Appendix

A Implementation details

Number of training samples: The results reported in the main text used 5 training samples for each concept in the retrieval experiments, and 10 training samples for each concept in the segmentation experiments. Below, in Sec. B, we provide additional results that sweep over the number of training samples.

Cycle loss prompts: We use multiple prompts for querying the concept with the cycle loss. In each epoch, we selected a template at random from the following list of prompts.
"This is a photo of a [CONCEPT]", "This photo contains a [CONCEPT]", "A photo of a [CONCEPT]", "This is an illustrations of a [CONCEPT]", "This illustrations contains a [CONCEPT]", "An illustrations of a [CONCEPT]", "This is a sketch of a [CONCEPT]", "This sketch contains a [CONCEPT]", "A sketch of a [CONCEPT]", "This is a diagram of a [CONCEPT]", "This diagram contains a [CONCEPT]", "A diagram of a [CONCEPT]", "A [CONCEPT]", "We see a [CONCEPT]", "[CONCEPT]", "We see a [CONCEPT] in this photo", "We see a [CONCEPT] in this image", "We see a [CONCEPT] in this illustration", "We see a [CONCEPT] photo", "We see a [CONCEPT] image", "We see a [CONCEPT] illustration", "[CONCEPT] photo", "[CONCEPT] image", "[CONCEPT] illustration".

Contrastive loss: We apply all contrastive losses with a temperature hyperparameter (denoted by $Temp = 0.25$), dividing each cosine similarity in Eq.1,2. The value of $Temp$ was selected using a validation set (details about hyper-parameter search below).

Ground-truth regularization for training the set encoder f_θ: For training the set encoder f_θ, we use a regularization term that maximizes the cosine similarity of the predicted word embedding $\mathbf{w}_c^0$, with its ground truth word embedding $\mathbf{g}_c$, keeping $\mathbf{w}_c^0$ close to its ground truth value. Namely, $\ell_{GT}(\mathbf{w}_c^0, \mathbf{g}_c) = -\cos(\mathbf{w}_c^0, \mathbf{g}_c)$, where $\mathbf{g}_c$ is the word embedding of the concept type (e.g. the embedding of "dog"). If the concept type includes more than a single word, we take the first one.

Architecture: When using CLIP, we always used ViT-B/32 Vision Transformer.

Normalized embeddings: Wherever we use a textual or visual encoder output, we first normalize the embedding vector to unit norm. The embedding can be viewed as lying on a hypersphere.

Training the alignment matrix A with images and captions: In addition to updating the alignment matrix $\mathbf{A}$ during training of f_θ with text (Section 4.2), we also update $\mathbf{A}$ by mapping from captions to images. Specifically, for every image I embedded with CLIP $v = h^{\mathcal{I}}(I)$, we took the respective caption S embedded with $u = h^{\mathcal{T}}(S)$, and trained $\mathbf{A}$ to project from the embedded captions to the embedded images by minimizing an L_2 loss:

$$\ell\big(I, S, \mathbf{A}\big) = ||h^{\mathcal{I}}(I) - \mathbf{A}h^{\mathcal{T}}(S)||^2 \tag{A.3}$$

Using the alignment matrix A for the fine-tuning stage: In practice, in the fine tuning stage (Eq. (2)), we replace η_c by $\mathbf{A} \cdot \eta_c$.

Training procedure for the set encoder f_θ: We train f_θ in an alternating fashion. One batch with COCO images and one batch with augmented COCO captions.

Hyperparameters: Hyper parameters were tuned one at a time, on a validation set, to maximize the MRR metric in the retrieval task.

We train f_θ for 300 epochs. Batch size was set to 256. We used the Adam [33] optimizer with a learning rate of 0.0001 for both the cycle loss and the alignment loss (eq. A.3). DeepSet's hidden dimension was set to 4096, and the dropout rate was set to 0.25. The contrastive loss temperature was set to $Temp = 0.25$. To optimize the word embeddings, we used 30 epochs, with a learning rate of 0.01. The weight of the ground-truth regularization term λ_{gt} was set to 512.

We used the following ranges to search for hyper parameters: (1) number of epochs $\in [100, 200, 300, 500, 1000]$ (2) batch size $\in [128, 256, 512]$ (3) learning rate $\in [0.01, 0.001, 0.0001, 0.00001]$ (4) DeepSet's hidden dimension $\in [512, 1024, 2048, 4096]$ (5) dropout rate $\in [0.15, 0.25, 0.35, 0.5]$ (6) $Temp \in [0.15, 0.25, 0.35, 0.5]$ (7) number of fine-tuning personalization epochs $\in [10, 20, 30, 40, 50, 60]$ (8) fine-tuning learning rate $\in [0.01, 0.001, 0.0001]$ (9) $\lambda_{gt} \in [1, 2, 4, 8, \ldots 2048]$.

Randomization: *Model:* For each of our 5 repetitions we trained a new f_θ model. *Few-shot training data:* When selecting a subset of few images from the few-shot training data, we made sure that the random seed (and subsets) are consistent between PALAVRA and the baselines (e.g. COLLIE, AvgIm, etc ...).

Training COLLIE and Adapter: To use multiple concepts that share the same concept type (category name), with the COLLIE and adapter baselines, we have assigned a unique [CONCEPT] phrase for concept. The phrase is composed of its class (e.g., skirt) and a unique ID number (e.g., "skirt 241").

B Additional Results

Accuracy versus number of shots: Fig. A.1 shows the performance of our model and the baselines as a function of the number of few-shot training samples used to learn each personalized concept. DeepFashion2 performance improves as the number

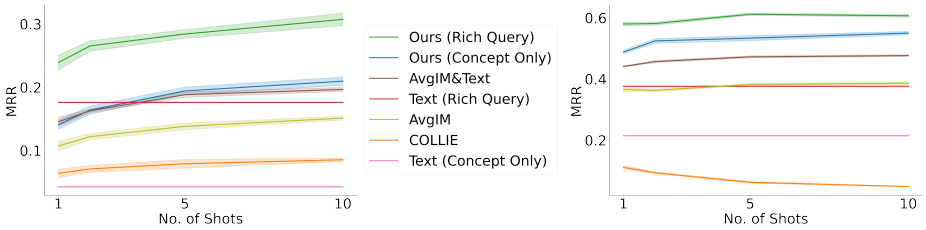

Fig. A.1. MRR for image retrieval on DeepFashion2 (left) and YTVOS (right) as function of the number of shots used to learn each personalized concept. DeepFashion2 performance improves as we increase the number of shots. YTVOS saturates early, probably because the training images of each concept have less variability, since they were all extracted from the same video.

of shots increases. YTVOS saturates early, probably because the training images of each concept have less variability since they were all extracted from the same video.

Short versus detailed captions: Table A.1 shows retrieval results on DeepFashion2 when using longer, detailed captions. As expected, all text-based methods demonstrate an increase in retrieval metrics across the board, indicating that they can successfully leverage additional information. Our method remains at the front even in this scenario, highlighting that the benefits of personalized concepts persist even when detailed descriptions are provided.

	MRR	Recall@5	Recall@10
PALAVRA (Ours)	**33.8 ± 0.5%**	**47.5 ± 0.9%**	**61.9 ± 0.9%**
PALAVRA w.o. tuning	27.8 ± 0.3%	36.4 ± 0.6%	48.4 ± 0.6%
AvgIM+Text	20.9 ± 0.6%	29.0 ± 0.7%	38.4 ± 0.6%
Text (CLIP)	24.3 ± 0.0%	31.7 ± 0.0%	43.4 ± 0.0%

Table A.1. Retrieval results using detailed captions. As expected, all compared methods show improved performance when provided with extra textual information. Notably, our method maintains the advantage even in such a scenario, showing that it can yield an increase in performance even when the concepts are described in detail.

In Sec. F.2 below we explain the data collection procedure of the "detailed" and "short" queries.

COLLIE sensitivity to prompt:

In Sec. 6.1 we demonstrated that COLLIE performance degrades when using rich textual queries. Here we describe results showing that COLLIE is even sensitive to much simpler queries. Namely, template queries that only add a *prefix prompt*.

When the text query includes only the [CONCEPT] tag, as in COLLIE's training procedure, COLLIE achieves a 13.4% average MRR score on DeepFashion2 retrieval

test set. When the query text is a sentence with a prefix, its score drops sharply. For example, the query "This is a photo of a [CONCEPT]" results in an MRR score of 5.3%, the query "This looks like a [CONCEPT]" score is 4.6%, and the query "In this image, there is a [CONCEPT]" yields 3.3%.

Qualitative results for semantic segmentation: In Fig. A.2 we show curated qualitative segmentation results. We observe that our model can successfully segment the correct object instance even in scenarios with visually similar distractors. On the other hand, our model can sometimes fail to distinguish between multiple relevant candidates, or segment other objects which exist near the target. However, this last limitation may be an artifact of the underlying segmentation method.

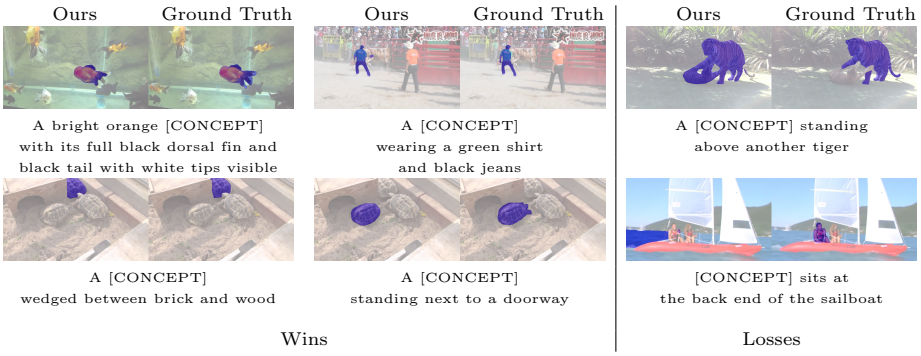

Fig. A.2. Qualitative examples of PALAVRA used for semantic segmentation. **Left and middle:** successful segmentations, where the correct specific object is identified and extracted from the image despite similar distractors (other turtles). **Right:** Typical failure cases: distractors are segmented along with the personalized object (top), or the textual descriptions draw CLIP's attention away from the main object (bottom).

C Ablation Study

To understand how various components of our approach contribute to its performance, we conducted an ablation study. We report validation and test metrics for DeepFashion2 and YTVOS.

We first ablate model components that affect training f_θ. We report the results without fine-tuning, to reveal how they affect the training of the set encoder. We call this stage **"no tuning"**.

Then we compare components that affect the fine-tuning stage. We call this stage **"with tuning"**. Specifically, we compare the following components:

1. **PALAVRA** is our approach described in Sec. 4. We tested it both **with tuning** and **no tuning**.
2. **no text augment** shows the results of f_θ trained only with visual concepts that exist in the MS-COCO vocabulary, and without using the extended vocabulary for augmentation.

3. **Only** $\ell_{\mathbf{GT}}$ does not use the cycle loss for training f_θ (see Eq. 1).
4. **Only** $\ell_{\mathbf{Cycle}}$ does not use the ground-truth regularization term for training f_θ (see Eq. 1).
5. **Only tuning** initializes $\mathbf{w}_c^0$ randomly, instead of using the prediction made by the set encoder f_θ.
6. **no alignment** shows the performance of our method when replacing the alignment matrix $\mathbf{A}$ by an identity mapping.

Table A.2 shows the results of the ablation experiments. Several points worth discussing. First, PALAVRA without tuning improves by 25% compared to "Text Only" (in Table 1), for both DeepFashion2 (22.1 vs. 17.6) and YTVOS (47.1 vs. 37.6). This result indicates that f_θ learns to predict the word embeddings of visual concepts, and these concepts are better than using their vanilla CLIP embeddings.

Next, we find that text augmentation with extended vocabulary (Sec. 4.1) improves concept learning with f_θ. It yields an improvement of ∼16% for DeepFashion2 (22.1 vs. 19.1) and ∼6% for YTVOS (47.1 vs. 44.4).

Combining a cycle loss with the ground truth (GT) regularization term is effective. When combined with the GT regularization term, the cycle loss improves by ∼16% for DeepFashion2 (22.1 vs 19.2) ∼14% for YTVOS (47.1 vs 41.4). However, when the GT regularization term is deactivated and only the cycle loss is used, f_θ fails to generalize (16.1 in DeepFashion2 and 37.3 in YTVOS). We hypothesize that this effect is similar to the effect observed with inversion to the latent space of GANs [63]. There, inversions into sparse regions of the latent space can better satisfy a cyclic reconstruction loss, but they behave poorly under interpolation. Our f_θ could similarly learn to invert into sparse regions of CLIP's input space. By adding the GT regularization term, our inversions are encouraged to reside in better-behaved regions of the input space, namely those observed during CLIP's training. In these regions, the semantics of the latent space hold better and the model can better generalize.

When f_θ is replaced by a random initialization, the performance degrades by ∼6% for YTVOS (57.1 vs 61.2) and ∼3% for DeepFashion2 (27.5 vs 28.4). Showing the synergy between the two personalization steps.

Finally, integrating the alignment matrix $\mathbf{A}$ showed an improvement of ∼8% for DeepFashion2 (28.4 vs. 26.3) and ∼5% for YTVOS (61.2 vs. 58.1).

D Personalization of Other Vision & Language

Vision & Language models other than CLIP may also benefit from an extended vocabulary of personalized concepts. It is likely that a similar approach to ours can still be applied. For example, in models like M-DETR [31], f_θ could map from the CNN output space to the *input* space of RoBERTa. The alignment matrix $\mathbf{A}$ can close the cycle, mapping from the *output* of RoBERTa to the CNN output.

E Segmentation Details and Analysis

E.1 Baselines and hyper parameters

Our segmentation experiments use the framework of Zabari et al. [74]. The method leverages transformer interpretability methods to identify image regions that relate to

DeepFashion2	Validation		Test	
	MRR	Recall@5	MRR	Recall@5
no tuning				
PALAVRA (Ours)	**26.9 ± 0.2**	**35.9 ± 0.2**	**22.1 ± 0.2**	**29.6 ± 0.3**
no text augment	21.8 ± 1.9	29.2 ± 2.3	19.1 ± 0.2	25.7 ± 0.3
Only ℓ_{GT}	23.3 ± 0.4	31.9 ± 0.5	19.2 ± 0.5	25.1 ± 0.8
Only ℓ_{Cycle}	19.3 ± 0.5	26.8 ± 0.8	16.1 ± 0.4	21.6 ± 0.8
with tuning				
PALAVRA (Ours)	**36.2 ± 1.3**	**53.7 ± 2.0**	**28.4 ± 0.7**	**39.2 ± 1.3**
Only tuning	32.1 ± 0.6	44.1 ± 0.7	27.5 ± 1.0	37.9 ± 1.8
no alignment	32.9 ± 0.4	47.8 ± 1.1	26.3 ± 0.2	36.9 ± 1.6
YTVOS	Validation		Test	
	MRR	Recall@5	MRR	Recall@5
no tuning				
PALAVRA (Ours)	**47.3 ± 0.5**	**68.5 ± 0.5**	**47.1 ± 0.8**	**70.3 ± 0.8**
no text augment	45.0 ± 0.3	63.8 ± 0.5	44.4 ± 0.3	65.6 ± 0.4
Only ℓ_{GT}	40.8 ± 0.8	59.3 ± 1.3	41.4 ± 0.2	62.0 ± 0.1
Only ℓ_{Cycle}	35.5 ± 1.1	50.8 ± 2.4	37.3 ± 1.1	55.8 ± 2.1
with tuning				
PALAVRA (Ours)	**59.0 ± 0.8**	**76.2 ± 1.1**	**61.2 ± 0.4**	**78.7 ± 0.4**
Only tuning	57.3 ± 0.9	76.1 ± 0.9	57.8 ± 0.3	77.1 ± 0.8
no alignment	56.5 ± 0.7	74.1 ± 0.3	58.1 ± 0.3	75.2 ± 0.9

Table A.2. Ablation study results, highlighting the importance of various framework components. See the text for a full description of each setting and an analysis of the results.

a given textual prompt. In this process, the text-encoding branch is only used to supply an embedding vector which is matched with the image branch. As such, the embedding vector can be easily replaced with another vector from any source. We leverage this property for all of our baselines.

When conducting an image-based search (AvgIM), we replace the embedding vector with the normalized average embedding of a small set of images depicting the target object. For the AvgIM&Text baseline, we further average this image embedding with the text embedding of the query text.

To compare with COLLIE, we generate the embeddings using their adapter setup. For our method, we utilize the original CLIP text encoder but substitute our learned input word embeddings for the concept token.

Hyper parameters were tuned on the validation set and kept fixed for all methods. We use a resizing factor of 0.5 and generate 3 'clicks' from the relevancy maps for the single image segmentation method. All other parameters were unchanged from the baseline implementation of Zabari et al. [74].

E.2 Rich queries versus concept-only queries

In the main manuscript, we noted that surprisingly the segmentation model performed better when provided with a text target of the form "A photo of a [CONCEPT]" (i.e. a "Concept-only" query) than when provided with a rich textual caption.

To investigate this behavior, we turn to an analysis of the local relevance maps, which are used to guide the segmentation. Our investigation reveals that often, when the rich query describes other objects within the image, CLIP's attention drifts towards those objects. That is, CLIP struggles with leveraging relational information in the text and instead splits its focus between several objects mentioned in the rich query. Figure A.7 provides a qualitative visualization of this effect.

To *quantitatively test this hypothesis*, we re-ran segmentation, this time masking out relevancy scores of the background, except for objects which are also valid retrieval candidates. Now, context objects were no longer valid candidates. Indeed, we found that with this manipulation, rich queries do outperform concept-only queries, as in retrieval (Fig. A.3).

We conclude that a good caption for CLIP-guided personalized segmentation should describe the object or its immediate vicinity, and not its relation to other objects.

Last, we further investigated whether COLLIE demonstrates a similar sensitivity to rich queries. COLLIE's segmentation performance in the two scenarios is shown in Fig. A.4. We observe that, in contrast to our own approach and the baseline CLIP, COLLIE's performance on the segmentation task does not appear to be sensitive to the level of detail in the query.

E.3 Qualitative Analysis

In this section, we provide an additional qualitative analysis of segmentation using PALAVRA. We compare our results with the recent baseline COLLIE. Figure A.5 shows examples of successful segmentation, and Figure A.6 shows some failure modes.

F Evaluation datasets

We provide details for creating our two new benchmark datasets, based on DeepFashion and YTVOS.

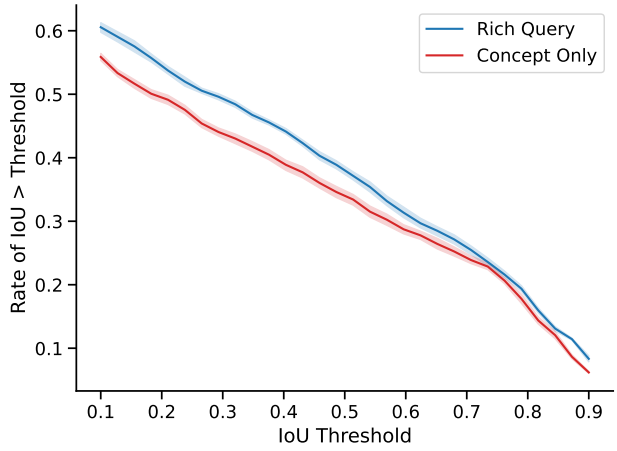

Fig. A.3. Rich queries outperform concept-only queries when context objects are not valid segmentation candidates.

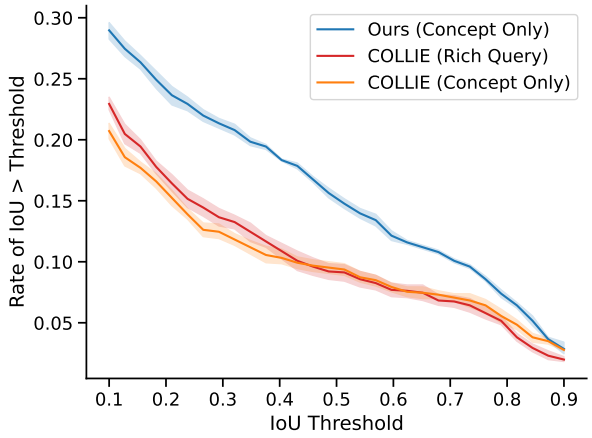

Fig. A.4. COLLIE performance when supplied with rich queries and with concept-only queries. The performance of COLLIE does not depend on the rich query, indicating that the additional information is largely ignored in the case of segmentation.

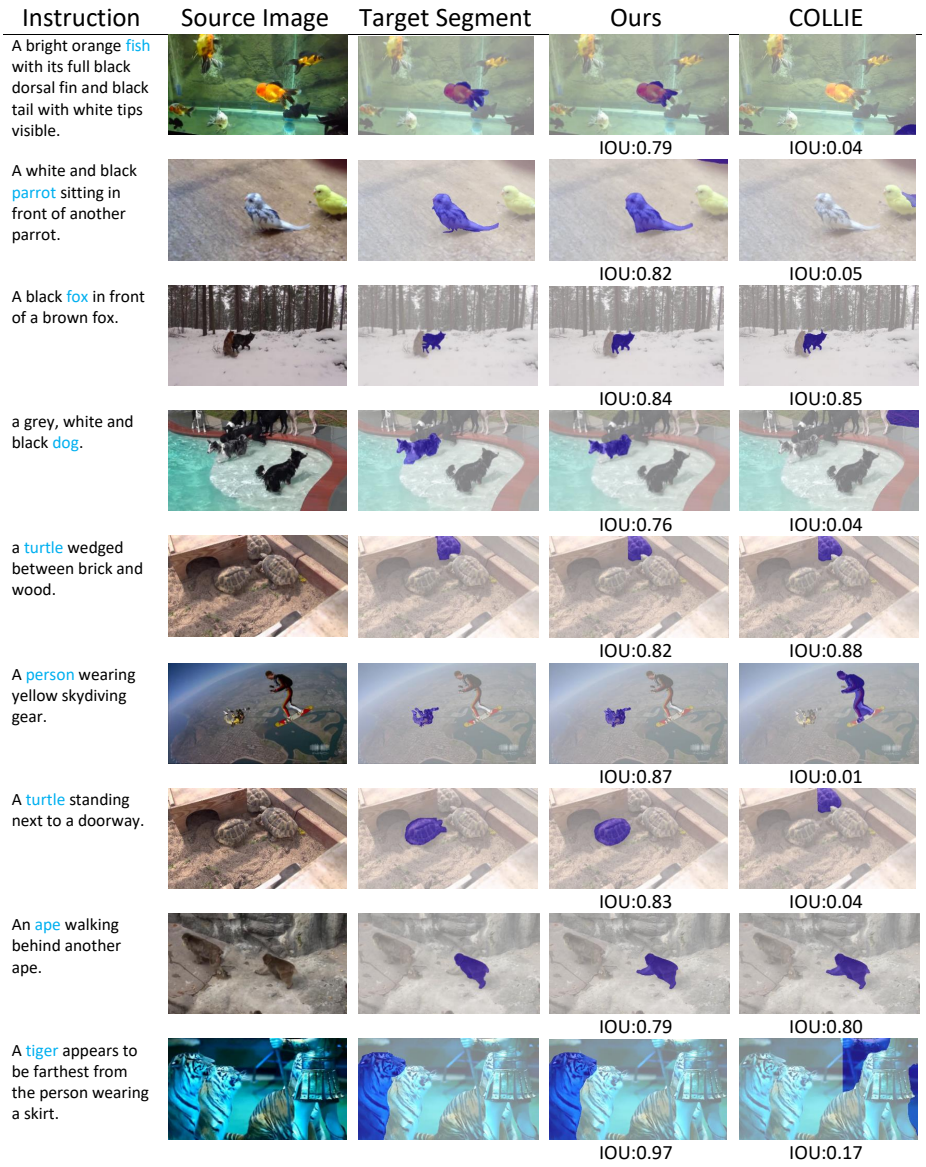

Fig. A.5. Examples of successful segmentation. For visualization purposes, we replaced the [CONCEPT] tag by the name of its concept type, and highlighted it in cyan.

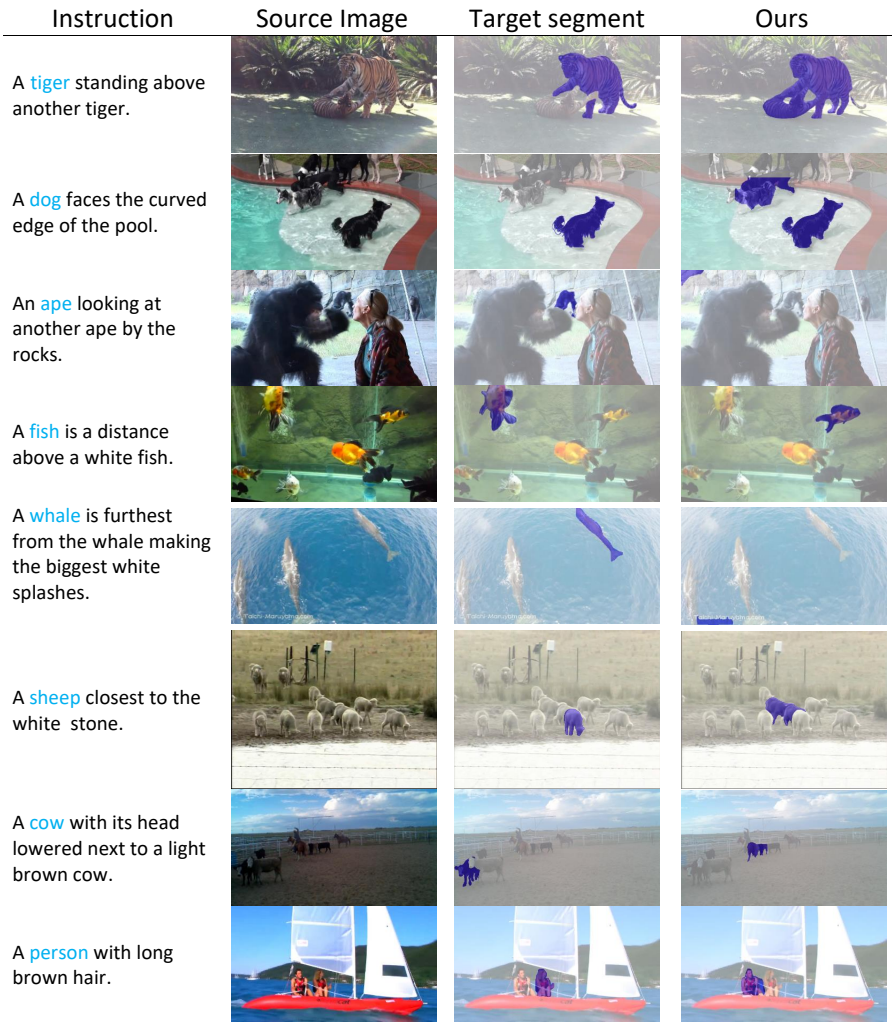

Fig. A.6. Examples of segmentation failures. For visualization purposes, we replaced the [CONCEPT] tag by the name of its concept type, and highlighted it in cyan.

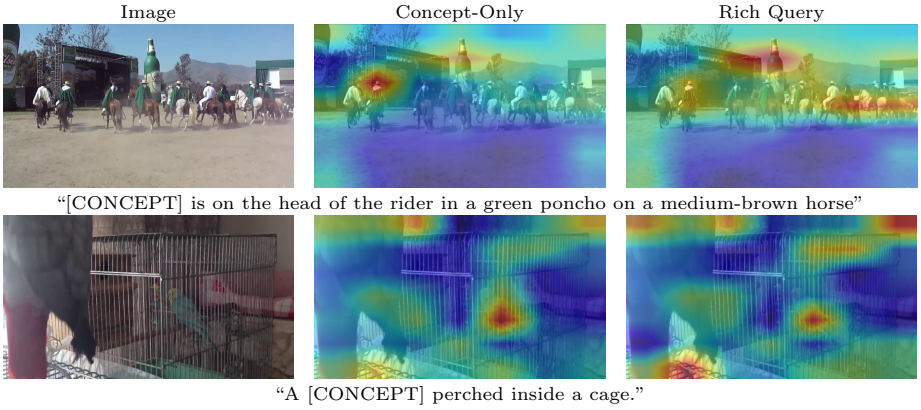

Fig. A.7. Qualitative examples of 'attention drift' when using rich queries. When the descriptor mentions other objects, CLIP's attention visually drifts away from the target concept and towards other objects described in the query. For example, in the top row, focus moves from the hat and towards the brown horses. On the bottom row, focus moved away from the parrot and towards the empty cage at the bottom of the frame.

F.1 DeepFashion2

To ensure that DeepFashion2 benchmark items are included in a rich visual context, items were included in the dataset if they obey the following criteria: (1) Have at least 5 images with a proper scale (zoom). Specifically, the item covers no more than 50% of the image. (2) There are at least 15 images of the same item in total. The set yielded ~1700 images and 100 unique fashion items (concepts) which met these criteria. Each unique fashion item was assigned a unique [CONCEPT] tag.

Next, we explain how we annotated a subset of this data with textual descriptions, and how we selected the evaluation set.

We manually curated a subset of 652 images (out of 1700) that contain a person wearing a fashion item and at least one additional object for a context. We did not consider mobile phones or mirrors as valid context, as these objects are abundant in the dataset. For each image, we collected a textual description that refers to each fashion item. For instance, *The [CONCEPT] is facing a glass store display."*.

To provide diverse captions, we instruct the raters to *avoid* trivial captions such as *"a [CONCEPT] in front of a mirror"*. We also instructed them to avoid describing the item itself, because the same item appears in several evaluation images, and we wished to have a textual query which is specific to one single image.

We randomly sampled an evaluation set (out of the 652 images), by sampling 5 images per concept, or less, if not available. This results in 450 evaluation images and 1250 images for training. Finally, we made a concept-based split by randomly splitting the dataset to 50 validation concepts and 50 test concepts.

Annotations for DeepFashion2 with Amazon Mechanical Turk

To simplify the instructions for collecting textual annotations, we used the fact that every fashion item is worn by a person. When describing the images, we simply asked the raters to relate to the person in the image in context of the objects in the scene, and in a post-processing step, we replace every mention of the word "person"

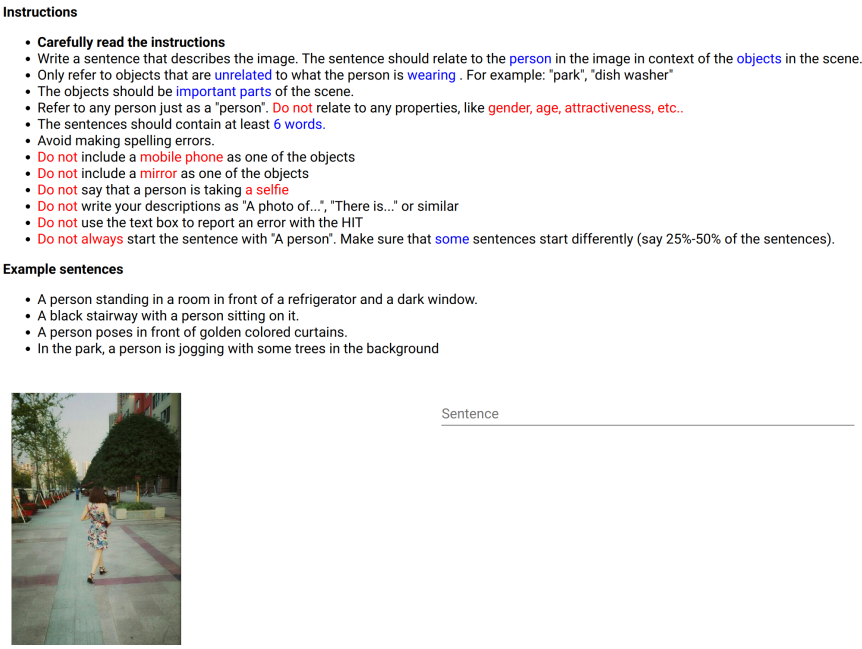

Fig. A.8. Instructions for collecting textual descriptions for images of the DeepFashion2 benchmark.

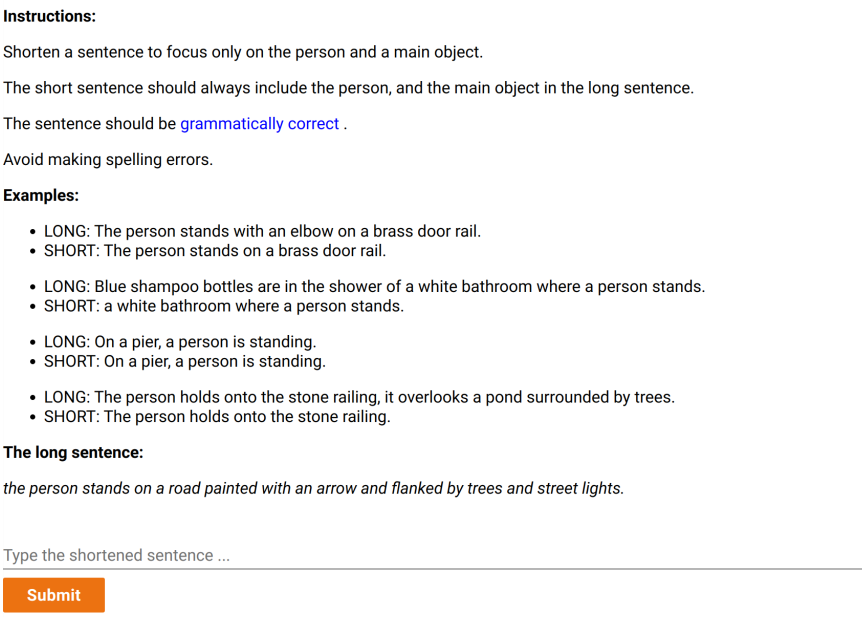

Fig. A.9. Instructions for summarizing textual descriptions for images of the DeepFashion2 benchmark.

by the "[CONCEPT]" token. Additional instructions were inspired by the instructions provided for collecting captions for the COCO dataset (See appendix of [40]).

Finally, to maintain the quality of the textual descriptions, we only worked with the raters after they passed our qualification test, making sure that they followed the instructions when describing 5-10 images. In addition, we only worked with raters with AMT "masters" qualification, demonstrating a high degree of approval rate in a wide range of tasks. We paid the raters 0.2$ for annotating each image.

Fig. A.8 provides an example of the data collection API for textual annotation of images for the DeepFashion2 benchmark.

F.2 Summarizing textual annotations with AMT

As explained in Sec. B, for DeepFashion2 we created two types of captions for each image, in order to quantify the effect of caption length. We expected that image retrieval with short textual queries will pose a greater challenge, because they contain less information about the target image, leading to queries that are more ambiguous.

To create the set of "short" text queries, we took the set of image captions described in Sec. F.1, which we now denote as "detailed" captions, and asked the AMT raters to summarize each caption. Given a detailed caption, their goal was to describe the concept in the context of a *single* object in the scene. An example of a caption and its summarized version is: "*White cabinets, some with open drawers, are alongside and behind the [CONCEPT].*" was summarized to "*White cabinets are behind the [CONCEPT].*"

Fig. A.9 provides an example of the data collection API to summarize textual descriptions.

Similarly to the previous section, to maintain the quality of the textual descriptions, we only worked with raters after they passed our qualification test and have AMT "masters" qualification. We paid the raters 0.1$ for summarizing each caption.

Finally, for most of the DeepFashion2 experiments throughout the paper, we used the more challenging "short" queries. In Sec. B we describe the evaluation results with the "detailed" queries.

F.3 Youtube-VOS

Overview

We created an image segmentation benchmark of personalized visual concepts given a textual query using Youtube-VOS (YTVOS) [72]. YTVOS is a dataset for instance segmentation in video, which includes 4000+ videos, 90+ categories, and 7800+ unique object instances. The original videos were $3-6$ second long with a 30 FPS frame rate. The dataset contains a subset of the frames, sampled at rate of 6 FPS. To transform the dataset into an image personalization benchmark, we take the last frame of each video (scene) for evaluation and the object instances that appear in it as target concepts. Earlier frames that contain that object are used as candidate frames for few-shot training. See examples in Figures 5(left), A.2, A.5, A.6 and A.7.

For building the concept set, we consider each object instance (e.g. each animal in the frame) as a unique personalized concept. We chose training samples such that their object instantiations are not trivially solved by simple visual template matching with the last (evaluation) frame. To that end, we use the following criteria: For each object instance, we consider all the previous video frames that contain it. We keep only the

frames where: (1) the object's segmentation mask has a zero intersection-over-union (IOU) score when compared with its mask at the last frame (i.e. the evaluation target) and (2) the center of the mask moved at least 150 pixels when compared to the final frame. We discard any object instance that does not have at least 4 training examples left at the end of this filtering process. Finally, we take a box crop of the images around the selected masks and use them as training examples.

Annotation with AMT

We annotated the instances in the evaluation frame with captions using AMT. We instructed the AMT workers to concisely describe what makes a specific entity distinct, compared with similar entities in the image, and, if possible, preferring descriptions that relate to one object that is nearby.

Finally, similar to the previous sections, to maintain the quality of the textual descriptions, we only worked with raters after they passed our qualification test and have AMT "masters" qualification. We paid the raters 0.3$ for every textual description.

Fig. A.10 provides an example of the data collection API for textual annotation of images for the Youtube-VOS benchmark.

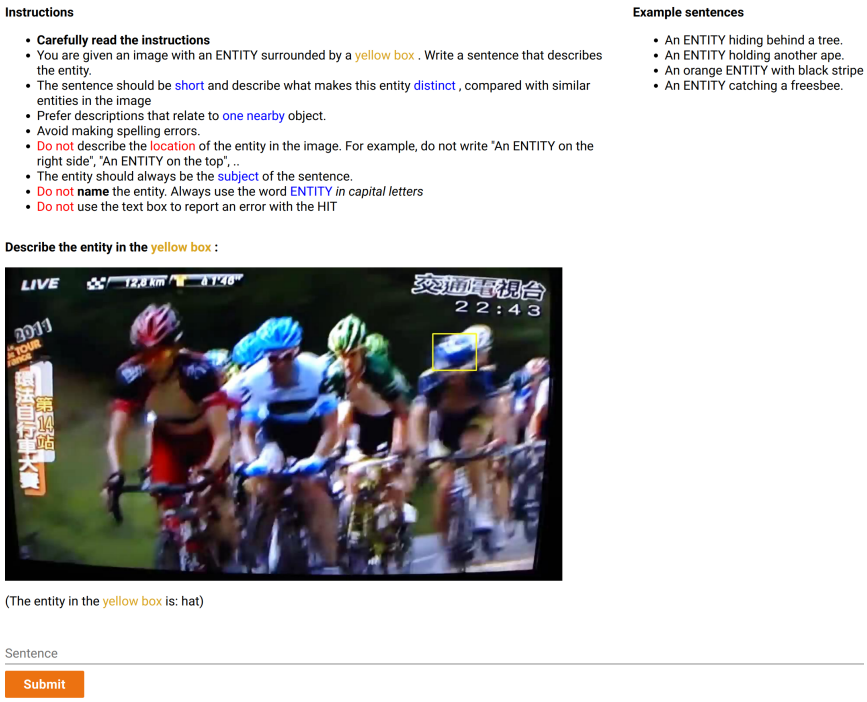

Fig. A.10. Instructions for collecting textual descriptions for object instances in the Youtube-VOS benchmark.

Personalized image retrieval: We also created an image retrieval variant of YTVOS. We extract a set of images that correspond to the collected captions, where every image in the retrieval set was extracted from a wide box cropped around every instance in

each evaluation frame. The box size was set to twice the size of the instance mask *on each axis* (that is, four times the area), to allow it to display some information about the context of the instance.