Adversarial Autoencoders

Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow
Google Brain

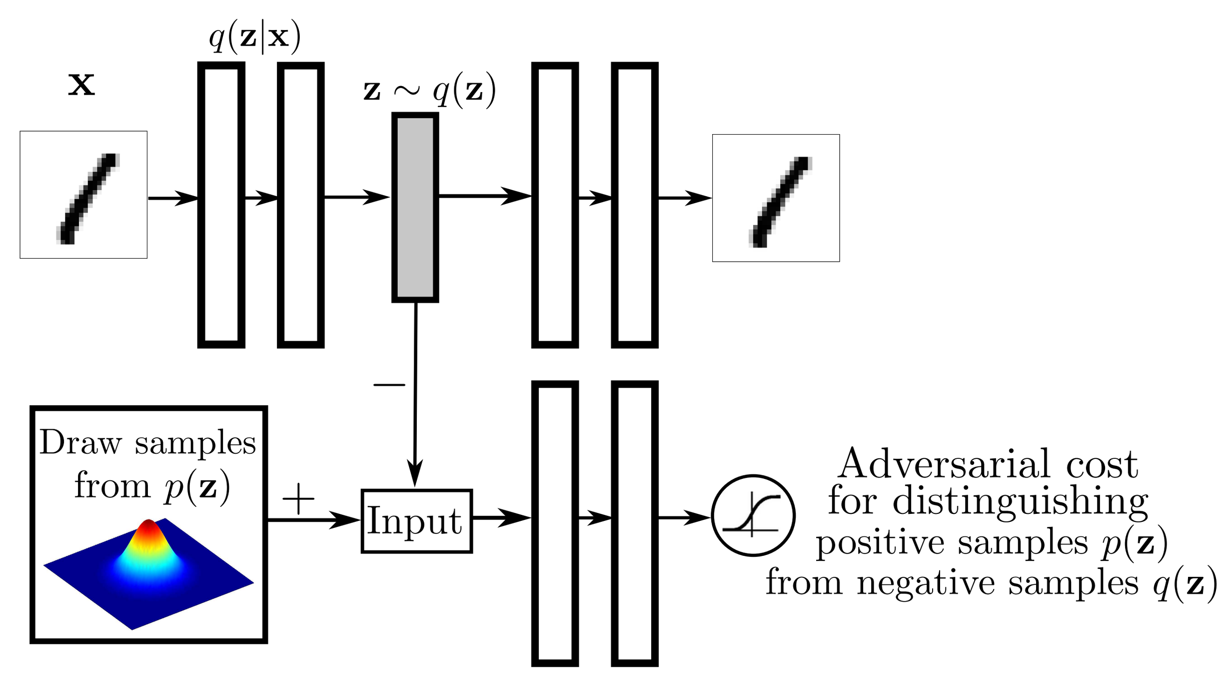
ICLR 2016

Adversarial Autoencoders (AAE)

We propose a new method for regularizing autoencoders by imposing an arbitrary prior on the latent representation of the autoencoder with GAN framework.

Training has two stages:

- The autoencoder updates the encoder and the decoder to minimize the reconstruction error.
- The adversarial network first updates its discriminator to tell the apart the true samples from the generated samples then updates its generator to confuse the discriminator.



Relationship to VAE

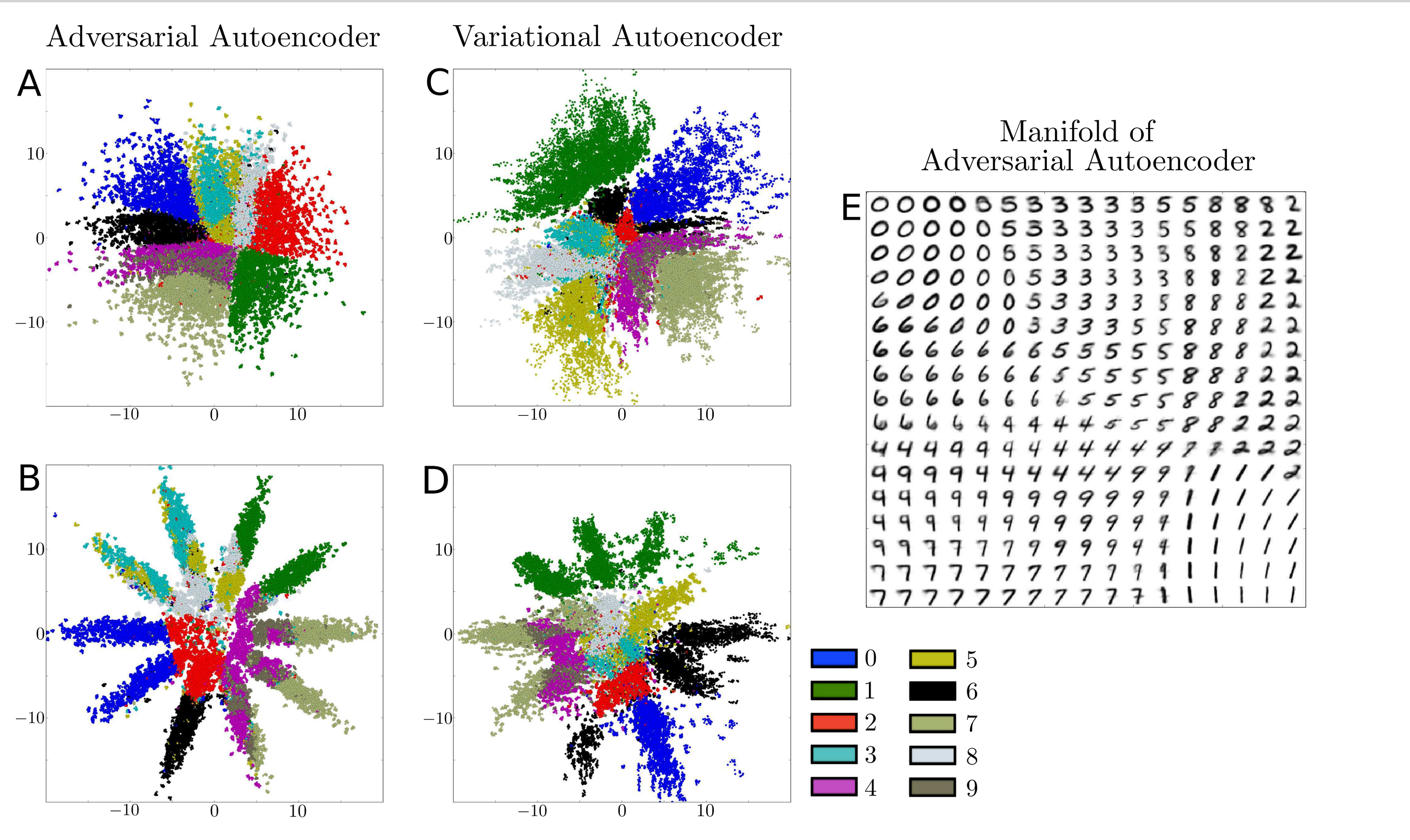


Figure 1: (a) 2-D Gaussian code of AAE (b) mixture of 10 2-D Gaussian code of AAE (c) 2-D Gaussian code of VAE (d) mixture of 10 2-D Gaussian code of VAE (e) AAE manifold.

Semi-Supervised AAE

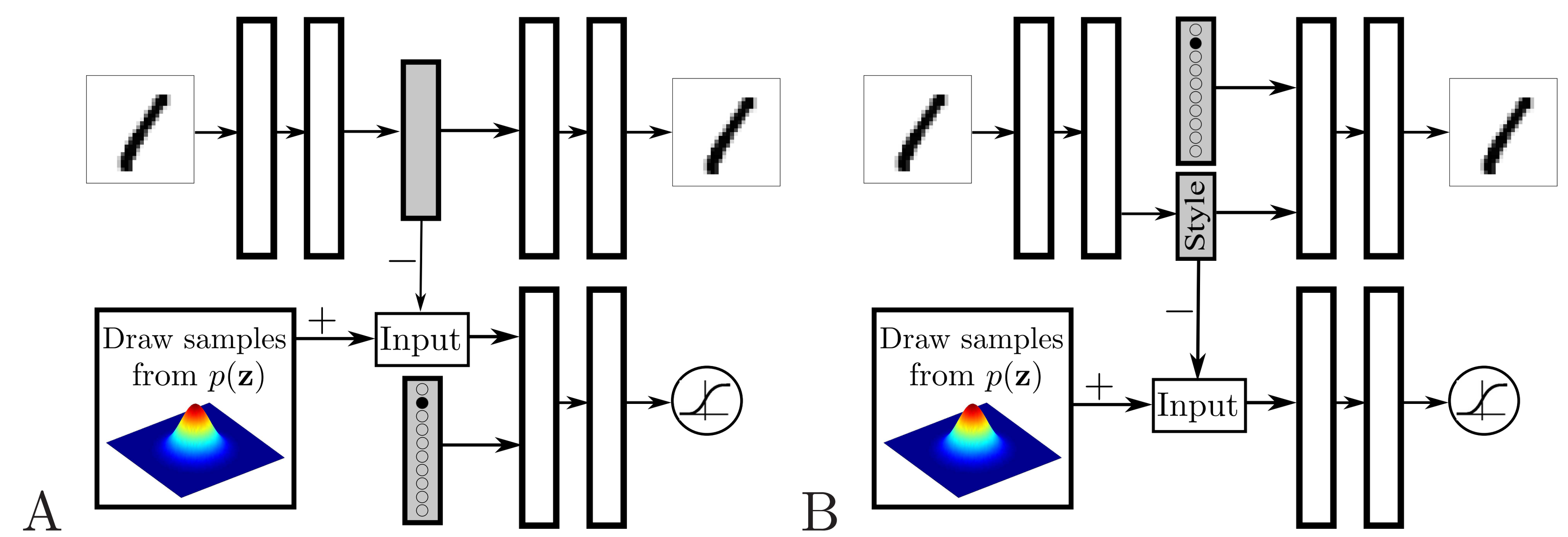


Figure 2: Two methods for semi-supervised learning with AAE (a) Regularizing the hidden code by providing a one-hot vector to the discriminative network. (b) Disentangling the label information from the hidden code by providing the one-hot vector to the generative model.

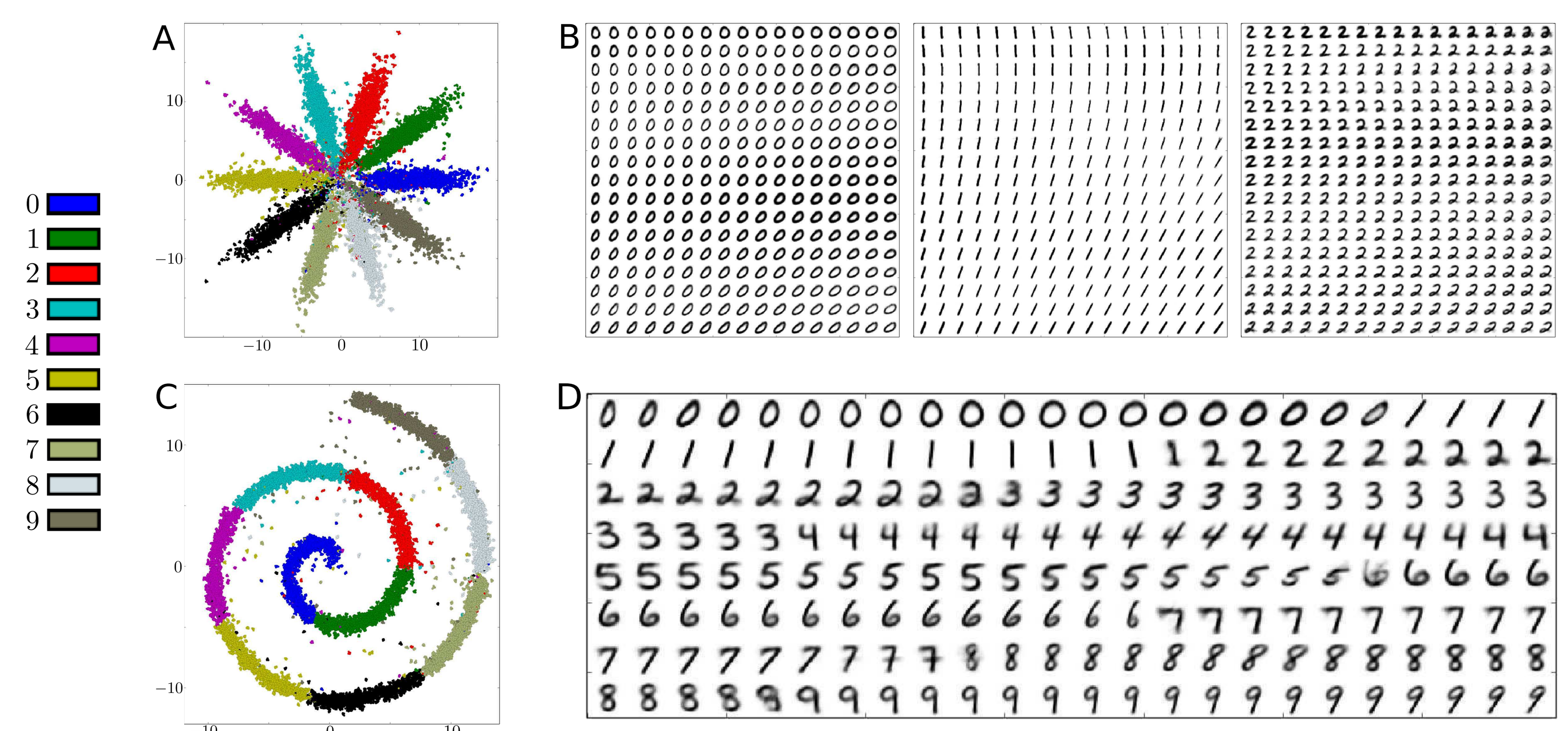


Figure 3: Leveraging label information to better regularize the hidden code. (a,b) Training the coding space to match a mixture of 10 2-D Gaussians. (c,d) Same but for a swiss roll distribution.

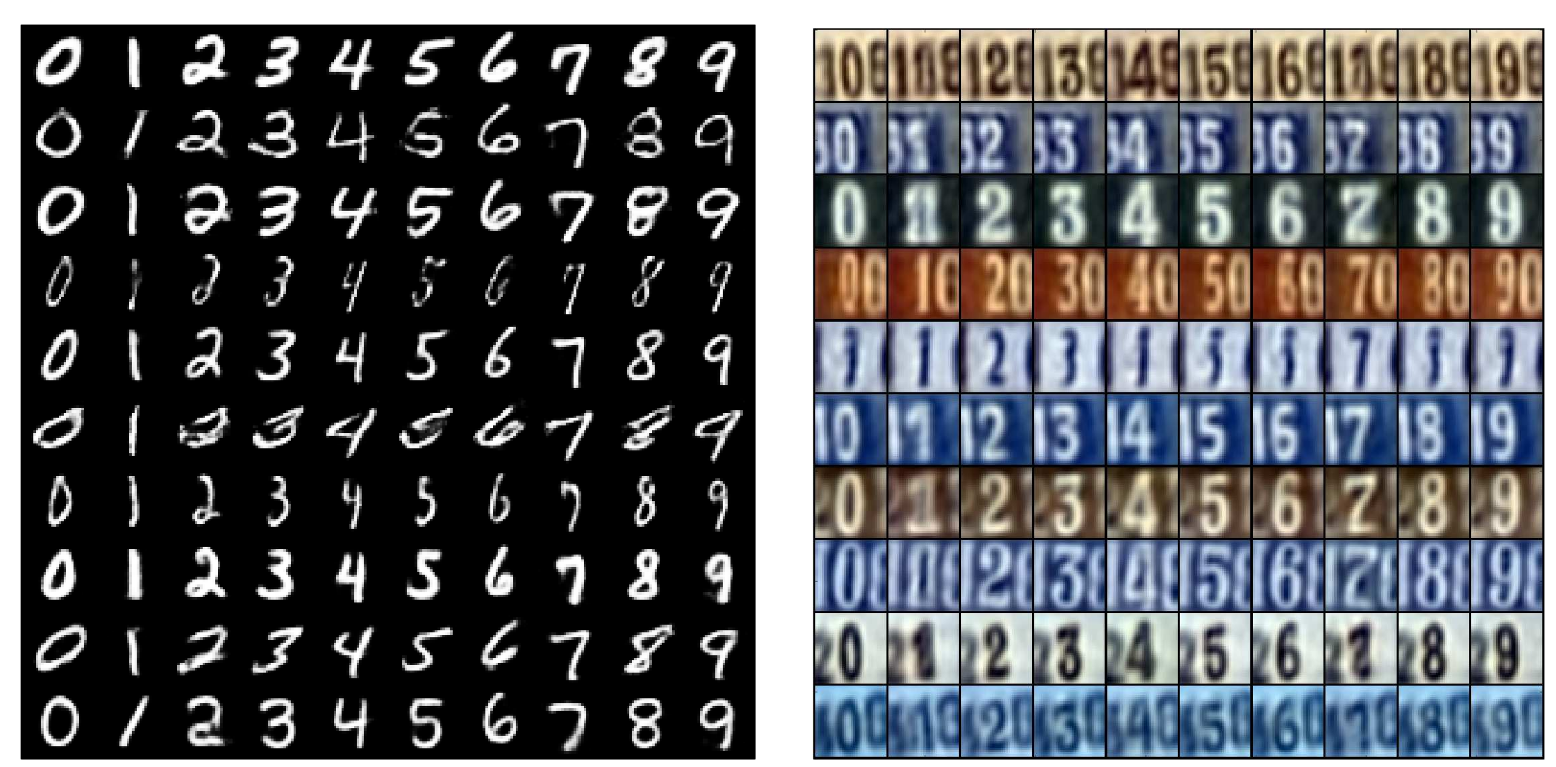


Figure 4: Disentangling content and style, MNIST and SVHN.

Likelihood Evaluation

Figure 5: Samples generated from an adversarial autoencoder trained on MNIST and TFD. The last column shows the closest training images in pixel-wise Euclidean distance to those in the second-to-last column.

	MNIST (10K)	MNIST (10M)	TFD (10K)	TFD (10M)
DBN	138 ± 2	-	1909 ± 66	-
Stacked CAE	121 ± 1.6	-	2110 ± 50	-
Deep GSN	214 ± 1.1	-	1890 ± 29	-
GAN	225 ± 2	386	2057 ± 26	-
GMMN + AE	282 ± 2	-	2204 ± 20	-
AAE	$\mathbf{340 \pm 2}$	**427**	$\mathbf{2252 \pm 16}$	**2522**

Table 1: Log-likelihood of test data on MNIST and Toronto Face dataset. Higher values are better. On both datasets we report the Parzen window estimate of the log-likelihood obtained by drawing 10K or 10M samples from the trained model.

Acknowledgments

We would like to thank Ilya Sutskever, Oriol Vinyals, Jon Gauthier, Sam Bowman and other members of the Google Brain team for helpful discussions. We also thank the developers of TensorFlow, which we used for all of our experiments.