Face Image Retrieval with Attribute Manipulation: Supplementary Materials

Alireza Zaeemzadeh
University of Central Florida
zaeemzadeh@eecs.ucf.edu

Shabnam Ghadar
Adobe Inc.
ghadar@adobe.com

Baldo Faieta
Adobe Inc.
bfaieta@adobe.com

Zhe Lin
Adobe Inc.
zlin@adobe.com

Nazanin Rahnavard
University of Central Florida
nazanin@eecs.ucf.edu

Mubarak Shah
University of Central Florida
shah@crcv.ucf.edu

Ratheesh Kalarot
Adobe Inc.
kalarot@adobe.com

1. Implementation Details

Query generation: For CelebA [3] dataset, the standard testing set is used to both generate the queries and as the gallery set. For the synthetic dataset, the latent space of the StyleGAN [2] is sampled to produce the $100,000$ images. In the main manuscript, the synthetic images are used for the qualitative evaluations, as the gallery set is much larger.

To generate the queries, we randomly select 1000 images from the gallery set as the query face. We make sure that, after changing one attribute in these query images (the query attribute), there is at least one *similar* image the gallery set. Here, we use the ground truth attributes to define *similarity*. For our experiments, we consider two images similar if they have the exact same ground truth attribute values. Then, we use the query face and the query attribute to create either the modification vector (used by the GAN-based methods) or the modification text (used by the compositonal leaning methods). Out of 40 attributes in the CelebA data, 5 attributes are not related to facial features and are removed. These attributes are: `blurry`, `necktie`, `earrings`, `hat`, and `necklace`. Furthermore, to generate modification text and to generate queries, the attributes that describe the same feature are considered as one attribute. For example, CelebA contains ground truth for *black hair*, *brown hair*, *blonde hair*, and *grey hair*. We consider these four attribute as one, when generating the queries. Here are some example query modification texts: `add/remove eyeglasses`, `make hair black/brown/blonde/grey`, `make face young/old`, `add/remove hair`, `add/remove smile`, and `change gender to male/female`.

To generate the modification vector for our method, we just set the corresponding entry to 0 or 1. We use binary modification vector in our experiments for a fair comparison with the text based methods. However, our method is capable of accepting any value between 0 and 1 for the modification vector, which will be illustrated shortly.

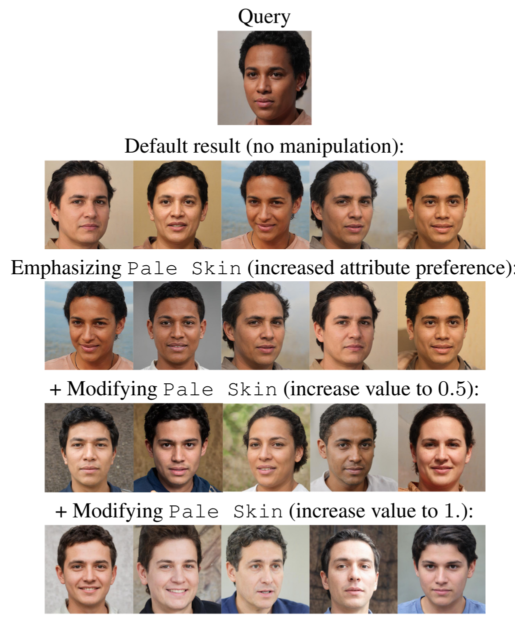
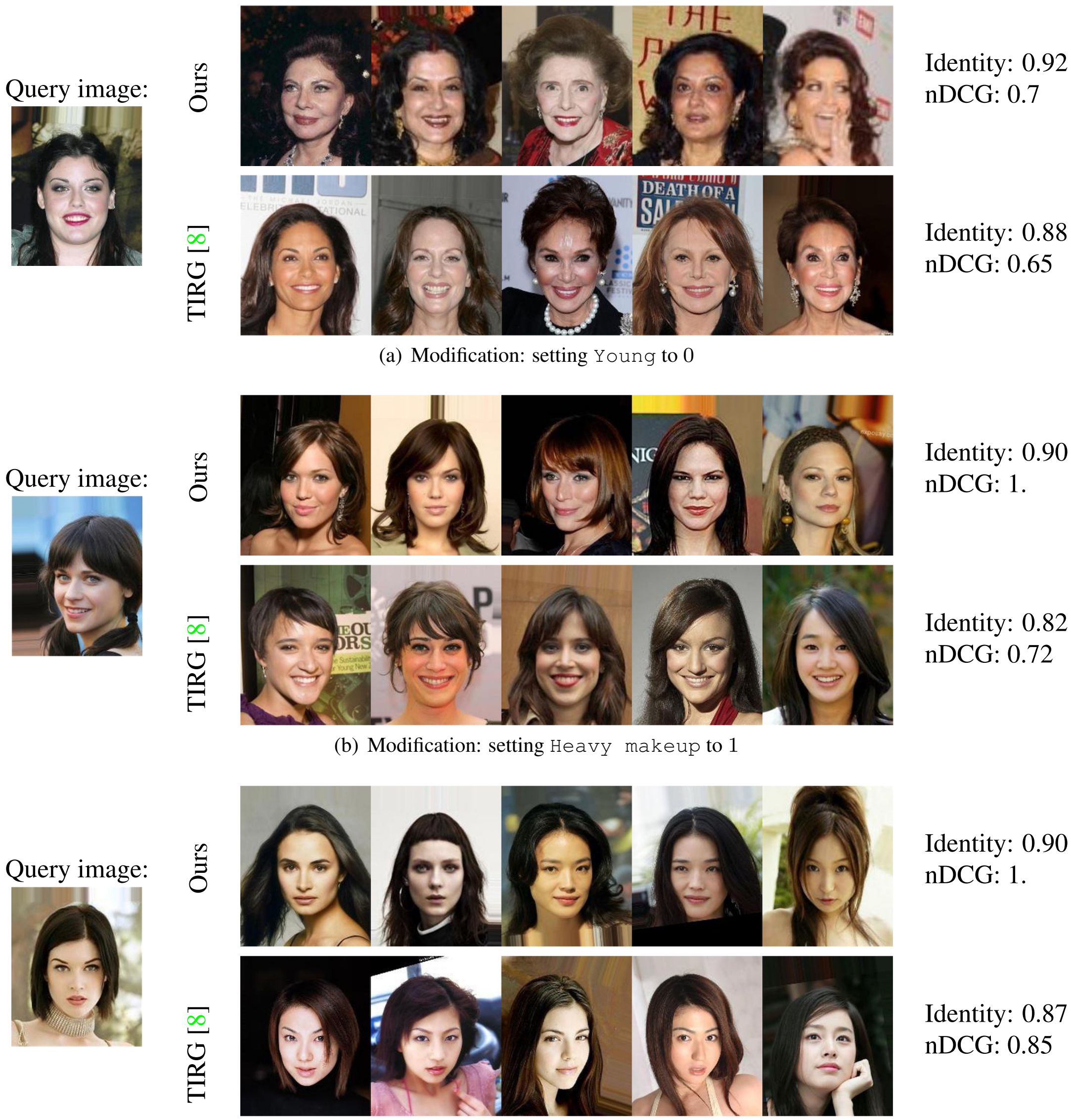


Figure 1. An example of modifying the retrieval results using continuous, real-valued, modification vector. The attribute intensity for `Pale Skin` for the query face is estimated as 0.12. The user is able to modify the results by increasing it to 0.5 and then to 1.

To retrieve images using the method in [4], we first use the image embeder to embed the query face. Then, the attribute operator corresponding to the attribute being adjusted is applied to obtain the modified query. The closest faces to this modified query vector in the gallery set are then retrieved and sorted using their Euclidean distance. For the feature extractor, which is a building block of the image embeder architecture in [4], we use Inception Resnet V1 architecture, as described in [7] and trained on VGGFace2 [1].

Training: For the compositional learning baselines, the full training set is used. For each training image, we generate all the possible query modification texts, as discussed earlier. All these possible queries are used to train the model



(a) Modification: setting `Young` to 0

(b) Modification: setting `Heavy makeup` to 1

(c) Modification: setting `Mouth slightly open` to 0

Figure 2. Examples of retrieved images by our method and the compositional learning method in [8] and their corresponding nDCG and identity similarity. (a) Changing the attribute `Young` to 0, (b) Changing the attribute `Heavy makeup` to 1, and (c) Changing the attribute `Mouth slightly open` to 0. In all of these examples, our method outperforms the baseline in both the evaluation metrics. Qualitatively, the retrieved images by method can modify the attribute, while preserving the other attributes, such as skin tone, hair color, smiling, etc, better.

using the code provided by the authors. On the other hand, for our method, we use a subset of CelebA training set and its corresponding attribute ground truth to obtain the attribute direction in a pretrained StyleGAN. For that, we first select a subset of images such that we have both positive and negative for all the attributes. Then, the selected samples are encoded onto the latent space using the encoder proposed in [5]. Then the latent vectors are used to obtain the sparse and orthogonal attribute directions as proposed in the main manuscript. The same number of samples and same encoder are used to extract the attribute directions for the GAN-based baseline [6], using the code provided by the authors.

2. Additional Experiments

Figure 1 illustrates a retrieval example using synthetic images and real-valued modification vector, as opposed to binary. In this example, the user is modifying the attribute `Pale Skin`. The estimated intensity of this attribute in the query is 0.12, but the user is able to modify the retrieval results by increasing it to 0.5 or 1. Here, we have first emphasized this attribute in the results, by increasing the preference value, to make the changes in attribute intensity more dominant. This example shows how our method can successfully utilize a modification vector to manipulate the results in a continuous manner, a capability which modification text cannot provide.

To compare the retrieved images using our method and the baseline in [8], Figure 2 shows a few examples of retrieved images and their corresponding performance metrics using the CelebA dataset and after modifying an attribute. For a fair comparison with the text-based baseline, we only use binary values as the modification for this experiment. For example, in Figure 2(a), we want to retrieve images similar to the query images, while changing the value for attribute `Young` to 0. Note that our method is able to preserve most the other attributes, such as skin tone, hair color, makeup, smiling, etc, while being able to modify the specified attribute, i.e. age. Similarly, for the other examples, the retrieved images by our method are more similar to the query images and to the other retrieved images, both in terms of identity and facial attributes. We argue that this because the latent space of GAN contains all the necessary information necessary to reconstruct the image, while the embedding space generated by the compositional learning methods does not need to satisfy such requirement. Also, our method is able to disentangle the attributes more effectively and can modify an attribute, while preserving other attributes and the identity.

To illustrate this, Figure 3 and Figure 4 show a few examples of editing multiple attributes in faces using the obtained attribute directions for synthetic and real faces, respectively. To achieve this, the latent vector corresponding to the starting point face, marked with the red square, is moved along two attribute directions. these figures show that our obtained attribute directions are more disentangled, compared to the method proposed in [6]. For example, in Figure 3, attributes `Pale Skin` and `Chubby` affect the attribute `Smiling` in faces edited using the baseline directions, an artifact that is not present in faces edited by our obtained directions. Furthermore, in Figure 3(b), manipulating the attribute `Black Hair` using the method in [6] affect the identity. The difference is even more apparent for real faces, Figure 4, where the baseline modifications lead to a lot more artifacts and more impact on the identity, compared to ours.

The quantitative results presented in Table 1 in the main manuscript also suggest that the directions obtained by our method are more disentangled compared to [6], as our method is able to consistently achieve better nDCG, while having similar or better identity similarity. This means that our sparse attribute directions affect the identity and other attributes less. We argue that this is due to the fact that the

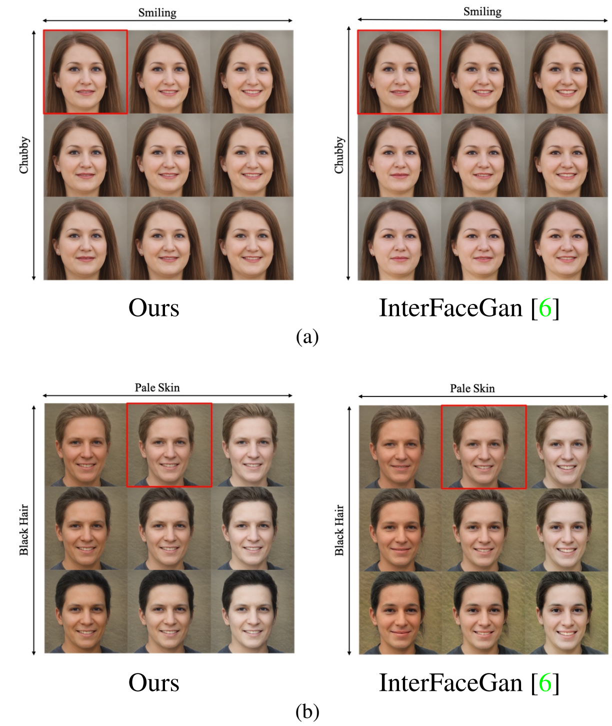


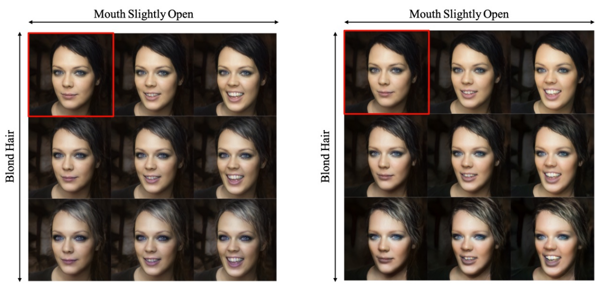
Figure 3. Attribute manipulation results using our method and the method proposed in [6] on two synthetic images. The latent vector corresponding to the starting point, marked with red square, is gradually moved along to different attributes' directions. Notice the impact of adjusting attributes `Chubby` and `Pale skin` on the smile in images edited using [6].

direction obtained by our method are sparse, meaning that they affect as few entries in the latent vector as possible. This encourages the learned directions to only affect the entries that are most relevant to their corresponding attribute. Figure 6 in the main manuscript shows how the energy of the sparse attribute directions are concentrated on a small percentage of the entries. Similarly, Figure 5 in this document shows the how the energy of the attribute vector is distributed across the layers. The energy ratio for each layer is calculated as the ratio of the ℓ_2 norm of the latent vector in that layer to the overall norm, i.e., $\frac{\|\boldsymbol{w}_l\|_2}{\|\boldsymbol{w}^+\|_2}$. For example, the energy of the attribute vectors that only affect color of skin or hair, `Black Hair` and `Pale Skin`, is mostly concentrated in the layers that are responsible for fine features of the face, i.e., the last few layers of the synthesis network. On the other hand the attribute vectors that affect the coarse structures in the face, such as `Eyeglasses`, `Bangs`, `Baldness`, `Smiling`, etc, are mainly concentrated in the first few layers.

Finally, Table 1 compares the GAN-based methods' performance, in terms of nDCG and identity similarity, for different number of training samples used to obtain the attribute directions. Our proposed method is consistently more data-efficient compared to the baseline. This can be

Table 1. Normalized discounted cumulative gain (nDCG) and identity similarity for the GAN-based methods using different number of training faces to obtain the attribute directions, averaged over 1000 queries. Here we are calculating the metrics on the top-5 images

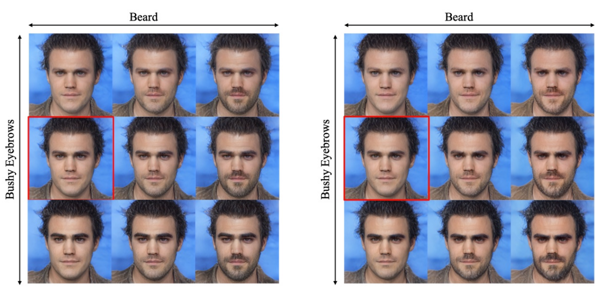
Number of training samples	3,500		14,000		20,000	
Method	nDCG	Identity Similarity	nDCG	Identity Similarity	nDCG	Identity Similarity
InterFaceGAN [6] (Identity constrained)	0.79	0.817	0.81	0.860	0.82	0.859
Ours (Identity constrained)	0.82	0.830	0.83	0.864	0.85	0.864
InterFaceGAN [6] (best nDCG)	0.83	0.831	0.88	0.839	0.90	0.841
Ours (best nDCG)	0.85	0.840	0.90	0.847	0.92	0.848



Ours

InterFaceGan [6]

(a)



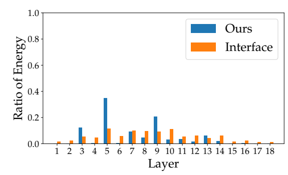
Ours

InterFaceGan [6]

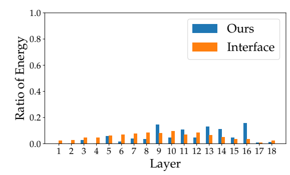
(b)

Figure 4. Attribute manipulation results using our method and the method proposed in [6] on two images from CelebA dataset. The latent vector corresponding to the starting point, marked with red square, is gradually moved along to different attributes' directions. The obtained directions by the baseline leads to more artifacts compared to the directions obtained by our method.

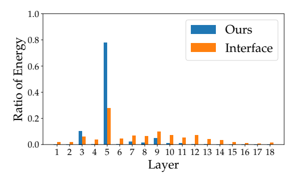
due to the fact that we enforce both orthogonality and sparsity constraints during the training, which makes the solution space much smaller. Also, comparing the results with Table 1 in the main manuscript, our proposed method can compete with the compositional-learning methods even with only $3,500$ training samples, while these baselines use the full training set, containing more than $160,000$ samples.



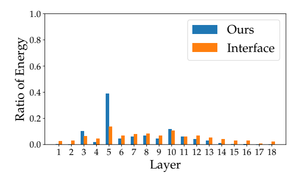
(a) Baldness



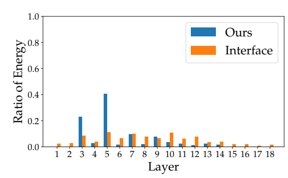
(b) Black Hair



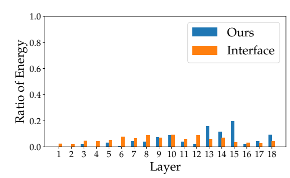
(c) Eyeglasses



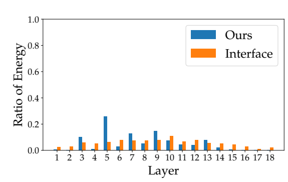
(d) Smiling



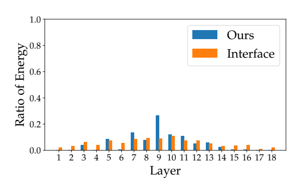
(e) Bangs



(f) Pale Skin



(g) Chubby



(h) Mustache

Figure 5. Ratio of ℓ_2 norm of different attribute vectors in each layer over the total ℓ_2 norm of the vector, for our method and InterFaceGAN [6]. Our method often concentrates most of the energy of the vector in a few layers. For example, vectors corresponding to `Bangs` and `Baldness` have a similar energy profile and only manipulated the layers corresponding to the coarse structures, i.e., first few layers. On the other hand, vectors corresponding to `Black hair` and `Pale skin` mainly change the last few layers, which are responsible for finer structures in the face.

References

[1] Qiong Cao, Li Shen, Weidi Xie, Omkar M. Parkhi, and Andrew Zisserman. VGGFace2: A dataset for recognising faces across pose and age. In *Proceedings - 13th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2018*, 2018. 1

[2] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving

the image quality of stylegan. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2020. 1

[3] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015. 1

[4] Tushar Nagarajan and Kristen Grauman. Attributes as Operators. *European Conference on Computer Vision*, 2018. 1

[5] Elad Richardson, Yuval Alaluf, Or Patashnik, Yotam Nitzan, Yaniv Azar, Stav Shapiro, and Daniel Cohen-Or. Encoding in Style: a StyleGAN Encoder for Image-to-Image Translation. *arXiv preprint arXiv:2008.00951*, 2020. 2

[6] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of GANs for semantic face editing. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2020. 2, 3, 4

[7] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. Inception-v4, inception-ResNet and the impact of residual connections on learning. In *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, 2017. 1

[8] Nam Vo, Lu Jiang, Chen Sun, Kevin Murphy, Li Jia Li, Li Fei-Fei, and James Hays. Composing text and image for image retrieval-An empirical odyssey. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019. 2, 3