Supplementary Material: **Discovering the Spatial Extent of Relative Attributes**

Paper ID: 1639

In this document, we (1) show additional qualitative examples of our discovered visual chains, (2) provide more detailed analysis comparing our approach to the detection-only baseline that uses only the detection-term and not local-smoothness to discover the chains, and (3) show qualitative examples of attribute editing on faces.

1. Qualitative examples of discovered visual chains

Along with this pdf file, we include two html files that display our discovered visual chains for each attribute in LFW10 and UTZap50k, respectively. We show the single top-ranked one, as measured by ranking accuracy on the validation set; see Eqn. 3 in the main paper. Each chain has 240 images, and is ordered according to predicted attribute strength using the ranker trained on the image patches in the chain.

LFW10: ./chains_LFW10/chains.html (or you can manually access chains_LFW10/chains.html)
UTZap50k: ./chains_utzap50k/chains.html (or you can manually access chains_utzap50k/chains.html)

Our chains are visually-coherent, even when the appearance of the underlying visual concept changes drastically over the attribute spectrum. For example, for the attribute "Comfort" in UTZap50k, the top-ranked visual chain consistently captures the heel-part of the shoe, even though its appearance changes significantly across the attribute spectrum. Due to the precise localization of the attribute, we are able to learn an accurate ordering of the images. Note that while here we only display the top-ranked visual chain, our final ensemble image-representation combines the localizations of the top-60 ranked chains to discover the full spatial extent of the attribute (see Fig. 5 in the main paper).

2. Contribution of local-smoothness

In this section, we expand upon our ablation studies on the contribution of local-smoothness (Sec. 4.3 in the main paper). We find that the attributes in UTZap50k are more difficult to localize than those in LFW10, since their appearance varies more substantially over the attribute spectrum. Therefore, we focus on UTZap50k, and compare the top-3 ranked visual chains discovered by our approach (i.e., detection+local-smoothness) versus the baseline that does not use the local-smoothness term (i.e., detection-only). Specifically, we ask a human annotator to mark the number of outlier detections that do not visually agree with the majority detections in the chain discovered by each approach. This table shows the average number of outliers in the top-3 ranked visual chains per attribute on UTZap50k:

	Open	Sporty	Pointy	Comfort	Mean
Detection-only	28.3	**0**	12.7	**0.3**	10.3
Ours	**6.3**	**0**	**3.0**	**0.3**	**2.4**

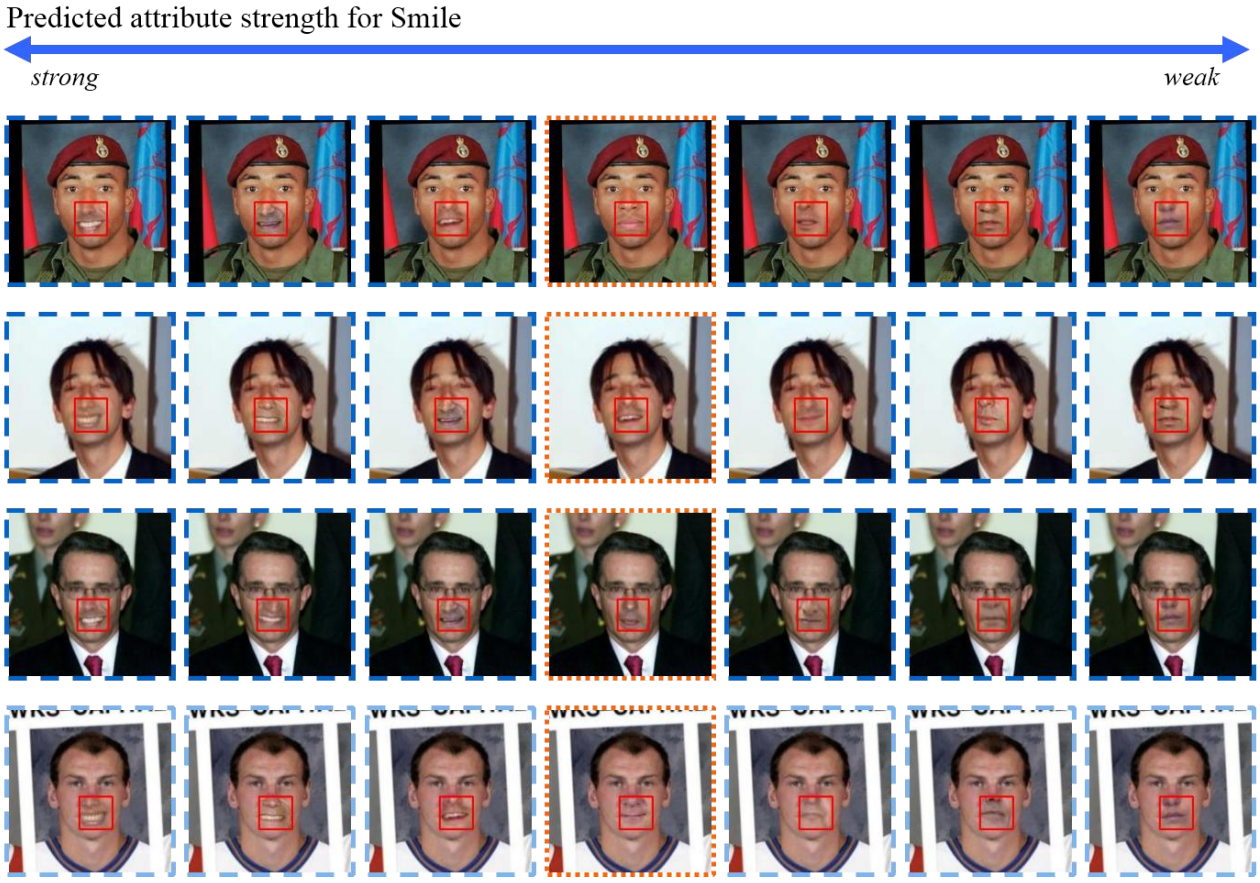
Table 1. Average number of outliers per chain in the top-3 visual chains for Ours vs. the Detection-only baseline.

Our approach produces significantly fewer outliers for "Open" and "Pointy" (for "Sporty" and

"Comfort", both our approach and the baseline produce equally few errors). Upon close inspection for "Open" and "Pointy", we find that the baseline approach tends to discover multiple-modes during iterative growing; i.e., the detector accumulates outlier patterns in each iteration, which eventually lead to multiple modes in the final chain. Our approach is more robust to the development of multiple-modes because the local-smoothness term propagates information from the good neighbors to correct most of the detection errors (see Fig. 7 in main paper). Thus, in our approach, error accumulation is minimized over each iteration of chain-growing. Note that local-smoothness can also propagate errors if the neighboring patches are incorrect. However, since there are many more good detections than bad ones, we find that errors are often corrected rather than propagated.

3. Attribute editor for faces

Finally, we show additional results of attribute editing. In addition to the shoe editing results that we show in Section 4.4 of the main paper, we also attempted to edit faces in LFW10. Here are some example editing results for the "Smile" attribute:



The middle column shows the query image whose attribute (automatically localized in red box) we want to edit. With the automatically localized attribute regions, we can edit how "smiley" someone is by only editing the appearance of the mouth region. From left to right, the images change gradually from very smiley to not-smiley while keeping the person's identity unchanged.