**A Style-Based Generator Architecture for Generative Adversarial Networks**

Tero Karras, Samuli Laine, Timo Aila

**Abstract**

We propose an alternative generator architecture for generative adversarial networks, borrowing from style transfer literature. The new architecture leads to an automatically learned, unsupervised separation of high-level attributes (e.g., pose and identity when trained on human faces) and stochastic variation in the generated images (e.g., freckles, hair), and it enables intuitive, scale-specific control of the synthesis. The new generator improves the state-of-the-art in terms of traditional distribution quality metrics, leads to demonstrably better interpolation properties, and also better disentangles the latent factors of variation. To quantify interpolation quality and disentanglement, we propose two new, automated methods that are applicable to any generator architecture. Finally, we introduce a new, highly varied and high-quality dataset of human faces.스타일 전달 문헌에서 차용하여 생성적 적대 네트워크에 대한 대안 생성기 아키텍처를 제안합니다. 새로운 아키텍처는 자동으로 학습되고 감독되지 않은 높은 수준의 속성(예: 사람 얼굴에 대해 훈련될 때 포즈 및 ID)과 생성된 이미지(예: 주근깨, 머리카락)의 확률적 변화를 분리하고 직관적이고 확장성을 가능하게 합니다. 합성의 특정 제어. 새로운 생성기는 기존의 분포 품질 메트릭 측면에서 최첨단을 개선하고, 명백하게 더 나은 보간 속성으로 이어지며, 변동의 잠재 요인을 더 잘 풀 수 있습니다. 보간 품질과 얽힘을 정량화하기 위해 모든 생성기 아키텍처에 적용할 수 있는 두 가지 새로운 자동화 방법을 제안합니다. 마지막으로 새롭고 다양한 고품질 사람 얼굴 데이터 세트를 소개합니다.

**1. Introduction**

The resolution and quality of images produced by generative methods — especially generative adversarial networks (GAN) [22] — have seen rapid improvement recently [30, 45, 5]. Yet the generators continue to operate as black boxes, and despite recent efforts [3], the understanding of various aspects of the image synthesis process, e.g., the origin of stochastic features, is still lacking. The properties of the latent space are also poorly understood, and the commonly demonstrated latent space interpolations [13, 52, 37] provide no quantitative way to compare different generators against each other.

생성 방법, 특히 GAN(Generative Adversarial Networks)에 의해 생성된 이미지의 해상도와 품질은 최근에 빠르게 개선되었습니다. 그러나 Generator는 계속해서 블랙박스로 작동하고 있으며 최근의 노력에도 불구하고[3], 확률적 특징의 기원과 같은 영상 합성 과정의 다양한 측면에 대한 이해가 여전히 부족합니다. 잠재 공간의 속성도 제대로 이해되지 않고 있으며 일반적으로 입증된 잠재 공간 보간은 서로 다른 생성기를 서로 비교할 수 있는 정량적 방법을 제공하지 않습니다.

Motivated by style transfer literature [27], we re-design the generator architecture in a way that exposes novel ways to control the image synthesis process. Our generator starts from a learned constant input and adjusts the “style” of the image at each convolution layer based on the latent code, therefore directly controlling the strength of image features at different scales. Combined with noise injected directly into the network, this architectural change leads to automatic, unsupervised separation of high-level attributes (e.g., pose, identity) from stochastic variation (e.g., freckles, hair) in the generated images, and enables intuitive scale-specific mixing and interpolation operations. We do not modify the discriminator or the loss function in any way, and our work is thus orthogonal to the ongoing discussion about GAN loss functions, regularization, and hyperparameters [24, 45, 5, 40, 44, 36].

스타일 전달 문헌에 동기를 부여하여 이미지 합성 프로세스를 제어하는 ​​새로운 방법을 노출하는 방식으로 생성기 아키텍처를 재설계합니다. 생성기는 학습된 상수 입력에서 시작하여 잠재 코드를 기반으로 각 컨볼루션 레이어에서 이미지의 "스타일"을 조정하므로 다양한 스케일에서 이미지 기능의 강도를 직접 제어합니다. 네트워크에 직접 주입된 노이즈와 결합된 이 아키텍처 변경은 생성된 이미지의 확률적 변화(예: 주근깨, 머리카락)에서 상위 수준 속성(예: 포즈, 아이덴티티)의 감독되지 않은 자동 분리로 이어지며 직관적인 확장을 가능하게 합니다. 특정 혼합 및 보간 작업. 우리는 판별기 또는 손실 함수를 어떤 식으로든 수정하지 않으며 따라서 우리의 작업은 GAN 손실 함수, 정규화 및 하이퍼파라미터에 대한 진행 중인 논의와 직교합니다.

Our generator embeds the input latent code into an intermediate latent space, which has a profound effect on how the factors of variation are represented in the network. The input latent space must follow the probability density of the training data, and we argue that this leads to some degree of unavoidable entanglement. Our intermediate latent space is free from that restriction and is therefore allowed to be disentangled. As previous methods for estimating the degree of latent space disentanglement are not directly applicable in our case, we propose two new automated metrics — perceptual path length and linear separability — for quantifying these aspects of the generator. Using these metrics, we show that compared to a traditional generator architecture, our generator admits a more linear, less entangled representation of different factors of variation.

생성기는 입력 잠재 코드를 중간 잠재 공간에 삽입하며, 이는 네트워크에서 변동 요인이 표현되는 방식에 중대한 영향을 미칩니다. 입력 잠재 공간은 훈련 데이터의 확률 밀도를 따라야 하며, 이것이 어느 정도 피할 수 없는 얽힘으로 이어진다고 주장합니다. 우리의 중간 잠재 공간은 그러한 제한에서 자유롭고 따라서 풀릴 수 있습니다. 잠재 공간 얽힘의 정도를 추정하는 이전 방법은 우리의 경우 직접 적용할 수 없으므로 생성기의 이러한 측면을 정량화하기 위해 지각 경로 길이와 선형 분리 가능성이라는 두 가지 새로운 자동화 메트릭을 제안합니다. 이러한 메트릭을 사용하여 기존 생성기 아키텍처와 비교하여 생성기가 다양한 변동 요인에 대해 보다 선형적이고 덜 얽힌 표현을 허용한다는 것을 보여줍니다.

Finally, we present a new dataset of human faces (Flickr-Faces-HQ, FFHQ) that offers much higher quality and covers considerably wider variation than existing high-resolution datasets (Appendix A). We have made this dataset publicly available, along with our source code and pre-trained networks.1 The accompanying video can be found under the same link.

마지막으로 기존의 고해상도 데이터 세트(부록 A)보다 훨씬 더 높은 품질을 제공하고 훨씬 더 넓은 변형을 포함하는 새로운 인간 얼굴 데이터 세트(Flickr-Faces-HQ, FFHQ)를 제시합니다. 우리는 소스 코드 및 사전 훈련된 네트워크와 함께 이 데이터 세트를 공개적으로 사용할 수 있도록 만들었습니다.1 함께 제공되는 비디오는 동일한 링크에서 찾을 수 있습니다.

**2. Style-based generator**

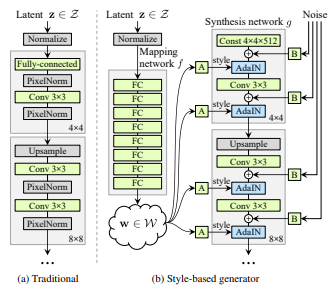
****

Figure 1. While a traditional generator [30] feeds the latent code though the input layer only, we first map the input to an intermediate latent space W, which then controls the generator through adaptive instance normalization (AdaIN) at each convolution layer. Gaussian noise is added after each convolution, before evaluating the nonlinearity. Here “A” stands for a learned affine transform, and “B” applies learned per-channel scaling factors to the noise input. The mapping network f consists of 8 layers and the synthesis network g consists of 18 layers— two for each resolution . The output of the last layer is converted to RGB using a separate 1 × 1 convolution, similar to Karras et al. [30]. Our generator has a total of 26.2M trainable parameters, compared to 23.1M in the traditional generator.

그림 1. 기존의 생성기[30]는 입력 레이어를 통해서만 잠재 코드를 제공하지만 먼저 입력을 중간 잠재 공간 W에 매핑한 다음 각 컨볼루션 레이어에서 적응형 인스턴스 정규화(AdaIN)를 통해 생성기를 제어합니다. 가우스 잡음은 비선형성을 평가하기 전에 각 컨볼루션 후에 추가됩니다. 여기서 "A"는 학습된 아핀 변환을 나타내고 "B"는 학습된 채널별 스케일링 계수를 노이즈 입력에 적용합니다. 매핑 네트워크 f는 8개의 레이어로 구성되고 합성 네트워크 g는 각 해상도 〖(4〗^2-1024^2)에 대해 2개씩 18개의 레이어로 구성됩니다. 마지막 레이어의 출력은 Karras et al.과 유사한 별도의 1 × 1 컨볼루션을 사용하여 RGB로 변환됩니다. [30]. 기존 생성기의 23.1M에 비해 생성기에는 총 26.2M의 훈련 가능한 매개변수가 있습니다.

Traditionally the latent code is provided to the generator through an input layer, i.e., the first layer of a feedforward network (Figure 1a). We depart from this design by omitting the input layer altogether and starting from a learned constant instead (Figure 1b, right). Given a latent code z in the input latent space Z, a non-linear mapping network first produces w ∈ W (Figure 1b, left). For simplicity, we set the dimensionality of both spaces to 512, and the mapping f is implemented using an 8-layer MLP, a decision we will analyze in Section 4.1. Learned affine transformations then specialize w to styles that control adaptive instance normalization (AdaIN) [27, 17, 21, 16] operations after each convolution layer of the synthesis network g. The AdaIN operation is defined as전통적으로 잠재 코드는 입력 계층, 즉 피드포워드 네트워크의 첫 번째 계층을 통해 생성기에 제공됩니다(그림 1a). 입력 레이어를 완전히 생략하고 대신 학습된 상수에서 시작하여 이 설계에서 출발합니다(그림 1b, 오른쪽). 입력 잠재 공간 Z의 잠재 코드 z가 주어지면 비선형 매핑 네트워크 f:Z→W는 먼저 w ∈ W를 생성합니다(그림 1b, 왼쪽). 단순화를 위해 두 공간의 차원을 512로 설정하고 매핑 f는 4.1절에서 분석할 8계층 MLP를 사용하여 구현됩니다. 학습된 아핀 변환은 합성 네트워크 g의 각 컨볼루션 계층 이후 적응 인스턴스 정규화(AdaIN) [27, 17, 21, 16] 작업을 제어하는 ​​스타일 y=(y\_s,y\_b)에 w를 특화합니다. AdaIN 연산은 다음과 같이 정의됩니다.



where each feature map is normalized separately, and then scaled and biased using the corresponding scalar components from style y. Thus the dimensionality of y is twice the number of feature maps on that layer.

여기서 각 기능 맵 는 개별적으로 정규화되고 스타일 y의 해당 스칼라 구성 요소를 사용하여 크기가 조정되고 바이어스됩니다. 따라서 y의 차원은 해당 레이어의 피쳐 맵 수의 두 배입니다.

Comparing our approach to style transfer, we compute the spatially invariant style y from vector w instead of an example image. We choose to reuse the word “style” for y because similar network architectures are already used for feedforward style transfer [27], unsupervised image-toimage translation [28], and domain mixtures [23]. Compared to more general feature transforms [38, 57], AdaIN is particularly well suited for our purposes due to its efficiency and compact representation.

스타일 전송에 대한 접근 방식을 비교하여 예제 이미지 대신 벡터 w에서 공간적으로 불변 스타일 y를 계산합니다. 유사한 네트워크 아키텍처가 이미 피드포워드 스타일 전송[27], 비지도 이미지-이미지 변환[28] 및 도메인 혼합[23]에 사용되기 때문에 y에 대해 "스타일"이라는 단어를 재사용하기로 선택했습니다. 보다 일반적인 기능 변환[38, 57]과 비교할 때 AdaIN은 효율성과 간결한 표현으로 인해 우리의 목적에 특히 적합합니다.

테이블이(가) 표시된 사진

자동 생성된 설명

Table 1. Frechet inception distance (FID) for various generator designs (lower is better). In this paper we calculate the FIDs using 50,000 images drawn randomly from the training set, and report the lowest distance encountered over the course of training.표 1. 다양한 발전기 설계에 대한 FID(Frechet inception distance)(낮을수록 좋음) 이 논문에서는 훈련 세트에서 무작위로 추출한 50,000개의 이미지를 사용하여 FID를 계산하고 훈련 과정에서 발생하는 가장 낮은 거리를 보고합니다.

Finally, we provide our generator with a direct means to generate stochastic detail by introducing explicit noise inputs. These are single-channel images consisting of uncorrelated Gaussian noise, and we feed a dedicated noise image to each layer of the synthesis network. The noise image is broadcasted to all feature maps using learned perfeature scaling factors and then added to the output of the corresponding convolution, as illustrated in Figure 1b. The implications of adding the noise inputs are discussed in Sections 3.2 and 3.3.마지막으로 명시적 노이즈 입력을 도입하여 확률적 세부 정보를 생성하는 직접적인 수단을 생성기에 제공합니다. 이것은 상관관계가 없는 가우스 노이즈로 구성된 단일 채널 이미지이며 합성 네트워크의 각 레이어에 전용 노이즈 이미지를 제공합니다. 노이즈 이미지는 학습된 성능 스케일링 계수를 사용하여 모든 기능 맵에 브로드캐스트된 다음 그림 1b와 같이 해당 컨볼루션의 출력에 추가됩니다. 노이즈 입력 추가의 의미는 섹션 3.2 및 3.3에서 논의됩니다.

**2.1. Quality of generated images**

Before studying the properties of our generator, we demonstrate experimentally that the redesign does not compromise image quality but, in fact, improves it considerably. Table 1 gives Frechet inception distances (FID) [25] for various generator architectures in CELEBA-HQ [30] and our new FFHQ dataset (Appendix A). Results for other datasets are given in Appendix E. Our baseline configuration (A) is the Progressive GAN setup of Karras et al. [30], from which we inherit the networks and all hyperparameters except where stated otherwise. We first switch to an improved baseline (B) by using bilinear up/downsampling operations [64], longer training, and tuned hyperparameters. A detailed description of training setups and hyperparameters is included in Appendix C. We then improve this new baseline further by adding the mapping network and AdaIN operations (C), and make a surprising observation that the network no longer benefits from feeding the latent code into the first convolution layer. We therefore simplify the architecture by removing the traditional input layer and starting the image synthesis from a learned 4 × 4 × 512 constant tensor (D). We find it quite remarkable that the synthesis network is able to produce meaningful results even though it receives input only through the styles that control the AdaIN operations.

생성기의 속성을 연구하기 전에 재설계가 이미지 품질을 손상시키지 않지만 실제로는 상당히 향상된다는 것을 실험적으로 보여줍니다. 표 1은 CELEBA-HQ 및 새로운 FFHQ 데이터 세트(부록 A)의 다양한 발전기 아키텍처에 대한 Frechet 개시 거리(FID)를 제공합니다. 다른 데이터 세트에 대한 결과는 부록 E에 나와 있습니다. 기본 구성(A)은 Karras 등의 Progressive GAN 설정입니다. [30], 달리 명시된 경우를 제외하고 네트워크와 모든 하이퍼파라미터를 상속합니다. 먼저 쌍선형 업/다운샘플링 작업, 더 긴 훈련 및 조정된 하이퍼파라미터를 사용하여 개선된 기준선(B)으로 전환합니다. 훈련 설정 및 하이퍼파라미터에 대한 자세한 설명은 부록 C에 포함되어 있습니다. 그런 다음 매핑 네트워크와 AdaIN 작업(C)을 추가하여 이 새로운 기준선을 더욱 개선하고 네트워크가 더 이상 잠재 코드를 첫 번째 컨볼루션 레이어 따라서 기존 입력 레이어를 제거하고 학습된 4 × 4 × 512 상수 텐서(D)에서 이미지 합성을 시작하여 아키텍처를 단순화합니다. 우리는 합성 네트워크가 AdaIN 작업을 제어하는 ​​스타일을 통해서만 입력을 수신함에도 의미 있는 결과를 생성할 수 있다는 사실을 발견했습니다.

Finally, we introduce the noise inputs (E) that improve the results further, as well as novel mixing regularization (F) that decorrelates neighboring styles and enables more finegrained control over the generated imagery (Section 3.1).마지막으로 결과를 더욱 향상시키는 노이즈 입력(E)과 인접 스타일을 상관 관계를 해제하고 생성된 이미지를 보다 세밀하게 제어할 수 있는 새로운 혼합 정규화(F)를 소개합니다(섹션 3.1).

We evaluate our methods using two different loss functions: for CELEBA-HQ we rely on WGAN-GP [24], while FFHQ uses WGAN-GP for configuration A and nonsaturating loss [22] with R1 regularization [44, 51, 14] for configurations B–F. We found these choices to give the best results. Our contributions do not modify the loss function.

우리는 두 가지 다른 손실 함수를 사용하여 방법을 평가합니다. CELEBA-HQ의 경우 WGAN-GP [24]에 의존하는 반면 FFHQ는 구성 A에 WGAN-GP를 사용하고 R1 정규화 [44, 51, 14]에 대해 비포화 손실 [22]을 사용합니다. 구성 B–F. 최상의 결과를 제공하기 위해 이러한 선택을 찾았습니다. 우리의 기여는 손실 함수를 수정하지 않습니다.

텍스트, 가장, 사람, 그룹이(가) 표시된 사진

자동 생성된 설명

Figure 2. Uncurated set of images produced by our style-based generator (config F) with the FFHQ dataset. Here we used a variation of the truncation trick [42, 5, 34] with for resolutions . Please see the accompanying video for more results.

그림 2. FFHQ 데이터 세트를 사용하여 스타일 기반 생성기(구성 F)에서 생성된 선별되지 않은 이미지 세트. 여기서 우리는 4^2-32^2 해상도에 대해 인 절단 트릭의 변형을 사용했습니다. 더 많은 결과를 보려면 첨부된 비디오를 참조하십시오.

We observe that the style-based generator (E) improves FIDs quite significantly over the traditional generator (B), almost 20%, corroborating the large-scale ImageNet measurements made in parallel work [6, 5]. Figure 2 shows an uncurated set of novel images generated from the FFHQ dataset using our generator. As confirmed by the FIDs, the average quality is high, and even accessories such as eyeglasses and hats get successfully synthesized. For this figure, we avoided sampling from the extreme regions of W using the so-called truncation trick [42, 5, 34] — Appendix B details how the trick can be performed in W instead of Z. Note that our generator allows applying the truncation selectively to low resolutions only, so that highresolution details are not affected.

스타일 기반 생성기(E)가 기존 생성기(B)에 비해 FID를 거의 20%로 상당히 개선하여 병렬 작업에서 수행된 대규모 ImageNet 측정을 뒷받침하는 것을 관찰했습니다[6, 5]. 그림 2는 생성기를 사용하여 FFHQ 데이터 세트에서 생성된 큐레이트되지 않은 새로운 이미지 세트를 보여줍니다. FID에서 확인했듯이 평균 품질이 높고 안경, 모자와 같은 액세서리도 성공적으로 합성됩니다. 이 그림의 경우 소위 절단 트릭을 사용하여 W의 극단 영역에서 샘플링을 피했습니다. [42, 5, 34] — 부록 B에서는 Z 대신 W에서 트릭을 수행하는 방법을 자세히 설명합니다. 생성기는 고해상도 세부 사항이 영향을 받지 않도록 저해상도에서만 선택적으로 잘립니다.

All FIDs in this paper are computed without the truncation trick, and we only use it for illustrative purposes in Figure 2 and the video. All images are generated in resolution.이 문서의 모든 FID는 자르기 트릭 없이 계산되며 그림 2와 비디오에서 설명 목적으로만 사용합니다. 모든 이미지는 해상도로 생성됩니다.

**2.2. Prior art**

improving the discriminator by, e.g., using multiple discriminators [18, 47, 11], multiresolution discrimination [60, 55], or self-attention [63]. The work on generator side has mostly focused on the exact distribution in the input latent space [5] or shaping the input latent space via Gaussian mixture models [4], clustering [48], or encouraging convexity [52].예를 들어, 다중 판별자[18, 47, 11], 다중 해상도 판별[60, 55] 또는 자기 주의를 사용하여 판별자를 개선합니다. 생성기 측의 작업은 입력 잠재 공간의 정확한 분포[5] 또는 가우시안 혼합 모델, 클러스터링 또는 볼록성 장려를 통한 입력 잠재 공간의 형성에 주로 초점을 맞췄습니다.

Recent conditional generators feed the class identifier through a separate embedding network to a large number of layers in the generator [46], while the latent is still provided though the input layer. A few authors have considered feeding parts of the latent code to multiple generator layers [9, 5]. In parallel work, Chen et al. [6] “self modulate” the generator using AdaINs, similarly to our work, but do not consider an intermediate latent space or noise inputs.

최근의 조건부 생성기는 별도의 임베딩 네트워크를 통해 생성기의 많은 계층에 클래스 식별자를 제공하는 반면, 잠재성은 여전히 ​​입력 계층을 통해 제공됩니다. 몇몇 작성자는 잠재 코드의 일부를 여러 생성기 계층에 제공하는 것을 고려했습니다[9, 5]. 병행 작업에서 Chen et al. 우리 작업과 유사하게 AdaIN을 사용하여 발전기를 "자체 변조"하지만 중간 잠재 공간이나 노이즈 입력을 고려하지 않습니다.

**3. Properties of the style-based generator**

Our generator architecture makes it possible to control the image synthesis via scale-specific modifications to the styles. We can view the mapping network and affine transformations as a way to draw samples for each style from a learned distribution, and the synthesis network as a way to generate a novel image based on a collection of styles. The effects of each style are localized in the network, i.e., modifying a specific subset of the styles can be expected to affect only certain aspects of the image.우리의 생성기 아키텍처는 스타일에 대한 스케일별 수정을 통해 이미지 합성을 제어하는 ​​것을 가능하게 합니다. 매핑 네트워크와 아핀 변환은 학습된 분포에서 각 스타일에 대한 샘플을 추출하는 방법으로 볼 수 있고 합성 네트워크는 스타일 컬렉션을 기반으로 새로운 이미지를 생성하는 방법으로 볼 수 있습니다. 각 스타일의 효과는 네트워크에 국한됩니다. 즉, 스타일의 특정 하위 집합을 수정하면 이미지의 특정 측면에만 영향을 줄 수 있습니다.

To see the reason for this localization, let us consider how the AdaIN operation (Eq. 1) first normalizes each channel to zero mean and unit variance, and only then applies scales and biases based on the style. The new per-channel statistics, as dictated by the style, modify the relative importance of features for the subsequent convolution operation, but they do not depend on the original statistics because of the normalization. Thus each style controls only one convolution before being overridden by the next AdaIN operation.

이 지역화의 이유를 보기 위해 AdaIN 연산(Eq. 1)이 먼저 각 채널을 0의 평균과 단위 분산으로 정규화한 다음 스타일에 따라 척도와 편향을 적용하는 방법을 살펴보겠습니다. 스타일에 따라 새로운 채널별 통계는 후속 컨볼루션 작업에 대한 기능의 상대적 중요도를 수정하지만 정규화 때문에 원래 통계에 의존하지 않습니다. 따라서 각 스타일은 다음 AdaIN 작업에 의해 재정의되기 전에 하나의 회선만 제어합니다.

**3.1. Style mixing**

To further encourage the styles to localize, we employ mixing regularization, where a given percentage of images are generated using two random latent codes instead of one during training. When generating such an image, we simply switch from one latent code to another — an operation we refer to as style mixing— at a randomly selected point in the synthesis network. To be specific, we run two latent codes through the mapping network, and have the corresponding control the styles so that applies before the crossover point and after it. This regularization technique prevents the network from assuming that adjacent styles are correlated.

스타일의 현지화를 더욱 장려하기 위해 혼합 정규화를 사용합니다. 여기서 주어진 비율의 이미지는 훈련 중에 하나가 아닌 두 개의 임의의 잠재 코드를 사용하여 생성됩니다. 이러한 이미지를 생성할 때 합성 네트워크에서 무작위로 선택된 지점에서 하나의 잠재 코드에서 다른 잠재 코드로 전환하기만 하면 됩니다(스타일 혼합이라고 하는 작업). 구체적으로, 매핑 네트워크를 통해 두 개의 잠재 코드 를 실행하고 해당 가 스타일을 제어하여 이 교차 지점 이전에 적용되고 가 교차 지점 이후에 적용되도록 합니다. 이 정규화 기술은 네트워크가 인접한 스타일이 상관관계가 있다고 가정하지 않도록 합니다.

가장, 사람, 쥐고있는, 그룹이(가) 표시된 사진

자동 생성된 설명

Figure 3. Two sets of images were generated from their respective latent codes (sources A and B); the rest of the images were generated by copying a specified subset of styles from source B and taking the rest from source A. Copying the styles corresponding to coarse spatial resolutions brings high-level aspects such as pose, general hair style, face shape, and eyeglasses from source B, while all colors (eyes, hair, lighting) and finer facial features resemble A. If we instead copy the styles of middle resolutions from B, we inherit smaller scale facial features, hair style, eyes open/closed from B, while the pose, general face shape, and eyeglasses from A are preserved.Finally, copying the fine styles from B brings mainly the color scheme and microstructure.그림 3. 각각의 잠재 코드(소스 A 및 B)에서 두 세트의 이미지가 생성되었습니다. 나머지 이미지는 소스 B에서 지정된 스타일 하위 집합을 복사하고 나머지는 소스 A에서 가져옴으로써 생성되었습니다. 거친 공간 해상도 에 해당하는 스타일을 복사하면 다음과 같은 고수준 측면을 가져옵니다. 포즈, 일반적인 헤어 스타일, 얼굴 모양 및 안경은 소스 B의 반면 모든 색상(눈, 머리카락, 조명) 및 미세한 얼굴 특징은 A와 비슷합니다. 대신 중간 해상도의 스타일을 복사하면 B에서 작은 스케일의 얼굴 특징, 머리 스타일, 눈 뜨거나 감고 있는 눈을 B에서 상속하고 포즈, 일반적인 얼굴 모양 및 안경을 A에서 유지합니다. 마지막으로 미세 스타일을 복사합니다B에서 주로 색 구성표와 미세 구조를 가져옵니다.

테이블이(가) 표시된 사진

자동 생성된 설명

Table 2. FIDs in FFHQ for networks trained by enabling the mixing regularization for different percentage of training examples. Here we stress test the trained networks by randomizing 1 . . . 4 latents and the crossover points between them. Mixing regularization improves the tolerance to these adverse operations significantly. Labels E and F refer to the configurations in Table 1.

표 2. 다양한 비율의 훈련 예시에 대해 혼합 정규화를 활성화하여 훈련된 네트워크에 대한 FFHQ의 FID 여기서 우리는 1을 무작위화하여 훈련된 네트워크를 스트레스 테스트합니다. . . 4개의 잠재성과 그들 사이의 교차점. 혼합 정규화는 이러한 불리한 작업에 대한 내성을 크게 향상시킵니다. 레이블 E 및 F는 표 1의 구성을 나타냅니다.

텍스트, 사람이(가) 표시된 사진

자동 생성된 설명

Figure 4. Examples of stochastic variation. (a) Two generated images. (b) Zoom-in with different realizations of input noise. While the overall appearance is almost identical, individual hairs are placed very differently. (c) Standard deviation of each pixel over 100 different realizations, highlighting which parts of the images are affected by the noise. The main areas are the hair, silhouettes, and parts of background, but there is also interesting stochastic variation in the eye reflections. Global aspects such as identity and pose are unaffected by stochastic variation.

그림 4. 확률적 변동의 예. (a) 두 개의 생성된 이미지. (b) 입력 노이즈의 다른 구현으로 확대. 전체 모양은 거의 동일하지만 개별 머리카락은 매우 다르게 배치됩니다. (c) 노이즈의 영향을 받는 이미지 부분을 강조 표시하는 100가지 다른 구현에 대한 각 픽셀의 표준 편차. 주요 영역은 머리카락, 실루엣 및 배경 부분이지만 눈 반사에도 흥미로운 확률론적 변화가 있습니다. 정체성 및 포즈와 같은 전역적 측면은 확률적 변동의 영향을 받지 않습니다.

Table 2 shows how enabling mixing regularization during training improves the localization considerably, indicated by improved FIDs in scenarios where multiple latents are mixed at test time. Figure 3 presents examples of images synthesized by mixing two latent codes at various scales. We can see that each subset of styles controls meaningful high-level attributes of the image.

표 2는 훈련 중에 혼합 정규화를 활성화하면 테스트 시간에 여러 잠재성이 혼합되는 시나리오에서 개선된 FID로 표시되는 현지화를 상당히 개선하는 방법을 보여줍니다. 그림 3은 두 개의 잠재 코드를 다양한 규모로 혼합하여 합성한 이미지의 예를 보여줍니다. 스타일의 각 하위 집합이 이미지의 의미 있는 상위 수준 속성을 제어한다는 것을 알 수 있습니다.

**3.2. Stochastic variation**

There are many aspects in human portraits that can be regarded as stochastic, such as the exact placement of hairs, stubble, freckles, or skin pores. Any of these can be randomized without affecting our perception of the image as long as they follow the correct distribution.

사람의 초상화에는 머리카락, 수염, 주근깨, 피부 모공의 정확한 위치와 같이 확률론적으로 간주될 수 있는 많은 측면이 있습니다. 올바른 분포를 따르는 한 이미지에 대한 우리의 인식에 영향을 미치지 않고 이들 중 어떤 것도 무작위화할 수 있습니다.

사람, 가장, 젊은, 그룹이(가) 표시된 사진

자동 생성된 설명

Figure 5. Effect of noise inputs at different layers of our generator. (a) Noise is applied to all layers. (b) No noise. (c) Noise in fine layers only (d) Noise in coarse layers only . We can see that the artificial omission of noise leads to featureless “painterly” look. Coarse noise causes large-scale curling of hair and appearance of larger background features, while the fine noise brings out the finer curls of hair, finer background detail, and skin pores.

그림 5. 다양한 생성기 레이어에서 노이즈 입력의 영향. (a) 모든 레이어에 노이즈가 적용됩니다. (b) 소음이 없습니다. (c) 미세 레이어에만 있는 노이즈 (d) 거친 레이어에만 있는 노이즈. 노이즈의 인위적인 생략이 특징 없는 "회화적인" 모습으로 이어짐을 알 수 있습니다. 거친 노이즈는 머리카락의 대규모 컬링 및 더 큰 배경 특징을 유발하는 반면, 미세 노이즈는 머리카락의 더 미세한 컬, 더 미세한 배경 디테일 및 피부 모공을 가져옵니다.

Let us consider how a traditional generator implements stochastic variation. Given that the only input to the network is through the input layer, the network needs to invent a way to generate spatially-varying pseudorandom numbers from earlier activations whenever they are needed. This consumes network capacity and hiding the periodicity of generated signal is difficult — and not always successful, as evidenced by commonly seen repetitive patterns in generated images. Our architecture sidesteps these issues altogether by adding per-pixel noise after each convolution.

기존 생성기가 확률적 변동을 구현하는 방법을 살펴보겠습니다. 네트워크에 대한 유일한 입력이 입력 계층을 통한다는 점을 감안할 때 네트워크는 필요할 때마다 이전 활성화에서 공간적으로 변화하는 의사 난수를 생성하는 방법을 발명해야 합니다. 이것은 네트워크 용량을 소모하고 생성된 신호의 주기성을 숨기는 것은 어렵습니다. 생성된 이미지에서 흔히 볼 수 있는 반복적인 패턴에서 알 수 있듯이 항상 성공적인 것은 아닙니다. 우리의 아키텍처는 각 컨볼루션 후에 픽셀당 노이즈를 추가하여 이러한 문제를 모두 회피합니다.

Figure 4 shows stochastic realizations of the same underlying image, produced using our generator with different noise realizations. We can see that the noise affects only the stochastic aspects, leaving the overall composition and high-level aspects such as identity intact. Figure 5 further illustrates the effect of applying stochastic variation to different subsets of layers. Since these effects are best seen in animation, please consult the accompanying video for a demonstration of how changing the noise input of one layer leads to stochastic variation at a matching scale.

그림 4는 다른 노이즈 구현으로 생성기를 사용하여 생성된 동일한 기본 이미지의 확률적 구현을 ​​보여줍니다. 노이즈가 확률적 측면에만 영향을 미치고 전체 구성과 동일성과 같은 상위 수준 측면은 그대로 유지됨을 알 수 있습니다. 그림 5는 확률론적 변화를 레이어의 다른 하위 집합에 적용하는 효과를 추가로 보여줍니다. 이러한 효과는 애니메이션에서 가장 잘 나타나므로 한 레이어의 노이즈 입력을 변경하면 일치하는 규모에서 확률적 변화가 발생하는 방법에 대한 데모를 보려면 첨부된 비디오를 참조하십시오.

We find it interesting that the effect of noise appears tightly localized in the network. We hypothesize that at any point in the generator, there is pressure to introduce new content as soon as possible, and the easiest way for our network to create stochastic variation is to rely on the noise provided. A fresh set of noise is available for every layer, and thus there is no incentive to generate the stochastic effects from earlier activations, leading to a localized effect.

노이즈의 영향이 네트워크에 밀접하게 국한되어 나타나는 것이 흥미롭습니다. 우리는 생성기의 어느 지점에서나 가능한 한 빨리 새로운 콘텐츠를 도입해야 한다는 압력이 있으며, 우리 네트워크가 확률적 변동을 생성하는 가장 쉬운 방법은 제공된 노이즈에 의존하는 것이라고 가정합니다. 모든 레이어에 대해 새로운 노이즈 세트를 사용할 수 있으므로 이전 활성화에서 확률론적 효과를 생성할 인센티브가 없어 국소 효과가 발생합니다.

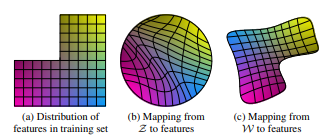


Figure 6. Illustrative example with two factors of variation (image features, e.g., masculinity and hair length). (a) An example training set where some combination (e.g., long haired males) is missing. (b) This forces the mapping from Z to image features to become curved so that the forbidden combination disappears in Z to prevent the sampling of invalid combinations. (c) The learned mapping from Z to W is able to “undo” much of the warping.

그림 6. 두 가지 변형 요인(이미지 특징, 예: 남성다움 및 머리 길이)이 있는 예시. (a) 일부 조합(예: 긴 머리 남성)이 누락된 예제 훈련 세트입니다. (b) 이것은 Z에서 이미지 특징으로의 매핑이 곡선이 되도록 하여 금지된 조합이 Z에서 사라지도록 하여 잘못된 조합의 샘플링을 방지합니다. (c) Z에서 W로의 학습된 매핑은 많은 뒤틀림을 "실행 취소"할 수 있습니다.

**3.3. Separation of global effects from stochasticity**

The previous sections as well as the accompanying video demonstrate that while changes to the style have global effects (changing pose, identity, etc.), the noise affects only inconsequential stochastic variation (differently combed hair, beard, etc.). This observation is in line with style transfer literature, where it has been established that spatially invariant statistics (Gram matrix, channel-wise mean, variance, etc.) reliably encode the style of an image [20, 39] while spatially varying features encode a specific instance.

이전 섹션과 함께 제공되는 비디오에서는 스타일 변경이 전체적 효과(포즈, 정체성 변경 등)를 갖지만 노이즈는 중요하지 않은 확률적 변화(다른 방식으로 빗질한 머리카락, 수염 등)에만 영향을 미친다는 것을 보여줍니다. 이 관찰은 공간적으로 변하지 않는 통계(Gram 행렬, 채널별 평균, 분산 등)가 이미지의 스타일을 안정적으로 인코딩하는 반면 공간적으로 변화하는 특징은 인코딩한다는 스타일 전달 문헌[20, 39]과 일치합니다. 특정 인스턴스.

In our style-based generator, the style affects the entire image because complete feature maps are scaled and biased with the same values. Therefore, global effects such as pose, lighting, or background style can be controlled coherently. Meanwhile, the noise is added independently to each pixel and is thus ideally suited for controlling stochastic variation. If the network tried to control, e.g., pose using the noise, that would lead to spatially inconsistent decisions that would then be penalized by the discriminator. Thus the network learns to use the global and local channels appropriately, without explicit guidance.

스타일 기반 생성기에서 스타일은 전체 이미지에 영향을 줍니다. 전체 기능 맵이 동일한 값으로 크기 조정되고 바이어스되기 때문입니다. 따라서 포즈, 조명 또는 배경 스타일과 같은 전역 효과를 일관되게 제어할 수 있습니다. 한편, 노이즈는 각 픽셀에 독립적으로 추가되므로 확률적 변동을 제어하는 ​​데 이상적입니다. 네트워크가 예를 들어 노이즈를 사용하여 포즈를 제어하려고 하면 공간적으로 일관되지 않은 결정이 발생하여 판별자에 의해 불이익을 받게 됩니다. 따라서 네트워크는 명시적인 지침 없이 글로벌 및 로컬 채널을 적절하게 사용하는 방법을 배웁니다.

**4. Disentanglement studies**

There are various definitions for disentanglement [54, 50, 2, 7, 19], but a common goal is a latent space that consists of linear subspaces, each of which controls one factor of variation. However, the sampling probability of each combination of factors in Z needs to match the corresponding density in the training data. As illustrated in Figure 6, this precludes the factors from being fully disentangled with typical datasets and input latent distributions.

얽힘에 대한 다양한 정의가 있지만[54, 50, 2, 7, 19], 공통 목표는 각각 하나의 변동 요인을 제어하는 ​​선형 부분 공간으로 구성된 잠재 공간입니다. 그러나 Z에 있는 각 요인 조합의 샘플링 확률은 훈련 데이터의 해당 밀도와 일치해야 합니다. 그림 6에서 볼 수 있듯이 이는 요인이 일반적인 데이터 세트 및 입력 잠재 분포와 완전히 분리되는 것을 방지합니다.

A major benefit of our generator architecture is that the intermediate latent space W does not have to support samling according to any fixed distribution; its sampling density is induced by the learned piecewise continuous mapping f(z). This mapping can be adapted to “unwarp” W so that the factors of variation become more linear. We posit that there is pressure for the generator to do so, as it should be easier to generate realistic images based on a disentangled representation than based on an entangled representation. As such, we expect the training to yield a less entangled W in an unsupervised setting, i.e., when the factors of variation are not known in advance [10, 35, 49, 8, 26, 32, 7].생성기 아키텍처의 주요 이점은 중간 잠재 공간 W가 고정 분포에 따라 샘플링을 지원할 필요가 없다는 것입니다. 샘플링 밀도는 학습된 조각별 연속 매핑 f(z)에 의해 유도됩니다. 이 매핑은 변형 요인이 보다 선형이 되도록 W를 "워프 해제"하도록 조정할 수 있습니다. 얽힌 표현을 기반으로 하는 것보다 얽힌 표현을 기반으로 하는 사실적인 이미지를 생성하는 것이 더 쉬워야 하기 때문에 생성기가 그렇게 해야 한다는 압력이 있다고 가정합니다. 따라서 우리는 훈련이 감독되지 않은 설정, 즉 변동 요인을 미리 알 수 없는 경우 덜 얽힌 W를 산출할 것으로 기대합니다[10, 35, 49, 8, 26, 32, 7].

Unfortunately the metrics recently proposed for quantifying disentanglement [26, 32, 7, 19] require an encoder network that maps input images to latent codes. These metrics are ill-suited for our purposes since our baseline GAN lacks such an encoder. While it is possible to add an extra network for this purpose [8, 12, 15], we want to avoid investing effort into a component that is not a part of the actual solution. To this end, we describe two new ways of quantifying disentanglement, neither of which requires an encoder or known factors of variation, and are therefore computable for any image dataset and generator.

불행히도 얽힘을 정량화하기 위해 최근에 제안된 메트릭[26, 32, 7, 19]은 입력 이미지를 잠재 코드에 매핑하는 인코더 네트워크를 필요로 합니다. 우리의 기준 GAN에는 그러한 인코더가 없기 때문에 이러한 메트릭은 우리의 목적에 부적합합니다. 이 목적을 위해 네트워크를 추가하는 것이 가능하지만[8, 12, 15], 실제 솔루션의 일부가 아닌 구성 요소에 노력을 투자하는 것을 피하고 싶습니다. 이를 위해 우리는 얽힘을 정량화하는 두 가지 새로운 방법을 설명합니다. 두 가지 방법 모두 인코더나 알려진 변동 요인이 필요하지 않으므로 모든 이미지 데이터 세트 및 생성기에 대해 계산할 수 없습니다.

**4.1. Perceptual path length**

As noted by Laine [37], interpolation of latent-space vectors may yield surprisingly non-linear changes in the image. For example, features that are absent in either endpoint may appear in the middle of a linear interpolation path. This is a sign that the latent space is entangled and the factors of variation are not properly separated. To quantify this effect, we can measure how drastic changes the image undergoes as we perform interpolation in the latent space. Intuitively, a less curved latent space should result in perceptually smoother transition than a highly curved latent space.

Laine이 언급한 바와 같이, 잠재 공간 벡터의 보간은 이미지에서 놀랍게도 비선형 변화를 생성할 수 있습니다. 예를 들어, 양쪽 끝점에 없는 기능은 선형 보간 경로의 중간에 나타날 수 있습니다. 이는 잠재 공간이 얽혀 있고 변동 요인이 제대로 분리되지 않았다는 신호입니다. 이 효과를 정량화하기 위해 잠재 공간에서 보간을 수행할 때 이미지가 얼마나 급격한 변화를 겪는지 측정할 수 있습니다. 직관적으로 덜 휘어진 잠재 공간은 고도로 휘어진 잠재 공간보다 지각적으로 더 부드러운 전환을 가져옵니다.

As a basis for our metric, we use a perceptually-based pairwise image distance [65] that is calculated as a weighted difference between two VGG16 [58] embeddings, where the weights are fit so that the metric agrees with human perceptual similarity judgments. If we subdivide a latent space interpolation path into linear segments, we can define the total perceptual length of this segmented path as the sum of perceptual differences over each segment, as reported by the image distance metric. A natural definition for the perceptual path length would be the limit of this sum under infinitely fine subdivision, but in practice we approximate it using a small subdivision epsilon . The average perceptual path length in latent space Z, over all possible endpoints, is therefore

메트릭의 기반으로 메트릭이 인간의 지각 유사성 판단과 일치하도록 가중치가 맞는 두 VGG16 [58] 임베딩 간의 가중치 차이로 계산되는 지각 기반 쌍별 이미지 거리[65]를 사용합니다. 잠재 공간 보간 경로를 선형 세그먼트로 세분화하면 이 분할된 경로의 총 지각 길이를 이미지 거리 메트릭으로 보고된 각 세그먼트에 대한 지각 차이의 합으로 정의할 수 있습니다. 지각 경로 길이에 대한 자연스러운 정의는 무한히 미세한 세분화에서 이 합의 한계이지만 실제로는 작은 세분화 ε ϵ= 10^(-4)를 사용하여 근사화합니다. 따라서 가능한 모든 끝점에 대한 잠재 공간 Z의 평균 지각 경로 길이는 다음과 같습니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

Where , G is the generator (i.e., g ◦f for style-based networks), and d(·, ·) evaluates the perceptual distance between the resulting images. Here slerp denotes spherical interpolation [56], which is the most appropriate way of interpolating in our normalized input latent space [61]. To concentrate on the facial features instead of background, we crop the generated images to contain only the face prior to evaluating the pairwise image metric. As the metric d is quadratic [65], we divide by . We compute the expectation by taking 100,000 samples.

여기서, G는 생성기(즉, 스타일 기반 네트워크의 경우 g ◦f)이고 d(·, ·)는 결과 간의 지각 거리를 평가합니다. 이미지. 여기에서 slerp는 구형 보간[56]을 나타내며, 이는 정규화된 입력 잠재 공간[61]에서 보간하는 가장 적절한 방법입니다. 배경 대신 얼굴 특징에 집중하기 위해 쌍별 이미지 메트릭을 평가하기 전에 얼굴만 포함하도록 생성된 이미지를 자릅니다. 메트릭 d가 2차이므로 [65] 로 나눕니다. 100,000개의 샘플을 취하여 기대치를 계산합니다.

테이블이(가) 표시된 사진

자동 생성된 설명

Table 3. Perceptual path lengths and separability scores for various generator architectures in FFHQ (lower is better). We perform the measurements in Z for the traditional network, and in W for stylebased ones. Making the network resistant to style mixing appears to distort the intermediate latent space W somewhat. We hypothesize that mixing makes it more difficult for W to efficiently encode factors of variation that span multiple scales.

표 3. FFHQ의 다양한 생성기 아키텍처에 대한 인식 경로 길이 및 분리성 점수(낮을수록 좋음). 우리는 전통적인 네트워크에 대해 Z로 측정을 수행하고 스타일 기반 네트워크에 대해 W에서 측정을 수행합니다. 스타일 믹싱에 저항하는 네트워크를 만드는 것은 중간 잠재 공간 W를 다소 왜곡시키는 것으로 보입니다. 우리는 혼합이 W가 여러 척도에 걸쳐 있는 변동 요인을 효율적으로 인코딩하는 것을 더 어렵게 만든다고 가정합니다.

Computing the average perceptual path length in W is carried out in a similar fashion:

W에서 평균 지각 경로 길이를 계산하는 것은 유사한 방식으로 수행됩니다.

텍스트이(가) 표시된 사진

자동 생성된 설명

where the only difference is that interpolation happens in W space. Because vectors in W are not normalized in any fashion, we use linear interpolation (lerp).

여기서 유일한 차이점은 보간이 W 공간에서 발생한다는 것입니다. W의 벡터는 어떤 방식으로든 정규화되지 않기 때문에 선형 보간(lerp)을 사용합니다.

Table 3 shows that this full-path length is substantially shorter for our style-based generator with noise inputs, indicating that W is perceptually more linear than Z. Yet, this measurement is in fact slightly biased in favor of the input latent space Z. If W is indeed a disentangled and “flattened” mapping of Z, it may contain regions that are not on the input manifold— and are thus badly reconstructed by the generator — even between points that are mapped from the input manifold, whereas the input latent space Z has no such regions by definition. It is therefore to be expected that if we restrict our measure to path endpoints, i.e., , we should obtain a smaller while is not affected. This is indeed what we observe in Table 3.

표 3은 이 전체 경로 길이가 노이즈 입력이 있는 스타일 기반 생성기에서 훨씬 더 짧음을 보여줍니다. 이는 W가 Z보다 지각적으로 더 선형임을 나타냅니다. 그러나 이 측정은 실제로 입력 잠재 공간 Z에 유리하게 약간 편향되어 있습니다. W가 실제로 Z의 얽혀 있고 "평탄화된" 매핑인 경우 입력 매니폴드에서 매핑된 지점 사이에서도 입력 매니폴드에 없는 영역을 포함할 수 있으므로 생성기에 의해 잘못 재구성됩니다. 공간 Z에는 정의에 따라 그러한 영역이 없습니다. 따라서 측정을 경로 끝점, 즉 t ∈{0,1}로 제한하면 가 영향을 받지 않는 동안 더 작은 를 얻어야 합니다. 이것은 실제로 우리가 표 3에서 관찰한 것입니다.

Table 4 shows how path lengths are affected by the mapping network. We see that both traditional and style-based generators benefit from having a mapping network, and additional depth generally improves the perceptual path length as well as FIDs. It is interesting that while improves in the traditional generator, becomes considerably worse, illustrating our claim that the input latent space can indeed be arbitrarily entangled in GANs.

표 4는 경로 길이가 매핑 네트워크의 영향을 받는 방식을 보여줍니다. 우리는 전통적인 생성기 및 스타일 기반 생성기 모두 매핑 네트워크를 갖는 이점이 있으며 추가 깊이는 일반적으로 FID뿐만 아니라 지각 경로 길이를 향상시킨다는 것을 알 수 있습니다. 흥미로운 점은 가 전통적인 생성기에서 향상되는 반면 는 상당히 악화되어 입력 잠재 공간이 실제로 GAN에서 임의로 얽힐 수 있다는 주장을 보여줍니다.

테이블이(가) 표시된 사진

자동 생성된 설명

Table 4. The effect of a mapping network in FFHQ. The number in method name indicates the depth of the mapping network. We see that FID, separability, and path length all benefit from having a mapping network, and this holds for both style-based and traditional generator architectures. Furthermore, a deeper mapping network generally performs better than a shallow one.

표 4. FFHQ에서 매핑 네트워크의 효과. 메서드 이름의 숫자는 매핑 네트워크의 깊이를 나타냅니다. 우리는 FID, 분리성 및 경로 길이가 모두 매핑 네트워크를 사용함으로써 이점을 얻음을 알 수 있으며 이는 스타일 기반 및 기존 생성기 아키텍처 모두에 적용됩니다. 또한 깊은 매핑 네트워크는 일반적으로 얕은 매핑 네트워크보다 성능이 좋습니다.

**4.2. Linear separability**

If a latent space is sufficiently disentangled, it should be possible to find direction vectors that consistently correspond to individual factors of variation. We propose another metric that quantifies this effect by measuring how well the latent-space points can be separated into two distinct sets via a linear hyperplane, so that each set corresponds to a specific binary attribute of the image.

잠재 공간이 충분히 얽혀 있으면 개별 변동 요인에 일관되게 대응하는 방향 벡터를 찾는 것이 가능해야 합니다. 우리는 잠재 공간 포인트가 선형 초평면을 통해 두 개의 개별 세트로 얼마나 잘 분리되어 각 세트가 이미지의 특정 이진 속성에 해당하는지 측정하여 이 효과를 정량화하는 또 다른 메트릭을 제안합니다.

In order to label the generated images, we train auxiliary classification networks for a number of binary attributes, e.g., to distinguish male and female faces. In our tests, the classifiers had the same architecture as the discriminator we use (i.e., same as in [30]), and were trained using the CELEBA-HQ dataset that retains the 40 attributes available in the original CelebA dataset. To measure the separability of one attribute, we generate 200,000 images with z ∼ P(z) and classify them using the auxiliary classification network. We then sort the samples according to classifier confidence and remove the least confident half, yielding 100,000 labeled latent-space vectors.

생성된 이미지에 레이블을 지정하기 위해 예를 들어 남성과 여성의 얼굴을 구별하기 위해 여러 이진 속성에 대한 보조 분류 네트워크를 훈련합니다. 우리의 테스트에서 분류기는 우리가 사용하는 판별기와 동일한 아키텍처를 갖고(즉, [30]에서와 동일), 원래 CelebA 데이터 세트에서 사용 가능한 40개의 속성을 유지하는 CELEBA-HQ 데이터 세트를 사용하여 훈련되었습니다. 한 속성의 분리 가능성을 측정하기 위해 z ∼ P(z)를 갖는 200,000개의 이미지를 생성하고 보조 분류 네트워크를 사용하여 분류합니다. 그런 다음 분류기 신뢰도에 따라 샘플을 정렬하고 가장 신뢰도가 낮은 절반을 제거하여 100,000개의 레이블이 지정된 잠재 공간 벡터를 생성합니다.

For each attribute, we fit a linear SVM to predict the label based on the latent-space point —z for traditional and w for style-based — and classify the points by this plane. We then compute the conditional entropy H(Y |X) where X are the classes predicted by the SVM and Y are the classes determined by the pre-trained classifier. This tells how much additional information is required to determine the true class of a sample, given that we know on which side of the hyperplane it lies. A low value suggests consistent latent space directions for the corresponding factor(s) of variation.

각 속성에 대해 선형 SVM을 피팅하여 잠재 공간 점(전통적인 경우 z, 스타일 기반인 경우 w)을 기반으로 레이블을 예측하고 이 평면으로 점을 분류합니다. 그런 다음 조건부 엔트로피 H(Y |X)를 계산합니다. 여기서 X는 SVM에 의해 예측된 클래스이고 Y는 사전 훈련된 분류기에 의해 결정된 클래스입니다. 이것은 샘플이 초평면의 측면에 놓여 있다는 것을 알고 있는 경우 샘플의 실제 클래스를 결정하는 데 필요한 추가 정보의 양을 알려줍니다. 낮은 값은 해당 변동 요인에 대한 일관된 잠재 공간 방향을 나타냅니다.

We calculate the final separability score as , where enumerates the 40 attributes. Similar to the inception score [53], the exponentiation brings the values from logarithmic to linear domain so that they are easier to compare.

최종 분리성 점수를 로 계산합니다. 여기서 i는 40개의 속성을 열거합니다. 시작 점수[53]와 유사하게 지수화는 값을 대수에서 선형 영역으로 가져와 비교하기 쉽도록 합니다.

Tables 3 and 4 show that W is consistently better separable than Z, suggesting a less entangled representation.

표 3과 4는 W가 Z보다 일관되게 더 잘 분리 가능함을 보여주며 덜 얽힌 표현을 제안합니다.

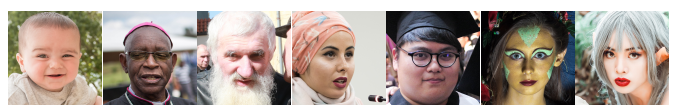


Figure 7. The FFHQ dataset offers a lot of variety in terms of age, ethnicity, viewpoint, lighting, and image background.

그림 7. FFHQ 데이터 세트는 연령, 민족, 관점, 조명 및 이미지 배경 측면에서 매우 다양합니다.

Furthermore, increasing the depth of the mapping network improves both image quality and separability in W, which is in line with the hypothesis that the synthesis network inherently favors a disentangled input representation. Interestingly, adding a mapping network in front of a traditional generator results in severe loss of separability in Z but improves the situation in the intermediate latent space W, and the FID improves as well. This shows that even the traditional generator architecture performs better when we introduce an intermediate latent space that does not have to follow the distribution of the training data.

또한 매핑 네트워크의 깊이를 증가시키면 W의 이미지 품질과 분리 가능성이 모두 향상되며, 이는 합성 네트워크가 본질적으로 얽힌 입력 표현을 선호한다는 가설과 일치합니다. 흥미롭게도 기존 생성기 앞에 매핑 네트워크를 추가하면 Z의 분리도가 크게 손실되지만 중간 잠재 공간 W의 상황이 개선되고 FID도 개선됩니다. 이것은 훈련 데이터의 분포를 따를 필요가 없는 중간 잠재 공간을 도입할 때 전통적인 생성기 아키텍처조차도 더 나은 성능을 보인다는 것을 보여줍니다.

**5. Conclusion**

Conclusion Based on both our results and parallel work by Chen et al. [6], it is becoming clear that the traditional GAN generator architecture is in every way inferior to a style-based design. This is true in terms of established quality metrics, and we further believe that our investigations to the separation of high-level attributes and stochastic effects, as well as the linearity of the intermediate latent space will prove fruitful in improving the understanding and controllability of GAN synthesis.

결론은 우리의 결과와 Chen et al.의 병렬 작업을 모두 기반으로 합니다. [6]기존의 GAN 생성기 아키텍처가 모든 면에서 스타일 기반 설계보다 열등하다는 것이 분명해지고 있습니다. 이것은 확립된 품질 지표의 측면에서 사실이며, 우리는 또한 높은 수준의 속성과 확률적 효과의 분리와 중간 잠복 공간의 선형성에 대한 조사가 GAN 합성의 이해와 제어 가능성을 개선하는 데 효과적일 것이라고 믿습니다.

We note that our average path length metric could easily be used as a regularizer during training, and perhaps some variant of the linear separability metric could act as one, too. In general, we expect that methods for directly shaping the intermediate latent space during training will provide interesting avenues for future work.

평균 경로 길이 메트릭은 훈련 중에 정규화로 쉽게 사용될 수 있으며 선형 분리성 메트릭의 일부 변형도 하나로 작동할 수 있습니다. 일반적으로 훈련 중 중간 잠재 공간을 직접 형성하는 방법이 향후 작업에 흥미로운 길을 제공할 것으로 기대합니다.

**6. Acknowledgements**

We thank Jaakko Lehtinen, David Luebke, and Tuomas Kynka¨anniemi for in-depth discussions and helpful com- ¨ ments; Janne Hellsten, Tero Kuosmanen, and Pekka Janis ¨ for compute infrastructure and help with the code release.

깊이 있는 토론과 유용한 논평을 해주신 Jaakko Lehtinen, David Luebke, Tuomas Kynka¨anniemi에게 감사드립니다. Janne Hellsten, Tero Kuosmanen 및 Pekka Janis ¨ 컴퓨팅 인프라 및 코드 릴리스 지원.