Seeing the Abstract: Translating the Abstract Language for Vision Language Models

Supplementary Material

A. Overview

In this supplementary, we provide additional details on the implementation of the presented approach and additional experimental analysis on ACT. In Section B we present further details on the dataset discussed in our work, the attribute extraction pipeline for the preliminary analysis, and extensive details on ACT. Section C presents additional analysis on the latent space of the Vision Language Model (VLM) and the cross-dataset evaluation on FACAD. Finally, in Section D, we show additional interesting qualitative results of the retrieval on DeepFashion using ACT.

B. Additional details

In this section we provide additional details on the dataset used for the presented analysis (Sec. B.1) and report implementation details, especially focusing on the extraction pipeline leveraged for fashion language analysis (Sec. B.2) and prompting information (Sec. B.3).

B.1. Fashion datasets vs LAION 400M

In Fig. 1 we show some data samples taken from the fashion-related DeepFashion and FACAD, as well as two samples randomly picked from LAION 400M. It is clear how, apart from the image content, the descriptions differ greatly: both DeepFashion and FACAD provide longer descriptions with more abstract-oriented properties, such as "chic" and "street ready", to describe both appearance and feeling precisely; on the other hand, LAION 400M focuses the descriptions on the visually grounded aspects to briefly describe the content. This difference is further highlighted in the adjective wordclouds of FACAD Fig. 2 and LAION Fig. 3. While FACAD has a balanced distribution between concrete and abstract adjectives, LAION 400M exhibits a noticeable bias towards concrete adjectives. The word cloud is predominantly filled with concrete descriptors with a strong emphasis on color-related terms like "white" and "black". For exact statistical numbers, we refer the reader to Tab. 1 in the main paper.

B.2. Analysis extraction pipeline

Our attribute extraction and classification pipeline uses the Python-based spaCy library to tokenize, tag, and extract adjectives and attributes from the item descriptions. The pipeline follows 4 main phases: *i) preparation phase* where we instantiate the NLP model and tokenizer; *ii) tagging phase* where the NLP model performs POS tagging; *iii) extraction phase* where we extract the adjectives/attributes from the descriptions; *iv) classification phase* where we classify the adjectives/attributes into abstract and concrete following a lexicon. We briefly discuss the pipeline in the following paragraphs and will release the code upon acceptance.

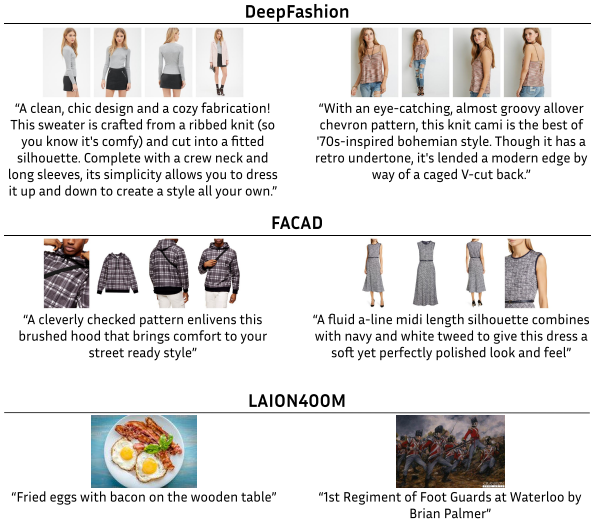

Figure 1. Samples taken from the datasets considered for analysis in our main paper. Compared to the abstract-oriented fashion datasets, LAION 400M mainly provides short and concrete details about the image content.

Figure 2. Wordcloud of concrete (blue) and abstract (red) adjectives in FACAD descriptions. The larger font indicates a higher frequency. Abstract adjectives make up a large portion of the entire dataset, being as frequent as concrete ones (see Tab. 1 of the main paper).

i) Preparation phase. We first instantiate a language processor using the pre-trained spaCy `en_core_wb_sm` model and then modify its tokenizer, keeping hyphenated words as a single token (*e.g.*, `mind-blowing` will be considered a single word, instead of `[mind, blowing]`). This

Figure 3. Wordcloud of concrete (blue) and abstract (red) adjectives in LAION descriptions. The larger font indicates a higher frequency. It's clear how concrete adjectives are more frequent than abstract ones (see Tab. 1 of the main paper).

choice is guided by our observation that hyphenated words are common in the fashion language to convey particular details, *e.g.,* `v-neckline`, `must-have`, `happy-mood`, etc.

ii) Tagging phase. With the spaCy[17] model configured, we then process each description by *tokenizing* and *tagging* them. We use the tokenizer introduced in the previous step to parse the description and separate the words into tokens *i.e.,* atomic words in the sentence, then we conduct POS tagging using the spaCy model. This step takes the list of tokens and analyzes their syntactic relationship, building a Part-of-Sentence (POS) tree. In this tree, each token is associated with a tag (*e.g.,* adjective, noun, verb, etc.) and connected to other tokens through syntactic relationships, *e.g.,* `green jacket` will tag `green` as an adjective, and connect it to `jacket` through an `adjective modifier` relationship. We will refer to this tree as our `tagged description`, for simplicity.

iii) Extraction phase. With the description successfully tagged, we can then extract the attributes. In particular, when analyzing the adjectives only (*i.e.,* dataset statistics in Sec. 3 of the main paper), we keep all the words that were tagged as "adjectives" or that had an "adjective modifier" relationship in the tagged description. Finally, to avoid word repetitions, we lemmatize the words using the spaCy model built-in lemmatizer.

Similarly, when analyzing attributes (which we define as adjective+noun couples), we follow the same procedure to first locate all adjective tokens. To extract the attribute, we then check whether each of these tokens constitutes a valid attribute with its relationship head (the word it refers to or modifies). To do so, we apply the following rules:

1. If the head is either an "adjective", or was tagged as the "subject" of the sentence, lemmatize it and create the attribute (token, head).
2. If the token was a "compound word" (*i.e.,* a concept expressed through multiple words, such as "knee length"), extract all the (token, compound) couples as attributes.

iv) Classification phase. Finally, we classify the attributes following the methodology and lexicon presented in [5]. It is worth noting that some adjectives were not present in the lexicon, and we found many words' concreteness was context-dependent. Words like `cool` and `sharp` were rightfully considered concrete in the lexicon since, in a common context, they refer to the "sensation of cold" and the "ability to cut" respectively, which are properly assessable with senses. In the fashion domain, however, these words change in meaning and rather describe the style of the clothing, which we define as abstract. We manually assigned concrete/abstract classes to these words following suggestions from domain experts.

Examples. We provide some examples of the attribute extraction results. In blue we highlight the concrete attributes extracted, while in red the abstract ones.

> A duo of dark stripes elongate a midweight cotton polo, styled in a comfortable fit, with long sleeves, and a sporty zipper in the placket.

> Stay comfy and chic while showing off your bump in this scalloped neck sheath, featuring flattering seam, and a knee length skirt.

B.3. Prompt details

Captioning. For captioning purposes, we utilize recent state-of-the-art instructed VLMs, referred as captioning models, allowing for a detailed description of the fashion item of interest. Specifically, we experiment with Int4-quantized versions of Qwen2-VL [35] (`Qwen2-VL-7B-Instruct-GPTQ`) and CogVLM2 [16] (`cogvlm2-llama3-chat-19B`) that are queried to focus on item class using the vision prompt p_v:

> p_v: Briefly describe in detail the ⟨`class`⟩ only in less than 10 tokens, use visually grounded properties like shape, pattern and garment details.

where ⟨`class`⟩ denotes the class of the fashion item of interest. The caption generation is limited to 77 tokens.

Language rewriting. Language rewriting relies on the use of a LLM for a first rephrased version of the provided abstract query in more concrete terms. For the main experimentation, we use an instructed version of Llama-3 [11] (`Llama-3.1-8B-Instruct`) loaded for inference in half precision. The model is guided to focus on concrete aspects of the original descriptions, discarding information that downstream VLMs cannot leverage for matching. In practice, we use the prompt:

> p_r: Briefly describe the fashion item in less than 10 tokens in a natural language sentence in a discriminative way. Avoid specialized language and listing properties. Focus on shape, patterns, color (if indicated) and garment details. Remove information on how to pair and complement. Strictly remove suggestions on occasions when to wear it.

We further add *four* In-Context samples [10] to provide demonstrations of the desired rewriting outcomes:

> This sheer Georgette top features a high collar and shirred shoulders. It features long sleeves and buttoned cuffs $=\rangle$ A sheer, high-collared Georgette top with shirred shoulders, long sleeves, and buttoned cuffs.

> Represent your love for Long Beach hip-hop with this old school graphic of Snoop Dogg on a muscle tee. To balance some softness into this edgy piece, tuck it into a skater skirt and finish it off with flatforms $=\rangle$ A graphic muscle tee featuring Snoop Dogg.

> Busy mornings and stacked social calls are no match for this throw-on-and-go tunic! Its boxy silhouette and ultra-soft knit fabrication will have you taking on the day in breezy comfort, while a scoop back lends it some unexpected sartorial side-eye. Plus, you can just as easily throw on this short-sleeved number over leggings to effortlessly transition from day to night. $=\rangle$ A casual, boxy tunic with a scoop back, made from soft knit fabric.

> Cut from a crisp cotton woven splashed with a richly ornate paisley print, this short-sleeved shirt is guaranteed to be a standout in your collection. Its boldness is pared down by its sleek and refined structure (a slim fit, classic collar, and buttoned front). Just because it's a standout doesn't mean it isn't versatile - wear it with everything from chinos to joggers for a sharp dose of style. $=\rangle$ A crisp cotton shirt with a paisley print, slim fit, classic collar, and buttoned front.

The provided rewritten descriptions are randomly drawn from the DeepFashion evaluation set and hence removed from the final performance testing. These sample descriptions are the only captions manually re-written with human involvement. A similar prompt is used with other LLMs when required to ablate on the language rewriting choice, *e.g.,* the smaller models Llama-3.2 [11] (`Llama-3.2-3B-Instruct`) and Phi-3 [1] (`Phi-3-mini-4k-instruct`) that are loaded in full precision. Due to the complexity of brief and ungrammatical FACAD descriptions, for FACAD cross-dataset setting we refine the prompt to maintain a similar style and description length:

> p_r: Briefly rewrite in a discriminative way to focus on shape, patterns, color (if indicated) and garment details, preserve all information. Use a single sentence with description length and writing style similar to original descriptions. Remove information on how to pair and complement. Strictly remove suggestions on occasions when to wear it.

and present results relying on a Int4-quantized `Phi-3-medium-4k-instruct`, based on empirical observations.

Tab. 1 and Tab. 2 reports qualitatives on language rewriting for DeepFashion and FACAD, respectively. As can be noted, while for DeepFashion language rewriting allows to focus on concrete-information about the fashion item of interest, FACAD rewriting mostly focus on correcting the description grammar-wise.

C. Additional Analysis

In this section we present additional analysis on our presented approach ACT. Sec C.1 explores the representation shift that occurs, while Sec. C.2 presents a quantitative evaluation of the FACAD cross-dataset analysis.

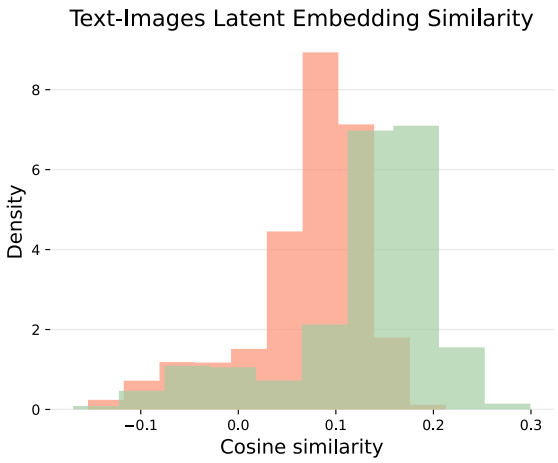

Figure 4. Density distribution of the cosine similarity with respect to the ground truth image embeddings of original abstract queries embeddings and ACT ones.

C.1. VLM latent space

We investigate the latent transformations induced by ACT on representations of abstract textual descriptions. To as-

Original query (q^A)	This floral print maxi dress features a front slit and self-tie crossback straps. Team this piece with flat sandals and sunnies for a warm weather look that all ease.
Language rewritten (q')	A floral print maxi dress with a front slit and self-tie crossback straps.
Original query (q^A)	Crafted from a stretchy ribbed knit with a racerback design and the text "Where There's a Will, There's a Way", this sporty tank is a shortcut to street style that makes a statement (literally). Plus, its slightly cropped cut means it's the cute partner in crime your high-waisted bottoms have been missing.
Language rewritten (q')	A sporty tank top with a racerback design and a printed message, made from stretchy ribbed knit, featuring a slightly cropped cut.
Original query (q^A)	There are certain pieces that will always bring a boho style aesthetic to mind, and this boxy top is one of them. It's crafted from an open floral crochet with a scalloped hem and short sleeves. We're showcasing the circle crochet trim along the round neckline with a turquoise necklace to really knock it out of the park.
Language rewritten (q')	A boxy top with an open floral crochet pattern, scalloped hem, and short sleeves, featuring a circle crochet trim along the round neckline.
Original query (q^A)	The easiest way to take-on a rushed morning starts with this heathered top! Cut from an ultra-lightweight fabric into a slouchy fit, make this long-sleeved piece will be your next go-to. With it's classic design, this layer will seamlessly work into any outfit for a relaxed aesthetic.
Language rewritten (q')	A slouchy, long-sleeved top made from ultra-lightweight fabric.
Original query (q^A)	Cookie-cutter so isn't your style. This muscle tee gets you with its "I'm Original" graphic, raw-cut hem, and cutout back. Rock a lacy bandeau underneath and put your unique spin on the look.
Language rewritten (q')	A muscle tee featuring an "I'm Original" graphic, a raw-cut hem, and a cutout back.

Table 1. Examples of input abstract-oriented descriptions and their corresponding LLM rewritten descriptions, using `Llama-3.1-8B-Instruct` on DeepFashion. All inferences rely on the same p_r prompt, see Sec. B.3.

Original query (q^A)	chewed hem amp up the vintage vibe of these high waisted stretch denim jeans featuring 3d whisker minor abrasion and hand sanding through the thigh
Language rewritten (q')	high-waisted stretch denim jeans with a chewed hem, 3D whisker abrasion, and hand-sanded thighs for a vintage look.
Original query (q^A)	an ortholite footbed and cushion soft padding ensure all day comfort in a versatile chukka boot in a sharp silhouette
Language rewritten (q')	a versatile chukka boot with a sharp silhouette, featuring an ortholiste footbed and cushion soft padding for all-day comfort.
Original query (q^A)	ripped hem and heavily sanded highlight bring instant old favorite status to a vintage button up that s easy to wear layered or on it own
Language rewritten (q')	a vintage button-up shirt with a ripped hem and heavily sanded highlight, versatile for layering or wearing alone.
Original query (q^A)	a faux wrap top cut in a cool linen blend feature a wide sash at the waist that creates a pretty peplum effect
Language rewritten (q')	a faux wrap top in a cool linen blend with a wide sash at the waist for a peplum effect
Original query (q^A)	a slit at the front brings easy movement to this stretch denim skirt finished with a softly fraying hem
Language rewritten (q')	a stretch denim skirt with a front slit and a softly fraying hem

Table 2. Examples of input abstract-oriented descriptions and their corresponding LLM rewritten descriptions, using `Phi-3-medium-4k-instruct` on FACAD. All inferences rely on the same p_r prompt, see Sec. B.3.

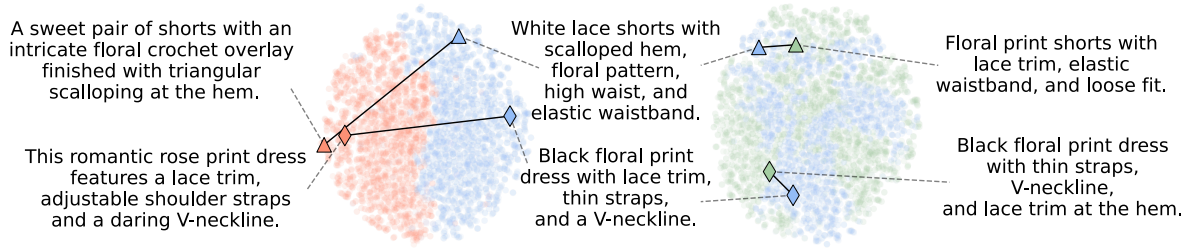

Figure 5. t-SNE of VLM latent space for textual embeddings in the evaluation set of DeepFashion. Connected markers denote different descriptions associated to the same data sample. **Left:** Abstract queries embeddings are separated from concrete ones, as those obtained from captioning models, `Qwen2-VL` in this case. **Right:** ACT allows to reduce the separation in the latent space.

sess this, we compute the cosine similarity between embeddings of original abstract descriptions and their corresponding ground truth visual representations within a shared latent space. These similarities are then compared to those obtained after applying the ACT transformation to the original abstract descriptions, see Fig. 4. The results demonstrate a clear improvement in alignment, as evidenced by a shift in the similarity distribution toward higher values relative to the baseline. This suggests that ACT enhances the semantic correspondence between text and images.

To further examine the structure of the latent space, we employ t-SNE [33] to visualize the embeddings of textual descriptions of the retrieval VLM. Figure 5 contrasts the original and shifted embeddings with those associated with concrete representations derived from a captioning model. The visualization reveals that, while embeddings from abstract descriptions are initially separated from their concrete counterparts, the ACT-transformed embeddings exhibit improved mixing. This blending demonstrates the establishment of a shared representation space where textual representations of abstract and concrete descriptions overlap. To show that ACT effectively shifts in the language space, we report the textual descriptions corresponding to marked embeddings, where we can observe a clear coherence in concrete terms. The results indicate that ACT effectively bridges the gap between abstract and concrete descriptions of the same fashion data, leading to unified multimodal understanding.

C.2. Cross-dataset evaluation with FACAD

We conducted additional analysis on a cross-dataset setting where models are trained or utilize data from DeepFashion and are evaluated on a 5K subsample of FACAD dataset. Tab. 3 shows the retrieval performance of zero-shot models, models fine-tuned on DeepFashion and two different variants of the proposed ACT. On one side, `ACT-df` considers both the language rewriting and representation shift components of the strategy. On the other, `ACT-df w/o LR` evaluates the representation shift without language rewriting. The results indicate that the shift characterization effectively bridges the representation of abstract-oriented descriptions, achieving an average improvement of +1.2% compared to the best-performing zero-shot model. The strategy consistently achieves second-best results falling behind the fine-tuned SigLIP model. However, challenges arise when applying large language models (LLMs) for language rewriting, as the performance deteriorate *w.r.t* the zero-shot model. A further qualitative investigation, shown in Fig. 6, reveals that the concise and grammatically intricate nature of the original FACAD descriptions (21 words on average per description in FACAD, compared to 53 words on average in DeepFashion) often leads LLMs to hallucinate or lack critical information, reducing their effectiveness in bridging the shift between concrete-oriented and abstract-oriented descriptions. This limitation not only impacts performance but also opens promising research directions in the natural language processing fields, particularly in designing LLMs better equipped in understanding and processing attribute-rich, concise text.

Model	R@1	R@5	R@10	H@1	H@5	H@10
Zero-Shot						
SigLIP	0.085	0.291	0.404	0.441	0.678	0.766
CLIP	0.022	0.073	0.109	0.107	0.247	0.333
O-CLIP	0.064	0.207	0.292	0.332	0.576	0.671
EVA-CLIP	0.054	0.176	0.254	0.277	0.494	0.597
F-CLIP	0.063	0.206	0.288	0.322	0.554	0.654
OF-CLIP	0.058	0.183	0.259	0.295	0.529	0.641
Cross-Dataset (DeepFashion → FACAD)						
SigLIP-ft-df	**0.093**	**0.322**	**0.438**	**0.482**	**0.708**	**0.786**
CLIP-ft-df	0.023	0.076	0.116	0.115	0.257	0.349
O-CLIP-ft-df	0.066	0.213	0.299	0.341	0.584	0.680
EVA-CLIP-ft-df	0.051	0.171	0.244	0.265	0.477	0.576
F-CLIP-ft-df	0.063	0.207	0.289	0.328	0.559	0.649
OF-CLIP-ft-df	0.062	0.201	0.282	0.316	0.570	0.666
ACT-df (Ours)	0.082	0.284	0.388	0.426	0.655	0.743
ACT-df w/o LR (Ours)	<u>0.089</u>	<u>0.303</u>	<u>0.413</u>	<u>0.460</u>	<u>0.694</u>	<u>0.776</u>

Table 3. Results on FACAD. In **bold** the best results, while <u>underlined</u> are the second best. ACT proves to be second best method *w.r.t.* zero-shot and fine-tuned models in the cross-dataset setting where DeepFashion is used for fine-tuning or database construction.

Figure 6. Some failure cases of ACT on FACAD. On the left, we report failures due to the generic FACAD descriptions: despite ACT retrieving fitting candidate images, the retrieval is considered a failure because the exact item was not presented. On the right, we report failures due to language rewriting hallucinations, highlighting the hallucinated elements.

D. Qualitatives results

In this section, we provide some qualitative results of text-to-image retrieval using ACT. The queries, specifically designed to qualitatively evaluate the retrieval performance of ACT, are manually crafted to contain a single abstract attribute, and as such, ground truth annotations are not available. In Fig. 7, we illustrate some query specializations across different types of t-shirts. ACT can properly leverage the abstract attributes available (*e.g.,* "subtle colors" *vs.* "happy colors") to present suitable candidates during retrieval. In Fig. 8, "serious neckline" results in blouses with a tight or high collar; meanwhile, when queried with a "relaxed neckline", ACT presents candidate images with large and soft v-necklines. Similarly, when asked to retrieve a dress with a "modern silhouette", ACT retrieves sleeveless short dresses with a tight fit. A "traditional silhouette", on the other hand, leads to mostly sleeved dresses with typically longer, airy skirts. In Fig. 9, we further demonstrate that ACT can interpret styles precisely: when queried for a "casual dinner dress", ACT presents mono-colored one-piece dresses that is uncluttered with little-to-no accessories; a "formal dinner dress", while also featuring mono-colored, results in mostly maxi-length dress with sophisticated details, such as asymmetrical hemline, tailored cuts with emphasis on the waist and decorative belts. Likewise, in another example of querying "sweet" cardigans and "edgy" cardigans: the former retrieves cardigans featuring a looser fit, particularly in the sleeves; while results from the latter are mostly dark-colored with tight sleeves.

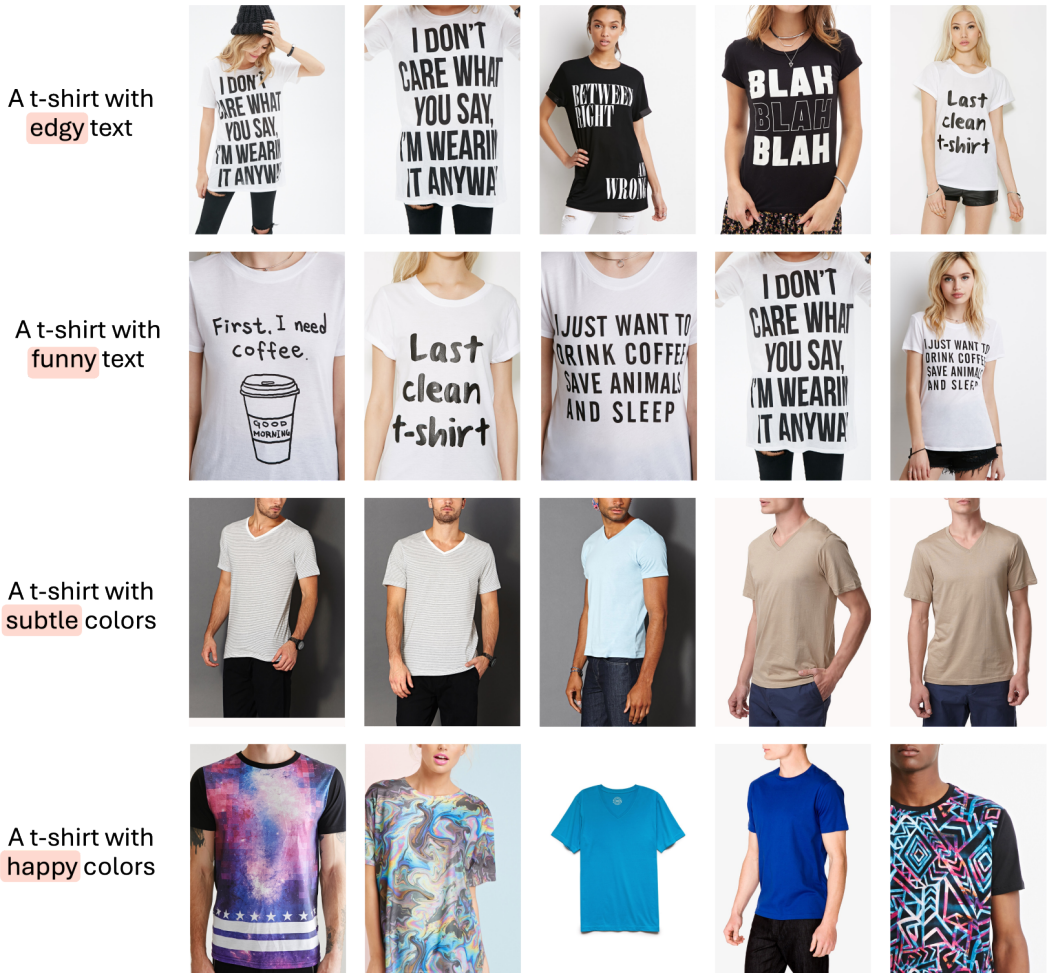

Figure 7. Qualitative examples of retrieval using our ACT, on the test split of DeepFashion, using abstract attributes. In this example, we provide specialization over different types of t-shirts. As demonstrated, ACT allows us to specialize our retrieval by changing the abstract adjectives used (*e.g.,* "subtle colors" vs. "happy colors").

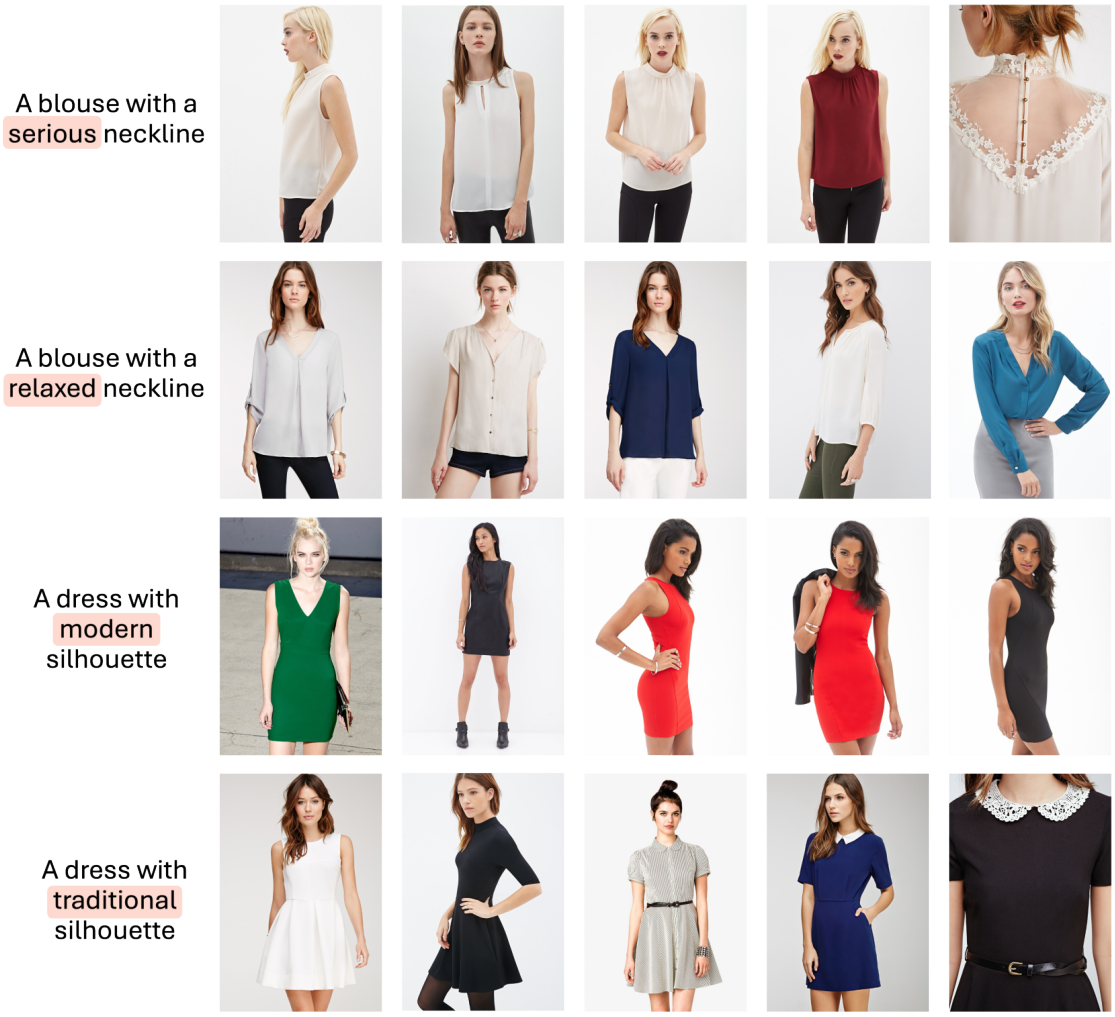

Figure 8. Further qualitative examples of retrieval using our ACT, on the test split of DeepFashion, using abstract attributes. In the top two rows, ACT differentiates between necklines, presenting candidate images with tight or high collars for "serious" styles, and large, soft v-necklines for "relaxed" ones. In the bottom two rows, ACT retrieves sleeveless, short and fitted dresses for a "modern silhouette", and mostly sleeved dresses with typically longer, airy skirts for a "traditional silhouette"

Figure 9. Further retrieval results from our ACT on the test split of DeepFashion showcase that ACT can also interpret abstract global details such as styles well. In the top two rows, ACT distinguishes between "casual" and "formal" dinner dresses, with the former being mono-colored and minimalist, and the latter featuring mostly maxi-length and more sophisticated design. In the bottom two rows, a similar distinction is made between "sweet" and "edgy" cardigans, with the former having a looser fit and the latter being darker with tighter sleeves.