Fantastic Style Channels and Where to Find Them: A Submodular Framework for Discovering Diverse Directions in GANs

Enis Simsar[1] Umut Kocasari[2] Ezgi Gülperi Er[2] Pinar Yanardag[2]
[1]Technical University of Munich [2]Boğaziçi University
enis.simsar@tum.de umut.kocasari@boun.edu.tr ezgi.er@boun.edu.tr yanardag.pinar@gmail.com

Channels from the same cluster modifying *nose* in FFHQ. Channels from the same cluster modifying *pattern* in Fashion.

Channels from the same cluster modifying *hair* in Metfaces.

Channels from different clusters in LSUN Cars. Channels from different clusters in AFHQ Cats.

Figure 1: Our submodular framework uses the notion of *clusters* to select the most representative and diverse set of *style channels*. Channels performing similar or different manipulations are shown in the clusters above. The input images are displayed in the first column.

Abstract

The discovery of interpretable directions in the latent spaces of pre-trained GAN models has recently become a popular topic. In particular, StyleGAN2 has enabled various image generation and manipulation tasks due to its rich and disentangled latent spaces. However, the discovery of such directions is typically made either in a supervised manner, which requires annotated data for each desired manipulation, or in an unsupervised manner, which requires a manual effort to identify the directions. As a result, existing work typically finds only a handful of directions in which controllable edits can be made. In this study, we design a novel submodular framework that finds the most representative and diverse subset of directions in the latent space of StyleGAN2. Our approach takes advantage of the latent space of channel-wise style parameters, so-called stylespace, in which we cluster channels that perform similar manipulations into groups. Our framework promotes diversity by using the notion of clusters and can be efficiently solved with a greedy optimization scheme. We evaluate our framework with qualitative and quantitative experiments and show that our method finds more diverse and disentangled directions.

1. Introduction

Recent GAN models such as StyleGAN2 [12] and BigGAN [2] have achieved phenomenal success due to their ability to produce images with high visual quality and fidelity. StyleGAN, in particular, introduces a *style-based* approach to transform random latent vectors into realistic images. Unlike traditional GAN architectures [23, 10], style-based designs first transform the random latent vectors into an intermediate latent code using a mapping function, and then modify the channel-wise activation statistics of the model. Due to its rich and disentangled latent spaces, several approaches have been proposed to study the structure

of the latent space of StyleGAN2 in a more principled way [24, 8]. Some of these works aim to discover specific directions such as *expression* or *gender* using supervision [24], while others propose unsupervised approaches to identify semantically meaningful directions [8, 31]. Typically, the identified directions are used to modify the image semantics by shifting the latent code by a certain amount in the identified direction to increase or decrease the desired property. However, while supervised methods such as [24] manage to find the directions the user is interested in, they are limited since it is not always possible to find labeled data for the desired attribute. On the other hand, unsupervised methods such as [8, 31] find a certain number of directions, but the user has to manually explore what these directions are capable of. Not only this approach provide limited insight into the manipulation capabilities of latent space, but it is also time consuming for the user to explore these directions.

Recently, it has been shown that the StyleGAN2 method provides a variety of different latent spaces suitable for different image editing and manipulation tasks. For example, it has been shown that the $\mathcal{W}+$ space is suitable for image inversion [1, 30], while $\mathcal{S}$, the space of channel-wise style parameters (so-called *stylespace*), allows disentangled edits [34]. This space offers rich editing capabilities where an arbitrary style channel is responsible for a particular edit, such as *smile, eye color, hair type*. In other words, it is possible to perform disentangled manipulations by perturbing channel-wise style parameters of the image. While some previous work [21, 34] explores stylespace to find specific channels that perform a desired in a supervised manner, types of manipulations stylespace has to offer in a fine-grained and unsupervised manner has not yet been explored.

In this work we aim to find a subset of *diverse* and *representative* directions in latent spaces. We consider the search for directions in the latent space as a combinatorial optimization problem, where we view the latent space as a discretized set of items using the notion of style channels. Our task is then to select a *subset* of channels that *covers* the stylespace, while respecting the diversity in terms of types of manipulations they perform. This aspect is particularly important since stylespace provides more than 9K style channels and there is redundancy in what these channels cover. In particular, it has been shown that there are over 300 style channels dedicated to the control of the *hair*, and over 180 channels dedicated to the *ears* or *background* [34]. Therefore, an objective function should consider *diversity* into account when covering the stylespace. In other words, if a channel modifying the *hair style* attribute is already selected, the gain of covering another hair style channel should diminish. To address this issue, we design a novel framework that considers representativeness of the channels while incorporating diversity. Our diversity objective benefits from clustering the latent space, where channels that perform similar edits are grouped under the same cluster (see Figure 1). Our framework then ensures that selecting a channel from a cluster that has not yet been explored yields a higher gain. We formulate this task as a monotone submodular function maximization, for which there is a simple greedy algorithm guarantees that the solution obtained is almost as good as the best possible solution [15]. Our contributions are as follows:

- We propose the problem of finding diverse and representative style channels in the latent space of StyleGAN2 and design a submodular objective function that exhibits a natural property of diminishing returns, for which we can efficiently provide a near-optimal solution [17].

- To the best of our knowledge, our framework is the first work to propose a submodular framework for finding latent directions, and the first attempt to provide a complete guide to discovering semantically meaningful *groups* of style channels.

2. Related Work

Recent research has shown that the latent space of GANs contains semantically meaningful directions that can be used for editing images in a variety of ways [8, 9, 31]. Our approach builds on recent successes in discovering disentangled directions using *stylespace*. We also benefit concepts from document summarization in the NLP literature [15] to design our submodular framework.

Several techniques are proposed to exploit the latent structure of GANs in supervised and unsupervised ways. Supervised approaches to exploit the latent space typically use pre-trained classifiers to guide the optimization process and discover directions. [24] trains a Support Vector Machine (SVM) [19] with labeled data such as *age, gender* and *expression*. The normal vector of the resulting hyperplane is used as the latent direction. [7] uses an externally trained classifier to discover directions for cognitive image attributes in the latent space of BigGAN. Other approaches attempt to find interpretable directions in an unsupervised manner. [31] uses a classifier-based technique that finds a collection of directions that correlate with a variety of image modifications. [9] presents an approach that is self-supervised and uses task-specific edit functions. [25] directly uses closed-form optimization of the intermediate weight matrix of GANs and selecting the eigenvectors with the largest eigenvalues as directions. GANSpace [8] uses principal component analysis (PCA) [33] on randomly sampled latent vectors from the intermediate layers of BigGAN and StyleGAN2 and treats the generated principal components as latent directions. [36] uses a self-supervised contrastive learning-based method to discover interpretable

directions in the latent space of pre-trained BigGAN and StyleGAN2 models.

Existing work either provides limited exploration of stylespace in a supervised manner [34] or aims to identify relevant style channels using text-based prompts with a CLIP model [21]. In particular, [34] retrieves relevant channels based on a region or attribute classifier. However, what kind of manipulations stylespace has to offer in a fine-grained way has not yet been explored.

An important component of our framework requires that the channels in the stylespace are grouped into clusters. As discussed later in Section 3, we use the notion of *clusters* to measure the diversity when covering the stylespace. There are several works that use clustering in the latent space of GAN models. We use clustering as a form of identifying similar channels and use this insight as a way to diversify coverage. [5] aims to edit an image based on a particular part of a reference image using k-means [16]. Given a reference image and a target image, they exchange style codes based on regional differences to transfer the appearance of an object. [4] improves upon [5] by finding more successful image-specific manipulation directions and eliminating the per-image matching overhead. [20] clusters the feature maps to find meaningful and interpretable semantic classes that can be used to create segmentation masks. Compared to these methods, we aim to cluster style channels directly based on the regions they modify and use this as a way to diversify channel selection.

3. Methodology

In our work, we view the latent space of StyleGAN2 as a discrete set of items using the notion of style channels in stylespace. The task we are interested in is then to select a subset of representative and diverse channels that *cover* the stylespace. An overview of our framework can be found in Figure 2. Our method benefits from clustering style channels by grouping channels that perform similar manipulations. These clusters are then used in our submodular framework to promote diversity.

3.1. Background on Stylespace

The generation process of StyleGAN2 consists of several latent spaces, namely $\mathcal{Z}$, $\mathcal{W}$, $\mathcal{W}+$, and $\mathcal{S}$. More formally, let $\mathcal{G}$ be a generator which is a mapping function $\mathcal{G} : \mathcal{Z} \rightarrow \mathcal{X}$, where $\mathcal{X}$ is the target image domain. The latent code $\mathbf{z} \in \mathcal{Z}$ is drawn from a prior distribution $p(\mathbf{z})$, typically chosen as a Gaussian distribution. The $\mathbf{z}$ vectors are transformed into an intermediate latent space $\mathcal{W}$ using a mapper function consisting of 8 fully connected layers. The latent vectors $\mathbf{w} \in \mathcal{W}$ are then transformed into channel-wise style parameters and form the *stylespace*, denoted $\mathcal{S}$, which is the latent space that determines the style parameters of the image. It has been shown that [34] style channels provide the most disentangled, complete, and informative space compared to others. However, it is still largely unexplored what style channels are capable of.

3.2. Background on Submodularity

Let $\mathcal{V}$ represent a set of elements $\mathcal{V} = \{v_1, \dots v_n\}$, often called as the *ground set*. Let $\mathcal{F} : 2^{\mathcal{V}} \rightarrow \mathbb{R}$ represent a function that gives a real value for any subset $\mathcal{P} \subseteq \mathcal{V}$. The task we are interested in is then to select a small subset $|\mathcal{P}| \leq n$ that maximizes the function such that $\mathcal{P}^* \in \arg\max_{\mathcal{P} \subseteq \mathcal{V}} \mathcal{F}(\mathcal{P})$. Solving this problem is intractable in general, but it has been shown that a greedy algorithm can be used to solve this equation almost optimally with an approximation factor of $(1 - 1/e)$, under the condition that the function $\mathcal{F}$ is monotone, non-decreasing, and submodular [28]. The greedy algorithm simply starts with an empty set and at each iteration adds the item that maximizes the objective function. In other words, the solution $\mathcal{P}^*$ obtained by the greedy algorithm is a constant factor approximation to the best possible solution (say $\mathcal{P}_{\text{opt}}$) such that $\mathcal{F}(\mathcal{P}^*) \geq (1 - 1/e)\ \mathcal{P}_{\text{opt}} \approx 0.63\ \mathcal{F}(\mathcal{P}_{\text{opt}})$. More formally, submodularity is defined as:

Definition 1 *The function $\mathcal{F}$ is called submodular if for every $\mathcal{P}$ the following inequality holds: $\mathcal{F}(\mathcal{P} \cup \{v\}) - \mathcal{F}(\mathcal{P}) \leq \mathcal{F}(\mathcal{R} \cup \{v\}) - \mathcal{F}(R)$, if $R \subseteq P \subseteq \mathcal{V}$ and $v \in \mathcal{V} \setminus \mathcal{P}$. This form of submodularity directly satisfies the diminishing returns property; the* value *of the addition of v never becomes larger as the context becomes larger [17].*

3.3. A Submodular Framework to Cover Stylespace

Let $\mathcal{V}$ represent the set of style channels in the stylespace. Then, we are interested in selecting a small subset of channels $\mathcal{P} \subseteq \mathcal{V}$ that are most representative and diverse. To measure the overall *coverage* or *fidelity* of the channels in $\mathcal{P}$, we can define a set function as follows,

$$\mathcal{F}_{coverage}(\mathcal{P}) = \sum_{v_i \in \mathcal{V}, v_j \in \mathcal{P}} \mathcal{F}_{\text{sim}}(v_i, v_j) \qquad (1)$$

which simply computes the similarity between the summary set $\mathcal{P}$ and the ground set $\mathcal{V}$. In other words, it measures some form of coverage of $\mathcal{V}$ by $\mathcal{P}$. $\mathcal{F}_{\text{sim}}$ measures the similarity between two channels (see Section 3.3.2).

However, this function does not take diversity into account, since the value of the covering a particular type of edit (such as *hair* or *background*) never diminishes. For example, such a coverage function might favor selecting several background channels without considering diversity, since background is one of the most popular types of edits in stylespace (see Appendix Figure 1. In contrast, if we already have a channel that modifies the background in our summary set $\mathcal{P}$, then we want the gain for selecting another background channel to *decrease*. A common approach is

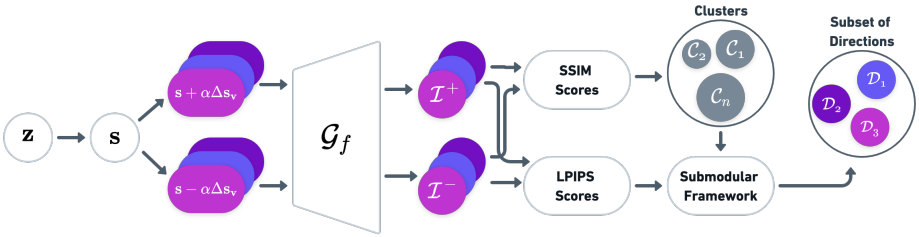

Figure 2: We randomly sample M latent vectors $\mathbf{z} \in \mathcal{Z}$, which are transformed into style vectors $\mathbf{s}$. An arbitrary channel v in $\mathcal{S}$ are perturbed by a certain amount α in positive and negative directions such that $(\mathbf{s} + \alpha\Delta\mathbf{s_v})$ and $(\mathbf{s} - \alpha\Delta\mathbf{s_v})$, where $\Delta\mathbf{s_v}$ is a vector containing all zeros except one of its dimensions, which is equal to one for channel v. LPIPS and SSIM scores are computed for the images obtained from the perturbed vectors, which are then used to generate clusters and select channels using the submodular framework.

to apply a diversity regularization to our objective function [15], where we aim to reward items selected from different groups of directions such that:

$$\mathcal{F}_{diversity}(\mathcal{P}) = \sum_{k=1}^{K} \left(\log \left(1 + \sum_{v_i \in \mathcal{C}_k \cap \mathcal{P}} \mathcal{F}_{\text{reward}}(v_i) \right) \right) \tag{2}$$

where the ground set $\mathcal{V}$ of style channels is partitioned into K separate clusters. The clusters $\mathcal{C}_k$ are disjoint, where $k = 1, \ldots K$ and $\bigcup_k \mathcal{C}_k = \mathcal{V}$. For each style channel v_i, we have a reward $\mathcal{F}_{\text{reward}}(v_i) \geq 0$, which indicates the importance of adding channel v_i to the empty set (see Section 3.3.1).

Let us explain the intuition behind $\mathcal{F}_{diversity}$ in more detail. The idea is that when a channel is selected, the gain decreases for channels from the same cluster due to the concave function $\log(1 + x)$. For example, suppose that the candidate channels in cluster $\mathcal{C}_1$ are v_1 and v_2, which have rewards of 5 and 4, respectively. Similarly, the cluster $\mathcal{C}_2$ has a candidate channel, v_3 with a score of 3. When we evaluate the objective function in Eq. (2) for the first time, we select v_1 since it has the largest marginal gain. However, the next time we choose channel v_3, even though the score of v_2 is higher, because $\log(5 + 4) < \log(5) + \log(3)$. Intuitively, this means that selecting a channel from a cluster that has not yet been explored will yield a higher gain than selecting a channel from a cluster that we already covered. Thus, the objective function rewards diversity by selecting elements from different clusters and prevents popular channels such as *background* from dominating the selected set.

Then, the overall objective function we want to solve is a combination of both:

$$\mathcal{F}(\mathcal{P}) = \mathcal{F}_{coverage}(\mathcal{P}) + \lambda \mathcal{F}_{diversity}(\mathcal{P}) \tag{3}$$

where $\lambda \geq 0$ is the tradeoff coefficient between coverage and diversity. Since we are interested in selecting a small subset, we aim to maximize the following objective function,

$$\mathcal{P}^* = \underset{\mathcal{P} \subseteq \mathcal{V}: |\mathcal{P}| \leq n}{\arg\max} \; \mathcal{F}(\mathcal{P}) \tag{4}$$

subject to a cardinality constraint n, which denotes the total number of channels in the set $\mathcal{P}^*$. This objective function combines two aspects in which we are interested: 1) it encourages the selected set to be *representative* of the stylespace, and 2) it positively rewards *diversity*. Finding the exact subset that maximizes this equation is intractable. However, it has been shown that maximizing a monotone submodular function under a cardinality constraint can be solved near optimally using a *greedy* algorithm [17]. In particular, if a function $\mathcal{F}$ is submodular, monotone and takes only non-negative values, then a greedy algorithm approximates the optimal solution of the Eq. (4) within a factor of $(1 - 1/e)$ [17]. Note that this property is particularly attractive because it is a worst-case bound. In most cases, the quality of the obtained solution of submodular optimization problems is much better than this bound suggests [15].

Theorem 1 *Given two functions $\mathcal{F} : 2^{\mathcal{V}} \to \mathbb{R}$ and $f : \mathbb{R} \to \mathbb{R}$, the composition $\mathcal{F}' = f \circ \mathcal{F} : 2^{\mathcal{V}} \to \mathbb{R}$ is non-decreasing submodular, if $\mathcal{F}$ is non-decreasing concave and $\mathcal{F}$ is non-decreasing submodular. [15]*

Claim 1 : The function in Eq. (3) *is submodular.*

Proof The $\mathcal{F}_{coverage}(\mathcal{P})$ is a sum of modular functions with non-negative weights (hence, monotone). Similarly, the sum of non-negative rewards in $\mathcal{F}_{diversity}(\mathcal{P})$ is also monotone. This monotone function is surrounded by a non-decreasing concave function $\log(1 + x)$. Applying a concave function to a monotone function, we obtain a submodular function (see Theorem 1). Finally, the sum of a collection of submodular functions is submodular [27], so $\mathcal{F}(\mathcal{P})$ in Eq. (3) *is submodular.*

3.3.1 Reward of channels

Our framework requires a singleton reward associated with each style channel for the diversity objective. To this end, we use the LPIPS[37] metric as a proxy for the reward score, where channels with more perceptual changes have a larger value. First, we sample M random latent vectors $\mathbf{z}$

$\in \mathcal{Z}$ and pass them through the mapping network of StyleGAN2 to obtain their corresponding style vectors $\mathbf{s}$. Given an arbitrary channel $v^1 \in \mathcal{S}$, we perturb the value of channel v in each style vector $\mathbf{s}$, while leaving the other channels unchanged, and generate modified images, $\mathcal{G}(\mathbf{s} + \alpha\Delta\mathbf{s_v})$ and $\mathcal{G}(\mathbf{s} - \alpha\Delta\mathbf{s_v})$. $\Delta\mathbf{s_v}$ is a vector containing all zeros except one of its dimensions, which is equal to one for channel v, and α denotes the magnitude of the perturbation. We run both images through the VGG16 [26] network and compute the L2 difference between their feature embeddings. This process is repeated for M style vectors and the average LPIPS score for each channel v is calculated as the reward value $\mathcal{F}_{\text{reward}}(v)$.

3.3.2 Clustering Stylespace

Our method quantifies diversity by using the notion of clusters $\mathcal{C}_k, k = 1, \dots K$, where channels performing similar edits are grouped together. Using the same approach as above, we first obtain the perturbed images for each style vector $\mathbf{s}$ such that $\mathcal{G}(\mathbf{s}+\alpha\Delta\mathbf{s_v})$ and $\mathcal{G}(\mathbf{s}-\alpha\Delta\mathbf{s_v})$. Then we compute the structural similarity index (SSIM) [32], which is a metric for measuring the similarity between two images. In particular, we obtain the image difference between two images, where the difference is represented as a value in the range [0, 255]. This process is repeated for each style channel in $\mathcal{S}$ for a total of M style vectors, resulting in $|\mathcal{S}| \times M$ matrices of SSIM scores. We then compute the cosine distance between the SSIM matrix of each style channel, with the distance between channels averaged over M style vectors. We use the resulting matrix as a distance matrix in agglomerative clustering [6] to cluster style channels into groups. We have experimented with both agglomerative clustering and k-means algorithms and found that they yield similar clusters. We use agglomerative clustering to group the channels since it uses a precomputed distance matrix to speed up the clustering process and does not require tuning the number of clusters. We note that clustering at individual layers leads to finer-grained clusters for models such as FFHQ. For such large models, we perform clustering at each layer and then group the clusters based on the regions they modify (e.g., *hair, ear, background*) using a segmentation model [14].

SSIM scores are also used to compute the similarity between two style channels. Given two style channels v_i and v_j, $\mathcal{F}_{\text{sim}}$ in Eq. (1) is calculated as the cosine similarity between the SSIM matrix of each channel, averaged over M style codes.

4. Experiments

We conduct several qualitative experiments to demonstrate the effectiveness of the submodular framework and compare our method to supervised [34] and unsupervised methods [8, 25]. We also explore clusters of StyleGAN2 on a variety of datasets, including FFHQ [12], LSUN Cars [35], AFHQ Cats [3], Metfaces [11], and Fashion. For Fashion model, we train a StyleGAN2 model with dataset collected from [29, 18]. Finally, we present two applications that leverage our framework to allow users to explore stylespace.

[1]We drop the subscript of v in the rest of this paper for clarity.

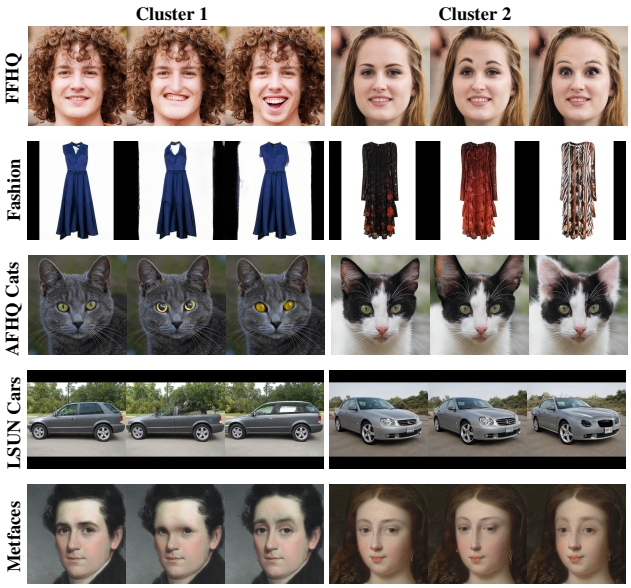

Figure 3: **Sample clusters for various datasets**. The first column represents the input image and the remaining columns show the manipulation performed by a random channel in the cluster.

4.1. Experimental Setup

For all experiments, we use the StyleGAN2 model [11] with truncation value *0.7*. For the LPIPS and SSIM scores, α is set to 20 and the number of style codes is set to $M = 128$. It takes 1 hour to compute LPIPS and SSIM scores. We use Scikit-learn [22] for agglomerative clustering, with the distance threshold parameter set to *0.7*, resulting in about 20 to 40 clusters depending on the layer. Clustering per layer takes 5-15 seconds. Following [34], we exclude RGB layers as they cause entangled manipulations, and we exclude the last 4 blocks as they represent very fine-grained features that are difficult to use for editing tasks. For the submodular framework, we use the diversity tradeoff λ as 25. For our experiments, we use a single NVIDIA Titan RTX GPU.

4.2. Qualitative Results

Clustering Stylespace Our submodular framework relies on the clusters to encourage diversity. Figure 1 and Figure 3 show clusters from the FFHQ, Fashion, AFHQ Cats, LSUN Cars, and Metfaces datasets. We note that clusters that modify similar regions are grouped together, such as

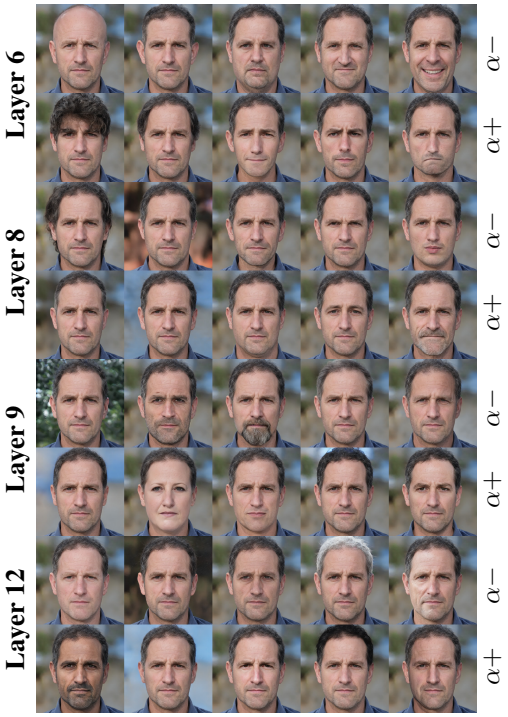

Figure 4: **Top 5 channels ranked by our submodular framework for individual layers**. As can be seen from the results, our method is able to select diverse channels for each layer.

smile, expression in FFHQ, *neck type, pattern* in Fashion, *eye color, ear type* in AFHQ Cats, *roof type, bumper type* in LSUN Cars, *eyebrow type, expression* in Metfaces, shown in Figure 3.

Covering Stylespace Our framework is flexible in terms of which groups of layers to cover. We can choose to cover only channels from a single layer or from multiple layers. In either case, one just needs to form the clusters based on the particular layers of interest. Next, we investigate both cases.

- **Single layers** We first experiment with selecting a subset of channels on single layers. Figure 4 shows the top 5 channels for individual layers $L = 6, 8, 9, 12$. We see that our framework selects diverse channels for each layer, such as channels that modify *hair, ear, face, expression, mouth* as in layer $L = 6$ or *background, gender, beard, hair, expression* as in layer $L = 9$. Note that performing submodular ranking allows us to get the top channels for each layer, but is still not sufficient to cover the stylespace, as channels that perform similar edits may be ranked at top for different layers and cause redundancy. For example, channels that change *background* are placed at the top in different layers (see first and second channels in layers $L = 9, 8, 12$, respectively). Therefore, submodular selection at multiple layers is required to achieve adequate stylespace coverage, as we show below.

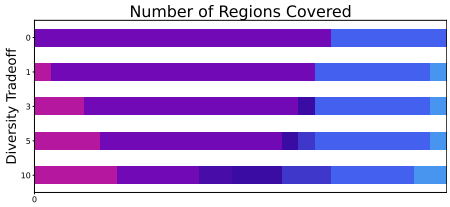

Figure 5: **The effect of the diversity tradeoff**. The number of regions (indicated by different colors) covered by the top 25 channels in FFHQ. Our model covers more regions as we increase the diversity tradeoff λ due to diminishing returns.

- **Multiple layers** Figure 6 shows the top 10 channels ranked by our method considering multiple layers. As can be seen from the results, our method selects a variety of channels that modify regions such as *background, hair, face, mouth, eye, ear, and clothing*. We note that our method places a channel that modifies *background* first, as this is one of the most popular types of editing offered by the stylespace. Covering another *background* channel then has diminishing returns thanks to the submodularity property, and preference is given to channels that modify other diverse regions before placing another background channel at the 8^{th} position.

- **Diversity tradeoff** We also examine the effects of the diversity parameter λ (see Figure 5). When the diversity parameter $\lambda = 0$, we find that the number of regions in the top 25 channels covers only two regions. When we increase the parameter λ, we find that more regions are covered and the balance between regions improves since the submodular framework accounts for diversity.

4.3. Comparison with Unsupervised Methods

Next, we compare our results with the state-of-the-art unsupervised methods GANSpace [8] and SeFa [25]. GANSpace applies PCA to randomly sampled w vectors of StyleGAN2 and uses the resulting principal components as directions. SeFa uses a closed-form approach where it factorizes the weight matrix and uses the resulting eigenvectors with the highest eigenvalues as directions. We used the official implementations for both methods[2] and obtained the top 10 principal components for GANSpace and the top 10 eigenvectors for SeFa methods using the default parameters. Note that since the directions vary by the choice of layers used in SeFa, we experimented with all options (layers 0-1, 2-5, 6-13, all) and chose layers

[2] http://github.com/harskish/ganspace, http://github.com/genforce/sefa

Figure 6: Comparison of the top 10 directions for Ganspace[8], SeFa[25] and our method. The first column shows the original image.

6-13 because they have the most diverse and semantically meaningful directions (see Appendix Figure 4). As can be seen from Figure 6, our method yields more disentangled and diverse directions compared to GANSpace and SeFa. For example, while both GANSpace and SeFA change semantics in the input, such as *gender, age, eyeglasses*, while also changing other semantics such as *background, position, highlight* at the same time. In contrast, our method performs disentangled edits by changing one semantic at a time. To verify our observations, we also conduct a user study with $N = 25$ participants. For the user study, we list the top 10 manipulations of each method along with the original image and ask the following questions:

(Q1) *'How disentangled do you think the change in each image is?' (Note that disentanglement is the degree to which each latent dimension captures at most one attribute.)* (1=Not Disentangled 5=Very disentangled)

(Q2) *'How semantically meaningful do you think the change in each image is?'* (1=Not Semantically Meaningful 5=Very Semantically Meaningful)

Model	GANSpace	SeFa	Ours
Q1	2.46 $\pm$ 0.45	2.91 $\pm$ 0.41	4.32 $\pm$ 0.31
Q2	3.45 $\pm$ 0.41	3.26 $\pm$ 0.28	4.20 $\pm$ 0.29

Table 1: Comparison with GANSpace and SeFa for *Disentanglement* $\uparrow$ (Q1) and *Semantically Meaningful* $\uparrow$ (Q2).

As can be seen from Table 1, our method has more disentangled and semantically meaningful directions. All results are statistically significant with a p-value of < 0.0001. Our method shows a significant performance especially on the *disentanglement* question, with an improvement of 49% over the closest competitor since we operate in $\mathcal{S}$-space, while other methods operate in $\mathcal{W}$-space.

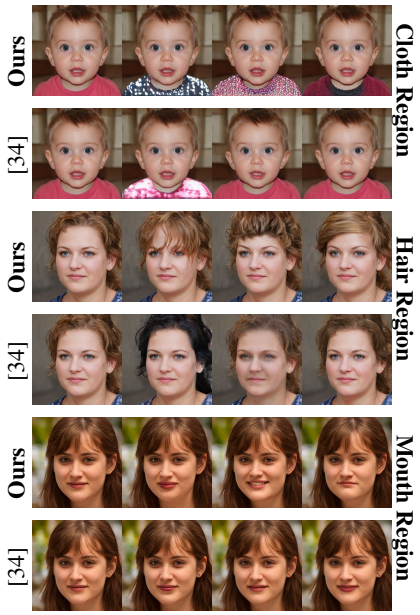

Figure 7: **Comparison of channels retrieved using our method and [34]** for *cloth, hair* and *mouth* regions. Our method can capture more diverse channels than [34].

4.4. Comparison with Supervised Methods

Both our work and [34] use stylespace to find style channels that can be used as directions. While our method proposes an unsupervised method for finding the top channels in stylespace, [34] uses a supervised approach where channels are retrieved based on a specific region (such as *mouth*) or based on a specific attribute classifier. Since [34] does not provide a way to list the top channels in stylespace, we compare our results with [34] as follows: we select three regions; *hair, mouth* and *background*. Then, using the official implementation of [34][3], we determined top 3 channels for a given region. For our method, we determined 3 clusters with the highest match for a given region and selected a random channel from the obtained clusters. Figure 7 shows the results for both methods. As can be seen from the figure, our method is able to obtain diverse channels for the regions *clothing, hairstyle* and *mouth*. To verify our observations, we also conduct a user study with $N = 25$ participants. We list the results for each method with the original image on the left and ask the question *'How diverse do you think the changes are? (1=Not Diverse 5=Very Diverse)'* to participants [4]. As can be seen from the results in Table 2, our method shows significantly better diversity than [34] with a p-value of < 0.0001. This is due to the fact that [34] retrieves channels without considering their similarity, while our method considers channels from different clusters.

Model	[34]	Ours
Cloth	2.26 $\pm$ 1.63	4.32 $\pm$ 0.48
Hair	2.68 $\pm$ 1.16	4.35 $\pm$ 0.18
Mouth	2.16 $\pm$ 0.38	3.64 $\pm$ 0.64

Table 2: Comparison with supervised method [34] on *Diversity* $\uparrow$.

4.5. Applications

Our framework also opens up possibilities for interesting applications that help users discover new directions.

Interactive Editing Users can navigate the stylespace by drawing a region of interest such as *hair* and retrieving relevant clusters and corresponding channels. Figure 8 shows the *background* region with the retrieved clusters (a random channel from each cluster is shown). *See Appendix Figure 2 for more examples*.

Exploration Platform We also provide a web-based platform called *Style Atlas* at `http://catlab-team.github.io/styleatlas` where users can explore the stylespace in a fine-grained way (see Appendix Figure 3). This tool allows users to explore the manipulations made by specific channels based on the region and discover style channels of interest.

[3] `http://github.com/betterze/StyleSpace`

[4] Note that since both methods use the $\mathcal{S}$-space for disentangled edits, we do not compare for disentanglement.

Figure 8: **Filtered clusters based on a region specified by the user**. The two images in the upper left show the input image and the user-specified region. The remaining images show randomly selected channels from the retreived clusters.

5. Social Impact and Limitations

Our method uses a pre-trained GAN model as input, so it is limited to manipulating GAN -generated images. However, it can be extended to real images using GAN inversion methods [38] by encoding the real images into the latent space. Like any image synthesis tool, our method poses similar misuse concerns and dangers, as it can be applied to images of people or faces for malicious purposes, as discussed in [13]. Our method is currently applicable to style-based GAN methods such as StyleGAN2, since it directly benefits from the stylespace. However, we also note that our architecture is applicable to any GAN model where the latent space can be represented as a collection of discretized items. We leave the exploration of our framework to other GAN models such as BigGAN to future work. However, we also note that our architecture is applicable to any GAN model where the latent space can be represented as a collection of discretized items.

6. Conclusion

In this work, we consider the selection of diverse edits in the latent space of StyleGAN2 as a coverage problem. We formulate our framework as a submodular optimization for which we provide an efficient solution. Moreover, we provide a complete guide to the stylespace in which one can explore hundreds of diverse directions formed by style channels using clusters. In our experiments, we have shown that our method can identify a variety of manipulations, and performs diverse and disentangled edits.

Acknowledgments This publication has been produced benefiting from the 2232 International Fellowship for Outstanding Researchers Program of TUBITAK (Project No: 118c321). We also acknowledge the support of NVIDIA Corporation through the donation of the TITAN X GPU and GCP research credits from Google. We thank Irem Simsar for proof-reading our paper.

References

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4432–4441, 2019.

[2] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. *CoRR*, abs/1809.11096, 2018.

[3] Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8185–8194, 2020.

[4] Min Jin Chong, Wen-Sheng Chu, Abhishek Kumar, and David Forsyth. Retrieve in style: Unsupervised facial feature transfer and retrieval, 2021.

[5] Edo Collins, Raja Bala, Bob Price, and Sabine Süsstrunk. Editing in style: Uncovering the local semantics of gans, 2020.

[6] Kazimierz Florek, Jan Łukaszewicz, Julian Perkal, Hugo Steinhaus, and Stefan Zubrzycki. Sur la liaison et la division des points d'un ensemble fini. In *Colloquium mathematicum*, volume 2, pages 282–285, 1951.

[7] Lore Goetschalckx, Alex Andonian, Aude Oliva, and Phillip Isola. Ganalyze: Toward visual definitions of cognitive image properties. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5744–5753, 2019.

[8] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. *arXiv preprint arXiv:2004.02546*, 2020.

[9] Ali Jahanian, Lucy Chai, and Phillip Isola. On the" steerability" of generative adversarial networks. *arXiv preprint arXiv:1907.07171*, 2019.

[10] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[11] Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. In *Proc. NeurIPS*, 2020.

[12] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. *CoRR*, abs/1812.04948, 2018.

[13] Pavel Korshunov and Sébastien Marcel. Deepfakes: a new threat to face recognition? assessment and detection. *arXiv preprint arXiv:1812.08685*, 2018.

[14] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5549–5558, 2020.

[15] Hui Lin and Jeff Bilmes. A class of submodular functions for document summarization. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pages 510–520, 2011.

[16] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[17] George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1):265–294, 1978.

[18] Net-a-porter luxury fashion, beauty & lifestyle for women, https://net-a-porter.com.

[19] William S Noble. What is a support vector machine? *Nature biotechnology*, 24(12):1565–1567, 2006.

[20] Daniil Pakhomov, Sanchit Hira, Narayani Wagle, Kemar E. Green, and Nassir Navab. Segmentation in style: Unsupervised semantic image segmentation with stylegan and clip, 2021.

[21] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. *arXiv preprint arXiv:2103.17249*, 2021.

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[23] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[24] 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*, 2020.

[25] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in gans. *arXiv preprint arXiv:2007.06600*, 2020.

[26] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[27] Peter Stobbe and Andreas Krause. Efficient minimization of decomposable submodular functions. *arXiv preprint arXiv:1010.5511*, 2010.

[28] Maxim Sviridenko. A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters*, 32(1):41–43, 2004.

[29] The global destination for modern luxury, https://farfetch.com.

[30] Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. Designing an encoder for stylegan image manipulation. *ACM Transactions on Graphics (TOG)*, 40(4):1–14, 2021.

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

[32] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[33] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[34] Zongze Wu, Dani Lischinski, and Eli Shechtman. Stylespace analysis: Disentangled controls for stylegan image generation, 2020.

[35] 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.

[36] Oğuz Kaan Yüksel, Enis Simsar, Ezgi Gülperi Er, and Pinar Yanardag. Latentclr: A contrastive learning approach for unsupervised discovery of interpretable directions. *arXiv preprint arXiv:2104.00820*, 2021.

[37] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[38] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *European Conference on Computer Vision*, pages 592–608. Springer, 2020.