

Location, Location, Generation: Fine-Tuning a VLM for Real Estate Descriptions

Stanford CS224N Custom Project

Carey Chang

Department of Computer Science
Stanford University
ckchang@stanford.edu

Niko Ustin

Department of Computer Science
Stanford University
nikou@stanford.edu

Abstract

Home descriptions in real estate listings play a significant role in the successful sale of a property. However, high-quality descriptions require a particular real-estate-specific tone, syntax, and content. As a result, generic VLMs often produce subpar, vague home descriptions. We tested whether a pre-trained VLM (LLaVA-OneVision-7b), fine-tuned using LoRA on home descriptions, home images, and key home details, generates better home descriptions than the same pre-trained VLM without any fine-tuning. We found that while there was high hallucination rates for both models, the fine-tuned VLM had more similar coverage rates of key provided home details and ultimately generated descriptions that were preferred for the majority of listings (79.9%) in the LLM-as-a-judge buyer preference evaluation.

1 Key Information

- Mentor: Tristan Thrush
- Late Days:

Team member	Late days remaining
Niko Ustin	2
Carey Chang	0
Total pooled	2
Team size	2
Late days applied for the report	1

2 Introduction

Real estate listing descriptions are the perfect application for multimodal generation: given a set of property photos and basic home attributes, the goal is to produce a concise, persuasive description that is both visually grounded and factually consistent. This problem is interesting and important because listing text directly affects how buyers perceive a home and how quickly it attracts attention, yet most listings are still written manually or with heavily templated tools. Automating this process could reduce real estate agent workload and standardize quality, but only if the generated descriptions match the distinctive voice, diction, and content used in practice.

Despite recent progress in VLMs, real-estate description generation remains difficult for several reasons. First, the task requires visual grounding across multiple images: a description should reflect salient features (layout, finishes, outdoor space, views) while avoiding incorrect claims. Prompting a general VLM with a few images and minimal context tends to produce text that is fluent but stylistically generic, inconsistent across listings, or insufficiently tailored to the conventions of

real-estate marketing. Second, the task requires structured factual control: models may omit key facts that buyers expect (beds/baths/sqft) or hallucinate attributes not supported by the prompt or photos. Finally, the data itself is heterogeneous: photos vary widely in quality, coverage, and ordering, and the “ground-truth” descriptions reflect subjective marketing choices rather than a purely objective captioning target.

This problem has not been fully solved largely because prior work in multimodal generation often optimizes for broad capabilities or benchmark tasks rather than domain-specific style and constraints. Generic pretrained VLMs are trained on diverse web-scale corpora and therefore learn broad visual–text pairs, but they do not internalize the consistent rhetorical structure, tone, and lexical choices that agents use to market homes. Traditional template-based systems can enforce factual consistency but struggle to incorporate visual nuance and produce natural, non-repetitive wording. Domain adaptation is challenging because end-to-end fine-tuning of large VLMs is expensive and risks degrading general multimodal understanding, while small datasets can encourage overfitting or surface level style mimicry without improving visual grounding.

To address these gaps, our approach adapts a strong pretrained VLM to the real-estate domain using parameter-efficient fine-tuning on a curated dataset of listings paired with multiple photos and human-written descriptions. We represent each example in an instruction-following, chat-style format that explicitly encodes both the desired writing persona and the property’s structured facts alongside the images, enabling supervised learning of domain-appropriate style while maintaining a clear conditioning signal for factual content. We then fine-tune only a small subset of parameters (via lightweight adapters) to improve the model’s ability to produce agent-style language and to better connect visual signals and structured attributes to the generated text, while keeping most of the pretrained model fixed for stability and efficiency. Key limitations remain: the model can still overgeneralize common marketing phrases, performance depends on photo quality and coverage, and purely automatic evaluations may not fully capture persuasive quality or subtle grounding errors, motivating added human analysis.

3 Related Work

Recent work in vision–language models (VLMs) has focused on combining pretrained vision encoders with large language models to enable multimodal reasoning and generation. The core idea is straightforward: take a model that already understands images (a vision encoder) and a model that already understands and generates text (an LLM), and find a way to connect them so the language model can have vision capabilities. This allows the resulting VLM to benefit from the advanced capabilities of SOTA pretrained LLMs and vision encoders while not having to train a complete VLM from scratch.

Earlier approaches such as CLIP learned strong image–text representations using contrastive training by pushing image and text embeddings closer together when they describe the same content, and farther apart when they don’t [1]. This created powerful representations; however, it was much more useful for determining whether an image and caption match rather than for text generation. Transformer-based fusion models like ViLBERT, UNITER, and ALBEF took a different approach, using cross-modal attention layers to let vision and language tokens attend to each other, enabling tasks that require grounded reasoning across both modalities [2, 3, 4].

More recent multimodal LLM systems extend this line of work by connecting vision encoders to language models through lightweight connectors, allowing a single generative interface for tasks such as captioning, visual question answering, and multimodal dialogue. The connector’s job is to translate the vision encoder’s output into a sequence of token embeddings that the language model can interpret. Models such as Flamingo, BLIP-2, and InstructBLIP explore different connector strategies, including cross-attention modules and query-based transformers [5, 6, 7], while systems like MiniGPT-4 and LLaVA show that relatively simple MLP projection layers combined with instruction tuning can produce strong multimodal results [8, 9]. LLaVA in particular is a common baseline for multimodal generation tasks due to its simplicity and strong performance.

However, prior work also highlights several limitations of generative VLMs, including hallucinated visual attributes and object inconsistencies with target images [10], sensitivity to prompts and instruction data, and weaknesses in tasks requiring precise quantities or grounded reasoning [11]. For our

task of generating real estate listing descriptions from property photos, these limitations are especially relevant, since a good description needs to accurately report specific details like bedroom count and square footage without fabricating information. These challenges motivate careful evaluation of VLM outputs and analysis of how training strategies affect generation quality. Our work builds on this literature by evaluating and comparing baseline and fine-tuned VLM behavior on a structured generation task, focusing specifically on factual coverage and hallucination in generated descriptions, as well as semantic quality.

4 Approach

We frame real estate description generation as a vision-language task: given a set of property images and structured metadata, generate a natural-language listing description. Each training sample is a chat-formatted conversation with a system prompt establishing a professional real estate writer persona, an input containing 6 property images and a text prompt with structured details (city, beds, baths, sqft, year built), and a target output containing the ground-truth listing description from Redfin.

We ingested sold listings from Redfin via RapidAPI, collecting images, metadata, and original listing descriptions. Images were downloaded and validated (valid JPEG, minimum 300x300px, >10KB), filtering out listings with fewer than 6 valid photos, as those typically did not contain comprehensive photos of both the interior and exterior of the house. However, listings often had 10-20 images, but passing all these images to the VLM would be prohibitively expensive. We decided to only pass in 6 photos per listing. To choose the 6, we encoded all images for each listing into 512-dimensional feature vector image embeddings using CLIP ViT-B/32 [1]. We then applied K-means clustering ($k=6$) with the standard k-means++ centroid initialization over the CLIP embeddings and selected the image nearest each cluster centroid. This was in hopes of ensuring visual diversity across the selected images.

We selected LLaVA-OneVision-7B [12] for its multi-image capabilities. Many VLMs that we looked into only take a single image, whereas LLaVA-OneVision was trained on single-image, multi-image, and video data jointly, making it well-suited for our task, where the model must reason across 6 property photos simultaneously. At 7B parameters, it was also small enough to fine-tune with LoRA given our compute constraints. Our baseline evaluates the unfinetuned model with no task-specific training. The model received the same system prompt, 6 images, and property details used in training, and generated a description. More specifics are discussed in Experimental details in the Experiments section. LLaVA-OneVision is built on the Qwen2 language model and the SigLIP vision encoder. We used the HuggingFace Transformers implementation [13] (llava-hf/llava-onevision-qwen2-7b-ov-hf).

5 Experiments

5.1 Data

We used real estate listing data from Redfin via RapidAPI to pull listing information from recently sold properties across 13 regions in the Bay Area. We collected the listing images and description, as well as some variables, namely square footage, number of beds, number of baths, year built, and city. In total, our data include 6,084 home listings and their 36,504 selected images. We stored the listing information and images in a Google Cloud Services bucket, caching the data in a persistent location in Modal during our first training run to decrease training time. We reserved 10% of the data set for testing, 10% for validation, and the remaining 80% for training. The task provided to the VLM is as described above in the Approach section.

5.2 Evaluation method

We evaluate generated descriptions across five complementary metrics spanning factual accuracy, informativeness, fluency, lexical similarity, and buyer appeal. For the first three evaluations, we use a VLM (Gemini 2.5 Flash) to extract and judge properties of the generated descriptions.

Factual Accuracy (Hallucination Rate): For each generated description, we prompt Gemini to determine whether it states a value for the number of bedrooms, number of bathrooms, year built,

or square footage that contradicts the ground-truth property metadata. We report the per-field hallucination rate as the fraction of listings where the generated description asserts an incorrect value for that field.

Coverage: We prompt Gemini to assess whether the generated description mentions each of the four key property details: bedrooms, bathrooms, year built, and square footage. We report coverage as the fraction of listings where each field appears in the description. A model that hallucinates less but also covers fewer details may simply be omitting information rather than accurately conveying it, so coverage and hallucination rate are best interpreted together.

Buyer Preference (Pairwise LLM Judge): We prompt Gemini to adopt the perspective of a prospective home buyer and choose which of two descriptions, from the baseline model and the fine-tuned model, would make them more likely to engage with the listing (e.g., save it, schedule a tour). To control for position bias, we randomly shuffle the presentation order of the two descriptions. We report win rates for each model and the tie rate.

CIDeR-D: We compute CIDeR-D to measure the lexical similarity between each generated description and the real estate agent’s original description. CIDeR-D uses TF-IDF weighted n-gram overlap, giving higher weight to informative n-grams that are distinctive to a given reference while down-weighting common phrases. Higher scores indicate closer alignment with the reference description’s specific word choices and phrasing. We chose CIDeR-D over the generic CIDeR in order to penalize descriptions whose length deviates significantly from the reference length.

Perplexity (GPT-2): We compute perplexity under a pretrained GPT-2 language model as a reference-free measure of fluency. Perplexity reflects how natural and predictable the generated text is to a general-purpose language model; lower perplexity indicates more fluent, natural-sounding prose. We report perplexity for the generated descriptions as well as for the human-written reference descriptions to establish a fluency ceiling.

5.3 Experimental details

We fine-tuned LLaVA-OneVision-7B using a QLoRA/LoRA approach, which allows us to adapt the model by training a small number of additional parameters while keeping the pretrained backbone largely fixed. More specifically, we loaded the base weights in 4-bit quantization and inserted low-rank adapters into the language model’s projection layers in both the self-attention blocks (query/key/value/output projections) and the feed-forward blocks (gating, up-projection, and down-projection). We also adapted the multimodal projection MLP that bridges the vision encoder’s pooled features into the language model’s hidden space. The vision encoder itself stayed frozen; we only trained the LoRA parameters and their associated adapter components. This keeps the memory footprint manageable, makes optimization more stable, and helps avoid catastrophic forgetting of the general visual knowledge the model already has, while allowing it to generate more precise descriptions for this task.

For training data, we used instruction-style examples: each one has a fixed system instruction, then a user turn that combines multiple images with a short text prompt encoding structured property attributes. The training objective is standard next-token prediction on the assistant’s description, with the loss computed only over the assistant portion of the sequence (prompt tokens masked). We trained on a single GPU (NVIDIA A100 on Modal) with gradient accumulation to achieve a larger effective batch size. We ran a single epoch and saved only the learned adapter weights (plus the processor/tokenizer artifacts) for lightweight deployment on top of the unchanged base model. For inference, we also used a single NVIDIA A100 on Modal, generating outputs with greedy decoding and a maximum token limit of 512.

5.4 Results

We ran all of the evaluations, as described in the Evaluation Metrics section, on the real estate agent descriptions from the original data set, the pre-trained VLM with no fine-tuning (LLaVA-OneVision), and our LoRA fine-tuned VLM. Note that CIDeR-D was calculated for the baseline VLM and the fine-tuned VLM relative to the real estate agent’s description, so there is no CIDeR-D metric for the real estate agent descriptions. For the buyer preference metric, we asked the VLM to choose between the baseline VLM and the fine-tuned VLM, so there is no metric for the real estate agent descriptions.

Table 1: Comparison of Reference, Baseline, and Fine-Tuned VLM Listing Descriptions

Metric	Real Estate Agent	Baseline VLM	Fine-Tuned VLM
Mean hallucinated fields (out of 4)	0.301	0.195	0.707
Mean coverage score (0–1)	0.478	0.831	0.758
CIDEr-D	—	0.0151	0.0178
Perplexity	57.92	14.49	14.15
<i>Hallucination rate by field</i>			
Beds	0.034	0.035	0.030
Baths	0.062	0.031	0.033
Year built	0.007	0.000	0.000
Sqft	0.198	0.129	0.644
<i>Coverage rate by field</i>			
Beds	0.817	0.956	0.911
Baths	0.757	0.925	0.870
Year built	0.132	0.661	0.544
Sqft	0.210	0.782	0.717
<i>Buyer preference (Gemini)</i>			
Preferred (%)	—	19.9	79.9

The fine-tuned VLM hallucinated the most fields, with a mean hallucinated fields of 0.707 (out of 4: beds, baths, year built, and square footage), followed by the real estate agents with 0.301, and then the baseline with 0.195. The existence of hallucinations was expected because, as discussed in the Related Work section, this is a common challenge with VLMs. The high hallucination rates for the fine-tuned VLM and the real estate agents was an unexpected result. The baseline had the highest mean coverage score with 0.831, followed by the fine-tuned VLM with 0.758, and then the real estate agents with 0.478. This result makes sense given that real estate agents may choose not to include certain fields in a home description and the fine-tuned VLM is more real estate agent-like than the baseline. The CIDEr-D score for the baseline and fine-tuned VLMs are similar, with the fine-tuned VLM having a slightly higher score of 0.0178 relative to 0.0151 for the baseline. The two VLMs had very similar perplexity as well, with 14.15 for the fine-tuned model and 14.49 for the baseline, both much lower than the real estate agent descriptions with 57.92. We expected a greater difference between the CIDEr-D scores and perplexity.

The hallucination rates for the beds, baths, and year built fields were very similar across all three sets of descriptions, with the exception of the real estate agent descriptions having a higher hallucination rate for the bathrooms field at 0.062, relative to 0.031 and 0.033 for the baseline and fine-tuned VLM, respectively. The hallucination rates differed more significantly for the square footage field, with 0.644 for the fine-tuned VLM, followed by 0.198 for the real estate agents, and the best 0.129 for the baseline. The higher hallucination rate for square footage was a surprise, in particular for the fine-tuned VLM. We believe that the increased appearance of hallucinations may be due to the model’s increased exposure to numerical values during fine-tuning. The training pairs contain specific numbers for beds, baths, sqft, and year built across hundreds of listings, and the model may struggle to associate the correct numbers with the correct property at inference time. Regarding the hallucination among real estate agents, we believe this may be a limitation of our data, as Redfin sometimes provides faulty information for older or renovated listings.

For all four fields, the coverage rates were highest for the baseline, followed by the fine-tuned VLM, and then the real estate agents with the lowest. The starkest differences in coverage rates are for the year built field, with the baseline at 0.661, the fine-tuned VLM at 0.544, and the real estate agents at 0.132, as well as the square footage field with the baseline at 0.782, the fine-tuned VLM at 0.717, and the real estate agents at 0.210. As mentioned, the ordering of the coverage rates was predicted, however, the gap in coverage rates between the real estate agents and the fine-tuned VLM was not.

The VLM taking the perspective of a home buyer preferred the fine-tuned VLM descriptions for the vast majority of the listings (79.9%), with a preference for the baseline for 19.9% of the listings and a tie for 0.2%. This result was expected, as it was the main goal of trying to build a fine-tuned VLM for this task. This result is also likely the most significant, as the task is ultimately to convince potential buyers to engage further with the listing.

6 Analysis

We conducted a qualitative analysis of baseline and fine-tuned model outputs to understand when each system succeeded and failed, what the fine-tuned model learned from training, and where our automated evaluation metrics break down.

Furniture Descriptions vs. Listing Prose: The baseline model frequently describes staged furniture visible in photos rather than the property itself. The baseline mentions sofas in 99 descriptions, coffee tables in 85, and rugs in 29, none of which appear in any human-written reference listing. For example, one baseline output describes "a plush sofa and stylish chairs, complemented by a sleek coffee table and a decorative vase," where an agent would instead describe the room's dimensions, finishes, and layout. The fine-tuned model reduces but does not eliminate this behavior (78 sofas, 58 coffee tables), suggesting that exposure to agent-written listings helps the model begin to distinguish between captioning a photo and describing a property for sale.

Square Footage Errors: Through visual inspection, the baseline seemed to often truncate square footage values by dropping the leading digit: 1,712 sqft becomes 712. This truncation pattern likely stems from how the model tokenizes and generates multi-digit numbers. On the other hand, the mistakes made in square footage reporting by the finetuned model seem to be numbers that are more plausible, and do not fit into a specific systematic category of error. This may be because fine-tuning on listings where sqft appears consistently in natural sentences helps the model learn the general format and magnitude of square footage values, but in turn confuses the model with reproducing the exact number.

What the Fine-Tuned Model Learned: The fine-tuned model successfully learned to incorporate property metadata into natural sentences and to adopt the general register of listing prose. It uses real estate jargon with appropriate frequency: "open floor plan" appears in 176 fine-tuned descriptions vs. 109 baseline, and "natural light" in 390 vs. 348. However, overuse relative to references suggests the model learned that these phrases are common in listings without fully learning when to use them in context.

However, the model fails to learn terms requiring genuine visual grounding or local knowledge. "Turnkey" (31 real estate agent, 0 VLM), "move-in ready" (52 real estate agent, 0 VLM), and "chef's kitchen" (50 real estate agent, 0 VLM) never appear in either model's output. These terms require understanding property condition from images, which is beyond what our LoRA fine-tuning was able to achieve.

Understanding LLM-as-Judge The buyer-preference evaluation shows that the fine-tuned model was preferred 79.9% of the time, but this was often driven by the baseline's obvious failure modes: furniture descriptions and truncated square footage. As mentioned before, the finetuned model often generates more plausible metrics. We also evaluated the Gemini judge's reasoning for its decisions. In many of the fine-tuned model's wins, the stated rationale cites the baseline model's implausible square footage or generic staging language. However, these numeric errors should be treated as equal, which demonstrates a limitation in using LLM-as-Judge.

7 Conclusion

The goal of this project was to test whether a fine-tuned VLM could outperform a general, pre-trained VLM at generating home descriptions for real estate listings using home images and a few key home details. We found that while the hallucination rates for the provided home details were higher for the fine-tuned VLM, coverage rates were more similar to the real estate agent descriptions. More importantly, however, we found that the fine-tuned VLM was preferred for most listings (79.9%) over the baseline VLM on the buyer preference evaluation, which achieves the ultimate goal of

creating listing descriptions that motivate potential buyers to engage further with the listing. The high hallucination rates across both models speak to the challenges with hallucination that VLMs experience, as discussed in the Related Work section. Trying a larger training data set, improving data quality, and running more epochs are potential paths for future work that might reduce hallucination rates. One key limitation is that we tried a single fine-tuning approach on one baseline VLM. Other fine-tuning approaches and baseline VLMs might have produced different results. To scope the task, we selected 6 images from the full set of home images for each listing; therefore, another limitation is that the selected photos may not be representative of the home and likely miss visual home details. Additionally, certain important information that a real estate agent might include in a home description was not passed to the VLMs in these experiments (school district, neighborhood, etc.). One last limitation, is that several of the evaluations rely on an LLM-as-a-judge approach, which does not always produce reliable results, as many of the pitfalls of LLMs are inherited. Future work could include trying DPO on human comparison pairs, increasing the number of listings used for training, and increasing the number of images used for each listing.

Team contributions

The team started with a highly collaborative ideation phase, in which Carey and Niko worked together to research and discuss various project ideas. In the implementation phase, Carey put together the data collection and processing pipeline, set up and ran the baseline model on Modal, and put together some of the evaluations. Niko set up the Google Cloud bucket, put together some of the evaluations, and fine-tuned the model. Throughout the whole project, both Carey and Niko were heavily involved in discussing ideas about implementation.

References

- [1] 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. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021.
- [2] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32, 2019.
- [3] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *European conference on computer vision*, pages 104–120. Springer, 2020.
- [4] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705, 2021.
- [5] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.
- [6] Junnan Li, Dongxu Li, Caiming Xiong, and Steven C. H. Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023.
- [7] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in neural information processing systems*, 36:49250–49267, 2023.

- [8] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.
- [9] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.
- [10] Yifan Li, Yifan Du, Kun Zhou, Jinteng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 292–305, 2023.
- [11] Roni Paiss, Ariel Ephrat, Omer Tov, Shiran Zada, Inbar Mosseri, Michal Irani, and Tali Dekel. Teaching clip to count to ten. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3170–3180, 2023.
- [12] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.
- [13] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In Qun Liu and David Schlangen, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics.