Bilinear Models of Parts and Appearances in Generative Adversarial Networks

James Oldfield, Christos Tzelepis, Yannis Panagakis, Mihalis A. Nicolaou, and Ioannis Patras

Abstract—Recent advances in the understanding of Generative Adversarial Networks (GANs) have led to remarkable progress in visual editing and synthesis tasks, capitalizing on the rich semantics that are embedded in the latent spaces of pre-trained GANs. However, existing methods are often tailored to specific GAN architectures and are limited to either discovering global semantic directions that do not facilitate localized control, or require some form of supervision through manually provided regions or segmentation masks. In this light, we present an architecture-agnostic approach that jointly discovers factors representing spatial parts and their appearances in an entirely unsupervised fashion. These factors are obtained by applying a semi-nonnegative tensor factorization on the feature maps, which in turn enables context-aware local image editing with pixel-level control. In addition, we show that the discovered appearance factors correspond to saliency maps that localize concepts of interest, without using any labels. Experiments on a wide range of GAN architectures and datasets show that, in comparison to the state of the art, our method is far more efficient in terms of training time and, most importantly, provides much more accurate localized control.

Index Terms—GANs, Interpretability, Local Image Editing

1 Introduction

Generative Adversarial Networks (GANs) [1] constitute the state of the art (SOTA) for the task of image synthesis. However, despite the remarkable progress in this domain through improvements to the image generator's architecture [2], [3], [4], [5], [6], [7], their inner workings remain to a large extent unexplored. Developing a better understanding of the way in which high-level concepts are represented and composed to form synthetic images is important for a number of downstream tasks such as generative model interpretability [8], [9], [10] and image editing [11], [12], [13], [14], [15], [16]. In modern generators however, the synthetic images are produced through an increasingly complex interaction of a set of per-layer latent codes in tandem with the feature maps themselves [4], [5], [6] and/or with skip connections [7]. Furthermore, given the rapid pace at which new architectures are being developed, demystifying the process by which these vastly different networks model the constituent parts of an image is an ever-present challenge. Thus, many recent advances are architecture-specific [17], [18], [19] and a general-purpose method for analyzing and manipulating convolutional generators remains elusive.

A popular line of GAN-based image editing research concerns itself with learning so-called "interpretable directions" in the generator's latent space [10], [11], [12], [13], [14], [15], [20], [21], [22], [23]. Once discovered, such representations of high-level concepts can be manipulated to bring about predictable changes to the images. One important question in this line of research is how latent representations are combined to form the appearance at a particular *local* region of the image. Whilst some recent methods attempt to tackle this problem [17], [19], [24], [25], [26], [27], [28], the current state-of-the-art methods come with a number of important drawbacks and limitations. In particular, existing techniques require prohibitively long training times [17], [26], costly Jacobian-based optimization [26], and the requirement of semantic masks [17] or manually specified regions of interest [26]. Furthermore, whilst these methods [17], [26] successfully find directions affecting local changes, optimization must be performed on a per-region basis, and the resulting directions do not provide *pixel-level* control [26].

In this light, we present a fast unsupervised method for *jointly* learning factors for interpretable parts and their appearances (we thus refer to our method as *PandA*) in pre-trained convolutional generators. Our method allows one to both interpret and edit an image's style at discovered local semantic regions of interest, using the learnt appearance representations. We achieve this by formulating a constrained optimization problem with a semi-nonnegative tensor decomposition of the dataset of deep feature maps $\mathcal{Z} \in \mathbb{R}^{M \times H \times W \times C}$ in a convolutional generator. This allows one to accomplish a number of useful tasks, prominent examples of which are shown in Fig. 1. Firstly, our learnt representations of appearance across samples can be used for the popular task of local image editing [17], [26] (for example, to change the colour or texture of a cat's ears as shown in Fig. 1 (a)). Whilst the state-of-the-art methods [17], [26] provide fine-grained control over a target region, they

- *J. Oldfield and I. Patras are with the School of Electronic Engineering and Computer Science, Queen Mary University of London. E-mail: j.a.oldfield@qmul.ac.uk*
- *C. Tzelepis is with the Department of Computer Science, City University of London.*
- *Y. Panagakis is with the Department of Informatics and Telecommunications, National and Kapodistrian University of Athens and Archimedes AI, Athena RC.*
- *M. A. Nicolaou is with the Computation-based Science and Technology Research Center, at the Cyprus Institute.*

Manuscript received Apr 07, 2023; revised Oct 13, 2023.

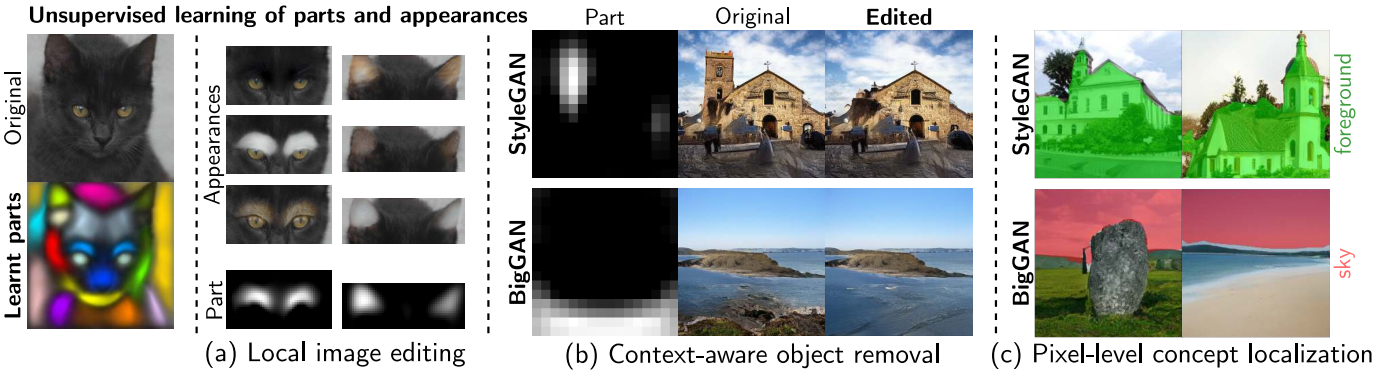

Fig. 1: We propose an unsupervised method for learning a set of factors that correspond to interpretable parts and appearances in a dataset of images. These can be used for multiple tasks: (a) local image editing, (b) context-aware object removal, and (c) producing saliency maps for learnt concepts of interest.

adopt an "annotation-first" approach, requiring an end-user to first manually specify a ROI. By contrast, our method fully exploits the unsupervised learning paradigm, wherein such concepts are discovered automatically and without any manual annotation. These discovered semantic regions can then be chosen, combined, or even modified by an end-user as desired for local image editing.

More interestingly still, through a generic decomposition of the feature maps our method identifies representations of common concepts (such as "background") in *all* generator architectures considered (all 3 StyleGANs [4], [5], [6], ProgressiveGAN [3], and BigGAN [7]). This is a surprising finding, given that these generators are radically different in architecture. By then editing the feature maps using these appearance factors, we can thus, for example, *remove* specific objects in the foreground (Fig. 1 (b)) in all generators, seamlessly replacing the pixels at the target region with the background appropriate to each image.

However, our method is useful not only for local image editing, but also provides a straightforward way to localize the learnt appearance concepts in the images. By expressing activations in terms of our learnt appearance basis, we are provided with a *visualization* of how much of each of the appearance concepts are present at each spatial location (i.e., saliency maps for concepts of interest). By then thresholding the values in these saliency maps (as shown in Fig. 1 (c)), we can localize the learnt appearance concepts (such as sky, floor, or background) in the images–without the need for supervision at any stage.

We show exhaustive experiments on 5 different architectures [3], [4], [5], [6], [7] and 5 datasets [4], [29], [30], [31], [32]. Our method is not only orders of magnitude faster than the SOTA, but also showcases superior performance at the task of local image editing, both qualitatively and quantitatively. Our contributions can be summarized as follows:

- We present an architecture-agnostic unsupervised framework for learning factors for both the parts and the appearances of images in pre-trained GANs, that enables local image editing. In contrast to the SOTA [17], [26], our approach requires neither semantic masks nor manually specified ROIs, yet offers more precise pixel-level control.

TABLE 1: A high-level comparison of our method to the SOTA for local image editing. "Training time" denotes the total training time required to produce the images for the quantitative comparisons for StyleGAN2 (at layer 5 with feature maps of size $512 \times 16 \times 16$). We use "style diversity" to refer to the ability to make a large number of visual changes at each semantic part.

	StyleSpace	LowRankGAN	ReSeFa	**PandA**
Manual ROI–free	✓	✗	✗	✓
Semantic mask–free	✗	✓	✓	✓
Pixel-level control	✗	✗	✗	✓
Architecture-agnostic	✗	✓	✓	✓
Style diversity	✓	✗	✗	✓
Training time (mins)	177.12	324.21	347.79	**0.87**

- Through a semi-nonnegative tensor decomposition of the generator's feature maps, we show how one can learn sparse representations of semantic parts of images by formulating and solving an appropriate constrained optimization problem.
- We show that the proposed method learns appearance factors that correspond to semantic concepts (e.g., background, sky, skin), which can be localized in the image through saliency maps.
- A rigorous set of experiments show that the proposed approach allows for more accurate local image editing than the SOTA, while taking only a fraction of the time to train.

2 Related Work

Generative Adversarial Networks (GANs) [1] continue to push forward the state of the art for the task of image synthesis through architectural advances such as the use of convolutions [2], progressive growing [3], and style-based architectures [4], [5], [6]. Understanding the representations induced by these networks for interpretation [8], [9], [10] and control [8], [11], [12], [13], [14], [15], [17], [23], [26], [33], [34] has subsequently received much attention.

However, whilst several methods identify ways of manipulating the latent space of GANs to bring about global semantic changes–either in a supervised [8], [13], [35], [36] or unsupervised [11], [12], [14], [15] manner–many of them struggle to apply *local* changes to regions of interest in the

image. In this framework of local image editing, one can swap certain parts between images [16], [18], [37], [38], [39], [40], or modify the style at particular regions [17], [19], [24], [25], [26], [27], [28]. This is achieved with techniques such as clustering [18], [25], [27], [28], manipulating the AdaIN [41] parameters [17], [24], or/and operating on the feature maps themselves [24], [25], [27] to aid the locality of the edit. Other approaches employ additional latent spaces or architectures [19], [40], require the computation of expensive gradient maps [17], [24] and semantic segmentation masks/networks [17], [19], [42], or require manually specified regions of interest [26]. In contrast to related work, our method automatically learns both the parts and a diverse set of global appearances, in a fast unsupervised procedure without any semantic masks. Additionally, our method allows for *pixel-level* control [26]: the ability to *precisely* target specific pixels in the image. For example, one can choose to modify a single eye only in a face, which is not possible with the SOTA [26]. Our method and its relationship to the SOTA for local image editing is summarized in Table 1.

From a methodological standpoint, most closely related to our method are the works of Collins et al. [18], [43]. Both of these perform clustering in the activation space for parts-based representations in generators [18] and CNNs [43] respectively. However, [43] considers only discriminative networks for locating common semantic regions in CNNs, whilst we additionally focus on image editing tasks in GANs. On the other hand, [18] does not jointly learn representations of *appearances*. Therefore [18] is limited to swapping parts between two images, and is additionally StyleGAN-specific, unlike our method that offers a generic treatment of convolutional generators.

3 Methodology

In this section, we detail our approach for jointly learning interpretable parts and their appearances in pre-trained GANs, in an unsupervised manner. We begin by establishing the notation used throughout the paper in Section 3.1. We then introduce our proposed separable model in Section 3.2, and our optimization objective in Section 3.3. In Section 3.4 we describe our initialization strategies.

3.1 Notation

We use uppercase (lowercase) boldface letters to refer to matrices (vectors), e.g., $\mathbf{X}$ ($\mathbf{x}$), and calligraphic letters for higher-order tensors, e.g., $\mathcal{X}$. We refer to each element of an N^{th} order tensor $\mathcal{X}$ using N indices, i.e., $\mathcal{X}(i_1, i_2, \ldots, i_N) \triangleq x_{i_1 i_2 \ldots i_N} \in \mathbb{R}$. The **mode-$n$ fibers** of a tensor are the column vectors formed when fixing all but one of the indices (e.g., $\mathbf{x}_{:jk} \in \mathbb{R}^{I_1}$). For a tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$, its mode-$n$ fibers can be stacked along the columns a matrix, giving us the **mode-n unfolding** denoted as $\mathbf{X}_{(n)} \in \mathbb{R}^{I_n \times \bar{I}_n}$ with $\bar{I}_n = \prod_{\substack{t=1 \\ t \neq n}}^{N} I_t$ [44]. We denote a pre-trained convolutional GAN generator with G, and use $G_{[:l]}$ to refer to the partial application of the last l layers of the generator only.

3.2 A separable model of parts and appearances

A convolutional generator maps each latent code $\mathbf{z}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ to a synthetic image $\mathcal{X}_i \in \mathbb{R}^{\tilde{H} \times \tilde{W} \times \tilde{C}}$ via a sequence of 2D transpose convolutions. The intermediate convolutional features $\mathcal{Z}_i \in \mathbb{R}^{H \times W \times C}$ at each layer have a very particular relationship to the output image. Concretely, each spatial activation [45] (which can be thought of as a spatial coordinate in the feature maps in Fig. 2 indexed with an (h, w) tuple) affects a specific patch in the output image [18]. At each of these spatial positions, a channel fiber $\mathbf{z}_{ihw:} \in \mathbb{R}^C$ lies depth-wise along the activation tensor, determining its content. With this understanding, we propose to factor the spatial and channel modes *separately* with a tensor decomposition, providing an intuitive separation into representations of the images' parts and appearances. Subsequently editing the feature maps with the learnt appearances at target semantic part(s) provides a simple interface for local image editing. We suggest that representations of a set of interpretable parts for local image editing should have two properties:

1) *Non-negativity*: the representations ought to be *additive* in nature, thus corresponding to semantic parts of the images [46].
2) *Sparsity*: the parts should span disjoint spatial regions, capturing different localized patterns in space, as opposed to global ones [47], [48].

Concretely, given the dataset of N samples' intermediate feature maps $\mathcal{Z} \in \mathbb{R}^{N \times H \times W \times C}$ from the pre-trained generator, each sample i's mode-3 unfolding $\mathbf{Z}_{i(3)} \in \mathbb{R}^{C \times S}$ contains in its columns the channel-wise activations at each of the $S \triangleq H \cdot W$ spatial positions in the feature maps.[1] We propose a separable factorization of the form

$$\mathbf{Z}_{i(3)} = \mathbf{A} \mathbf{\Lambda}_i \mathbf{P}^\top \tag{1}$$

$$= \underbrace{\begin{bmatrix} | & & | \\ \mathbf{a}_1 & \cdots & \mathbf{a}_{R_C} \\ | & & | \end{bmatrix}}_{\text{Appearance}} \underbrace{\begin{bmatrix} \lambda_{i11} & \lambda_{i12} & \cdots \\ \vdots & \ddots & \\ \lambda_{iR_C1} & \cdots & \lambda_{iR_CR_S} \end{bmatrix}}_{\text{Sample } i\text{'s coefficients}} \underbrace{\begin{bmatrix} — & \mathbf{p}_1{}^\top & — \\ & \vdots & \\ — & \mathbf{p}_{R_S}{}^\top & — \end{bmatrix}}_{\text{Parts}}, \tag{2}$$

where $\mathbf{A} \in \mathbb{R}^{C \times R_C}$ are the global appearance factors and $\mathbf{P} \geq \mathbf{0} \in \mathbb{R}^{S \times R_S}$ are the global parts factors, jointly learnt across many samples in a dataset. Intuitively, the coefficients λ_{ijk} encode how much of appearance $\mathbf{a}_j$ is present at part $\mathbf{p}_k$ in sample i's feature maps $\mathbf{Z}_{i(3)}$. We show our proposed separable decomposition schematically in Fig. 2. Each *non-negative* parts factor $\mathbf{p}_k \in \mathbb{R}^S \geq \mathbf{0}$ spans a spatial sub-region of the feature maps, corresponding to a semantic part. The various appearances and textures present throughout the dataset are encoded in the appearance factors $\mathbf{a}_j \in \mathbb{R}^C$ and lie along the depth-wise channel mode of the feature maps. This formulation facilitates modelling the multiplicative interactions [49] between the parts and appearance factors. Concretely, due to the outer product, the factors relating to the parts control the spatial regions at which the various appearance factors are present. The parts factors thus function similarly to semantic masks, however (in contrast to related work [17]) are learnt jointly and in an entirely unsupervised manner. This is particularly useful for datasets for which segmentation masks are not readily available.

[1] Intuitively, $\mathbf{Z}_{i(3)} \in \mathbb{R}^{C \times S}$ can be viewed simply as a 'reshaping' of the i^{th} sample's feature maps that combines the height and width modes into a single S-dimensional 'spatial' mode.

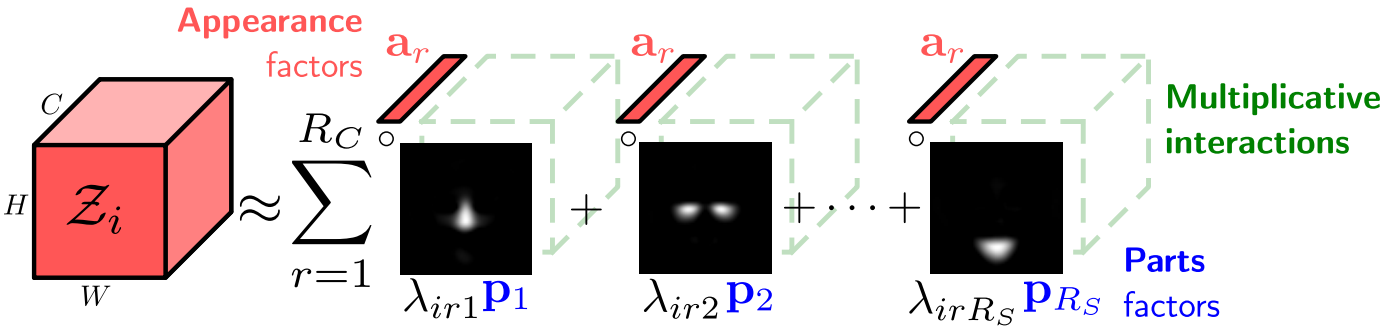

Fig. 2: An overview of our method. We decompose a dataset of generator's activations $\mathcal{Z}_i \in \mathbb{R}^{H\times W\times C}$ with a separable model. Each factor has an intuitive interpretation: the factors for the spatial modes $\mathbf{p}_j$ control the parts, determining at *which spatial locations* in the feature maps the various appearances $\mathbf{a}_k$ are present, through their multiplicative interactions.

3.3 Objective

We propose to solve a constrained optimization problem that leads to the two desirable properties outlined in Section 3.2. We impose hard non-negativity constraints on the parts factors $\mathbf{P}$ to achieve property 1, and encourage both factor matrices $\mathbf{A}, \mathbf{P}$ to be column-orthonormal for property 2 (which has been shown to lead to sparser representations [47], [48], [50], [51], and has intricate connections to clustering [52], [53]). We achieve this by formulating a single reconstruction objective as follows. Let $\mathcal{Z} \in \mathbb{R}^{N\times C\times S}$ be a batch of N samples' mode-3 unfolded intermediate activations. Then our constrained optimization problem is

$$\min_{\mathbf{A},\mathbf{P}} \mathcal{L}(\mathcal{Z},\mathbf{A},\mathbf{P}) = \min_{\mathbf{A},\mathbf{P}} \sum_{i=1}^{N} ||\mathbf{Z}_i - \mathbf{A}\left(\mathbf{A}^\top \mathbf{Z}_i \mathbf{P}\right)\mathbf{P}^\top||_F^2 \quad (3)$$
$$\text{s.t. } \mathbf{P} \geq \mathbf{0},$$

where $\mathbf{\Lambda}_i \triangleq \mathbf{A}^\top \mathbf{Z}_i \mathbf{P}$ are the sample-specific coefficients in Eq. (1), which can be viewed as an 'encoding' of the feature maps in the parts and appearance bases. Given the symmetric encoder-decoder form of the objective function, a good reconstruction naturally leads to orthogonal factor matrices (e.g., $\mathbf{P}^\top\mathbf{P} \approx \mathbf{I}_{R_S}$ for $\mathbf{P} \in \mathbb{R}^{S\times R_S}$ with $S \geq R_S$) without the need for additional hard constraints [54]. What's more, each parts factor (column of $\mathbf{P}$) is encouraged to span a distinct spatial region to simultaneously satisfy both the non-negativity and orthonormality-via-reconstruction constraints. However, this problem is non-convex. We thus propose to break the problem into two sub-problems in $\mathbf{A}$ and $\mathbf{P}$ separately, applying a form of block-coordinate descent [55], optimizing each factor matrix separately whilst keeping the other fixed. The gradients of the objective function in Eq. (3) with respect to the two factor matrices (see the supplementary material for the derivation) are given by

$$\nabla_{\mathbf{P}}\mathcal{L} = 2\left(\sum_{i=1}^{N} \bar{\mathbf{P}}\mathbf{Z}_i^\top \bar{\mathbf{A}}\bar{\mathbf{A}}\mathbf{Z}_i\mathbf{P} + \mathbf{Z}_i^\top \bar{\mathbf{A}}\bar{\mathbf{A}}\mathbf{Z}_i\bar{\mathbf{P}}\mathbf{P} - 2\mathbf{Z}_i^\top \bar{\mathbf{A}}\mathbf{Z}_i\mathbf{P}\right), \quad (4)$$

$$\nabla_{\mathbf{A}}\mathcal{L} = 2\left(\sum_{i=1}^{N} \bar{\mathbf{A}}\mathbf{Z}_i\bar{\mathbf{P}}\bar{\mathbf{P}}\mathbf{Z}_i^\top \mathbf{A} + \mathbf{Z}_i\bar{\mathbf{P}}\bar{\mathbf{P}}\mathbf{Z}_i^\top \bar{\mathbf{A}}\mathbf{A} - 2\mathbf{Z}_i\bar{\mathbf{P}}\mathbf{Z}_i^\top \mathbf{A}\right), \quad (5)$$

with $\bar{\mathbf{P}} \triangleq \mathbf{P}\mathbf{P}^\top$ and $\bar{\mathbf{A}} \triangleq \mathbf{A}\mathbf{A}^\top$. After a gradient update for the parts factors $\mathbf{P}$, we project them onto the non-negative orthant [55] with $\max\{\mathbf{0}, \cdot\}$ to enforce the non-negativity constraint. This leads to our alternating optimization algorithm, outlined in Algorithm 1.

Algorithm 1: Block-coordinate descent solution to Eq. (3)

Input : $\mathcal{Z} \in \mathbb{R}^{M\times C\times S}$ (M lots of mode-3-unfolded activations), $R_C, R_S \in \mathbb{R}$ (ranks), $\lambda \in \mathbb{R}$ (learning rate), and T (# iterations).
Output: $\mathbf{P} \in \mathbb{R}^{S\times R_S}$, $\mathbf{A} \in \mathbb{R}^{C\times R_C}$ (*parts* and *appearance* factors).
Initialise
  $\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}^\top \leftarrow \text{SVD}\left(\mathbf{Z}_{(2)}\mathbf{Z}_{(2)}^\top\right)$;
  $\mathbf{A}^{(1)} \leftarrow \mathbf{U}_{:R_C}$;
  $\mathbf{P}^{(1)} \sim \mathcal{U}(0, 0.01)$;
for $t = 1$ **to** T **do**
  $\mathbf{P}^{(t+1)} \leftarrow$
  $\max\left\{\mathbf{0}, \mathbf{P}^{(t)} - \lambda \cdot \nabla_{\mathbf{P}^{(t)}}\mathcal{L}\left(\mathcal{Z}, \mathbf{A}^{(t)}, \mathbf{P}^{(t)}\right)\right\}$;
  `// PGD step`
  $\mathbf{A}^{(t+1)} \leftarrow \mathbf{A}^{(t)} - \lambda \cdot \nabla_{\mathbf{A}^{(t)}}\mathcal{L}\left(\mathcal{Z}, \mathbf{A}^{(t)}, \mathbf{P}^{(t+1)}\right)$;
end

3.3.1 *Editing with the parts and appearance factors*

Upon convergence of Algorithm 1, to modify an image i at region k with the j^{th} appearance with desired magnitude $\alpha \in \mathbb{R}$, we compute the forward pass from layer l onwards in the generator with $\mathcal{X}_i' = G_{[l:]}\left(\mathbf{Z}_i + \alpha\mathbf{a}_j\hat{\mathbf{p}}_k^\top\right)$, with $\hat{\mathbf{p}}_k$ being the normalized parts factor of interest. Intuitively, this operation adds $\alpha\mathbf{a}_j$ at the non-zero spatial positions in $\mathbf{p}_k$ (as is depicted graphically in a single term of Fig. 2).

3.4 Initialization

Let $\mathcal{Z} \in \mathbb{R}^{N\times C\times S}$ be a batch of N mode-3 unfolded feature maps as in Section 3.3. A common initialization strategy [48], [56], [57] for non-negative matrix/tensor decompositions is via a form of HOSVD [58], [59]. Without non-negativity constraints, the channel factor matrix subproblem has a closely related closed-form solution given by the first R_C left-singular vectors of the mode-2 unfolding of the activations expressed in terms of the parts basis (proof given in Appendix B of [60]). We thus initialize the channel factors

at time-step $t = 1$ with $\mathbf{A}^{(1)} \triangleq \mathbf{U}_{:R_C}$ where $\mathbf{U}_{:R_C}$ are the first R_C-many left-singular vectors of $\mathbf{Z}_{(2)}\mathbf{Z}_{(2)}^\top$. Later on in Section 5.1.2 we demonstrate the benefits of this choice, including its usefulness for locating interpretable appearances.

4 Objective Function Analysis and Refinement

In this section, we first aim to provide a complementary interpretation of the objective in the parts subproblem of Eq. (3) from the perspective of clustering. Using this insight we then suggest a more general graph clustering formulation of a 'refinement' step to specialize the global parts factors to sample-specific parts factors.

4.1 Parts subproblem as regularised k-means

The parts factor subproblem in Eq. (3) is motivated in Section 3.2 from the point of view of a tensor factorization–with the parts factors controlling whereabouts the appearance factors are present via the outer product. However, many works in the literature also show a tight connection between NMF-style objectives and the k-means clustering objective [47], [48], [52], often differing only in the constraints imposed. To view the parts' objective as a form of clustering we first show how it can be written in the same form as the Projective Nonnegative Matrix Factorisation (PNMF) [47], [48]. Treating the column-orthonormal appearance factors $\mathbf{A} \in \mathbb{R}^{C\times R_C}$ as a constant, the parts factors $\mathbf{P} \in \mathbb{R}^{S\times R_S}$ are learnt by minimizing the objective in PandA's parts subproblem as

$$\arg\min_{\mathbf{P}\geq\mathbf{0}} \sum_{i=1}^{N} ||\mathbf{Z}_i - \mathbf{A}(\mathbf{A}^\top\mathbf{Z}_i\mathbf{P})\mathbf{P}^\top||_F^2 \tag{6}$$

$$= \arg\min_{\mathbf{P}\geq\mathbf{0}} \sum_{i=1}^{N} \text{tr}\left(\mathbf{Z}_i^\top\mathbf{Z}_i\right) - 2\text{tr}\left(\mathbf{P}\mathbf{P}^\top\mathbf{Z}_i^\top\mathbf{A}\mathbf{A}^\top\mathbf{Z}_i\right) \tag{7}$$
$$+\text{tr}\left(\mathbf{P}\mathbf{P}^\top\mathbf{Z}_i^\top\mathbf{A}\mathbf{A}^\top\mathbf{A}\mathbf{A}^\top\mathbf{Z}_i\mathbf{P}\mathbf{P}^\top\right)$$

$$= \arg\min_{\mathbf{P}\geq\mathbf{0}} \sum_{i=1}^{N} -2\text{tr}\left(\mathbf{P}\mathbf{P}^\top\mathbf{Z}_i^\top\mathbf{A}\mathbf{A}^\top\mathbf{Z}_i\right) \tag{8}$$
$$+\text{tr}\left(\mathbf{P}\mathbf{P}^\top\mathbf{Z}_i^\top\mathbf{A}(\mathbf{A}^\top\mathbf{A})\mathbf{A}^\top\mathbf{Z}_i\mathbf{P}\mathbf{P}^\top\right)$$

$$= \arg\min_{\mathbf{P}\geq\mathbf{0}} \sum_{i=1}^{N} \underbrace{-2\,\text{tr}\left(\mathbf{P}^\top\mathbf{W}_i\mathbf{P}\right)}_{\mathcal{L}_a} + \underbrace{\text{tr}\left(\mathbf{P}\mathbf{P}^\top\mathbf{W}_i\mathbf{P}\mathbf{P}^\top\right)}_{\mathcal{L}_b}, \tag{9}$$

where $\mathbf{W}_i \triangleq (\mathbf{A}^\top\mathbf{Z}_i)^\top(\mathbf{A}^\top\mathbf{Z}_i) \in \mathbb{R}^{S\times S}$ is the Gram matrix of the S-many channel fibers' coordinates *in the appearance basis*. As identified in [47], [48], $\mathcal{L}_a$ is the (negative) k-means clustering objective formulated in [52] in terms of cluster indicator matrix $\mathbf{P}$ for our specific data $\mathbf{A}^\top\mathbf{Z}_i$. The second term $\mathcal{L}_b$ can be seen as a 'regularization' term that encourages orthogonality in $\mathbf{P}$. Concretely, by inspecting Eq. (9)'s gradient, given by

$$\sum_{i=1}^{N} \underbrace{-4\mathbf{W}_i\mathbf{P}}_{-2\cdot\nabla_\mathbf{P}\mathcal{L}_a} + \underbrace{2\mathbf{P}\mathbf{P}^\top\mathbf{W}_i\mathbf{P} + 2\mathbf{W}_i\mathbf{P}\mathbf{P}^\top\mathbf{P}}_{\nabla_\mathbf{P}\mathcal{L}_b},$$

one can see that it is the presence of this second term $\mathcal{L}_b$ that means an orthogonal parts factor solution always corresponds to an optima of the original parts subproblem: whenever $\mathbf{P}\mathbf{P}^\top = \mathbf{I}$ the gradient is $\mathbf{0}$. We therefore have a second interpretation of PandA's parts factor subproblem, when considered in isolation: as a *sum* of regularized 'soft' k-means clustering terms on the channel fibers' coordinates in the appearance basis $\mathbf{A}^\top\mathbf{Z}_i$ for all $i = 1, \ldots, N$ data samples. Solving the parts subproblem can thus alternatively be seen to give in $\mathbf{P}$ a *global* soft cluster assignment indicator matrix for the channel fibers, shared between all N samples.

4.1.1 Clustering in the appearance basis

Recall that in Section 3.4 $\mathbf{A} \in \mathbb{R}^{C\times R_C}$ is initialized via the HOSVD. When the full appearance basis is used (i.e., $R_C = C$) $\mathbf{A}$ is an orthogonal matrix. In this scenario, clustering the raw activations $\mathbf{Z}_i$ and clustering the activations' coordinates in the appearance basis $\mathbf{A}^\top\mathbf{Z}_i$ lead to equivalent solutions. This is because in the expanded objective of Eq. (9), only the Gram matrix $\mathbf{W}_i$ appears, and an orthogonal $\mathbf{A}$ means $\mathbf{W}_i = \left(\mathbf{A}^\top\mathbf{Z}_i\right)^\top\left(\mathbf{A}^\top\mathbf{Z}_i\right) = \mathbf{Z}_i^\top(\mathbf{A}\mathbf{A}^\top)\mathbf{Z}_i = \mathbf{Z}_i^\top\mathbf{Z}_i$. Interestingly however, if $\mathbf{A}$ is chosen to be low-rank ($R_C < C$) PandA's parts subproblem step can be seen as implicitly performing PCA (or dimensionality reduction more generally) on the channel fibers as a 'preprocessing step' to k-means, clustering the channel fibers represented in terms of only the leading R_C principal components. Not only does this reduce the computational cost when $R_C < C$, but might also alleviate problems arising from the curse of dimensionality [61] when C is large (e.g., $C = 2048$ at BigGAN's first layer).

4.2 Graph clustering refinement

As described in [62], the formulation in Eq. (3) for learning parts and appearances makes the implicit assumption that the samples are spatially aligned. However, this does not always hold in practice, and therefore the global parts are not always immediately useful for datasets with no alignment. To alleviate this requirement, we propose a fast optional "refinement" step of the learnt global parts factors $\mathbf{P} \in \mathbb{R}^{S\times R_s}$ to specialize them to sample-specific parts factors $\tilde{\mathbf{P}}_i \in \mathbb{R}^{S\times R_s}$ for sample i. Concretely, we propose to optimize the following more general objective:

$$\min_{\tilde{\mathbf{P}}_i\geq\mathbf{0}} \mathcal{L}_G(\mathbf{W}_i, \mathbf{A}, \tilde{\mathbf{P}}_i) = \min_{\tilde{\mathbf{P}}_i\geq\mathbf{0}} -2\text{tr}\left(\tilde{\mathbf{P}}_i^\top\mathbf{W}_i\tilde{\mathbf{P}}_i\right) \tag{10}$$
$$+\text{tr}\left(\tilde{\mathbf{P}}_i\tilde{\mathbf{P}}_i^\top\mathbf{W}_i\tilde{\mathbf{P}}_i\tilde{\mathbf{P}}_i^\top\right).$$

We note that, with column-orthonormal appearance factors $\mathbf{A}$, setting $\mathbf{W}_i := (\mathbf{A}^\top\mathbf{Z}_i)^\top(\mathbf{A}^\top\mathbf{Z}_i)$ gives the same solution to the original refinement objective presented in [62], whilst the use of a *sparse* $\mathbf{W}_i$ can be used to better capture the local neighborhood relationships [63] in this more general formulation.

As observed in the literature [47], [64], the form of Eq. (10) involves only the Gram matrix–a measure of pairwise similarity used in many graph clustering methods involving trace maximization problems of the same form [52], [65], [66]. Thus, one may view Eq. (10) as a specific kind of graph clustering, where $\mathbf{W}_i$ is an *affinity matrix*. Each

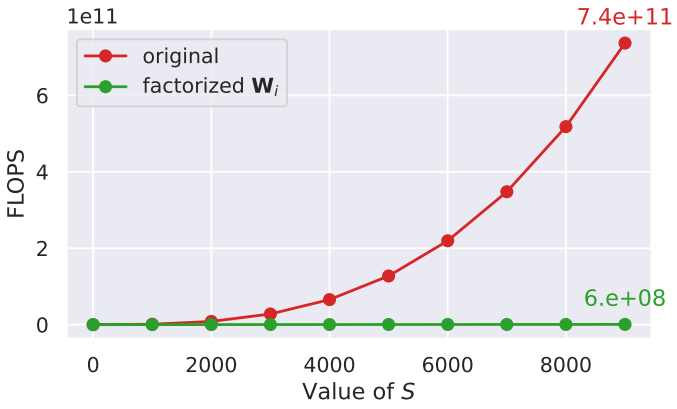

Fig. 3: Estimated number of FLOPS in each term of the original gradient computation Eq. (11) vs the factorized form Eq. (12) (using $R_C = 512, R_S = 16$).

element w_{ijk} describes the similarity (in the inner product sense) between all pairs (j, k) of sample i's channel fibers. The gradient for the unconstrained objective is given by:

$$\nabla_{\tilde{\mathbf{P}}_i}\mathcal{L}_G = 2\tilde{\mathbf{P}}_i\tilde{\mathbf{P}}_i^\top\mathbf{W}_i\tilde{\mathbf{P}}_i + 2\mathbf{W}_i\tilde{\mathbf{P}}_i\tilde{\mathbf{P}}_i^\top\tilde{\mathbf{P}}_i - 4\mathbf{W}_i\tilde{\mathbf{P}}_i. \quad (11)$$

We analyze in Section 5.3 the benefits of this refinement step and the impact of various choices of affinity matrix, comparing the global parts factors to the refined factors.

4.3 Gradient descent with low-rank affinity matrices

The gradient in Eq. (11) computed naively materializes the full affinity matrix $\mathbf{W}_i \in \mathbb{R}^{S\times S}$. At later layers in the generator, S can be very large[2], and consequently the gradient can become computationally expensive to compute. However, $\mathbf{W}_i$ is often low-rank by construction and/or positive semidefinite, thus admitting the decomposition $\mathbf{W}_i = \mathbf{X}_i^\top\mathbf{X}_i$ for some $\mathbf{X}_i \in \mathbb{R}^{R_C\times S}$ with $R_C = \text{rank}(\mathbf{W}_i)$. In the case of fully-connected affinity matrices, it can be written simply as the Gram matrix of the activations with $\mathbf{X}_i = \mathbf{A}^\top\mathbf{Z}_i$. When the affinity matrix is PSD more generally, the compact SVD of $\mathbf{W}_i$ can be used with $\mathbf{X}_i = \mathbf{\Sigma}^{1/2}\mathbf{U}^\top$. Then, through the associativity of matrix multiplication, matrices of size $S \times S$ need never be computed when descending the gradients by writing Eq. (11) instead as:

$$\nabla_{\tilde{\mathbf{P}}_i}\mathcal{L}_G = 2\tilde{\mathbf{P}}_i\mathbf{Y}_i^\top\mathbf{Y}_i + 2\mathbf{X}_i^\top\mathbf{Y}_i\left(\tilde{\mathbf{P}}_i^\top\tilde{\mathbf{P}}_i\right) - 4\mathbf{X}_i^\top\mathbf{Y}_i, \quad (12)$$

with $\mathbf{Y} \triangleq \mathbf{X}_i\tilde{\mathbf{P}}_i \in \mathbb{R}^{R_C\times R_S}$. We show in Fig. 3 the estimated number of FLOPS necessary for the two computations. As can be seen, this can greatly decrease the computational cost when S is very large. We highlight that a smarter ordering of matrix multiplications can also be used to speed up the computation of the gradients in Eqs. (4) and (5) of the main paper's original objectives even further, in a similar fashion. This is detailed in the supplementary material with corresponding estimations of FLOPS.

[2]e.g., $S = 16384$ at $l = 10$.

5 Experiments

In this section we present a series of experiments to validate the method and explore its properties. We begin in Section 5.1 by focusing on using the method for interpretation: showing how one can generate saliency maps for concepts of interest and remove the foreground at target locations. Following this, we showcase local image editing results on 5 GANs in Section 5.2. Finally, we present ablation studies in Section 5.4 to further justify and motivate our method.

5.1 Interpreting the appearance vectors

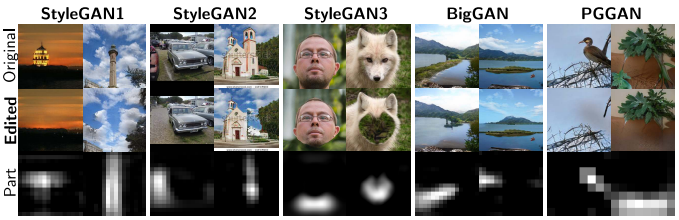

Fig. 4: Our architecture-agnostic method discovers a representation of the "background" concept directly in the feature maps, which allows us to remove objects in a context-aware manner in the same way for all 5 generators.

Using the learnt appearance basis $\mathbf{A}$, one can straightforwardly visualize "how much" of each column is present at each spatial location via a change of basis. In particular, the element at row c and column s of the activations expressed in terms of the appearance basis $\mathbf{A}^\top\mathbf{Z}_i \in \mathbb{R}^{R_C\times S}$ encodes how much of appearance c is present at spatial location s, for a particular sample i of interest. This transformation provides a visual understanding of the concepts controlled by the columns by observing the semantic regions in the image at which these values are the highest.

5.1.1 *Generic concepts shared across GAN architectures*

The analysis above leads us to make an interesting discovery. We find that our model frequently learns an appearance vector for a high-level "background" concept in *all* 5 generator architectures. This is a surprising finding–one would not necessarily expect these radically different architectures to encode concepts in the same manner (given that many existing methods are architecture-specific), let alone that they could be extracted with a single unsupervised approach. We can thus use this learnt "background" appearance vector to remove objects in a context-aware manner, as shown on all 5 generators and numerous datasets in Fig. 4.

5.1.2 *Visualizing and localizing appearance vectors*

Through the change of basis $\mathbf{A}^\top\mathbf{Z}_i$ we can identify the pixels in the image that are composed of the concept k of interest (e.g., the "background" concept), offering an interpretation of the images' semantic content. We first compute the saliency map $\mathbf{m}_{ik} = \mathbf{a}_k^\top\mathbf{Z}_i \in \mathbb{R}^S$, whose elements encode the magnitude of concept k at each spatial position in the i^{th} sample. This can be reshaped into a square matrix and visualized as an image to localize the k^{th} concept in the image, as shown in row 2 of Fig. 5. We then additionally perform a simple binary classification following [14]. We

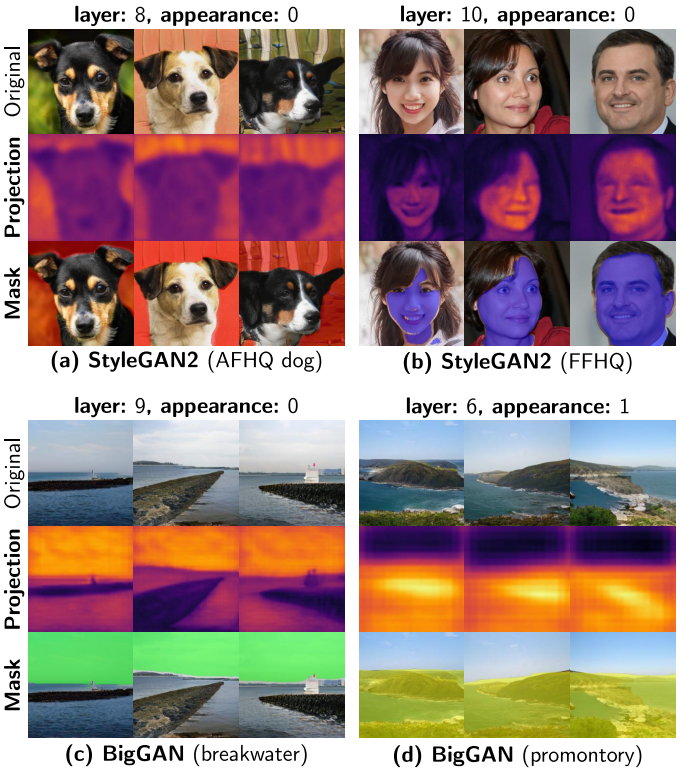

Fig. 5: Visualizing the coordinates in the appearance basis (2^{nd} row), one can interpret how much of each appearance vector is present at each spatial patch. For example, we see appearance vectors at various layers very clearly corresponding to (a) background, (b) skin, (c) sky, and (d) foreground.

classify each pixel j as an instance of concept k or not with $\tilde{m}_{ikj} = [m_{ikj} \geq \mu_k]$, where $\mu_k = \frac{1}{N \cdot S} \sum_{n,s} m_{nks} \in \mathbb{R}$ is the mean magnitude of the k^{th} concept in N samples. We show this in row 3 of Fig. 5 for various datasets and GANs. For example, this analysis allows us to identify (and localize) appearance vectors in various generators that control concepts including "foreground", "skin", and "sky", shown in Fig. 5 (b-d). We find this visualization to be most useful for understanding the first few columns of $\mathbf{A}$, which control the more prominent high-level visual concepts in the dataset due to our SVD-based initialization outlined in Section 3.4.

5.2 Local image editing

Next, we showcase our method's ability to perform local image editing in pre-trained GANs, on 5 generators and 5 datasets (ImageNet [29], AFHQ [30], FFHQ [4], LSUN [31], and MetFaces [32]). In Fig. 6 we show a number of interesting local edits achievable with our method, using both the global and refined parts factors. Whilst we can manipulate the style at common regions such as the eyes with the global parts factors, the refined parts factors allow one to target regions such as an individual's clothes, or their background. One is not limited to this set of learnt parts however. For example, one can draw a ROI by hand at test-time or modify an existing part–an example of this is shown in the supplementary material. This way, pixel-level control

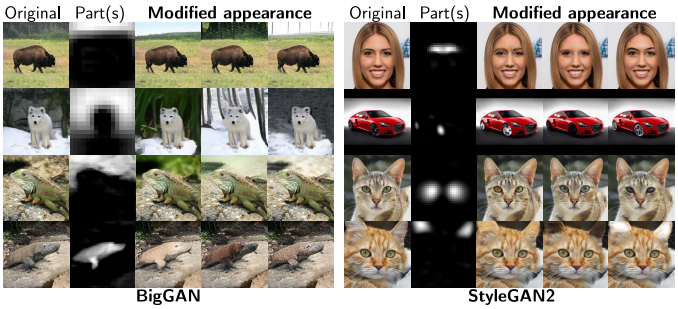

Fig. 6: Local image editing on a number of architectures and datasets, using both the global and refined parts factors. At each column, the original image is edited at the target part with a different appearance vector.

(e.g., opening only a single eye of a face) is achievable in a way that is not possible with the SOTA methods [17], [26].

We next compare our method to state-of-the-art GAN-based image editing techniques in Fig. 7. In particular, we train our model at layer 5 using $R_S := 8$ global parts factors, with no refinement, and $R_C := C$. As can be seen, SOTA methods such as LowRank-GAN [26] excel at enlarging the eyes in a photo-realistic manner. However, we frequently find the surrounding regions to change as well. This is seen clearly by visualizing the mean squared error [18] between the original images and their edited counterparts, shown in every second row of Fig. 7. Additional comparisons are found in the supplementary material. We further quantify this ability to affect local edits in the section that follows.

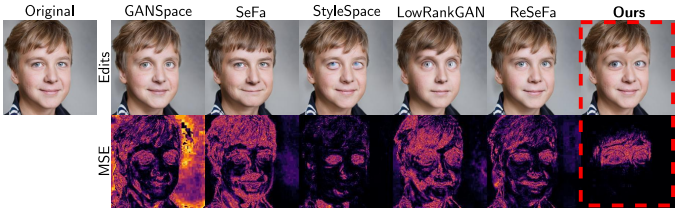

Fig. 7: Qualitative comparison to SOTA methods editing the 'eyes' ROI (using editing strength $\alpha := 30$ for the proposed method). We also show the mean squared error [18] between the original images and their edited counterparts, highlighting the regions that change.

5.2.1 *Quantitative results*

We compute the ratio of the distance between the pixels of the original and edited images in the region of 'disinterest', over the same quantity with the region of interest (call it ROIR). Concretely, we compute

$$\text{ROIR}(\mathcal{M}, \mathcal{X}, \mathcal{X}') = \frac{1}{N} \sum_{i=1}^{N} \frac{|| \, (\mathbf{1} - \mathcal{M}) * (\mathcal{X}_i - \mathcal{X}'_i) \, ||}{||\mathcal{M} * (\mathcal{X}_i - \mathcal{X}'_i) \, ||}, \quad (13)$$

where $\mathcal{M} \in [0,1]^{H \times W \times C}$ is an $H \times W$ spatial mask (replicated along the channel mode) specifying the region of interest, $\mathbf{1}$ is a 1-tensor, and $\mathcal{X}, \mathcal{X}' \in \mathbb{R}^{N \times \tilde{H} \times \tilde{W} \times \tilde{C}}$ are the batch of original and edited versions of the images respectively. A small ROIR indicates more 'local' edits, through desirable change to the ROI (large denominator) and little change

TABLE 2: ROIR (↓) for 10k FFHQ samples per local edit.

	Eyes	Nose	Open mouth	Smile
GANSpace [11]	2.80±1.22	4.89±2.11	3.25±1.33	2.44±0.89
SeFa [12]	5.01±1.90	6.89±3.04	3.45±1.12	5.04±2.22
StyleSpace [17]	1.26±0.70	1.70±0.82	1.24±0.44	2.06±1.62
LowRankGAN [26]	1.78±0.59	5.07±2.06	1.82±0.60	2.31±0.76
ReSeFa [67]	2.21±0.85	2.92±1.29	1.69±0.65	1.87±0.75
Ours	**1.04±0.33**	**1.17±0.44**	**1.04±0.39**	**1.05±0.38**

elsewhere (small numerator). We compute this metric for our method and SOTA baselines in Table 2, for a number of regions of interest. As can be seen, our method consistently produces more local edits than the SOTA for a variety of regions of interest. We posit that the reason for this is due to our operating directly on the feature maps, where the spatial activations have a direct relationship to a patch in the output image.

5.3 Parts factor refinement

Finally, we showcase the benefit of our optional parts factors refinement process for data with no alignment. In row 2 of Fig. 8, we show the global parts factors overlaid over the target samples. Clearly, for extreme poses (or in the case of data with no alignment, such as with animals and cars), these global parts will not correspond perfectly to the specific sample's parts. After a few projected gradient descent steps of PandA's original refinement objective, we see (row 3 of Fig. 8) that the refined parts factors span the specific parts of the individual samples more successfully.

5.3.1 Graph clustering

Despite the success of [62]'s original refinement step, we find that at generator layers with large spatial resolutions (e.g., $l = 10$ where $S = 128^2$) it may co-cluster semantically unrelated channel fibers when using a small number of parts. One example of this problem is shown in Fig. 9 (a), where the cat's face is co-clustered along with the ears. Appealing to the more general graph clustering perspective, one possible reason for this is that the Gram matrix used in the original refinement objective (i.e. $\mathbf{W}_i := (\mathbf{A}^\top \mathbf{Z}_i)^\top (\mathbf{A}^\top \mathbf{Z}_i)$) is *non-sparse*, and thus is overly permissive in considering semantically unrelated channel fibers as similar. Therefore, to better capture the *local* (in the inner product similarity sense) neighbourhood structure, we propose to follow the popular spectral clustering approach of using a mutual KNN graph [63] to limit the number of edges in the graph as desired. In particular, we sever the edges between non-neighboring vertices with $w_{ijk} := 0$ if channel fibers at spatial positions j, k are not considered to be mutual KNN-given neighbors in Euclidean space. This is outlined in detail in Algorithm 2, where setting the desired number of neighbors to $k = S - 1$ recovers the full Gram matrix in the original PandA formulation.

We show examples of the parts learnt with the original refinement objective and with the new graph clustering objective with sparse $\mathbf{W}_i$ in Fig. 9 (row 1). We also show local image edits using the two methods' parts in Fig. 9 (row 2). As can be seen, the new formulation clearly makes it possible to learn parts that span only single semantic regions to a greater extent than before. More results comparing the parts learnt with the two objectives can be found in the supplementary material.

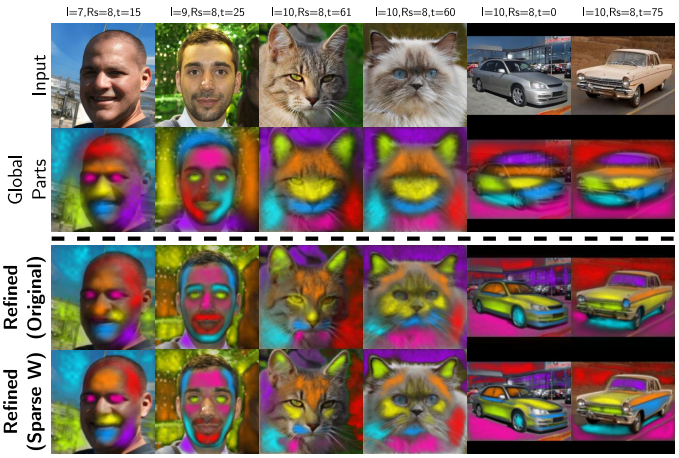

Fig. 8: Visualization of the global parts factors (2nd row), PandA's refined factors with the original fully-connected $\mathbf{W}_i$ (3rd row), and the refined factors with the new graph clustering objective with sparse $\mathbf{W}_i$ (4th row).

Algorithm 2: Graph clustering refinement

Input : $\mathbf{Z}_i \in \mathbb{R}^{C \times S}$, $\mathbf{A} \in \mathbb{R}^{C \times R_C}$, $\mathbf{P} \in \mathbb{R}^{S \times R_S}$, $T, \lambda \in \mathbb{R}$, and $k < S$ (desired # neighbors).
Output: $\tilde{\mathbf{P}}_i \in \mathbb{R}^{S \times R_S}$ (Refined parts factors)
Initialise
  $\tilde{\mathbf{P}}_i^{(1)} \leftarrow \mathbf{P}$;
  $\mathbf{G} \in \{0,1\}^{S \times S} \leftarrow$ MKNN_adj_matrix$(\mathbf{A}^\top \mathbf{Z}_i, k)$;
  $\mathbf{W}_i \leftarrow \mathbf{G} * \left((\mathbf{A}^\top \mathbf{Z}_i)^\top (\mathbf{A}^\top \mathbf{Z}_i)\right)$; `// sever edges`
for $t = 1$ **to** T **do**
  $\tilde{\mathbf{P}}_i^{(t+1)} \leftarrow$
  $\max\left\{\mathbf{0}, \tilde{\mathbf{P}}_i^{(t)} - \lambda \cdot \nabla_{\tilde{\mathbf{P}}_i^{(t)}} \mathcal{L}_G\left(\mathbf{W}_i, \mathbf{A}, \tilde{\mathbf{P}}_i^{(t)}\right)\right\}$
end

5.4 Ablation studies

In this subsection, we present a thorough study of the various components of our method, and the resulting learnt parts factors. Moreover, we find the method flexible to the choice of decomposition rank R_S. Higher values produce

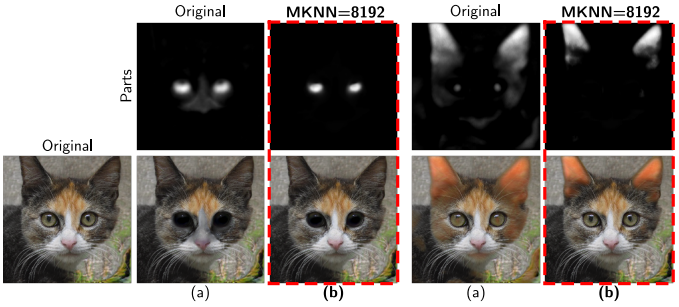

Fig. 9: Editing an image using the original refined parts (a) and with the parts learnt with the new graph clustering objective using a sparse affinity matrix (b).

more fine-grained parts (e.g. eyebrows), whilst lower values lead to more coarse-grained parts (e.g. whole faces). We refer readers to Sect. 2.7 of the supplementary material for many such additional ablation studies and visualizations.

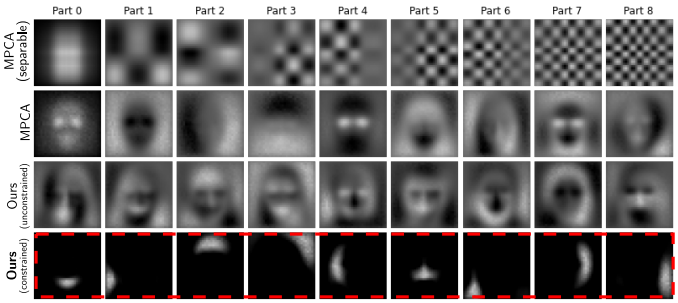

Fig. 10: Ablation study comparing the parts factors learnt with various constraints and formulations.

5.4.1 *Constraints and form*

We first study the impact of the non-negativity constraints on the parts factors, and the importance of operating on the mode-3 unfolded $\mathbf{Z}_{i(3)} \in \mathbb{R}^{C \times H \cdot W}$ tensors (rather than their original 3rd-order form $\mathcal{Z}_i \in \mathbb{R}^{H \times W \times C}$). We show along the rows of Fig. 10 the resulting parts factors using various forms of decomposition and constraints. In particular, naively applying MPCA [59] (row 1) to decompose $\mathcal{Z}_i$ imposes a separable structure *between* the spatial modes, restricting its ability to capture semantic spatial regions. Moreover, even when combining the spatial modes and decomposing $\mathbf{Z}_{i(3)}$, the solution given by MPCA [59] (row 2) and by optimizing our method *without* any non-negativity constraints (row 3) leads to parts factors spanning the entire spatial window. This is due to the non-additive nature of the parts. However, as shown in row 4 of Fig. 10, only our constrained method successfully finds local, non-overlapping semantic regions of interest.

Training time An important benefit of our method is the lack of need to compute expensive gradient maps or Jacobians with respect to target regions [17], [26]. To quantify this, we benchmark the total time needed to train the methods to produce the results in Table 2. This is summarized in Table 1 (and shown in detail in the supplementary material). We find that our method takes less than $1/400^{\text{th}}$ of the training time of LowRankGAN [26], and $1/170^{\text{th}}$ the time of StyleSpace [17]–greatly speeding up the task of local image editing. Alternatively, one can use an optimizer such as Adam [68] in an autograd framework (e.g., PyTorch [69]) to compute Algorithm 1. We find this removes some sensitivity to the learning rate that comes with vanilla gradient descent. However, we find that when solving our objective manually with the gradients in Eq. (5), our method takes less than $1/3^{\text{rd}}$ of the time to train. This is a particularly useful performance boost when decomposing later layers in the network.

5.5 Concept rewriting

In this section, we propose a intuitive way of 'rewriting' [16] the semantic concepts found by our method via basic arithmetic on the coefficients of the appearance basis.

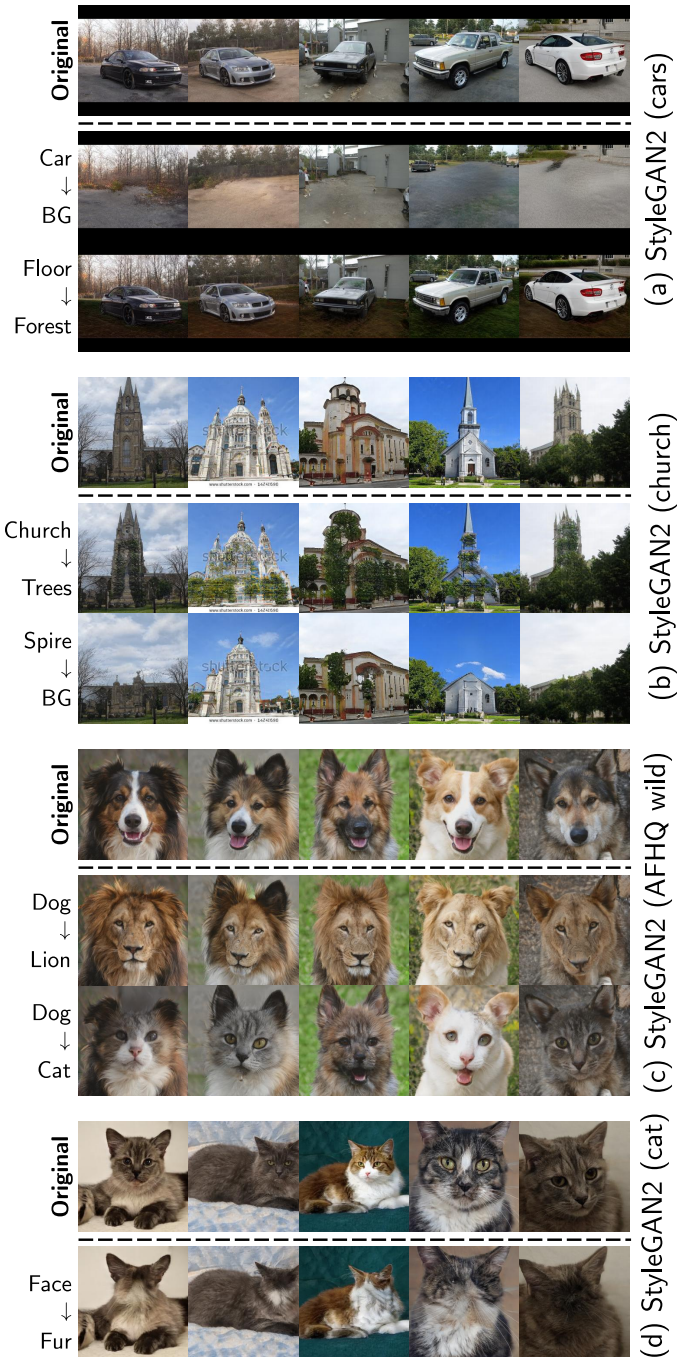

Fig. 11: Concept rewriting via manipulation of the rows of the images' feature maps in the appearance basis.

In particular, we aim to visually replace all instances of target concepts in the image with a new source concept (e.g., to change all 'concrete' to 'trees' in an image). Let $\mathbf{Z}_i \in \mathbb{R}^{C \times S}$ be a sample's activations, and $\mathbf{A} \in \mathbb{R}^{C \times C}$ be the learnt orthogonal appearance factor matrix. We always have $\mathbf{Z}_i = \mathbf{A}\mathbf{A}^\top \mathbf{Z}_i = \mathbf{A}\hat{\mathbf{Z}}_i$. Viewing the feature maps $\mathbf{Z}_i$ as a linear combination of the columns of $\mathbf{A}$, the j^{th} row of the coefficients, $\hat{\mathbf{Z}}_i(j,:) \in \mathbb{R}^S$ can be understood as specifying 'how much' of appearance vector j is at the various spatial positions in the feature maps. However, the coefficients are not strictly positive, and a negative amount of $\mathbf{a}_j$ will control a different high-level concept in the image to a positive

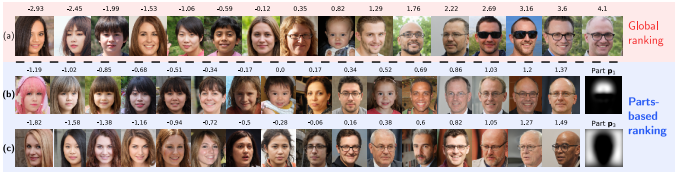

Fig. 12: Ranking the dataset by the same appearance at different parts: e.g. hair on the fringe / hair around the head. Here we use 100k images with a truncation value of 0.7 (please zoom for detail).

amount of $\mathbf{a}_j$. To disentangle this effect resulting from the mixed signs of $\hat{\mathbf{Z}}_i$ we can split the row vector controlling the j^{th} appearance concept into its positive and negative parts

$$\hat{\mathbf{Z}}_i(j,:) = \left[\hat{\mathbf{Z}}_i(j,:)\right]_+ + \left[\hat{\mathbf{Z}}_i(j,:)\right]_-, \tag{14}$$

where the terms on the left and the right denote the positive and negative amounts of concept j at each spatial position respectively. To replace the instances of concept j in the image with concept k we propose to simply move the positive coefficients in the j^{th} row to the k^{th} row. Concretely, we set:

$$\hat{\mathbf{Z}}_i(k,:) := \hat{\mathbf{Z}}_i(k,:) + \alpha \left[\hat{\mathbf{Z}}_i(j,:)\right]_+, \tag{15}$$

$$\hat{\mathbf{Z}}_i(j,:) := \left[\hat{\mathbf{Z}}_i(j,:)\right]_-. \tag{16}$$

Intuitively, at each spatial position that previously contained a positive amount of concept j, we now instead add concept k. The image with its concepts rewritten can then be generated with $G_{[:l]}(\mathbf{A}\hat{\mathbf{Z}}_i)$. In Fig. 11 we show many instances of possible manipulations with this technique with particular chosen columns of $\mathbf{A}$. For example, one can replace the church concrete with trees, or the face of cats with fur. Interestingly, at earlier layers of StyleGAN2 trained on AFHQ (wild), we find abstract concepts controlling whole animal species. It's thus possible to perform what resembles image-to-image translation by changing dogs to lions, or to cats, as shown in Fig. 11 (c). Like with Section 5.1.2, we find this to work best at earlier layers, and for only the first few columns of $\mathbf{A}$, where the high-level concepts are most easily located.

5.6 Parts-based ranking

Finally, we consider the downstream task of *ranking* images according to a particular target concept. We can compute the amount of a target concept j in image i's feature maps with

$$r_{ij} = \mathbf{a}_j^\top \mathbf{Z}_i \mathbb{1} \in \mathbb{R}, \tag{17}$$

where $\mathbb{1} \in \mathbb{R}^S$ is a vector of ones, summing the contribution over all spatial positions. This scalar r_{ij} can then be used to rank the images according to attribute j of interest. Whilst such a method is useful for many ranking tasks, this is theoretically straight forward to achieve in many existing frameworks [11], [12] that find interpretable directions in the $\mathcal{W}$ latent space. Concretely, one could perform the orthogonal projection $r_{ij} = \frac{\mathbf{u}_j^\top \mathbf{w}_i}{\mathbf{u}_j^\top \mathbf{u}_j}$ of latent code $\mathbf{w}_i \in \mathbb{R}^d$ onto interpretable direction $\mathbf{u}_j \in \mathbb{R}^d$ of interest. However, PandA's joint decomposition of parts and appearance factors allows for the more involved task of ranking the amount of an attribute j at a *particular* spatial part k. Achieving this is non-trivial in existing methods, where one has no direct spatial control. Using our jointly learnt parts factors, we can achieve part-based ranking for sample i by computing

$$r_{ijk} = \mathbf{a}_j^\top \mathbf{Z}_i \mathbf{p}_k \tag{18}$$

where the spatial part $\mathbf{p}_k \geq \mathbf{0}$ can be thought to act like a mask that zeros-out values everywhere but at the semantic part of interest. We show in Fig. 12 (a) the result of a global ranking of the appearance vector for the hair concept following Eq. (17). However, as shown in Fig. 12 (b-c), a part-based ranking following Eq. (18) can be performed to order images by the amount of hair over the fringe, or around the head, independently. This provides a much more fine-grained analysis for when one is interested in the amount of a concept at a particular spatial region only.

6 Conclusion

In this paper, we have presented a fast unsupervised algorithm for learning interpretable parts and their appearances in pre-trained GANs. We have shown experimentally how our method outperforms the state of the art at the task of local image editing, in addition to being orders of magnitude faster to train. We showed how one can identify and manipulate generic concepts in 5 generator architectures for tasks such as object removal. We also believe that our method's ability to visualize the learnt appearance concepts through saliency maps could be a useful tool for network interpretability.

6.1 Limitations

Whilst we have demonstrated that our method can lead to more precise control, the approach is not without its limitations. Such strictly local editing means that after modifying a precise image region, any expected influence on the rest of the image is not automatically accounted for. As a concrete example, one can remove trees from an image, but any shadow they may have cast elsewhere is not also removed automatically. Additionally, we find that methods editing the feature maps have a greater tendency to introduce artifacts relative to methods working on the latent codes. This is one potential risk with the freedom of pixel-level control–adding appearance vectors at arbitrary spatial locations does not always lead to photorealistic edits.

Acknowledgments This work was supported by the EU H2020 AI4Media No. 951911 project.

References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Adv. Neural Inform. Process. Syst.*, 2014.

[2] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Int. Conf. Learn. Represent.*, 2016.

[3] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," in *Int. Conf. Learn. Represent.*, 2018.

[4] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2019.

[5] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGAN," in *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2020.

[6] T. Karras, M. Aittala, S. Laine, E. Härkönen, J. Hellsten, J. Lehtinen, and T. Aila, "Alias-free generative adversarial networks," *Adv. Neural Inform. Process. Syst.*, vol. 34, 2021.

[7] A. Brock, J. Donahue, and K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis," in *Int. Conf. Mach. Learn.*, 2019.

[8] Y. Shen, J. Gu, X. Tang, and B. Zhou, "Interpreting the latent space of GANs for semantic face editing," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.

[9] D. Bau, J.-Y. Zhu, H. Strobelt, B. Zhou, J. B. Tenenbaum, W. T. Freeman, and A. Torralba, "GAN dissection: Visualizing and understanding generative adversarial networks," in *Int. Conf. Learn. Represent.*, 2019.

[10] C. Yang, Y. Shen, and B. Zhou, "Semantic hierarchy emerges in deep generative representations for scene synthesis," *Int. J. Comput. Vis.*, vol. 129, no. 5, pp. 1451–1466, 2021.

[11] E. Härkönen, A. Hertzmann, J. Lehtinen, and S. Paris, "GANSpace: Discovering interpretable GAN controls," in *Adv. Neural Inform. Process. Syst.*, vol. 33, 2020, pp. 9841–9850.

[12] Y. Shen and B. Zhou, "Closed-form factorization of latent semantics in GANs," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.

[13] Y. Shen, C. Yang, X. Tang, and B. Zhou, "InterFaceGAN: Interpreting the disentangled face representation learned by GANs," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020.

[14] A. Voynov and A. Babenko, "Unsupervised discovery of interpretable directions in the GAN latent space," in *Int. Conf. Mach. Learn.*, 2020, pp. 9786–9796.

[15] C. Tzelepis, G. Tzimiropoulos, and I. Patras, "WarpedGANSpace: Finding non-linear RBF paths in GAN latent space," in *Int. Conf. Comput. Vis.*, October 2021, pp. 6393–6402.

[16] D. Bau, S. Liu, T. Wang, J.-Y. Zhu, and A. Torralba, "Rewriting a deep generative model," in *Eur. Conf. Comput. Vis.* Springer, 2020, pp. 351–369.

[17] Z. Wu, D. Lischinski, and E. Shechtman, "StyleSpace analysis: Disentangled controls for stylegan image generation," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.

[18] E. Collins, R. Bala, B. Price, and S. Süsstrunk, "Editing in style: Uncovering the local semantics of GANs," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.

[19] H. Ling, K. Kreis, D. Li, S. W. Kim, A. Torralba, and S. Fidler, "Editgan: High-precision semantic image editing," in *Adv. Neural Inform. Process. Syst.*, 2021.

[20] Z. He, M. Kan, and S. Shan, "Eigengan: Layer-wise eigen-learning for gans," in *Int. Conf. Comput. Vis.*, 2021, pp. 14 408–14 417.

[21] R. Haas, S. Graßhof, and S. S. Brandt, "Tensor-based subspace factorization for StyleGAN," in *Proc. Int. Conf. Automatic Face and Gesture Recognit. (FG)*, 2021.

[22] ——, "Tensor-based emotion editing in the StyleGAN latent space," in *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, 2022.

[23] S. Bounareli, V. Argyriou, and G. Tzimiropoulos, "Finding directions in gan's latent space for neural face reenactment," 2022.

[24] R. Wang, J. Chen, G. Yu, L. Sun, C. Yu, C. Gao, and N. Sang, "Attribute-specific Control Units in StyleGAN for Fine-grained Image Manipulation," in *ACM Int. Conf. Multimedia*, Oct. 2021.

[25] T. Broad, F. F. Leymarie, and M. Grierson, "Network bending: Expressive manipulation of generative models in multiple domains," *Entropy*, 2022.

[26] J. Zhu, R. Feng, Y. Shen, D. Zhao, Z. Zha, J. Zhou, and Q. Chen, "Low-Rank Subspaces in GANs," in *Adv. Neural Inform. Process. Syst.*, 2021.

[27] C. Zhang, Y. Xu, and Y. Shen, "Decorating your own bedroom: Locally controlling image generation with generative adversarial networks," 2021.

[28] O. Kafri, O. Patashnik, Y. Alaluf, and D. Cohen-Or, "StyleFusion: A generative model for disentangling spatial segments," 2021.

[29] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2009, pp. 248–255.

[30] Y. Choi, Y. Uh, J. Yoo, and J.-W. Ha, "Stargan v2: Diverse image synthesis for multiple domains," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 8188–8197.

[31] F. Yu, A. Seff, Y. Zhang, S. Song, T. Funkhouser, and J. Xiao, "Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop," 2015.

[32] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, "Training generative adversarial networks with limited data," *Adv. Neural Inform. Process. Syst.*, vol. 33, pp. 12 104–12 114, 2020.

[33] M. Georgopoulos, J. Oldfield, G. G. Chrysos, and Y. Panagakis, "Cluster-guided image synthesis with unconditional models," 2021.

[34] R. Abdal, P. Zhu, N. J. Mitra, and P. Wonka, "StyleFlow: Attribute-conditioned exploration of StyleGAN-generated images using conditional continuous normalizing flows," *ACM Trans. Graph.*, vol. 40, no. 3, May 2021.

[35] L. Goetschalckx, A. Andonian, A. Oliva, and P. Isola, "Ganalyze: Toward visual definitions of cognitive image properties," in *Int. Conf. Comput. Vis.*, 2019, pp. 5744–5753.

[36] A. Plumerault, H. Le Borgne, and C. Hudelot, "Controlling generative models with continuous factors of variations," in *Int. Conf. Learn. Represent.*, 2020.

[37] K. Jakoel, L. Efraim, and T. R. Shaham, "GANs spatial control via inference-time adaptive normalization," in *IEEE Winter Conf. Applic. Comput. Vis.*, January 2022.

[38] M. J. Chong, W.-S. Chu, A. Kumar, and D. Forsyth, "Retrieve in style: Unsupervised facial feature transfer and retrieval," in *Int. Conf. Comput. Vis.*, October 2021.

[39] R. Suzuki, M. Koyama, T. Miyato, T. Yonetsuji, and H. Zhu, "Spatially controllable image synthesis with internal representation collaging," 2018.

[40] H. Kim, Y. Choi, J. Kim, S. Yoo, and Y. Uh, "Exploiting spatial dimensions of latent in gan for real-time image editing," in *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.

[41] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Int. Conf. Comput. Vis.*, Oct 2017.

[42] P. Zhu, R. Abdal, J. Femiani, and P. Wonka, "Barbershop: GAN-based Image Compositing using Segmentation Masks," *ACM Trans. Graph.*, Oct. 2021.

[43] E. Collins, R. Achanta, and S. Susstrunk, "Deep feature factorization for concept discovery," in *Eur. Conf. Comput. Vis.*, 2018, pp. 336–352.

[44] T. G. Kolda and B. W. Bader, "Tensor Decompositions and Applications," *SIAM Review*, vol. 51, no. 3, pp. 455–500, Aug. 2009.

[45] C. Olah, A. Satyanarayan, I. Johnson, S. Carter, L. Schubert, K. Ye, and A. Mordvintsev, "The building blocks of interpretability," *Distill*, 2018, https://distill.pub/2018/building-blocks.

[46] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, Oct. 1999.

[47] Z. Yang and E. Oja, "Linear and nonlinear projective nonnegative matrix factorization," *IEEE Trans. Neural Networks*, vol. 21, no. 5, pp. 734–749, 2010.

[48] Z. Yuan, Z. Yang, and E. Oja, "Projective nonnegative matrix factorization: Sparseness, orthogonality, and clustering," *Neural Process. Lett*, pp. 11–13, 2009.

[49] S. M. Jayakumar, W. M. Czarnecki, J. Menick, J. Schwarz, J. Rae, S. Osindero, Y. W. Teh, T. Harley, and R. Pascanu, "Multiplicative interactions and where to find them," in *Int. Conf. Learn. Represent.*, 2020.

[50] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix t-factorizations for clustering," in *ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, ser. KDD '06. New York, NY, USA: Association for Computing Machinery, 2006.

[51] Z. Yang and J. Laaksonen, "Multiplicative updates for non-negative projections," *Neurocomputing*, vol. 71, no. 1-3, pp. 363–373, 2007.

[52] C. Ding, X. He, and H. D. Simon, "On the equivalence of nonnegative matrix factorization and spectral clustering," in *Proceedings of the 2005 SIAM international conference on data mining*. SIAM, 2005, pp. 606–610.

[53] T. Li and C. Ding, "The relationships among various nonnegative matrix factorization methods for clustering," in *Sixth International Conference on Data Mining (ICDM'06)*. IEEE, 2006, pp. 362–371.

[54] Q. V. Le, A. Karpenko, J. Ngiam, and A. Y. Ng, "ICA with Reconstruction Cost for Efficient Overcomplete Feature Learning," in *Adv. Neural Inform. Process. Syst.*, 2013, p. 9.

[55] C.-J. Lin, "Projected Gradient Methods for Nonnegative Matrix Factorization," *Neural Computation*, vol. 19, no. 10, pp. 2756–2779, Oct. 2007.
[56] A. Cichocki, R. Zdunek, A. H. Phan, and S. ichi Amari, *Nonnegative Matrix and Tensor Factorizations - Applications to Exploratory Multi-way Data Analysis and Blind Source Separation.* Wiley, 2009.
[57] C. Boutsidis and E. Gallopoulos, "SVD based initialization: A head start for nonnegative matrix factorization," *Pattern Recognition*, vol. 41, no. 4, pp. 1350–1362, 2008.
[58] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, 1966.
[59] H. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos, "Mpca: Multilinear principal component analysis of tensor objects," *IEEE Trans. Neural Networks*, vol. 19, no. 1, pp. 18–39, 2008.
[60] D. Xu, S. Yan, L. Zhang, H.-J. Zhang, Z. Liu, and H.-Y. Shum, "Concurrent subspaces analysis," in *IEEE Conf. Comput. Vis. Pattern Recog.*, vol. 2, Jun. 2005.
[61] C. Ding, *Dimension Reduction Techniques for Clustering*. Boston, MA: Springer US, 2009, pp. 846–846.
[62] J. Oldfield, C. Tzelepis, Y. Panagakis, M. Nicolaou, and I. Patras, "Panda: Unsupervised learning of parts and appearances in the feature maps of GANs," in *Int. Conf. Learn. Represent.*, 2023.
[63] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.
[64] A. C. Türkmen, "A review of nonnegative matrix factorization methods for clustering," *arXiv preprint arXiv:1507.03194*, 2015.
[65] I. S. Dhillon, Y. Guan, and B. Kulis, "Kernel k-means: Spectral clustering and normalized cuts," in *ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, 2004, p. 551–556.
[66] ——, "A unified view of kernel k-means, spectral clustering and graph cuts," UTCS, Tech. Rep., 2004.
[67] J. Zhu, Y. Shen, Y. Xu, D. Zhao, and Q. Chen, "Region-based semantic factorization in GANs," in *Int. Conf. Mach. Learn.*, 2022.
[68] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[69] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Adv. Neural Inform. Process. Syst.* Curran Associates, Inc., 2019, pp. 8024–8035.

Yannis Panagakis received the B.Sc. degree in informatics and telecommunications from the University of Athens, and the M.Sc. and Ph.D. degrees in computer science from Aristotle University of Thessaloniki in Greece. He previously held research and academic positions with the Samsung AI Centre, Cambridge, Middlesex University London, and Imperial College London. He is currently an Associate Professor with the Department of Informatics and Telecommunications, National and Kapodistrian University of Athens and a Lead Researcher with Archimedes AI, Athena RC. His research focuses on advancing AI for robust information processing across modalities such as vision, audio, language, and neural signals. He is particularly interested in mathematical aspects of deep learning related to generalization, robustness, compression, and their interface with tensor methods. Yannis has published over 100 articles in leading journals and conferences and received several grants and awards.

Mihalis A. Nicolaou is Associate Professor at the Computation-based Science and Technology Research Center at the Cyprus Institute. He received the B.Sc. degree from the University of Athens, Greece and the M.Sc. and Ph.D. degrees from the Department of Computing, Imperial College London, UK. Previously, he held faculty and researcher positions at Goldsmiths, University of London and Imperial College London respectively. His research focuses on developing machine learning algorithms with favourable properties, such as efficiency, interpretability, and generalization, often in the context of large-scale deep learning. He is interested in applications ranging from computer vision to natural language processing, climate science, and health. He has published over 80 research articles in leading conferences and journals.

James Oldfield is currently a PhD student at Queen Mary University of London. Previously, he was a research intern at The Cyprus Institute. He has published in venues and journals including NeurIPS, ICLR, CVPR, IJCV, and the Proceedings of the IEEE. His recent research focuses on controllable and interpretable generative models of visual data.

Ioannis Patras is a Professor in Computer Vision and Machine Learning and leads the Centre for Multimodal AI in the school of Electronic Engineering and Computer Science in Queen Mary University of London. He obtained his BSc and MSc in the Computer Science Department in the University of Crete, his PhD in Delft University of Technology and held research and academic positions in the University of Amsterdam and The University of York. His current research is on Generative Multimodal AI. He is/has been in the organizing committee of Int'l Conf. Multimedia Retrieval 2023, ACM Multimedia 202, MMM2020, Face and Gesture Recognition 2008, ICMR2011, ICMR2018, ACM Multimedia 2013, BMVC 2009 and was the general chair of WIAMIS 2009. He is/was an associate editor in the Journal of Pattern Recognition, Computer Vision and Image Understanding, and Area Chair in all major Computer Vision conferences including, CVPR, ICCV, FG, ICMI, ACII, and BMVC. He has more than 250 publications in the most selective Journals and conferences in the field of Computer Vision. He has been the primary supervisor for more than 25 PhD students. He is a senior member of IEEE.

Christos Tzelepis received the Diploma degree in Electrical and Computer Engineering from Aristotle University of Thessaloniki, Greece, in 2011, and the PhD degree from Queen Mary, University of London, UK, in 2018. He is currently a Lecturer with the department of Computer Science at City, University of London. He has published at some of the most authoritative venues and journals in the fields of machine learning and computer vision (including NeurIPS, ICLR, ICCV, CVPR, IJCV, and T-PAMI). His research focuses on machine learning, computer vision, and Generative AI.