Non-Adversarial Image Synthesis with Generative Latent Nearest Neighbors

Yedid Hoshen[1, 2], Ke Li[3], and Jitendra Malik[2, 3]

[1]Hebrew University of Jerusalem
[2]Facebook AI Research
[3]UC Berkeley

Abstract

Unconditional image generation has recently been dominated by generative adversarial networks (GANs). GAN methods train a generator which regresses images from random noise vectors, as well as a discriminator that attempts to differentiate between the generated images and a training set of real images. GANs have shown amazing results at generating realistic looking images. Despite their success, GANs suffer from critical drawbacks including: unstable training and mode-dropping. The weaknesses in GANs have motivated research into alternatives including: variational auto-encoders (VAEs), latent embedding learning methods (e.g. GLO) and nearest-neighbor based implicit maximum likelihood estimation (IMLE). Unfortunately at the moment, GANs still significantly outperform the alternative methods for image generation. In this work, we present a novel method - Generative Latent Nearest Neighbors (GLANN) - for training generative models without adversarial training. GLANN combines the strengths of IMLE and GLO in a way that overcomes the main drawbacks of each method. Consequently, GLANN generates images that are far better than GLO and IMLE. Our method does not suffer from mode collapse which plagues GAN training and is much more stable. Qualitative results show that GLANN outperforms a baseline consisting of 800 GANs and VAEs on commonly used datasets. Our models are also shown to be effective for training truly non-adversarial unsupervised image translation.

1. Introduction

Generative image modeling is a long-standing goal for computer vision. Unconditional generative models attempt to learn functions that generate the entire image distribution given a finite number of training samples. Generative Adversarial Networks (GANs) [9] are a recently introduced technique for image generative modeling. They are used extensively for image generation owing to: i) training effective unconditional image generators ii) being almost the only method for unsupervised image translation between domains (but see NAM [15]) iii) being an effective perceptual image loss function (e.g. Pix2Pix [16]).

Along with their obvious advantages, GANs have critical disadvantages: i) GANs are very hard to train, this is expressed by a very erratic progression of training, sudden run collapses, and extreme sensitivity to hyper-parameters. ii) GANs suffer from mode-dropping - the modeling of only some but not all the modes of the target distribution. The birthday paradox can be used to measure the extent of mode dropping [2]: The number of modes modeled by a generator can be estimated by generating a fixed number of images and counting the number of repeated images. Empirical evaluation of GANs found that the number of modes is significantly lower than the number in the training distribution.

The disadvantages of GANs gave rise to research into non-adversarial alternatives for training generative models. GLO [4] and IMLE [23] are two such methods. GLO, introduced by Bojanowski et al., embeds the training images in a low dimensional space, so that they are reconstructed when the embedding is passed through a jointly trained deep generator. The advantages of GLO are i) encoding the entire distribution without mode dropping ii) the learned latent space corresponds to semantic image properties i.e. Euclidean distances between latent codes correspond to semantically meaningful differences. A critical disadvantage of GLO is that there is not a principled way to sample new images from it. Although the authors recommended fitting a Gaussian to the latent codes of the training images, this does not result in high-quality image synthesis.

IMLE was proposed by Li and Malik [23] for training generative models by sampling a large number of latent codes from an arbitrary distribution, mapping each to the image domain using a trained generator and ensuring

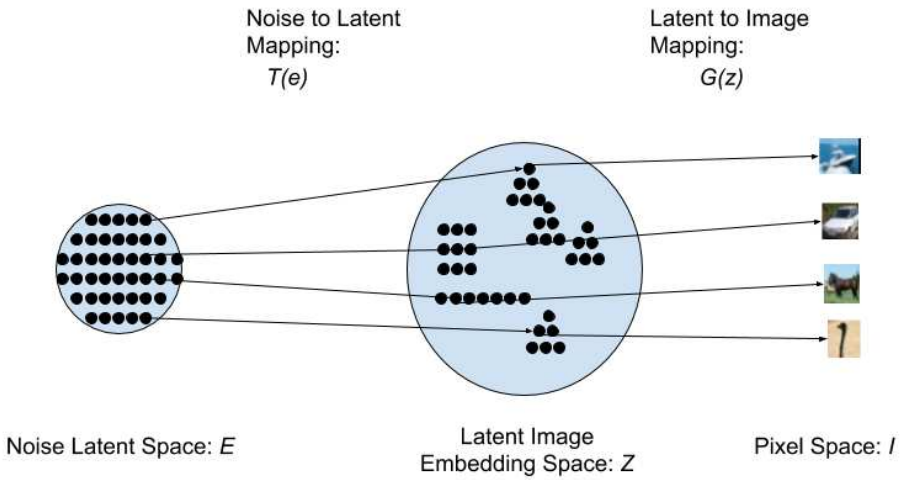

Figure 1. An illustration of our architecture: a random noise vector e is sampled and mapped to the latent space to yield latent code $z = T(e)$. The latent code is projected by the generator to yield image $I = G(z)$.

that for every training image there exists a generated image which is near to it. IMLE is trivial to sample from and does not suffer from mode-dropping. Like other nearest neighbor methods, IMLE is sensitive to the exact metric used, particularly given that the training set is finite. Recall that while the classic Cover-Hart result [7] tells us that asymptotically the error rate of the nearest neighbor classifier is within a factor of 2 of the Bayes risk, when we use a finite set of exemplars better choices of metrics give us better classifier performance. When trained directly on image pixels using an L_2 loss, IMLE synthesizes blurry images.

In this work, we present a new technique, Generative Latent Nearest Neighbors (GLANN), which is able to train generative models of comparable or better quality to GANs. Our method overcomes the metric problem of IMLE by first embedding the training images using GLO. The attractive linear properties of the latent space induced by GLO, allow the Euclidean metric to be semantically meaningful in the latent space $\mathcal{Z}$. We train an IMLE-based model to map between an arbitrary noise distribution $\mathcal{E}$, and the GLO latent space $\mathcal{Z}$. The GLO generator can then map the generated latent codes to pixel space, thus generating an image. Our method GLANN enjoys the best of both IMLE and GLO: easy sampling, modeling the entire distribution, stable training and sharp image synthesis. A schema of our approach is presented in Fig. 1.

We quantitatively evaluate our method using established protocols and find that it significantly outperforms other non-adversarial methods, while being usually better or competitive with current GAN based models. GLANN is also able to achieve promising results on high-resolution image generation and 3D generation. Finally, we show that GLANN-trained models are the first to perform truly non-adversarial unsupervised image translation.

2. Previous Work

Generative Modeling: Generative modeling of images is a long-standing problem of wide applicability. Early approaches included mixtures of Gaussian models (GMM) [39]. Such methods were very limited in image resolution and quality. Deep learning methods have continually been used for image generative models. Variational Autoencoders (VAEs) [20] were a significant breakthrough in deep generative modeling, introduced by Kingma and Welling. VAEs are able to generate images from the Gaussian distribution by making a variational approximation. Although VAEs are relatively simple to train and have solid theoretical foundations, they generally do not generate sharp images.

Several other non-adversarial training paradigms exist: Generative invertible flows [8], that were recently extended to high resolution [19] but at prohibitive computational costs. Another training paradigm is autoregressive image models e.g. PixelRNN/PixelCNN [29], where pixels are modeled sequentially. Autoregressive models are computationally expensive and underperform adversarial methods although they are the state of the art in audio generation (e.g. WaveNet [28]).

Adversarial Generative Models: Generative Adversarial Networks (GANs) were first introduced by Goodfellow et al. [9] and are the state-of-the-art method for training generative models. A basic discussion on GANs was given in Sec. 1. GANs have shown a remarkable ability for image generation, but suffer from difficult training and mode dropping. Many methods were proposed for improving GANs e.g. changing the loss function (e.g. Wasserstein GAN [1])

or regularizing the discriminator to be Lipschitz by: clipping [1], gradient regularization [10, 25] or spectral normalization [26]. GAN training was shown to scale to high resolutions [37] using engineering tricks and careful hyperparameter selection.

Evaluation of Generative Models: Evaluation of generative models is challenging. Early works evaluated generative models using probabilistic criteria (e.g. [39]). More recent generative models (particularly GANs) are not amenable to such evaluation. GAN generations have traditionally been evaluated using visual inspection of a handful of examples or by a user study. More recently, more principled evaluation protocols have emerged. Inception Scores (IS) which take into account both diversity and quality were first introduced by [31]. FID scores [11] were more recently introduced to overcome major flaws of the IS protocol [3]. Very recently, a method for generative evaluation which is able to capture both precision and recall was introduced by Sajjadi et al. [30]. Due to the hyperparameters sensitivity of GANs, a large scale study of the performance of 7 different GANs and VAE was carried out by Lucic et al. [24] over a large search space of 100 different hyperparameters, establishing a common baseline for evaluation.

Non-Adversarial Methods: The disadvantages of GANs motivated research into GAN alternatives. GLO [4], a recently introduced encoderless generative model which uses a non-adversarial loss function, achieves better results than VAEs. Due to the lack of a good sampling procedure, it does not outperform GANs (see Sec. 3.1). IMLE [23], a method related to ICP was also introduced for training unconditional generative models, however due to computational challenges and the choice of metric, it also does not outperform GANs. Chen and Koltun [5] presented a non-adversarial method for supervised image mapping, which in some cases was found to be competitive with adversarial methods. Hoshen and Wolf introduced an ICP-based method [13] for unsupervised word translation which contains no adversarial training. They also presented non-adversarial method, NAM [14, 15, 12], for unsupervised image mapping. The method relies on having access to a strong unconditional model of the target domain, which is typically trained using GANs.

3. Our method

In this section we present a method - GLANN - for synthesizing high-quality images without using GANs.

3.1. GLO

Classical methods often factorize a set of data points $\{x_1, x_2, .., x_T\}$ via the following decomposition:

$$x_i = W z_i \quad \forall i \tag{1}$$

Where z_i is a latent code describing x_i, and W is a set of weights. Such factorization is poorly constrained and is typically accompanied by other constraints such as low-rank, positivity (NMF), sparsity etc. Both W and z_i are optimized directly e.g. by alternating least squares or SVD. The resulting z_i are latent vectors that embed the data in a lower dimension and typically better behaved space. It is often found that attributes become linear operations in the latent space.

GLO [4] is a recently introduced deep method, which is different from the above in three aspects: i) Constraining all latent vectors to lie on a unit sphere or a unit ball. ii) Replacing the linear matrix W, by a deep CNN generator $G()$ which is more suitable for modeling images. iii) Using a Laplacian pyramid loss function (but we find that a VGG [32] perceptual loss works better).

The GLO optimization objective is written in Eq. 2:

$$arg \min_{G,\{z_i\}} \sum_i \ell(G(z_i), x_i) \quad s.t. \quad \|z_i\| = 1 \tag{2}$$

Bojanowski et al [4], implement ℓ as a Laplacian pyramid. All weights are trained by SGD (including the generator weights $G()$ and a latent vector z_i per each training image x_i). After training, the result is a generator $G()$ and a latent embedding z_i of each training image x_i.

3.2. IMLE

IMLE [23] is a recent non-adversarial technique that maps between distributions using a maximum likelihood criterion. Each epoch of IMLE consists of the following stages: i) M random latent codes e_j are sampled from a normal distribution ii) The latent codes are mapped by the generator resulting in images $G(e_j)$ iii) For each training example x_i, the nearest generated image is found such that: $e_i = arg\min_{e_j} \|G(e_j), x_i|_2^2$ iv) $G()$ is optimized using nearest neighbors as approximate correspondences $G = arg\min_{\tilde{G}} \sum_i \|\tilde{G}(e_i), x_i\|_2^2$ This procedure is repeated until the convergence of $G()$.

3.3. Limitations of GLO and IMLE

The main limitation of GLO is that the generator is not trained to sample from any known distribution i.e. the distribution of z_i is unknown and we cannot directly sample from it. When sampling latent variables from a normal distribution or when fitting a Gaussian to the training set latent codes (as advocated in [4]), generations that are of much lower quality than GANs are usually obtained. This prevents GLO from being competitive with GANs.

Although sampling from an IMLE trained generator is trivial, the training is not, a good metric might not be known, the nearest neighbor computation and feature extraction for each random noise generation is costly. IMLE typically results in blurry image synthesis.

3.4. GLANN: Generative Latent Nearest Neighbor

We present a method - GLANN - that overcomes the weaknesses of both GLO and IMLE. GLANN consists of two stages: i) embedding the high-dimensional image space into a "well-behaved" latent space using GLO. ii) Mapping between an arbitrary distribution (typically a multi-dimensional normal distribution) and the low-dimensional latent space using IMLE.

3.4.1 Stage 1: Latent embedding

Images are high-dimensional and distances between them in pixel space might not be meaningful. This makes IMLE and the use of simple metric functions such as L_1 or L_2 less effective in pixel space. In some cases perceptual features may be found under which distances make sense, however they are high dimensional and expensive to compute.

Instead our method first embeds the training images in a low dimensional space using GLO. Differently from the GLO algorithm, we use a VGG perceptual loss function. The optimization objective is written in Eq, 5:

$$arg \min_{\tilde{G},\{z_i\}} \sum_i \ell_{perceptual}(\tilde{G}(z_i), x_i) \quad s.t. \quad \|z_i\| = 1 \quad (3)$$

All parameters are optimized directly by SGD. By the end of training, the training images are embedded by the low dimensional latent codes $\{z_i\}$. The latent space $\mathcal{Z}$ enjoys convenient properties such as linearity. A significant benefit of this space is that a Euclidean metric in the $\mathcal{Z}$ space can typically yield more more semantically meaningful results than raw image pixels.

3.4.2 Stage 2: Sampling from the latent space

GLO replaced the problem of sampling from image pixels $\mathcal{X}$ by the problem of sampling from $\mathcal{Z}$ without offering an effective sampling algorithm. Although the original paper suggests fitting a Gaussian to the training latent vectors z_i, this typically does not result in good generations. Instead we propose learning a mapping from a distribution from which sampling is trivial (e.g. multivariate normal) to the empirical latent code distribution using IMLE.

At the beginning of each epoch, we sample a set of random noise codes $e_1..e_m..e_M$ from the noise distribution. Each one of the codes is mapped using mapping function T to the latent space - $\tilde{z}_m = T(e_m)$.

During the epoch, our method iteratively samples a mini-batch of latent codes from the set $\{z_1..z_t..z_T\}$ computed in the previous stage. For each latent code z_t, we find the nearest neighbor mapped noise vector (using a Euclidean distance metric):

$$e_t = \arg\min_{e_m} \|z_t - T(e_m)\|_2^2 \quad (4)$$

The approximate matches can now be used for finetuning the mapping function T:

$$T = \arg\min_{\tilde{T}} \sum_t \|z_t - \tilde{T}(e_t)\|_2^2 \quad (5)$$

This procedure is repeated until the convergence of $T()$. It was shown theoretically by Li and Malik [23], that the method achieves a form of maximum likelihood estimate.

3.4.3 Sampling new images

Synthesizing new images is now a simple task: We first sample a noise vector from the multivariate normal distribution $e \sim N(0, I)$. The new sample is mapped to the latent code space - $z_e = T(e)$.

By our previous optimization, $T()$ was trained such that latent code z_e lies close to the data manifold. We can therefore use the generator to project the latent code to image space by our GLO trained generator $I_e = G(z_e)$. I_e will appear to come from the distribution of the input images x.

It is also possible to invert this transformation by optimizing for the noise vector e given an image I:

$$e = arg\min_{\tilde{e}} \ell(G(T(\tilde{e})), I) \quad (6)$$

4. Experiments

To evaluate the performance of our proposed method, we perform quantitative and qualitative experiments comparing our method against established baselines.

4.1. Quantitative Image Generation Results

In order to compare the quality of our results against representative adversarial methods, we evaluate our method using the protocol established by Lucic et al. [24]. This protocol fixes the architecture of all generative models to be InfoGAN [6]. They evaluate 7 representative adversarial models (DCGAN, LSGAN, NSGAN, W-GAN, W-GAN GP, DRAGAN, BEGAN) and a single non-adversarial model (VAE). In [24], significant computational resources are used to evaluate the performance of each method over a set of 100 hyper-parameter settings, e.g.: learning rate, regularization, presence of batch norm etc.

Finding good evaluation metrics for generative models is an active research area. Lucic et al. argue that the previously used Inception Score (IS) is not a good evaluation metric, as the maximal IS score is obtained by synthesizing a single image from every class. Instead, they advocate using Frechet Inception Distance (FID) [11]. FID measures

Table 1. Quality of Generation (FID)

	Adversarial					Non-Adversarial		
Dataset	MM GAN	NS GAN	LSGAN	WGAN	BEGAN	VAE	GLO	Ours
MNIST	9.8 ± 0.9	6.8 ± 0.5	7.8 ± 0.6	$\mathbf{6.7} \pm 0.4$	13.1 ± 1.0	23.8 ± 0.6	49.6 ± 0.3	8.6 ± 0.1
Fashion	29.6 ± 1.6	26.5 ± 1.6	30.7 ± 2.2	21.5 ± 1.6	22.9 ± 0.9	58.7 ± 1.2	57.7 ± 0.4	$\mathbf{13.0} \pm 0.1$
Cifar10	72.7 ± 3.6	58.5 ± 1.9	87.1 ± 47.5	55.2 ± 2.3	71.4 ± 1.6	155.7 ± 11.6	65.4 ± 0.2	$\mathbf{46.5} \pm 0.2$
CelebA	65.6 ± 4.2	55.0 ± 3.3	53.9 ± 2.8	41.3 ± 2.0	$\mathbf{38.9} \pm 0.9$	85.7 ± 3.8	52.4 ± 0.5	46.3 ± 0.1

the similarity of the distributions of real and generated images by two steps: i) Running the Inception network as a feature extractor to embed each of the real and generated images ii) Fitting a multi-variate Gaussian to the real and generated embeddings separately, to yield means μ_r, μ_g and variances Σ_r, Σ_g for the real and generated distributions respectively. The FID score is then computed as in Eq. 7:

$$FID = \|\mu_r - \mu_g\|_2^2 + Tr(\Sigma_r + \Sigma_g - 2(\Sigma_r\Sigma_g)^{\frac{1}{2}}) \quad (7)$$

Lucic et al. evaluate the 8 baselines on 4 standard public datasets: MNIST [22], Fashion MNIST [35], CIFAR10 [21] and CelebA [36]. MNIST, Fashion-MNIST and CIFAR10 contain 50k color images and 10k validation images. MNIST and Fashion are 28×28 while CIFAR is 32×32.

For a fair comparison of our method, we use the same generator architecture used by Lucic et al. for our GLO model. We do not have a discriminator, instead, we use a VGG perceptual loss. Also differently from the methods tested by Lucic et al. we train an additional network $T()$ for IMLE sampling from the noise space to the latent space. In our implementation, $T()$ has two dense layers with 128 hidden nodes, with RelU and BatchNorm. GLANN actually uses fewer parameters than the baseline by not using a discriminator. Our method was trained with ADAM [18]. We used the highest learning rate that allowed convergence: 0.001 for the mapping network, 0.01 for the latent codes (0.003 for CelebA), generator learning rate was $0.1\times$ the latent code rate. 500 epochs were used for GLO training decayed by 0.5 every 50 epochs. 50 epochs were used for mapping network training.

Tab. 1 presents a comparison of the FID achieved by our method and those reported by Lucic et al. We removed DRAGAN and WGAN-GP for space consideration (and as other methods represented similar performance). The results for GLO were obtained by fitting a Gaussian to the learned latent codes (as suggested in [4]).

All GLO experiments used precisely the same perceptual loss as GLANN. The numbers for VAE were taken from [24] and used an L_1 loss. We run additional VAE experiments with the same perceptual loss as used by us. We obtained: MNIST 23.7 Fashion 41.2 Cifar10 86.0 CelebA 60.8. These results are competitive with GLO, but are much worse than ours.

On Fashion and CIFAR10, our method significantly outperforms all baselines - despite just using a single hyperparameter setting. Our method is competitive on MNIST, but as all methods performed well, it is hard to draw conclusions from it. A few other methods outperformed ours in terms of FID on CelebA, due to checkerboard patterns in our generated images. This is a well known phenomenon of deconvolutional architectures [27], which are now considered outdated. In Sec. 4.3, we show high-quality CelebA-HQ facial images generated by our method when trained using modern architectures.

Our method always significantly outperforms the VAE and GLO baselines (with the same perceptual loss), which are strong representatives of non-adversarial methods. One of the main messages in [24] was that GAN methods require a significant hyperparameter search to achieve good performance. Our method was shown to be very stable and achieved strong performance (top on two datasets) with a fixed hyperparameter setting. An extensive hyperparameter search can potentially further increase the performance our method, we leave it to future work.

To address the question of whether the fact that the perceptual loss uses VGG features trained on the ImageNet dataset is unfair, as the inception network used by FID was trained on ImageNet, we rerun GLANN with a perceptual loss based on a VGG network trained on the Place365 scene recognition dataset (which is significantly different from ImageNet). The FID scores were: MNIST 9.5 Fashion 13.5 Cifar10 56.7 CelebA 34.8. This is competitive and sometimes much better than the ImageNet VGG loss results. We can therefore conclude that the good performance of our method is not due to overfitting to the test metric.

4.2. Evaluation of Precision and Recall

FID is effective at measuring precision, but not recall. We therefore also opt for the evaluation metric recently presented by Sajjadi et al. [30] which they name PRD. PRD first embeds an equal number of generated and real images using the inception network. All image embeddings (real and generated) are concatenated and clustered into B bins ($B = 20$). Histograms $P(\omega)$, $Q(\omega)$ are computed for the

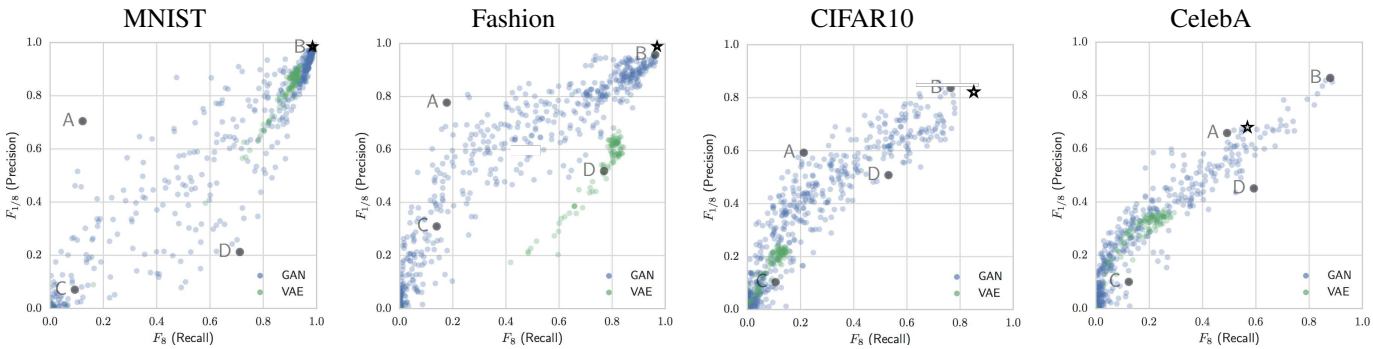

Figure 2. Precision-Recall measured by $(F_8, F_{\frac{1}{8}})$ for 4 datasets. The plots were reported by [30]. We marked the results of our model for each dataset by a star on the relevant plot.

number of images in each cluster from the real, generated data respectively. The precision (α) and recall (β) are defined:

$$\alpha(\lambda) = \sum_{\omega \in \Omega} min(\lambda P(\omega), Q(\omega)) \tag{8}$$

$$\beta(\lambda) = \sum_{\omega \in \Omega} min(P(\omega), \frac{Q(\omega)}{\lambda}) \tag{9}$$

The set of pairs $PRD = \{(\alpha(\lambda_i), \beta(\lambda_i))\}$ forms the precision-recall curve (threshold λ is sampled from an equiangular grid). The precision-recall curve is summarized by a variation of the F_1 score: F_β which is able to assign greater importance to precision or recall. Specifically $(F_8, F_{\frac{1}{8}})$ are used for capturing (recall, precision).

The exact numerical precision-recall values are not available in [30], they do provide scatter plots with the $(F_8, F_{\frac{1}{8}})$ pairs of all 800 models trained in [24]. We computed $(F_8, F_{\frac{1}{8}})$ for the models trained using our method as described in the previous section. The scores were computed using the authors' code. For ease of comparison, we overlay our scores over the scatter plots provided in [30]. Our numerical $(F_8, F_{\frac{1}{8}})$ scores are: MNIST $(0.971, 0.979)$, Fashion $(0.985, 0.963)$, CIFAR10 $(0.860, 0.825)$ and CelebA $(0.574, 0.681)$. The results for GLO with sampling by fitting a Gaussian to the learned latent codes (as suggested in [4]) were much worse: MNIST $(0.845, 0.616)$, Fashion $(0.888, 0.594)$, CIFAR10 $(0.693, 0.680)$, CelebA $(0.509, 0.404)$.

From Fig. 2 we can observe that our method generally performs better or competitively to GANs on both precision and recall. On MNIST our method and the best GAN method achieved near-perfect precision-recall. On Fashion our method achieved near perfect precision-recall while the best GAN method lagged behind. On CIFAR10 the performance of our method was also convincingly better than the best GAN model. On CelebA, our method performed well but did not achieve the top performance due to the checkerboard issue described in Sec. 4.2. Overall the performance of our method is typically better or equal to the baselines examined, this is even more impressive in view of the baselines being exhaustively tested over 100 hyperparameter configurations. We also note that our method outperformed VAEs and GLOs very convincingly. This provides evidence that our method is far superior to other generator-based non-adversarial models.

4.3. Qualitative Image Generation Results

We provide qualitative comparisons between our method and the GAN models evaluated by Sajjadi et al. [30] and also show promising results on high-resolution images.

As mentioned above, Sajjadi et al. [30] evaluated 800 different generative models in terms of precision and recall. They provided visual examples of their best performing model (marked as B) for each of the 4 datasets evaluated. In Fig. 3, we provide a visual comparison between random samples generated by our model (without cherry picking) vs. their reported results.

Our method and the best GAN method performed very well on MNIST and Fashion-MNIST. The visual examples are diverse and of high visual quality. On the CIFAR10 dataset, our examples are more realistic than those generated by the best GAN model trained by [24]. On CelebA our generated images are very realistic and with many fewer failed generations, but do suffer from some pixelization (discussed in Sec. 4.1). We note that GANs can generate very high quality faces (e.g. PGGAN [17]), however it appears that for the small architecture used by Lucic et al. and Sajjadi et al., GANs do not generate particularly high-quality facial images.

As a high resolution experiment, we trained GLANN on the CelebA-HQ dataset at 256×256 resolution. We used the network architecture from Mescheder et al [25], with 64 channels, latent code dimensionality of 256 and noise dimension of 100, learning rates of 0.003 for the latent codes and the noise to latent code mapping function, and 0.001 for the generator. We trained for 250 epochs, decayed by 0.5 every 10 epochs.

Figure 3. Comparison of synthesis by IMLE [23], GLO [4], GAN [24], Ours. First row: MNIST, Second row: Fashion, Third row: CIFAR10, Last row: CelebA64. The missing IMLE images were not reported in [23]. The GAN results are taken from [24], corresponding to the best generative model out of 800 as evaluated by the precision-recall metric.

Interpolation examples between two randomly sampled noises are presented in Fig. 4. Our model is able to generate high resolution images. The smooth interpolations illustrate that our model generalizes well to unseen images.

To show the ability of our method to scale to 1024×1024, we present two interpolations at this high resolution in Fig. 5. Note that not all interpolations at such high resolution were successful.

4.4. ModelNet Chair 3D Generation

We present preliminary results for 3D generation on the Chairs category of ModelNet [34]. The generator follows the 3DGAN architecture from [33]. GLANN was trained with ADAM and an L_1 loss. Some GLANN generated 3D samples are presented in Fig. 6.

4.5. Non-Adversarial Unsupervised Image Translation

As generative models are trained in order to be used in downstream tasks, we propose to evaluate generative models by the downstream task of cross domain unsupervised mapping. NAM [15] was proposed by Hoshen and Wolf for unsupervised domain mapping. The method relies on having a strong unconditional generative model of the output image domain. Stronger generative models perform better at this task. This required [15, 12] to use GAN-

Figure 4. Interpolation on CelebA-HQ at 256×256 resolution. The rightmost and leftmost images are randomly sampled from random noise. The interpolation are smooth and of high visual quality.

Figure 5. Interpolation on CelebA-HQ at 1024×1024 resolution.

Figure 6. Examples of 3D chairs generated by GLANN

based unconditional generators. We evaluated our model using the 3 quantitative benchmarks presented in [15] - namely: $MNIST \rightarrow SVHN$, $SVHN \rightarrow MNIST$ and $Car \rightarrow Car$. Our model achieved scores of 31.3%, 25.0% and 1.45 on the three tasks respectively. The results are similar to those obtained using the GAN-based unconditional models (although SVHN is a bit lower here). GLANN is therefore the first model able to achieve fully unsupervised image translation without the use of GANs.

5. Discussion

Loss function: In this work, we replaced the standard adversarial loss function by a perceptual loss. In practice we use ImageNet-trained VGG features. Zhang et al. [38] claimed that self-supervised perceptual losses work no worse than the ImageNet-trained features. It is therefore likely that our method will have similar performance with self-supervised perceptual losses.

Higher resolution: The increase in resolution between 64×64 to 256×256 or 1024×1024 was enabled by a simple modification of the loss function: the perceptual loss was calculated both on the original images, as well as on a bi-linearly subsampled version of the image. Going up to higher resolutions simply requires more sub-sampling levels. Research into more sophisticated perceptual loss will probably yield further improvements in synthesis quality.

Other modalities: In this work we focuses on image synthesis. We believe that our method can extend to many other modalities, particularly 3D and video. The simplicity of the procedure and robustness to hyperparameters makes application to other modalities much simpler than GANs. We showed some evidence for this assertion in Sec. 4.4. One research task for future work is finding good perceptual loss functions for domains outside 2D images.

6. Conclusions

In this paper we introduced a novel non-adversarial method for training generative models. Our method combines ideas from GLO and IMLE and overcomes the weaknesses of both methods. When compared on established benchmarks, our method outperformed the the most common GAN models that underwent exhaustive hyperparameter tuning. Our method is robust and simple to train and achieves excellent results. As future work, we plan to extend this work to higher resolutions and new modalities such as video and 3D.

References

[1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. In *ICLR*, 2017. 2, 3

[2] S. Arora and Y. Zhang. Do gans actually learn the distribution? an empirical study. *arXiv preprint arXiv:1706.08224*, 2017. 1

[3] S. Barratt and R. Sharma. A note on the inception score. *arXiv preprint arXiv:1801.01973*, 2018. 3

[4] P. Bojanowski, A. Joulin, D. Lopez-Paz, and A. Szlam. Optimizing the latent space of generative networks. In *ICML*, 2018. 1, 3, 5, 6, 7

[5] Q. Chen and V. Koltun. Photographic image synthesis with cascaded refinement networks. *ICCV*, 2017. 3

[6] X. Chen, X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel. InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. In *NIPS*. 2016. 4

[7] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 1967. 2

[8] L. Dinh, D. Krueger, and Y. Bengio. Nice: Nonlinear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014. 2

[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014. 1, 2

[10] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein gans. In *NIPS*, 2017. 3

[11] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637, 2017. 3, 4

[12] Y. Hoshen. Non-adversarial mapping with vaes. In *NIPS*, 2018. 3, 7

[13] Y. Hoshen and L. Wolf. An iterative closest point method for unsupervised word translation. *arXiv preprint arXiv:1801.06126*, 2018. 3

[14] Y. Hoshen and L. Wolf. Nam - unsupervised cross-domain image mapping without cycles or gans. In *ICLR Workshop*, 2018. 3

[15] Y. Hoshen and L. Wolf. Nam: Non-adversarial unsupervised domain mapping. In *ECCV*, 2018. 1, 3, 7, 8

[16] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 1

[17] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 6

[18] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *The International Conference on Learning Representations (ICLR)*, 2016. 5

[19] D. P. Kingma and P. Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *arXiv preprint arXiv:1807.03039*, 2018. 2

[20] D. P. Kingma and M. Welling. Auto-encoding variational bayes. In *ICLR*, 2014. 2

[21] A. Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. 5

[22] Y. LeCun and C. Cortes. MNIST handwritten digit database. 2010. 5

[23] K. Li and J. Malik. Implicit maximum likelihood estimation. *arXiv preprint arXiv:1809.09087*, 2018. 1, 3, 4, 7

[24] M. Lucic, K. Kurach, M. Michalski, S. Gelly, and O. Bousquet. Are gans created equal? a large-scale study. *arXiv preprint arXiv:1711.10337*, 2017. 3, 4, 5, 6, 7

[25] L. Mescheder, S. Nowozin, and A. Geiger. Which training methods for gans do actually converge?, booktitle = International Conference on Machine Learning (ICML), year = 2018. 3, 6

[26] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. In *ICLR*, 2018. 3

[27] A. Odena, V. Dumoulin, and C. Olah. Deconvolution and checkerboard artifacts. *Distill*, 1(10):e3, 2016. 5

[28] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016. 2

[29] A. v. d. Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016. 2

[30] M. S. Sajjadi, O. Bachem, M. Lucic, O. Bousquet, and S. Gelly. Assessing generative models via precision and recall. *arXiv preprint arXiv:1806.00035*, 2018. 3, 5, 6

[31] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016. 3

[32] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015. 3

[33] J. Wu, C. Zhang, T. Xue, W. T. Freeman, and J. B. Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *NIPS*, 2016. 7

[34] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015. 7

[35] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 5

[36] S. Yang, P. Luo, C. C. Loy, and X. Tang. From facial parts responses to face detection: A deep learning approach. In *ICCV*, pages 3676–3684, 2015. 5

[37] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena. Self-attention generative adversarial networks. *arXiv preprint arXiv:1805.08318*, 2018. 3

[38] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. *arXiv preprint arXiv:1801.03924*, 2018. 8

[39] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *ICCV*, 2011. 2, 3