Unsupervised Panoptic Interpretation of Latent Spaces in GANs Using Space-Filling Vector Quantization

Anonymous authors
Paper under double-blind review

Abstract

Generative adversarial networks (GANs) learn a latent space whose samples can be mapped to real-world images. Such latent spaces are difficult to interpret. Some earlier supervised methods aim to create an interpretable latent space or discover interpretable directions that require exploiting data labels or annotated synthesized samples for training. However, we propose using a modification of vector quantization called space-filling vector quantization (SFVQ), which quantizes the data on a piece-wise linear curve. SFVQ can capture the underlying morphological structure of the latent space and thus make it interpretable. We apply this technique to model the latent space of pretrained StyleGAN2 and BigGAN networks on various datasets. Our experiments show that the SFVQ curve yields a general interpretable model of the latent space that determines which part of the latent space corresponds to what specific generative factors. Furthermore, we demonstrate that each line of SFVQ's curve can potentially refer to an interpretable direction for applying intelligible image transformations. We also showed that the points located on an SFVQ line can be used for controllable data augmentation.

1 Introduction

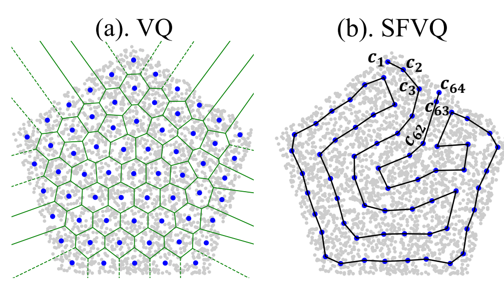

Figure 1: Codebook vectors (*blue points*) of a 6 bit (a) vector quantization, and (b) space-filling vector quantization (*curve in black*) on a pentagon distribution (*gray points*). Voronoi regions for VQ are shown *in green*.

Generative adversarial networks (GANs) (Goodfellow et al., 2014) are powerful generative models applied to various applications, e.g. data augmentation (Antoniou et al., 2017; Shorten & Khoshgoftaar, 2019), image editing (Härkönen et al., 2020; Yüksel et al., 2021; Shen & Zhou, 2021; Voynov & Babenko, 2020; Tzelepis et al., 2021; Aoshima & Matsubara, 2023; Abdal et al., 2021; Wang et al., 2018b; Alaluf et al., 2022; Roich et al., 2022; Pehlivan et al., 2023; Liu et al., 2023; Jahanian et al., 2019; Plumerault et al., 2020; Yang et al., 2021; Goetschalckx et al., 2019; Shen et al., 2020; Wu et al., 2021), video generation (Wang et al., 2018a). For image data, GANs map a latent space to an output image space by learning a non-linear mapping (Voynov & Babenko, 2020). After learning such mapping, GANs can create realistic high-resolution images by sampling from the latent space (Karras et al., 2019). However, this latent space is a black box such that it is difficult to interpret the mapping between the latent space and generative factors such as gender, age (Shen et al., 2020). In addition, the interpretable directions to change these factors are not known (Voynov & Babenko, 2020). Hence, having a comprehensive interpretation of the latent space is an important research problem that, if solved, leads to more controllable generations.

In the literature, supervised and unsupervised methods exist to find interpretable directions in the latent space. Supervised methods (Jahanian et al., 2019; Plumerault et al., 2020; Yang et al., 2021; Goetschalckx et al., 2019; Shen et al., 2020; Wu et al., 2021; Shen et al., 2022) require data collections together with the use of pretrained classifiers or human labelers to label the collected data with respect to the user-predefined directions (Shen & Zhou, 2021). In addition, these methods only find the directions that the user defines (Voynov & Babenko, 2020). On the other hand, in unsupervised methods (Härkönen et al., 2020; Shen & Zhou, 2021; Voynov & Babenko, 2020; Yüksel et al., 2021; Tzelepis et al., 2021; Aoshima & Matsubara, 2023; Song et al., 2023b;a) the user has to choose the hyper-parameter K (the number of interpretable directions to discover) before training, where a large value for K results in discovering repetitive directions (Yüksel et al., 2021). In all these unsupervised methods, there is no prior knowledge about the specific transformation each of these K discovered directions yields. Hence, the user has to do an exhaustive search over all available K directions to determine which directions are practical and what they refer to. For instance, as GANSpace (Härkönen et al., 2020) applies principal component analysis (PCA) on the latent space, the number of directions (K) to be examined is large (equal to the latent space dimension). Also, as stated in Härkönen et al. (2020), not all PCA directions are necessarily useful to change a generative factor.

In this paper, we used a modification of vector quantization, called space-filling vector quantization (SFVQ) (Vali & Bäckström, 2023), to interpret the latent spaces of the pretrained StyleGAN2 (Karras et al., 2020) and BigGAN (Brock et al., 2018) models using FFHQ (Karras et al., 2019), AFHQ (Choi et al., 2020), LSUN Cars (Yu et al., 2015), CIFAR10 (Krizhevsky et al., 2009), and ImageNet (Deng et al., 2009) datasets. Regarding the intrinsic arrangement of SFVQ's codebook vectors (see Fig. 1(b)), SFVQ can capture the underlying structure of the latent space such that subsequent codebook vectors refer to similar contents. In contrast to supervised approaches, our unsupervised method neither needs human labeling nor puts any constraint on the learned latent space as it uses the original learned latent spaces from pretrained models. Moreover, our method does not need any hyper-parameter tuning, e.g. choosing the number of directions (K) as in Shen & Zhou (2021); Voynov & Babenko (2020); Yüksel et al. (2021); Tzelepis et al. (2021); Aoshima & Matsubara (2023); Song et al. (2023b;a) or tuning the coefficients of the training loss terms as in Voynov & Babenko (2020); Yüksel et al. (2021); Tzelepis et al. (2021); Aoshima & Matsubara (2023). In our proposed method, to explore the latent space structure and find its interpretable directions, the only required effort is that the user has to visually observe the generated images from learned SFVQ's codebook vectors (Fig. 2(a), Fig. 4(a)) only once. By observing the generated images, the user would have prior knowledge of the potential edit type for a discovered direction in advance, contrary to other unsupervised methods. Therefore, we reduce the search effort to achieve the desired edit by only searching for the suitable layers in StyleGAN2 or BigGAN to modify. Our method's implementation is publicly available at https://github.com/AnonymousSubmission7/GANs-SFVQ.git.

Our experiments show that our proposed method makes StyleGAN2's latent space interpretable such that the user knows what type of generations to expect from each part of the latent space regarding age, gender, pose, accessories (for FFHQ), color, breed, pose (for AFHQ) and class of data (for CIFAR10). Furthermore, we discovered that the directions of the space-filling lines (lines connecting SFVQ's codebook vectors) can be used as interpretable directions leading to meaningful image transformations. We improved SFVQ's training (first contribution below) and observed that the learned space-filling lines are mainly located inside the latent space. Hence, by interpolating between subsequent codebook vectors, we have numerous meaningful latent vectors (located on the SFVQ's lines), which we use for controllable data augmentation. Our main contributions are:

- Contrary to Vali & Bäckström (2023) where the outlier codebook vectors were an issue, we improved the *initialization* and *codebook expansion* scheme of SFVQ such that we did not encounter any outlier codebook vectors in our experiments (Sec. 2.2.1 and Sec. 2.2.2).
- We explored SFVQ from a new viewpoint and discovered that SFVQ lines can refer to interpretable directions (Sec. 4.2 and Sec. 4.3). This SFVQ's property was not explored in Vali & Bäckström (2023).
- We found joint interpretable directions that can change several attributes at the same time (Sec. 4.7).
- We uncovered another new property of SFVQ, which is controllable data augmentation (Sec. 4.8).
- This is the first time that SFVQ is introduced to this field of research to interpret the latent spaces of GANs.

2 Methods

2.1 Space-filling vector quantization (SFVQ)

A space-filling curve is a piece-wise continuous curve created by recursion, and if the recursion repeats infinitely, the curve fills a multi-dimensional distribution (Sagan, 2012). Motivated by space-filling curves, space-filling vector quantization (SFVQ) (Vali & Bäckström, 2023) designs vector quantization (VQ) as mapping of data on a space-filling curve, whose corner points are the codebook vectors of VQ. Similar to space-filling curves, SFVQ is trained recursively. SFVQ first starts with $N = 4$ codebook vectors (2 bit) and then, it expands the codebook by doubling the number of codebook vectors at each recursion step. The recursion continues until SFVQ reaches $N = \log_2(B_{target})$ codebook vectors, where B_{target} is the SFVQ's target bitrate. Fig. 1 illustrates a 6 bit VQ and SFVQ applied on a 2D pentagon distribution.

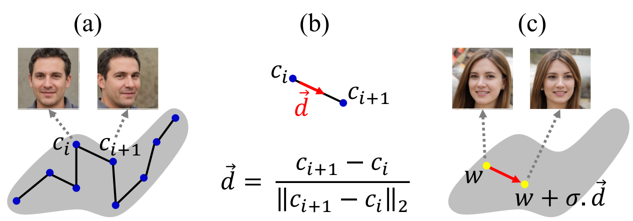

Figure 2: (a) Observation of generated images from learned SFVQ codebook (similar to Fig. 4(a)) to find a sensible direction between c_i and c_{i+1}. (b) Computation of the direction d. (c) Applying the direction on a random latent vector w by shift magnitude of σ.

2.2 Proposed method

2.2.1 SFVQ initialization

As mentioned, SFVQ training starts with $N = 4$ codebook vectors. The *initialization* of these four codebook vectors significantly impacts the final learned SFVQ curve obtained at the end of training. In Vali & Bäckström (2023), these codebook vectors were initialized randomly (from a normal distribution $\mathcal{N}(0, 1)$), and as the SFVQ codebook was expanded to reach the target bitrate, there were some outlier codebook vectors which ended up out of the latent space.

In this paper, we changed the codebook *initialization*. Since pretrained models are available, we sample 10^3 random vectors z from the normal distribution and generate their corresponding latent vectors in the layer where we intend to train the SFVQ (e.g. intermediate $\mathcal{W}$ space in StyleGAN2). Then, we compute the Euclidean norm of all latent vectors and sort them in the ascending order. We split these sorted vectors into four groups and initialize the codebook vectors with the mean of these four groups. From a geometrical viewpoint, this initial SFVQ curve spans from one end of latent Euclidean space to its other end and brings a desirable order to SFVQ codebook vectors which aligns with the intrinsic SFVQ codebook arrangement, as the curve starts from low norm to high norm latent vectors.

2.2.2 SFVQ codebook expansion

When training SFVQ, *codebook expansion* occurs at the beginning of a recursion step and the codebook size is doubled. In Vali & Bäckström (2023), the new codebook vectors are defined in the center of the line connecting two adjacent codebook vectors (which already exist on the curve), i.e. $c_{new} = (c_i + c_{i+1})/2$, where c_i is the i-th codebook vector. c_{new} can be useless as it might be located outside the latent space and it takes a long time (or many training batches) to be pushed inside. However, we define the new codebook by shifting the existing codebook vectors slightly such that $c_{new} = 0.99\, c_i + 0.01\, c_{i+1}$. Now, the new codebooks are more likely to reside inside the latent space, and thus, after being selected actively during training, they will be optimized to better locations. In contrast to Vali & Bäckström (2023), our proposed *codebook expansion* and *initialization* for SFVQ lead to no outlier codebooks throughout our experiments.

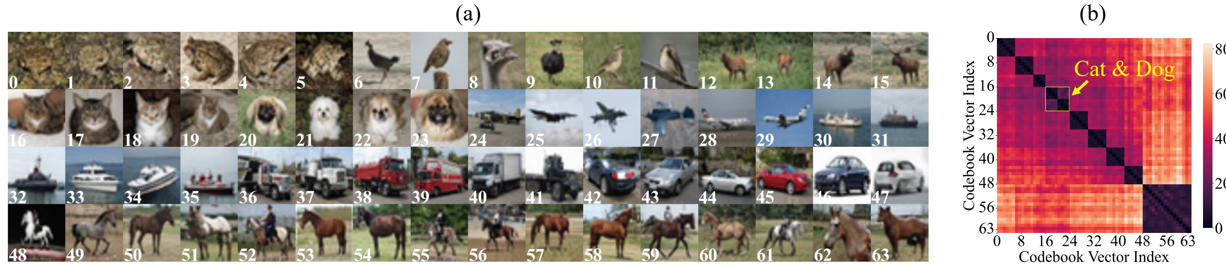

Figure 3: (a) Generated images from codebook of a 6 bit SFVQ trained on $\mathcal{W}$ space of StyleGAN2 pretrained on CIFAR10. (b) Heatmap of Euclidean distances between all codebook vectors.

2.2.3 SFVQ interpretable directions

After training SFVQ on the latent space and obtaining the learned codebook, we generate the images corresponding to SFVQ codebook vectors (Fig. 2(a) and Fig. 4(a)). This visualization reveals the underlying structure of the latent space regarding generative factors, which we discuss in Sec. 4.1. In this paper, we take one step forward to extract more information from SFVQ's curve, which results in finding interpretable directions. Fig. 2 shows how we find the interpretable directions using SFVQ. Because of the intrinsic arrangement in SFVQ codebook vectors, subsequent images refer to similar contents, i.e., they share many similar features while they are different in minimal attributes. For example, in Fig. 2(a), the images for two subsequent codebook vectors of c_i and c_{i+1} share most of the attributes except the rotation. Hence, we can infer that the direction (d) connecting these two vectors (Fig. 2(b)) refers to the rotation direction. Then, by shifting any latent vector along this direction (Fig. 2(c)), we observe the change in rotation attribute.

By a quick observation of the subsequent generated images from the SFVQ codebook, the user can simply spot the interpretable direction. Hence, the user has a prior knowledge of the direction, and the only required action is to find the proper layers of the GAN to edit along this direction (Härkönen et al., 2020). In this way, the user achieves the desired edit with less search effort compared to other unsupervised methods (Härkönen et al., 2020; Shen & Zhou, 2021; Voynov & Babenko, 2020; Yüksel et al., 2021; Tzelepis et al., 2021; Aoshima & Matsubara, 2023), in which apart from the layer-wise search, they should do an exhaustive search over all K discovered directions to inspect whether they are practical and what direction they refer to.

3 Experiments

To evaluate how SFVQ can be used to interpret the latent spaces in GANs, similarly to GANSpace (Härkönen et al., 2020), we chose the intermediate latent space ($\mathcal{W}$) of StyleGAN2 (Karras et al., 2020) and the first linear layer of BigGAN512-deep (Brock et al., 2018), and then trained the SFVQ on these layers. These layers are more favorable for interpretation because they render more disentangled representations, they are not constrained to any specific distribution, and they suitably model the structure of the real data (Karras et al., 2019; Härkönen et al., 2020; Shen et al., 2020). For StyleGAN2, we employ the pretrained models on FFHQ, AFHQ, LSUN Cars, CIFAR10 datasets, and also the pretrained BigGAN on ImageNet.

We trained the SFVQ with various bitrates ranging from 2 to 12 bit (4 to 4096 codebook vectors). Since the training of SFVQ is not sensitive to hyper-parameter tuning, we adopt a general setup that works for all pretrained models and datasets. In this setup, we trained SFVQ with the batch size of 64 over 100 k number of training batches (for each recursion step) using Adam optimizer with the initial learning rate of 10^{-3}. We used a learning rate scheduler such that during each recursion step, we halved the learning rate after 60 k and 80 k training batches.

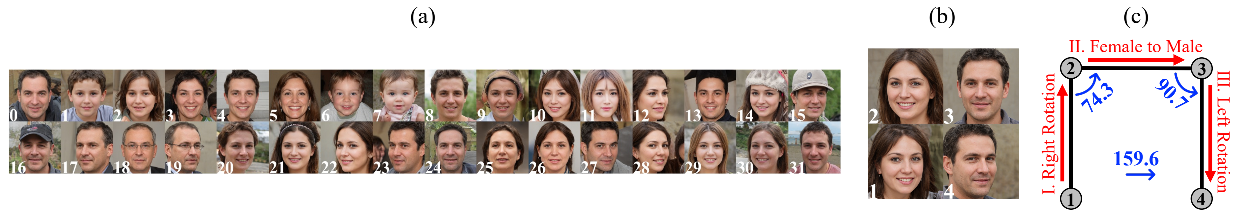

Figure 4: (a) Generated images from codebook of a 5 bit SFVQ trained on $\mathcal{W}$ space of StyleGAN2 pretrained on FFHQ. (b) Similar generated images for a 2 bit SFVQ, and (c) their semantic directions. Numbers (*in blue*) show the angle between directions.

4 Results and discussions

4.1 StyleGAN2: universal interpretation

To explore a universal interpretation of latent space, we apply the SFVQ on that distribution and plot the generated images from the obtained SFVQ's codebook vectors. According to the inherent arrangement of SFVQ's codebook vectors, we expect SFVQ to capture a universal morphology of the latent space. As the first experiment, we apply the SFVQ on the intermediate latent space ($\mathcal{W}$) of StyleGAN2 (Karras et al., 2020) pretrained on the CIFAR10 dataset. During training, the number of extracted latent vectors is unbiased for all CIFAR10 classes. In Fig. 3(a), we plot the generated images corresponding to 6 bit SFVQ codebook, i.e. each image corresponds to a codebook vector. At first glance, we observe a clear arrangement with respect to the image class, such that images from an identical category are organized into groups. Also, apart from the *horse* class, all animal types and industrial vehicles are located next to each other. Furthermore, there are some visible similarities for subsequent codebook vectors within a class, such as similar objects' rotation, scale, color, and background.

To inspect the learned SFVQ from another viewpoint, we plotted the heatmap of Euclidean distances between all SFVQ's codebook vectors in Fig. 3(b). Again, we observe a clear separation between different classes, as each dark box shows a data class. It is important to note that the SFVQ captures this class separation property because of its inherent orderliness and in a completely unsupervised way. Also, we spot a bigger dark box shared between *cat* and *dog* classes because they are the most similar classes and reside close to each other in the latent space.

In the second experiment, we applied a 5 bit SFVQ on the $\mathcal{W}$ space of the pretrained StyleGAN2 on the FFHQ dataset. Images corresponding to the SFVQ's codebook are represented in Fig. 4(a). We observe similarities among neighboring codebook vectors such as baby-aged faces for indices 6-7, hat accessory for indices 13-16, eyeglasses for indices 18-19, rotation from right to left from index 17 to 20, and rotation from left to right from index 27 to 31. Based on our investigations, the StyleGAN2's $\mathcal{W}$ space for FFHQ, AFHQ, and LSUN Cars are much denser and entangled than CIFAR10 because they are trained on not very diverse data like CIFAR10. That is why the learned SFVQ curve shown in Fig. 4(a) does not show a perfect distinctive universal interpretation. We provided a similar figure for a 6 bit SFVQ for the AFHQ dataset in Appendix A.1.

As the third experiment, we examined a 2 bit SFVQ applied on the $\mathcal{W}$ space of StyleGAN2 pretrained on the FFHQ dataset and displayed the generated images in Fig. 4(b). We observe a clear separation between females and males, while we only have two individual identities, each representing the average face for females and males. From this SFVQ curve, we can infer some more interesting properties. We hypothesize that each SFVQ line corresponds to an interpretable direction shown in Fig. 4(c). Direction I (direction from codebook vector 1 to 2) is for changing rotation to the right, direction II refers to the gender change, and direction III is for changing rotation to the left. We also compute the angles between these directions in degrees, which somehow confirms our hypothesis. Direction II is almost orthogonal to two other directions, and directions I and III are approximately inverse.

4.2 StyleGAN2: interpretable directions

Fig. 4(b) reminds us of the PCA-based method of GANSpace (Härkönen et al., 2020) which finds PCA directions as interpretable directions. Similar to the first two PCA directions of GANSpace that refer to the change of gender and rotation, the SFVQ lines are also located along the directions in which the training data has the most variance, i.e. gender and rotation. However, in the pretrained $\mathcal{W}$ space of StyleGAN2, the interpretable directions are not necessarily orthogonal to each other, and that is why only the first 100 (out of 512) GANSpace's PCA orthogonal directions lead to noticeable changes (Härkönen et al., 2020). In contrast, in the SFVQ curve, each direction (SFVQ's line) can potentially work for a meaningful and obvious change. These observations and discussions motivate us to use the SFVQ's curve to discover interpretable directions, which we study in the following.

We applied SFVQ curves (from 2 to 12 bit) on the $\mathcal{W}$ space of StyleGAN2 pretrained on FFHQ, AFHQ, and LSUN Cars datasets and observed the generated images of SFVQ curves. By observation, we spotted some useful interpretable directions, shown in Fig. 5. Columns (a) and (b) represent the discovered direction from two SFVQ's subsequent codebook vectors, column (c) is the test vector in the latent space to which we apply the direction, and column (d) is the final result after applying the direction. Similar to the GANSpace naming convention, the term Wi-Wj means we only manipulate the style blocks within the range [i-j]. Note that we take the directions only from SFVQ's subsequent codebook vectors, but not from two necessarily similar though far apart codebook vectors. Otherwise, one can accidentally find directions by taking two codebook vectors from an ordinary VQ that might lead to a meaningful direction. To show the practicality of the directions better, we applied them only on one identical test image (except for the *Beard* and *Bald* directions which are specified to males).

One great advantage of our proposed method over some other approaches is that it almost keeps the identity of the test image (column (c)) fixed when applying the interpretable directions. Another advantage is that we could find some new and unique directions that were not found in previous methods, such as *Hat*, *Beard* for FFHQ, *Age*, *Bicolor* for AFHQ, and *Classic* for LSUN Cars. These unique directions are not limited only to these ones, as users can find other directions by their own observations. More importantly, our approach detected an inclusive set of directions, whereas other methods in the literature could only find a portion of them. It is important to note that the directions for the AFHQ dataset are class-agnostic (see Appendix A.2), i.e. the direction for one animal works for other animal species because in Fig. 5 we found the directions from *Wolf* and *Cat* classes, but we applied them to a *Dog* class. However, some directions do not necessarily work for all animal species in the AFHQ because the transformations are restricted by the dataset bias of individual animal classes (Jahanian et al., 2019) (see Appendix A.2). Another interesting observation is how the *Hat* direction (discovered for males) works logically but differently for females.

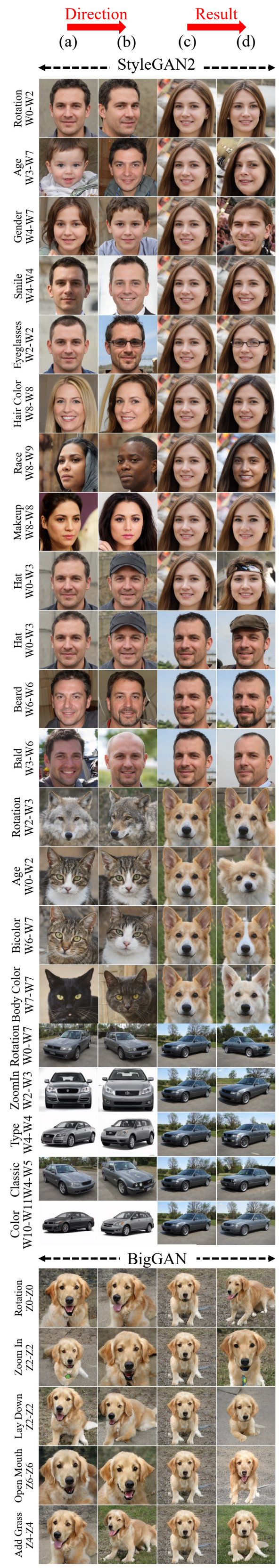

Figure 5: SFVQ interpretable directions.

4.3 BigGAN: interpretable directions

BigGAN (Brock et al., 2018) samples a random vector z from a normal prior distribution $p(z)$ and maps it to an image. Since in BigGAN the intermediate layers also take the random vector z as input (i.e. *skip-z connections*), the vector z has the most effect on the generated output image. Hence, we should find the semantic directions in $p(z)$ space. However, as $p(z)$ is an isotropic distribution, it is difficult to find useful directions from it (Härkönen et al., 2020). Therefore, similar to GANSpace, we first train the SFVQ on the first linear layer ($\mathcal{L}$) of

BigGAN to search for interpretable directions within this space, and afterward, we transfer these directions back to $p(z)$ space. To this end, we sample 10^6 random vectors from $p(z)$ and generate their corresponding vectors in $\mathcal{L}$ space. We map these vectors (in $\mathcal{L}$) to the learned SFVQ codebook vectors such that each sample will be mapped to its closest codebook vector using Euclidean distance. Therefore, for each codebook vector in $\mathcal{L}$, we have its corresponding samples in $p(z)$. Finally, for each codebook vector in $\mathcal{L}$, we compute its corresponding codebook vector in $p(z)$ by taking the mean of the vectors in $p(z)$ which get mapped to this SFVQ codebook vector. We obtain the corresponding SFVQ curve in $p(z)$ by doing this operation for all codebook vectors. Now, we use this computed SFVQ codebook (in $p(z)$) to interpret the latent space of BigGAN. Note that to compute the SFVQ curve for BigGAN, we selected a class label and kept it fixed.

We computed the SFVQ curve over different bitrates (from 2 to 12 bit) in the $p(z)$ space of BigGAN for *golden retriever* class and discovered some interpretable directions, which are shown in Fig. 5. Columns (a) and (b) represent the discovered direction from two SFVQ's subsequent codebook vectors, column (c) is the test vector in the $p(z)$ to which we apply the direction, and column (d) is the final result after applying the direction. Similar to the GANSpace naming convention, the term Zi-Zj means we only manipulate the *skip-z connections* within the range [i-j]. Apart from basic geometrical directions (*Rotation* and *Zoom In*), we discovered some more specific directions such as *Lay Down* and *Open Mouth* as found in Yüksel et al. (2021), and *Add Grass* as found in Härkönen et al. (2020). Note that the discovered directions by SFVQ for *golden retriever* class are class-agnostic, i.e. they also work for other classes (see Appendix A.3).

4.4 Qualitative comparison

We compared our interpretable directions with GANSpace (Härkönen et al., 2020), LatentCLR (Yüksel et al., 2021), and SeFa (Shen & Zhou, 2021) qualitatively and quantitatively. The reason for choosing these methods is that their interpretable directions for StyleGAN2-FFHQ were readily available in their GitHub repositories. Hence, we skipped other methods that were not trained on StyleGAN2-FFHQ or did not share their directions. We focus on StyleGAN2-FFHQ for comparisons, because we planned to use the pretrained networks of Zhang et al. (2017); Karkkainen & Joo (2021); Jiang et al. (2021); Doosti et al. (2020); Deng et al. (2019) for face attributes rating for our quantitative comparisons. For Sefa, there were no annotations for the discovered directions of StyleGAN2-FFHQ. Hence, we used their interactive tool to examine their first $K = 25$ semantics and find their interpretable directions. In SeFa's interactive tool, there were only three possible options for the layer-wise edits ($W0$-$W1$, $W2$-$W5$, $W6$-$W13$), but to have a fair comparison, we further searched precisely and found the best layer-wise edits for each direction. We provided the information of SeFa's interpretable directions in Appendix A.5.

Fig. 6 shows the qualitative comparison. We provided similar comparison over 50 different random vectors in Supplementary Material (shown as $\mathcal{SM}$ in the rest of the paper). To have a fair comparison, we use the same amount of shift (σ) toward each direction because, as mentioned in Tzelepis et al. (2021), it is the advantage of a direction if it reaches the desired change in the attribute within a shorter path. The image in the red square is the initial test image to which we apply the changes. For *Smile* direction, our method opens and closes the smile effectively in both positive and negative paths while keeping the identity and age attributes almost fixed. Whereas GANSpace is highly entangled with age attribute and SeFa changes the identity. In *Hair Color* direction, our method keeps the identity better than the others. However, GANSpace and SeFa alter the face highlights, LatentCLR is highly entangled with gender, and SeFa is highly entangled with age. For *Age* direction, our method covers a wider range of ages than LatentCLR, while LatentCLR and SeFa are highly entangled with gender, and SeFa alters the identity too much. For *Gender* direction, GANSpace and SeFa are entangled with age, and our method remains in the valid range of generations better than GANSpace. In *Bald* direction, our method keeps the identity better than SeFa, while SeFa adds beard to the face, and our method renders a better baldness than LatentCLR. To see a more comprehensive qualitative comparison, we encourage the readers to make subjective comparisons with different random vectors using our GitHub demo directory, or to inspect subjective comparisons over 50 random vectors in $\mathcal{SM}$.

4.5 Quantitative comparison

For quantitative comparison, we adopted the evaluation criteria and pretrained networks (to rate an image's attributes) used in Tzelepis et al. (2021) and Aoshima & Matsubara (2023). We used Zhang et al. (2017) to spot the face bounding box, FairFace (Karkkainen & Joo, 2021) to rate the age, race, and gender attributes, CelebA-HQ (Jiang et al., 2021) to measure the smile attribute, Hopenet (Doosti et al., 2020) to find the face direction (yaw, pitch, roll attributes), and ArcFace (Deng et al., 2019) to evaluate how much face's identity is preserved after shifting along a direction.

We sampled 10^3 vectors from $\mathcal{N}(0, 1)$ and generated their corresponding latent vectors in the $\mathcal{W}$ space of StyleGAN2. Following Tzelepis et al. (2021), to assess a discovered direction for each latent vector, we create a sequence of images by shifting the latent vector for 20 steps in both positive and negative paths along that direction. Therefore, each sequence contains 41 images such that the original intact image is in the middle. Then, for each image within this sequence, we use the above-mentioned pretrained networks to measure its attributes. Next, we calculate the correlation between the step index (from 1 to 41) and each attribute scores. Thus, for each direction, we obtain a vector of seven correlation values (one per each attribute) which is then L1-normalized similar to Tzelepis et al. (2021). Table 1 shows the results averaged over 10^3 latent vectors for our method, GANSpace, LatentCLR and SeFa.

Table 1 shows that for *Gender* direction, our method works better than others with a higher correlation to gender attribute, whereas GANSpace and SeFa are correlated with age and race attributes. For *Age* direction, our method has almost the same correlation to age attribute as LatentCLR, but higher than SeFa. However, LatentCLR and SeFa remarkably change the gender and race attributes, which is undesirable, whereas our method changes the smile and pitch attributes, which is visually more acceptable (see Fig. 6). For *Smile* direction, SeFa renders a higher correlation to smile attribute than others. However, the smile direction of GANSpace and LatentCLR methods are improperly correlated the most with age and pitch attributes, respectively. In *Rotation* direction, similar to other methods, our method is mainly correlated with the yaw attribute but with less correlation than GANSpace and LatentCLR, while it changes other face rotation's attributes (i.e. pitch and roll) more than them. Our *Rotation* direction causes less changes in gender attributes compared to others, specifically SeFa. Note that in Table 1, if a method is not listed for a direction, it means, that direction does not exist for the method.

We also compared our method with others on how they can preserve the identity when shifting the latent vectors for various shift values over different directions. Table 2 provides the identity scores (averaged over 10^3 latent vectors) that range from 0 to 1, such that a higher value means a higher similarity to the original test image in terms of identity. We observe that our method keeps the identity better than others with a big margin for *Smile*, *Race*, and *Hair Color* directions. For *Gender* direction, our method perform much better than GANSpace and comparable to SeFa. In *Age* direction, our method performs comparable to LatentCLR and SeFa. Based on qualitative comparisons (Fig. 6), since LatentCLR apply minimal changes to the face compared to others, it gives higher identity scores for *Rotation* and *Bald* directions. Ignoring the LatentCLR scores, our method performs comparable to GANSpace and SeFa in *Rotation* direction, and bet-

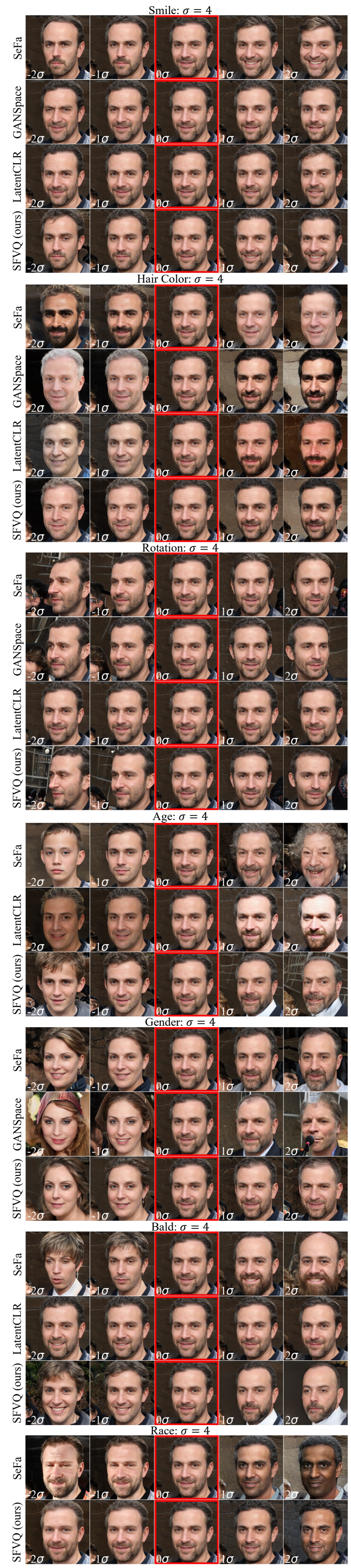

Figure 6: Qualitative Comparison.

ter than SeFa for *Bald* direction. Furthermore, we computed the *commutativity error* (defined in Aoshima & Matsubara (2023)) for our method, GANSpace, LatentCLR, and SeFa over all directions. As expected, all four methods are *commutative* because they all apply linear transformations on the latent vectors. In the end, we believe that the best way to compare the interpretable directions between different methods is still the subjective comparisons. Therefore, to better see the efficiency of our discovered directions over other methods, we encourage the readers to make subjective comparisons with different random vectors using our GitHub demo directory, or to inspect subjective comparisons over 50 random vectors in $\mathcal{SM}$.

4.6 Ablation study

We did an ablation study on the effect of different SFVQ's bitrates (from 2 to 12 bit) on the interpretations of StyleGAN2 models on FFHQ, AFHQ, and LSUN Cars datasets. Regarding *universal interpretation*, for all bitrates, we observe the inherent structure in SFVQ's subsequent codebook vectors sharing similar generative factors such as rotation, background, and accessories for FFHQ. When increasing the bitrate, we see more diversity in the images (e.g. more identities for FFHQ) because we model the latent space with more clusters (or codebook vectors). We provided images corresponding to SFVQ codebooks from 2 to 8 bit in Appendix A.6.

Regarding *interpretable directions*, a higher SFVQ bitrate allows the curve to get more turned and twisted in the latent space, increasing the chance of spotting more detailed or intricate directions. Based on our investigations, the directions that alter images more structurally can be found from lower bitrates and vice versa. For example, for StyleGAN2-FFHQ, we found *rotation*, *gender* and *age* directions from 2, 5, and 6 bit SFVQ, respectively. On the other hand, we detected the directions that cause a partial change on the face such as *smile*, *hair color*, *makeup*, *race*, and *bald* from 12 bit SFVQ.

Table 1: Quantitative comparison of interpretable directions. Values in each row show the L1-normalized correlation of each direction to the attributes.

Direction	Method	Gender	Age	Smile	Race	Yaw	Pitch	Roll
Gender	GANSpace	0.63	0.12	0.047	0.13	0.0093	0.05	0.0074
	SeFa	0.74	0.11	0.06	0.06	0.018	0.0056	0.01
	SFVQ (Ours)	**0.87**	0.0027	0.037	0.039	0.011	0.031	0.0052
Age	LatentCLR	0.31	**0.38**	0.034	0.24	0.014	0.0049	0.0067
	SeFa	0.48	0.25	0.081	0.14	0.0074	0.029	0.013
	SFVQ (Ours)	0.11	0.37	0.14	0.018	0.09	0.24	0.057
Smile	GANSpace	0.047	0.53	0.0052	0.36	0.0005	0.057	0.0008
	LatentCLR	0.15	0.15	0.17	0.1	0.052	0.32	0.056
	SeFa	0.26	0.12	**0.46**	0.023	0.0011	0.12	0.0047
	SFVQ (Ours)	0.31	0.078	0.4	0.077	0.022	0.052	0.061
Race	SeFa	0.066	0.17	0.34	0.37	0.0057	0.027	0.025
	SFVQ (Ours)	0.037	0.07	0.33	**0.52**	0.0015	0.031	0.0011
Rotation	GANSpace	0.11	0.037	0.0023	0.022	**0.76**	0.063	0.0032
	LatentCLR	0.12	0.051	0.11	0.0027	0.67	0.032	0.01
	SeFa	0.3	0.024	0.01	0.0049	0.51	0.15	0.0036
	SFVQ (Ours)	0.084	0.013	0.16	0.021	0.58	0.07	0.072
Bald	LatentCLR	0.25	0.083	0.17	0.16	0.032	0.23	0.073
	SeFa	0.24	0.32	0.053	0.31	0.024	0.013	0.043
	SFVQ (Ours)	0.47	0.14	0.12	0.019	0.059	0.16	0.027
HairColor	GANSpace	0.006	0.084	0.47	0.4	0.012	0.014	0.0083
	LatentCLR	0.33	0.099	0.27	0.22	0.0009	0.03	0.046
	SeFa	0.027	0.4	0.43	0.08	0.02	0.013	0.027
	SFVQ (Ours)	0.11	0.015	0.47	0.38	0.0022	0.012	0.019

4.7 Joint interpretable directions

By observing images of the SFVQ's curve (Fig. 4(a)) to find interpretable directions, we can also discover joint interpretable directions from subsequent codebook vectors that differ in multiple attributes. By *joint*, we mean to change, for example, *rotation* and *gender* attributes simultaneously. In fact, joint directions are the directions in which multiple attributes are entangled. Supervised methods cannot find joint directions because they use pretrained networks or labeled data with respect to only one attribute. Furthermore, finding joint directions will be laborious for the unsupervised methods of Härkönen et al. (2020); Shen & Zhou (2021); Voynov & Babenko (2020); Yüksel et al. (2021); Tzelepis et al. (2021); Aoshima & Matsubara (2023) because 1) their training strategy was not designed for this task, 2) they have to blindly search over all K found directions and hope to find the direction to change their desirable joint attributes. However in our method, the prior knowledge of potential directions achieved by observing the SFVQ's curve helps to find the desirable joint directions quickly. Fig. 7 shows some joint directions found by our proposed method. Note that the joint directions are not limited to these, as users can discover their desired directions by their own inspections.

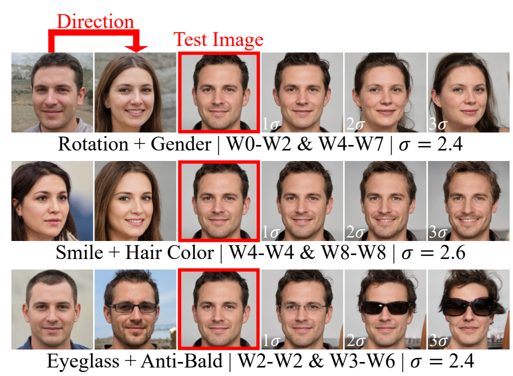

Figure 7: Some examples of SFVQ joint interpretable directions.

Figure 8: Generated images from 20 equally-spaced points on the line connecting two neighboring codebook vectors of VQ and two subsequent codebook vectors of SFVQ trained on the $\mathcal{W}$ space of StyleGAN2 pre-trained on FFHQ dataset.

4.8 Controllable data augmentation

According to the training objective of SFVQ to map input vectors on the line connecting subsequent codebook vectors, SFVQ has the property that its lines are mainly located inside the distribution's space. This property is desirable for controllable data augmentation because we have many meaningful points (located on SFVQ's curve) available to generate valid images. By looking at images corresponding to SFVQ's curve in Fig. 4(a), we have an idea of the possible generations from each part of the curve. For instance, to generate baby-aged faces we select 20 equally-spaced points on the line connecting codebook vectors of indices 6 to 7 in Fig. 4(a), and we plot the generations corresponding to these 20 points in the middle row of Fig. 8. In a similar way, we take 20 equally-spaced points on the line connecting two subsequent codebook vectors of indices 15 and 16 in Fig. 4(a) and generate their corresponding images in the bottom row of Fig. 8. We observe that all 20 generations contain the *hat* accessory for a male person.

Table 2: Identity preservation scores of interpretable directions for our proposed method, GANSpace, and LatentCLR. Values in a row show the identity scores for different shifts (σ).

Direction	Method	1-4σ	5-8σ	9-12σ	13-16σ	17-20σ
Gender	GANSpace	0.85	0.58	0.39	0.22	0.078
	SeFa	**0.94**	**0.78**	**0.63**	**0.51**	**0.4**
	SFVQ (Ours)	0.93	0.76	0.61	0.47	0.35
Age	LatentCLR	0.93	0.73	0.53	0.39	**0.29**
	SeFa	**0.95**	**0.78**	**0.59**	**0.42**	**0.29**
	SFVQ (Ours)	0.94	0.76	0.56	0.39	0.25
Smile	GANSpace	0.94	0.76	0.57	0.41	0.29
	LatentCLR	0.95	0.8	0.63	0.47	0.32
	SeFa	0.93	0.78	0.62	0.47	0.35
	SFVQ (Ours)	**0.96**	**0.85**	**0.72**	**0.59**	**0.48**
Race	SeFa	0.93	0.7	0.48	0.33	0.22
	SFVQ (Ours)	**0.98**	**0.88**	**0.74**	**0.6**	**0.48**
Rotation	GANSpace	0.93	0.77	0.62	0.51	0.42
	LatentCLR	**0.98**	**0.92**	**0.85**	**0.79**	**0.75**
	SeFa	0.93	0.77	0.62	0.5	0.4
	SFVQ (Ours)	0.93	0.76	0.61	0.49	0.4
Bald	LatentCLR	**0.96**	**0.86**	**0.73**	**0.61**	**0.51**
	SeFa	0.95	0.8	0.64	0.49	0.36
	SFVQ (Ours)	**0.96**	0.84	0.7	0.55	0.41
HairColor	GANSpace	0.98	0.88	0.75	0.62	0.51
	LatentCLR	0.96	0.81	0.63	0.47	0.36
	SeFa	0.97	0.86	0.71	0.56	0.44
	SFVQ (Ours)	**0.99**	**0.97**	**0.94**	**0.89**	**0.83**

We also take the line connecting two neighboring codebook vectors (under Euclidean distance) of a 5 bit VQ and plot similar generations in the top row of Fig. 8. To have more diverse generations, for all generations in Fig. 8, we added normal noise ($\mathcal{N}(0, 0.3)$) to the selected points. As expected, all generations of SFVQ consistently follow the properties of their corner points, such that they are all faces of babies or males wearing hats. However, the generations for two neighboring codebook vectors of VQ do not follow any specific rule as we observe changes in gender, age, and race among them. Thus, here, by *controllable*, we mean that the users have control over what type of images with specific characteristics they intend to generate.

5 Conclusions

Generative adversarial networks (GANs) are well-known image synthesis models widely used to generate high-quality images. However, there is still not sufficient control over generations in GANs because their latent spaces act as a black box and are thus hard to interpret. In this paper, we used the unsupervised space-filling vector quantizer (SFVQ) technique to get a universal interpretation of the latent distribution of GANs and to find their interpretable directions. Our experiments showed that the SFVQ can capture the underlying morphological structure of the latent space and discover better and more consistent interpretable directions compared to GANSpace, LatentCLR, and SeFa methods. SFVQ gives the user proper control for generating images and manipulating them, and reduces the search effort for finding the desired direction of a change. SFVQ is a generic tool for modeling distributions that is neither restricted to any specific neural network architecture nor any data type (e.g. image, video, speech, and etc.).

References

Rameen Abdal, Peihao Zhu, Niloy J Mitra, and Peter Wonka. Styleflow: Attribute-conditioned exploration of stylegan-generated images using conditional continuous normalizing flows. *ACM Transactions on Graphics (ToG)*, 40(3):1–21, 2021.

Yuval Alaluf, Omer Tov, Ron Mokady, Rinon Gal, and Amit Bermano. Hyperstyle: Stylegan inversion with hypernetworks for real image editing. In *Proceedings of CVPR*, pp. 18511–18521, 2022.

Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017.

Takehiro Aoshima and Takashi Matsubara. Deep curvilinear editing: Commutative and nonlinear image manipulation for pretrained deep generative model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5957–5967, 2023.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2018.

Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8188–8197, 2020.

Nicos Christofides. Worst-case analysis of a new heuristic for the traveling salesman problem. *Graduate School of Industrial Administration, Carnegie Mellon University*, 1976.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, 2009.

Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4690–4699, 2019.

Bardia Doosti, Shujon Naha, Majid Mirbagheri, and David J Crandall. Hope-net: A graph-based model for hand-object pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 6608–6617, 2020.

Merrill M Flood. The traveling-salesman problem. *Operations research*, 4(1):61–75, 1956.

Lore Goetschalckx, Alex Andonian, Aude Oliva, and Phillip Isola. Ganalyze: Toward visual definitions of cognitive image properties. In *Proceedings of ICCV*, pp. 5744–5753, 2019.

Ian 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.

Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. In *Proceedings of NeurIPS*, pp. 9841–9850, 2020.

Ali Jahanian, Lucy Chai, and Phillip Isola. On the" steerability" of generative adversarial networks. In *International Conference on Learning Representations*, 2019.

Yuming Jiang, Ziqi Huang, Xingang Pan, Chen Change Loy, and Ziwei Liu. Talk-to-edit: Fine-grained facial editing via dialog. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13799–13808, 2021.

David S Johnson and Lyle A McGeoch. The traveling salesman problem: A case study in local optimization. *Local search in combinatorial optimization*, 1997.

Kimmo Karkkainen and Jungseock Joo. Fairface: Face attribute dataset for balanced race, gender, and age for bias measurement and mitigation. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 1548–1558, 2021.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.

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/CVF conference on computer vision and pattern recognition*, pp. 8110–8119, 2020.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

Hongyu Liu, Yibing Song, and Qifeng Chen. Delving stylegan inversion for image editing: A foundation latent space viewpoint. In *Proceedings of CVPR*, pp. 10072–10082, 2023.

Hamza Pehlivan, Yusuf Dalva, and Aysegul Dundar. Styleres: Transforming the residuals for real image editing with stylegan. In *Proceedings of CVPR*, pp. 1828–1837, 2023.

Antoine Plumerault, Hervé Le Borgne, and Céline Hudelot. Controlling generative models with continuous factors of variations. *arXiv preprint arXiv:2001.10238*, 2020.

Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *ACM Transactions on graphics (TOG)*, 42(1):1–13, 2022.

Hans Sagan. *Space-filling curves*. Springer Science & Business Media, 2012. URL `https://doi.org/10.1007/978-1-4612-0871-6`.

Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 1532–1540, 2021.

Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9243–9252, 2020.

Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. Interfacegan: Interpreting the disentangled face representation learned by gans. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(4): 2004–2018, 2022. doi: 10.1109/TPAMI.2020.3034267.

Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

Yue Song, T Anderson Keller, Nicu Sebe, and Max Welling. Latent traversals in generative models as potential flows. *arXiv preprint arXiv:2304.12944*, 2023a.

Yue Song, Jichao Zhang, Nicu Sebe, and Wei Wang. Householder projector for unsupervised latent semantics discovery. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 7712–7722, 2023b.

Christos Tzelepis, Georgios Tzimiropoulos, and Ioannis Patras. Warpedganspace: Finding non-linear rbf paths in gan latent space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6393–6402, 2021.

Mohammad Hassan Vali and Tom Bäckström. Interpretable latent space using space-filling curves for phonetic analysis in voice conversion. In *Proceedings of Interspeech*, 2023.

Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the gan latent space. In *Proceedings of International Conference on Machine Learning*, pp. 9786–9796, 2020.

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Guilin Liu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. Video-to-video synthesis. *Advances in Neural Information Processing Systems*, 31, 2018a.

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of CVPR*, pp. 8798–8807, 2018b.

Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for stylegan image generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12863–12872, 2021.

Huiting Yang, Liangyu Chai, Qiang Wen, Shuang Zhao, Zixun Sun, and Shengfeng He. Discovering interpretable latent space directions of gans beyond binary attributes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12177–12185, 2021.

Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015.

Oğuz Kaan Yüksel, Enis Simsar, Ezgi Gülperi Er, and Pinar Yanardag. Latentclr: A contrastive learning approach for unsupervised discovery of interpretable directions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 14263–14272, 2021.

Shifeng Zhang, Xiangyu Zhu, Zhen Lei, Hailin Shi, Xiaobo Wang, and Stan Z Li. S3fd: Single shot scale-invariant face detector. In *Proceedings of the IEEE international conference on computer vision*, pp. 192–201, 2017.

A Appendix

A.1 StyleGAN2: universal interpretation of AFHQ dataset

Similar to what was discussed in Sec. 4.1 of the paper, we apply the SFVQ to capture a universal morphology of the latent space, and we expect that subsequent codebook vectors in SFVQ refer to similar images. Hence, we applied a 6 bit SFVQ on the $\mathcal{W}$ space of the StyleGAN2 model pretrained on the AFHQ dataset. Images corresponding to the SFVQ's codebook vectors are represented in Fig. 9. We can see that similar animal species are generally located next to each other. In addition, there are some other similarities among neighboring codebook vectors, such as change in rotation (from right to left) when moving from index 0 to index 10, change in rotation (from left to right) when moving from index 26 to index 34, light-colored animals for indices 22-25, bi-colored animals for indices 26-29, and baby-aged cats for indices 61-62.

Figure 9: Codebook of a 6 bit SFVQ trained on $\mathcal{W}$ space of StyleGAN2 pretrained on AFHQ dataset.

A.2 Class-agnostic directions for StyleGAN2 pretrained of AFHQ dataset

According to what was discussed in Sec. 4.2 of the paper, in this section we aim to test whether and how the discovered direction of *Bicolor* (in Fig. 5 of the paper) is class-agnostic across different AFHQ animal classes. To this end, we applied this direction to all existing animal species in the AFHQ dataset and represented the results in Fig. 10. We observe that this direction works well for *Cat* and *Dog* classes because there exists enough data (i.e. cats and dogs with bicolored faces) within the AFHQ dataset. Therefore, the learned latent space supports this transformation. In addition, this transformation more or less works for *Wolf* class, since *Wolf* looks like *Siberian husky* (which exists in AFHQ dataset), and this transformation leads the *Wolf* class to become similar to a *Siberian husky*. However, the *Bicolor* direction does not work for other animal classes of *Fox*, *Leopard*, *Cheetah*, *Tiger*, and *Lion*. The reason is that the learned latent space is constrained by dataset bias of individual classes (Jahanian et al., 2019). In other words, the learned latent space does not support this transformation for them since there is no image with a bicolored face from these animal classes within the AFHQ dataset. The σ value determines the magnitude of the step we take toward the *Bicolor* direction. To make sure whether this direction works for these five animal classes, we used a larger σ value (bigger steps) for them. We observe that even with larger steps, not only there is no meaningful transformation effect of the desired direction, but also, in the very last step (3σ), the images turn to become unrealistic by having some artifacts.

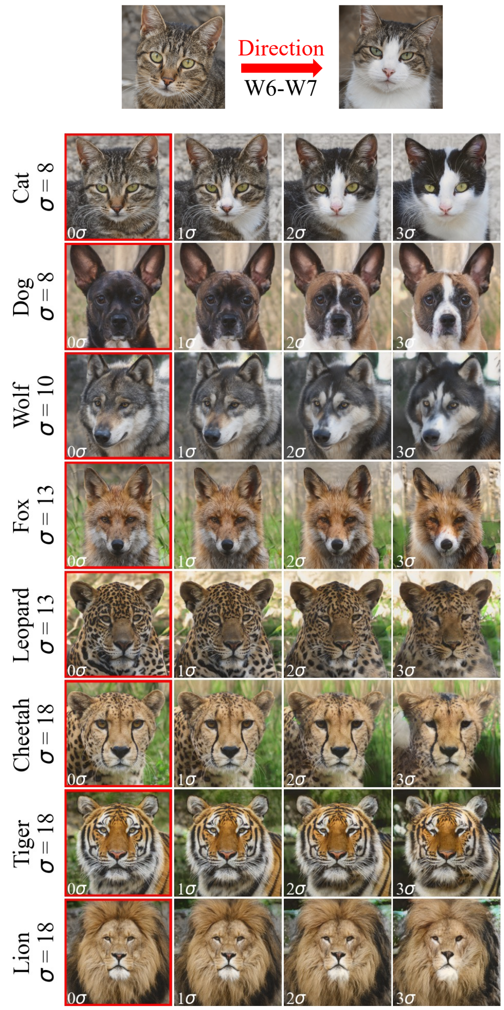

Figure 10: Applying *Bicolor* direction to different animal species of AFHQ dataset.

A.3 Class-agnostic directions for BigGAN pretrained on ImageNet dataset

As discussed in Sec. 4.3 of the paper, we found that the discovered directions by SFVQ (in $p(z)$ space of BigGAN) for the *golden retriever* class are class-agnostic. It means that the detected directions also work when applied to other data classes within the ImageNet dataset. To confirm this, we applied all five directions found for the *golden retriever* (in Fig. 5 of the paper) on the *husky* class, and we illustrated the results in Fig. 11. The image in the middle column (in red square) is the initial test image to which we apply the directions, such that we step along both sides of a direction. According to the figure, all five directions are valid for *husky* class, resulting in meaningful and expected transformations.

A.4 Subsidiary study: traveling salesman problem

Space-filling vector quantization (SFVQ) has some parallels with the classic *traveling salesman* problem (TSP) (Flood, 1956) in bringing order to a set of codebook vectors. One could ask whether we can achieve a better codebook arrangement than SFVQ by applying an ordinary vector quantization (VQ) as usual and, afterward, use one of the *traveling salesman* solutions to reorganize VQ codebook vectors. The scenario of TSP is that we have a list of cities (codebook vectors) and the distances between them, then we aim to discover the shortest possible route to visit each city only once. TSP is an NP-hard problem to solve. We can interpret these cities as the codebook vectors of a VQ. If we learn an 8 bit VQ as usual and intend to rearrange the codebook vectors to achieve the shortest route, then there are 256 ! possible permutations for rearrangement. This is an astronomically large number (8.5×10^{506}). It is thus practically infeasible to

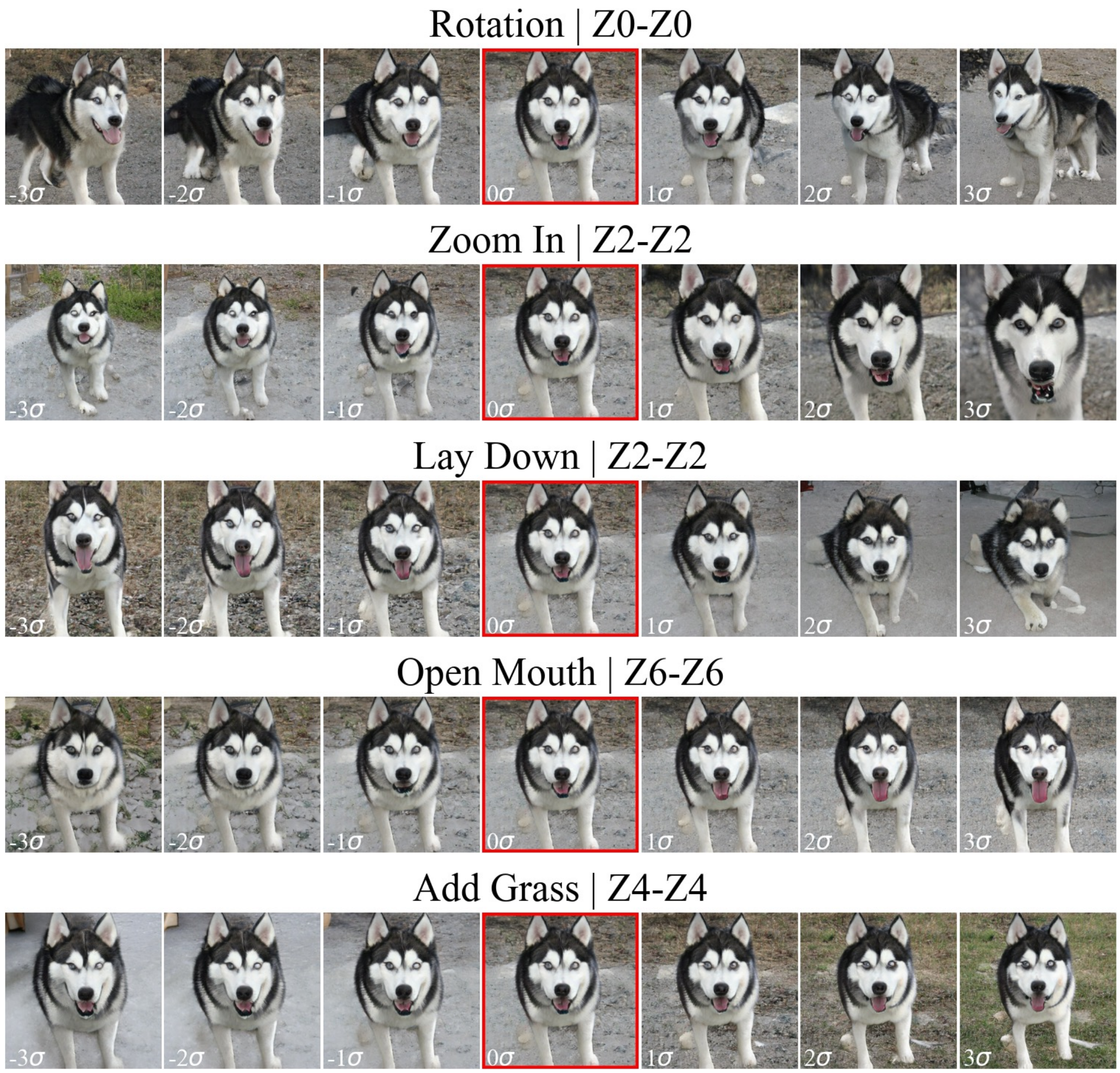

Figure 11: Class-agnostic directions; applying five SFVQ's discovered directions for the *golden retriever* class on the *husky* class using BigGAN pretrained on ImageNet dataset.

do an exhaustive search for all possible permutations in most relatively high bitrate cases of VQ. Hence, it is recommended to use heuristic TSP solvers that have lower computational complexity such as nearest neighbor (Johnson & McGeoch, 1997), greedy (Johnson & McGeoch, 1997), Christofides (Christofides, 1976).

To compare the performance of TSP heuristic solutions with the SFVQ, we examine their ability to model three sparse sample distributions of *circles*, *moons*, and *spiral* in 3D space. We chose the distributions to be sparse because it makes the task more challenging. We trained ordinary VQ and SFVQ with 9 bit (with identical initialization and hyper-parameter settings). After training the VQ, we rearranged its codebook vectors using the nearest neighbor (NN) and Christofides TSP heuristic solvers. Fig. 12 demonstrates the results such that the order in the space-filling line is shown with color coding (light to dark color = first to last codebook vector) for both methods of VQ+TSP and SFVQ. Since the training objective of SFVQ is different from VQ, SFVQ locates the codebook vectors such that the line connecting the subsequent codebook vectors desires mainly to fill up the distribution space and as a result, the line ends up landing inside the distribution space. To affirm this fact, compare the upper and lower parts of *spiral* dataset arranged by VQ+Christofides and VQ+NN methods. VQ locates fewer codebook vectors for these two parts of the *spiral* data, and thus we observe a narrow line that does not fill the distribution's space appropriately. Furthermore, we notice more unfavorable jumps (lines outside the distribution or lines breaking the arrangement) for VQ+TSP methods than the SFVQ due to their improper codebook arrangement. Therefore, we generally observe that the SFVQ achieves a much better codebook arrangement than VQ+TSP for all three distributions.

A.5 SeFa interpretable directions

There were no annotations for SeFa (Shen & Zhou, 2021) interpretable directions for StyleGAN2 pretrained on FFHQ dataset. Therefore, we used the interactive tool provided in SeFa's GitHub repository and examined the first $K = 25$ semantics of StyleGAN2-FFHQ model. In the interactive tool, there were only three options available ($W0$-$W1$, $W2$-$W5$, $W6$-$W13$) to do the layer-wise edits and manipulate the latent vector only for those layers. Hence, to obtain more precise interpretable directions, we further searched for the best layer-wise edits for each semantic (or interpretable direction). We provided the details of SeFa's interpretable directions in Table 3.

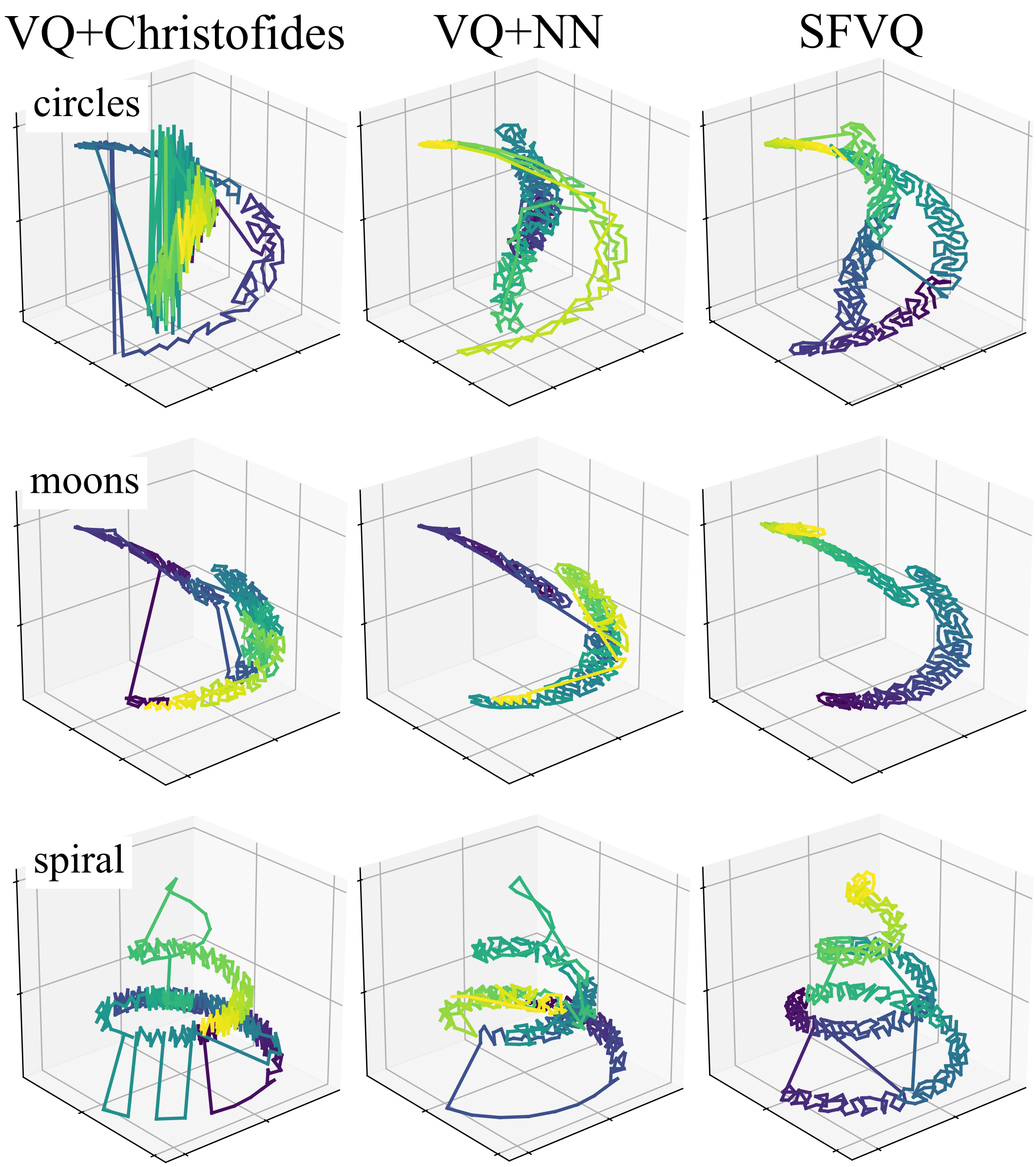

Figure 12: Comparison of the codebook arrangement property of SFVQ with ordinary VQ post-processed by *traveling salesman* heuristic solvers over three sparse distributions.

A.6 Learned SFVQ codebooks on StyleGAN2-FFHQ

As mentioned in Sec. 4.6 of the paper, here we provided the learned SFVQ curves trained on the $\mathcal{W}$ space of pretrained StyleGAN2 on the FFHQ dataset. Fig. 13 to Fig. 19 demonstrate the generated images from learned SFVQ codebooks with the bitrates from 2 to 8 bit. We also provided similar figures for bitrates of 9 to 12 bit in our GitHub repository. The learned SFVQ codebooks and their corresponding generated images for pretrained StyleGAN2 on the FFHQ, AFHQ, and LSUN Cars for the bitrates ranging from 2 to 12 bit are available in our GitHub repository.

Table 3: SeFa (Shen & Zhou, 2021) interpretable directions

Direction	Semantic Index	StyleGAN2 Layers
Gender	2	[4-6]
Rotation	5	[0-2]
Eyeglasses	8	[0-1]
Race	8	[6-11]
Age	9	[3-4]
Bald	15	[4-5]
Hair Color	18	[7-8]
Smile	21	[4-5]
Beard	21	[8-9]
Fat	22	[2-5]
Makeup	24	[6-7]

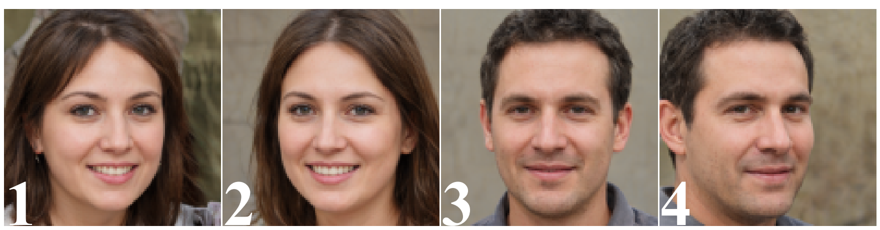

Figure 13: Codebook of a 2 bit SFVQ trained on $\mathcal{W}$ space of StyleGAN2 pretrained on FFHQ.

Figure 14: Codebook of a 3 bit SFVQ trained on $\mathcal{W}$ space of StyleGAN2 pretrained on FFHQ.

Figure 15: Codebook of a 4 bit SFVQ trained on $\mathcal{W}$ space of StyleGAN2 pretrained on FFHQ.

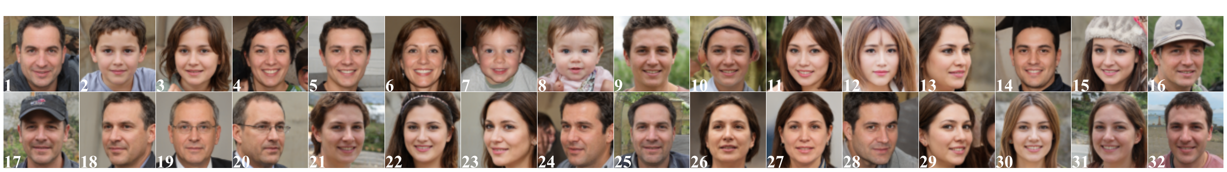

Figure 16: Codebook of a 5 bit SFVQ trained on $\mathcal{W}$ space of StyleGAN2 pretrained on FFHQ.

Figure 17: Codebook of a 6 bit SFVQ trained on $\mathcal{W}$ space of StyleGAN2 pretrained on FFHQ.

Figure 18: Codebook of a 7 bit SFVQ trained on $\mathcal{W}$ space of StyleGAN2 pretrained on FFHQ.

Figure 19: Codebook of a 8 bit SFVQ trained on $\mathcal{W}$ space of StyleGAN2 pretrained on FFHQ.