X-GAN: Improving Generative Adversarial Networks with ConveX Combinations

Oliver Blum, Biagio Brattoli, and Björn Ommer

Heidelberg University, HCI / IWR, Germany
o.blum90@gmail.com
biagio.brattoli@iwr.uni-heidelberg.de
ommer@uni-heidelberg.de

Abstract. Recent neural architectures for image generation are capable of producing photo-realistic results but the distributions of real and faked images still differ. While the lack of a structured latent representation for GANs results in mode collapse, VAEs enforce a prior to the latent space that leads to an unnatural representation of the underlying real distribution. We introduce a method that preserves the natural structure of the latent manifold. By utilizing neighboring relations within the set of discrete real samples, we reproduce the full continuous latent manifold. We propose a novel image generation network X-GAN that creates latent input vectors from random convex combinations of adjacent real samples. This way we ensure a structured and natural latent space by not requiring prior assumptions. In our experiments, we show that our model outperforms recent approaches in terms of the missing mode problem while maintaining a high image quality.

1 Introduction

One big challenge in computer vision is the understanding of images and their manifold. Other than discriminative tasks like image/action classification[30, 6] and pose estimation[5, 2], generative approaches provide a much deeper understanding of the nature of the given data. Therefore, the generation of synthetic data which resembles the real data distribution is major in modern research.

The recent trend of deep generative networks in computer vision led to several models capable to produce photo-realistic images [1, 8, 12, 25, 23]. These approaches are mostly based on Generative Adversarial Networks (GAN) [13] and Variational Auto Encoders [27] (VAE). GANs are based on an adversarial game between generator and discriminator, which has been proven successful in generating realistic images. However, GANs are known to suffer from the missing mode problem [28, 13]: If the generator becomes strong in producing images of certain modes the adversarial loss suppresses the generation of more challenging samples. In contrast to the real distribution, for GANs dense areas in the fake data distribution can be observed where images are easily generated, while areas with more difficult samples are not well represented. VAEs are trained to encode and decode images. They are often applied to tackle mode collapse. To fill the

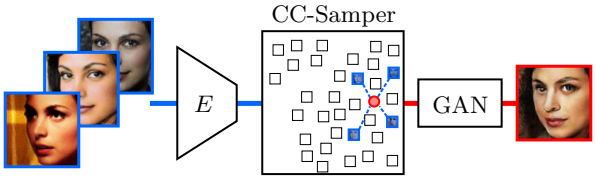

Fig. 1. The principle X-GAN setup. For the creation of latent variables that are used as input of the GAN module, the X-GAN utilities convex combinations of adjacent discrete samples to create new samples from the continuous manifold. This ensures a natural representation of the latent space without enforcing a prior to the distribution.

space in between the encoded real data representations, VAEs sample from a prior. A Kullback-Leibler-divergence loss [15] (KL-divergence loss) pushes the latent vectors to this prior, that is assumed to match the underlying real distribution. However, this is not necessarily the case: The impact of the overpowering effect of the regularization term (the prior) on VAE training is often referenced in the VAE literature [4, 16, 18, 9, 10]. To address the issues of mode collapse and overpowering effect a combination of VAE and GANs is proposed in several publications [7, 1, 20]. However, all of the proposed methods either still involve a KL-divergence regularization [1, 20] or perform the latent vector generation for the GAN training independent of the AE training such that missing modes cannot be fully solved [7]. Other methods rely on the strategy of better representing the latent space [3, 10]. However, they still rely on the idea of fitting prior distributions. Currently, the challenge of representing the complex manifold of real images has not yet been addressed satisfactorily.

In this paper, we approach the problem from a different perspective by avoiding any assumption about the probability distribution function but sampling from the entire continuous latent manifold of the given dataset such that no modes are missed. In our model, samples are drawn from the full continuous latent distribution using a non-parametric approach based on a convex combination of adjacent points in the encoding space, without requiring an explicit definition of the latent density function. As this novel approach of drawing latent samples with conveX combinations is our main contribution, we call our model X-GAN.

The dense representation of the latent manifold, provided by means of the convex combinations, makes it possible to condition the generator directly using the position in the latent vectors. Based on this we propose a new method for conditioning our network.

In our experiments, we show that our model outperforms recent methods in terms of missing mode and image realism. In the ablation studies, we substantiate the benefits of the convex combination and the conditioning strategy by illustrating the drawbacks of replacing them with alternative approaches.

2 Related Work

In this section, we introduce several deep generative architectures that attempt to tackle the known issues of mode collapse and overpowering effect. We show that solving both issues with one model is still an unsolved problem.

Variational Auto-Encoder VAEs rely on the idea of compressing their input with an encoder-network and decoding this latent representation with a generator-network. The network then is trained with an image reconstruction loss. An additional KL-divergence loss pushes the latent representation to resemble a prior distribution, from which samples can be drawn to generate new images. This approach makes the VAE robust against mode collapse. However, the underlying true distribution often is poorly represented by the chosen prior [4, 16, 18, 9, 10]. Moreover, VAEs include a distance metric describing the difference between original and generated image. An imperfect choice of this distance metric entails unnatural artifacts in the faked images [11].

Generative Adversarial Network GANs come with a convenient solution for these problems by learning the loss by an adversarial game between generator and discriminator, without the requirement of enforcing a prior. However, a missing control over the latent space easily results in mode collapse [13, 28] as described in section 1.

GAN+VAE Several recent approaches intend to tackle the previously mentioned issues by combining VAEs and GANs [20, 1, 7]. The VAE/GAN trains a discriminator along with a VAE. As distance metric, for the autoencoder loss they utilize features provided by the discriminator instead of the pixel-wise distance metrics. This provides a way to learn a distance metric without the drawbacks of the GAN training. The cVAE uses a similar approach which also includes class conditioning. While the the VAE/GAN and the cVAE-GAN utilize a prior to draw latent vectors from the continuous distribution, in our model, we use convex combinations. This comes with the advantage of avoiding the overpowering effect.

The Mode-Regularized-GAN [7] (MD-GAN) suggest training a GAN along with an autoencoder without using the KL-divergence loss. In the training procedure, the GAN and the autoencoder are trained alternately with a shared generator network. The MD-GAN attempts to structure the latent space with the autoencoder such that all modes are represented in the latent space. The latent vectors of the GAN module are sampled from a distribution that is independent of the encoded image distribution of the autoencoder. Other than the MD-GAN the encoder of the X-GAN is responsible to define the embedding manifold within the latent space. The latent vectors that are passed to the generator are sampled from this manifold by applying the convex combination strategy. As the autoencoder and the GAN sample from the same shared distribution, the autoencoder also prevents mode collapse for the GAN training.

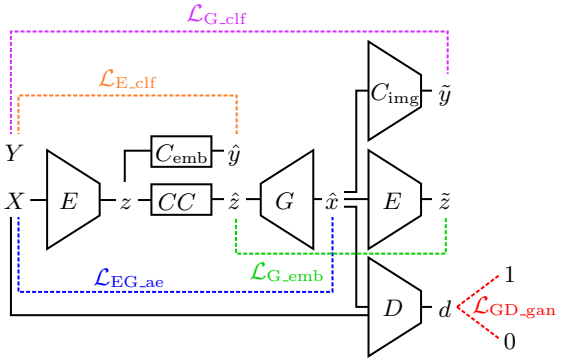

Fig. 2. This figure visualizes the X-GAN architecture. During testing, the X-GAN encodes images $x \in X$ using the encoder E into the latent space $z \in Z$. For the creation of latent variables $\hat{z}$, the X-GAN utilities convex combinations of adjacent real samples. This task is performed by the convex combination sampler CC. The created latent variables are passed on to the generator G that generates synthetic images $\hat{x}$. During the training phase, the images are passed further to a discriminator D, the previously mentioned encoder E and the classification network C_{img}. The network C_{emb} enforces class information into the sample by predicting the label $\hat{y}$ of the latent vectors z. The colored dashed lines that connect two parameters stand for the loss functions with which the X-GAN is trained.

Latent Space Based Approaches Apart from combining VAE and GAN, many recent papers focus on optimizing the latent representation.

To better model the single modes of real data distributions, Dilokthanakul et al. [10] utilize a Gaussian mixture model instead of a single Gaussian for the KL-divergence loss. This approach mitigates the issue of overpowering.

The Generative Latent Optimization framework [3] (GLO) represents a slim generative solution that only relies on one generative network. The latent variables themselves are optimized along with the generator by an image-reconstruction-loss. This ensures a solid structured latent space.

The Plug & Play Generative Network (PPGN) [25] involves a network to sample class conditioned latent variables. This ensures that the network generates samples of all classes. This strategy reduces the severity of the missing mode problem as it ensures that all classes are represented in the faked images.

Similarly to the above introduced models, the X-GAN also aims at improving the latent representation. As we focus on side-stepping the overpowering effect, other than the Dilokthanakul et al. [10] and Bojanowski et al. [3] we do not utilize any prior for the latent distribution. Like Nguyen et al. [25] we intend to support the latent structuring by inducing labels without the requirement of the concatenation of sparse label vectors to the dense layer information. In contrast to the PPGN we do not train a conditioning network, but directly encode the

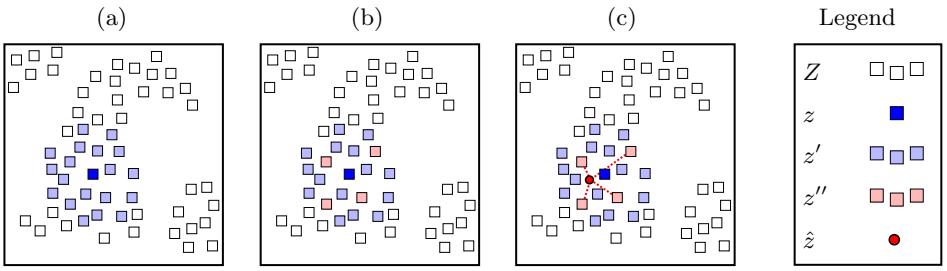

Fig. 3. The three steps for the convex combination strategy. (a) Chose one latent vector $z \in Z$ (dark blue) from the full latent distribution Z and get z', the m nearest neighbors of z in the latent space. (b) Randomly select k samples $z'' \subset z'$. (c) Get $\hat{z}$ as a randomly weighted mean of z'' (see equation 1).

conditioning information in the position of the latent vector in the embedding manifold. This supports the encoder finding a solid structure for the latent space which forms the basis for our convex combination approach.

3 Method

In our model, we use the convex combination strategy to ensure that samples from the entire manifold are created. This approach has no requirement of assuming a prior distribution for the latent space. Thus, our model does not suffer from the overpowering effect. In this section, we will first describe our novel approach to directly sample from the data distribution. Afterwards, we illustrate the model architecture and the training procedure by explaining the utilized loss functions. Finally, we explain our conditioning strategy in the resulting smooth latent distribution.

3.1 Convex Combination Strategy

The objective of the convex combination strategy is to provide the generator with samples from the underlying continuous manifold. As the representation of photo-realistic images is a high dimensional, complex manifold, a powerful strategy of drawing samples is required. However, common methods like Kernel-Density-Estimation or Gaussian-Mixture-Models fail due to the curse of dimensionality: Stone [31] proved that the complexity of any estimator grows exponentially with the dimensionality of the density function. However, drawing samples from a distribution does not necessarily require the explicit definition of such a function. A natural approach, in this case, is utilizing neighborhood relations: We assume that, the space can be approximated as convex for a small neighborhood, such that we can perform a local linear estimation of the manifold. Thus,

Algorithm 1 Training algorithm for X-GAN

Require: $\{X, Y\}$: training set (images, labels). $\{\theta_E, \theta_G, \theta_D, \theta_{C_{\text{emb}}}, \theta_{C_{\text{img}}}\}$: initial parameters for $E, G, D, C_{\text{emb}}, C_{\text{img}}$ (C_{img} pre-trained on $\{X, Y\}$ and fixed during the training). λ: parameter balancing GAN loss compared to the other loss functions. $f(\cdot)$: image feature extracting function.

while θ_G has not converged **do**
- $Z \leftarrow E(X)$
- Sample $\{x, y, z\} \sim \{X, Y, Z\}$;
- $\hat{z} \leftarrow \text{CC}^Z_{k,m}(z)$ (see equation 1) with $\hat{z} \sim P_{\hat{z}}$
- $\mathcal{L}_{\text{E\_clf}} \leftarrow -\mathbb{E}_{x \sim P_r}[y \log C_{\text{emb}}(E(x))]$
- $\mathcal{L}_{\text{EG\_ae}} \leftarrow |f(x_i) - f(G(E(x)))|$
- $\mathcal{L}_{\text{GD\_gan}} \leftarrow -(\mathbb{E}_{x \sim p_r}[log(D(x))] + \mathbb{E}_{\hat{z} \sim p_{\hat{z}}}[log(1 - D(G(\hat{z})))])$
- $\mathcal{L}_{\text{G\_emb}} \leftarrow |\hat{z} - E(G(\hat{z}))|^2$
- $\mathcal{L}_{\text{G\_clf}} \leftarrow -\mathbb{E}_{\hat{z} \sim P_{\hat{z}}}[y\ log(C_{\text{img}}(G(\hat{z})))]$
- $\theta_E \xleftarrow{\text{adam}} -\nabla_{\theta_E}(\mathcal{L}_{\text{E\_clf}} + \mathcal{L}_{\text{EG\_ae}})$
- $\theta_G \xleftarrow{\text{adam}} -\nabla_{\theta_G}(\mathcal{L}_{\text{EG\_ae}} - \lambda\mathcal{L}_{\text{GD\_gan}} + \mathcal{L}_{\text{G\_emb}} + \mathcal{L}_{\text{G\_clf}})$
- $\theta_D \xleftarrow{\text{adam}} -\nabla_{\theta_D}(\lambda\mathcal{L}_{\text{GD\_gan}})$
- $\theta_{C_{\text{emb}}} \xleftarrow{\text{adam}} -\nabla_{\theta_{C_{\text{emb}}}}(\mathcal{L}_{\text{E\_clf}})$

we propose to use random convex combinations of adjacent samples to draw new points from the continuous manifold.

For a latent vector $z \in Z$, ideally only m directly adjacent neighbors z' are chosen. Points on an s-dimensional manifold are expected to have $m = 2s$ directly adjacent neighbors (1D: left, right; 2D: left, right, up, down; 3D: behind, in front, ...). In section 4 we utilize a tool for intrinsic dimensionality estimation to determine an ideal value for m. The convex combination of several samples is expected to combine the corresponding individual features. As the latent manifolds from which we sample are high dimensional ($s \sim 50$) we cannot utilize all directly adjacent neighbors: If too many nearest neighbors are chosen, they only share the high-level features as the set is too big to produce an image that exhibits distinct unique features of all individuals. We thus randomly pick k samples $z'' \subset z'$. We then create a set of weights $w_i = \frac{|\hat{w}_i|}{\sum_i |\hat{w}_i|}$, with $i \in \{1, .., k\}$, where $\hat{w}_i \sim \mathcal{N}$ are drawn from a uniform distribution. Given the simplex of weights w_i we create a sample $\hat{z}$ by a convex combination of z''

$$\hat{z} = \sum_{i=0}^{k} w_i \; z''_i : \text{CC}^Z_{m,k}(z) \text{ , with } \hat{z} \sim P_{\hat{z}}. \tag{1}$$

With the created latent vectors $\hat{z}$ we fill the space between the discrete latent representations z and this way reconstruct the full continuous manifold.

3.2 Image Generation Procedure

This section explains how new images are generated with X-GAN. Thereby, we explain the central role of the convex combination (CC) strategy for guar-

anteeing the generation of various and realistic images. The utilized network architecture is shown in figure 2.

Given a set of images X we use the encoder E to produce an embedding $Z = E(X)$. To create new latent vectors $\hat{z}$ we utilize our novel CC-sampler module. It draws samples from the continuous latent manifold which is defined by the discrete set of image representations Z and thus ensures that no mode representations are missed. Our convex combination strategy samples directly from the real manifold without assuming any prior. On the contrary, VAE enforce a prior on the manifold which may not match the real data distribution, hence it suffers from the overpowering effect. Finally an image $\hat{x} = G(\hat{z})$ can be produced using the generator G. Notice that we only need the networks E and G for the generation of new images.

3.3 Training Procedure

In the following, the training procedure of our proposed method is explained step-by-step. The summarized procedure is shown in algorithm 1 and a visualization of the network architecture can be found in figure 2. The notation of the utilized losses follows the structure $\mathcal{L}_{N\_\mathrm{T}}$, with N as the network which is optimized using $\mathcal{L}_{N\_\mathrm{T}}$ and the T as the loss type.

The parameter λ balances the GAN loss against to the remaining loss functions. We empirically found that further weighting factors do not improve the results.

Encoder Training Along with the generator, the encoder is updated with an L1 autoencoder loss on features of the real and the fake images $\mathcal{L}_{EG\_\mathrm{ae}}$ (see algorithm 1 l.7). Therefor, we utilize an image feature extraction function $f(\cdot)$. For the autoencoder loss, no convex combinations are used in the training. The reason for this is that the autoencoder is responsible to shape the basic structure of the latent space. This is done by assigning real images to their corresponding latent representation. VAE would fill the continuous space in between real image embeddings with faked samples that minimize the reconstruction error of all surrounding images. This way of combining features does not necessarily result in realistic images [11]. Thus, we train the autoencoder with discrete latent vectors and utilize a learned GAN loss to train the generator to reconstruct the full continuous distribution defined by convex combinations. Apart from the autoencoder loss, the encoder is trained to minimize a classification loss $\mathcal{L}_{E\_\mathrm{clf}}$ (see algorithm 1 l.6) to cluster the embedding vectors according to their label y in the embedding space. This strategy of inducing conditioning helps to structure the latent space.

GAN Training So far we have a basic encoding space and we trained E and G to reconstruct the discrete images of the training set. However, we did not train the generator to produce images from latent vectors which are positioned in between the discrete image representations. We propose an adversarial loss

$\mathcal{L}_{\text{GD\_gan}}$ (see algorithm 1 l.8) to fill the space in between samples. To generate input latent variables for the GAN we apply our novel convex combination module to sample from the continuous manifold.

Stabilizing Loss Functions We propose to additionally train G with an L2 loss $\mathcal{L}_{\text{G\_emb}}$ (see algorithm 1 l.9) between the input latent vector $\hat{z}$ and the encoded fake image latent vector $E(G(\hat{z}))$. This loss can only be minimized if the generator creates images, that can be mapped back precisely to the latent space. Thus, sudden jumps from one to the other image in the image space are suppressed by this loss. This way, a smooth image transition (such as in figure 5) is encouraged.

An additional classification loss $\mathcal{L}_{\text{G\_clf}}$ (see algorithm 1 l.10) is applied on the synthetic images to reward the generation of recognizable objects. Therefore, it must be ensured that the $\hat{z}$ (equation 1) is uniquely assignable to one label by selecting z_i'' from the same class.

3.4 Conditioning

An important feature of generative models is the ability to produce samples that are conditioned on certain characteristic classes. The common procedure of passing a label vector to the generator is to concatenate it to one or several layers of the network. Although this forms a joint representation, neural architectures have problems to capture the complex associations between the two different modalities of sparse and dense information [19]. This is especially the case for fine-grained datasets where the number of classes becomes very large. A more natural way of conditioning, that does not require sparse information, would be to directly encode the class label in the position of a latent vector within the manifold. However, commonly, the full continuous manifold in latent space is not well defined, but a prior distribution is assumed from which samples are drawn. Our model, thanks to the convex- combination strategy, produces a latent space that allows a more natural conditioning of the latent vectors by their position in latent space: The embedding classifier of our model clusters the latent representation according to their class label by means of a classification loss. This supports the encoder in creating a well structured latent space.

In practice, given a set of images x from a common class y we can assume that their embedding vectors $z = E(x)$, can be found in the same cluster in the latent space. Thus, a new sample $\hat{z}$ produced by equation 1 will also be placed in this cluster and is naturally conditioned by its position in latent space. This way not only class conditioning but also fine-grained mode conditioning is possible through the encoding in the latent vector.

In table 3 and figure 6 (e) we show that our conditioning method produces images that are more realistic than those gained with models utilizing the approach of label concatenation.

4 Experiments

In this section, we describe the experimental setup, the datasets and the model parameters used. Moreover, we explain our evaluation procedures and present their results. In several ablation experiments, we illustrate the advantages of convex combination based sampling.

4.1 Implementation Details

The encoder, generator, and discriminator are all implemented as 6-Layer-CNNs. They utilize batch-normalization [14] and leaky-ReLU as activation function. C_{emb} is a fully connected network with two layers. C_{img} is a VGG-19 network [30], pre-trained on the given dataset and fixed during training. This ensures useful gradients from the beginning of training and avoids overfitting in a later stage. As a feature descriptor f we choose features from intermediate layers of the VGG network as proposed by Chen et al. [8]. Additionally, we add the features of a second VGG-19 network that is trained as a discriminator along with the rest of the model. We set the latent variable size to 512 and the batch size to 80. We choose $\lambda = 2$. That means that we weight the GAN loss twice as high as the remaining loss functions. We train the network with the Adam optimizer [17] with a learning rate of 10^{-4}. For training we only need 16 epochs, what is far less then what is required to train the cVAE-GAN ($\sim$ 50 epochs) or the MD-GAN ($\sim$ 100 epochs). For the convex combination sampling, we set the parameters m (total number of nearest neighbors) by estimating the intrinsic dimensionality s of the embedding manifold with a Maximum Likelihood Estimation [21]. For our latent embedding we find a mean (over the classes) of $s = 48$ resulting in $m \approx 100$ (see 3.1). To maintain individual features for each sample we draw $\hat{z}$ as randomly weighted convex combinations of $k = 5$ adjacent samples. In the evaluation we compare our model with cVAE-GAN [1][1] and MD-GAN [7][2].

Our model is applied to two different datasets: The FaceScrub dataset [26] with images of 530 different individuals and the 102 Category Flower dataset [24]. For the FaceScrub dataset, we followed the alignment procedure suggested by [1]. We want to show that we can maintain a high image quality even for non-aligned image data. Thus, other than Bao et al. [1] we do not perform any preprocessing on the 102 Category flower dataset.

4.2 Visual Image Quality Comparison

In figure 4 we compare our results with MD-GAN and cVAE-GAN. For the FaceScrub dataset, the MD-GAN generations exhibit noise and artifacts while the cVAE-GAN and the X-GAN produce photo-realistic images. While the cVAE-GAN performs well on the FaceScrub dataset the generated flowers look worse. In the cVAE-GAN paper [1] the authors also evaluate their model with these two

[1] cVAE-GAN implementation: https://github.com/tatsy/keras-generative
[2] MD-GAN implementation: https://github.com/wiseodd/generative-models

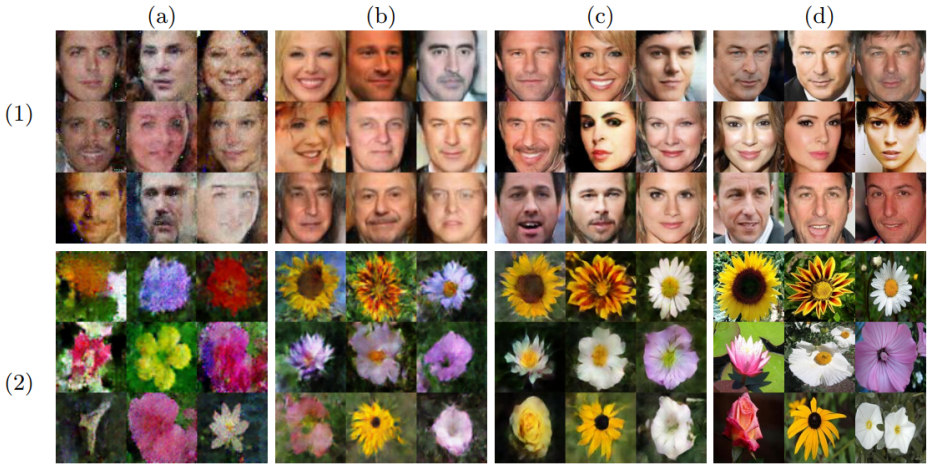

Fig. 4. Comparison of generated samples from different methods on the (1) FaceScrub dataset [24] and (2) 102 Category Flower dataset [26] (bottom row). The images are generated by (a) MD-GAN [7], (b) cVAE-GAN [1], (c) X-GAN (this work) (d) real. Using the MD-GAN artifacts are clearly visible, the cVAE-GAN and the X-GAN produce photo-realistic images for the aligned faces. For the more challenging flower dataset, which has not been preprocessed, the image quality for the X-GAN is better than for the cVAE-GAN

.

datasets and present appealing results. However, other than we do they align the images of both datasets. The fact that still photo-realistic images are generated with X-GAN for the unaligned flower dataset shows, that our model is robust against the missing alignment.

4.3 Discriminator Score for Missing Mode Evaluation

Che et al. [7] propose to utilize a third party discriminator D^* to quantify the robustness of a model towards the missing mode problem. After training D^* on a set of real and fake images the discriminator is applied to a set of test images. The mode of the images that are clearly recognized as real ($D^* \approx 1$) is obviously not represented in the fake image distribution. Thus, the number of test images for which $D^* \approx 1$ serves as a proxy for the degree of mode collapse. As this approach requires a strong discriminator D^* we use the FaceNet [29][3] which is trained on the CASIA dataset [22] to produce a face embedding. On top of the FaceNet, we stack a two-layer neural network (D^*). For our experiments, we used three different thresholds $t \in \{0.9, 0.95, 0.97\}$ that determine for which D^* score an image is assumed to be not represented in the fake data. The mean over five runs of this experiment is documented in table 1. In total 4412 test images are used

[3] weights/code for FaceNet: https://github.com/davidsandberg/facenet

Table 1. This table shows the D^*-score for the MD-GAN [7], cVAE-GAN [1] and X-GAN (our model). To obtain this score a third-party-discriminator is trained on a set of real and a set of fake images. Afterwards, it is evaluated on a set of test images. Test samples for which $D^* \approx 1$ are assumed not to be represented by the faked images. Thus, a high D^*-score is a proxy for mode collapse.

Model	$t = 0.90$ [# samples]	$t = 0.95$ [# samples]	$t = 0.97$ [# samples]
MD-GAN	2737 ± 34	2299 ± 62	1952 ± 64
CVAEGAN	325 ± 47	87 ± 22	28 ± 14
X-GAN	$\mathbf{134 \pm 32}$	$\mathbf{15 \pm 10}$	$\mathbf{2 \pm 2}$

for the experiment. For a strict threshold of $t = 0.97$, our model exhibits almost no missing representations anymore (2 ± 2), while underrepresented modes for cVAE-GAN are found for 28 ± 14 images. The MD-GAN, even for $t = 0.97$, collapses to almost 50% of the existing modes.

4.4 Image Morphing

The embedding loss (algorithm 1 l.9) ensures that a smooth transition in the latent space corresponds to a smooth transition in the image space. To visually underpin this, in figure 5 we show the results of an image morphing experiment. The corners of the shown image array are real images while the space in between is filled with fake samples. The fact that this is only noticeable in a detailed investigation shows that our synthesized images smoothly supplement the underlying real distribution.

Table 2. This table shows the inception score computed on FaceScrub dataset [24] as a mean of 5×100K generated images.

Model	Inception Score
Real data	2.372 ± 0.003
cVAE-GAN [1]	1.757 ± 0.002
X-GAN	$\mathbf{1.831 \pm 0.003}$

4.5 Inception score

The realism of generated images is characterized by a high diversity in the distribution and easily recognizable objects. Salimans et al. [28] propose the Inception Score which relies on the output score of a pre-trained InceptionNet[32]. A high diversity is characterized by a high entropy of the overall score distribution, while a recognizable object corresponds to a low entropy for a single image.

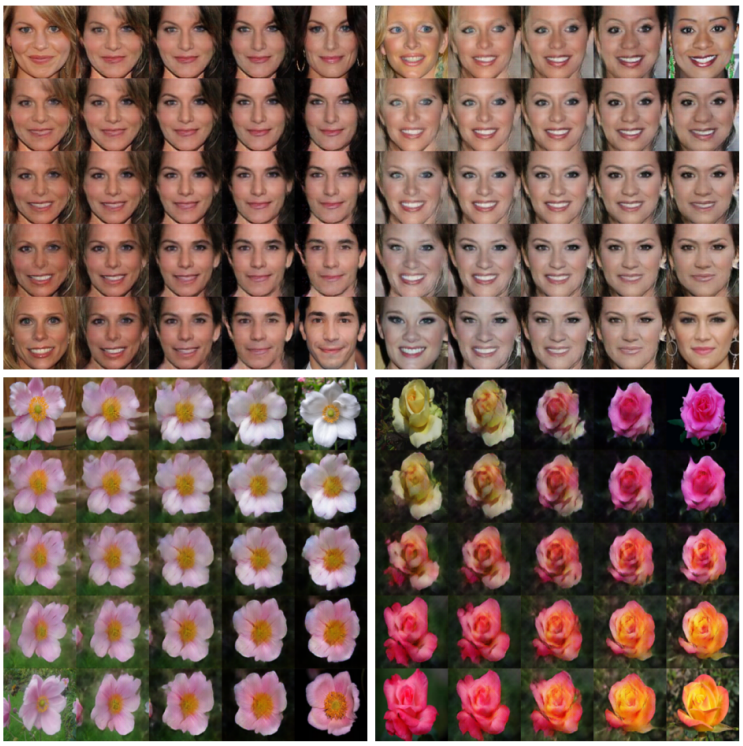

Fig. 5. This figure shows morphing experiments. For each of the shown image-arrays, the four corner images are real and the space in between is filled with fake images gained from weighted convex combinations. The corner images of the upper right image originate from unseen test data. Especially for the illustrated faces, the jump from real to fake data is hardly visible. On a closer inspection of the flower image-arrays, one can recognize the smoother background. The fine features of the flower itself are preserved. This shows that the generated images complement the underlying real distribution well.

In table 2 we compare the inception score of generated images of our model with cVAE-GAN for the faces dataset. Images generated with our model achieve higher scores.

4.6 Ablation Study

In this section, we quantitatively (see table 3) and qualitatively (see figure 6) evaluate the impacts of removing or replacing parts of X-GAN. Here, we not only illustrate the benefits of the proposed convex combination strategy compared to conventional methods but also present an unsupervised version of the X-GAN.

Unsupervised X-GAN An unsupervised version of the proposed X-GAN can be obtained by removing the losses $\mathcal{L}_{\text{E\_clf}}$ and $\mathcal{L}_{\text{G\_clf}}$. Table 3 shows that our

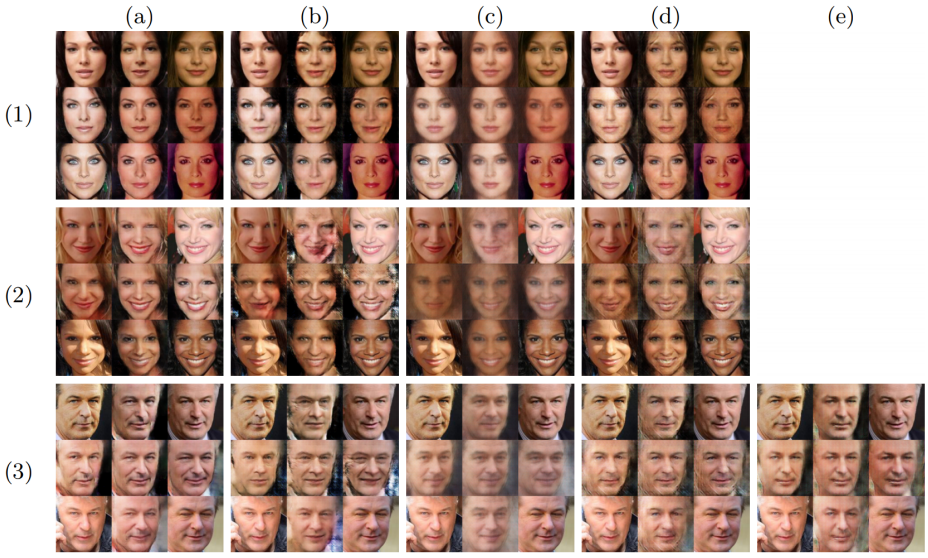

Fig. 6. This figure shows image morphing experiments for each of the performed ablation studies. The images in the corners of the image array are real, while the space in between is filled with faked images. (a) X-GAN, (b) unsupervised X-GAN, (c) KL-Loss, (d) KDE-sampling, (e) concatenate label vector. For each of the image arrays, the four corner images are real, while the space in between is filled with faked images. (1), (2) different individuals in the corner images, (3) different images of the same individual in the corner images.

model still performs better then the supervised cVAE-GAN. Even though The discriminator score deteriorates, it is still on the same scale as the cVAE-GAN. In the visual evaluation (figure 6 (b)) it can be seen that the generated images are still sharp and detailed. However, some exhibit artifacts that are not observed for the original X-GAN.

KL-Loss In this experiment, the convex combination strategy is replaced by the KL-loss based approach proposed by Rosca et al. [27]. This approach enforces a prior to the latent distribution. This results in a deterioration of the inception score and the discriminator score (see table 3). In the visual analysis in figure 6 (c) it can be seen that no distinct facial details can be found in the generated bury samples anymore. Unlike the X-GAN generations, there is a clear jump between from the real to the fake images visible.

KDE Sampling The kernel density estimation (KDE) is a non-parametric approach to estimate the continuous distribution from a set of discrete samples. In this experiment, the creation of latent vectors with convex combinations is replaced by KDE sampling. However, the estimation of the high dimensional

latent space may not sufficiently represent the complex manifold (see section 3.1). This explains the deterioration of the inception and discriminator score (see table 3) compared to the convex combination sampling. The visual artifacts in the generated images underpin the quantitative observations (see figure 6 (d)).

Conditioning by the Concatenation of Label Vectors Beside the convex combination strategy, another special feature of the X-GAN model is the novel conditioning approach (see section 3.4). In this ablation experiment, we omit the loss $\mathcal{L}_{\text{E\_clf}}$ and condition the latent variables by the concatenation of a label vector to the latent vector. Neural architectures face difficulties when mixing up sparse and dense information. Thus, as it can be seen in table 3 the conditioning by concatenation performs significantly worse than the X-GAN regarding the inception score and the discriminator score. In figure 6 (e) a close look shows that the faked images in this ablation study exhibit less distinct features compared to the original X-GAN. As the label vector is discrete, it is pointless to morph between images of different classes. Thus, only row (3) is shown.

Table 3. This table shows the quantitative results of the ablation study. The evaluation metrics are the inception score (see section 4.5) and the discriminator score (see section 4.3). The evaluation approves that the convex combination strategy outperforms conventional approaches (KL-loss and KDE-sampling). Moreover, the superiority of the novel conditioning approach compared to the conventional concatenation of label and latent vectors is approved. Finally, we show that also an unsupervised version of the X-GAN achieves decent inception and discriminator scores.

Model	Inception Score	D^* Score [# samples]
X-GAN	1.831 ± 0.003	15 ± 10
unsupervised	1.795 ± 0.003	153 ± 28
KL-loss	1.612 ± 0.002	497 ± 43
KDE-sampling	1.711 ± 0.003	203 ± 34
concat. label	1.727 ± 0.003	76 ± 21

5 Conclusion

In this paper, we propose a novel strategy of generating a latent vector representation by convex combinations. This enables us to reconstruct the full continuous distribution of the manifold defined by the given dataset. We evaluate our method on two datasets. With various qualitative and quantitative evaluation approaches, we show the X-GAN architecture represents a major step forward in terms of tackling the issue of mode collapse and the overpowering effect.

References

1. Bao, J., Chen, D., Wen, F., Li, H., Hua, G.: Cvae-gan: Fine-grained image generation through asymmetric training. arXiv preprint arXiv:1703.10155 (2017)
2. Bautista, M.A., Sanakoyeu, A., Tikhoncheva, E., Ommer, B.: Cliquecnn: Deep unsupervised exemplar learning. In: Advances in Neural Information Processing Systems. pp. 3846–3854 (2016)
3. Bojanowski, P., Joulin, A., Lopez-Paz, D., Szlam, A.: Optimizing the latent space of generative networks. arXiv preprint arXiv:1707.05776 (2017)
4. Bowman, S.R., Vilnis, L., Vinyals, O., Dai, A.M., Jozefowicz, R., Bengio, S.: Generating sentences from a continuous space. arXiv preprint arXiv:1511.06349 (2015)
5. Brattoli, B., Büchler, U., Wahl, A.S., Schwab, M.E., Ommer, B.: Lstm self-supervision for detailed behavior analysis. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
6. Büchler, U., Brattoli, B., Ommer, B.: Improving spatiotemporal self-supervision by deep reinforcement learning. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
7. Che, T., Li, Y., Jacob, A.P., Bengio, Y., Li, W.: Mode regularized generative adversarial networks. arXiv preprint arXiv:1612.02136 (2016)
8. Chen, Q., Koltun, V.: Photographic image synthesis with cascaded refinement networks. arXiv preprint arXiv:1707.09405 (2017)
9. Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., Abbeel, P.: Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In: Advances in Neural Information Processing Systems. pp. 2172–2180 (2016)
10. Dilokthanakul, N., Mediano, P.A., Garnelo, M., Lee, M.C., Salimbeni, H., Arulkumaran, K., Shanahan, M.: Deep unsupervised clustering with gaussian mixture variational autoencoders. arXiv preprint arXiv:1611.02648 (2016)
11. Dosovitskiy, A., Brox, T.: Generating images with perceptual similarity metrics based on deep networks. In: Advances in Neural Information Processing Systems. pp. 658–666 (2016)
12. Esser, P., Sutter, E., Ommer, B.: A variational u-net for conditional appearance and shape generation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8857–8866 (2018)
13. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. pp. 2672–2680 (2014)
14. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning. pp. 448–456 (2015)
15. Joyce, J.M.: Kullback-leibler divergence. In: International Encyclopedia of Statistical Science, pp. 720–722. Springer (2011)
16. Kaae Sønderby, C., Raiko, T., Maaløe, L., Kaae Sønderby, S., Winther, O.: How to train deep variational autoencoders and probabilistic ladder networks. arxiv preprint. arXiv preprint arXiv:1602.02282 (2016)
17. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
18. Kingma, D.P., Salimans, T., Welling, M.: Improving variational inference with inverse autoregressive flow. arXiv preprint arXiv:1606.04934 (2016)

19. Kwak, H., Zhang, B.T.: Ways of conditioning generative adversarial networks. arXiv preprint arXiv:1611.01455 (2016)
20. Larsen, A.B.L., Sønderby, S.K., Larochelle, H., Winther, O.: Autoencoding beyond pixels using a learned similarity metric. arXiv preprint arXiv:1512.09300 (2015)
21. Levina, E., Bickel, P.J.: Maximum likelihood estimation of intrinsic dimension. In: Advances in neural information processing systems. pp. 777–784 (2005)
22. Li, S., Yi, D., Lei, Z., Liao, S.: The casia nir-vis 2.0 face database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 348–353 (2013)
23. Milbich, T., Bautista, M., Sutter, E., Ommer, B.: Unsupervised video understanding by reconciliation of posture similarities. In: Proceedings of the IEEE International Conference on Computer Vision (2017)
24. Ng, H.W., Winkler, S.: A data-driven approach to cleaning large face datasets. In: Image Processing (ICIP), 2014 IEEE International Conference on. pp. 343–347. IEEE (2014)
25. Nguyen, A., Yosinski, J., Bengio, Y., Dosovitskiy, A., Clune, J.: Plug & play generative networks: Conditional iterative generation of images in latent space. arXiv preprint arXiv:1612.00005 (2016)
26. Nilsback, M.E., Zisserman, A.: Automated flower classification over a large number of classes. In: Computer Vision, Graphics & Image Processing, 2008. ICVGIP'08. Sixth Indian Conference on. pp. 722–729. IEEE (2008)
27. Rosca, M., Lakshminarayanan, B., Warde-Farley, D., Mohamed, S.: Variational approaches for auto-encoding generative adversarial networks. arXiv preprint arXiv:1706.04987 (2017)
28. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. In: Advances in Neural Information Processing Systems. pp. 2234–2242 (2016)
29. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 815–823 (2015)
30. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
31. Stone, C.J.: Optimal global rates of convergence for nonparametric regression. The annals of statistics pp. 1040–1053 (1982)
32. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Computer Vision and Pattern Recognition (CVPR) (2015), http://arxiv.org/abs/1409.4842