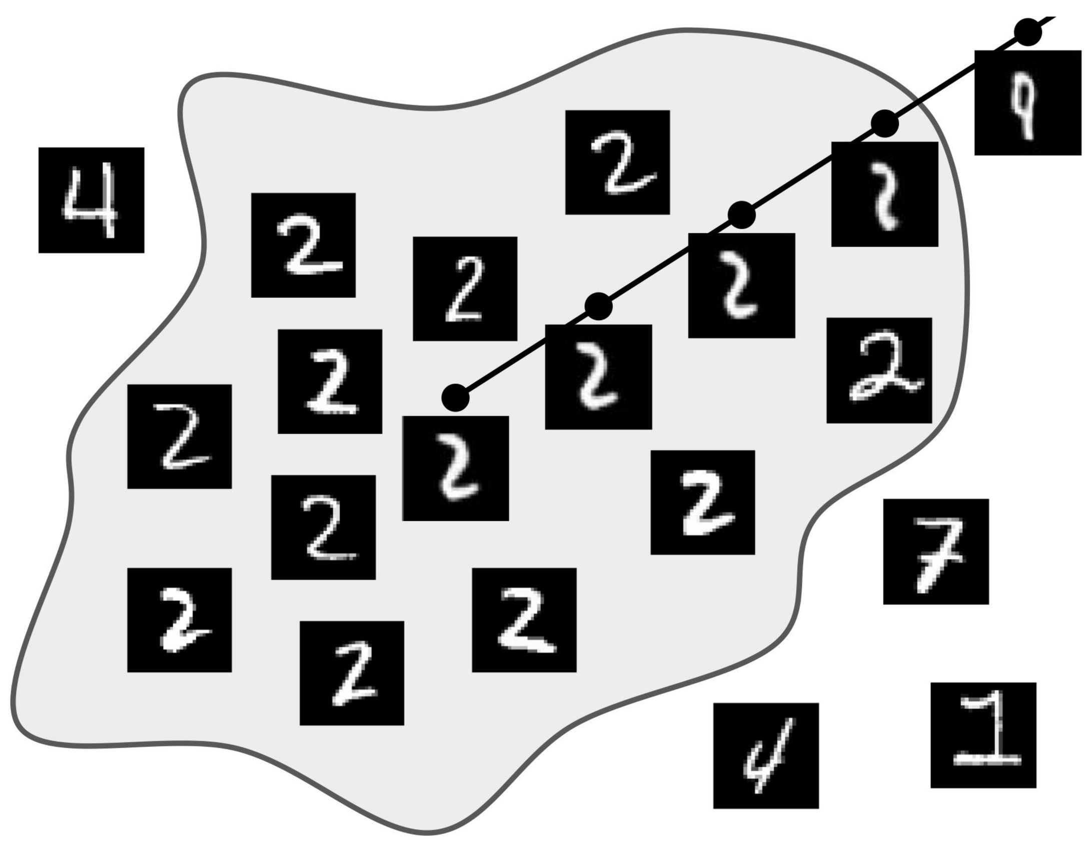
Understanding and Improving Interpolation in Autoencoders via an Adversarial Regularizer

David Berthelot*, Colin Raffel*, Aurko Roy, and **Ian Goodfellow** (*equal contribution)

Abstract

Autoencoders provide a powerful framework for learning compressed representations by encoding all of the information needed to reconstruct a data point in a latent code. In some cases, autoencoders can "interpolate": By decoding the convex combination of the latent codes for two datapoints, the autoencoder can produce an output which semantically mixes characteristics from the datapoints. In this paper, we propose a regularization procedure which encourages interpolated outputs to appear more realistic by fooling a critic network which has been trained to recover the mixing coefficient from interpolated data. We then develop a simple benchmark task where we can quantitatively measure the extent to which various autoencoders can interpolate and show that our regularizer dramatically improves interpolation in this setting. We also demonstrate empirically that **our regularizer produces latent codes which are more effective on downstream tasks, suggesting a possible link between interpolation abilities and learning useful representations.**

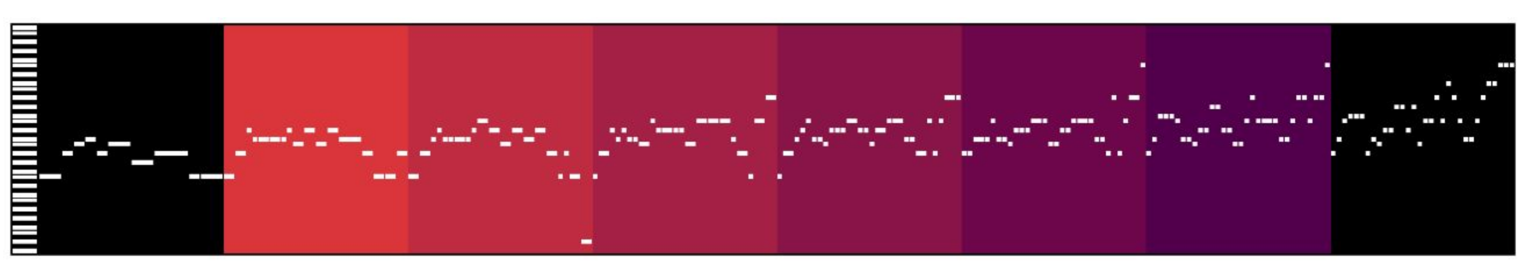
Interpolation and representations



Interpolation as a benchmark

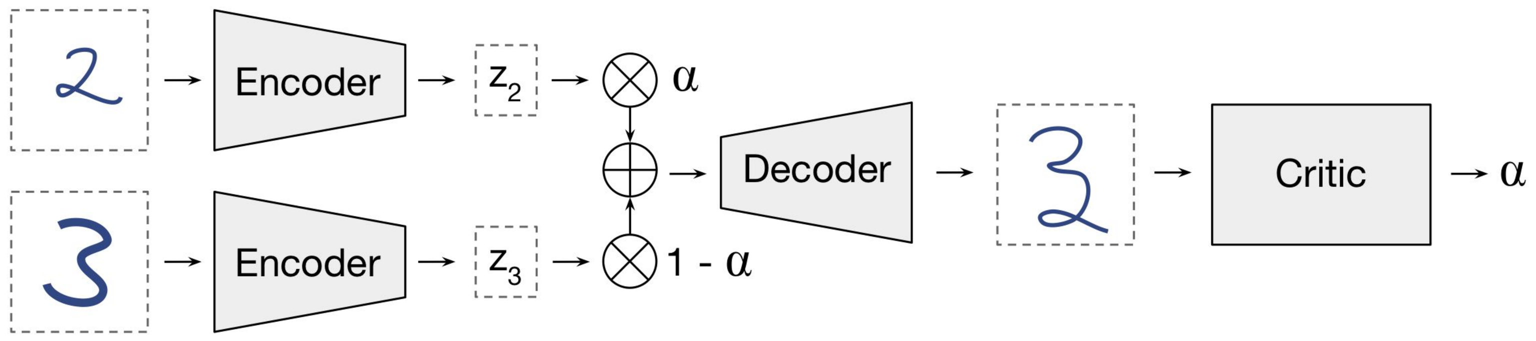
Radford et al., *"Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks"*

Ha & Eck, *"A Neural Representation of Sketch Drawings"*



Roberts et al., *"A Hierarchical Latent Vector Model for Learning Long-Term Structure in Music"*

ACAI

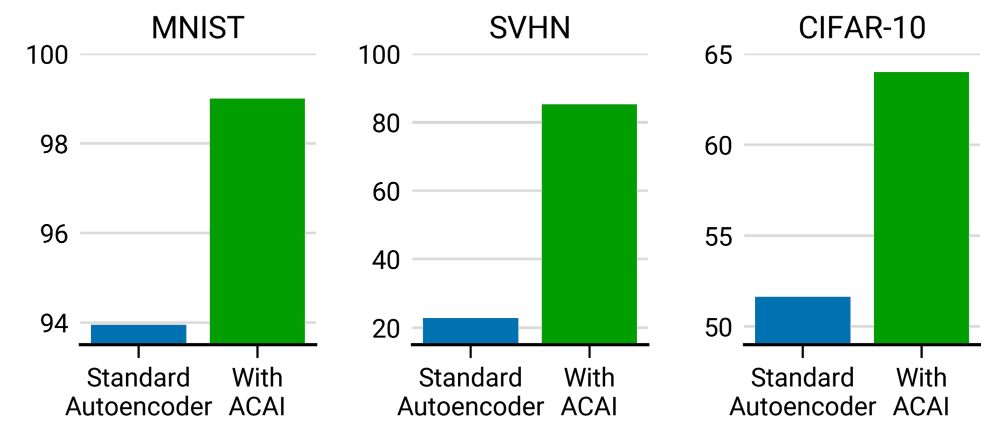


$$\hat{x}_\alpha = g_\phi(\alpha z_1 + (1-\alpha) z_2)$$

$$\mathcal{L}_d = \|d_\omega(\hat{x}_\alpha) - \alpha\|^2 + \|d_\omega(\gamma x + (1-\gamma)\hat{x})\|^2$$

$$\mathcal{L}_{f,g} = \|x - g_\phi(f_\theta(x))\|^2 + \lambda \|d_\omega(\hat{x}_\alpha)\|^2$$

Representation learning



Dataset	d_z	Baseline	Denoising	VAE	AAE	VQ-VAE	ACAI
MNIST	32	94.90±0.14	96.00±0.27	96.56±0.31	70.74±3.27	97.50±0.18	**98.25±0.11**
	256	93.94±0.13	98.51±0.04	98.74±0.14	90.03±0.54	97.25±1.42	**99.00±0.08**
SVHN	32	26.21±0.42	25.15±0.78	29.58±3.22	23.43±0.79	24.53±1.33	**34.47±1.14**
	256	22.74±0.05	77.89±0.35	66.30±1.06	22.81±0.24	44.94±20.42	**85.14±0.20**
CIFAR-10	256	47.92±0.20	**53.78±0.36**	47.49±0.22	40.65±1.45	42.80±0.44	52.77±0.45
	1024	51.62±0.25	60.65±0.14	51.39±0.46	42.86±0.88	16.22±12.44	**63.99±0.47**

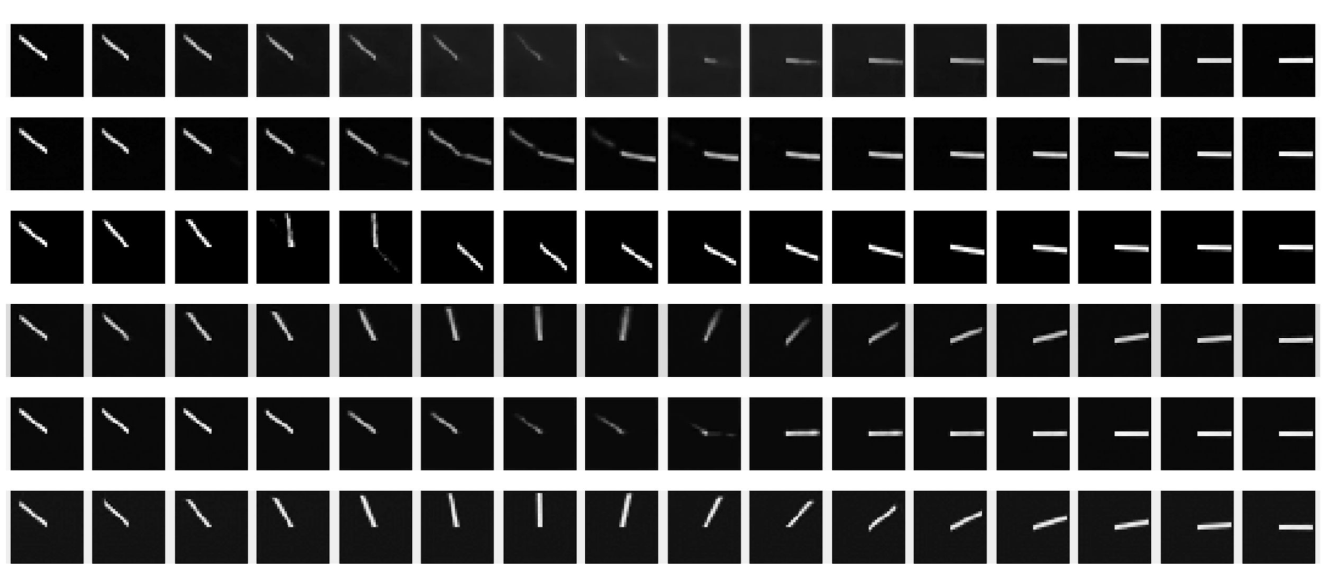
Single-layer classifier accuracy

Dataset	d_z	Baseline	Denoising	VAE	AAE	VQ-VAE	ACAI
MNIST	32	77.56	82.59	75.74	79.19	82.39	**94.38**
	256	53.70	70.89	83.44	81.00	**96.80**	96.17
SVHN	32	19.38	17.91	16.83	17.35	15.19	**20.86**
	256	15.62	**31.49**	11.36	13.59	18.84	24.98

Clustering accuracy

Synthetic line task

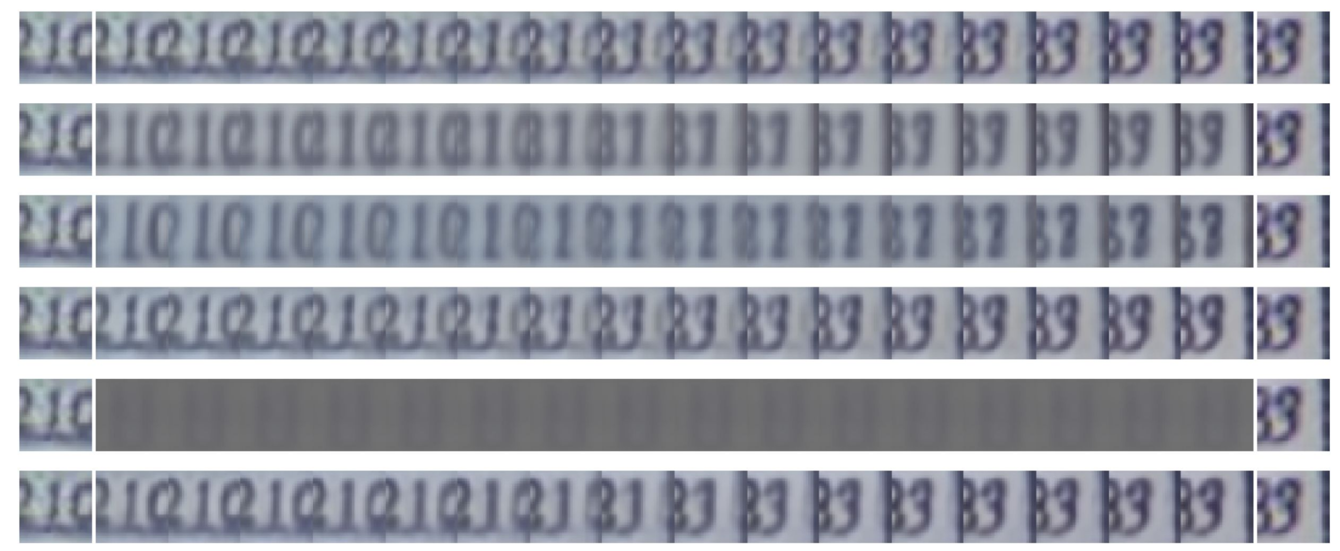
Metric	Baseline	Denoising	VAE	AAE	VQ-VAE	ACAI
Mean Distance $(\times 10^{-3})$	6.88±0.21	4.21±0.32	1.21±0.17	3.26±0.19	5.41±0.49	**0.24±0.01**
Smoothness	0.44±0.04	0.66±0.02	0.49±0.13	0.14±0.02	0.77±0.02	**0.10±0.01**



Interpolation examples

MNIST

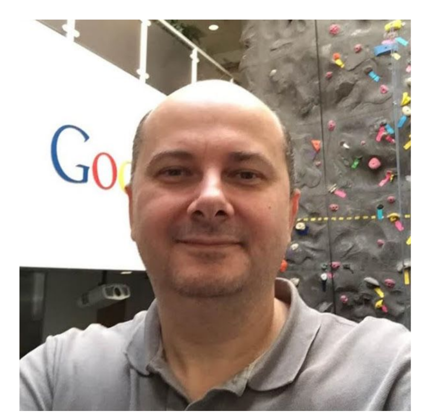
SVHN



CelebA

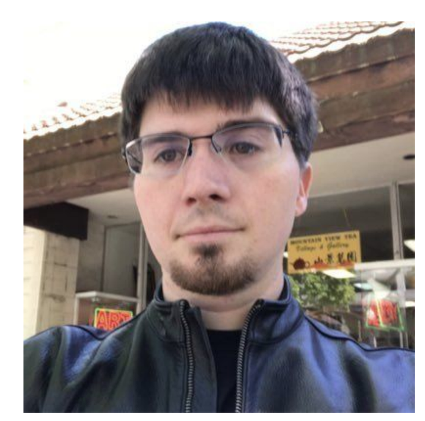
Pointers

Paper: `https://arxiv.org/abs/1807.07543`
Code: `http://github.com/brain-research/acai`
This poster: `http://bit.ly/acai-poster`
E-mails: `dberth@google.com, craffel@gmail.com`



David

Aurko



Ian