Supplementary Material for Adversarial Variational Bayes: Unifying Variational Autoencoders and Generative Adversarial Networks

Lars Mescheder [1] Sebastian Nowozin [2] Andreas Geiger [1 3]

Abstract

In the main text we derived Adversarial Variational Bayes (AVB) and demonstrated its usefulness both for black-box Variational Inference and for learning latent variable models. This document contains proofs that were omitted in the main text as well as some further details about the experiments and additional results.

I. Proofs

This section contains the proofs that were omitted in the main text.

The derivation of AVB in Section 3.1 relies on the fact that we have an explicit representation of the optimal discriminator $T^*(x,z)$. This was stated in the following Proposition:

Proposition 1. *For $p_\theta(x \mid z)$ and $q_\phi(z \mid x)$ fixed, the optimal discriminator T^* according to the objective in* (3.3) *is given by*

$$T^*(x,z) = \log q_\phi(z \mid x) - \log p(z). \tag{3.4}$$

Proof. As in the proof of Proposition 1 in Goodfellow et al. (2014), we rewrite the objective in (3.3) as

$$\int \big(p_{\mathcal{D}}(x) q_\phi(z \mid x) \log \sigma(T(x,z)) + p_{\mathcal{D}}(x) p(z) \log(1 - \sigma(T(x,z)))\big) \mathrm{d}x \mathrm{d}z. \tag{I.1}$$

This integral is maximal as a function of $T(x,z)$ if and only if the integrand is maximal for every (x,z). However, the function

$$t \mapsto a \log(t) + b \log(1-t) \tag{I.2}$$

attains its maximum at $t = \frac{a}{a+b}$, showing that

$$\sigma(T^*(x,z)) = \frac{q_\phi(z \mid x)}{q_\phi(z \mid x) + p(z)} \tag{I.3}$$

or, equivalently,

$$T^*(x,z) = \log q_\phi(z \mid x) - \log p(z). \tag{I.4}$$

□

To apply our method in practice, we need to obtain unbiased gradients of the ELBO. As it turns out, this can be achieved by taking the gradients w.r.t. a fixed optimal discriminator. This is a consequence of the following Proposition:

Proposition 2. *We have*

$$\mathrm{E}_{q_\phi(z|x)}\left(\nabla_\phi T^*(x,z)\right) = 0. \tag{3.6}$$

Proof. By Proposition 1,

$$\mathrm{E}_{q_\phi(z|x)}\left(\nabla_\phi T^*(x,z)\right) = \mathrm{E}_{q_\phi(z|x)}\left(\nabla_\phi \log q_\phi(z \mid x)\right). \tag{I.5}$$

For an arbitrary family of probability densities q_ϕ we have

$$\begin{aligned}\mathrm{E}_{q_\phi}\left(\nabla_\phi \log q_\phi\right) &= \int q_\phi(z) \frac{\nabla_\phi q_\phi(z)}{q_\phi(z)} \mathrm{d}z \\ &= \nabla_\phi \int q_\phi(z) \mathrm{d}z = \nabla_\phi 1 = 0.\end{aligned} \tag{I.6}$$

Together with (8.5), this implies (3.6). □

In Section 3.3 we characterized the Nash-equilibria of the two-player game defined by our algorithm. The following Proposition shows that in the nonparametric limit for $T(x,z)$ any Nash-equilibrium defines a global optimum of the variational lower bound:

Proposition 3. *Assume that T can represent any function of two variables. If (θ^*, ϕ^*, T^*) defines a Nash-equilibrium of the two-player game defined by* (3.3) *and* (3.7)*, then*

$$T^*(x,z) = \log q_{\phi^*}(z \mid x) - \log p(z) \tag{3.8}$$

and (θ^*, ϕ^*) is a global optimum of the variational lower bound in (2.4).

Proof. If (θ^*, ϕ^*, T^*) defines a Nash-equilibrium, Proposition 1 shows (3.8). Inserting (3.8) into (3.5) shows that (ϕ^*, θ^*) maximizes

$$\mathrm{E}_{p_{\mathcal{D}}(x)} \mathrm{E}_{q_\phi(z|x)}\big(-\log q_{\phi^*}(z \mid x) + \log p(z) + \log p_\theta(x \mid z)\big) \tag{I.7}$$

as a function of ϕ and θ. A straightforward calculation shows that (8.7) is equal to

$$\mathcal{L}(\theta, \phi) + \mathrm{E}_{p_{\mathcal{D}}(x)} \mathrm{KL}(q_\phi(z \mid x), q_{\phi^*}(z \mid x)) \tag{I.8}$$

where

$$\begin{aligned}\mathcal{L}(\theta, \phi) := \mathrm{E}_{p_{\mathcal{D}}(x)} \Big[& - \mathrm{KL}(q_\phi(z \mid x), p(z)) \\ & + \mathrm{E}_{q_\phi(z \mid x)} \log p_\theta(x \mid z) \Big]\end{aligned} \tag{I.9}$$

is the variational lower bound in (2.4).

Notice that (8.8) evaluates to $\mathcal{L}(\theta^*, \phi^*)$ when we insert (θ^*, ϕ^*) for (θ, ϕ).

Assume now, that (θ^*, ϕ^*) does not maximize the variational lower bound $\mathcal{L}(\theta, \phi)$. Then there is (θ', ϕ') with

$$\mathcal{L}(\theta', \phi') > \mathcal{L}(\theta^*, \phi^*). \tag{I.10}$$

Inserting (θ', ϕ') for (θ, ϕ) in (8.8) we obtain

$$\mathcal{L}(\theta', \phi') + \mathrm{E}_{p_{\mathcal{D}}(x)} \mathrm{KL}(q_{\phi'}(z \mid x), q_{\phi^*}(z \mid x)), \tag{I.11}$$

which is strictly bigger than $\mathcal{L}(\theta^*, \phi^*)$, contradicting the fact that (θ^*, ϕ^*) maximizes (8.8). Together with (3.8), this proves the theorem. □

II. Adaptive Contrast

In Section 4 we derived a variant of AVB that contrasts the current inference model with an adaptive distribution rather than the prior. This leads to Algorithm 2. Note that we do not consider the $\mu^{(k)}$ and $\sigma^{(k)}$ to be functions of ϕ and therefore do not backpropagate gradients through them.

Algorithm 2 Adversarial Variational Bayes with Adaptive Constrast (AC)

```
$i \leftarrow 0$
while not converged do
    Sample $\{x^{(1)}, \dots, x^{(m)}\}$ from data distrib. $p_{\mathcal{D}}(x)$
    Sample $\{z^{(1)}, \dots, z^{(m)}\}$ from prior $p(z)$
    Sample $\{\epsilon^{(1)}, \dots, \epsilon^{(m)}\}$ from $\mathcal{N}(0, 1)$
    Sample $\{\eta^{(1)}, \dots, \eta^{(m)}\}$ from $\mathcal{N}(0, 1)$
    for $k = 1, \dots, m$ do
        $z_\phi^{(k)}, \mu^{(k)}, \sigma^{(k)} \leftarrow \mathrm{encoder}_\phi(x^{(k)}, \epsilon^{(k)})$
        $\bar{z}_\phi^{(k)} \leftarrow \frac{z_\phi^{(k)} - \mu^{(k)}}{\sigma^{(k)}}$
    end for
    Compute $\theta$-gradient (eq. 3.7):
    $g_\theta \leftarrow \frac{1}{m} \sum_{k=1}^m \nabla_\theta \log p_\theta \left( x^{(k)}, z_\phi^{(k)} \right)$
    Compute $\phi$-gradient (eq. 3.7):
    $g_\phi \leftarrow \frac{1}{m} \sum_{k=1}^m \nabla_\phi \big[ -T_\psi \left( x^{(k)}, \bar{z}_\phi^{(k)} \right) + \frac{1}{2} \| \bar{z}_\phi^{(k)} \|^2 + \log p_\theta \left( x^{(k)}, z_\phi^{(k)} \right) \big]$
    Compute $\psi$-gradient (eq. 3.3) :
    $g_\psi \leftarrow \frac{1}{m} \sum_{k=1}^m \nabla_\psi \left[ \log \left( \sigma(T_\psi(x^{(k)}, \bar{z}_\phi^{(k)}) \right) + \log \left( 1 - \sigma(T_\psi(x^{(k)}, \eta^{(k)})) \right) \right]$
    Perform SGD-updates for $\theta$, $\phi$ and $\psi$:
    $\theta \leftarrow \theta + h_i\, g_\theta, \quad \phi \leftarrow \phi + h_i\, g_\phi, \quad \psi \leftarrow \psi + h_i\, g_\psi$
    $i \leftarrow i + 1$
end while
```

III. Architecture for MNIST-experiment

To apply Adaptive Contrast to our method, we have to be able to efficiently estimate the moments of the current inference model $q_\phi(z \mid x)$. To this end, we propose a network architecture like in Figure 8. The final output z of the network is a linear combination of basis noise vectors where the coefficients depend on the data point x, i.e.

$$z_k = \sum_{i=1}^m v_{i,k}(\epsilon_i) a_{i,k}(x). \tag{III.1}$$

The noise basis vectors $v_i(\epsilon_i)$ are defined as the output of small fully-connected neural networks f_i acting on normally-distributed random noise ϵ_i, the coefficient vec-

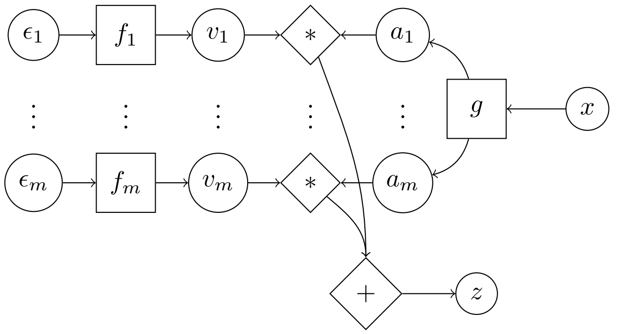


Figure 8. Architecture of the network used for the MNIST-experiment

tors $a_i(x)$ are defined as the output of a deep convolutional neural network g acting on x.

The moments of the z_i are then given by

$$\mathrm{E}(z_k) = \sum_{i=1}^{m} \mathrm{E}[v_{i,k}(\epsilon_i)] a_{i,k}(x). \tag{III.2}$$

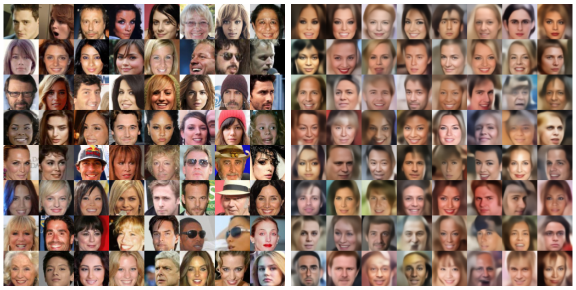
$$\mathrm{Var}(z_k) = \sum_{i=1}^{m} \mathrm{Var}[v_{i,k}(\epsilon_i)] a_{i,k}(x)^2. \tag{III.3}$$

By estimating $\mathrm{E}[v_{i,k}(\epsilon_i)]$ and $\mathrm{Var}[v_{i,k}(\epsilon_i)]$ via sampling once per mini-batch, we can efficiently compute the moments of $q_\phi(z \mid x)$ for all the data points x in a single mini-batch.

IV. Additional Experiments

celebA We also used AVB (without AC) to train a deep convolutional network on the celebA-dataset (Liu et al., 2015) for a 64-dimensional latent space with $\mathcal{N}(0, 1)$-prior. For the decoder and adversary we use two deep convolutional neural networks acting on x like in Radford et al. (2015). We add the noise ϵ and the latent code z to each hidden layer via a learned projection matrix. Moreover, in the encoder and decoder we use three RESNET-blocks (He et al., 2015) at each scale of the neural network. We add the log-prior $\log p(z)$ explicitly to the adversary $T(x, z)$, so that it only has to learn the log-density of the inference model $q_\phi(z \mid x)$.

The samples for celebA are shown in Figure 9. We see that our model produces visually sharp images of faces. To demonstrate that the model has indeed learned an abstract representation of the data, we show reconstruction results and the result of linearly interpolating the z-vector in the latent space in Figure 10. We see that the reconstructions are reasonably sharp and the model produces realistic images for all interpolated z-values.



(a) Training data (b) Random samples

Figure 9. Independent samples for a model trained on celebA.

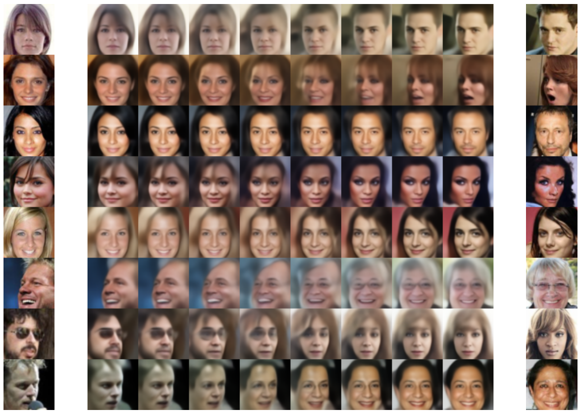


Figure 10. Interpolation experiments for celebA

MNIST To evaluate how AVB with adaptive contrast compares against other methods on a fixed decoder architecture, we reimplemented the methods from Maaløe et al. (2016) and Kingma et al. (2016). The method from Maaløe et al. (2016) tries to make the variational approximation to the posterior more flexible by using auxiliary variables, the method from Kingma et al. (2016) tries to improve the variational approximation by employing an Inverse Autoregressive Flow (IAF), a particularly flexible instance of a normalizing flow (Rezende & Mohamed, 2015). In our experiments, we compare AVB with adaptive contrast to a standard VAE with diagonal Gaussian inference model as well as the methods from Maaløe et al. (2016) and Kingma et al. (2016).

In our first experiment, we evaluate all methods on training a decoder that is given by a fully-connected neural network with ELU-nonlinearities and two hidden layers with 300 units each. The prior distribution $p(z)$ is given by a 32-dimensional standard-Gaussian distribution.

The results are shown in Table 3a. We observe, that both AVB and the VAE with auxiliary variables achieve a better (approximate) ELBO than a standard VAE. When evaluated using AIS, both methods result in similar log-

	ELBO	AIS	reconstr. error
AVB + AC	$\approx -85.1 \pm 0.2$	-83.7 ± 0.3	59.3 ± 0.2
VAE	-88.9 ± 0.2	-85.0 ± 0.3	62.2 ± 0.2
auxiliary VAE	-88.0 ± 0.2	-83.8 ± 0.3	62.1 ± 0.2
VAE + IAF	-88.9 ± 0.2	-84.9 ± 0.3	62.3 ± 0.2

(a) fully-connected decoder ($\dim(z) = 32$)

	ELBO	AIS	reconstr. error
AVB + AC	$\approx -93.8 \pm 0.2$	-89.7 ± 0.3	76.4 ± 0.2
VAE	-94.9 ± 0.2	-89.9 ± 0.4	76.7 ± 0.2
auxiliary VAE	-95.0 ± 0.2	-89.7 ± 0.3	76.8 ± 0.2
VAE + IAF	-94.4 ± 0.2	-89.7 ± 0.3	76.1 ± 0.2

(b) convolutional decoder ($\dim(z) = 8$)

	ELBO	AIS	reconstr. error
AVB + AC	$\approx -82.7 \pm 0.2$	-81.7 ± 0.3	57.0 ± 0.2
VAE	-85.7 ± 0.2	-81.9 ± 0.3	59.4 ± 0.2
auxiliary VAE	-85.6 ± 0.2	-81.6 ± 0.3	59.6 ± 0.2
VAE + IAF	-85.5 ± 0.2	-82.1 ± 0.4	59.6 ± 0.2

(c) convolutional decoder ($\dim(z) = 32$)

likelihoods. However, AVB results in a better reconstruction error than an auxiliary variable VAE and a better (approximate) ELBO. We observe that our implementation of a VAE with IAF did not improve on a VAE with diagonal Gaussian inference model. We suspect that this due to optimization difficulties.

In our second experiment, we train a decoder that is given by the shallow convolutional neural network described in Salimans et al. (2015) with 800 units in the last fully-connected hidden layer. The prior distribution $p(z)$ is given by either a 8-dimensional or a 32-dimensional standard-Gaussian distribution.

The results are shown in Table 3b and Table 3c. Even though AVB achieves a better (approximate) ELBO and a better reconstruction error for a 32-dimensional latent space, all methods achieve similar log-likelihoods for this decoder-architecture, raising the question if strong inference models are always necessary to obtain a good generative model. Moreover, we found that neither auxiliary variables nor IAF did improve the ELBO. Again, we believe this is due to optimization challenges.