

It Takes (Only) Two: Adversarial Generator-Encoder Networks

Dmitry Ulyanov

Skolkovo Institute of Science and Technology, Yandex
dmitry.ulyanov@skoltech.ru

Andrea Vedaldi

University of Oxford
vedaldi@robots.ox.ac.uk

Victor Lempitsky

Skolkovo Institute of Science and Technology
lempitsky@skoltech.ru

Abstract

We present a new autoencoder-type architecture that is trainable in an unsupervised mode, sustains both generation and inference, and has the quality of conditional and unconditional samples boosted by adversarial learning. Unlike previous hybrids of autoencoders and adversarial networks, the adversarial game in our approach is set up directly between the encoder and the generator, and no external mappings are trained in the process of learning. The game objective compares the divergences of each of the real and the generated data distributions with the prior distribution in the latent space. We show that direct generator-vs-encoder game leads to a tight coupling of the two components, resulting in samples and reconstructions of a comparable quality to some recently-proposed more complex architectures.

Introduction

Deep (Variational) Auto Encoders (AEs (Bengio 2009) and VAEs (Kingma and Welling 2014; Rezende, Mohamed, and Wierstra 2014)) and deep Generative Adversarial Networks (GANs (Goodfellow et al. 2014)) are two of the most popular approaches to generative learning. These methods have complementary strengths and weaknesses. VAEs can learn a *bidirectional* mapping between a complex data distribution and a much simpler prior distribution, allowing both generation and inference; on the contrary, the original formulation of GAN learns a *unidirectional* mapping that only allows sampling the data distribution. On the other hand, GANs use more complex loss functions compared to the simplistic data-fitting losses in (V)AEs and can usually generate more realistic samples.

Several recent works have looked for hybrid approaches to support, in a principled way, both sampling and inference like AEs, while producing samples of quality comparable to GANs. Typically this is achieved by training a AE jointly with one or more adversarial discriminators whose purpose is to improve the alignment of distributions in the latent space (Brock et al. 2017; Makhzani et al. 2016), the data space (Che et al. 2017; Larsen, Sønderby, and Winther 2015) or in the joint (product) latent-data space (Donahue,

Krähenbühl, and Darrell 2017; Dumoulin et al. 2017). Alternatively, the method of (Zhu et al. 2016) starts by learning a unidirectional GAN, and then learns a corresponding inverse mapping (the encoder) post-hoc.

While compounding autoencoding and adversarial discrimination does improve GANs and VAEs, it does so at the cost of added complexity. In particular, each of these systems involves at least three deep mappings: an encoder, a decoder/generator, and a discriminator. In this work, we show that this is unnecessary and that the advantages of autoencoders and adversarial training can be combined without increasing the complexity of the model.

In order to do so, we propose a new architecture, called an *Adversarial Generator-Encoder (AGE) Network*, that contains only two feed-forward mappings, the encoder and the generator, operating in opposite directions. As in VAEs, the generator maps a simple prior distribution in latent space to the data space, while the encoder is used to move both the real and generated data samples into the latent space. In this manner, the encoder induces two latent distributions, corresponding respectively to the *encoded real data* and the *encoded generated data*. The AGE learning process then considers the divergence of each of these two distributions to the original prior distribution.

There are two advantages of this approach. First, due to the simplicity of the prior distribution, computing its divergence to the latent data distributions reduces to the calculation of simple statistics over small batches of images. Second, unlike GAN-like approaches, real and generated distributions are never compared directly, thus bypassing the need for discriminator networks as used by GANs. Instead, the adversarial signal in AGE comes from learning the encoder to increase the divergence between the latent distribution of the generated data and the prior, which works against the generator, which tries to decrease the same divergence (Figure 1). Optionally, AGE training may include reconstruction losses typical of AEs.

The AGE approach is evaluated on a number of standard image datasets, where we show that the quality of generated samples is comparable to that of GANs (Goodfellow et al. 2014; Radford, Metz, and Chintala 2016), and the quality of reconstructions is comparable or better to that of the more

complex Adversarially-Learned Inference (ALI) approach of (Dumoulin et al. 2017), while training faster. We further evaluate the AGE approach in the conditional setting, where we show that it can successfully tackle the colorization problem that is known to be difficult for GAN-based approaches.

Other related work. Apart from the above-mentioned approaches, AGE networks can be related to several other recent GAN-based systems. Thus, they are related to improved GANs (Salimans et al. 2016) that proposed to use batch-level information in order to prevent mode collapse. The divergences within AGE training are also computed as batch-level statistics.

Another avenue for improving the stability of GANs has been the replacement of the classifying discriminator with the regression-based one as in energy-based GANs (Zhao, Mathieu, and LeCun 2017) and Wasserstein GANs (Arjovsky, Chintala, and Bottou 2017). Our statistics (the divergence from the prior distribution) can be seen as a very special form of regression. In this way, the encoder in the AGE architecture can be (with some reservations) seen as a discriminator computing a single number similarly to how it is done in (Arjovsky, Chintala, and Bottou 2017; Zhao, Mathieu, and LeCun 2017).

Adversarial Generator-Encoder Networks

This section introduces our Adversarial Generator-Encoder (AGE) networks. An AGE is composed of two parametric mappings: the *encoder* $e_\psi(\mathbf{x})$, with the learnable parameters ψ , that maps the data space \mathcal{X} to the latent space \mathcal{Z} , and the *generator* $g_\theta(\mathbf{z})$, with the learnable parameters θ , which runs in the opposite direction. We will use the shorthand notation $f(Y)$ to denote the distribution of the random variable $f(\mathbf{y})$, $\mathbf{y} \sim Y$.

The reference distribution Z is chosen so that it is easy to sample from it, which in turns allow to sample $g_\theta(Z)$ unconditionally be first sampling $\mathbf{z} \sim Z$ and then by feed-forward evaluation of $\mathbf{x} = g_\theta(\mathbf{z})$, exactly as it is done in GANs. In our experiments, we pick the latent space \mathcal{Z} to be an M -dimensional sphere \mathbb{S}^M , and the latent distribution to be a uniform distribution on that sphere $Z = \text{Uniform}(\mathbb{S}^M)$. We have also conducted some experiments with the unit Gaussian distribution in the Euclidean space and have obtained results comparable in quality.

The goal of learning an AGE is to align the real data distribution X to the generated distribution $g_\theta(Z)$ while establishing a correspondence between data and latent samples \mathbf{x} and \mathbf{z} . The real data distribution X is empirical and represented by a large number N of data samples $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N\}$. Learning amounts to tuning the parameter ψ and θ to optimize the AGE criterion, discussed in the next section. This criterion is based on an adversarial game whose saddle points correspond to networks that align real and generated data distribution ($g(Z) = X$). The criterion is augmented with additional terms that encourage the reciprocity of the encoder e and the generator g .

Adversarial distribution alignment

The GAN approach to aligning two distributions is to define an adversarial game based on a ratio of probabilities (Goodfel-

low et al. 2014). The ratio is estimated by repeatedly fitting a binary classifier that distinguishes between samples obtained from the real and generated data distributions. Here, we propose an alternative adversarial setup with some advantages with respect to GAN’s, including avoiding generator collapse (Goodfellow 2017).

The goal of AGE is to generate a distribution $g(Z)$ in data space that is close to the true data distribution X . However, direct matching of the distributions in the high-dimensional data space, as done in GAN, can be challenging. We propose instead to move this comparison *to the simpler latent space*. This is done by introducing a divergence measure $\Delta(P\|Q)$ between distributions defined in the latent space \mathcal{Z} . We only require this divergence to be non-negative and zero if, and only if, the distributions are identical ($\Delta(P\|Q) = 0 \iff P = Q$).¹ The encoder function e maps the distributions X and $g(Z)$ defined in data space to corresponding distributions $e(X)$ and $e(g(Z))$ in the latent space. Below, we show how to design an adversarial criterion such that minimizing the divergence $\Delta(e(X), e(g(Z)))$ in latent space induces the distributions X and $g(Z)$ to align in data space as well.

In the theoretical analysis below, we assume that encoders and decoders span the class of all measurable mappings between the corresponding spaces. This assumption, often referred to as *non-parametric limit*, is justified by the universality of neural networks (Hornik, Stinchcombe, and White 1989). We further make the **assumption** that there exists at least one “perfect” generator that matches the data distribution, i.e. $\exists g_0 : g_0(Z) = X$.

We start by considering a simple game with objective defined as:

$$\max_e \min_g V_1(g, e) = \Delta(e(g(Z))\|e(X)). \quad (1)$$

As the following theorem shows, perfect generators form saddle points (Nash equilibria) of the game (1) and all saddle points of the game (1) are based on perfect generators.

Theorem 1. *A pair (g^*, e^*) forms a saddle point of the game (1) if and only if the generator g^* matches the data distribution, i.e. $g^*(Z) = X$.*

The proofs of this and the following theorems are given in the supplementary material.

While the game (1) is sufficient for aligning distributions in the data space, finding such saddle points is difficult due to the need of comparing two empirical (hence non-parametric) distributions $e(X)$ and $e(g(Z))$. We can avoid this issue by introducing an intermediate reference distribution Y and comparing the distributions to that instead, resulting in the game:

$$\max_e \min_g V_2(g, e) = \Delta(e(g(Z))\|Y) - \Delta(e(X)\|Y). \quad (2)$$

Importantly, (2) still induces alignment of real and generated distributions in data space:

Theorem 2. *If a pair (g^*, e^*) is a saddle point of game (2) then the generator g^* matches the data distribution, i.e. $g^*(Z) = X$. Conversely, if the generator g^* matches the data*

¹We do not require the divergence to be a distance.

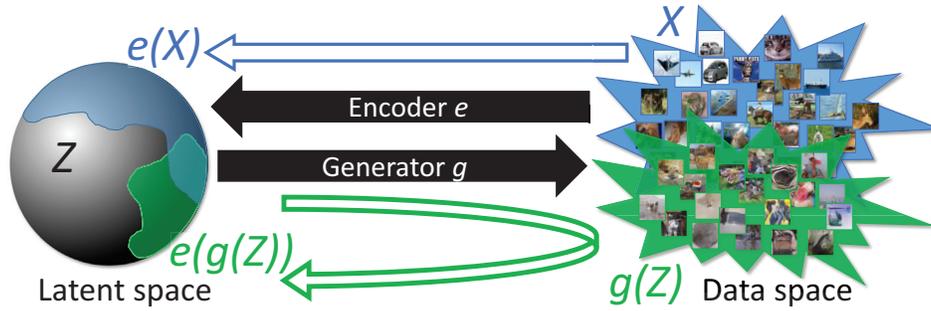


Figure 1: Our model (AGE network) has only two components: the generator g and the encoder e . The learning process adjusts their parameters in order to align a simple prior distribution Z in the latent space and the data distribution X . This is done by adversarial training, as updates for the generator aim to minimize the divergence between $e(g(Z))$ and Z (aligning green with gray), while updates for the encoder aim to minimize the divergence between $e(X)$ (aligning blue with gray) and to *maximize* the divergence between $e(g(Z))$ and Z (shrink green “away” from gray). We demonstrate that such adversarial learning gives rise to high-quality generators that result in the close match between the real distribution X and the generated distribution $g(Z)$. Our learning can also incorporate reconstruction losses to ensure that encoder-generator acts as autoencoder.

distribution, then for some e^* the pair (g^*, e^*) is a saddle point of (2).

The important benefit of formulation (2) is that, if Y is selected in a suitable manner, it is simple to compute the divergence of Y to the empirical distributions $e(g(Z))$ and $e(X)$. For convenience, in particular, we choose Y to coincide with the “canonical” (prior) distribution Z . By substituting $Y = Z$ in objective (2), the loss can be extended to include reconstruction terms that can improve the quality of the result. It can also be optimized by using stochastic approximations as described in section .

Given a distribution Q in data space, the encoder e and divergence $\Delta(\cdot\|Y)$ can be interpreted as extracting statistics $F(Q) = \Delta(e(Q)\|Y)$ from Q . Hence, game (2) can be thought of as comparing certain statistics of the real and generated data distributions. Similarly to GANs, these statistics are not fixed but evolve during learning.

We also note that, even away from the saddle point, the minimization $\min_g V_2(g, e)$ for a fixed e does not tend to collapse for many reasonable choice of divergence (e.g. KL-divergence). In fact, any collapsed distribution would inevitably lead to a very high value of the first term in (2). Thus, unlike GANs, our approach can optimize the generator for a fixed adversary till convergence and obtain a non-degenerate solution. On the other hand, the maximization $\max_e V_2(g, e)$ for some fixed g can lead to $+\infty$ score for some divergences.

Encoder-generator reciprocity and reconstruction losses

In the previous section we have demonstrated that finding a saddle point of (2) is sufficient to align real and generated data distributions X and $g(Z)$ and thus generate realistically-looking data samples. At the same time, this by itself does not necessarily imply that mappings e and g are reciprocal. Reciprocity, however, can be desirable if one wishes to reconstruct samples $\mathbf{x} = g(\mathbf{z})$ from their codes $\mathbf{z} = e(\mathbf{x})$.

In this section, we introduce losses that encourage encoder and generator to be reciprocal. Reciprocity can be measured

either in the latent space or in the data space, resulting in the loss functions based on reconstruction errors, e.g.:

$$L_X(g_\theta, e_\psi) = \mathbb{E}_{\mathbf{x} \sim X} \|\mathbf{x} - g_\theta(e_\psi(\mathbf{x}))\|_1, \quad (3)$$

$$L_Z(g_\theta, e_\psi) = \mathbb{E}_{\mathbf{z} \sim Z} \|\mathbf{z} - e_\psi(g_\theta(\mathbf{z}))\|_2^2. \quad (4)$$

Both losses (3) and (4) thus encourage the reciprocity of the two mappings. Note also that (3) is the traditional pixelwise loss used within AEs (L1-loss was preferred, as it is known to perform better in image synthesis tasks with deep architectures).

A natural question then is whether it is helpful to minimize both losses (3) and (4) at the same time or whether considering only one is sufficient. The answer is given by the following statement:

Theorem 3. *Let the two distributions W and Q be aligned by the mapping f (i.e. $f(W) = Q$) and let $\mathbb{E}_{\mathbf{w} \sim W} \|\mathbf{w} - h(f(\mathbf{w}))\|_2^2 = 0$. Then, for $\mathbf{w} \sim W$ and $\mathbf{q} \sim Q$, we have $\mathbf{w} = h(f(\mathbf{w}))$ and $\mathbf{q} = f(h(\mathbf{q}))$ almost certainly, i.e. the mappings f and h invert each other almost everywhere on the supports of W and Q . Furthermore, Q is aligned with W by h , i.e. $h(Q) = W$.*

Recall that Theorem 2 establishes that the solution (saddle point) of game (2) aligns distributions in the data space. Then Theorem 3 shows that when augmented with the latent space loss (4), the objective (2) is sufficient to ensure reciprocity.

Training AGE networks

Based on the theoretical analysis derived in the previous subsections, we now suggest the approach to the joint training of the generator in the encoder within the AGE networks. As in the case of GAN training, we set up the learning process for an AGE network as a game with the iterative updates over the parameters θ and ψ that are driven by the optimization of different objectives. In general, the optimization process combines the maximin game for the functional (2) with the optimization of the reciprocity losses (3) and (4).

In particular, we use the following game objectives for the generator and the encoder:

$$\hat{\theta} = \arg \min_{\theta} (V_2(g_{\theta}, e_{\bar{\psi}}) + \lambda L_{\mathcal{Z}}(g_{\theta}, e_{\bar{\psi}})) , \quad (5)$$

$$\hat{\psi} = \arg \max_{\psi} (V_2(g_{\bar{\theta}}, e_{\psi}) - \mu L_{\mathcal{X}}(g_{\bar{\theta}}, e_{\psi})) , \quad (6)$$

where $\bar{\psi}$ and $\bar{\theta}$ denote the value of the encoder and generator parameters at the moment of the optimization and λ, μ is a user-defined parameter. Note that both objectives (5), (6) include only one of the reconstruction losses. Specifically, the generator objective includes only the latent space reconstruction loss. In the experiments, we found that the omission of the other reconstruction loss (in the data space) is important to avoid possible blurring of the generator outputs that is characteristic to autoencoders. Similarly to GANs, in (5), (6) we perform only several steps toward optimum at each iteration, thus alternating between generator and encoder updates.

By maximizing the difference between $\Delta(e_{\psi}(g_{\bar{\theta}}(Z))\|Z)$ and $\Delta(e_{\psi}(X)\|Z)$, the optimization process (6) focuses on the maximization of the mismatch between the real data distribution X and the distribution of the samples from the generator $g_{\bar{\theta}}(Z)$. Informally speaking, the optimization (6) forces the encoder to find the mapping that aligns real data distribution X with the target distribution Z , while mapping non-real (synthesized data) $g_{\bar{\theta}}(Z)$ away from Z . When Z is a uniform distribution on a sphere, the goal of the encoder would be to uniformly spread the real data over the sphere, while cramping as much of synthesized data as possible together assuring non-uniformity of the distribution $e_{\psi}(g_{\bar{\theta}}(Z))$.

Any differences (misalignment) between the two distributions are thus amplified by the optimization process (6) and force the optimization process (5) to focus specifically on removing these differences. Since the misalignment between X and $g(Z)$ is measured after projecting the two distributions into the latent space, the maximization of this misalignment makes the encoder to compute features that distinguish the two distributions.

Experiments

We have validated AGE networks in two settings. A more traditional setting involves *unconditional* generation and reconstruction, where we consider a number of standard image datasets. We have also evaluated AGE networks in the *conditional* setting. Here, we tackle the problem of image colorization, which is hard for GANs. In this setting, we condition both the generator and the encoder on the grayscale image. Taken together, our experiments demonstrate the versatility of the AGE approach.

Unconditionally-trained AGE networks

Network architectures: In our experiments, the generator and the encoder networks have a similar structure to the generator and the discriminator in DCGAN (Radford, Metz, and Chintala 2016). To turn the discriminator into the encoder, we have modified it to output an M -dimensional vector and replaced the final sigmoid layer with the normalization layer that projects the points onto the sphere.

Divergence measure: As we need to measure the divergence between the empirical distribution and the prior distribution in the latent space, we have used the following measure. Given a set of samples on the M -dimensional sphere, we fit the Gaussian Normal distribution with diagonal covariance matrix in the embedding M -dimensional space and we compute the KL-divergence of such Gaussian with the unit Gaussian as

$$\text{KL}(Q\|\mathcal{N}(0, I)) = -\frac{1}{2} + \frac{1}{M} \sum_{j=1}^M \frac{s_j^2 + m_j^2}{2} - \log(s_j) \quad (7)$$

where m_j and s_j are the means and the standard deviations of the fitted Gaussians along various dimensions. Since the uniform distribution on the sphere will entail the lowest possible divergence with the unit Gaussian in the embedding space among all distributions on the unit sphere, such divergence measure is valid for our analysis above. We have also tried to measure the same divergence non-parametrically using Kozachenko-Leonenko estimator (Kozachenko and Leonenko 1987). In our initial experiments, both versions worked equally well, and we used a simpler parametric estimator in the presented experiments.

Hyper-parameters: We use ADAM (Kingma and Ba 2015) optimizer with the learning rate of 0.0002. We perform two generator updates per one encoder update for all datasets. For each dataset we tried $\lambda \in \{500, 1000, 2000\}$ and picked the best one. We ended up using $\mu = 10$ for all datasets. The dimensionality M of the latent space was manually set according to the complexity of the dataset. We thus used $M = 64$ for CelebA and SVHN datasets, and $M = 128$ for the more complex datasets of Tiny ImageNet and CIFAR-10.

Results: We evaluate unconditional AGE networks on several standard datasets, while treating the system (Dumoulin et al. 2017) as the most natural reference for comparison (as the closest three-component counterpart to our two-component system). The results for (Dumoulin et al. 2017) are either reproduced with the author’s code or copied from (Dumoulin et al. 2017).

In Figure 2, we present the results on the challenging Tiny ImageNet dataset (Russakovsky et al. 2015) and the SVHN dataset (Netzer et al. 2011). We show both samples $g(\mathbf{z})$ obtained for $\mathbf{z} \sim Z$ as well as the reconstructions $g(e(\mathbf{x}))$ alongside the real data samples \mathbf{x} . We also show the reconstructions obtained by (Dumoulin et al. 2017) for comparison. Inspection reveals that the fidelity of (Dumoulin et al. 2017) is considerably lower for Tiny ImageNet dataset.

In Figure 3, we further compare the reconstructions of CelebA (Liu et al. 2015) images obtained by the AGE network, ALI (Dumoulin et al. 2017), and VAE (Kingma and Welling 2014). Overall, the fidelity and the visual quality of AGE reconstructions are roughly comparable or better than ALI. Furthermore, ALI takes notoriously longer time to converge than our method, and the reconstructions of ALI after 10 epochs (which take six hours) of training look considerably worse than AGE reconstructions after 10 epochs (which take only two hours), thus attesting to the benefits of having a simpler two-component system.

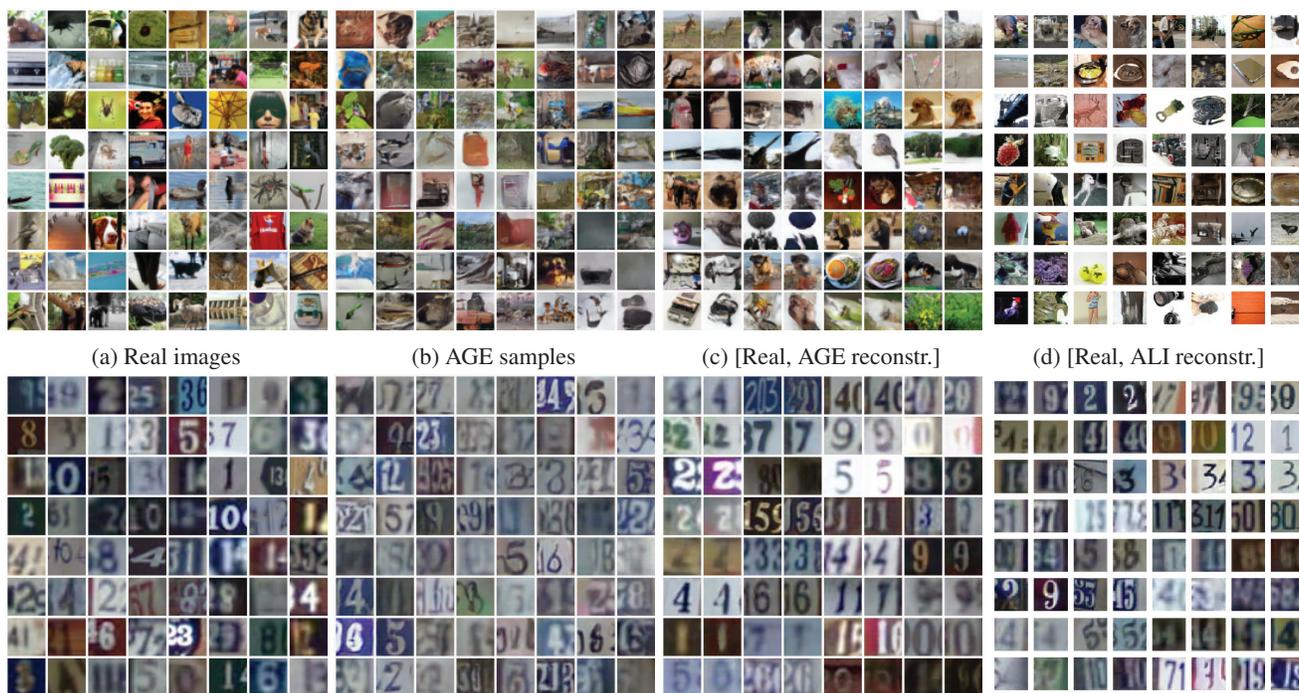


Figure 2: Samples (b) and reconstructions (c) for Tiny ImageNet dataset (top) and SVHN dataset (bottom). The results of ALI (Dumoulin et al. 2017) on the same datasets are shown in (d). In (c,d) odd columns show real examples and even columns show their reconstructions. Qualitatively, our method seems to obtain more accurate reconstructions than ALI (Dumoulin et al. 2017), especially on the Tiny ImageNet dataset, while having samples of similar visual quality.

Next we evaluate our method quantitatively. For the model trained on CIFAR-10 dataset we compute Inception score (Salimans et al. 2016). The AGE score is 5.90 ± 0.04 , which is higher than the ALI (Dumoulin et al. 2017) score of 5.34 ± 0.05 (as reported in (Warde-Farley and Bengio 2017)) and than the score of 4.36 ± 0.04 from (Salimans et al. 2016). The state-of-the-art from (Warde-Farley and Bengio 2017) is higher still (7.72 ± 0.13). Qualitative results of AGE for CIFAR-10 and other datasets are shown in supplementary material.

We also computed log likelihood for AGE and ALI on the MNIST dataset using the method of (Wu et al. 2016) with latent space of size 10 using authors source code. ALIs score is 721 while AGEs score is 746. The AGE model is also superior than both VAE and GAN, which scores are 705.375 and 328.772 respectively as evaluated by (Wu et al. 2016).

Finally, similarly to (Dumoulin et al. 2017; Donahue, Krähenbühl, and Darrell 2017; Radford, Metz, and Chintala 2016) we investigated whether the learned features are useful for discriminative tasks. We reproduced the evaluation pipeline from (Dumoulin et al. 2017) for SVHN dataset and obtained 23.7% error rate in the unsupervised feature learning protocol with our model, while their result is 19.14%. At the moment, it is unclear to us why AGE networks underperform ALI at this task.

Conditional AGE network experiments

Recently, several GAN-based systems have achieved very impressive results in the conditional setting, where the latent space is augmented or replaced with a second data space corresponding to different modality (Isola et al. 2017; Zhu et al. 2017). Arguably, it is in the conditional setting where the bi-directionality lacking in conventional GANs is most needed. In fact, by allowing to switch back-and-forth between the data space and the latent space, bi-directionality allows powerful neural image editing interfaces (Zhu et al. 2016; Brock et al. 2017).

Here, we demonstrate that AGE networks perform well in the conditional setting. To show that, we have picked the image colorization problem, which is known to be hard for GANs. To the best of our knowledge, while the idea of applying GANs to the colorization task seems very natural, the only successful GAN-based colorization results were presented in (Isola et al. 2017), and we compare to the authors' implementation of their pix2pix system. We are also aware of several unsuccessful efforts to use GANs for colorization.

To use AGE for colorization, we work with images in the *Lab* color space, and we treat the *ab* color channels of an image as a data sample \mathbf{x} . We then use the lightness channel L of the image as an input to both the encoder $e_{\psi}(\mathbf{x}|L)$ and the generator $g_{\theta}(\mathbf{z}|L)$, effectively conditioning the encoder and the generator on it. Thus, different latent variables \mathbf{z} will result in different colorizations \mathbf{x} for the same grayscale image L . The latent space in these experiments is taken to be

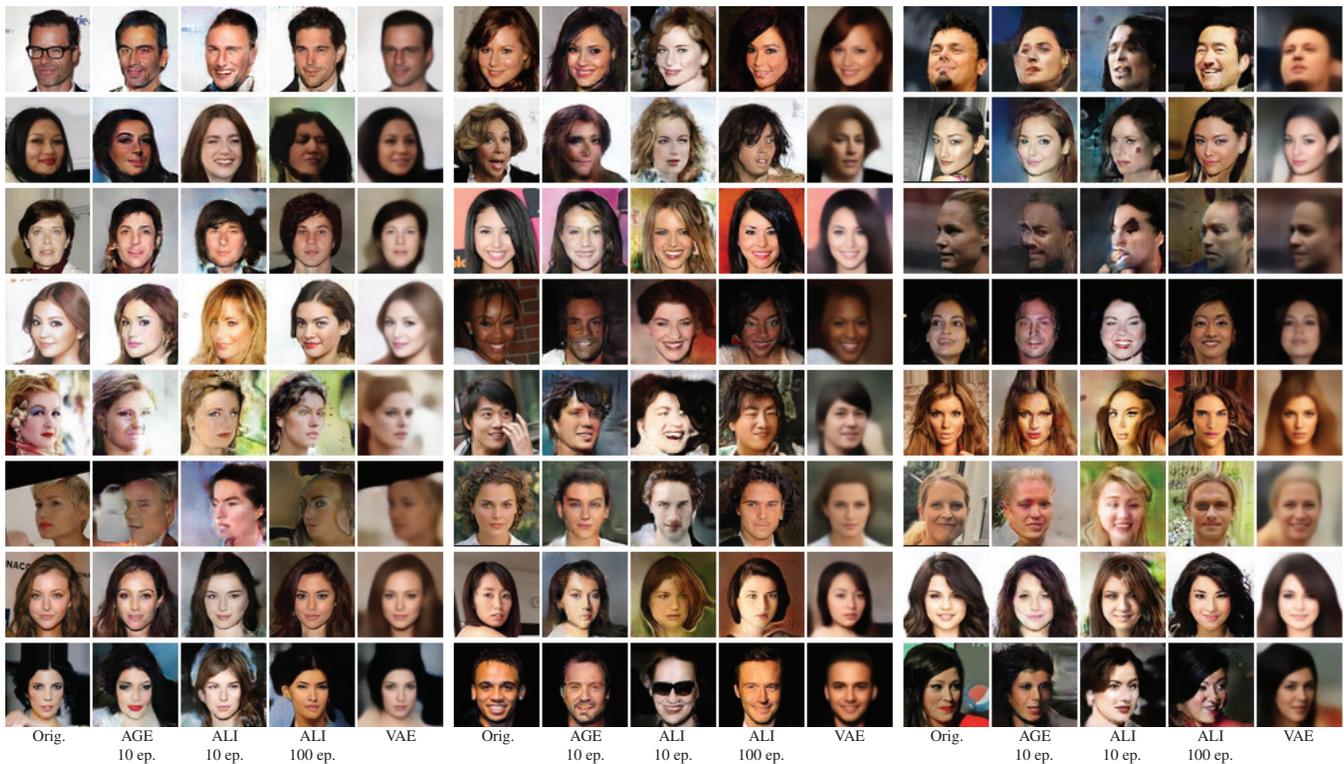


Figure 3: Reconstruction quality comparison of our method with ALI (Dumoulin et al. 2017) and VAE (Kingma and Welling 2014). The first column in each set shows examples from the test set of CelebA dataset. In the other columns the reconstructions for different methods are presented: column two for ours method, three and four for ALI and five for VAE. We train our model for 10 epochs and compare to ALI, trained for the same number of epochs (column three). Importantly one epoch for our method takes 3 times less time than for ALI. For a fair comparison we also present the results of ALI, trained till convergence.

three-dimensional.

The particular architecture of the generator takes the input image L , augments it with z variables expanded to constant maps of the same spatial dimensions as L , and then applies the ResNet type architecture (He et al. 2016; Johnson, Alahi, and Fei-Fei 2016) that computes x (i.e. the ab -channels). The encoder architecture is a convolutional network that maps the concatenation of L and x (essentially, an image in the Lab-space) to the latent space. The divergence measure is the same as in the unconditional AGE experiments and is computed “unconditionally” (i.e. each minibatch passed through the encoder combines multiple images with different L).

We perform the colorization experiments on Stanford Cars dataset (Krause et al. 2013) with 16,000 training images of 196 car models, since cars have inherently ambiguous colors and hence their colorization is particularly prone to the regression-to-mean effect. The images were downsampled to 64×64 .

We present colorization results in Figure 4. Crucially, AGE generator is often able to produce plausible and diverse colorizations for different latent vector inputs. As we wanted to enable pix2pix GAN-based system of (Isola et al. 2017) to produce diverse colorizations, we augmented the input to their generator architecture with three constant-valued maps (same as in our method). We however found that their system

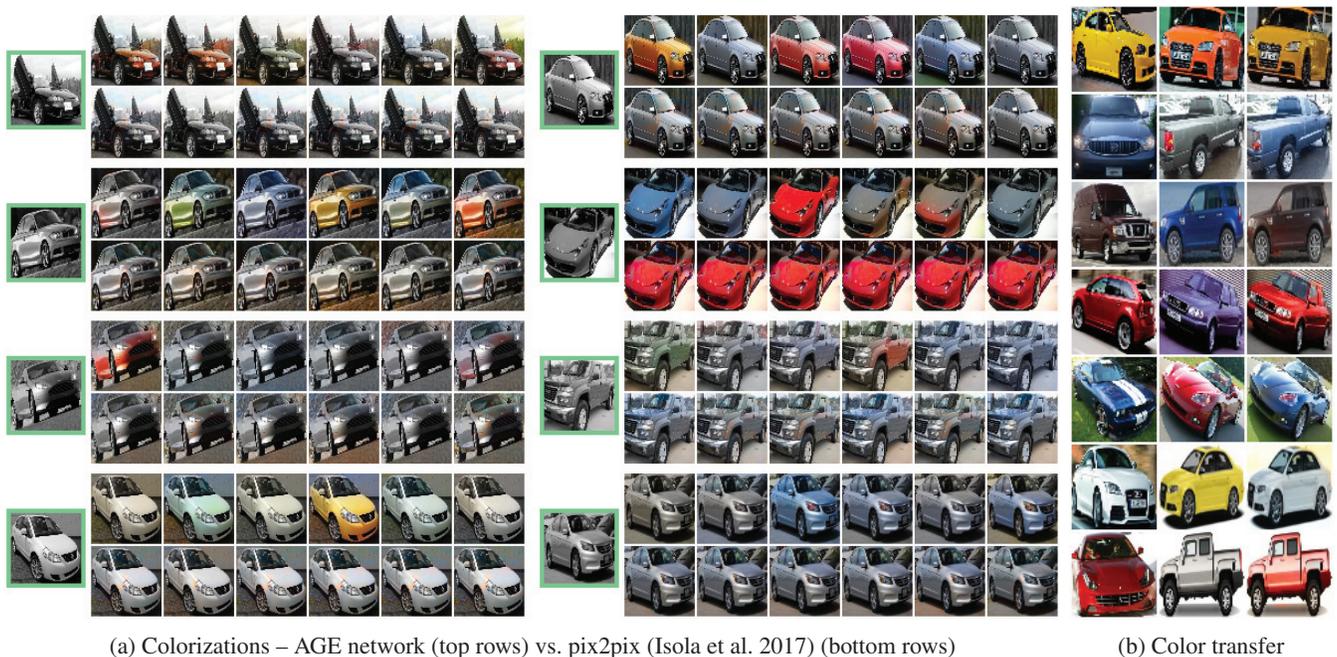
effectively learns to ignore such input augmentation and the diversity of colorizations was very low (Figure 4a).

To demonstrate the meaningfulness of the latent space learned by the conditional AGE training, we also demonstrate the color transfer examples, where the latent vector $z_1 = e_\psi(x_1|L_1)$ obtained by encoding the image $[x_1, L_1]$ is then used to colorize the grayscale image L_2 , i.e. $x_2 = g_\theta(z_1|L_2)$ (Figure 4b).

Conclusion

We have introduced a new approach for simultaneous learning of generation and inference networks. We have demonstrated how to set up such learning as an adversarial game between generation and inference, which has a different type of objective from traditional GAN approaches. In particular the objective of the game considers divergences between distributions rather than discrimination at the level of individual samples. As a consequence, our approach does not require training a discriminator network and enjoys relatively quick convergence.

We demonstrate that on a range of standard datasets, the generators obtained by our approach provides high-quality samples, and that the reconstructions of real data samples passed subsequently through the encoder and the generator are of better fidelity than in (Dumoulin et al. 2017). We have



(a) Colorizations – AGE network (top rows) vs. pix2pix (Isola et al. 2017) (bottom rows)

(b) Color transfer

Figure 4: (a) Each pane shows colorizations of the input grayscale image (emphasized) using conditional AGE networks (top rows) and pix2pix (Isola et al. 2017) with added noise maps (bottom rows). AGE networks produce diverse colorizations, which are hard to obtain using pix2pix. (b) In each row we show the result of color transfer using the conditional AGE network. The color scheme from the first image is transferred onto the second image.

also shown that our approach is able to generate plausible and diverse colorizations, which is not possible with the GAN-based system (Isola et al. 2017).

Our approach leaves a lot of room for further experiments. In particular, a more complex latent space distribution can be chosen as in (Makhzani et al. 2016), and other divergence measures between distributions can be easily tried.

Acknowledgments

This study was supported by grants 14.756.31.0001 and ERC 677195-IDIU.

References

- Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein GAN. *Proc. ICLR*.
- Bengio, Y. 2009. Learning deep architectures for AI. *Foundations and Trends in Machine Learning* 2(1):1–127.
- Brock, A.; Lim, T.; Ritchie, J. M.; and Weston, N. 2017. Neural photo editing with introspective adversarial networks. *Proc. ICLR*.
- Che, T.; Li, Y.; Jacob, A. P.; Bengio, Y.; and Li, W. 2017. Mode regularized generative adversarial networks. *Proc. ICLR*.
- Donahue, J.; Krähenbühl, P.; and Darrell, T. 2017. Adversarial feature learning. *Proc. ICLR*.
- Dumoulin, V.; Belghazi, I.; Poole, B.; Lamb, A.; Arjovsky, M.; Mastropietro, O.; and Courville, A. C. 2017. Adversarially learned inference. *Proc. ICLR*.
- Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A. C.; and Bengio, Y. 2014. Generative adversarial nets. In *Proc. NIPS*, 2672–2680.
- Goodfellow, I. J. 2017. NIPS 2016 tutorial: Generative adversarial networks. *CoRR* abs/1701.00160.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proc. CVPR*, 770–778.
- Hornik, K.; Stinchcombe, M.; and White, H. 1989. Multilayer feedforward networks are universal approximators. *Neural Networks* 2(5):359 – 366.
- Isola, P.; Zhu, J.-Y.; Zhou, T.; and Efros, A. A. 2017. Image-to-image translation with conditional adversarial networks. In *Proc. CVPR*.
- Johnson, J.; Alahi, A.; and Fei-Fei, L. 2016. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*.
- Kingma, D. P., and Ba, J. 2015. Adam: A method for stochastic optimization. *Proc. ICLR*.
- Kingma, D. P., and Welling, M. 2014. Auto-encoding variational bayes. *Proc. ICLR*.
- Kozachenko, L. F., and Leonenko, N. N. 1987. Sample estimate of the entropy of a random vector. *Probl. Inf. Transm.* 23(1-2):95–101.
- Krause, J.; Stark, M.; Deng, J.; and Fei-Fei, L. 2013. 3d object representations for fine-grained categorization. In *Proc. ICCV 3DRR Workshop*, 554–561.
- Larsen, A. B. L.; Sønderby, S. K.; and Winther, O. 2015.

Autoencoding beyond pixels using a learned similarity metric. *CoRR* abs/1512.09300.

Liu, Z.; Luo, P.; Wang, X.; and Tang, X. 2015. Deep learning face attributes in the wild. In *ICCV*, 3730–3738. IEEE Computer Society.

Makhzani, A.; Shlens, J.; Jaitly, N.; and Goodfellow, I. J. 2016. Adversarial autoencoders. *Proc. ICLR*.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; and Ng, A. Y. 2011. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*.

Radford, A.; Metz, L.; and Chintala, S. 2016. Unsupervised representation learning with deep convolutional generative adversarial networks. *Proc. ICLR*.

Rezende, D. J.; Mohamed, S.; and Wierstra, D. 2014. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M. S.; Berg, A. C.; and Li, F. 2015. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision* 115(3):211–252.

Salimans, T.; Goodfellow, I. J.; Zaremba, W.; Cheung, V.; Radford, A.; and Chen, X. 2016. Improved techniques for training gans. In *Advances in Neural Information Processing Systems (NIPS)*, 2226–2234.

Warde-Farley, D., and Bengio, Y. 2017. Improving generative adversarial networks with denoising feature matching. In *Proc. ICLR*.

Wu, Y.; Burda, Y.; Salakhutdinov, R.; and Grosse, R. B. 2016. On the quantitative analysis of decoder-based generative models. *CoRR* abs/1611.04273.

Zhao, J. J.; Mathieu, M.; and LeCun, Y. 2017. Energy-based generative adversarial network. *Proc. ICLR*.

Zhu, J.-Y.; Krähenbühl, P.; Shechtman, E.; and Efros, A. A. 2016. Generative visual manipulation on the natural image manifold. In *Proc. ECCV*.

Zhu, J.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR* abs/1703.10593.