Face Image Retrieval with Attribute Manipulation

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

Abstract

Current face image retrieval solutions are limited, since they treat different facial attributes the same and cannot incorporate user's preference for a subset of attributes in their search criteria. This paper introduces a new face image retrieval framework, where the input face query is augmented by both an adjustment vector that specifies the desired modifications to the facial attributes, and a preference vector that assigns different levels of importance to different attributes. For example, a user can ask for retrieving images similar to a query image, but with a different hair color, and no preference for absence/presence of eyeglasses in the results. To achieve this, we propose to disentangle the semantics, corresponding to various attributes, by learning a set of sparse and orthogonal basis vectors in the latent space of StyleGAN. Such basis vectors are then employed to decompose the dissimilarity between face images in terms of dissimilarity between their attributes, assign preference to the attributes, and adjust the attributes in the query. Enforcing sparsity on the basis vectors helps us to disentangle the latent space and adjust each attribute independently from other attributes, while enforcing orthogonality facilitates preference assignment and the dissimilarity decomposition. The effectiveness of our approach is illustrated by achieving state-of-the-art results for the face image retrieval task.

1. Introduction

The problem of image retrieval has been studied in many different applications, such as product search [31, 32] and face recognition [23]. The standard problem formulation for image to image retrieval task is, given a query image, find the most similar images to the query image among all the images in the gallery. However, in many scenarios, it is necessary to *improve* and/or *adjust* the retrieval results by incorporating either the user's feedback or by augmenting the query. This is due to the fact, in many cases, a perfect query image may not be readily available. Thus, it is desirable to give the user more control over the results. For example, in the context of fashion products, authors in [32, 13] exploit the user's feedback to refine the search results iteratively. For instance, the method in [32] asks the user a series of visual multiple-choice questions to refine the search results and to eliminate the semantic gap between the user and the retrieval system. Another parallel approach is to augment the query with additional information, e.g., adjustment text, to modify the search results [29]. This is most often done by mapping the multi-modal query onto a joint embedding space [8, 33, 29]. These approaches treat different semantics the same and cannot prioritize a subset of attributes. Thus, the user is not able to define a customized distance metric and to assign importance to the attributes.

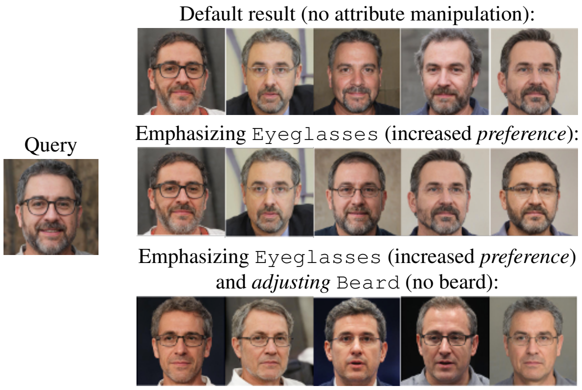
Default result (no attribute manipulation):
Query
Emphasizing Eyeglasses (increased *preference*):
Emphasizing Eyeglasses (increased *preference*)
and *adjusting* Beard (no beard):

Figure 1. Example of face image retrieval by considering both the attribute *adjustment* and attribute *preference* specified by the user.

In this work, we introduce a new formulation for the image search task in the context of face image retrieval; and augment the query with both an adjustment vector and a preference vector. The **adjustment vector** is used to change the *presence* of certain attributes in the retrieved images, and the **preference vector** is used to assign the *importance* of the attribute in the results. To the best of our knowledge,

[*] Work done as part of an internship at Adobe Inc.

this is the first work that can simultaneously adjust the attributes and assign preference values to them. Employing a preference vector gives the user the ability to customize the similarity criteria. For instance, having eyeglasses might be more important to the user than having the same hair color. This criteria cannot be specified using only the adjustment vector, which is a limitation of existing retrieval methods. On the other hand, adjustment vector enables the user to use an imperfect query image for the search and adjust the attributes to achieve the ideal results. Furthermore, employing an adjustment vector, as opposed to an adjustment text, provides us with more flexibility, as many facial attributes cannot be easily described in text, for example different shades of brown hair. In the example provided in Figure 1, the impact of assigning a larger preference value and adjusting attributes are illustrated. In the middle row, the user has emphasized the attribute `Eyeglasses`, by assigning a larger *preference* value to it, which leads to all the top-5 retrieved images containing eyeglasses. The user can further fine-tune the results by *adjusting* any subset of the attributes. The bottom row shows the retrieved images after both emphasizing the attribute `Eyeglasses` and adjusting the attribute `Beard`, as a result the beard has been removed and the eyeglasses are still present.

To achieve this, we employ the recent advancements in generative adversarial networks (GANs). It is shown that different semantic attributes are fairly disentangled in the latent space of StyleGAN [12, 11], even if the generator is trained in an unsupervised manner. This has been studied and experimentally verified in [12, 25]. This property provides us with an array of desirable features for face image retrieval. First, since the generator can be trained in an unsupervised manner, we do not need to have access to a lot of labeled data. A fairly small set of labelled data can be utilized to interpret the latent semantics learned by the generator. Second, the latent space provided by a well-trained StyleGAN provides us with an opportunity to both adjust the attributes and to assign preference to them. In that context, we propose to obtain a set of disentangled attribute vectors in the latent space of StyleGAN. To disentangle the obtained attribute vectors, we enforce both *orthogonality* and *sparsity* constraints on them. We argue that, by making the attribute vectors sparse, we can decouple the entangled attributes even further. This is due to the fact that such attribute vectors can manipulate their corresponding semantic by affecting only a small subset of entries of the latent vector. This promotes selectivity among both the entries of the latent vector and the layers of the generator of the StyleGAN. On the other hand, by enforcing orthogonality, we can translate the dissimilarity between each image pair into dissimilarity between the attributes, assign preference to attributes, and define an attribute-weighted distance metric. In short, our contributions can be summarized as follows:

- We introduce a new face image retrieval framework that can simultaneously adjust the facial attributes and assign preference to different attributes in the retrieval task, employing the latent space of GANs (Section 3);
- We propose a new method to extract the directions of different attributes in the latent space, by learning all the attribute directions simultaneously and enforcing orthogonality and sparsity constraints (Section 3.1);
- We utilize the learned attribute directions to define a weighted distance metric, to manipulate semantic attributes of the query, and to assign preference to different attributes for retrieval (Section 3.2); and
- The proposed method for image retrieval outperforms the recent state-of-the-art methods that use compositional learning or GANs for search (Section 4).

2. Related Work

Attribute-guided face image retrieval: There are many different approaches for image retrieval task based on metric learning such as [30, 4, 21, 3, 16], however they do not consider the task of retrieval with attribute manipulation. More similar to our attribute-guided retrieval setup, several methods utilize a query image and augment it with either an attribute adjustment text [29, 31, 8, 33, 9] or vector [32, 1]. Some of the prior work focuses on dialog-based interaction between the user and the retrieval agent, and improving the results in an iterative manner through user's feedback [8, 32, 13]. Most of the attribute-aware retrieval methods *need huge amounts of labelled data* to generate a semantically meaningful latent space and distance metric [1, 33, 9, 31, 29, 10]. The method in [29] employs a new operation, referred to as residual gating, to create the joint embedding space between the image and text queries, which leads to state-of-the-are results among compositional learning methods such as [16, 28, 17, 18, 22, 19]. In contrast, we propose to leverage the recent advancements in GAN architectures [6, 11, 12] and use the latent space generated by a GAN *trained in an unsupervised manner*, which significantly relaxes the requirements of access to labelled data. Furthermore, to the best of our knowledge, there has been no image retrieval method that can simultaneously *adjust* the attributes and assign *preference* to them.

Learning semantics in the latent space of GANs: Recent work have shown that the real image data can be represented in the latent space of GANs, and specifically StyleGAN, with manifolds that have little curvature [24, 25, 12]. Such smooth behaviour can be enhanced by using loss functions [14, 12] or by modifying the generator architecture [11, 27]. A major benefit of the StyleGAN architecture [11] is the introduction of an intermediate latent space that does not need to follow any fixed sampling distribution, and the linear behaviour in this space is further enforced in [12] using path length regularization. It has been shown that this

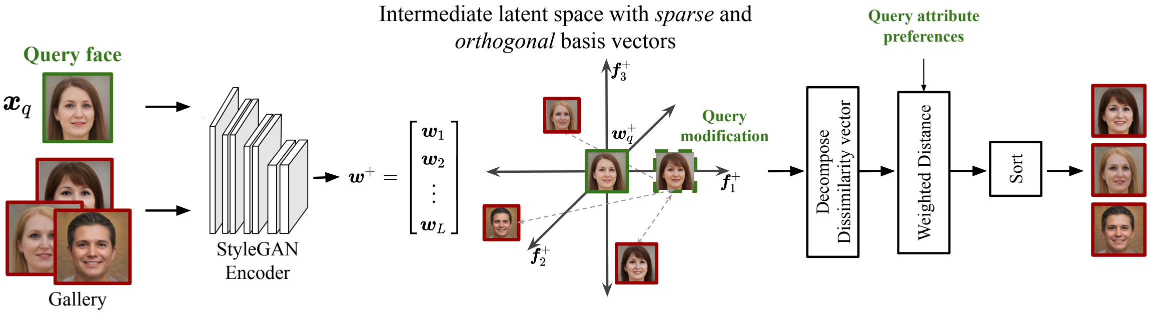
Query face
x_q
Gallery
StyleGAN Encoder
$w^+ = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_L \end{bmatrix}$
Intermediate latent space with *sparse* and *orthogonal* basis vectors
f_3^+
w_q^+
Query modification
f_1^+
f_2^+
Decompose Dissimilarity vector
Query attribute preferences
Weighted Distance
Sort

Figure 2. The overall architecture of the proposed face image retrieval framework. The intermediate latent space, $\mathcal{W}^+$, is generated by employing StyleGAN encoder proposed in [20]. Then, the orthogonal and sparse basis vectors $\{f_m\}_{m=1}^M$ are extracted using a fairly small set of face images with attribute annotations. Utilizing the basis vectors, we adjust the query, decompose the dissimilarity vectors, and assign preference to different attributes.

regularization leads to better Perceptual Path Length (PPL) score, which measures the perceptual score of the generated images after *linear* interpolation in the intermediate latent space. The authors in [25] employ this property and learn linear latent subspaces corresponding to different attributes. The authors in [25] proposed to orthogonalize the directions only during editing and in a sequential manner. This means that if the user wants to adjust multiple attributes, each new attribute direction is projected onto the null space of previous attributes. This approach has two main drawbacks. First, the final result depends on the order of applying the attribute adjustments. Second, the sequential orthogonal projection makes it more difficult to define an attribute-guided distance metric and make the image retrieval very computationally expensive. In contrast, we propose to learn the latent subspaces simultaneously, and enforce orthogonality on the subspaces during the learning process. Furthermore, we study the impact of enforcing sparsity on disentangling the attributes.

3. Our Approach

Assume we have a set of M predefined facial attributes. In this setting, the query can be defined as a triplet (x_q, a_q, p_q), where x_q is the query image, $a_q \in [0,1]^M$ is the vector specifying the intensity of each attribute (*attribute adjustment vector*), and $p_q \in \mathbb{R^+}^M$ is a vector containing positive real numbers indicating the *preference* for each attribute. The attribute adjustment vector (a_q) can be used to *adjust* the search query. For instance, if the user assigns an intensity of 0 to attribute `smiling`, the search results should not contain smiling faces, even though the query face is smiling. Also, the preference vector p_q is independent of the adjustment vector a_q, meaning that the value we assign as the preference value for each attribute does not depend on whether we are adjusting the attribute or not. The larger the preference value, the more similar the attribute should be to the query attribute. A preference value of 0 for a particular attribute means the user does not care about the presence/absence of that attribute. In this extreme case, the assigned attribute intensity will be ignored by the retrieval agent. The goal of our proposed framework is to rank the images in a gallery dataset based on the similarity with the query image, while considering both the *adjustments* and attribute *preferences* specified by the user.

To this end, we propose to perform the retrieval in the latent space of a StyleGAN [12]. This provides us with two desirable properties. First, as discussed in Section 2, it has been shown that different attributes can be manipulated fairly linearly in such a space [25, 12]. Second, using an unconditional StyleGAN gives us the opportunity to train it and its corresponding encoder using a large number of unlabeled data. We show how we can exploit a smaller number of labeled data to interpret the latent semantics learned by the StyleGAN.

The defining feature of StyleGAN architecture is the introduction of an intermediate latent vector, $w \in \mathcal{W}$. In short, the generator of the StyleGAN consists of two main components: a mapping network and a synthesis network. The mapping network transforms the input latent vector to the intermediate latent space $\mathcal{W}$. Then the intermediate latent vector w is used to modulate the convolution weights of the synthesis network, which generates the image. It has also been shown that this intermediate latent space is consistently more disentangled than the input latent space, meaning that the attributes can be classified using a linear classifier more accurately in $\mathcal{W}$ [11, 12]. Therefore, given a binary attribute, there exists a hyperplane in $\mathcal{W}$ that can separate the attribute classes. In other words, there exists a direction f, i.e., the direction orthogonal to the hyperplane, such that if we move the latent vector w along f, $w + \alpha f$, the class boundary can be crossed and the attribute can be turned to the opposite. Here α is a scalar which determines the displacement magnitude Such directions can be obtained by training a linear classifier in $\mathcal{W}$, using labelled data. We argue that if we obtain an orthogonal and sparse basis set in $\mathcal{W}$, where each basis vector corresponds to a single attribute, we can easily adjust the attributes and define a weighted distance metric to retrieve images.

The proposed retrieval framework can be summarized as follows. First, given a well-trained StyleGAN encoder trained on unlabeled data, a small set of labeled data (face images annotated with M attributes) are used to obtain an orthogonal basis set $\mathcal{F} = \{\boldsymbol{f}_m\}_{m=1}^{M}, \boldsymbol{f}_m \in \mathcal{W}, \forall m$, such that moving the latent vector along $\boldsymbol{f}_m$ only affects the m^{th} attribute (Section 3.1). Second, the obtained basis set $\mathcal{F}$ is used to adjust the attributes, to define a weighted distance metric in $\mathcal{W}$, and to retrieve images (Section 3.2). The overall framework is shown Figure 2. Below, we discuss each of these two steps in more details.

3.1. Extracting Orthogonal Basis Set for Disentangled Semantics

As mentioned earlier, it has been empirically verified that different facial attributes can be manipulated fairly linearly in the latent space of StyleGAN [24, 25, 11, 12]. However, when there is more than one attribute, the obtained directions may be correlated with each other, meaning that adjusting one attribute using its corresponding direction might affect other attributes as well. To tackle this issue, let us examine how the intermediate latent vector is utilized to generate images. The latent vector is transformed to generate *styles* for each convolution layer in the synthesis network, using an affine transform, i.e., $\boldsymbol{s}_l = A_l(\boldsymbol{w})$. Here, $\boldsymbol{s}_l$ stands for the style vector of l^{th} layer and $A_l(.)$ is the learned affine transform of the pretrained StyleGAN. Each entry in $\boldsymbol{s}_l$ is used to modulate the weights of a single convolution operator in the l^{th} layer. It has also been shown that instead of using a common latent vector $\boldsymbol{w}$ for all the layers, we can extend the latent space and improve the encoding performance by finding a separate latent vector for each layer $\boldsymbol{w}_l$ and producing the styles as $\boldsymbol{s}_l = A_l(\boldsymbol{w}_l)$. We refer to this space as the extended latent space $\mathcal{W}^+$ and represent the latent vector as the concatenation of layer-wise codes, $\boldsymbol{w}^+ = [\boldsymbol{w}_1^T, \boldsymbol{w}_2^T, \ldots, \boldsymbol{w}_L^T]^T \in \mathbb{R}^{d^+}$, and the attribute directions as $\boldsymbol{f}^+ \in \mathbb{R}^{d^+}$.

We argue that enforcing sparsity on the learned directions in $\mathcal{W}^+$ can effectively lead to disentangling the semantics and improved performance both for conditional image editing and the attribute-guided image retrieval. In other words, we look for attribute direction $\boldsymbol{f}^+ \in \mathcal{W}^+$ with minimum number of non-zero entries, while being able to classify the attributes accurately. This provides us with several advantages. First, it reduces the space of possible solutions and makes the learning problem more data-efficient. Thus, we are able to use a smaller set of labeled data to find the directions. Second, to manipulate the attribute in the latent space, $\boldsymbol{w}^+ + \alpha \boldsymbol{f}^+$, only a few entries of $\boldsymbol{w}^+$ are modified. Therefore, the learned direction $\boldsymbol{f}^+$ represents the minimum change necessary to manipulate the attribute. This leads to disentanglement of different attributes, as different attribute directions only modify a very small, probably non-overlapping, subset of the entries. Finally, enforcing sparsity on the filters learned in extended latent space $\mathcal{W}^+$ also encourages non-uniform modification of the latent vectors across layers, as most of the entries are zeros. This is significant because the first few layers generate coarse details and later layers generate the finer details. Modifying a subset of layers means that the method is able to manipulate only the detail levels that are relevant to the attribute, leading to better disentanglement and accuracy.

Algorithm 1 Finding Nearest Orthogonal Set to a Set of Vectors.

```
Input: A set of vectors $\{\boldsymbol{f}_m\}_{m=1}^{M}$
$c_m = \|\boldsymbol{f}_m\|_2, \forall m$
Create a matrix $\boldsymbol{F}$ whose columns are $\boldsymbol{f}_1/c_1, \boldsymbol{f}_2/c_2, \ldots, \boldsymbol{f}_1/c_M$
Compute $\hat{\boldsymbol{F}} = \boldsymbol{F}(\boldsymbol{F}^T\boldsymbol{F})^{-\frac{1}{2}}$
return $\{c_m\hat{\boldsymbol{f}}_m\}_{m=1}^{M}$, where $\hat{\boldsymbol{f}}_m$ is the $m^{\text{th}}$ column of $\hat{\boldsymbol{F}}$
```

Motivated by this, we propose to find an orthogonal and sparse basis set in the extended latent space, such that each basis vector corresponds to one of the attributes. More specifically, given a set of N latent vectors $\{\boldsymbol{w}_n^+\}_{n=1}^{N}$ and their corresponding attribute labels $\{\boldsymbol{y}_n\}_{n=1}^{N}$, we look for $\mathcal{F} = \{\boldsymbol{f}_m^+\}_{m=1}^{M}, \boldsymbol{f}_m^+ \in \mathcal{W}^+$, such that ${\boldsymbol{f}_m^+}^T \boldsymbol{f}_{m'}^+ = 0, m \neq m'$ and $\|\boldsymbol{f}_m^+\|_0 \leq \delta, \forall m$, where $\|.\|_0$ is the ℓ_0 norm of a vector and indicates its number of nonzero entries. The sparsity condition can be enforced by regularizing the ℓ_1 norm of the attribute directions, which is the convex relaxation of the ℓ_0 norm. For our experiments, we employ $20,000$ latent vectors ($N = 20,000$). Compared to many existing methods that use labelled data to create a semantically meaningful embedding, this is a large reduction in supervision requirements. For example, for quantitative comparisons with methods based on compositional learning in Section 4, their proposed models are trained with the full CelebA [15] training set, which contains about $160,000$ faces.

To enforce the orthogonality constraint, at each iteration of learning the attribute vectors, we replace the learned set of attribute directions with its nearest orthogonal set. This problem is closely related to Procrustes problems, in which the goal is to find the closest orthonormal matrix to a given matrix [7]. Algorithm 1 summarizes the operations performed at each iteration on the learned attribute directions to find their nearest orthogonal set. In short, a matrix $\boldsymbol{F}$ is created whose columns are the ℓ_2-normalized version of learned directions. Then, the nearest *orthonormal* matrix to $\boldsymbol{F}$ is computed by finding the matrix $\hat{\boldsymbol{F}}$ that minimizes $\|\boldsymbol{F} - \hat{\boldsymbol{F}}\|_F^2$, such that $\hat{\boldsymbol{F}}^T\hat{\boldsymbol{F}} = \boldsymbol{I}$, where $\|.\|_F$ denotes the Frobenius norm and $\boldsymbol{I}$ is identity matrix. It can be shown that the solution to this problem is given by $\hat{\boldsymbol{F}} = \boldsymbol{F}(\boldsymbol{F}^T\boldsymbol{F})^{-\frac{1}{2}}$. Then, the columns of the orthonormal matrix $\hat{\boldsymbol{F}}$ are rescaled to have the same norms as $\boldsymbol{F}$.

Algorithm 2 Extracting Orthogonal Basis Set for Disentangled Semantics

```
Input: Latent vectors $\{\boldsymbol{w}_n^+\}_{n=1}^N$ and their attribute labels
    $\{\boldsymbol{y}_n\}_{n=1}^N, \boldsymbol{y}_n \in \{0,1\}^M$, classification loss function
    $\mathcal{L}_c$, regularization parameter $\lambda$, and a learning rate $\beta$
Output: A set of $M$ orthogonal and sparse vectors, each
    corresponding to an attribute direction
Initialize the attribute directions $\{\boldsymbol{f}_m^+\}_{m=1}^M$ and biases
    $b_m$ randomly
repeat
    for each attribute $m = 1, \ldots, M$ do
        Calculate $\hat{y}_{m,n} = \boldsymbol{f}_m^{+T} \boldsymbol{w}_n^+ + b_m$
        Compute Loss $\mathcal{L}_m = \sum_n \mathcal{L}_c(y_{m,n}, \hat{y}_{m,n}) + \lambda \|\boldsymbol{f}_m^+\|_1$
        $\boldsymbol{f}_m^+ = \boldsymbol{f}_m^+ - \beta \nabla_f \mathcal{L}_m$
        $b_m = b_m - \beta \nabla_b \mathcal{L}_m$
    end for
    Replace $\{\boldsymbol{f}_m^+\}_{m=1}^M$ with its nearest orthogonal set using Alg 1
until convergence
Normalize $\boldsymbol{f}_m^+ = \boldsymbol{f}_m^+ / \|\boldsymbol{f}_m^+\|_2, \forall m$
return $\{\boldsymbol{f}_m^+\}_{m=1}^M$
```

Algorithm 2 provides the steps to extract the orthogonal sparse basis set, in more details. At each iteration, after updating all the attribute directions using the gradient of the loss function, Algorithm 1 is used to enforce the orthogonality condition, by projecting the current iterate onto the feasible set (set of orthonormal matrices). In optimization literature, this feasible set is referred to as Stiefel manifold and the act of projection is referred to as retraction. It is shown that gradient descent with retraction onto Stiefel manifold converges to a critical point, under very mild conditions (see Theorem 2.5 in [2]). Thus, Algorithm 2 is able to find the orthogonal basis set in a convergent manner. For our experiments, similar to prior research [25], we use hinge loss as the classification loss function $\mathcal{L}_c$.

3.2. Retrieval Using Orthogonal Decomposition

Dissimilarity decomposition and preference assignment: Given the obtained set of orthonormal directions, the query image $\boldsymbol{w}_q^+$, and any other latent vector $\boldsymbol{w}^+$, we decompose the dissimilarity vector $\boldsymbol{w}_q^+ - \boldsymbol{w}^+$ into its components. This can be done by projecting the dissimilarity vector onto each of the M attribute directions as:

$$\boldsymbol{d}_F = \boldsymbol{F}^T(\boldsymbol{w}_q^+ - \boldsymbol{w}^+) = \boldsymbol{F}^T \boldsymbol{w}_q^+ - \boldsymbol{F}^T \boldsymbol{w}^+, \qquad (1)$$

where columns of $\boldsymbol{F} \in \mathbb{R}^{d^+ \times M}$ contains the M orthonormal vectors obtained by Algorithm 2. m^{th} entry of $\boldsymbol{d}_F \in \mathbb{R}^M$ represents the inner product of $\boldsymbol{w}_q^+ - \boldsymbol{w}^+$ with $\boldsymbol{f}_m^+$. $\boldsymbol{d}_F$ is the component of the dissimilarity vector that lies inside the subspace spanned by our M attribute directions. We can also compute the residual displacement that is not represented in this subspace as:

$$\begin{aligned} \boldsymbol{d}_I &= (\boldsymbol{w}_q^+ - \boldsymbol{w}^+) - \boldsymbol{\mathcal{P}}_F(\boldsymbol{w}_q^+ - \boldsymbol{w}^+) \\ &= (\boldsymbol{I} - \boldsymbol{\mathcal{P}}_F)(\boldsymbol{w}_q^+ - \boldsymbol{w}^+), \end{aligned} \qquad (2)$$

where $\boldsymbol{\mathcal{P}}_F = \boldsymbol{F}\boldsymbol{F}^T \in \mathbb{R}^{d^+ \times d^+}$ is the orthogonal projection matrix onto the subspace spanned by these vectors. This residual subspace contains information about the identity as well as other visual and semantic attributes not included in our M predefined facial attributes. Therefore, for a given query latent vector $\boldsymbol{w}_q^+$ and the attribute preference vector $\boldsymbol{p}_q$, we propose the following weighted distance metric from any other latent vector w^+ as:

$$d(\boldsymbol{w}_q^+, \boldsymbol{w}^+, \boldsymbol{p}_q) = \boldsymbol{d}_F^T \boldsymbol{P} \boldsymbol{d}_F + \|\boldsymbol{d}_I\|_2^2, \qquad (3)$$

where $\boldsymbol{P}$ is an $M \times M$ diagonal matrix, whose diagonal entries contain the preference vector $\boldsymbol{p}_q$. The first term is the weighted Euclidean distance across different attribute directions (weighted attribute-aware distance), while the second term is the distance in the subspace not spanned by these directions (attribute-independent distance). This gives the user the ability to fine-tune the contribution of each component to achieve the desired result. In the special case, where $\boldsymbol{P}$ is set to identity matrix, this distance metric reduces to simple Euclidean distance in the latent space, $\|\boldsymbol{w}_q^+ - \boldsymbol{w}^+\|_2^2$.

Adjusting attributes: As mentioned earlier, we can adjust the m^{th} attribute in the query by moving its latent vector, $\boldsymbol{w}_q^+$, along the direction corresponding to the m^{th} attribute, $\boldsymbol{f}_m^+$, i.e., $\boldsymbol{w}_q^+ + \alpha \boldsymbol{f}_m^+$. Due to the definition of $\boldsymbol{d}_I$ and $\boldsymbol{d}_F$, this operation will not affect $\boldsymbol{d}_I$, as it represents the displacement in the subspace not spanned by the attribute direction. Furthermore, such adjustment will only affect the m^{th} entry of $\boldsymbol{d}_F$. We can write $\boldsymbol{d}_F$ for the adjusted latent vector as:

$$\boldsymbol{d}_F = \boldsymbol{F}^T(\boldsymbol{w}_q^+ + \alpha \boldsymbol{f}_m^+) - \boldsymbol{F}^T \boldsymbol{w}^+, \qquad (4)$$

which, due to orthonormality, simply translates into adding α to the m^{th} entry of $\boldsymbol{F}^T \boldsymbol{w}_q^+$. Multiple attributes can be adjusted at the same time by modifying their corresponding entries independently. Thus, we can manipulate the search results by updating $\boldsymbol{d}_F$ as:

$$\boldsymbol{d}_F = T(\boldsymbol{a}_q, \boldsymbol{w}_q^+, \boldsymbol{F}) - \boldsymbol{F}^T \boldsymbol{w}^+,$$

where $\boldsymbol{a}_q \in [0,1]^M$ is the attribute intensities provided by the user and $T(.)$ is an affine transform that maps the range $[0,1]$ to range of possible values for each entry of $\boldsymbol{F}^T \boldsymbol{w}_q^+$. The range of possible values, and therefore $T(.)$, can be obtained using the training set. Specifically, the output of $T(\boldsymbol{a}_q, \boldsymbol{w}_q^+, \boldsymbol{F})$ is an M-dimensional vector, whose m^{th} entry is set as $a_{q,m}(a_{\text{max}}^m - a_{\text{min}}^m) + a_{\text{min}}^m$, where a_{max}^m and a_{min}^m are the maximum and minimum value of $\boldsymbol{f}_m^{+T} \boldsymbol{w}_n$ over all the training feature vectors $\boldsymbol{w}_n$, respectively.

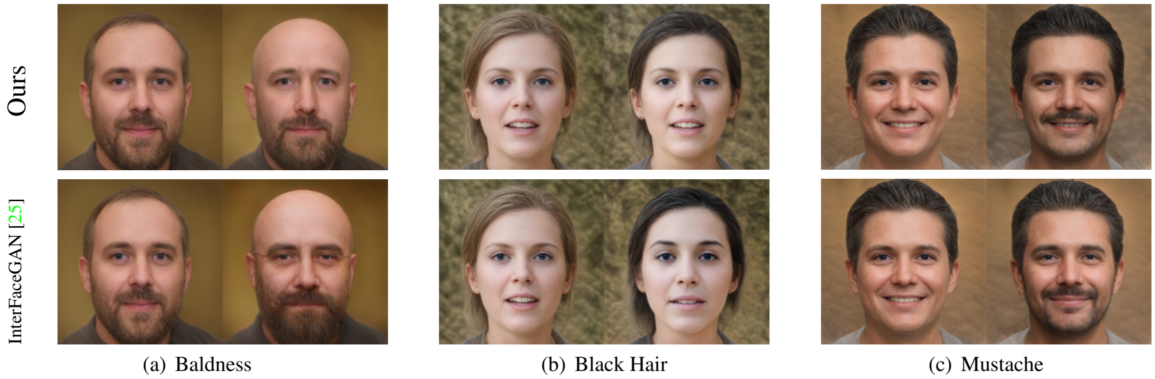

Figure 3. Qualitative evaluation of the learned attribute directions. In each pair of images, the image on the right is synthesized after moving the latent vector corresponding to the image on the left along an attribute direction. For attributes `Black Hair` and `Baldness`, the baseline is affecting the smile and the eyes as well, an artifact that is not present in the image manipulated by our method. For attribute `Mustache`, our method is able to add mustache to the face while not affecting the beard as much as the baseline.

Implementation Details: We encode the face images in the training, query, and gallery sets using the StyleGAN encoder proposed in [20], trained in an unsupervised manner on FFHQ [11] dataset. This encoder is trained using the StyleGAN generator in order to be able to map real images onto the latent space, $\mathcal{W}^+$. The latent vectors, $\{\boldsymbol{w}_n^+\}_{n=1}^N$, extracted from the training set are fed to Algorithm 2 to obtain the attribute directions $\{\boldsymbol{f}_m^+\}_{m=1}^M$. For latent vector of each query image, $\boldsymbol{w}_q^+$, the dissimilarity vector is computed by subtracting the query latent vector from each gallery latent vector. Using Equations (1) and (2), the dissimilarity vector is decomposed into $\boldsymbol{d}_F$ and $\boldsymbol{d}_I$, which are then used to compute the weighted distance (Equation (3)). This weighted distance metric is used to sort all the faces in gallery and retrieve the most similar images. The attributes can be adjusted either by moving the original latent vector, $\boldsymbol{w}_q^+$ along the corresponding attribute direct or, as shown in Equation (4), by modifying the projected latent vector.

4. Experiments

In this section, we evaluate our proposed face image retrieval framework. We employ the StyleGAN architecture and the training details as discussed in [12]. For obtaining the attribute directions, generating queries, and creating the gallery set, CelebA dataset [15] is used. $20,000$ samples, out of $160,000$ from the training set are used for training the attribute directions, while the full test set, containing $19,962$ faces, is used for creating queries and as the gallery data set. To the best of our knowledge, no other large-scale face dataset provides the ground truth for a large number of facial attributes. However, for qualitative results, we generate a much larger gallery set, containing $100,000$ faces, by sampling from the latent space.

The search performance is quantified using two evaluation metrics. **Normalized discounted cumulative gain (nDCG)**, which measures the similarity of the query attributes, after making the adjustments specified by the user, with the search results, while giving more weight to the top results. nDCG is closely related to top-k accuracy for binary attributes, while giving the top results larger weight in a logarithmic manner (which makes it more suitable for ranking problems). Furthermore, in contrast to top-k accuracy, nDCG can be used for real-valued attributes as well. **Identity Similarity** is calculated by embedding all the images onto the feature space generated by the Inception Resnet V1 architecture, as described in [26] and trained on VGGFace2 [5]. Then, the average cosine similarity between the embedded feature vector of the query face and the search results is used as a measure of identity similarity. Unless otherwise stated, the regularization parameter λ and the learning rate β are set to 5×10^{-3} and 10^{-2}, respectively in Algorithm 2. λ is selected from the set $\{0, 10^{-3}, 5 \times 10^{-3}, 10^{-2}\}$ by validating the obtained directions on the validation set of CelebA dataset. Best results for both the validation and test sets is achieved for $\lambda = 5 \times 10^{-3}$. The default value for attributes' preference is set to 1.

Qualitative Results: Figure 3 evaluates the obtained directions qualitatively for three attributes. In each pair of images, the left image is the starting point (image synthesized using a latent vector), and the image on the right shows the same image after adjusting a certain attribute (the image synthesized after moving the latent vector along the direction corresponding to the attribute). The top row illustrates the results obtained using the directions employing our proposed method and the middle row shows the results obtained by method in [25]. We argue that our proposed sparse attribute directions is able to preserve the identity better and also able to disentangle the attributes more accurately. For example, for attributes `Black Hair` and `Baldness`, the direction obtained by [25] is affecting the smile and shape of the eyes as well, an artifact that is not present in the image manipulated by our method. For attribute `Mustache`, our method is able to add mustache to the face while not affecting the beard as much as the baseline. This is due to the fact

Table 1. nDCG and identity similarity for different attribute-guided image retrieval methods, averaged over 1000 queries.

Number of retrieved images		5		10		20	
Method	Preference Assignment	nDCG	Identity Similarity	nDCG	Identity Similarity	nDCG	Identity Similarity
Attributes as Operators[17]	Not Applicable	0.730	0.824	0.720	0.823	0.711	0.824
TIRG [29]		0.794	0.847	0.781	0.844	0.776	0.840
Concat		0.804	0.841	0.806	0.838	0.805	0.822
Concat++		0.812	0.829	0.814	0.827	0.795	0.835
TIRG++ [29]		0.822	0.830	0.813	0.827	0.814	0.824
InterFaceGAN [25]	No Preference	0.568	0.838	0.570	0.835	0.571	0.832
	Identity Constrained	0.822	0.859	0.813	0.849	0.801	0.841
	Best nDCG	0.905	0.824	0.893	0.820	0.881	0.817
Ours	No Preference	0.595	0.849	0.586	0.845	0.583	0.841
	Identity Constrained	0.858	**0.864**	0.847	**0.855**	0.835	**0.846**
	Best nDCG	**0.923**	0.848	**0.917**	0.827	**0.909**	0.833

that, by enforcing sparsity, only the most relevant entries, and therefore layers, of the latent vector are modified.

Figure 4 shows a few examples of retrieval results using the synthetic gallery set. It is clear that our retrieval approach performs well on different attributes for both adjusting and emphasizing the attributes. It also shows that, our approach is able to adjust and emphasize multiple attributes at the same time, without affecting the other attributes much. For example, the last row in Figure 4(a) shows the results after adjusting three attributes, namely `Bangs`, `Black hair`, and `Eyeglasses`. Similarly, in the last row of Figure 4(b), the results are retrieved after adjusting attribute `Smile` and emphasizing attribute `Baldness`, by assigning it a larger value.

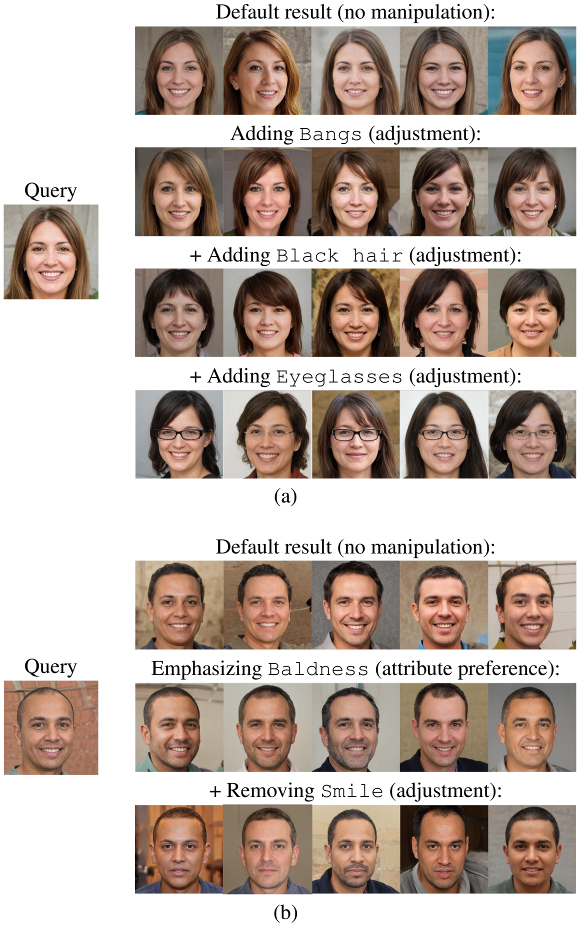
Default result (no manipulation):
Adding Bangs (adjustment):
Query
+ Adding Black hair (adjustment):
+ Adding Eyeglasses (adjustment):
(a)
Default result (no manipulation):
Query
Emphasizing Baldness (attribute preference):
+ Removing Smile (adjustment):
(b)

Figure 4. Qualitative evaluation of face image retrieval by considering both the *adjustment* and attribute *preference*. The user is able to both adjust multiple attributes in the query face and to customize the similarity metric by assigning preference to the attributes.

Quantitative Results: Table 1 shows the nDCG and identity similarity for adjusting a single attribute using different attribute-guided image retrieval methods, averaged over 1000 queries. TIRG stands for Text Image Residual Gating, which uses text input to adjust the attributes [29]. We use the implementation provided by authors of [29, 17] to train the baseline models, using the full CelebA dataset. Similar to TIRG, *Concat* uses text queries and concatenates the feature vector extracted from the text input with feature extracted from the query image to perform the retrieval. TIRG++ and Concat++ stand for their improved versions, which does not use triplet loss, as discussed in detail in [29]. Unlike our proposed method, text inputs are not able to adjust the attributes in a continuous fashion and can only remove or add the attributes. Thus, for a fair comparison, we limit the attribute intensity vector provided to our framework to a binary vector, i.e., $\boldsymbol{a}_q \in \{0,1\}^M$. However, our framework can also be used for continuous adjustment of attributes $\boldsymbol{a}_q \in [0,1]^M$. Furthermore, the compositional learning methods cannot assign different preference values to different attributes. Thus, we evaluate the GAN-based methods under *four* different settings: (i) **Best nDCG**: This setting represents the case where the attribute preference for the changed attribute (not all the attributes) is set such that the best nDCG is achieved for each query. In this scenario, the nDCG of our method is significantly larger than the methods based on compositional learning, while the achieving the same identity similarity. (ii) **Identity constrained**: The attribute preference is set such that the identity similarity is at least as good as the best compositional learning method for each query. In this scenario, our proposed framework outperforms other competitors both in nDCG and identity similarity. As expected, the

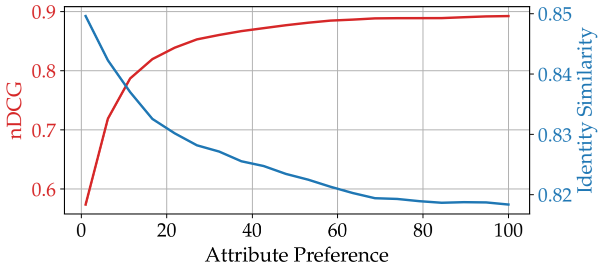

Figure 5. Impact of attribute preference on nDCG and identity similarity of the search results obtained by our method.

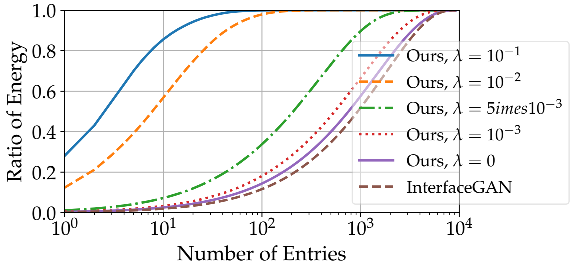

Figure 6. The energy concentrated in the top, most relevant, entries of the attribute vectors, averaged over all the attributes, for different values of the sparsity regularization parameter λ.

nDCG improvement is not as large as the previous scenario. This shows that our method is able to preserve identities, while improving the attribute similarities. (iii) **No preference:** This setting corresponds to the scenario where user has no preference and all the attributes are treated the same. In this setting GAN-based methods underperform compositional learning methods in terms of nDCG. This shows the importance of assigning preference in the GAN-based methods. It is also worthwhile to mention that the compositional learning methods implicitly assign preference to the attribute being adjusted, as these models are trained using losses to adjust attributes. (iv) **Fixed attribute preference value**: In this setting, the preference value is the same for all the queries and does not depend on nDCG or identity similarity. Figure 5 illustrates the average top-5 nDCG and identity similarity for each value of attribute preference, averaged over all the queries. As expected, as we increase the preference for the target attribute, the attribute nDCG increases while the identity similarity decreases. However, even for the largest average nDCG, i.e., maximum attribute preference, the identity similarity is still comparable to the baselines in Table 1. This shows how the user can utilize the attribute preference to achieve the desired trade-off between identity and attribute retrieval. We want to stress the fact the preference value is application-specific and cannot be optimized using the validation set.

Finally, to show the impact of sparsity on the selectivity of the attribute directions, Figure 6 illustrates the amount of energy in the most relevant entries of the attribute vectors for different values of sparsity regularization parameter λ, averaged over all the attributes. For instance, for the vectors trained using our method with $\lambda = 5 \times 10^{-3}$, about 1000 entries contain 95% the energy of the vector. This means that, in most cases, only 10% of the entries of a latent vector are modified to adjust the corresponding attribute. On the other hand, for vectors obtained using [25], the same amount energy is distributed over more than $5,000$ entries. Table 2 shows the impact of sparsity on the image retrieval performance. It is clear that increasing λ up to 10^{-2} on the attribute direction can increase the attribute retrieval accuracy, in terms of nDCG, while keeping the identity similarity about the same. This shows that the sparse directions can successfully adjust the attribute, while preserving the identity. The first row of the table, i.e., $\lambda = 0$ can also serve as an ablation study on the impact of enforcing only the orthogonality during the training. Comparing the results with the result of InterFaceGAN from Table 1, we can notice that, by only enforcing orthogonality, the same nDCG can be achieved with better identity similarity. It is worthwhile to mention that since our retrieval method depends on orthogonal decomposition of distances, we cannot report results without enforcing orthogonality. Additional implementation details and experiments, including more retrieval results on both CelebA and synthetic images, editing multiple attributes, and ablation study on number of training samples are provided in the supplementary materials.

Table 2. Top-5 nDCG and identity similarity for different levels of sparsity, i.e., different values of the regularization parameter λ, averaged over 1000 queries.

Regularization parameter	nDCG	Identity Similarity
$\lambda = 0$ (no sparsity constraint)	0.826	0.863
$\lambda = 10^{-3}$	0.847	0.863
$\lambda = 5 \times 10^{-3}$	**0.858**	0.864
$\lambda = 10^{-2}$	0.849	**0.866**

5. Conclusion

In this paper, a new formulation for face image retrieval is proposed. The new formulation considers a query face image, attribute modifiers, and attribute preference as input constraints to retrieve the most compatible face from a gallery set. While the attribute modifiers define which attributes to manipulate in the query image, the attribute preference sets the importance assigned to each attribute when compared to a gallery image. We propose a model that leverages the StyleGAN latent space characteristics to learn sparse and orthogonal attribute directions to increase control over each attribute and to allow adjusting multiple attributes at the same time, while reducing unwanted changes in the rest of the attributes. The proposed setup is evaluated on CelebA and compared to a set of state-of-the-art baselines showing improved retrieval performance.

References

[1] Kenan E. Ak, Ashraf A. Kassim, Joo Hwee Lim, and Jo Yew Tham. Learning Attribute Representations with Localization for Flexible Fashion Search. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018. 2

[2] Nicolas Boumal, P. A. Absil, and Coralia Cartis. Global rates of convergence for nonconvex optimization on manifolds. *IMA Journal of Numerical Analysis*, 2019. 5

[3] Biagio Brattoli, Karsten Roth, and Bjorn Ommer. MIC: Mining interclass characteristics for improved metric learning. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 2

[4] Fatih Cakir, Kun He, Xide Xia, Brian Kulis, and Stan Sclaroff. Deep metric learning to rank. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019. 2

[5] 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. 6

[6] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, 2014. 2

[7] John C Gower, Garmt B Dijksterhuis, and others. *Procrustes problems*, volume 30. Oxford University Press on Demand, 2004. 4

[8] Xiaoxiao Guo, Steven Rennie, Hui Wu, Gerald Tesauro, Yu Cheng, and Rogerio Schmidt Feris. Dialog-based Interactive Image Retrieval. In *Advances in Neural Information Processing Systems*, 2018. 1, 2

[9] Xintong Han, Zuxuan Wu, Phoenix X. Huang, Xiao Zhang, Menglong Zhu, Yuan Li, Yang Zhao, and Larry S. Davis. Automatic Spatially-Aware Fashion Concept Discovery. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. 2

[10] Xintong Han, Zuxuan Wu, Yu Gang Jiang, and Larry S. Davis. Learning fashion compatibility with bidirectional LSTMs. In *MM 2017 - Proceedings of the 2017 ACM Multimedia Conference*, 2017. 2

[11] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019. 2, 3, 4, 6

[12] 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. 2, 3, 4, 6

[13] Adriana Kovashka, Devi Parikh, and Kristen Grauman. WhittleSearch: Interactive Image Search with Relative Attribute Feedback. *International Journal of Computer Vision*, 2015. 1, 2

[14] Samuli Laine. Feature-based metrics for exploring the latent space of generative models. In *6th International Conference on Learning Representations, ICLR 2018 - Workshop Track Proceedings*, 2018. 2

[15] 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. 4, 6

[16] Ishan Misra, Abhinav Gupta, and Martial Hebert. From red wine to red tomato: Composition with context. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017. 2

[17] Tushar Nagarajan and Kristen Grauman. Attributes as Operators. *European Conference on Computer Vision*, 2018. 2, 7

[18] Hyeonwoo Noh, Paul Hongsuck Seo, and Bohyung Han. Image question answering using convolutional neural network with dynamic parameter prediction. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016. 2

[19] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, 2018. 2

[20] 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. 3, 6

[21] Artsiom Sanakoyeu, Vadim Tschernezki, Uta Buchler, and Bjorn Ommer. Divide and conquer the embedding space for metric learning. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019. 2

[22] Adam Santoro, David Raposo, David G.T. Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. In *Advances in Neural Information Processing Systems*, 2017. 2

[23] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015. 1

[24] Hang Shao, Abhishek Kumar, and P. Thomas Fletcher. The riemannian geometry of deep generative models. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2018. 2, 4

[25] 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, 5, 6, 7, 8

[26] 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. 6

[27] Ayush Tewari, Mohamed Elgharib, Gaurav Bharaj, Florian Bernard, Hans Peter Seidel, Patrick Perez, Michael Zollhofer, and Christian Theobalt. Stylerig: Rigging stylegan for

3d control over portrait images. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2020. 2

[28] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015. 2

[29] 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. 1, 2, 7

[30] Xun Wang, Haozhi Zhang, Weilin Huang, and Matthew R. Scott. Cross-Batch Memory for Embedding Learning. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2020. 2

[31] Xin Yang, Xuemeng Song, Xianjing Han, Haokun Wen, Jie Nie, and Liqiang Nie. Generative Attribute Manipulation Scheme for Flexible Fashion Search. In *SIGIR 2020 - Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020. 1, 2

[32] Zac Yu and Adriana Kovashka. Syntharch: Interactive image search with attribute-conditioned synthesis. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2020. 1, 2

[33] Bo Zhao, Jiashi Feng, Xiao Wu, and Shuicheng Yan. Memory-augmented attribute manipulation networks for interactive fashion search. In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017. 1, 2