Appendix

A. Original Rocchio formulation

The Rocchio algorithm [19] aims to generate the refined query vector $\boldsymbol{z}'_q \in \mathbb{R}^d$, by fusing representations of original query $\boldsymbol{z}_q$ with relevant and irrelevant (or positive and negative) feature vectors $\boldsymbol{z}_p$ and $\boldsymbol{z}_n$ obtained in the current stage of retrieval. This updated representation is then used to perform another round of retrieval. The Rocchio query refinement rule can formally be defined as follows:

$$\boldsymbol{z}'_q = \alpha \boldsymbol{z}_q + \beta \boldsymbol{z}_p - \gamma \boldsymbol{z}_n \tag{9}$$

The positive vector is computed from the representations of the most relevant candidates, whereas the negative features are aggregated from the irrelevant ones. Specifically, the aggregated positive and negative representations are computed by averaging:

$$\boldsymbol{z}_p = \frac{1}{|\boldsymbol{C_r}|} \sum_{i \in \boldsymbol{C_r}} \boldsymbol{z}_i \tag{10}$$

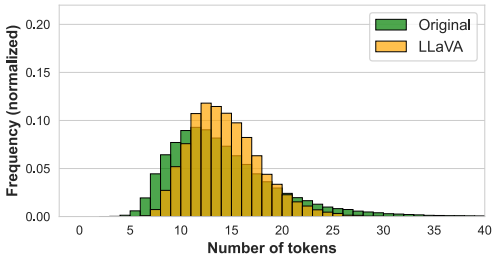
$$\boldsymbol{z}_n = \frac{1}{|\boldsymbol{C_{nr}}|} \sum_{i \in \boldsymbol{C_{nr}}} \boldsymbol{z}_i \tag{11}$$

where $\boldsymbol{C_r}$ and $\boldsymbol{C_{nr}}$ are the sets of the most relevant and non-relevant candidates with cardinalities $|\boldsymbol{C_r}|$ and $|\boldsymbol{C_{nr}}|$. In practice, the algorithm is often applied with $K = |\boldsymbol{C_r}| = |\boldsymbol{C_{nr}}|$, i.e., using the top-K candidates with the highest and lowest relevance with respect to the query.

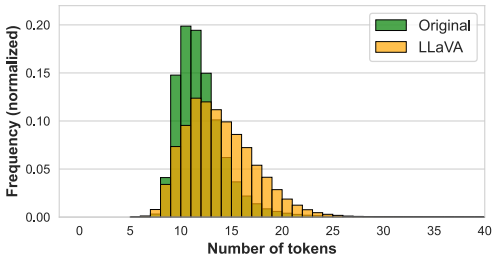
B. Generating Synthetic Captions with LLaVA-1.5

We utilized LLaVA-1.5-7B[6] [14], quantized in 8-bit, to generate a synthetic caption for each image in both Flickr30K and COCO. Given that these captions would later be used to obtain CLIP and BLIP-2 representations – which typically rely on short and descriptive textual inputs [20, 31] – we adopted a two-step prompting strategy to balance diversity and conciseness. First, we randomly selected a prompt from a predefined pool of five sentence-level description prompts, ensuring variability in the generated responses. Second, we provided a follow-up prompt explicitly instructing the model to generate a single-sentence description. The prompts and conversation template are shown in Table 7. Figure 6 shows the distributions of the lengths of the original and generated captions. As observed, the length distribution of generated captions differs slightly from that of human-written ones while still avoiding extreme values and remaining within a reasonable range.

[6]We used open-source checkpoints available via HuggingFace: https://huggingface.co/llava-hf/llava-1.5-7b-hf

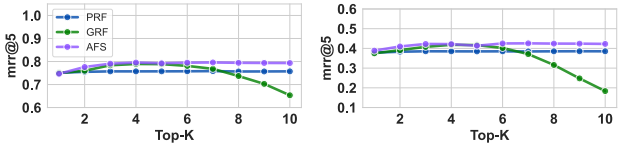


(a) Flickr30K



(b) COCO

Figure 6. Distribution of original and LLaVA-generated caption lengths.



(a) Flickr30K (b) COCO

Figure 7. **Retrieval with Rocchio: top-K items**. Retrieval metrics with a varying number of items in relevance feedback.

Dataset	Feedback	0.05	0.10	0.25	0.50
Flickr	GRF	0.716	0.714	0.707	0.709
	PRF	0.669	0.670	0.669	0.671
COCO	GRF	0.330	0.331	0.327	0.327
	PRF	0.295	0.295	0.295	0.294

Table 6. Retrieval performance (Hits@1) with different values of temperature.

C. Rocchio Temperature

In Equation 1, we define positive and negative weights for each item in relevance feedback. Specifically, we scale similarities between each item and the corresponding query representation by temperature τ to compute the weight of each item. These weights are then used in Equation 2 to guide query representations. Table 6 shows that using lower temperature values, which sharpen the weight distribution, leads to better retrieval performance.

Step	Role	Prompt
0		<prompt1> is randomly selected from the pool: • "Describe the image and main visual features in one sentence." • "Generate a short caption for this image, focusing on the visual details." • "Provide a concise description, one sentence, of the image visual features and surroundings." • "Write a brief caption, one sentence, that describes the image visual features." • "Summarize the image visual features in a precise caption, maximum one sentence." <prompt2>: • "Pay attention to the visual settings and details on the image. Write exactly one sentence under 10 words."
1	**User**	USER: <image><prompt1> ASSISTANT:
	Assistant	USER: <image><prompt1> ASSISTANT: <answer1>
2	**User**	USER: <image><prompt1> ASSISTANT: <answer1>USER: <prompt2> ASSISTANT:
	Assistant	USER: <image><prompt1> ASSISTANT: <answer1>USER: <prompt2> ASSISTANT: <answer2>

Table 7. Conversation template for two-step prompting: <answer2> is the generated synthetic caption for input image <image>.

D. Number of Candidates for Relevance Feedback

We evaluate the impact of the number of items in relevance feedback. According to Figure 7, the optimal performance in MRR@5 for both GRF and AFS is achieved with 4-5 relevant items on both datasets. Further increase of the relevance set leads to decay in MRR for GRF.

E. Attentive Feedback Summarizer

E.1. Attention Blocks Architecture

Section 3.3 presents an overview of the proposed attentive feedback summarizer architecture. This section provides a detailed description of how image patches and caption tokens are processed by the summarizer.

The first component of the AFS architecture is a cross-attention module (Figure 8) designed to fuse query representations with relevance feedback. Specifically, cross-attention queries are obtained for the token features of the user query. We denote these token-level representations of input query q with s tokens as $\boldsymbol{h}_q = [\boldsymbol{h}_{q,\texttt{CLS}}, \boldsymbol{h}_{q,1}, \boldsymbol{h}_{q,2}, \ldots, \boldsymbol{h}_{q,j}, \ldots, \boldsymbol{h}_{q,s}] \in \mathbb{R}^{s \times d_t}$, where d_t is the token dimensionality and $\boldsymbol{h}_{q,\texttt{CLS}}$ is a learnable CLS token. Furthermore, keys and values are obtained from relevance feedback, comprising patch- and word-level representations of the most relevant images and their corresponding AI-generated captions. Relevance feedback vector $\boldsymbol{r}_q = concat([\boldsymbol{r}_q^{\text{img}}, \boldsymbol{r}_q^{\text{cap}}])$ is a concatenated sequence of the patch and token embeddings for the top-K relevant images and their synthetic captions projected to $\mathbb{R}^{d_t}$. In detail, visual relevance sequence can be defined as $\boldsymbol{r}_q^{\text{img}} = [\boldsymbol{r}_{q,1}^{\text{img-1}}, \boldsymbol{r}_{q,2}^{\text{img-1}}, \ldots, \boldsymbol{r}_{q,p}^{\text{img-1}}, \ldots, \boldsymbol{r}_{q,1}^{\text{img-K}}, \boldsymbol{r}_{q,2}^{\text{img-K}}, \ldots, \boldsymbol{r}_{q,p}^{\text{img-K}}]$ for images divided into p patches $\boldsymbol{r}_{q,j}^{\text{img-c}} \in \mathbb{R}^{d_t}$. Similarly, the word-level relevance sequence from corresponding AI-generated captions is $\boldsymbol{r}_q^{\text{cap}} = [\boldsymbol{r}_{q,1}^{\text{cap-1}}, \boldsymbol{r}_{q,2}^{\text{cap-1}}, \ldots, \boldsymbol{r}_{q,s}^{\text{cap-1}}, \ldots, \boldsymbol{r}_{q,1}^{\text{cap-K}}, \boldsymbol{r}_{q,2}^{\text{cap-K}}, \ldots, \boldsymbol{r}_{q,s}^{\text{cap-K}}]$ with captions padded to length s and $\boldsymbol{r}_{q,j}^{\text{cap-c}} \in \mathbb{R}^{d_t}$.

The second component of the summarizer is a standard self-attention block that communicates information between query sequence tokens. Therefore, the output sequence of these two blocks can be defined as $\tilde{\boldsymbol{h}}_q = SelfAttn(CrossAttn(\boldsymbol{h}_q, \boldsymbol{r}_q)) \in \mathbb{R}^{s \times d_t}$. We use CLS token $\tilde{\boldsymbol{h}}_{q,\texttt{CLS}}$ as the output of the feedback summarizer and apply linear projection to obtain $\boldsymbol{z}_q^{\texttt{CLS}} \in \mathbb{R}^d$.

E.2. AFS Representations

In this section, we visualize representations $\boldsymbol{z}_q^{\texttt{CLS}}$ learned by the AFS model. We compare two AFS variants trained with different objectives: one using only the image-based loss (l_q^{img}), and one using a combined loss from both image and caption supervision ($l_q^{img} + l_q^{cap}$), corresponding to rows 2 and 3 in Table 2. As shown in Section 5.1, the variant trained with image-based loss achieves better retrieval performance. Figure 9 shows PCA projections of the query, image and AFS embeddings in two-dimensional space for Flickr30K with CLIP-ViT-B/32. The embeddings from the combined-loss model are positioned between the text and image embedding clusters, while the image-only

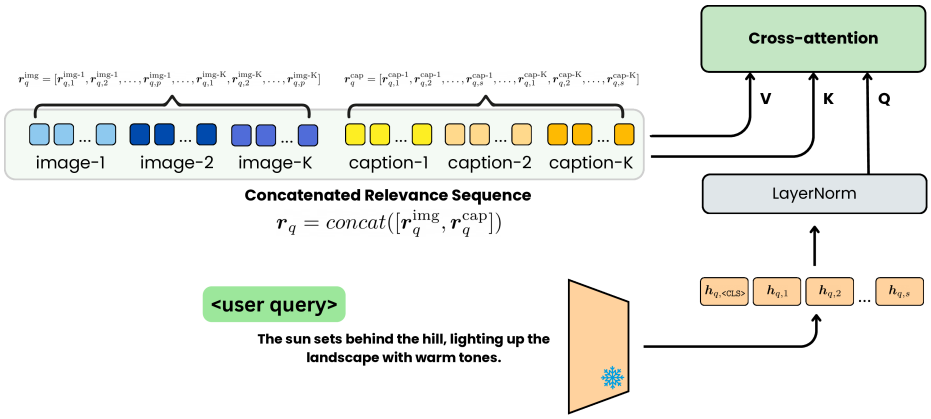
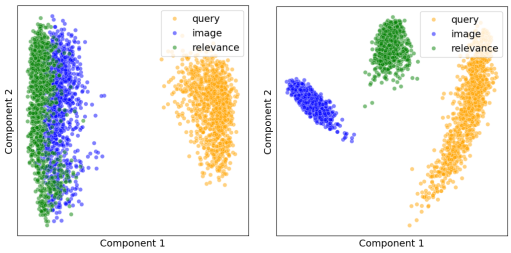


Figure 8. **Cross-attention architecture.**



(a) Image loss (b) Combined loss

Figure 9. Images, queries and AFS representations projected with PCA.

variant produces representations that remain closer to the image features. This pattern suggests that feedback representations more aligned with image features may be more effective for refining query representations.

E.3. Cross-attention Score Aggregation

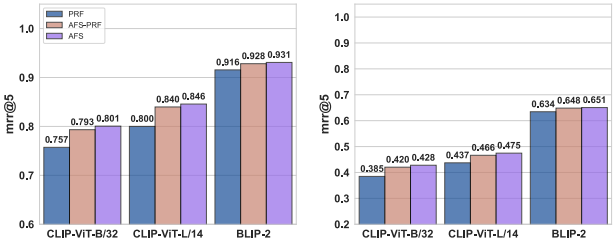
In Section 5.3 and Appendix E.5, we visualize cross-attention scores to demonstrate feedback summarization from fine-grained feedback.

In this section, we provide technical details on cross-attention score aggregation. Cross-attention scores are defined as $\boldsymbol{A}_q \in \mathbb{R}^{n_h \times s_q \times s_r}$, where n_h is the number of attention heads, s_q is the number of tokens in a user query, and s_r is the number of patch and token embeddings in a relevance sequence. First, we accumulate attention scores over attention heads and query tokens obtaining $\hat{\boldsymbol{A}}_q = [\hat{A}_{q,1}, \hat{A}_{q,2}, \ldots, \hat{A}_{q,s_r}] \in \mathbb{R}^{s_r}$, where each item is a scalar corresponding to a certain patch or token in relevance sequence $\boldsymbol{r}_q$. Then, we split $\hat{\boldsymbol{A}}_q$ into image and caption parts $\hat{\boldsymbol{A}}_q^{\text{img}} = [\hat{A}_{q,1}^{\text{img-1}}, \ldots, \hat{A}_{q,p}^{\text{img-1}}, \ldots, \hat{A}_{q,1}^{\text{img-K}}, \ldots, \hat{A}_{q,p}^{\text{img-K}}]$ and $\hat{\boldsymbol{A}}_q^{\text{cap}} = [\hat{A}_{q,1}^{\text{cap-1}}, \ldots, \hat{A}_{q,s}^{\text{cap-1}}, \ldots, \hat{A}_{q,1}^{\text{cap-K}}, \ldots, \hat{A}_{q,s}^{\text{cap-K}}]$ for K items used as relevance feedback. Finally, both sequences $\hat{\boldsymbol{A}}_q^{\text{img}}$ and $\hat{\boldsymbol{A}}_q^{\text{cap}}$ are independently normalized between 0 and 1 using min-max scaling. As a result, the normalized values are used as saliency scores for patches on relevant images and words in synthetic captions.

E.4. AFS without Synthetic Captions

As shown in Table 4, PRF does not improve over baseline retrieval models without relevance feedback (Table 4: rows 2, 7, and 12). In Sections 3.3 and 4.5, we introduced AFS as a strategy that combines PRF and GRF by aggregating local information from both image patches and synthetic captions. Here, we evaluate whether AFS can operate using retrieved image embeddings without synthetic captions. We refer to this variant as AFS-PRF.[7]

Figure 10 shows that AFS-PRF performs only slightly worse than the full AFS model while still outperforming PRF with Rocchio. This result suggests that AFS can serve as an effective tool for enabling pseudo-relevance feedback.



(a) Flickr30K (b) COCO

Figure 10. **AFS-PRF Evaluation**. Comparison between AFS without generated captions (AFS-PRF), its full version, and Rocchio-based PRF.

[7] We did not re-train the models using images only; instead, we used the same trained models from previous sections while ignoring synthetic captions at inference.

E.5. Cross-attention Visualization Examples

We provide more cross-attention visualizations obtained from the AFS model, supplementing Section 5.3. In Figures 14 and 15, saliency maps are obtained for CLIP-ViT-B/32 and CLIP-ViT-L/14, respectively. In both cases, the ground-truth images were not among the top-5 initially retrieved items and, hence, were excluded from relevance feedback aggregation. Nevertheless, the obtained attention maps highlight semantically meaningful regions, and the refined query representations lead to better retrieval performance. Furthermore, Figures 16 and 17, both using CLIP-ViT-L/14, show scenarios where the ground-truth image ranks improved after applying AFS.

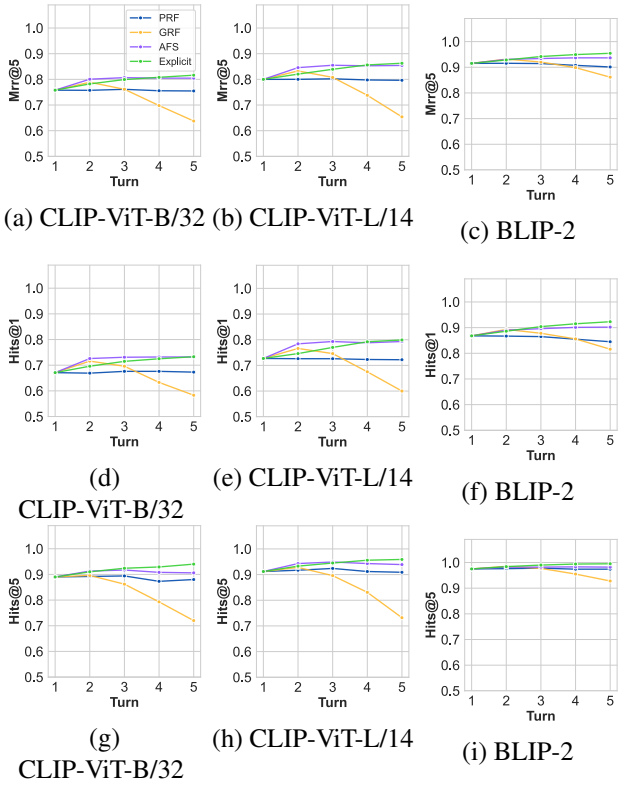
F. Multi-turn Retrieval Results

This section presents the extended results of the multi-turn retrieval with relevance feedback aggregation. Figures 11 and 12 show the performance metrics achieved across retrieval rounds for the Flickr30K and COCO datasets. The obtained results complement the findings discussed in Section 5.2. Specifically, explicit feedback with ground truth captions continuously improves retrieval performance according to all metrics. Generative relevance feedback, however, leads to performance degradation starting from the third retrieval round. The attentive feedback summarizer demonstrates an increase in performance at round 2, gradually converging in rounds 3-5, avoiding query drift without ground-truth captions.

G. Interactive Retrieval Demo

In Section 6, we outlined future directions for combining relevance feedback techniques with user interactions to create more interactive retrieval systems. While some recent work explores chat-based interactions via LLMs [8, 32], we investigate an alternative approach based on direct visual interaction with images. Specifically, we developed a prototype interface where users can retrieve images based on a textual query. The interface, then, allows users to annotate retrieved image regions by drawing bounding boxes indicating relevance or irrelevance to their search intent (Figure 13).

We explored how the bounding boxes could be integrated into the inference process of the AFS model working without synthetic captions (Appendix E.4). Specifically, we modified the cross-attention scores based on a simple heuristic: attention weights were increased for image patches corresponding to regions marked as relevant by the user and decreased for those identified as irrelevant. The magnitude of the increase is a hyperparameter and can be defined through configurations. These adjustments were applied at inference time without requiring any additional training or fine-tuning of the model.



(a) CLIP-ViT-B/32 (b) CLIP-ViT-L/14 (c) BLIP-2

(d) CLIP-ViT-B/32 (e) CLIP-ViT-L/14 (f) BLIP-2

(g) CLIP-ViT-B/32 (h) CLIP-ViT-L/14 (i) BLIP-2

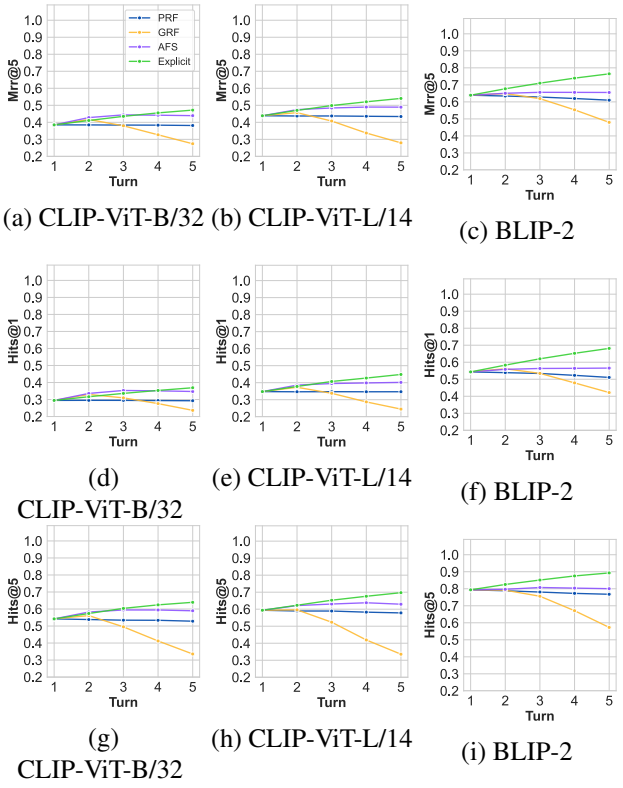
Figure 11. Multi-turn retrieval performance with relevance feedback on Flickr30K.

As shown in Figure 13, this approach effectively shifts the model's attention toward user-indicated regions, enabling more targeted retrieval responses. While our method is based on fixed weight adjustments, it demonstrates how user input can guide attention in relevance-based models. In the future, more sophisticated techniques could be explored. For instance, it is possible to add learnable token-level embeddings based on user feedback. This would require training the model with region-level relevance annotations, which are currently not available.

The code for the prototype is available at: https://github.com/bulatkh/visualref/tree/wacv_demo

H. Limitations

Our study systematically evaluated relevance feedback strategies with pre-trained VLM backbones. However, we do not compare against alternative query adjustment methods, such as query rewriting or prompt engineering with LLMs-in-the-loop. Future work can address this gap and explore the interplay between representation-level refinement and natural language query adjustments. This direction is particularly relevant for dialogue-aware image retrieval, where systems must model contextual coherence



(a) CLIP-ViT-B/32 (b) CLIP-ViT-L/14 (c) BLIP-2

(d) CLIP-ViT-B/32 (e) CLIP-ViT-L/14 (f) BLIP-2

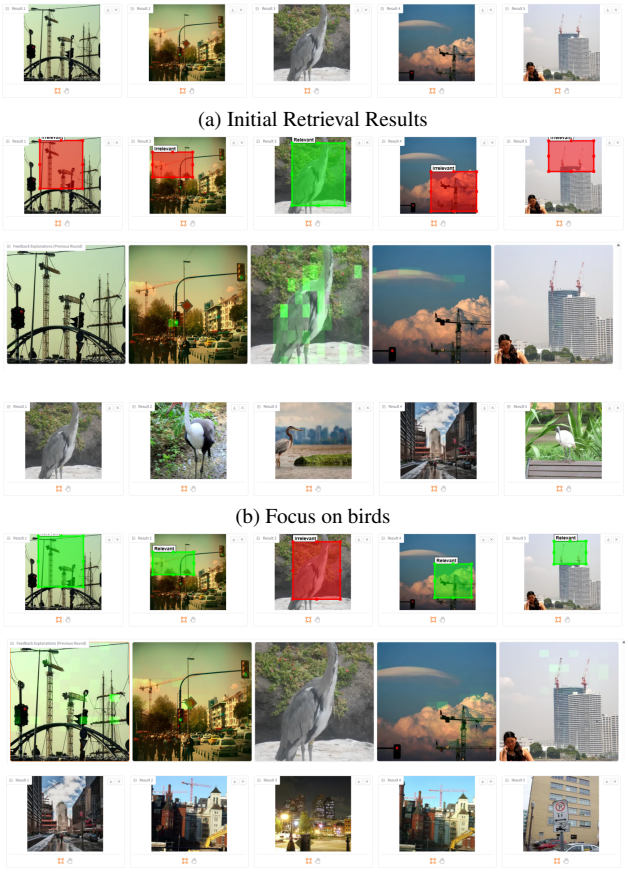
(g) CLIP-ViT-B/32 (h) CLIP-ViT-L/14 (i) BLIP-2

Figure 12. Multi-turn retrieval performance with relevance feedback on COCO.

across multiple modalities and turns, resolve ambiguous references, and encode rich dialogue context for effective prompting of VLMs, which often struggle with non-descriptive queries [20].

Further, GRF and AFS strategies introduce some additional overhead. GRF requires running an image captioning model in the background to generate captions for candidate images. AFS, on the other hand, operates at runtime, although its size remains modest (under 20 million parameters), especially compared to LLMs, which can contain billions of parameters. Therefore, we suggest that developers carefully weigh the trade-off between the retrieval gains offered by relevance feedback and the computational (and environmental) costs associated with deploying these methods.

Finally, we tested the proposed relevance feedback strategies on general-purpose image retrieval datasets. In future work, we are planning to evaluate these methods in domain-specific retrieval tasks.



(a) Initial Retrieval Results

(b) Focus on birds

(c) Focus on machines

Figure 13. **Interactive Retrieval Demo.** (a) Initial retrieval results for the textual query *"Crane in the city"*. (b) The user specifies their intent by marking the bird as relevant and industrial cranes as irrelevant. (c) The user instead focuses on industrial cranes. In both (b) and (c), the first row shows examples of manual relevance annotations, the second row visualizes cross-attention scores after applying the feedback, and the third row presents the updated retrieval results using AFS with user signals.

Figure 14. **Cross-attention visualization with CLIP-ViT-B/32**. This example shows a case when a ground-truth image was not used for relevance feedback aggregation, i.e., it was not initially among the top 5 retrieved images.

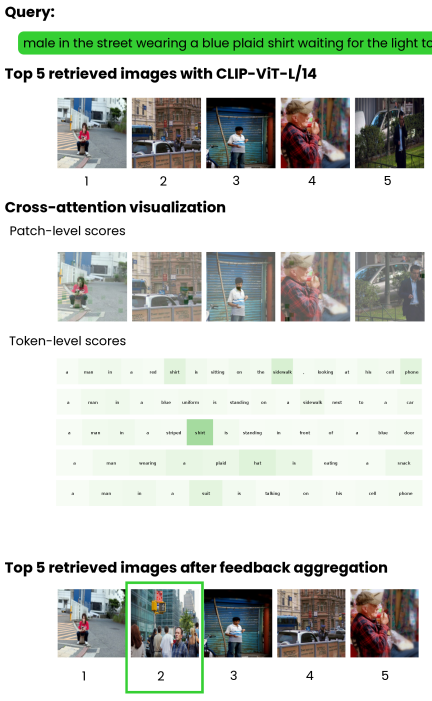


Figure 15. **Cross-attention visualization with CLIP-ViT-L/14**. The ground-truth image was not used for relevance feedback aggregation, i.e., it was not initially among the top 5 retrieved images.

Figure 16. **Cross-attention visualization with CLIP-ViT-L/14.** AFS increases the rank of the ground-truth image.

Figure 17. **Cross-attention visualization with CLIP-ViT-L/14**. AFS increases the rank of the ground-truth image.