# Meta-Personalizing Vision-Language Models to Find Named Instances in Video

# Supplementary Material

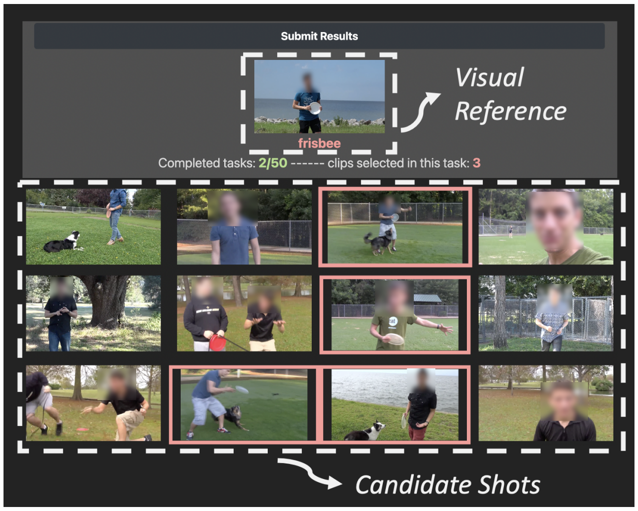

Figure 1. **Test-time Personalization Dataset annotation tool.** Our user interface contains two key components: (top) A video player that shows the visual reference of the target named instance. (bottom) A gallery of "clickable" candidate shots to be labeled as positives. The pink borders denote the selected positive samples, for the instance `"Zak's frisbee"`.

## Overview

In this supplementary, we first provide additional details of the algorithm for the automatic mining of named instances in videos (Section A). Then in Section B, we give additional details about the process of collecting annotations for our dataset. Section C provides additional implementation details of our approach and the evaluation metrics. Section D explores alternative approaches for mining instances, *i.e.*, using Part-of-Speech (POS) and Named Entity Recognition (NER). Section E discusses some limitations of our approach. Finally, Section F discusses additional qualitative results of personalized retrieval.

## A. Automatic Mining of Named Instances in Video

**Spotting Named Instances.** We provide more details here of how we spot named instances (Section 3.1 of the main paper). We keep up to four words after a possessive text pattern is matched based on text-visual similarity. Given a sequence of words $[q_1, \dots, q_4]$ we extract embeddings $f_l([q_1]), f_l([q_1, q_2]), \dots, f_l([q_1, \dots, q_4])$ with CLIP's text encoder. We then compute the cosine similarity with the visual reference embedding $f_v(s^*)$ and keep the longest sequence of words with cosine similarity greater than 0.3. This strategy allows us to find relatively clean instance names. For instance, let us suppose we string-match the candidate instance `this is my "dog waggy he is"`. Our approach would allow us to keep `dog waggy` as the instance name. We obtain the cleaned name given that additional words, `he is`, would yield a lower than 0.3 text-visual similarity.

**Filtering non-visual instances.** Regarding the filtering procedure for non-visual instances outlined in Section 3.1, we find that high visual-language similarity is observed when the candidate named instance features nouns or phrases that distinctly describe a visible object instance in the video. High visual-language similarity occurs for both general object categories (such as ”my car”) and more specific descriptions (”my 2018 Honda Civic”). Thresholds for spotting and filtering named instances were determined using cross-validation on a small curated validation set.

## B. *This-Is-My* Dataset

**Test-time Personalization Dataset $\mathcal{P}$.** We provide more details about our annotation tool (Section 4 of the main paper). Figure 1 (this supplemental) includes a screenshot of the annotation tool used to annotate the test-time personalization dataset. We implemented a simple user interface that shows the named instance (top) and a gallery of candidate shots (bottom). The interface auto-plays all candidate shots and allows the annotator to label the positive samples by clicking the video. Leveraging the interface, we are able to label the 1000 candidates for each instance in 20 minutes. Therefore, we spent around five hours annotating the 15 instances of the test-time personalization dataset.

## C. Additional Implementation Details

In all our experiments, we rely on the Adam optimizer [3] with a weight decay set to $10^{-5}$. The learning rate follows a cosine annealing schedule [5] with a maximum learning rate of 0.1. Next, we describe the implementation details for the two datasets.

**Baselines.** In the CLIP (language) baselines, we pass only the manually labeled object category (*e.g.*, ”dog”) to the text encoder, which prevents confusion caused by queries containing names of specific instances such as `"Zak's dog Coffee"` and `"My dog Biscuit"`.

**DeepFashion2 Experiments.** Each test-time training is performed for 40 epochs with a batch size of 512. In this setting, learning the 50 instance tokens takes about 10 minutes in total on a single GPU. We perform 10 rounds of meta-personalization for the pre-training of category features $C$, each round consisting of 32 pseudo instances per category (only a single training image per instance is available). Each training round consists of 10 epochs, and we use a batch size of 512. We identify 14 categories for DeepFashion2[1]. Our CLIP (language) baseline uses these categories in place of the learned instance tokens for retrieval.

***This-Is-My* Experiments.** Each test-time training is performed for 40 epochs with a batch size of 16, which takes less than two minutes on a single T4 GPU. We use 512 randomly chosen distractor shots at each training iteration during test-time personalization. Meta-personalization consists of 10 rounds of training, with each round lasting for 20 epochs. We use a batch size of 512 and do not include any distractor shots.

**Evaluation Metrics.** For completeness, we provide definitions for the retrieval metrics used in our experiments. Let $R_{ij}$ be indicators of whether the retrieved video shot for query $i$ at rank $j$ is a correct match, *i.e.*, $R_{ij} = 1$ if the $j$-th shot retrieved for query $i$ is showing the correct instance (in the right context), and $R_{ij} = 0$ otherwise. Let further $\text{rank}_i = \min\{j | R_{ij} = 1\}$. We then have

$$\text{MRR} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\text{rank}_i}, \tag{1}$$

$$\text{R@K} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}\{\text{rank}_i \leq K\}, \tag{2}$$

and

$$\text{mAP} = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} \frac{R_{ik}}{n_i} \, \text{P}_{ik}, \tag{3}$$

where $\text{P}_{ik} = \frac{1}{k} \sum_{j=1}^{k} R_{ij}$ is precision-at-$k$ for the $i$-th query, $n_i = \sum_{j=1}^{K} R_{ij}$ is the number of relevant shots for query $i$, $N$ is the number of queries and $K$ the number of shots in the retrieval dataset.

**Hyper-parameters.** We use COCO [4] classes as the object categories $l \in \mathcal{Y}$, since they represent common and general objects. We set the temperature parameter $\lambda = 0.1$ to a standard default value used in contrastive losses (*e.g.*, SimCLR [1]). We choose $\lambda_c = 0.5$ through cross-validation and find that our model's performance is robust with respect to different values of $\lambda_c$.

## D. Alternatives for Mining Instances

We explore alternative approaches for spotting named instances (Section 3.1 of the main paper). In our approach, we set the list of possessive text patterns for mining empirically. After annotating a small sample, we observed that those patterns yield a larger number and more precise set of named instances compared to other alternatives such as Part-of-Speech (POS) and Named Entity Recognition (NER) (see Table 1 in this supplemental). Interestingly, combining our approach for possessive text pattern matching with an additional NER filter can improve the precision of the mined instances. For simplicity, we do not apply the NER filter for our final collected dataset as described in the main paper.

## E. Discussion

We discuss potential limitations of our work and highlight key differences to prior work [2].

**Handling of multiple subjects featured in a video.** During data collection, we did not regulate the number of subjects featured in each clip; as a result, there could be instances where multiple subjects are presented. These multiple subjects may affect the precision of our mining approach, particularly if the subjects belong to the same visual category. Nevertheless, we have observed that specific objects are usually visually conspicuous when mentioned. For example, when the speaker mentions `This is my dog <Fido>`, a close-up or zoom-in shot of the dog are typically shown. In future work, the narration's contextual information (*e.g.*, `<Fido> eating`) could be utilized to differentiate instances with several subjects.

**Size of the *This-Is-My* test-time personalization dataset.** Our *This-Is-My* dataset contains a modest number of instances; however, the search space is very large (around 50K shots, including distractors). We also acknowledge that creating this dataset posed a considerable challenge, as it required identifying instances across numerous videos and annotating each shot per video.

**Differences to PALAVRA [2].** Our work distinguishes itself from [2] in three crucial aspects:

1. **Model:** [2] models instance tokens independently, whereas our method represents them as a weighted sum of shared category features learned through meta-personalization. As demonstrated in ablation (a) of

[1] List of DeepFashion2 categories: `['long sleeve dress', 'long sleeve top and skirt', 'long sleeve top and trousers', 'long sleeve top and vest dress', 'short sleeve top and shorts', 'short sleeve top and skirt', 'short sleeve top and sling dress', 'short sleeve top and trousers', 'shorts and vest', 'skirt and sling', 'skirt and vest', 'sling dress', 'trousers and vest', 'vest dress']`

Table 1. **Alternatives for mining instances.** We annotate 100 candidate instances for different mining approaches, including, Part-of-Speech (POS) and Named Entity Recognition (NER). We report the number of true named instances and the precision for each method. Possessives w/ NER filter denotes our mining approach combined with a filter that discards candidate instances without recognized entities. Combining possessives with POS yields much lower precision, thus not included in this table.

| | **w/o visual filter** | | **with visual filter** | |
|---|---|---|---|---|
| **Method** | # named instances | Precision | # named instances | Precision |
| POS (nouns) | 19 | 19.0% | 15 | 36.6% |
| NER | 21 | 21.0% | 17 | 38.3% |
| Possessives (ours) | 58 | 58.0% | **46** | 63.1% |
| Possessives w/ NER filter | 48 | 64.0% | 39 | **70.5%** |

Table 1 in the main paper, our design improves the model's generalization capabilities.

2. **Training Data:** Unlike [2], which requires a set of labeled examples per instance, we propose a method to mine training examples from narrated videos.

3. **Training Objective:** Our method proposes a contrastive training objective for test-time personalization, whereas [2] requires additional networks (see set encoder in [2])

***This-Is-My* vs. YTVOS [6] for personal video instance retrieval.** In contrast to [2], which used YTVOS by taking query and retrieval frames from the same video, we explore a more challenging scenario where query and retrieval shots are from different videos, showing the instance in different contexts.

## F. Qualitative Results for Contextualized Instance Retrieval

Figure 2 provides additional qualitative results for the contextualized instance retrieval task on the *This-Is-My* dataset. It compares the Top-5 retrievals of our approach and the CLIP (language) baseline given a language query. Both methods can successfully retrieve shots that match the generic context of the query, *e.g*., `eating food with a white plate` (third row). However, the baseline fails at retrieving the correct personalized instance, *e.g*., `"Zak's dog Coffee"` (third row). In the example of `Casey's son is standing at the beach without wearing shirt` (last row), the CLIP (language) baseline fails to find the personalized instance since it does not have a representation for the instance `"Casey's son"`. Contrarily, our model finds the correct named instance in the right context. This result is due to our model expanding the input space of the VLM by personalizing a representation of the learned instance while maintaining the general abilities of the underlying VLM.

Figure 3 shows successful and failure cases of our model on the contextualized instance retrieval task. (top) We observe that our method correctly retrieves the personalized instances even in challenging scenarios. For instance, it can retrieve small instances such as `"Casey's boosted board"` and `"Blippi's shows"` for different context queries. By keeping the VLM frozen, our method preserves the original VLM's capabilities to match natural language queries to the candidate set of shots. (bottom) While our method significantly improves the state-of-the-art in personalized retrieval, we observe some common failure cases. One typical example is discriminating between instances that are too similar. For instance, `"Sherry's road bike"` is confused with another black bike. Our method is also limited by the VLM's capabilities to understand actions such as `grabbing Sherry's road bike`. The former failure case could be addressed by leveraging additional cues from the transcript and the latter by leveraging progress on motion-aware VLMs.

## References

[1] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 2

[2] Niv Cohen, Rinon Gal, Eli A Meirom, Gal Chechik, and Yuval Atzmon. “This is my unicorn, Fluffy”: Personalizing frozen vision-language representations. In *ECCV*, 2022. 2, 3

[3] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015. 1

[4] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 2

[5] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 1

[6] Ning Xu, Linjie Yang, Yuchen Fan, Jianchao Yang, Dingcheng Yue, Yuchen Liang, Brian Price, Scott Cohen, and Thomas Huang. Youtube-vos: Sequence-to-sequence video object segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 585–601, 2018. 3

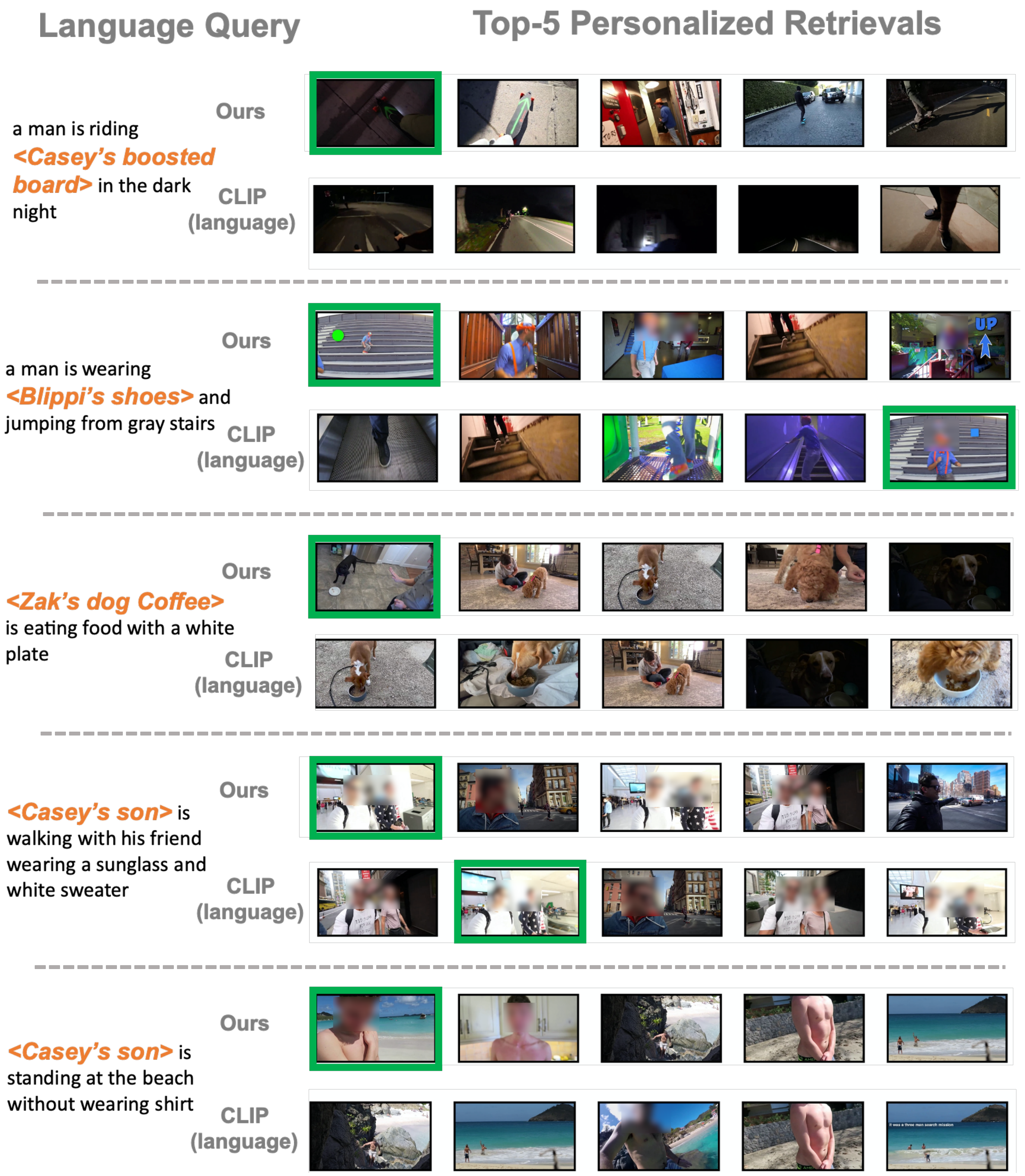
Language Query
Top-5 Personalized Retrievals
a man is riding <Casey's boosted board> in the dark night
Ours
CLIP (language)
a man is wearing <Blippi's shoes> and jumping from gray stairs
Ours
UP
CLIP (language)
<Zak's dog Coffee> is eating food with a white plate
Ours
CLIP (language)
<Casey's son> is walking with his friend wearing a sunglass and white sweater
Ours
CLIP (language)
<Casey's son> is standing at the beach without wearing shirt
Ours
CLIP (language)

Figure 2. **Qualitative Retrieval Comparison to the CLIP (language) Baseline.** While the baseline is able to accurately match the features in the scene that match the described context, it fails to retrieve the correct instance. In contrast, our personalized VLM successfully matches both context and personalized instance. Search prompts are shown on the left and correct retrievals are highlighted in green.

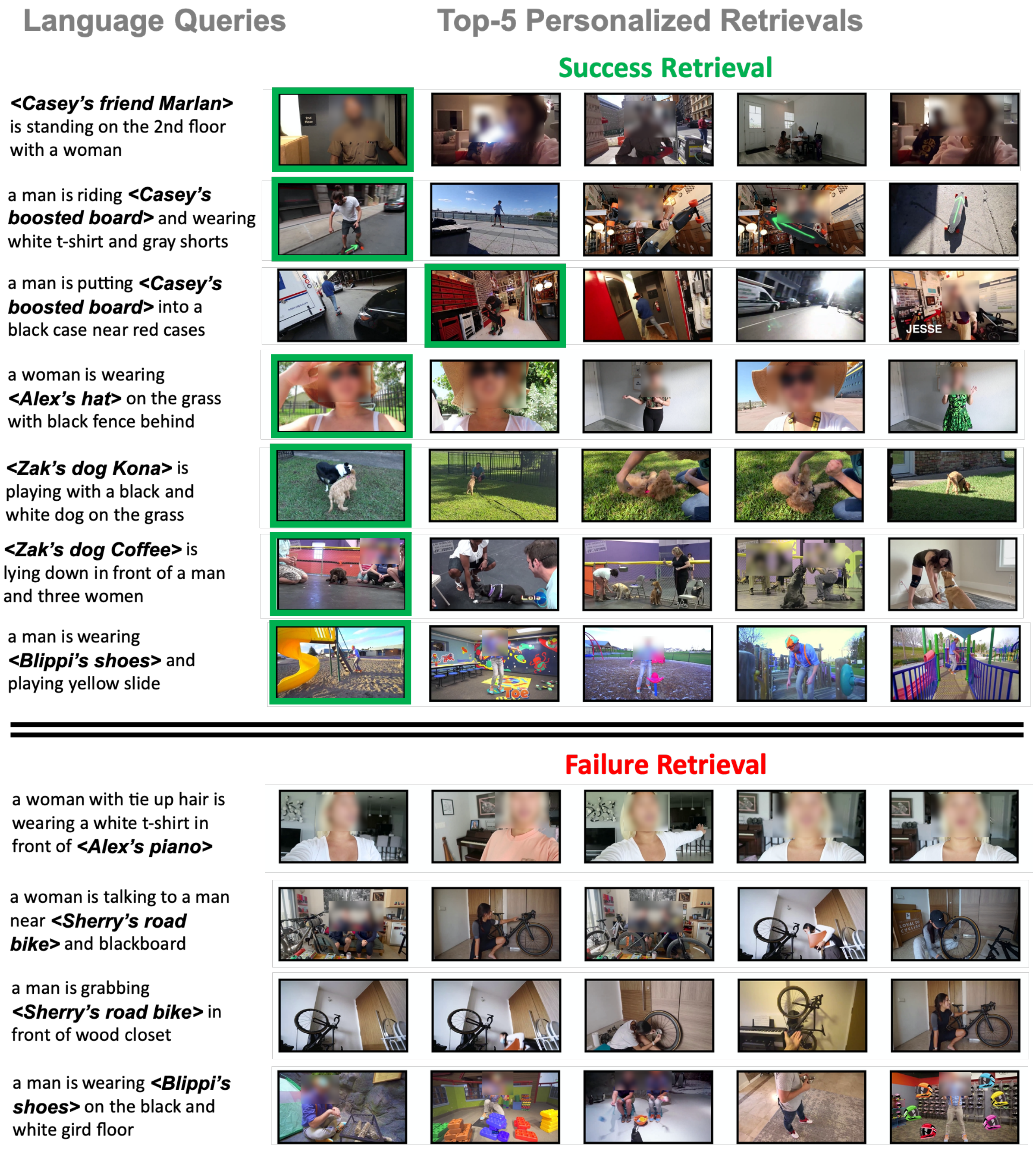
Language Queries
Top-5 Personalized Retrievals
Success Retrieval
<Casey's friend Marlan> is standing on the 2nd floor with a woman
a man is riding <Casey's boosted board> and wearing white t-shirt and gray shorts
a man is putting <Casey's boosted board> into a black case near red cases
JESSE
a woman is wearing <Alex's hat> on the grass with black fence behind
<Zak's dog Kona> is playing with a black and white dog on the grass
<Zak's dog Coffee> is lying down in front of a man and three women
a man is wearing <Blippi's shoes> and playing yellow slide
Failure Retrieval
a woman with tie up hair is wearing a white t-shirt in front of <Alex's piano>
a woman is talking to a man near <Sherry's road bike> and blackboard
a man is grabbing <Sherry's road bike> in front of wood closet
a man is wearing <Blippi's shoes> on the black and white gird floor

Figure 3. **Qualitative examples of retrieval using our personalization approach.** Top: Successful examples where the correct instance is retrieved within the top five retrieved shots. Bottom: Examples of failures where the correct instances is not retrieved within the top five retrieved shots.