Exploring the GANformer for Face Generation

Investigating the segmentation and smile augmentation potential

Romano Ferla
University of Twente, EEMCS, Data Management and Biometrics, Enschede

***Abstract*—Advancing the research in face applications is limited by proprietary databases and increasing data protection regulations, synthetically generated databases may provide a solution. In this work the GANformer, a hybrid generative image model, is explored for this application. While only trained for unconditioned face generation like many other models, this works shows the potential of two use cases. First, the unique implementation of the attention is examined for the application of segmentation. Results indicate segmenting behaviour is present, though post-processing is needed before its implementation in synthetic databases. Second, real labeled faces are reconstructed in latent space to find latent directions describing disentangled attributes. This concept is brought in practice by augmenting neutral to smiling faces, but could be applied on other expressions and attributes as well. In both the segmentation and the smile augmentation the results indicate that the GANformer is able to be used for multiple applications in synthetic database generation. This work can be use as basis as it opens up two directions for further research.**

Fig. 1: Cherry picked examples of smile augmentations on neutral faces using the GANformer.

I. Introduction

The performance of neural networks used in computer vision can increase logarithmically with the size of the training data [1], however larger databases are often not freely available. Common databases such ImageNet [2] (1M images) and VGGFace2 [3] (3.3M faces) are small compared to proprietary databases such as JFT-M300 [1] (300M images) and the dataset used to train FaceNet [4] (∼150M faces). Besides the limited size of the current available databases, the privacy aspect of biometric data makes it harder to compile larger databases. In recent years, privacy regulations have been developed such as the GDPR [5], which demands consent for processing biometric and thus personal data. A database like MS-Celeb-1M [6] (10M faces) was discontinued, it contained faces of journalists and digital right activists without their consent. As a solution, synthetic databases containing generated faces may solve this problem, mitigating the intensive process of collecting faces and risk of privacy breaches.

The generation of synthetic faces has made huge leaps forward in the past couple of years, notably the style based architecture of StyleGAN [7, 8] delivers high quality faces. With conditional generation, generative models can create variations upon these synthetic identities by controlling the output. In the work of Colbois, Freitas Pereira, and Marcel [9] variations of identities can be generated in an automatic manner using StyleGAN2. A benefit of their exploit method is that it can be done automatically and no additional training or networks are needed.

Meanwhile a new machine learning type called transformers was developed in the natural language processing (NLP) field [10]. The transformer utilizes the attention mechanism to enable interaction amongst each input element on a global scale. Unlike of a local interaction associated with the convolution operation, found in a.o. StyleGAN. The successes in NLP [11, 12] inspired the computer vision (CV) field, which in turn have shown promising results in many applications such as object detection and classification [13, 14, 15].

The GANformer by Hudson and Zitnick [16] is a hybrid generative image model, based on the style based architecture of StyleGAN while incorporating transformers. Like many other transformer based works it is unconditioned. Even so, the authors show additional outputs in the form of attention maps with segmenting behaviour. Unfortunately this is not shown in the case of face generation, while the additional segmentation information can yield a multi-purpose database. This makes it an interesting candidate to explore further for the sake of the creation of synthetic databases.

In this work the GANformer is explored and exploited to gain a broader set of functions than the face generator is trained for. First the segmentation potential of the attention mechanism is analysed. Furthermore the conditional generation is enabled by the reconstruction of labeled faces and using these to find latent directions in the latent space, in line with the method of [9]. In particular interest are variations in unambiguous expressions such as smiling. Other attributes e.g. pose and

illumination can also be acquired with other techniques such as 3D modelling. Our main research question is: *To what extent can the GANformer be used for synthetic face database generation?* This will be answered using three additional sub-questions:

- To what extend can we interpret the attention in the synthesis network as a semantic segmentation of the face?
- To what extend can we reconstruct existing identities from the latent space of the GANformer?
- To what extend can we control the smile expression while maintaining the same identity?

The paper is structured as follows: In Section II the related work is discussed. Next the interpretation of the attention is addressed in Section III, after which the reconstruction and semantic control are discussed in Section IV. Both subjects will contain a method that explains how the sub-questions are addressed, an experiments and results sections and a conclusion. The final conclusion in Section V will answer the main research question using the results from the sub-questions.

II. Related Work

A. Transformers

The transformer is a type of neural network block, in which the attention mechanism plays a major role [10]. While a convolution layer acts locally by shifting a kernel along the data, in attention an operation is applied on each input element and therefore enables global interaction between the elements.

A commonly used type of attention is self-attention, in which the data attends to itself. This is used in the Vision Transformer (ViT) of Dosovitskiy et al. [13], a discriminative classifier where all the convolutional layers are replaced by transformers. A downside of this approach is that with many input elements n, the computational load becomes excessive. The fact that every element attends to every other element results in a quadratic computational load $O(n^2)$.

For the interested reader more background information on transformers and the attention mechanism can be found in Appendix A1.

B. Face Generation using Transformers

One of the fields using transformers is synthetic face generation. Many works have been published, which can be roughly divided into two categories. A pure approach replacing all convolutional layers a transformer, such as done in the ViT [13], or using the transformer besides convolutional layers in a hybrid model. An overview of the discussed models is shown in Table I.

TABLE I: An overview of works using a transformer in face generation. Including the use of the transformer, whether the architecture is style based and the possibility for conditional generation.

Works	Transformer		Style-b.	Cond.
	Pure	Hybrid		
TransGAN [17]	✓			
ViTGAN [18]	✓			
HiT [19]	✓			
NutsAndBolts [20]	✓		✓	✓
StyleSwin [21]	✓		✓	
StyleFormer [22]	✓		✓	
VQGAN [23]		✓		✓
GANformer [16]		✓	✓	

Jiang, Chang, and Wang [17] propose TransGAN, a basic GAN architecture, where the commonly used convolutions are replaced by transformers. In such a pure transformer architecture the transformer synthesizes the image. The transformer architecture is very similar as seen in the vision transformer of [13]. To handle the quadratic computational complexity, the authors propose grid self-attention. The image features are separated into smaller grids, to reduce the number of elements in one attention operation. In many successive works, similar ideas with regards to a windowed transformer are used to reduce the computational load [18, 19, 20, 21]. Instead of using the vanilla GAN architecture, a style based architecture similar to StyleGAN [7] is used in [22] and more recently in [20, 21]. More information is elaborated on in Appendix A2.

Hybrid models use the transformer to benefit from the long range interaction property of the attention mechanism to act in the composition of the synthesized image. In the work of Esser, Rombach, and Ommer [23], a vector quantized GAN (VQGAN) made of CNNs is used. The transformer models the composition of an image auto-regressively using a discrete latent codebook.

In another work the GANformer is proposed by Hudson and Zitnick [16], a StyleGAN adaptation with bipartite attention. In bipartite attention, attention is applied between two disjoint sets. The intermediate latent vector w is broken down into m latent components. With the use of bipartite attention, the style information is propagated to the n image features. Each latent component can model long range spatial interactions to guide the synthesis process. An additional benefit of bipartite attention, is that the computational complexity is reduced to a bilinear complexity $O(mn)$. Next to the capabilities of generating faces, the bipartite attention mechanism gives insight into the synthesis of the faces using attention maps. Background information on this subject can be found in Appendix A3.

C. Conditional Face Generation

While face generation is an important step, one needs to have control over the network to generate synthetic identities with variations, while retaining the same identity. This requires a generative model based on a condition, such that semantics like pose and expressions can be changed. None of the transformer based models

provided such control over the synthesis process, taking the TransGAN[1], VQGAN[2] and GANformer [17, 23, 16] into consideration.

In the work of Colbois, Freitas Pereira, and Marcel [9] three approaches to acquiring conditional synthesis are distinguished. One can make a conditional model from scratch or retrain an existing non-conditional model to make it conditional. Another option is to use existing unconditional models to synthesize faces, which can be edited *a posteriori*. The last option that is presented is to exploit an existing model, assuming small changes within its feature or latent space result in changes in the semantics while retaining the same identity.

1) Conditional Conversion

Current models can be adjusted such that these take a conditional input besides their regular input. Architectures may have different implementations. In an early work the conditional GAN was proposed by Mirza and Osindero [24], the model takes a conditional input besides the initial noise vector. Outside of the computer vision field, the authors of Keskar et al. [25] train a language transformer model to condition on control codes that govern style, content, and task-specific behavior. The control code is prepended to the input sequence.

The major downside of this approach is that this requires to adjust and retrain the complete model.

2) A posteriori

In *a posteriori* editing, real or synthesized faces are changed. This approach requires two models, an existing face generation model and an image-2-image model.

Many works feature GANs where the generator has an auto-encoder structure, where the conditional information is inserted into its latent space. Some works are trained to change a selection of attributes, such as hair color, glasses, mustache and age [26, 27, 28]. Not all changes are useful in creating synthetic identities, such as glasses and gender. Other works focus on optimizing only one feature, such as expression or ageing [29, 30].

Rather than injecting conditional information into the latent space, another method uses two encoders for the identity and attribute face respectively and to combine their latents. A model is trained such that the synthesized face has the same identity as the identity face, but taking the attributes such as hair, pose and background from the attribute face [31, 32].

3) Exploiting

Instead of retraining or making additional models, the already trained deep networks can be exploited. Upchurch et al. [33] argue that if a discriminative CNN is able to classify a certain class with a linear classifier, this class must be linearly separable in one of the feature spaces of the CNN. By mapping images with binary classes into the feature space, such as *with beard* and *without beard*, one can determine the attribute vector as the difference between the mean of each class. To gain either attribute while preserving the identity, the reference image can be moved along this vector. The mapping is reversed to obtain the resulting image.

Similar to this idea many works create variations of identities by exploiting the latent spaces in StyleGAN [7, 8]. Such an exploit can be divided into two processes, the projection and the manipulation. The projection, reconstruction or embedding, into a latent space is an optimization process to find a latent vector, such that the synthesized image is similar to the target image. This can be done in both latent spaces Z and W. Other spaces such as $W+$ and StyleSpace are proposed by [34, 35] and [36], it is argued that these feature a higher disentanglement and completeness.

Using the projections, one can analyse the latent space to manipulate generated and real images. Shen et al. [37] show that semantic attributes are linearly separable in W. Using an auxiliary network, synthetic faces are classified on their attributes. Subsequently these faces are used as training data to fit linear support vector machines (SVM) to define hyperplanes. Synthetic and real projected faces can be manipulated by editing the distance to the hyperplane. In the work of Wu, Lischinski, and Shechtman [36] specific channels are changed to detect local and attribute changes. As an example the authors find four separate channels that control the visibility, the shape and the presence of an earring for the ear region. For the attributes an auxiliary classifier is used, the authors argue to only 10 to 30 faces are needed to detect the attribute.

A downside to the above approaches is that the generator generates unlabelled data which must be classified before it has any use. In [9], the authors project the labeled dataset Multi-PIE [38], containing 337 identities under 15 view points, 19 illumination conditions with 6 different facial expressions. Figure 12 shows an example of aligned faces. Besides removing the need of an auxiliary network, the projected identities incorporate a scale that describes the local range of an identity within the latent space.

In this work it is chosen to use the GANformer as a base model, in which the latent space will be exploited as done by [9]. This averts the resource intensive adjusting and training of a new model. Moreover with the idea of creating a synthetic database, the additional information that the attention maps may provide, such as segmentation can yield a multi-purpose database.

III. Attention and Segmentation

Hudson and Zitnick [16] mention that the attention as seen in the attention maps correspond to segmentation in lower layers and finer details in higher layers. This

[1]The authors show the results of interpolating in the latent space, but this only provides indirect control over the output.

[2]The model is capable of several tasks, such as completing images, depth-to-image reconstruction, semantically guided synthesis, pose guided human body generation and class-conditional samples. No fine grained solution to varying an identity is provided here.

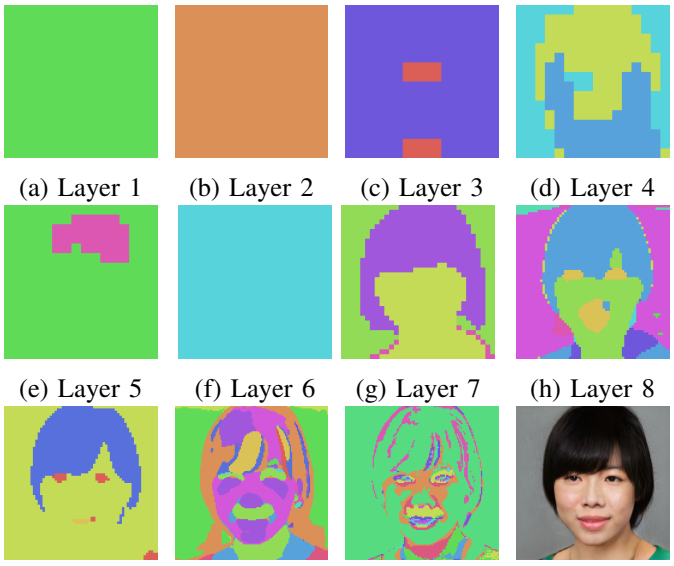


Fig. 2: Examples of the attention maps for one generated face using the default model. The rightmost bottom figure is the resulting face. As seen in layer 1, 2 and 6 only one latent component is represented. Layer 7 - 11 resemble a facial structure. Note that the resolution increases at higher layers.

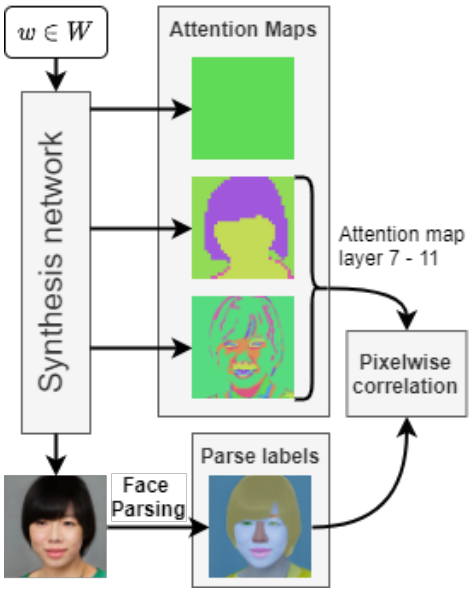


Fig. 3: The approach for inspecting semantic segmentation in attention maps, which is exported in each attention layer. The correlation between the parse labels and each latent component is determined. Only the layers showing facial traits are used.

Fig. 4: Example results of the face parser on generated faces. The left two parses are mostly successful, whereas the right two parses are contain artifacts, likely due to the low quality of the generated face.

is interesting as it may provide segmentation labels next to generated synthetic identities. However in the original work of the GANformer the interaction of the attention in face generation is not shown. This section will focus on answering the first sub-question: *To what extend can we interpret the attention in the synthesis network as a semantic segmentation of the face?*

To answer this question, the segmentation of the attention maps are analysed. Besides that, additional models with varying parameters will be trained from scratch to find out its effect on the attention mechanism. An example of the attention maps is shown in Figure 2.

A. Methods

1) Segmentation

A qualitative analysis will investigate whether segmented facial traits are present within the attention maps. As an objective addition, the correlation [39] is determined between the location of each active latent component and parse labels which are determined by a face parser [40]. Figure 3 shows the approach. It should be noted that the quality of the parses is not optimal, examples are shown in Figure 4. The parser can be unsuccessful in differentiating between the left and right facial features, such as the eyes, eyebrows and ear. Since in early tests it was seen that the attention maps did neither differentiate between the left nor right facial features, the labels for these traits were combined. The labels for glasses, earrings and hat are ignored as well, these are underrepresented and are not semantics of the face. This reduced the number of parse labels to 12.

2) Additional Models

To determine the robustness of the segmentation, additional models are used. By retraining a model, its repeatability on the segmentation is investigated. Moreover models with varying number of latent components and their dimensions are trained, to study the effect on the attention operation. These parameters have a direct effect on the attention computation.

B. Experiments and Results

A pre-trained model is provided in [41], this will be referred to as the default model. It should be noted that the model parameters of this model are different with regard to the model as described in the GANformer paper [16]. Some highlighted differences are clarified in the supplementary material in Appendix B.

First, the default model is analysed with regards to the segmentation. Then the training of additional models is elaborated on. At last the results of the additional models are compared and discussed.

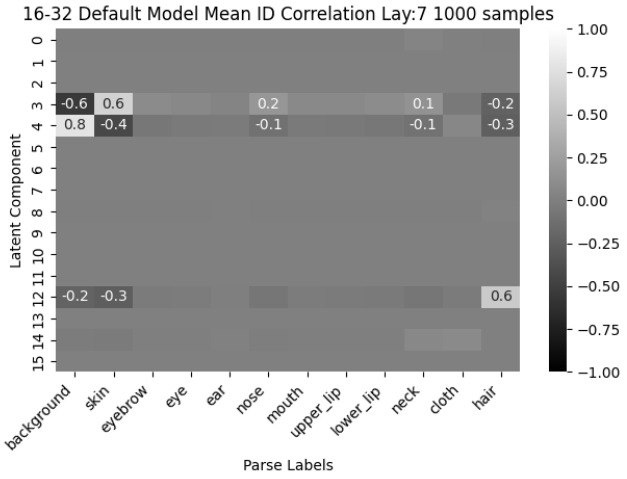


Fig. 5: The correlation between the latent components and the parse labels of layer 7 for the default network based on 1000 generated samples. Only if the correlation $|c| > 0.1$ the correlations are annotated within the figure.

1) Default model

With the default model 1000 faces and their corresponding attention maps were generated. The first 100 were used in the qualitative analysis, of which the first 10 were investigated more closely. For the correlation between the latent components and parse labels the whole set of 1000 samples was used.

The attention maps of layer 1 to 6 do not suggest any segmentation of facial traits, while these are seen from layer 7 onward. Therefore correlation between the active latent components and the parse labels was only determined from layer 7 until layer 11, as shown in Figures 5 and 23. While the other lower layers do contain some information, no clear patterns were seen that demanded further investigation. One phenomena does stand out in the lower layers, as layer 2 and 6 are always 'empty'. The 'empty' attention maps and segmentation of facial traits is discussed hereafter.

a) Empty Attention Maps

In layer 2 and 6, only one latent component is active, though this does not mean nothing happens. The resulting attention is still processed, effectively only one latent component attends to the image features. The latent component acts as a global latent variable, as seen in StyleGAN2.

Only the information of that latent is propagated during the attention as described in Equation (6), as the probabilities of other latent components is reduced to zero. The result of the Softmax term has only one non-zero column. Therefore the last matrix multiplication is effectively an outer product between the probabilities and the active latent component.

b) Segmentation

The attention maps from layer 7 onward can be roughly divided into two groups. In layer 7, 8 and 9, as shown in Figures 2g, 2h and 2i, only a few latent components seem to be active that each attend to segments of the face. The skin, hair, eyes and mouth are noticeable as segments. This is also shown in Figures 5, 23a and 23b, the labels corresponding to the regions tend to have a distinctive latent component.

The nose on the other hand does not have such specific attendance in these layers, but is incorporated with the latent component that attends the skin as well. It may be that the information of both features are combined due to their similar texture, rather than the very distinctive hair and eyes. Another possibility is that their features are described separately within the latent component, as seen with the skin and background in layer 9.

In the highest layers 10 and 11 the attention map seem to focus more on details, in line with the observation of [16]. The attention maps are shown in Figures 2j and 2k, the correlations are displayed in Figures 23c and 23d. It is worth noticing that some attributes are attended by more than one latent component, while some other latent components focus on more than one attribute. It is unclear why some attributes such as the hair and skin are attended to by more than one latent component in layer 10, as the uniform texture and colour of the output do not show such a segmentation.

Overall most attention maps are consistent in terms of the role latent components have in each layer over the generated samples. On the other hand, the role of the latent components seem to change in every layer. The segmented facial traits are not perfect, but note that some correlations have lower scores as the latent components and parse labels are not congruent. For example, latent component 3 in layer 7 attends to the whole facial region and the neck, but as the parse labels are divided into smaller elements including the eyes, mouth and nose, the correlation of each individual pair is rather low. In addition to that, the level of detail in the attention maps is spread. Segmentation can be improved by combining the information of latent components, such as taking the separation of background and skin in layer 7, but using the detail of the hair and eyes from layer 9. This suggests that the latent components of this model can be used to a certain extent with post-processing to acquire some segmented labels.

It should be noted that the attention maps are a simplification, as only the dominant latent component is shown. The probability maps of each latent component in every layer may provide more information, an example is shown in Figure 6. Especially in layer 10 and 11, more latent components seem to attend to the same image features. As a suggestion for future work, it should be investigated whether the added information of these maps are an addition to the segmentation information.

2) Training Additional Models

As the default model suggests an opportunity for segmentation, the additional models were (close) variants of

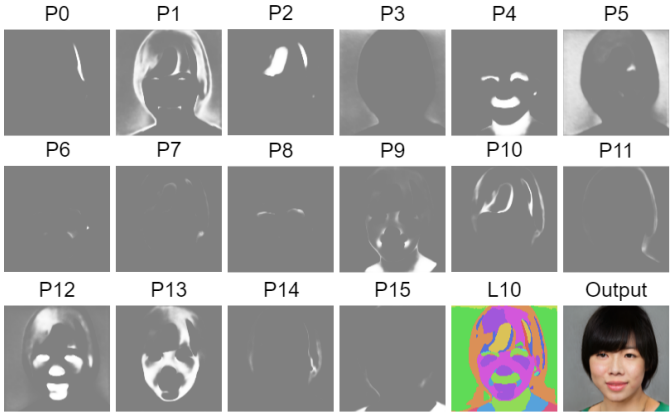


Fig. 6: The probability map for each latent component for attention layer 10. Each Pi indicates the probability map for that layer component.

TABLE II: FID of the retrained models $(16-32)$. Underlined models are used in the additional analysis.

k steps	1427	2515	5008	8005
FID #1	16.7	12.2	8.4	7.1
FID #2	18.5	12.6		
FID #3	18.2	12.9		

this model. The variants will be referred to as $(n-d)$ where n is the number of latent components and d their dimension.

The training algorithm was kept at default as provided in [41], except for the varying model parameters. The models were trained on the cropped and aligned FFHQ database [7] at a resolution of 256x256. The general face quality is determined using Fréchet Inception Distance [42], a common metric which compares distribution between the 50000 samples of training data and the generated data. A lower FID describes a higher quality generation.

For the repeatability of the results three models with the default parameters $(16-32)$ were trained for about 1 GPU week on ∼2500k steps. To estimate the amount of steps the default model was trained for, model 1 was trained further to 8005k steps. The default model had a FID of 7.35, the extended retrained model reached this number between 7000k and 7200k steps. Their FIDs are shown in Table II, the training progress is shown in Figure 22.

To study the effect of varying model parameters, eight variants were trained as well on about 2500k steps. The resulting FIDs for the trained models are shown in Table III, the training progress is shown in Figure 21. The dimension of the latent components seems to have the most significant influence on the quality of the generated faces, a higher dimension seems to be beneficial.

3) Segmentation of Additional Models

First the repeatability is discussed with the three $(16-32)$ models and the default model. Next for the varying parameters, the first $(16-32)$ model and four

TABLE III: FID of each trained model variant at about 2500k steps with the respective number of latent components and their dimension. The models with underlined annotation are chosen for the additional attention map analysis. For the $(16-32)$ models the mean is displayed.

FID	Latent Dimension			
	32	16	12	8
# Latent Components 32	/	14.9	20.3	28.1
# Latent Components 16	12.6	15.9	19.5	/
# Latent Components 12	12.5	16.0	/	/
# Latent Components 8	12.4	/	/	/

variants were chosen for further analysis: $(12-32)$, $(8-32)$, $(16-16)$ and $(16-12)$. This selection provides a wide range of latent components and their dimensions. Note that the correlation is based on 500 samples, a reduction to cope with the computational load of generating attention maps. Figure 24 shows all attention maps for one generated face for each evaluated model.

a) Repeatability

The correlations for model 1, 2, and 3 are shown in Figures 25, 26 and 27 respectively. The attention shows a similar behaviour, latent components attend in a holistic manner onto the image features. Therefore only a small number latent components are active and the information provided for semantic segmentation is rather low. The probability maps don't provide extra data for this case.

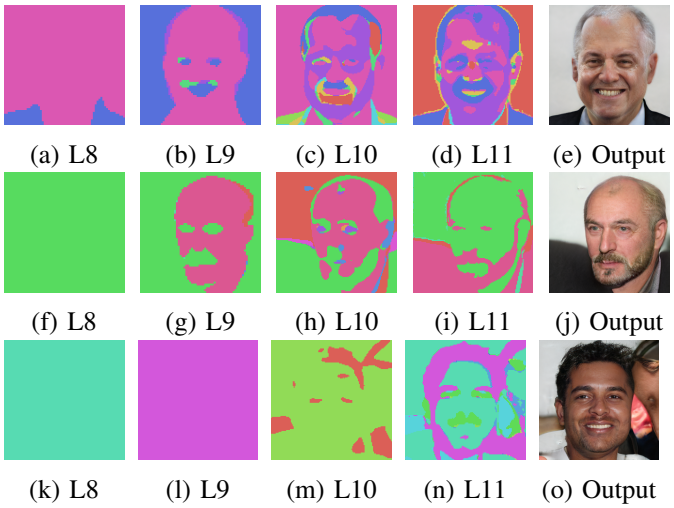
The correlations for model 1 trained for 1427k, 5008k and 8005k steps are shown in Figures 28, 29 and 30 respectively. For this model longer training results in more 'empty' maps and a detailed last layer. Like the 2500k model, the attention does not provide useful segmentation information. The attention in the 2500k and 8005k models are quite similar, which may suggest that one does not need a fully trained model to evaluate whether the model will be useful with regards to semantic segmentation.

It is clear that the trained models do not converge to the default model, the attention behaviour is significantly different. It might be that the default model is trained for deviating parameters, since the three trained models are highly similar. With this assumption, the process is repeatable. This may suggest that with knowing the training parameters of the default model, its highly informative attention can be acquired in a robust manner.

b) Varying Model Parameters

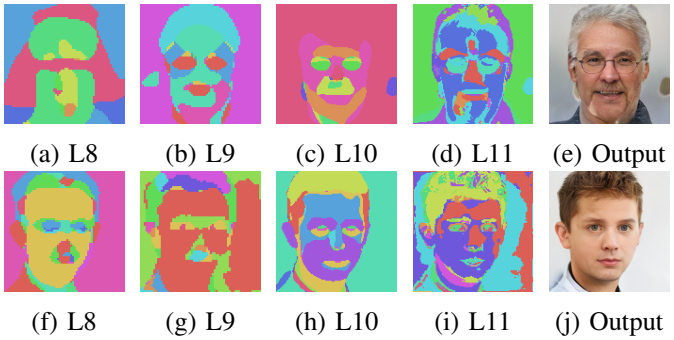
The attention maps and the correlations are shown in respectively Figures 7 and 8 and Figures 25 and 31 to 34. Note that for $(16-32)$ only model #1 is taken into account.

There is a clear distinction between models with $d=32$, versus a lower dimension of $d=[12,16]$. In the high dimensional models only a small number of latent components is active throughout the layers, attending in a holistic manner on the face. On the contrary the attention maps of $d=[12,16]$ seem to have more active



(a) L8 (b) L9 (c) L10 (d) L11 (e) Output
(f) L8 (g) L9 (h) L10 (i) L11 (j) Output
(k) L8 (l) L9 (m) L10 (n) L11 (o) Output

Fig. 7: Examples of the highest attention maps for one generated face per model, where the caption denotes the layer number or resulting face. The first row corresponds to model $8-32$, the second to model $12-32$ and the third to $16-32$.



(a) L8 (b) L9 (c) L10 (d) L11 (e) Output
(f) L8 (g) L9 (h) L10 (i) L11 (j) Output

Fig. 8: Examples of the highest attention maps for one generated face per model, where the caption denotes the layer number or resulting face. The first row corresponds to model $16-12$, the second to model $16-16$.

latent components in each layer, which provide a higher level of semantic segmentation.

It seems that the focus during training lies on maximizing the information into a minimum number of latent components. The many active components in the $d = [12, 16]$ models are mandatory, as the same level of detail of the $d = 32$ models has to be described within more latent components, due to a smaller embedding dimension. The results of the training in Table III supports this, as a smaller dimension seems to negatively impact the generation quality in terms of FID.

C. Conclusion on Attention Maps

The attention mechanism does not provide a direct means for segmentation, the attention layers are optimized to fool the discriminator, not to function as a segmentation tool. Nevertheless the default model does show segmented facial traits within several layers, especially the background, the skin and hair have high correlations. A suggestion for future research is to make smarter use of this by combining the information in these layers. The the probability maps may also be incorporated for layers with many active latent components, to maximize the information input.

A major downside is the fact that there is no guarantee that a model's attention maps will provide useful information for segmentation, all trained models do not reach the level of semantic segmentation as the default model.

With the same training parameters, the attention's behaviour seems to be robust to retraining. It is shown that by reducing the latent dimension, the amount of semantic segmentation can be increased. Although this does come with a lower general face quality as a trade-off.

Instead a better recommendation is to find out what distinguishes the default from the other models, as the default model does provide useful semantic segmentation information. The results suggest that one does not need a fully trained model to conclude this.

IV. Semantic Control

For synthetic face databases, generating identities is a necessary step for e.g. the use of training face recognition systems and 3D face reconstruction models. This requires control to augment the generated faces in order to get the variations. The GANformer is unconditioned, meaning that this is initially not possible. In this section the method as presented by [9] is used. An overview of the process is shown in Figure 9. The process for synthetic identity generation, as used in this work, consists out of three parts:

1) Expression faces of the Multi-PIE dataset [38] are projected into the latent space to acquire labelled latent vectors.
2) Using the labeled data for the each neutral-expression pair, the corresponding latent directions are computed.
3) Reference faces are created using random generated w latent variables, the faces are neutralized with regard of their expression. Augmented faces are determined by moving the reference face in the direction of the computed latent directions.

Since a projector has not been implemented on the GANformer, research is done to answer the sub-question: *To what extend can we reconstruct existing identities from the latent space of the GANformer?* Using this data the third sub-question will be answered: *To what extend can we control the smile expression while maintaining the same identity?*

The codebase of the implementation in StyleGAN2 can be found at [43]. It is adjusted where needed for compatibility with the GANformer.

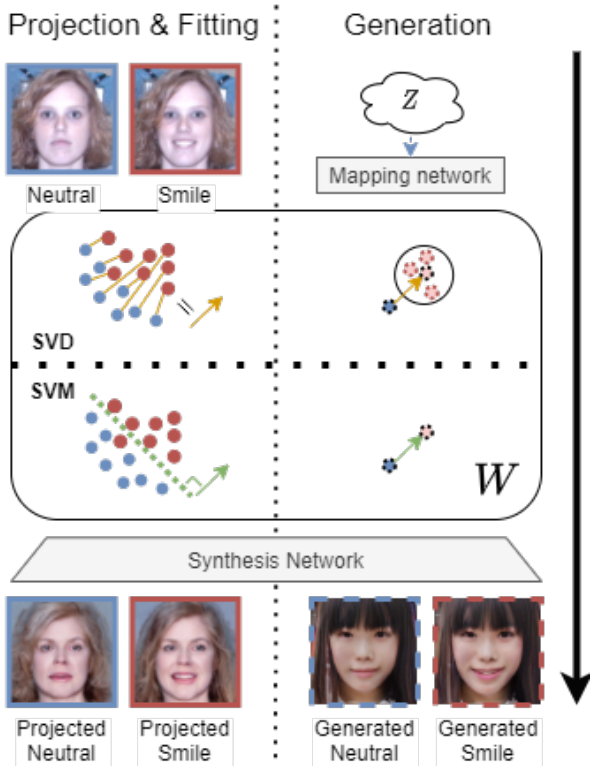


Fig. 9: The process to gain control over the latent space. Real faces are projected into the latent space, using either SVD or SVM the latent direction can be determined to augment sampled faces. Inspired by [9].

(a) Mean face (b) Step 30 (c) Step 60 (d) Target

Fig. 10: Typical projection of a generated face using the default projector provided by StyleGAN2, showing the results at step 0, 30, 60 and the target face. Left: the synthesis of the latent variable acquired by taking the mean over 10000 mapped vectors.

A. Methods

1) Projection

The GANformer is not provided with a projector like the one implemented in StyleGAN2 [8]. However since the GANformer is also a style-based architecture, the StyleGAN2 projector was used as a base. More information about the projector can be found in Appendix A2a.

To asses the performance of the projector, it was first applied on generated faces. After this real faces from the Multi-PIE dataset [38] were projected in order to find the latent directions of the attributes.

a) Projection of Generated Faces

The default projector was most of the times unable to find a suitable projection and even a face, a typical result is shown in Figure 10. Nevertheless the low resulting losses pointed out that there are many local minimums, while the projected latent variables w where not close to the latent variables of the generated faces.

Subspace: It appears that $W_{face} \subset W$ such that W_{face} describes faces, but the projector is unable to retain the faces within this subspace. On the other hand, the mapping network is able to map random vectors $z \rightarrow w \in W_{face}$. Generation from W, by drawing standard normal sample vectors w, results in non facial images, as shown in Figures 37 and 38.

In order to analyse the subspace and present a proper solution, the distribution of the subspace was approximated by two methods. For both methods a set of 10000 mappings w_m was used, note that this is the same number as used in the projector for the initial latent. The first method estimates the mean μ_m and standard deviation σ_m of the distribution of the latent variables w_m. Samples were drawn with $w \sim N(\mu_m, \sigma_m)$. The second method approximated the distribution using singular value decomposition (SVD) as shown in Equation (1). Samples are drawn from a standard normal distribution and scaled and transformed using S and U acquired from the SVD. At last the vectors are translated with mean μ_m and sample latent vectors w are acquired.

The synthesized results are shown in Figures 40 and 41 for the normal and SVD method respectively. Based on a cosine dissimilarity on the latents, distances between the approximated samples and the mapped samples are similar. Even though the synthesized approximations do result in face like images, the quality is inferior to the mapped synthesized samples. Many approximated samples contain artifacts and random patterns, a similar effect as seen in the mean face as shown in Figure 10a. This concludes that there is a subspace W_{face}, however it is shown that it can not be fully described by one of the two approximation methods.

$$\begin{aligned} U, S, V &= \text{SVD}\left(\frac{X - \mu_m}{\sqrt{dim(X) - 1}}\right) \\ L &\sim N(0, 1) \\ w &= USL + \mu_m \end{aligned} \tag{1}$$

Mahanalobis Distance: The LPIPS loss of the projector only retains the overall colour of the synthesized image and is therefore unable to retain w_p within W_{face}. Since W_{face} can be roughly approximated, the Mahanalobis distance was added to regularize the projected latent w_p. Note that the Mahanalobis distance is only an extra measure and cannot be applied solely, as it does not compare w_p to w_{target}, since the latter is unknown when projecting non-generated faces.

The Mahanalobis distance was implemented such that it determined the distance of the projected vector w_p to the distribution of the subspace, based on μ_m and its covariance as shown in Equation (2). The inverse

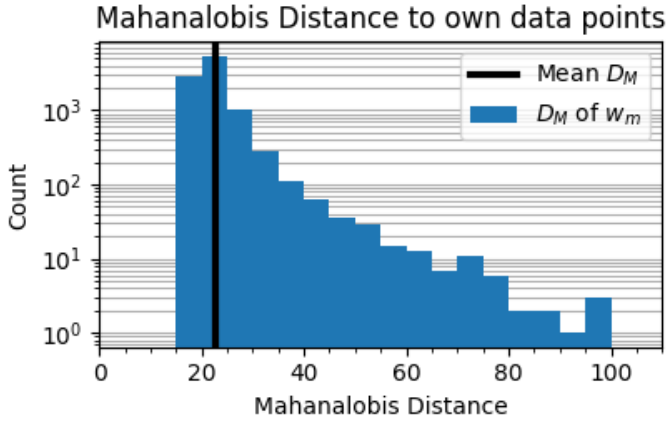


Fig. 11: The Mahanalobis distance of based on 10000 w_m variables, between μ_m and each of the 10000 w_m variables. The distances are between [16.3, 98.5].

covariance matrix S_m^{-1} will scale the loss inversely proportional to the scale of the approximated distribution.

$$D_M(w_p) = \sqrt{(w_p - \mu_m)^T S_m^{-1} (w_p - \mu_m)} \quad (2)$$

The Mahanalobis distance can be applied on the same mapped latents on which μ_m and S_m^{-1} are based on. In Figure 11 this distribution of distances is shown, with a mean D_M of 22.6. More interesting is that $min(D_M) = 16.3$, suggesting that $\mu_m \notin W_{face}$. This may explain that the mean image, as shown in Figure 10a, has similar artifacts in the synthesized faces as the approximated generations, which are based on the mean of the mapped distribution as well. The weight for the Mahanalobis loss should not be too high.

Mahanalobis Loss: Both the initial latent vector and the Mahanalobis distance are based on μ_m. Therefore $D_M = 0|_{t=0}$ and a weight a should be added, such that $a = 0|_{t=0}$, to prevent keeping the projection at μ_m.

Several loss functions were implemented, shown in Equation (3), these included a linear increasing loss L_{Mlin}, a quadratic increasing L_{Mq-inc} and a quadratic increasing and decreasing curve L_{Mq-cur}. Each loss consisted out of a scaling factor a_w and a time dependent factor, which is 0 at time step $t = 0$ and 1 at time step $t = T$. L_{Mlin} was included with a offset b. The parameters were optimized by minimizing the cosine distance between the generated latents and their projected latents.

$$\begin{aligned} L_{Mlin} &= a * D_M + b \\ &\text{where } a = a_w * \frac{t}{T} \\ L_{Mq-inc} &= a * D_M \\ &\text{where } a = 4 * a_w * \frac{t}{T}^2 + 4 * a_w * \frac{t}{T} \\ L_{Mq-cur} &= a * D_M \\ &\text{where } a = -4 * a_w * \frac{t}{T}^2 + 4 * a_w * \frac{t}{T} \end{aligned} \quad (3)$$

b) Projection of Real Faces

With a working projector real faces can be projected that serve as training data for the latent directions. In an ideal case, the projected faces are identical to the target faces and a perfect mapping is made into the latent space. However it is likely that the latent space is not complete. Important is that the difference between the projected neutral and attribute faces describe only that certain attribute, to acquire disentangled latent directions. Therefore the projected faces within an identity must have a similar identity.

The number of optimization steps was reduced to 500, as 1000 did not show a major improvement. Before the subjects of the PIE-dataset were projected, their faces were cropped and aligned using the same tool are applied on the FFHQ dataset.

The quality of the projected faces are validated using FaceVACS 9.6, where $0.5 = 0.1\%$ FAR. The score between aligned and projected pairs, describe the performance of the projector. In addition to that it illustrates the GANformer's latent space completeness in making all types of varying faces. Whereas the scores between faces in an identity describe the consistency of the projected faces and their value as training data for the latent directions.

For benchmark purposes the method is applied on StyleGAN2 as well, like the GANformer the number optimization steps for the projector was set at 500. Note that the fully trained model of StyleGAN2 is used, which is trained between 3x to 14x longer as stated in Appendix B.

2) Latent Directions and Exploration

The latent of the projected faces can be used to explore the latent space and as a training set to determine the latent directions. For the exploration two methods are applied, t-distributed stochastic neighbor embedding (t-SNE) and SVD. For the latent directions, the SVD is used as well as the method from [9], training a linear SVM.

t-SNE is an unsupervised statistical method to visualize high dimensional data in a low dimensional space [44]. It can provide a view on how the projected latents are related to one another. Note that due to the implementation, local pairwise distances are preserved. More information about t-SNE can be found in Appendix C.

As mentioned earlier the latent space of the GANformer is hard to describe, it may be that the classes are not linearly separable. Using SVD, the latent space can be explored and a linear combination may be able to describe the change between neutral and the specific expression. The first step all pairwise vectors are determined, that is $w_{i,neutral} - w_{i,expression}$. Then SVD is applied on these vectors, as shown in Equation (1), where X are the vectors and μ_m the mean of those vectors. By sampling L from a standard normal distribution, multiple direction vectors are generated. These can be added to the reference faces to find a

suitable expression variation. As additional parameters the number of singular components can be adjusted to acquire a more or less specific movement and the resulting sample vectors of USL can be scaled with a scalar. The multiple samples show variations around the expression acquired by moving the neutral latent with the mean.

In the SVM method, to goal is to fit a linear SVM as a hyperplane between the neutral and the expression class. The normal onto this hyperplane describes the direction between the neutral and expression. Among other statistics the mean of the distance between both classified populations to the hyperplane is determined. This distance reflects the range of the Multi-PIE dataset, which provides a scale as in how much variation can be applied while preserving the identity. While statistics can describe the performance of the fit, whether the normal on to the learned hyperplane makes any sense, must be concluded through the generation process.

3) Generation of Synthetic Identities

The generation of synthetic identities is done in two subsequent steps, generate reference faces and adding augmentations. The reference faces are sampled by generating faces from Z, which return its mapped w vector and the face itself. To ensure neutral and frontal reference faces, the reference latent is first neutralized.

In [9] the authors note that StyleGAN2 mainly generates smiling and neutral faces. Therefore the neutralization is done by moving the sampled latent towards the neutral direction along the neutral smile vector.[3]

To prevent similar faces when creating a synthetic database, a *interclass threshold* is applied. This threshold is based on the squared Euclidean distance between the embeddings of the generated face and all other generated faces. The embedding is determined by a pre-trained Inception-Resnet v2 model trained on MSCeleb [45].

The second part consists out of generating the augmented identities. The SVD method samples direction vectors while varying the number of singular components as mentioned before. For the SVM method each attribute face is determined by adding the mean of the latent direction, normal to the found hyperplane.

B. Experiments and Results

The default model as referred to Section III-B is used for this section as well.

1) Projection of Generated Faces

The additional Mahanalobis loss in the projector retains the projections within the W_{face}, three variations of loss functions are described in Equation (3). With various preliminary tests the optimal range of the parameter a_w for each loss function was decided. With a too high a_w, the Mahanalobis loss was too strict and the resulting projection had loss of detail and similar artifacts as shown in Figure 10a, as the projection was pulled towards the mean of the W_{face}. A too low a_w resulted in a higher chance to the projection to leave W_{face}, similar as shown in Figure 10c. The effect of offset b was not significant for low values between 0 and 10, however for higher values it performed better on the outliers. Every loss function had a similar performance without significant differences, as shown in Table IV. Noticeable differences between the projections and the generated faces were the direction of the eyes and the teeth structure. The identity, pose, illumination and expression were generally the same. For the subsequent sections the linear loss function with $a_w = 0.1$ and $b = 5$ was used.

TABLE IV: The results based on the cosine distance between the target and projected latents for each loss function.

	Quad Curve							
a_w	0.005	0.01	0.05	0.1	0.5	1	5	10
Mean	0.09	0.17	0.11	0.11	0.08	0.12	0.15	0.16
Median	0.03	0.09	0.04	0.05	0.04	0.05	0.15	0.15
Max	0.37	0.45	0.36	0.25	0.31	0.51	0.29	0.34
Min	0.02	0.02	0.02	0.02	0.02	0.02	0.05	0.06
	Quad Increase							
a_w	0.001	0.005	0.01	0.05	0.1	0.5	1	5	10
Mean	0.27	0.14	0.11	0.09	0.16	0.10	0.12	0.18	0.18
Median	0.25	0.03	0.04	0.03	0.05	0.05	0.06	0.18	0.18
Max	0.58	0.43	0.41	0.41	0.53	0.32	0.30	0.40	0.31
Min	0.02	0.02	0.02	0.02	0.02	0.02	0.03	0.07	0.07
	Linear								
a_w	0.1	0.1	0.1	1	1	1	10	10	10
b	0	10	50	0	10	50	0	10	50
Mean	0.19	0.13	0.14	0.15	0.09	0.11	0.13	0.14	0.11
Median	0.09	0.03	0.08	0.10	0.04	0.04	0.14	0.10	0.06
Max	0.62	0.52	0.34	0.48	0.28	0.40	0.25	0.30	0.42
Min	0.02	0.02	0.01	0.02	0.02	0.02	0.05	0.05	0.03

2) Projection of Multi-PIE Faces

The expression dataset was projected, which was done in well over 5 days, while only in 21 hours using StyleGAN2.[4]

The projection of the expression dataset resulted in 1095 alignment projection pairs with 208 identities ranging from 2 to 8 expressions divided over 1 to 4 sessions. There were 2964 unique pairs within the identities themselves. The aligned identities may have another appearance over the sessions, such as different hairstyle or clothing.

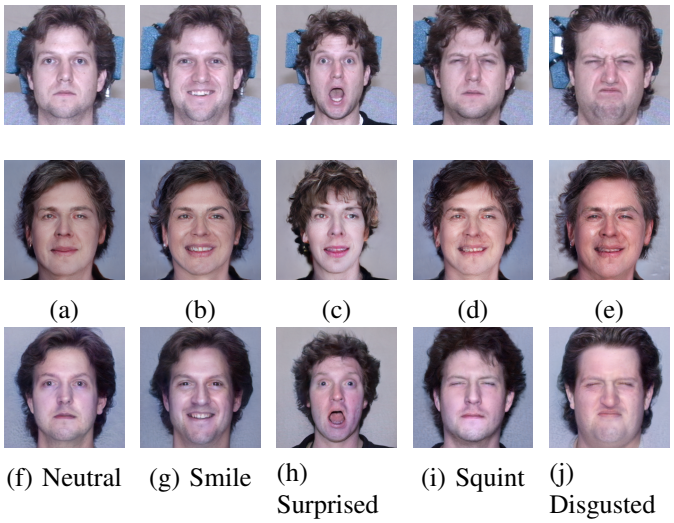
a) Alignment Projection Pairs

The FaceVACS scores of the alignment and projected pairs are shown in Figure 13a. The the pairs are regarded as non-mated, Figure 44 shows a selection of the best projected pairs. A projected identity is shown in Figure 12.

Only the smile expression seems to be successfully projected in most cases. The model seems to be unable

[3]The pose and illumination are neutralized as well in [9].

[4]Using 8 Intel Xeon Silver 4216 cores and one NVIDIA RTX6000 GPU. The pose and illumination would have taken 22 and 60 days respectively. Results of the pose and illumination projections are shown in Figures 42 and 43 as an indication of the network projection performance. Note that the pose and illumination can also be acquired via other methods such as 3D face modeling.



(a) (b) (c) (d) (e)

(f) Neutral (g) Smile (h) Surprised (i) Squint (j) Disgusted

Fig. 12: Example of one aligned (upper) and projected (GANformer middle, StyleGAN2 bottom) identity with a neutral face and various expressions.

to project open mouths and subtle details as found in other expressions. The FFHQ dataset is likely to consists mostly out of neutral and smiling faces, rather that screaming and disgusted faces. Besides the expression there are some other observations that shows the low variability of the latent space; in some cases genders are swapped, the model seems to have difficulties with projecting faces with a dark skin tones and some projected identities seem similar while the aligned faces are not. Examples are shown in Figures 45 and 46.

While the projection of generated faces was very successful, the projection of the Multi-PIE expression faces under performs. The projection of generated faces is a more trivial task as the optimal solution does exist somewhere in the latent space. Nevertheless the GANformer has an understanding of a holistic face. Note that the expression of smile is quite interpretable, while expressions such as surprised and disgusted are less clear. The next sections will look specifically at the smile expression. As the latents of the projected faces are the training data for the latent directions and the majority of the expressions but smile is not correctly projected.

b) Within Projected Identities

For the latent direction to be disentangled, the neutral expression pairs need to be similar in identity. Note that this is not a metric to asses the quality of the expression itself, in some cases an expression face looks very neutral and results in a high FaceVACS score. Figure 13b shows the FaceVACS scores for each neutral - expression pair. The projected faces shows a left-skewed distribution with a mean of 0.766. It is noticed that the variations of the subjects of the sessions such as hairstyle can have a major impact on the resulting face. Note that the aligned expression faces have a high similarity with a mean of 0.990, despite the variation in expressions.

The goal is to find identities with only some changing attributes. As a suggestion for future work, after the first neutral face is projected, the subsequent projections of that identity can start at this neutral face latent, rather than the distant mean face. This will likely reduce the excessive computational load and a higher consistency among the generated faces within an identity.

c) StyleGAN2

The projections of StyleGAN2 seem to be much more similar to the aligned faces, as shown in Figures 12 and 13a. The model is able to project significantly more details of the identity, among others: facial structure, wrinkles, expression including scream, surprised and disgusted and hairstyle. In a few cases the glasses or earrings are removed during the projection. Besides this the projected faces are sometimes washed out, nevertheless the majority is considered as non-mated.

The score distribution of the projections within the identities is similar that of GANformer. Although the majority of the projections are in the range of $[0.8, 0.9]$ rather than $[0.9, 1.0]$. Although with the GANformer some expressions were incorrectly projected and resulted in similar faces, the expressions are projected truthfully in the case of StyleGAN2. It is interesting that for these high quality projections the are not regarded as more similar, even though each projected sample is a result of an independent optimization process. Further research is needed whether this is due to the synthetic nature of the projections.

3) Latent Space Analysis and Generating Synthetic Identities

In this section the results of t-SNE, SVD and SVM are shown and analysed to explore the latent space and create smile augmented faces.

a) t-SNE

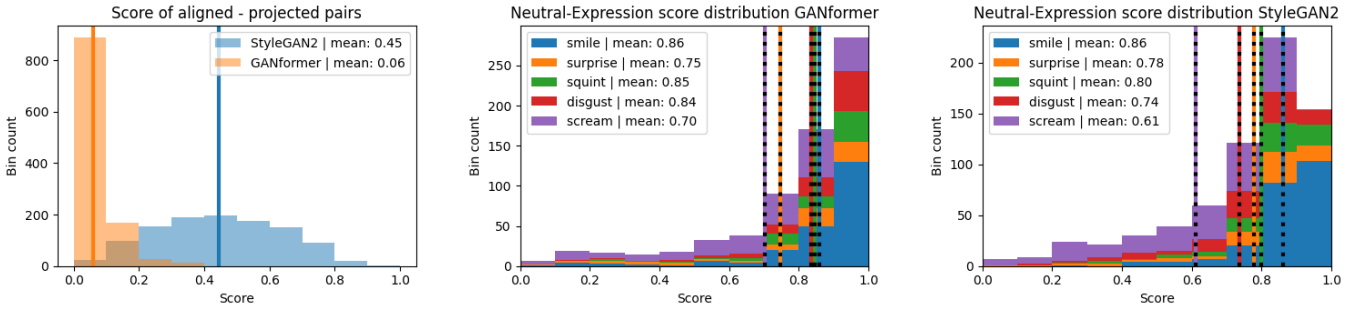
A t-SNE analysis is carried out on unfiltered and filtered projected latents of the GANformer as well unfiltered projected latents of StyleGAN2, shown respectively in Figures 47, 48 and 49. The filtered results were filtered on a minimal FaceVACS scores within the projections of an identity.

The unfiltered results are distorted, even for identities that are clustered in the filtered results of the GANformer. As there is no low dimensional structure present in the unfiltered data, it is unlikely that the latent directions can be acquired. Filtering the training data is thus required.

The filtered results of the GANformer are clustered in identities, like the t-SNE of the StyleGAN2 latents. Note that some identities have a more distorted mapping in the filtered GANformer results, which suggests that the used filter is not optimal for removing distant latents.

b) SVD

The SVD method was used on the GANformer to analyse the latent space around the face, moved with the mean neutral smile vector. The neutral smile pairs with various thresholds on their FaceVACS score were used



(a) (b) (c)

Fig. 13: (a): The FaceVACS score distribution of the aligned - projected pairs using the GANformer and StyleGAN2. (b): The FaceVACS score distribution of the neutral - expression pairs using the GANformer. (c): The FaceVACS score distribution of the neutral - expression pairs using StyleGAN2. Zoom in for better visibility.

to determine the relevant SVD parameters. In all cases the mean vector did not seem to transform the reference face to a smiling face, neither was adding singular components showing clear changes towards smiling. Instead, by adding additional singular components the face decayed quickly, as shown in Figure 14.

For StyleGAN2 the unfiltered mean vector was used. In contrast to the GANformer, the mean neutral smile vector did seem to represent the smile direction. The added singular components augmented attributes such as hair style and skin colour, as shown in Figure 14. This variation is likely due to the orthogonal property of the SVD. The latent space of StyleGAN is linearly separable, therefore any other direction induced by the multiple neutral-smile vectors acts on other linearly separable attributes.

The contrasting results between the models support the fact that the latent space of StyleGAN2 is better described. This is something that is seen throughout this method. As the GANformer needs an additional loss to retain the projector within W_{face}. In addition to that the lack of completeness is shown with the fact that all but smile are not projected correctly. On the other hand the space between faces is described as shown in the interpolations in Z and W from mapped z samples in Figures 38 and 39. It is expected that with longer training the latent space will be more fully described.

c) SVM

In a linearly separable latent space, fitting a linear SVM between the projected neutral and smile latents, the normal onto the hyperplane describes the neutral - smile latent direction. The training data is acquired without supervision and therefore the labeled neutral and smile may not be actually true to their label. While the unfiltered set resulted in an accuracy of 99.6%, due to the poor quality the latent direction was not representing anything.

Finding a filter which resulted in a proper latent direction was not a trivial task, as the classes needed to be balanced and the FaceVACS score does not imply correct labels. A proper filter was implemented by removing latents based on their FaceVACS entry if their score was $s < 0.9$. A balanced neutral - smile ratio was provided as shown in Table V.

TABLE V: Confusion matrix of linear SVM fit on GANformer projected neutral and smile latents.

neutral-smile 156		Predicted Neutral	Predicted Smile		
True value	Neutral	82	5	Accuracy	0.92
	Smile	7	62	True neutral rate	0.94
				True expression rate	0.90

The latent direction did seem to be oriented in the neutral smile direction, causing smiles on neutral faces, Figure 15 shows an example. In general the identities are kept, but the neutralisation was not always successful. In addition to this, latent direction is not completely disentangled, the pose, hair color, eye direction, appearance of glasses and teeth structure changes over the augmentations. It should be noted that in some instances, the latent direction did not change the expression at all.

The linear SVM on unfiltered projected StyleGAN2 latents was fitted perfectly and resulted in a representative latent direction, as shown in Figure 15. In contrast to the GANformer, the neutralisation did work on all samples. Like the GANformer, the latent direction was not completely free of entanglement and male faces got more feminine towards the smiling direction. It is noted that the reference faces are closer to the smile than the neutral face in the SVD method, as in SVM mean distance from hyperplane is taken, rather that the distance between the two populations in the SVD method.

Figure 16 show the FaceVACS results between the neutral and scaled smile faces for both models. While the GANformer shows a higher similarity, the actual variety in smile is smaller than in StyleGAN2. A higher scale is needed for the same variation in smile. This small augmentation is likely due to the scale acquired during the training of the latent direction. Some projected neutral faces have slight smiles and while some projected smile faces are very similar to the neutral

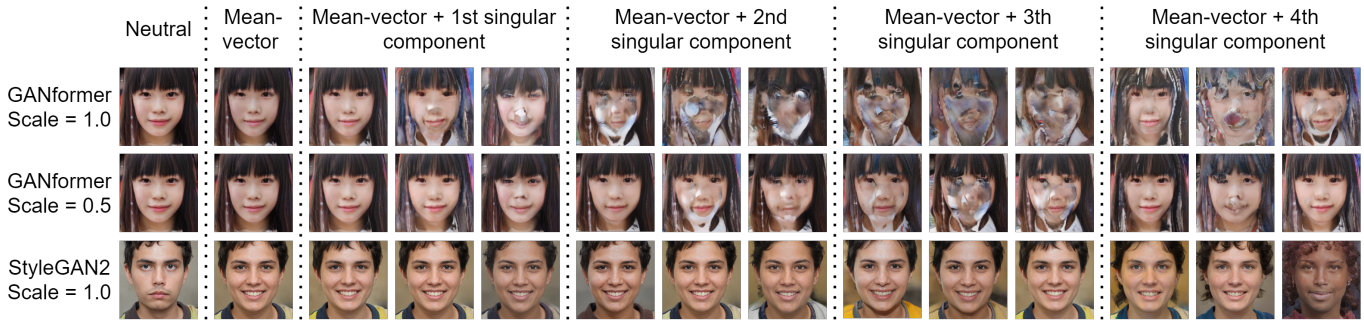


Fig. 14: The results of the SVD generation on one identity for both models. The upper two rows use the GANformer, while the bottom row uses StyleGAN2. The augmentations are done on the reference face using the mean vector and additional sampled singular components. In the middle row the vector USL is scaled with 0.5, while the two other rows have a scaling of 1.0.

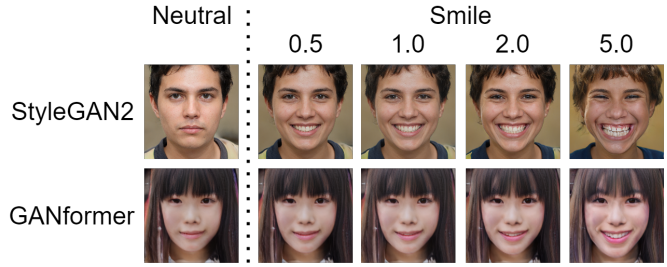
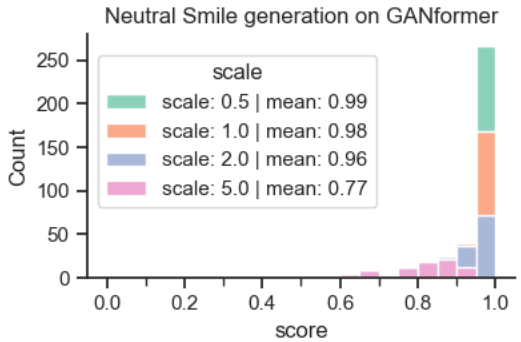


Fig. 15: The results of the SVM generation on one identity for both models. The first column shows the reference face, whereas the resulting columns show the augmentation with an increasing scale.

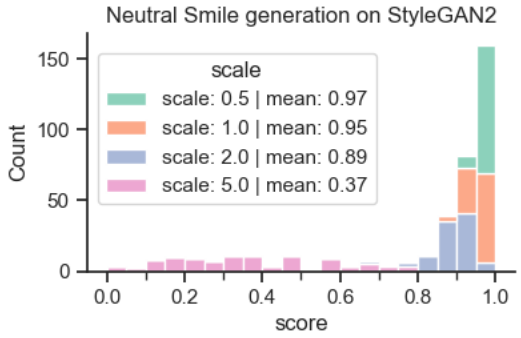
faces. Therefore the populations are closer to each other and the inherent scale too small. While the filter filters deviating identities to encourage disentangled data, this does not guarantee a well defined neutral and smile pair in terms of expression. In fact, it may even encourage pairs that are too similar.

This also explains the fact that the neutralisation of the GANformer does not work as well as of StyleGAN2, as a higher scale is needed as well. Generations with a higher neutralisation scales are shown in Figures 17 and 50.

The neutral smile latent direction of the GANformer is not completely disentangled. One cause can be the that the latent space has a limited separability and requires more training. In addition to that, it may be that the found latent direction is not optimal. As noted finding a latent direction is not trivial, it is very sensitive to the selection neutral smile latents. Another factor that may influence the entanglement is the use of attention in the architecture of the GANformer. As seen in Figures 5 and 23 the eyes and mouth do have overlapping latent components, interaction between these two attributes is possible, such as the changes in eye direction. On the contrary, the hair tends to change in some identities as well, while the latent components attending to the hair is negatively correlated with those of skin, eyes, eyebrows and mouth. The fact that this only happens to a small set



(a)



(b)

Fig. 16: (a): The FaceVACS score distribution of the generated neutral smile pairs per scale using the GANformer. (b): The FaceVACS score distribution of the generated neutral smile pairs per scale using the StyleGAN2. Zoom in for better visibility.

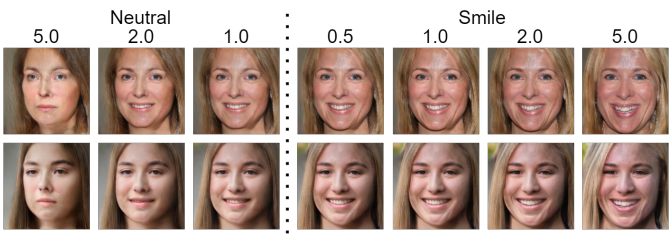


Fig. 17: Effect of scaling the the distance for both neutralisation and the smiling attribute on two generated identities using the GANformer.

of samples makes it less likely that this issue is present due to the architecture, but rather an entangled latent space.

The fact that a very specific set is needed is a downside of the projection method. This method needs to provide reliable results, such that the training set for the SVM consists out of representative data. As a recommendation for future research to improve this method, a system that classifies the magnitude of the smile and neutral expressions should be added. This can advance this work in two directions. First, the current training set is based on similar identities, but this still allows mislabeled faces. The classifier may serve as an extra quality control for a better latent direction. Second, the current evaluation is done qualitatively and with the use of FaceVACS. As shown some generated smile and neutral faces are mislabeled. With the classifier the expressions of the generated faces and the effect of the scaling can be quantized, which leads in turn to a better comparison between other models such as StyleGAN2.

To improve the separability of the latent space, the model should be trained for more steps. This will also improve the representation of under represented expressions in the latent space, such as scream and disgust.

C. Conclusion on Semantic Control

The first goal of this section was to evaluate whether existing identities can be reconstructed from the latent space of the GANformer. The projector on the current model needs a regularization term, to retain the projected latent in the subspace of faces, W_{face}. The FaceVACS verification shows that none of the existing projected pairs have the same identity. The qualitative results show that only holistic attributes of the existing identities are projected similarly. While the neutral and smile expressions of existing identities are also found in the projected faces, other more ambiguous and detailed expressions are not projected correctly. It is argued that these other expressions are under represented in the training data and therefore not represented in the latent space. As a suggestion for future work, more training may lift this limitation. Even though the quality of the expressions vary, the FaceVACS score suggests that the projected faces of one existing identity do have the same identity, a requirement for disentangled labeled data.

The second objective was to investigate to what extent the smile expression can be controlled while maintaining the same identity. It is shown that projected latents are noisy and need to be filtered. While the SVD method works on StyleGAN2, neither the mean vector and added singular components can construct smiling faces in the GANformer. The variations by adding the orthogonal singular components emphasizes the linear separability and high descriptiveness of the StyleGAN2 latent space, while the lack of in the used GANformer model.

The SVM method is, with a specific subset of projected latents, able to find a latent direction that controls the neutral smile attributes. The scale is inherently small due to the training set. Therefore a larger scale is needed to find similar degree of change in expression as found in StyleGAN2. The found latent direction is not completely disentangled. The latent direction is likely not optimal, in addition to that the latent space is only linearly separable to a certain extend. In some cases the latent direction does not control the smile expression.

Two suggestions are provided. A classifier should be used to classify the magnitude of the projected and generated expressions, this improves the filtering and makes the evaluation and comparison more explicit. In addition to that, it is argued that the shortcomings concerning the various expressions and entanglement are mostly due to the lack of training. With further training the latent space will likely head towards the descriptiveness and disentanglement of the fully trained latent space of StyleGAN2.

V. Conclusion

In this work the GANformer model is explored for two uses in creating synthetic face databases, without the need of training the model for the specific applications, but general face synthesis instead.

First the attention is investigated on the use of segmentation. This unique property makes this model beneficial to use over well known models such as StyleGAN2. The results show segmenting behaviour, though more work is needed to put this into practice. Suggestions for future work are to use multiple layers and probability maps and to fine tuning model and training parameters.

With augmentation multiple faces of the same identity can be created, this requires generation based on a condition. Using the reconstruction of faces to find latent directions, it is seen that some control over the neutral smile direction is gained. It is suggested to include a expression classifier to improve and evaluate the process. In addition to that the model should be trained longer for a higher descriptiveness to include other expression and to achieve a higher level of separability.

In both the segmentation and the smile augmentation cases compelling results are shown and indicate the possibilities to use the GANformer for multiple applications in synthetic face database generation. However further work is needed in both objectives to acquire working solutions and to create synthetic face databases.

Acknowledgement

I would like to thank my peers, who worked at the same time at their thesis, for our daily motivational stand-up, their emotional support and useful discussions.

REFERENCES

[1] Chen Sun et al. "Revisiting Unreasonable Effectiveness of Data in Deep Learning Era". In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Oct. 2017.

[2] J. Deng et al. "ImageNet: A Large-Scale Hierarchical Image Database". In: 2009.

[3] Qiong Cao et al. "VGGFace2: A Dataset for Recognising Faces across Pose and Age". In: *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)*. 2018, pp. 67–74. DOI: 10.1109/FG.2018.00020.

[4] Florian Schroff, Dmitry Kalenichenko, and James Philbin. "FaceNet: A Unified Embedding for Face Recognition and Clustering". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2015.

[5] European Commission. *General data protection regulation - processing of special categories of personal data*. 2018. URL: https://gdpr-info.eu/art-9-gdpr/.

[6] Jules. Harvey Adam. LaPlace. *Exposing.ai*. 2021. URL: https://exposing.ai/msceleb (visited on 01/01/2021).

[7] Tero Karras, Samuli Laine, and Timo Aila. "A Style-Based Generator Architecture for Generative Adversarial Networks". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 4396–4405. DOI: 10.1109/CVPR.2019.00453. URL: https://ieeexplore.ieee.org/document/8953766.

[8] Tero Karras et al. "Analyzing and Improving the Image Quality of StyleGAN". In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 8107–8116. DOI: 10.1109/CVPR42600.2020.00813. URL: https://ieeexplore.ieee.org/document/9156570.

[9] Laurent Colbois, Tiago de Freitas Pereira, and Sébastien Marcel. "On the use of automatically generated synthetic image datasets for benchmarking face recognition". In: *2021 IEEE International Joint Conference on Biometrics (IJCB)*. 2021, pp. 1–8. DOI: 10.1109/IJCB52358.2021.9484363.

[10] Ashish Vaswani et al. "Attention Is All You Need". In: *CoRR* abs/1706.03762 (2017). URL: http://arxiv.org/abs/1706.03762.

[11] Jacob Devlin et al. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *CoRR* abs/1810.04805 (2018). URL: http://arxiv.org/abs/1810.04805.

[12] Alec Radford et al. "Language models are unsupervised multitask learners". In: *OpenAI blog* 1.8 (2019), p. 9.

[13] Alexey Dosovitskiy et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *CoRR* abs/2010.11929 (2020). URL: https://arxiv.org/abs/2010.11929.

[14] Kai Han et al. "A Survey on Visual Transformer". In: *CoRR* abs/2012.12556 (2020). URL: https://arxiv.org/abs/2012.12556.

[15] Salman Khan et al. "Transformers in Vision: A Survey". In: *ACM Comput. Surv.* (Dec. 2021). Just Accepted. ISSN: 0360-0300. DOI: 10.1145/3505244. URL: https://doi.org/10.1145/3505244.

[16] Drew A. Hudson and C. Lawrence Zitnick. "Generative Adversarial Transformers". In: *CoRR* abs/2103.01209 (2021). URL: https://arxiv.org/abs/2103.01209.

[17] Yifan Jiang, Shiyu Chang, and Zhangyang Wang. *TransGAN: Two Transformers Can Make One Strong GAN*. 2021. URL: https://proceedings.neurips.cc/paper/2021/hash/7c220a2091c26a7f5e9f1cfb099511e3-Abstract.html.

[18] Kwonjoon Lee et al. "ViTGAN: Training GANs with Vision Transformers". In: *CoRR* abs/2107.04589 (2021). URL: https://arxiv.org/abs/2107.04589.

[19] Long Zhao et al. "Improved Transformer for High-Resolution GANs". In: *CoRR* abs/2106.07631 (2021). URL: https://arxiv.org/abs/2106.07631.

[20] Rui Xu et al. "STransGAN: An Empirical Study on Transformer in GANs". In: *CoRR* abs/2110.13107 (2021). URL: https://arxiv.org/abs/2110.13107.

[21] Bowen Zhang et al. "StyleSwin: Transformer-based GAN for High-resolution Image Generation". In: *CoRR* abs/2112.10762 (2021). URL: https://arxiv.org/abs/2112.10762.

[22] Jeeseung Park and Younggeun Kim. "Styleformer: Transformer based Generative Adversarial Networks with Style Vector". In: *CoRR* abs/2106.07023 (2021). URL: https://arxiv.org/abs/2106.07023.

[23] Patrick Esser, Robin Rombach, and Bjorn Ommer. "Taming Transformers for High-Resolution Image Synthesis". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2021, pp. 12873–12883. URL: https://openaccess.thecvf.com/content/CVPR2021/papers/Esser_Taming_Transformers_for_High-Resolution_Image_Synthesis_CVPR_2021_paper.pdf.

[24] Mehdi Mirza and Simon Osindero. "Conditional Generative Adversarial Nets". In: *CoRR* abs/1411.1784 (2014). URL: http://arxiv.org/abs/1411.1784.

[25] Nitish Shirish Keskar et al. “CTRL: A Conditional Transformer Language Model for Controllable Generation”. In: *CoRR* abs/1909.05858 (2019). URL: http://arxiv.org/abs/1909.05858.

[26] Yunjey Choi et al. “StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018.

[27] Zhenliang He et al. “AttGAN: Facial Attribute Editing by Only Changing What You Want”. In: *IEEE Transactions on Image Processing* 28.11 (2019), pp. 5464–5478. DOI: 10.1109/TIP.2019.2916751.

[28] Ming Liu et al. “STGAN: A Unified Selective Transfer Network for Arbitrary Image Attribute Editing”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.

[29] Grigory Antipov, Moez Baccouche, and Jean-Luc Dugelay. “Face aging with conditional generative adversarial networks”. In: *2017 IEEE International Conference on Image Processing (ICIP)*. 2017, pp. 2089–2093. DOI: 10.1109/ICIP.2017.8296650.

[30] Hui Ding, Kumar Sricharan, and Rama Chellappa. “ExprGAN: Facial Expression Editing With Controllable Expression Intensity”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 32.1 (Apr. 2018). URL: https://ojs.aaai.org/index.php/AAAI/article/view/12277.

[31] Jianmin Bao et al. “Towards Open-Set Identity Preserving Face Synthesis”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 6713–6722. DOI: 10.1109/CVPR.2018.00702.

[32] P.H. Vine. *Training Facial Recognition with Synthetic Faces*. Aug. 2021. URL: http://essay.utwente.nl/88198/.

[33] Paul Upchurch et al. “Deep Feature Interpolation for Image Content Changes”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017.

[34] 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 (ICCV)*. Oct. 2019.

[35] Rameen Abdal, Yipeng Qin, and Peter Wonka. “Image2StyleGAN++: How to Edit the Embedded Images?” In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.

[36] 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 (CVPR)*. June 2021, pp. 12863–12872.

[37] Yujun Shen et al. “Interpreting the Latent Space of GANs for Semantic Face Editing”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.

[38] Ralph Gross et al. “Multi-PIE”. In: *2008 8th IEEE International Conference on Automatic Face Gesture Recognition*. 2008, pp. 1–8. DOI: 10.1109/AFGR.2008.4813399.

[39] URL: https://numpy.org/doc/stable/reference/generated/numpy.corrcoef.html.

[40] shaoanlu. *Face Toolbox Keras*. URL: https://github.com/shaoanlu/face_toolbox_keras.

[41] Drew Arad Hudson. *gansformer*. URL: https://github.com/dorarad/gansformer/tree/3c0bcdee049d82318cb2f9327c8d6c7808664cd2.

[42] Martin Heusel et al. “GANs Trained by a Two Time-Scale Update Rule Converge to a Nash Equilibrium”. In: *CoRR* abs/1706.08500 (2017). URL: http://arxiv.org/abs/1706.08500.

[43] Laurent Colbois, Tiago de Freitas Pereira, and Sébastien Marcel. *bob.paper.ijcb2021_synthetic_dataset*. URL: https://gitlab.idiap.ch/bob/bob.paper.ijcb2021_synthetic_dataset.

[44] Laurens van der Maaten and Geoffrey Hinton. “Visualizing Data using t-SNE”. In: *Journal of Machine Learning Research* 9.86 (2008), pp. 2579–2605. URL: http://jmlr.org/papers/v9/vandermaaten08a.html.

[45] Idiap Research Institute. *InceptionResnetV2.py*. URL: https://gitlab.idiap.ch/bob/bob.learn.tensorflow/blob/39471f5bb2ae42cf6ef7fcc69e305d76a8b44ff9/bob/learn/tensorflow/network/InceptionResnetV2.py.

[46] Ian J. Goodfellow et al. *Generative Adversarial Networks*. 2014.

[47] Ian J. Goodfellow. “NIPS 2016 Tutorial: Generative Adversarial Networks”. In: *CoRR* abs/1701.00160 (2017). URL: http://arxiv.org/abs/1701.00160.

[48] Richard Zhang et al. “The Unreasonable Effectiveness of Deep Features as a Perceptual Metric”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018.

APPENDIX

A. Technical Background

1) Transformer

The transformer model is a type of neural network block, like the multi-layer perceptron (MLP) and convolutional neural layer in a CNN. It operates using an attention mechanism to enable long range interactions. The transformer was introduced by [10] and consists out of an encode-decoder architecture. Subsequent works such as Generative Pre-trained Transformer 2 (GPT-2) [12] and Vision Transformer (ViT) [13] only use the encoder. While the original transformer encoder replaced CNN's and recurrent neural networks with a residual multi-head attention and and a residual MLP stacked, it seems that in literature sometimes only the residual multi-head attention is referred to as the transformer and the MLP is left out or replaced with a CNN, such as in the work of the GANformer [16]. The attention mechanism provides insight, which is one of the additional benefits [10, 14, 16].

This section will describe the attention mechanism and some supporting elements of the transformer model.

a) Scaled Dot-Product Attention

The multi-head attention is the core part of the transformer model. In this section, the computation for one head is explained, whereas multi-head attention is a combination of multiple heads. The operation of one head is referred to as scaled dot-product attention (SDPA) and is shown in Figure 18. SDPA takes three inputs matrices, a query Q, a key K and a value V. Multiple types of attention mechanisms have been developed, such as the well known self-attention and the bipartite attention used in the GANformer. The various attention mechanism often differ in their input and whether the SDPA is partitioned with for example a shifting window. In self-attention the three matrices are constructed out of one input matrix X using trainable linear projection matrices as shown in Equation (5). Here X is build up from a set of N input vectors with an embedding dimension of D. For text N refers to the embedding of words and for images N can be related to the pixels or a patch in an image.

$$X \in \mathbb{R}^{\mathrm{N}\times\mathrm{D}} \tag{4}$$

$$Q = XW^Q \quad K = XW^K \quad V = XW^V \tag{5}$$

Using the Q, K and V matrices the SDPA is performed as shown in Equation (6). The result is a weighted value matrix, where each element is weighted by the attention score on all elements of the input sequence X. First the matrix multiplication QK^T gets the inner product of the matrices. The resulting score is a degree of attention, a similarity between the elements. The scores are normalized, which enhances the gradient stability [14]. Next the $Softmax$ converts the values to probabilities. The last operation multiplies the probabilities with the value matrix V to acquire a weighted value matrix. Note that the input and output matrices have the same dimension.

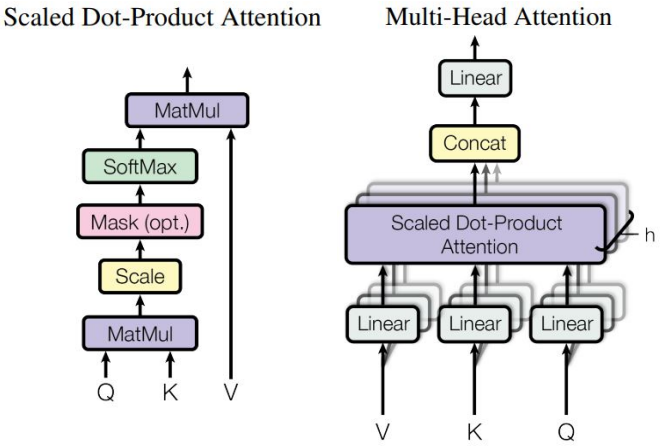


Fig. 18: Scaled Dot-Product Attention (left) and Multi-Head Self-Attention (right) from [10]

Due to the fact that every element attends to every other element, global interaction is enabled. A downside is the quadratic computational load $O(n^2)$ with large dimensions or many elements.

$$Attention(Q, K, V) = Softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{6}$$

$$Attention(Q, K, V) \in R^{N\times d_v} \tag{7}$$

b) Multi-Head Self Attention

With a single attention mechanism as the one explained above the focus between different interactions are averaged. By adding parallel attention mechanisms, equally important interactions are not influenced by one another. In addition to that, each attention mechanism has its own weight matrices W. These transform the input to different subspaces, each eventually trained for specific patterns and tasks [10].

The combination of self-attention is the multi-head self-attention (MSA), as shown in Equation (8), where each attention mechanism is a head. A schematic of MSA can be found in Figure 18.The outcome of each head is concatenated and transformed with a weight matrix W^0 as shown in Equation (8). Note that the output has the same dimension as input matrix X. To incorporate the output information of the attention mechanism, a residual connection with normalisation such as $LayerNorm(X + MSA(X))$ is performed.

$$\begin{aligned} MSA(Q, K, V) &= Concat(h_1, ..., h_h)W_O \\ \text{where } h_i &= Attention(QW_i^Q, KW_i^K, VW_i^V) \end{aligned} \tag{8}$$

c) Positional Encoding

While CNNs and RNNs process the input data in a structured manner, due to the matrix multiplications of

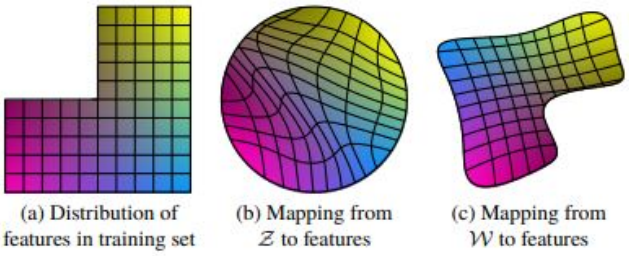


Fig. 19: A visual example of the entanglement problem in the original GAN. (a) In the training data some variations between attributes are missing, such as in the combination with gender and facial hair. (b) To match the latent space Z with the distribution of the training data, the latent space is warped and entangled. (c) By introducing a mapping network, these attributes can get disentangled. Figure from [7].

all elements this is not inherently present in the attention mechanism. Therefore information about the position of the elements need to be added. A common method is to add this information to the elements embeddings. Multiple works have discussed how this should be included for text and image based models, variating between fixed, learned, absolute, relative, 1D and 2D. It turns out that the implementation method is not relevant, as long as the information is included [10, 13].

2) GANs and StyleGAN

The generative adversarial network (GAN) architecture consists out of generator G and a discriminator D that are training in a competitive game [46, 47]. In face generation the generator takes a random input vector z and generates an output $G(z)$ that resembles a face based on the distribution of the training data, such as FFHQ [8]. The goal of the generator is to fool the discriminator by generating realistic images. The discriminator is either presented with a real face x or with a fake face $G(z)$ and needs to detect whether it is real or fake.

A problem in image generation is that the distribution of latent space Z should match to the distribution of the training set. This results in entanglement of attributes in the latent space, as illustrated in Figure 19. The authors of [7, 8] propose a style-based generator, consisting out of a mapping f and a synthesis g network. A schematic overview is shown in Figure 20. The sampled vector z is first embedded into an intermediate latent vector $f(z) = w$. Since the distribution of latent space W can be learned, it tends to result in a more disentangled and linear space. The synthesis network learns a constant, before each convolutional layer the disentangled style vector w inserted. With stochastic inputs at each layer minor variations affecting for example freckles and hair are introduced.

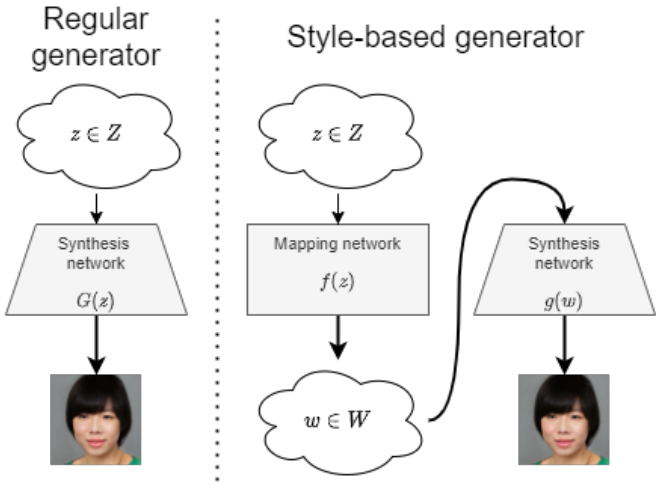


Fig. 20: A schematic overview of a regular generator and a style based generator. The regular generator builds the image on top of the latent z. In the style based generator a mapping network f maps $z \rightarrow w$ in an intermediate latent space W. The image is build up from a constant value in the synthesis network, the style vector is transferred in every layer.

a) Projector

The StyleGAN2 projector provides an optimization framework to find a representative latent vector w that synthesizes a similar looking image to a given target image [8]. The initial latent w is the average over 10000 mappings. Besides this the noise maps that are inserted in the layers of the synthesis network are optimised as well. Due to this the loss function is based on an image quality term and a weighted noise regularization term as well. According to the authors the latter prevents that the target image signal creeps into the noise maps. The image quality term is based on the Learned Perceptual Image Patch Similarity (LPIPS) distance [48].

3) GANformer

The GANformer is a style based model that includes transformers within the mapping and synthesis networks [16]. The authors recognize shortcomings in both the StyleGAN model and the transformer model using self-attention. First the main differences with StyleGAN considering the implementation of the transformers are clarified. After this bipartite attention, an alternative to self-attention for high dimensional data is discussed in the following section.

StyleGAN uses an intermediate latent vector w, as an input to the synthesis network. A downside is that the vector is applied globally, the vector is not able to emphasize or only influence parts of the face such as the hair or skin. The GANformer breaks the latent vector down into a number of equally sized latent components Y. The latent components attend to the image, which is effectively a spatial operation. Therefore the style of each latent component are applied locally in the image.

While in StyleGAN the style vector is inserted before every convolutional layer, the latent components prop-

agate from the start through the synthesis network. Instead of the modulation, the vectors attend to the image features. Note that the value of the latent components is not adjusted by the attention mechanism.

a) Bipartite Attention

In the following sections the notation of [16] is kept, where the intermediate latent vector w equals the flattened latent components Y.

In self-attention all pairwise relations of the input are considered and each single element is updated by attending to all other elements. This process is computationally expensive when it is applied on high dimensional data such as images. Combining the idea of the latent vector w and attention the authors of the GANformer propose a more general attention mechanism, called bipartite attention as shown in Equation (9). Instead of acting on itself, a bipartite attention features a bipartite graph. Its inputs can be divided into two disjoint and independent sets X and Y, in which every element of each set is connected to all the elements of the other set. It is a generalisation such that when $X = Y$, regular self-attention is applied.

In the GANformer $X^{n\times d}$ are the image features which ultimately result in the generated image or face, where the size of the image is represented by $n = H \times W$. $Y^{m\times d}$ represents the latent variable, the concurrent format for the intermediate latent vector w in the regular StyleGAN2. m is the number of latent components, whereas d is the dimension of either the image features or the latent components, equal in both X and Y. Since $m \in [8, 32]$ and thus $m < n$, the computational complexity is reduced to a bilinear complexity of $O(mn)$.

$$a(X, Y) = Attention(q(X), k(Y), v(Y)) \tag{9}$$

In general, bipartite attention $a(A, B)$ propagates information from B to A. Note that the reverse operation $a(B, A)$ returns the same probability matrix, but transposed: $AB^T = (BA^T)^T$. The propagation is the result of the last operation between the probabilities between A and B and the value matrix.

b) Simplex Attention

Besides the bipartite input, two update rules are proposed that include scaling and bias, these are the simplex and duplex update rule. In simplex Equation (10), the information is distributed in a single direction over the bipartite graph.

The image features X are normalized to zero-mean and unit-variance using Equation (11). The result of the bipartite attention is mapped by γ and β to function as scale and bias respectively on the normalized image features. Note that this operation is very similar to adaptive instance normalization (AdaIN), the operation used to fuse the style vector w with the image features in the original StyleGAN [7]. The difference is that the style information is based on the attention between the latent components and the image features, rather a learned affine transformation of the style vector.

$$u^s(X, Y) = \gamma(a(X, Y)) \odot \omega(X) + \beta(a(X, Y)) \tag{10}$$

$$\omega(X) = \frac{X - \mu(X)}{\sigma(X)} \tag{11}$$

c) Duplex Attention

In the duplex attention, information propagates in both directions between the image features X and latent variables Y. Instead of only storing the style content, the latent variables Y form a key value structure on their own, $Y = (K^{m\times d}, V^{m\times d})$. Here the value V stores the style just as Y in the simplex attention, where the addition is that the key K tracks the centroids of the semantic segmentent it attends to. The duplex update rule is shown in Equation (12). Note that the latter part of the computation is similar to the simplex update rule Equation (10), in which information is propagated from the latent components to the image features.

However before this update is calculated, information is also propagated from the image features to the centroids K. The centroids K are defined by as in Equation (13), note the order difference from X to Y.

Using this duplex structure the information flows both ways. X updates the latent centroids K, whereas the latents values V update the image features X. Note that in this work duplex attention is used.

$$\begin{aligned} u^d(X, Y) = \gamma(a(X, K, V)) \odot \omega(X) \\ + \beta(a(X, K, V)) \end{aligned} \tag{12}$$

$$K = a(Y, X) \tag{13}$$

4) Attention Maps

The GANformer returns the synthesised face as well as multiple attention maps. The attention maps are produced in each attention layer and represents the attention distribution between the latent components. These maps can give insight into the attention mechanism and the generative process. In the results on the CLEVR model published by the authors, it is shown that the role of the latent variables changes throughout the model. Whereas layer components show segmenting behaviour in the lower layers, the latent component in higher layers seem to represent the surface normal.

Unfortunately such a analysis is not done for generated faces by the FFHQ face model. The behaviour of the attention on generating faces will be analysed, an example of which can be seen in Figure 2. In contrast to the CLEVR scenes, a face is not a collection of individual elements, but more like continuously connected facial semantics.

The attention maps are a simplified graphical representation of the probabilities as calculated in $a(X, Y)$, that is the output of $Softmax(\frac{QK^T}{\sqrt{d_k}})$, thus before the

value V is multiplied as shown in the attention Equation (6). For each image feature element, its probability with each centroid of its latent component is given. The latent component with the highest probability will be shown for that pixel in the attention map. Therefore the attention maps shows at every location the dominant latent component. The attention maps do not show how the image features are affected by the rest of the network, such as the convolutions, or how the centroids K are updated.

B. Implementation Details

As mentioned in Section III-B the model acquired from [41] is different than described in the paper [16]. In this section the most noteworthy differences in the implementation details of the default model is compared to the details described in the paper.

Exact information about the model used in the paper is not provided, nor is it clear for how many steps the pre-trained model is actually trained for for a valid comparison. On the Github [41], the author note that the models are trained between 5k-15k kimg-steps. This versus 50-70k kimg-steps for StyleGAN2.

One difference is that $Y^{n=16,d=32}$ for the pre-trained model, while the authors note that their dimensions were $Y^{n=8,d=16}$. Their choice was based on performance, though not disclosing more information about the performance metric and performance differences between other implementations.

Besides that, the authors mention that multi-head attention was used, while the default model only has one head. This is odd, as all the attention in one layer, thus all pairwise relations between the image and the latent components, is averaged. This can be problematic as equally important pairwise attentions, for example attention to the mouth and the hair, influence one another. In earlier work, such as in the vision transformer[13], multi-head attention was applied as well.

Another major difference is the addition of a global latent component, a vector with dimension d similar to the latent component dimension. Before bipartite attention is applied, the global latent modulates the image features X uniformly. The idea is that the global latent modulates "holistic aspects of the image such as global lighting conditions, global style properties for e.g. faces, etc."[41].

C. t-SNE

t-SNE is an unsupervised statistical method to visualize high dimensional data in a low dimensional space [44]. Its visualisation it is used for data exploration. t-SNE estimates the distribution of high dimensional points, or latents in this work. This distribution is then reflected in an iterative manner to a lower dimensional space, by minimizing the Kullback–Leibler (KL) divergence between the high and low dimensional probability distributions.

The probability distribution of the high dimensional points is acquired with Equation (14). p_{ij} represents the similarity between the points i and j. The set of all similarities represents its probability distribution. The probability of picking a certain pair (i, j) is then proportional to the similarity of the pair. As the smaller distances have a high probability, the local pairwise structure of the space is maintained when minimizing the KL divergence.

$$p_{ij} = \frac{exp(-||x_i - x_j||^2)/2\sigma^2}{\sum_k \sum_{l=k} exp(-||x_k - x_l||^2)/2\sigma^2} \tag{14}$$

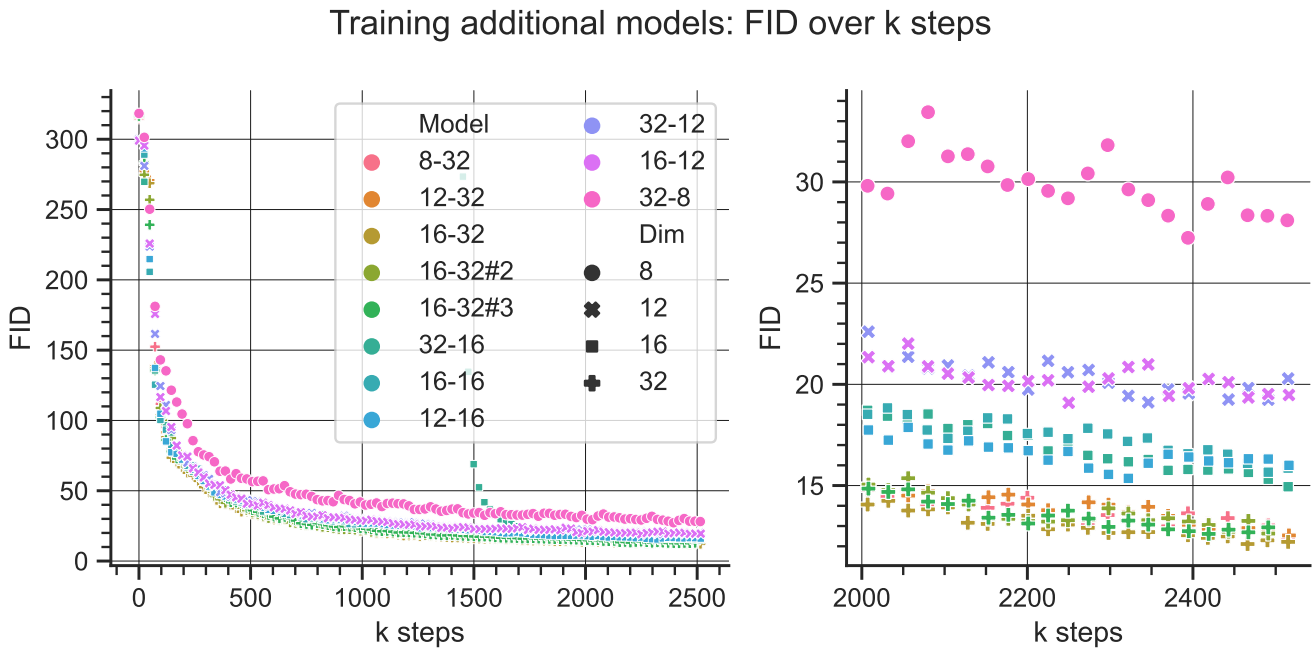


Fig. 21: The FID quality during the training for each trained model. The left plot shows the complete training, the right plot shows the last 5000 kimg steps in detail. Each latent dimension is indicated with a different marker.

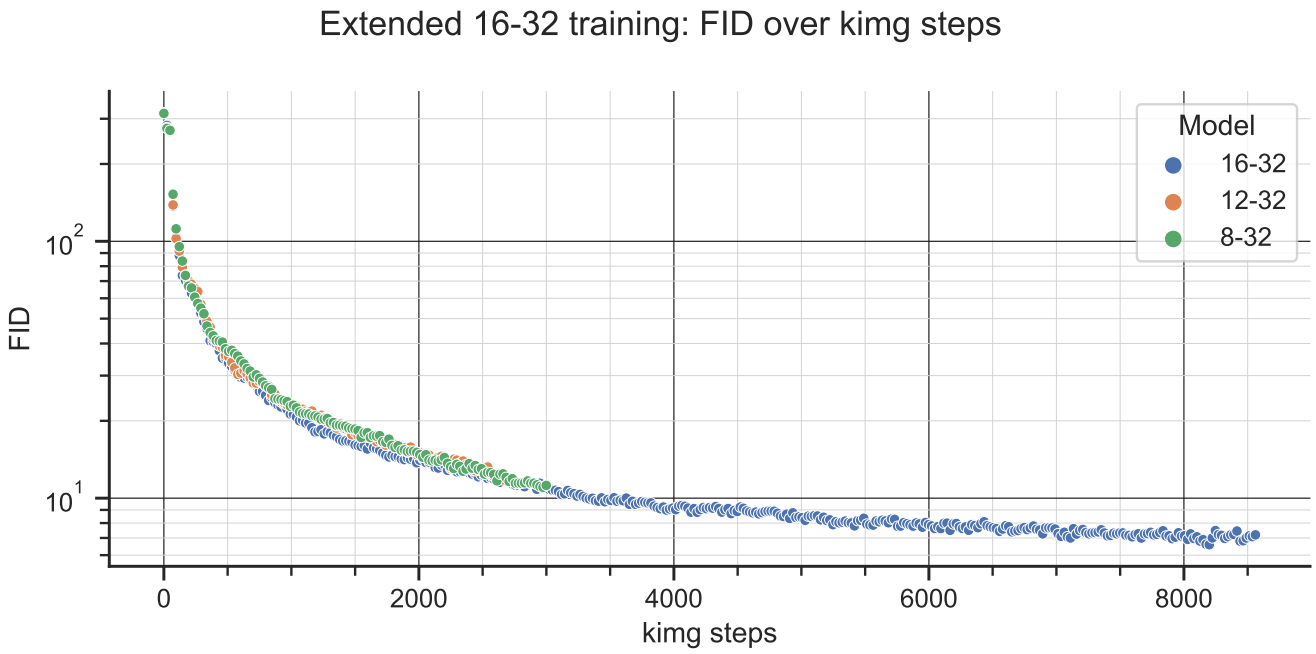
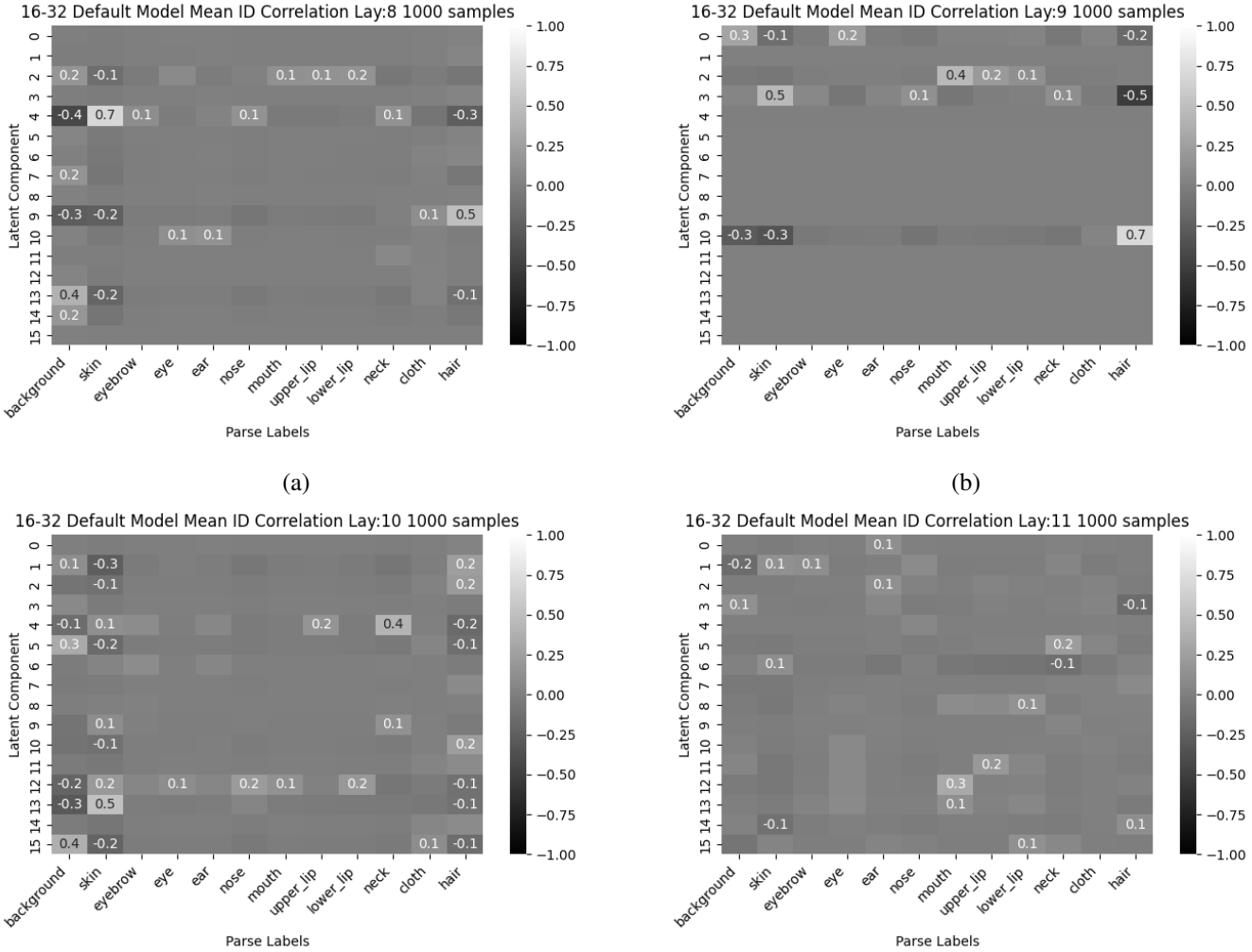


Fig. 22: The FID quality during the training for the extended training of the $16 - 32$ model.



(a) (b)

(c) (d)

Fig. 23: The correlation between the latent components and the parse labels of layer 8, 9, 10 and 11 for the **default** network based on 1000 generated samples. Only if the correlation $|c| > 0.1$ the correlations are annotated within the figure.

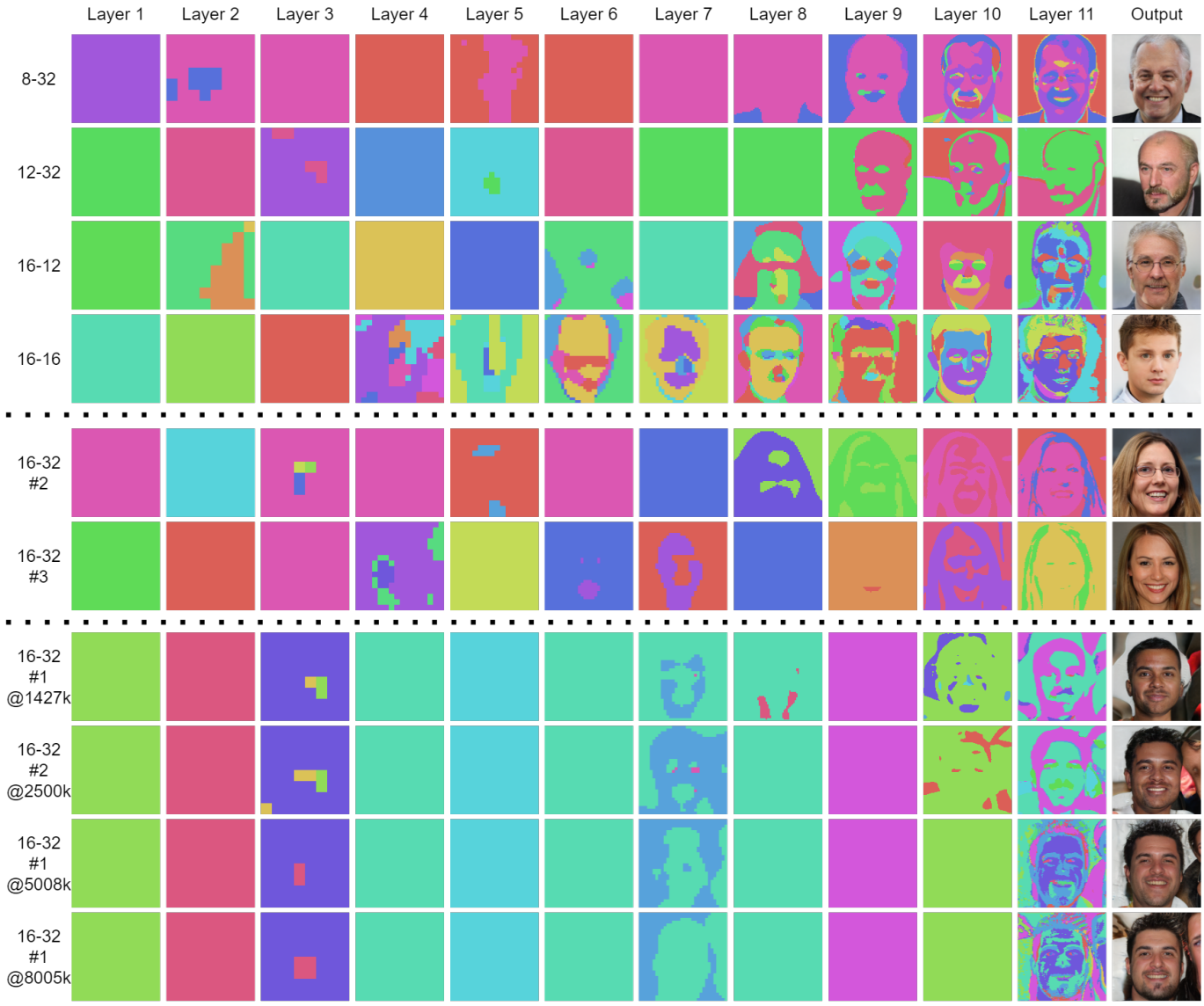


Fig. 24: Example of attention maps and outputs for each layer for each analysed trained model.

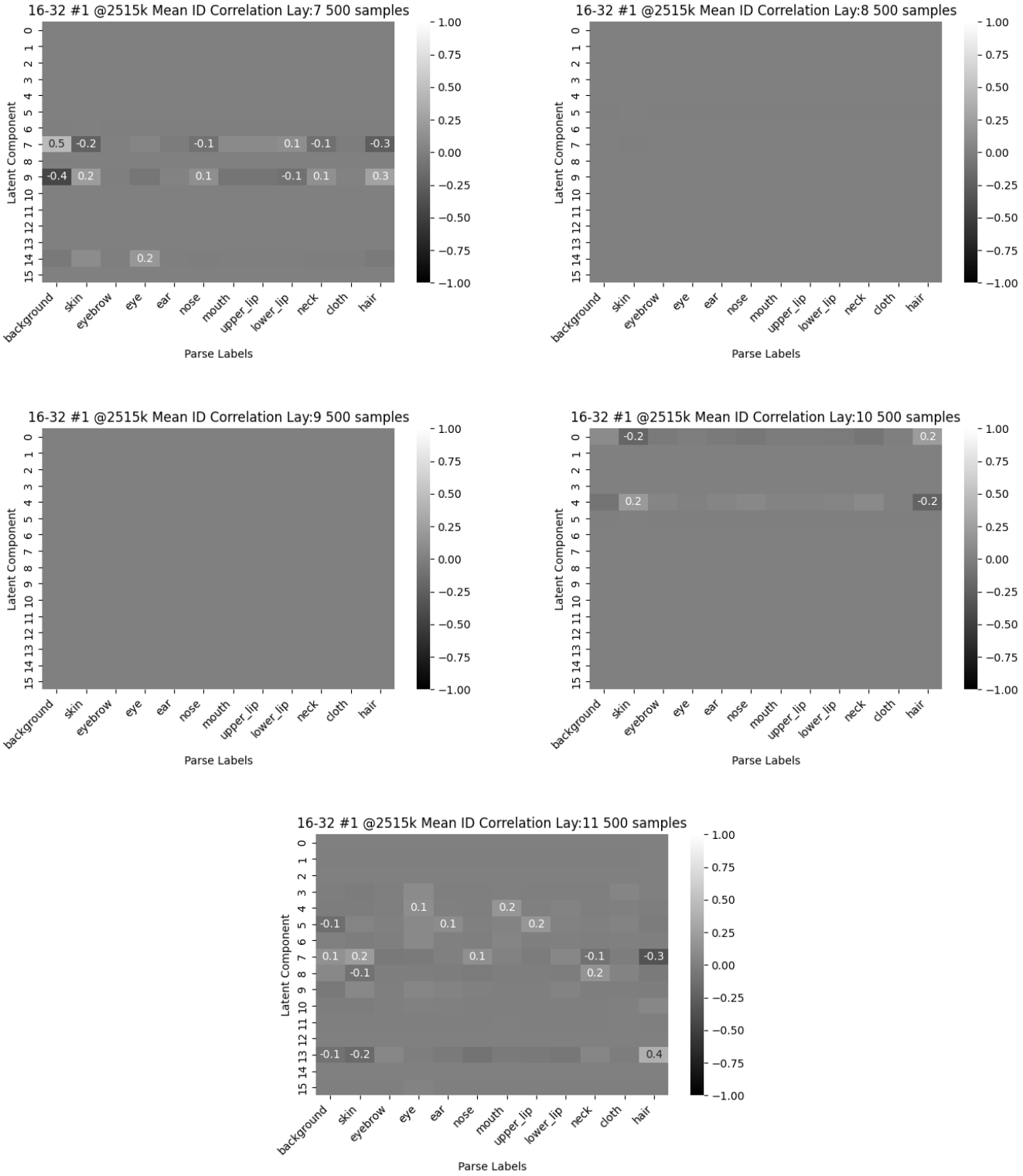


Fig. 25: The correlation between the latent components and the parse labels of layer 7, 8, 9, 10 and 11 for the **16-32 #1** network based on 500 generated samples. Only if the correlation $|c| > 0.1$ the correlations are annotated within the figure.

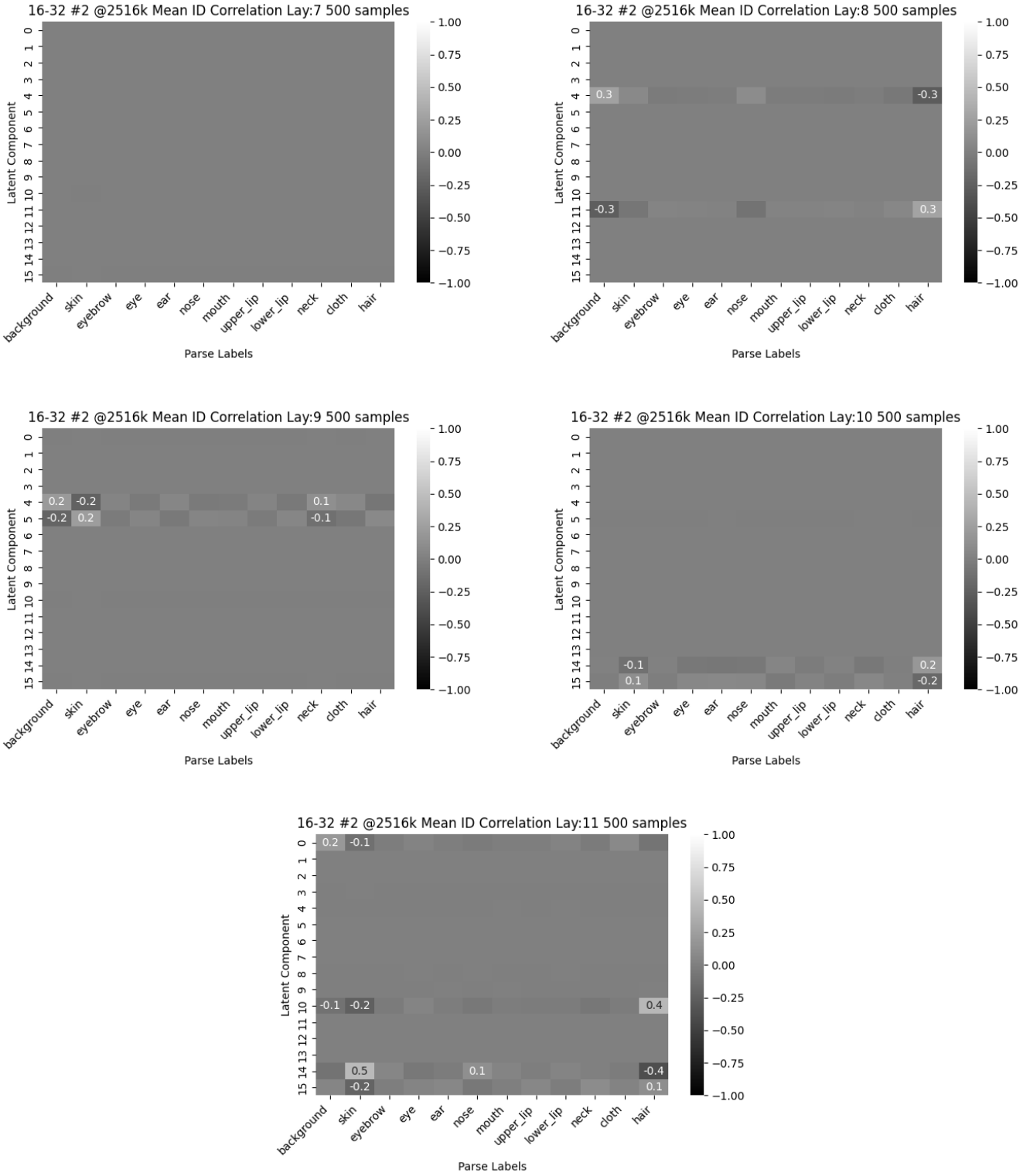


Fig. 26: The correlation between the latent components and the parse labels of layer 7, 8, 9, 10 and 11 for the **16-32 #2** network based on 500 generated samples. Only if the correlation $|c| > 0.1$ the correlations are annotated within the figure.

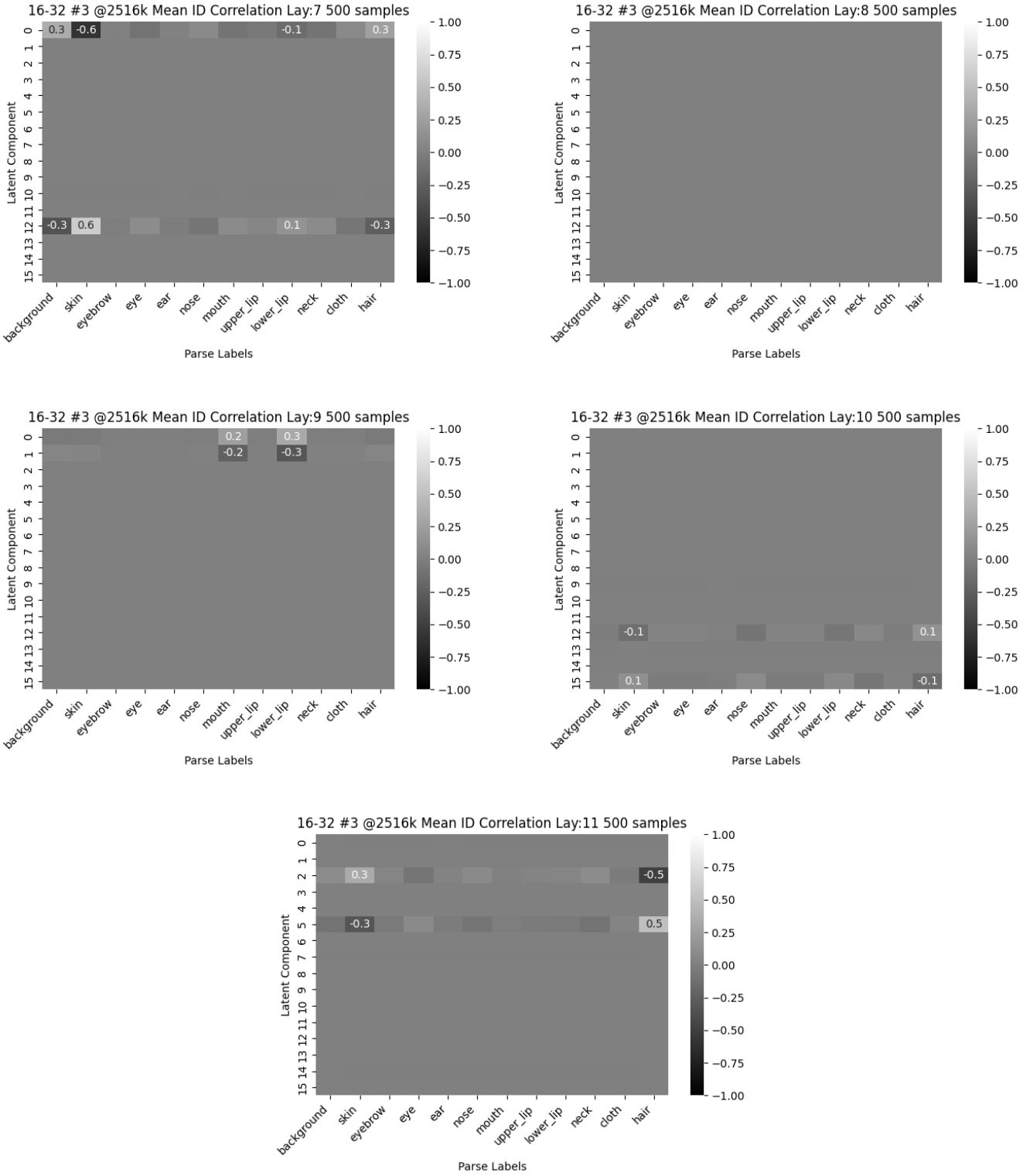


Fig. 27: The correlation between the latent components and the parse labels of layer 7, 8, 9, 10 and 11 for the **16-32 #3** network based on 500 generated samples. Only if the correlation $|c| > 0.1$ the correlations are annotated within the figure.

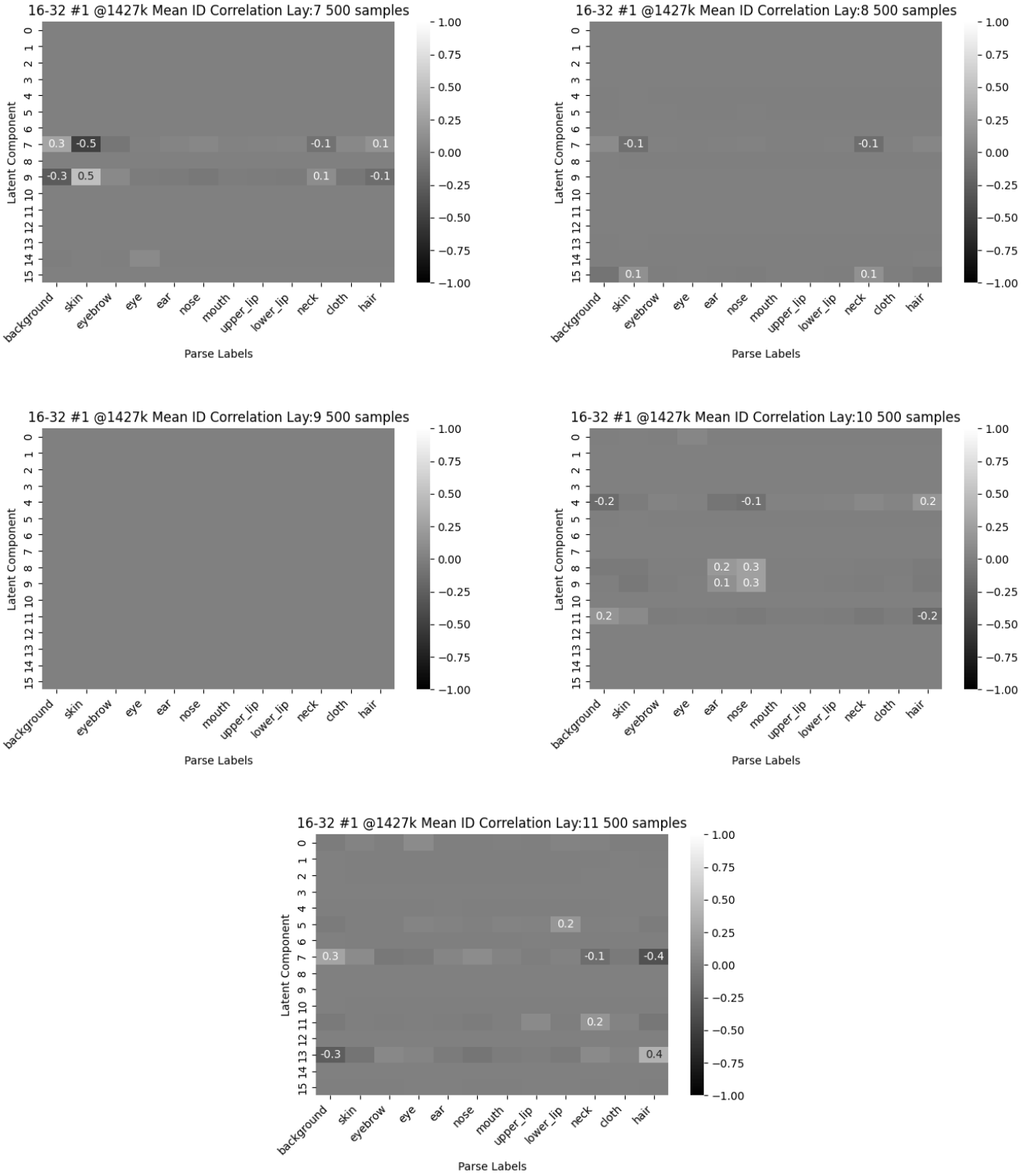


Fig. 28: The correlation between the latent components and the parse labels of layer 7, 8, 9, 10 and 11 for the **16-32 #1** network trained for 1427k steps based on 500 generated samples. Only if the correlation $|c| > 0.1$ the correlations are annotated within the figure.

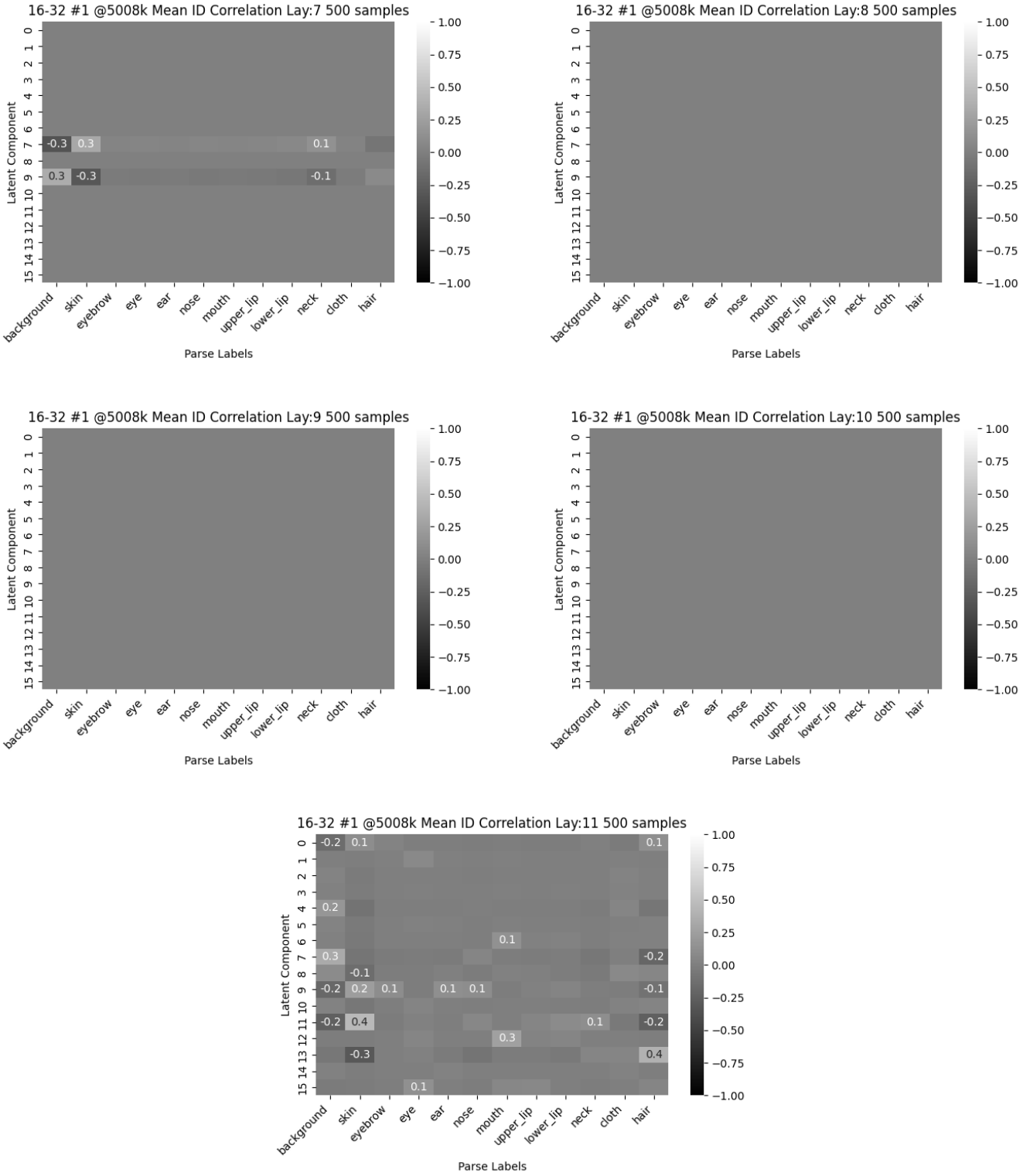


Fig. 29: The correlation between the latent components and the parse labels of layer 7, 8, 9, 10 and 11 for the **16-32 #1** network trained for 5008k steps based on 500 generated samples. Only if the correlation $|c| > 0.1$ the correlations are annotated within the figure.

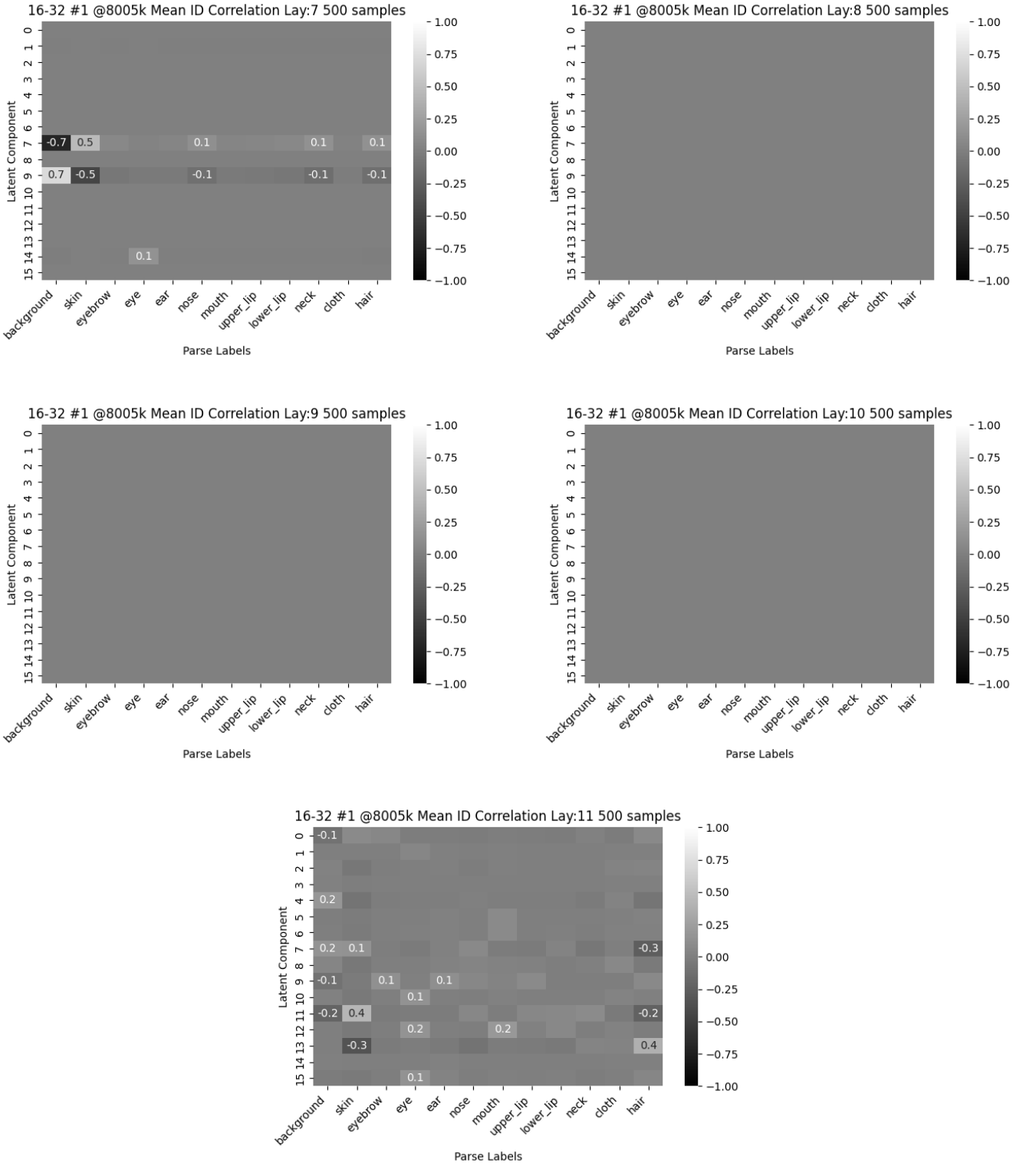


Fig. 30: The correlation between the latent components and the parse labels of layer 7, 8, 9, 10 and 11 for the **16-32 #1** network trained for 8005k steps based on 500 generated samples. Only if the correlation $|c| > 0.1$ the correlations are annotated within the figure.

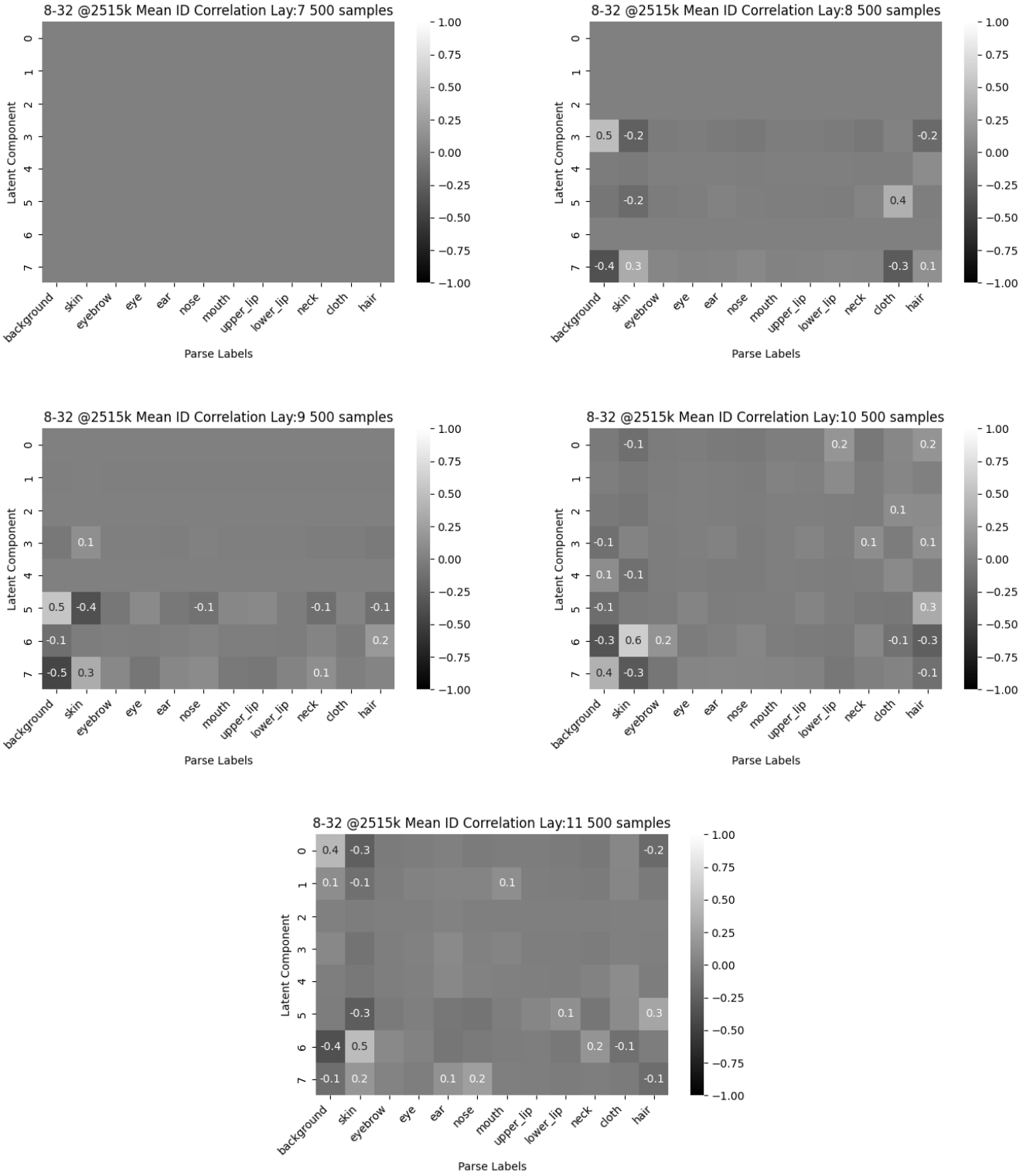


Fig. 31: The correlation between the latent components and the parse labels of layer 7, 8, 9, 10 and 11 for the **8-32** network based on 500 generated samples. Only if the correlation $|c| > 0.1$ the correlations are annotated within the figure.

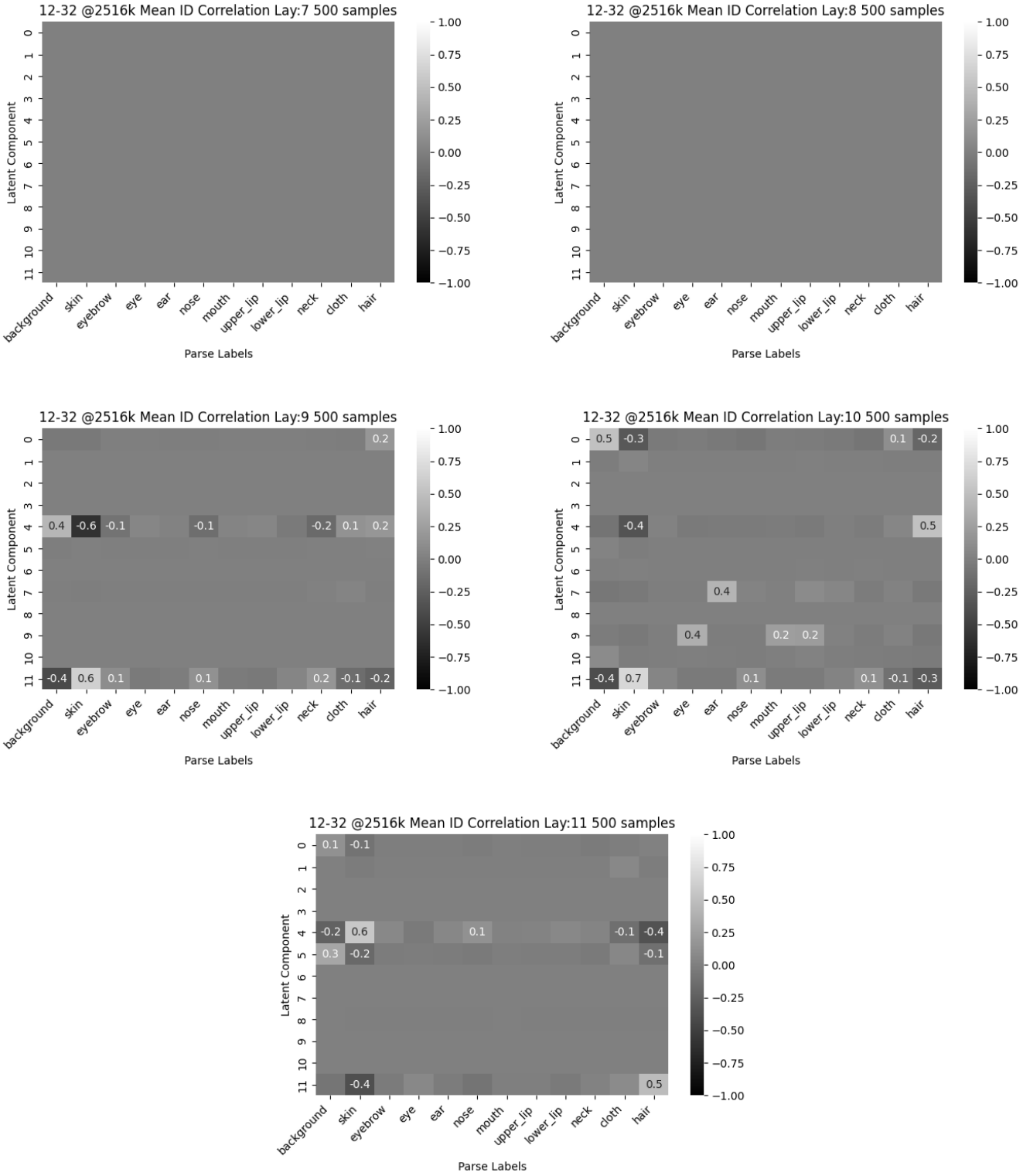


Fig. 32: The correlation between the latent components and the parse labels of layer 7, 8, 9, 10 and 11 for the **12-32** network based on 500 generated samples. Only if the correlation $|c| > 0.1$ the correlations are annotated within the figure.

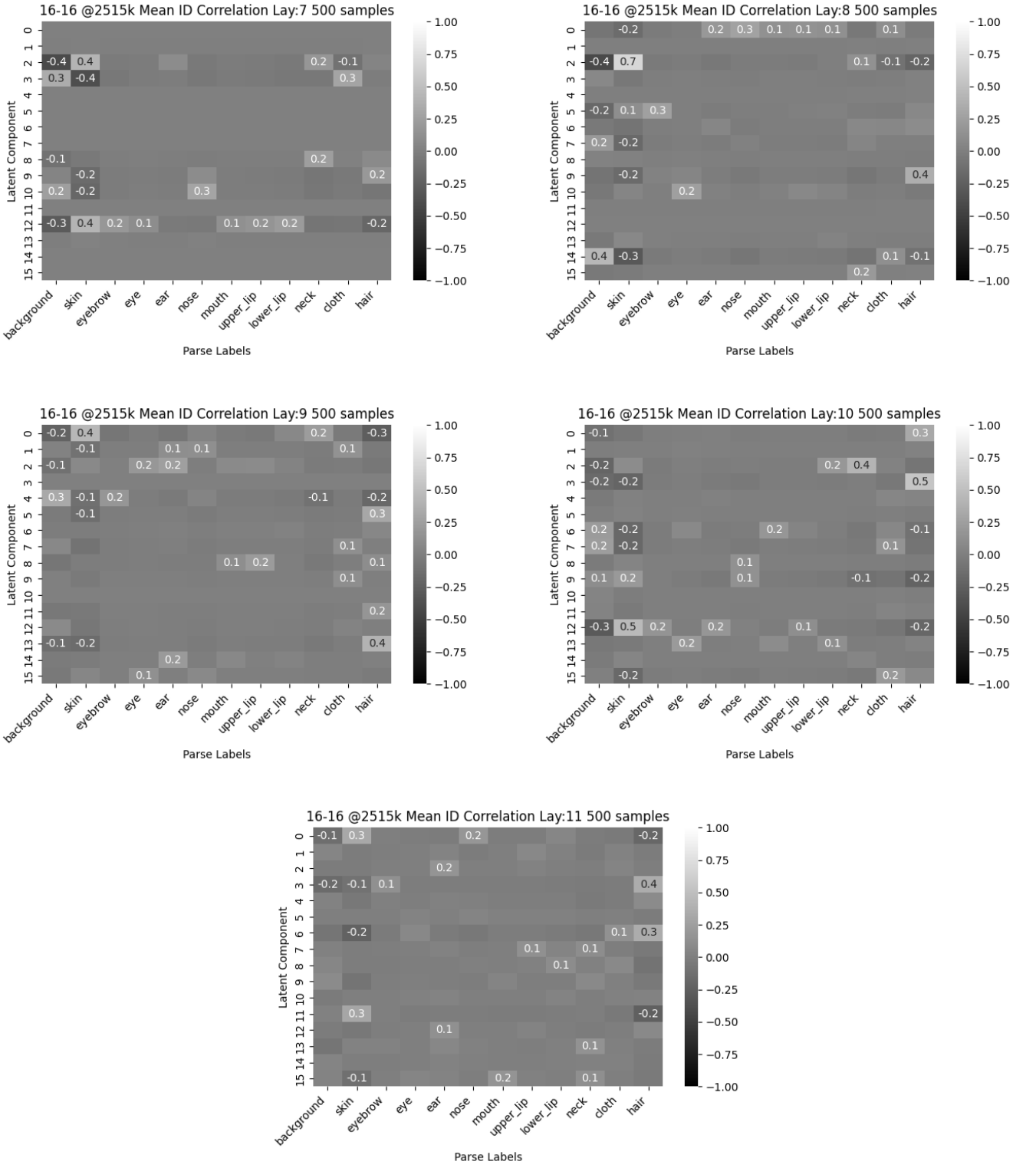


Fig. 33: The correlation between the latent components and the parse labels of layer 7, 8, 9, 10 and 11 for the **16-16** network based on 500 generated samples. Only if the correlation $|c| > 0.1$ the correlations are annotated within the figure.

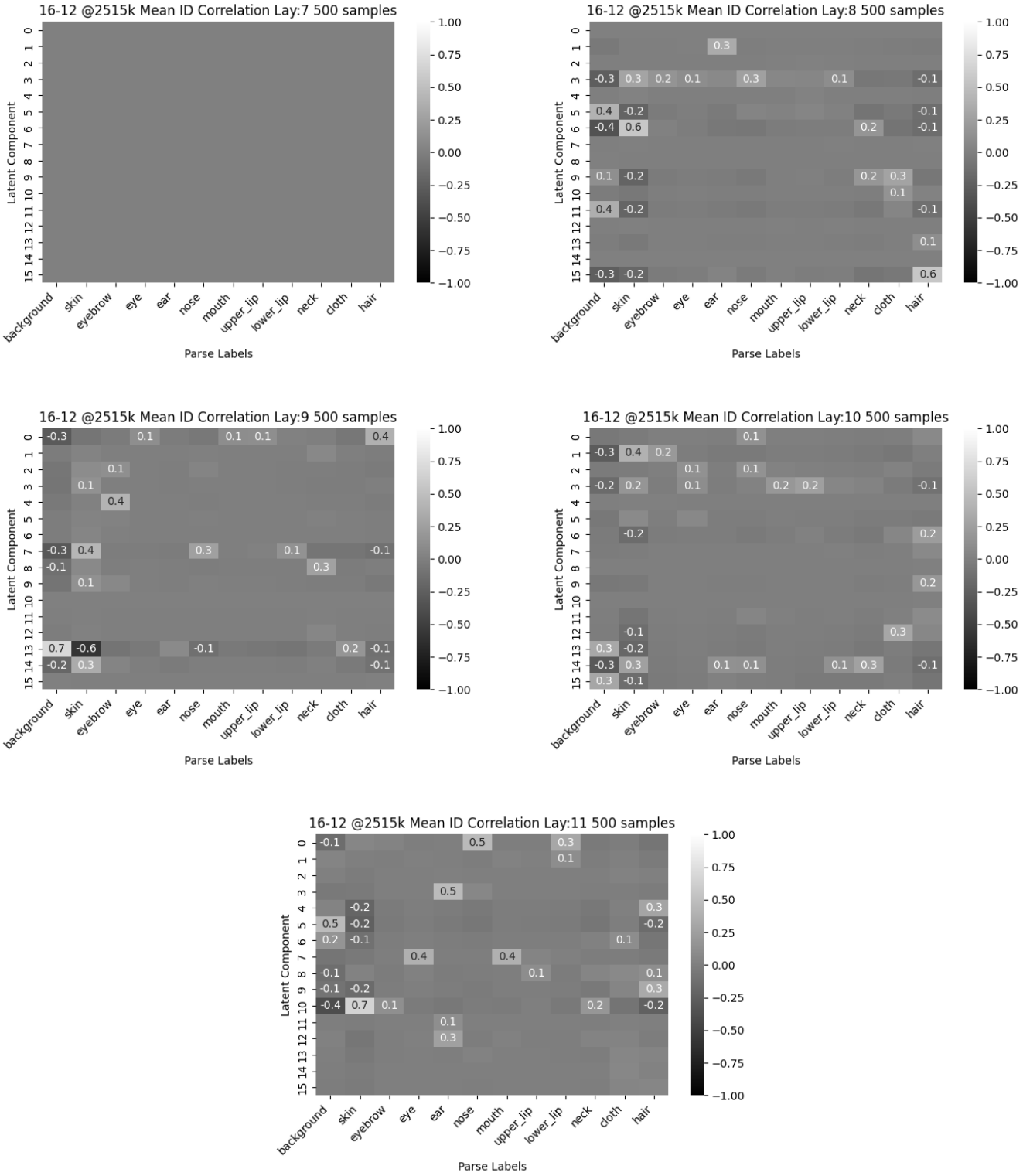


Fig. 34: The correlation between the latent components and the parse labels of layer 7, 8, 9, 10 and 11 for the **16-12** network based on 500 generated samples. Only if the correlation $|c| > 0.1$ the correlations are annotated within the figure.

Fig. 35: Generated faces from standard normal sampling in the Z space

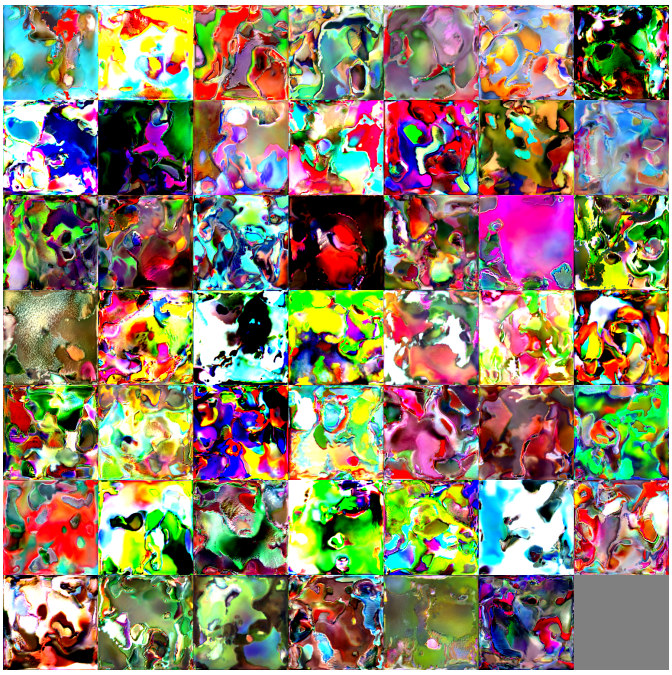


Fig. 37: Generated faces from standard normal sampling in the W space.

Fig. 36: Face generation by interpolating 5 random z vectors using 8 steps.

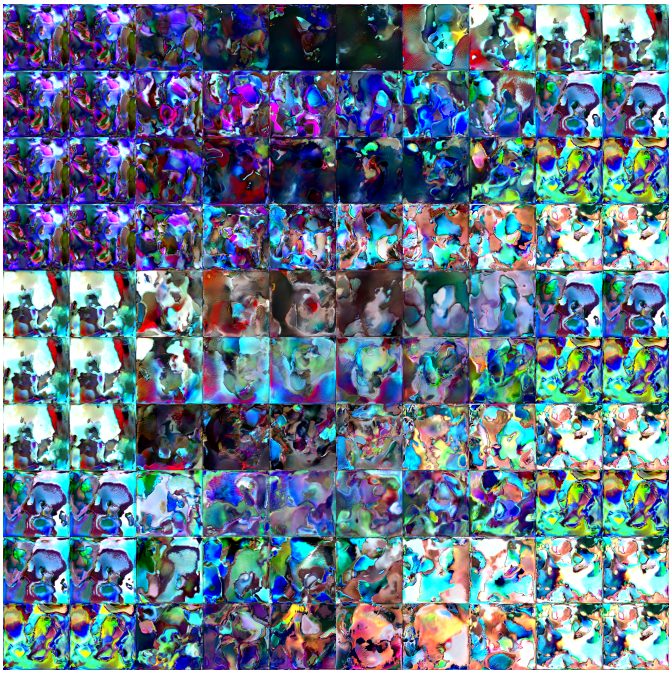


Fig. 38: Face generation by interpolating 5 random w vectors using 8 steps.

Fig. 39: Face generation by mapping 5 z vectors to w vectors and interpolating these using 8 steps.

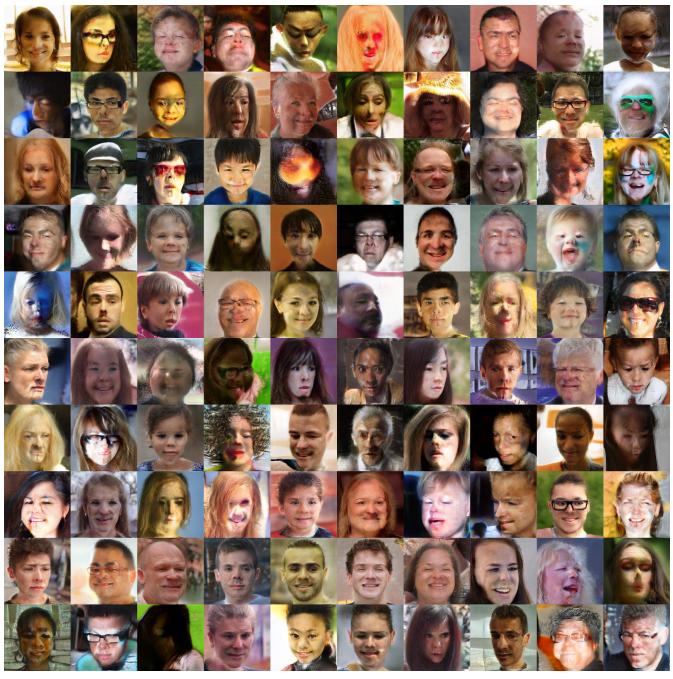


Fig. 40: Face generation from random w based on a normal distribution acquired by 10000 mappings.

Fig. 41: Face generation from random w based on reconstructive sampling of the distribution using SVD acquired by 10000 mappings.

Fig. 42: A example of pose projection using the GANformer.

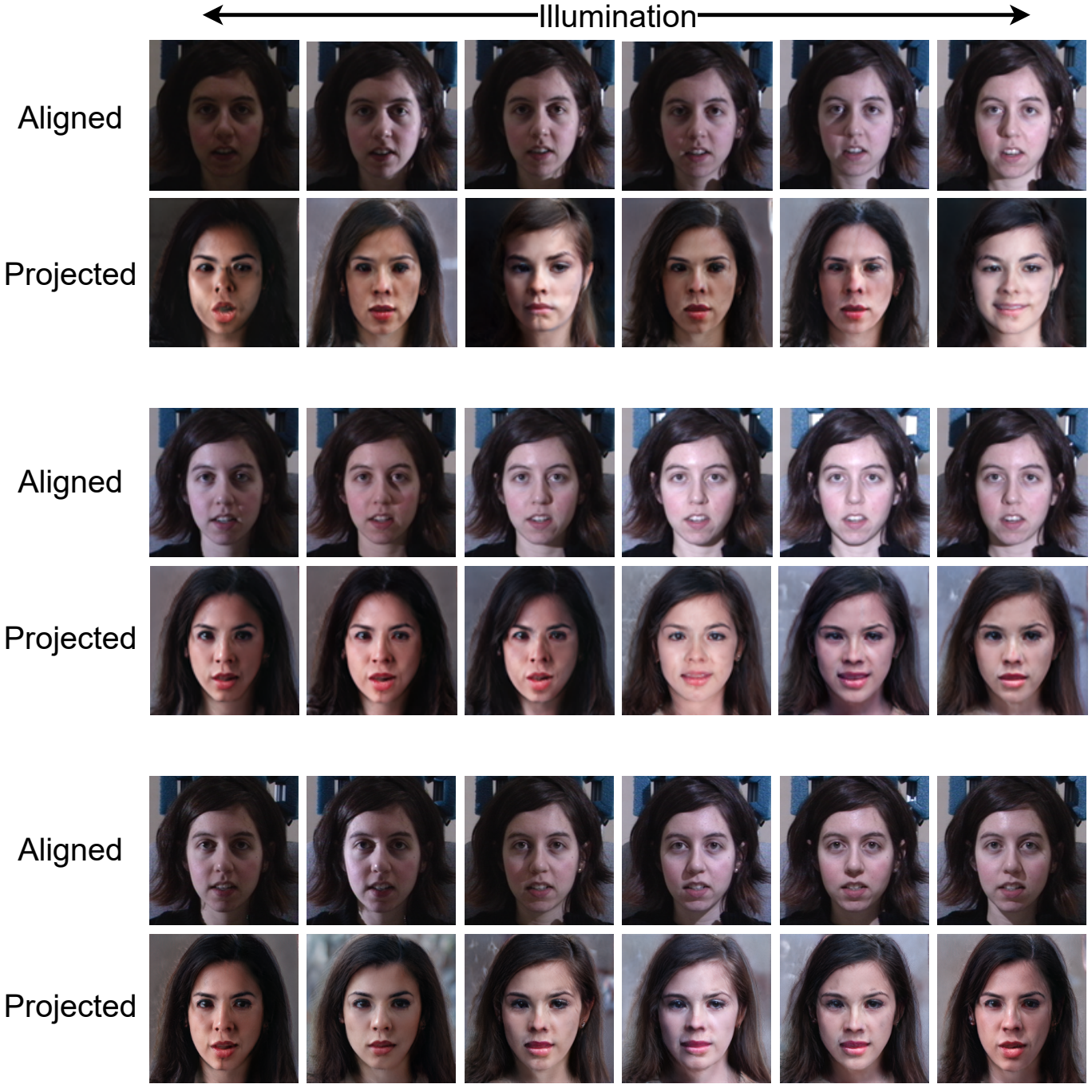


Fig. 43: A example of illumination projection using the GANformer.

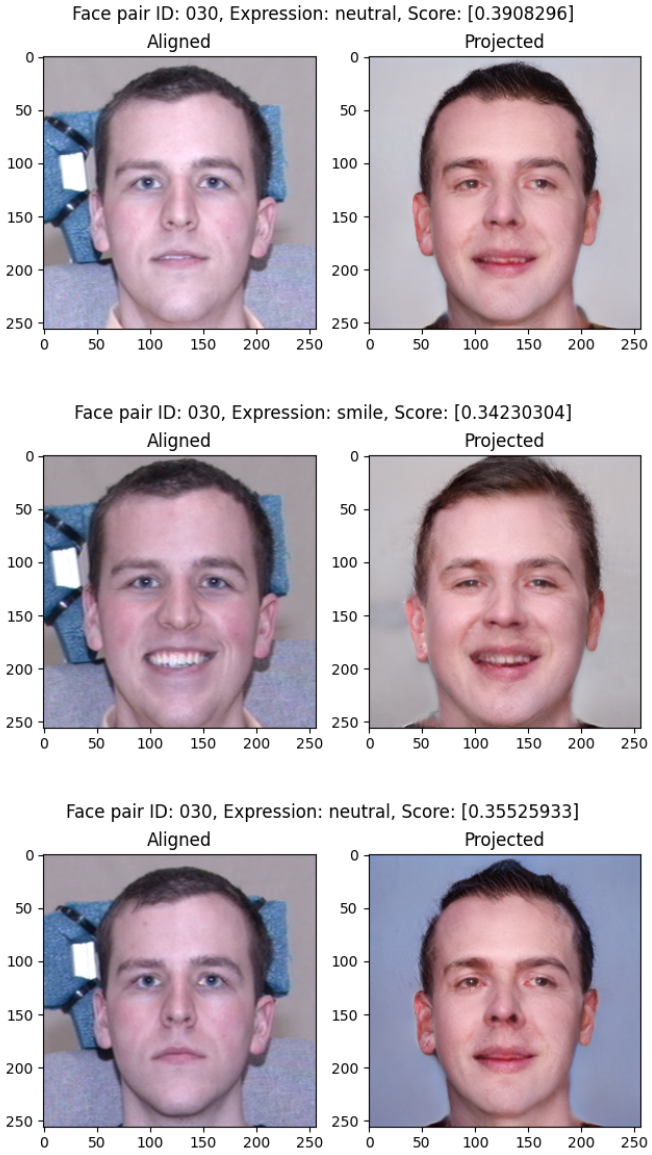


Fig. 44: A selection of the highest scores alignment projection pairs.

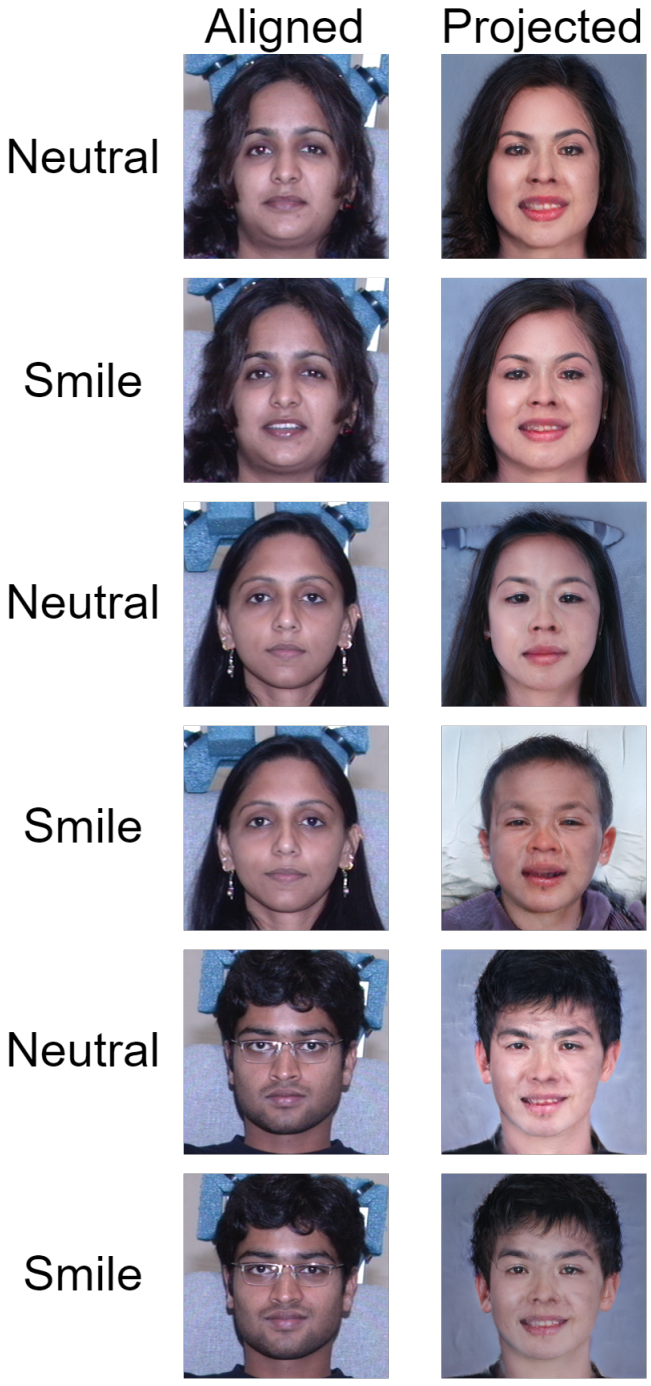


Fig. 45: An example of projecting faces with a dark skin tone using the GANformer.

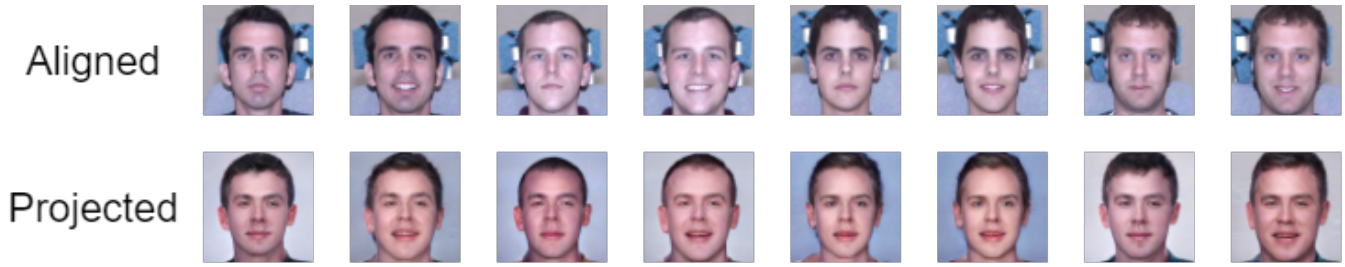


Fig. 46: An example of similar projected faces using the GANformer while their aligned origins are more distanced.

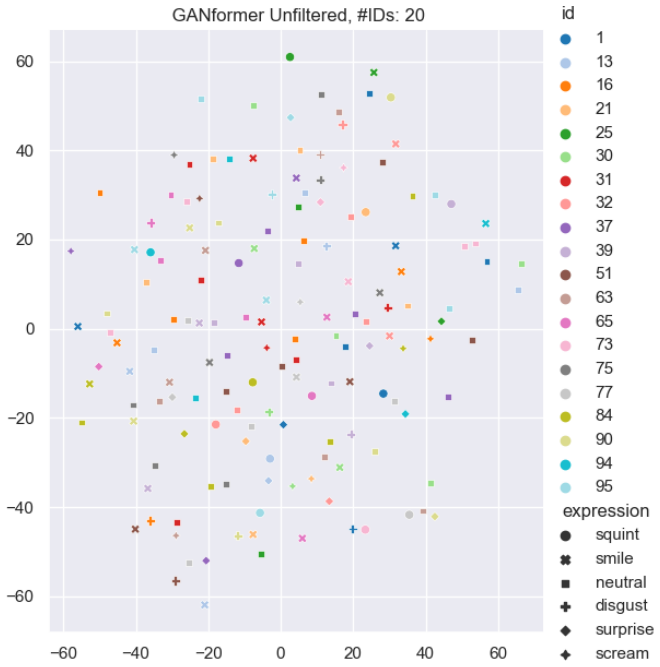


Fig. 47: A t-SNE analysis on unfiltered projected latents of the GANformer model. With a perplexity of 10.0

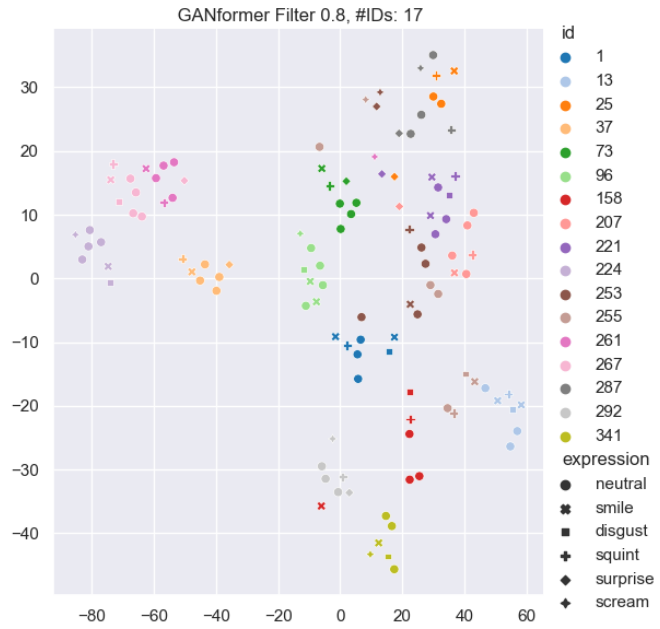


Fig. 48: A t-SNE analysis on filtered projected latents with a threshold of 0.8 of the GANformer model. With a perplexity of 10.0.

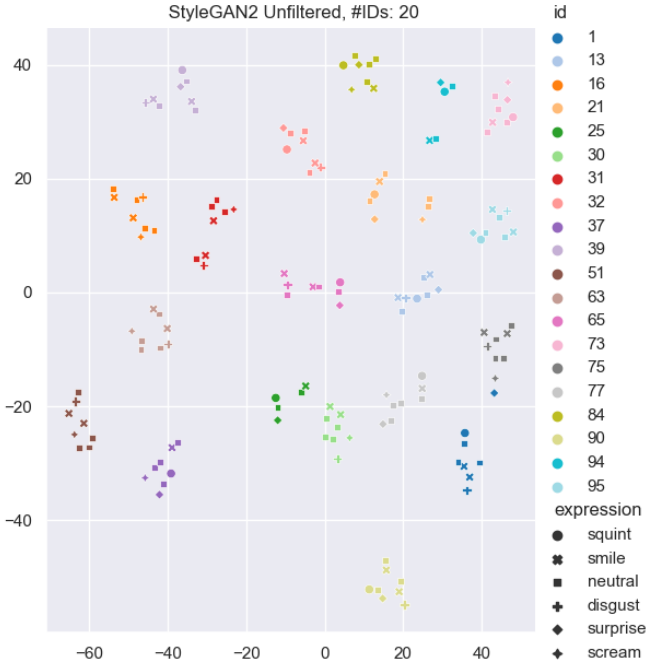


Fig. 49: A t-SNE analysis on projected latents of the StyleGAN2 model. With a perplexity of 10.0.

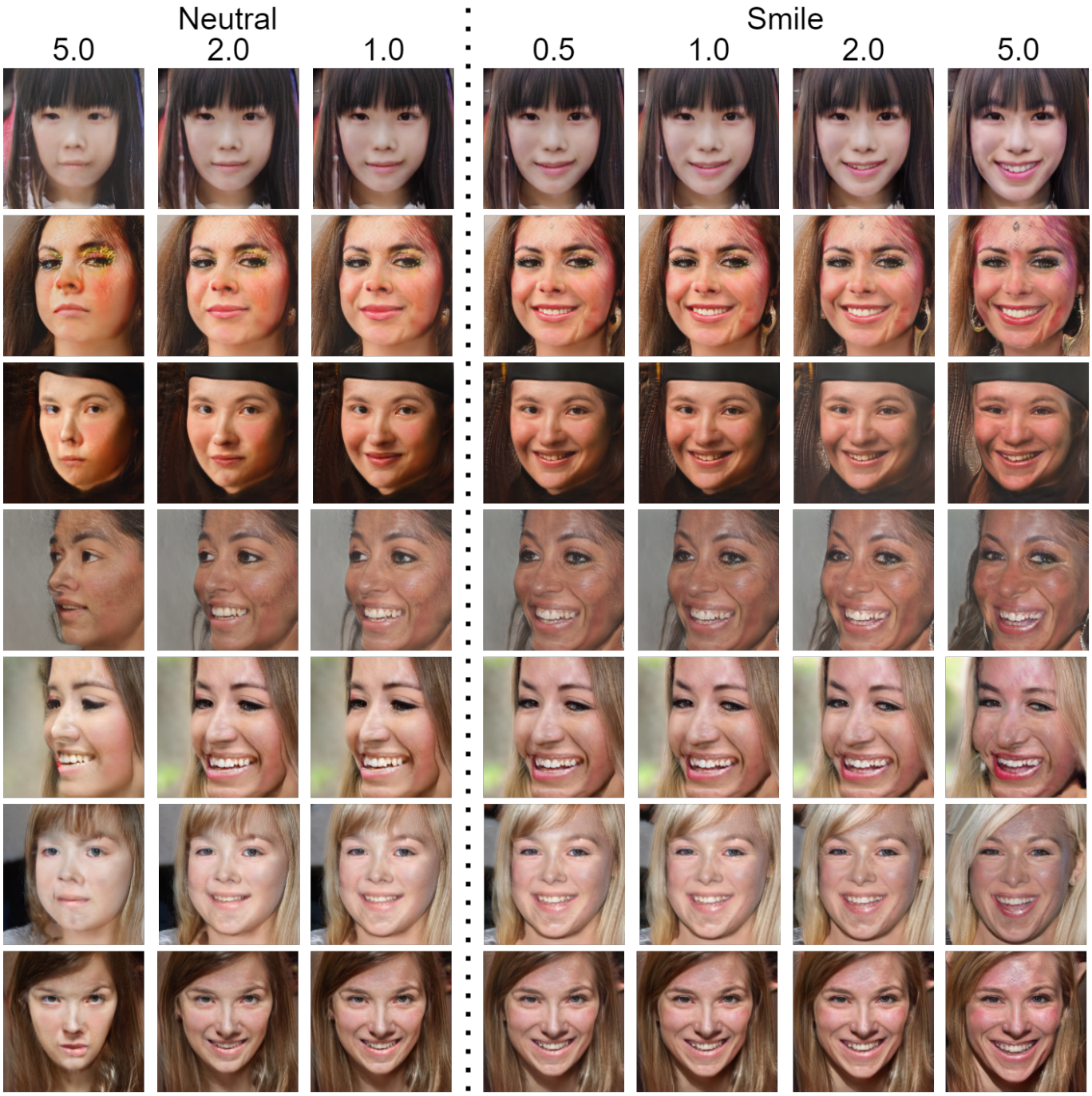


Fig. 50: Effect of scaling the the distance for both neutralisation and the smiling attribute.