cDVGAN: One Flexible Model for Multi-class Gravitational Wave Transient Generation

Tom Dooney, Open Universiteit

Email: td1@ou.nl

23/01/2024

Contents

Tom Dooney, td1@ou.nl

Introduction

- Simulating realistic GWs and glitch events offers many benefits
  - Learn underlying distributions
  - Bypass expensive simulations
  - Data augmentation – class imbalance
  - Validate detection schemes
  - Mock data challenges

- We focus on the Generative Adversarial Network (GAN) framework for transient time-domain observations (0.25s)

- cDVGAN – Conditional Derivative GAN
  - Conditions on multiple signal classes –glitches, bursts, CBC
  - Uses derivative signals as a form of regularization
  - Extension: cDVGAN2 (second-order derivatives)
  - One flexible model - More user control
  - We can leverage class mixing
    - Generate data outside training distribution

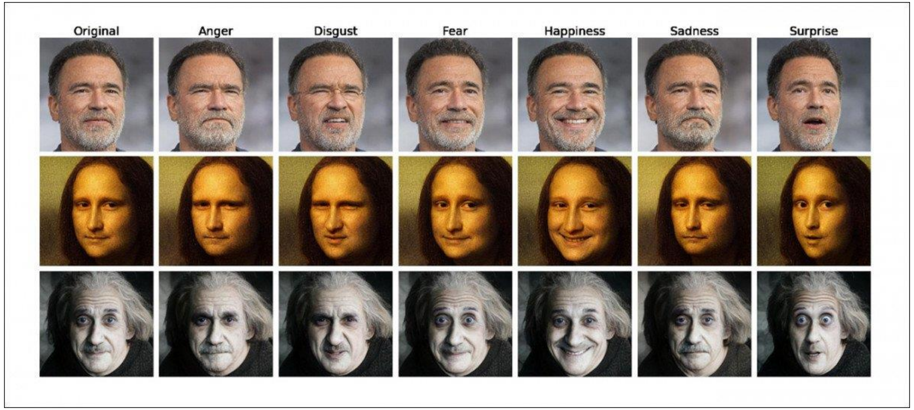


Fig: Output of 2D GAN conditioned on emotions

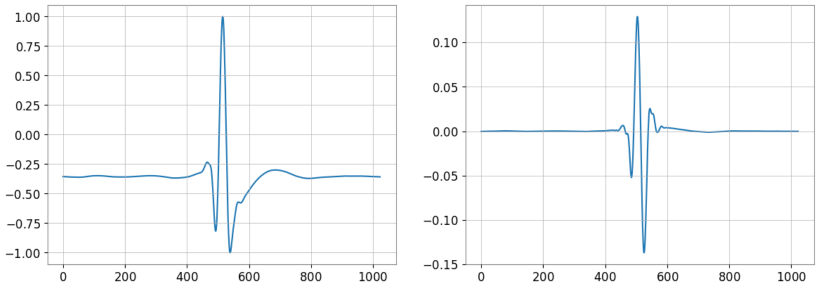
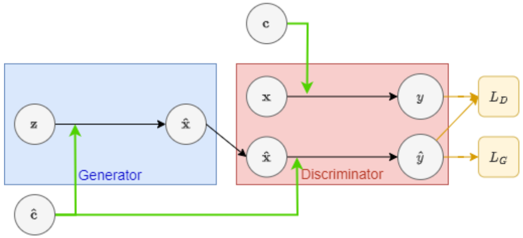


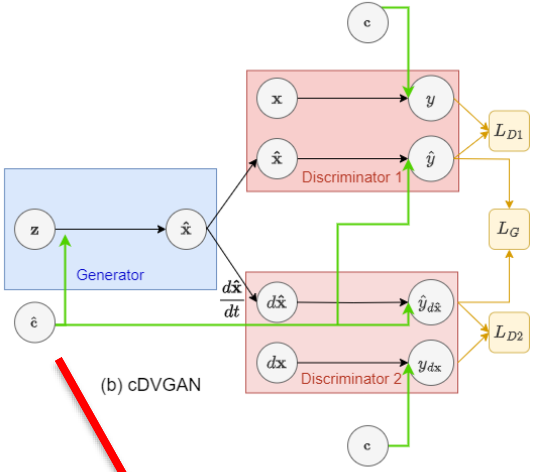
Fig: A blip glitch and its derivative signal

cGAN and cDVGAN

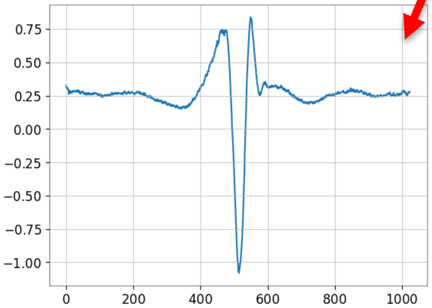
- cDVGAN(2) uses extra discriminator(s)
  - First (and second) order derivatives
  - Wasserstein GAN

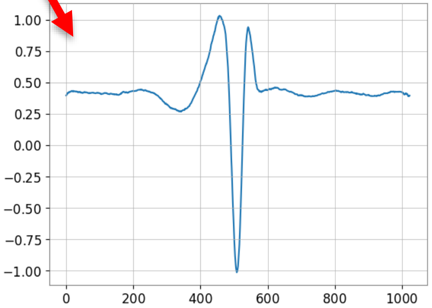


(a) Typical cGAN



(b) cDVGAN





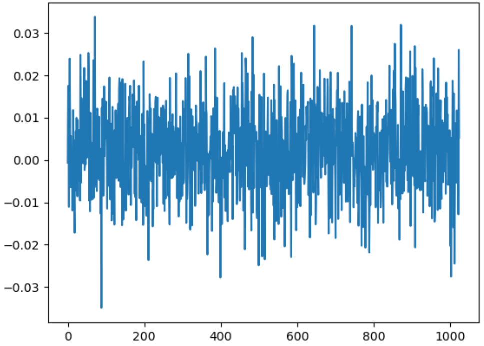
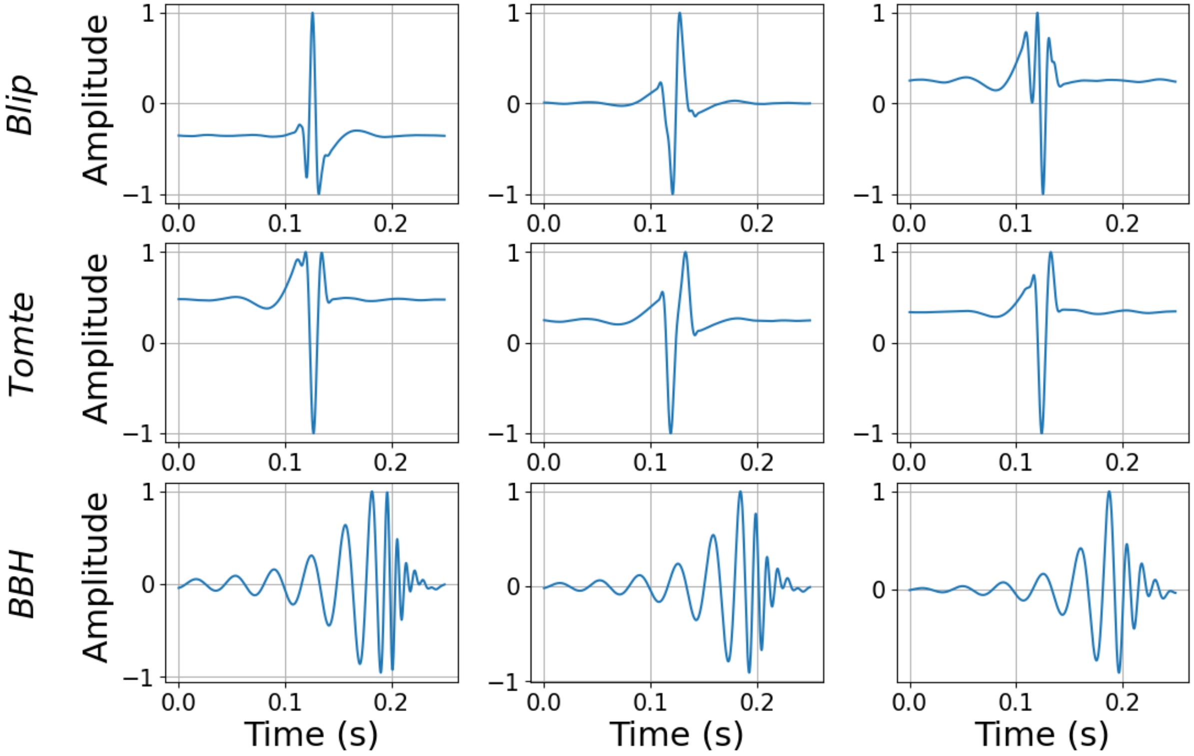


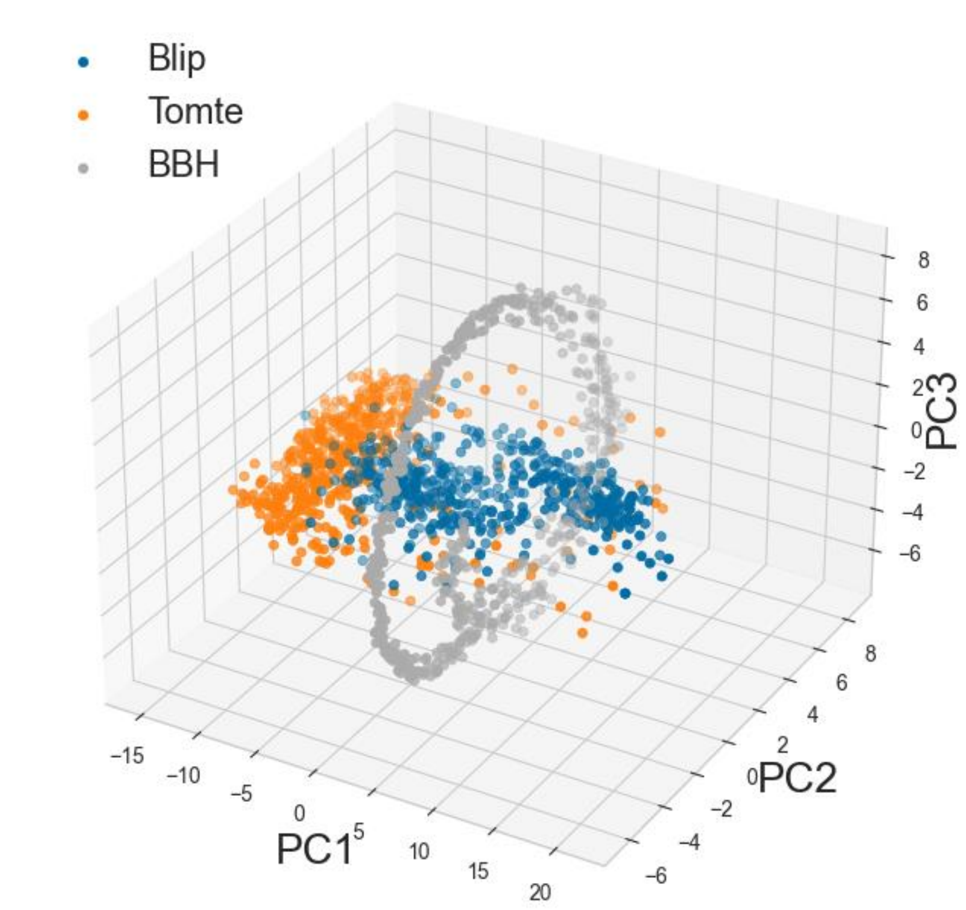
Fig: GAN output during training

Training Data

- Gravity Spy Glitches denoised
  - Blip and Tomte (O3)
  - Whiten and bandpass (20, 350) Hz
  - Savitzky-Golay filters
- BBHs from IMRPhenomD
  - 30-160 solar masses (m1>m2)
  - Plus polarization only
- 7,500 samples
  - 2,500 each class
  - 4,096 Hz
  - 1,024 data points (0.25s)
- 500 epochs

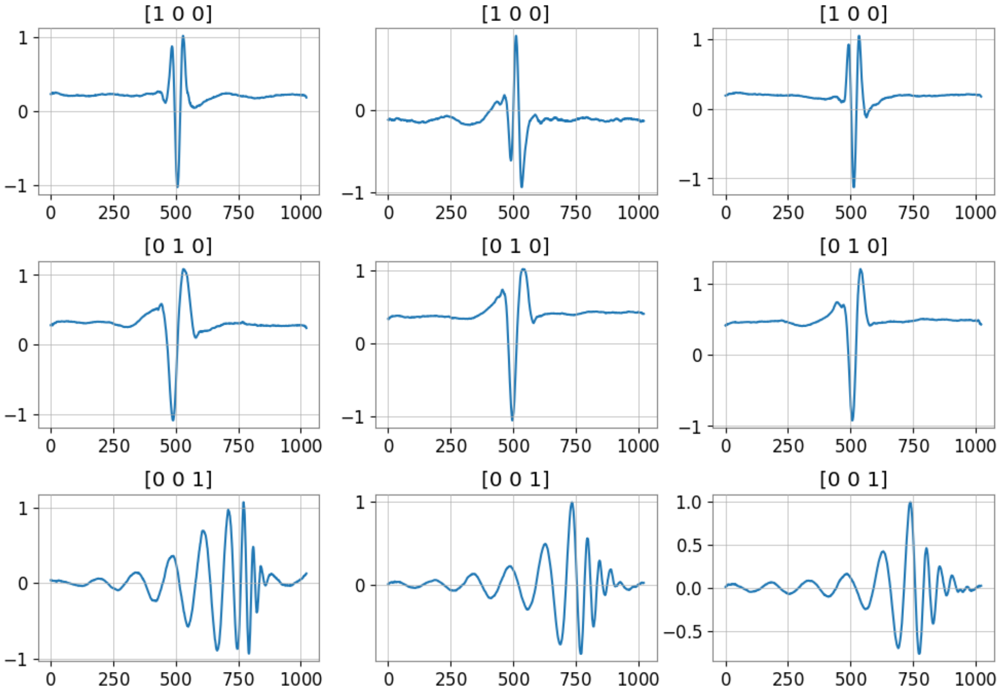


PCA on training data



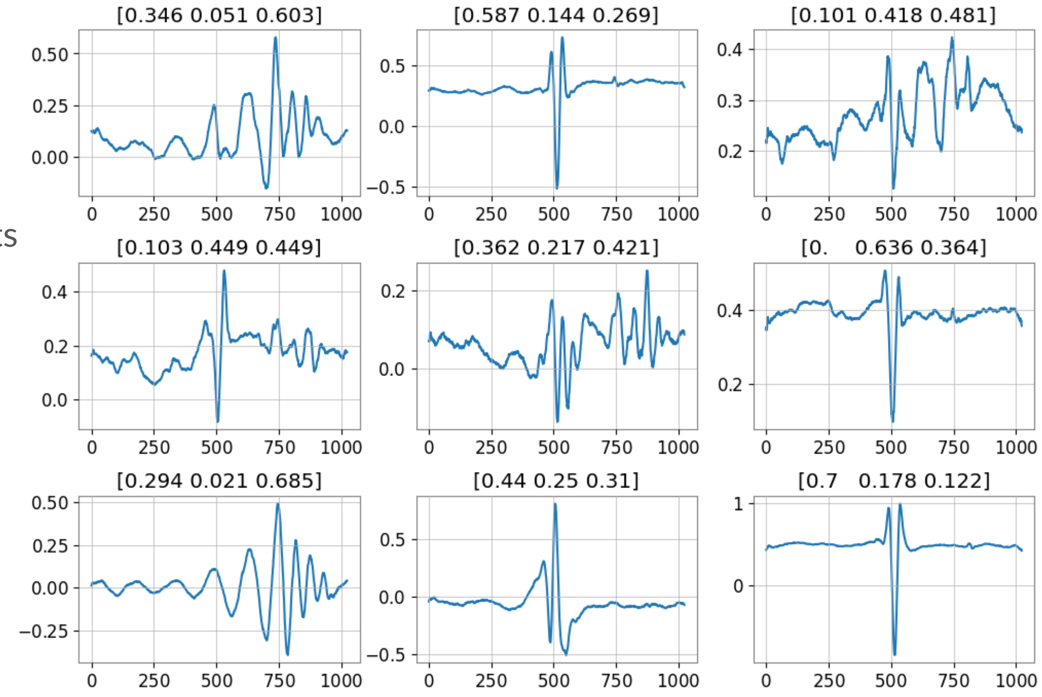
cDVGAN– Vertex Dataset

- Standard signal classes



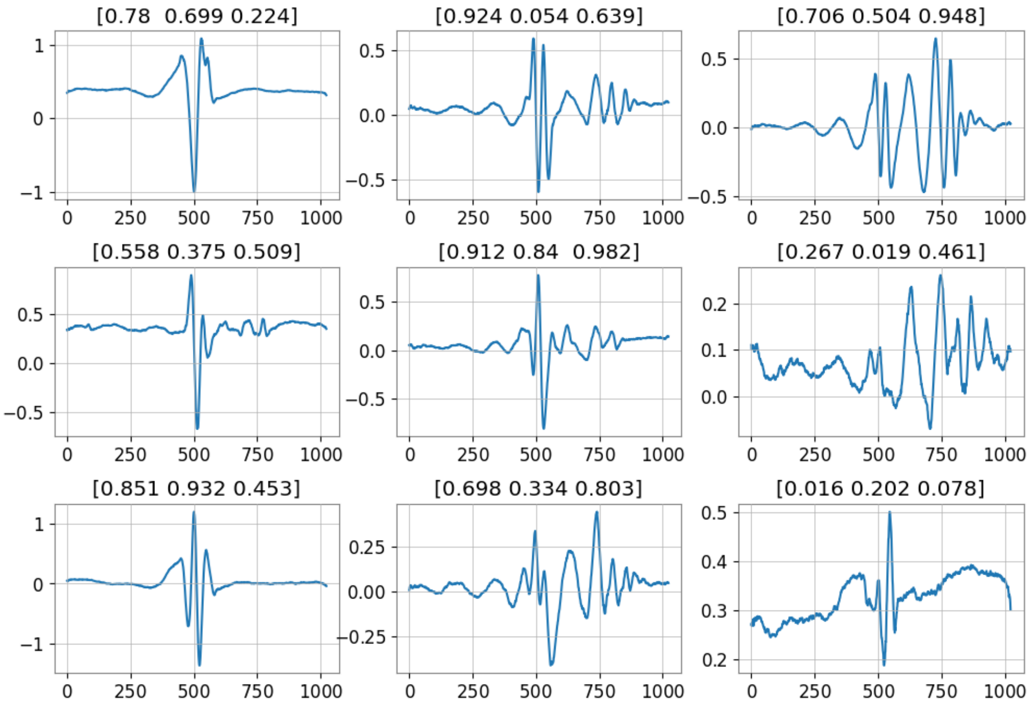
cDVGAN – Simplex Dataset

- Points on a k=2 simplex
  - Triangle
- Simplest surface that intersects all training classes
- Class vector sums to 1

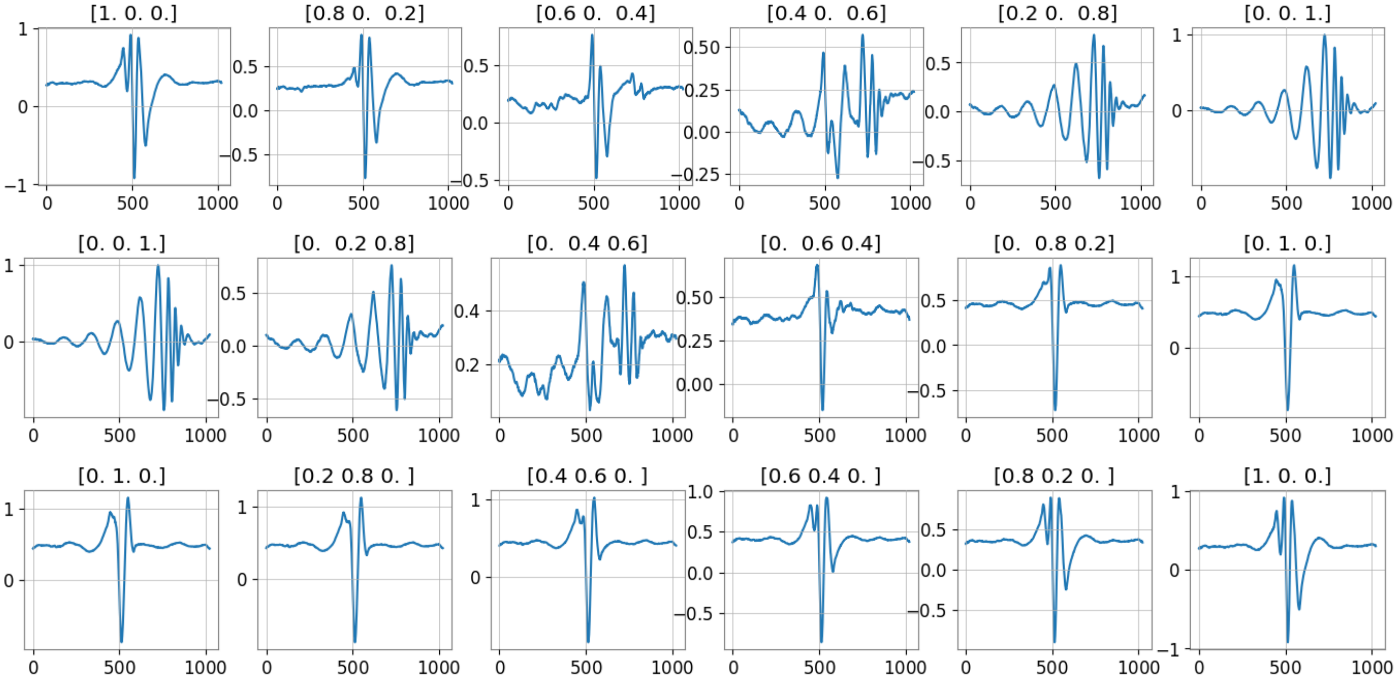


cDVGAN – Uniform Dataset

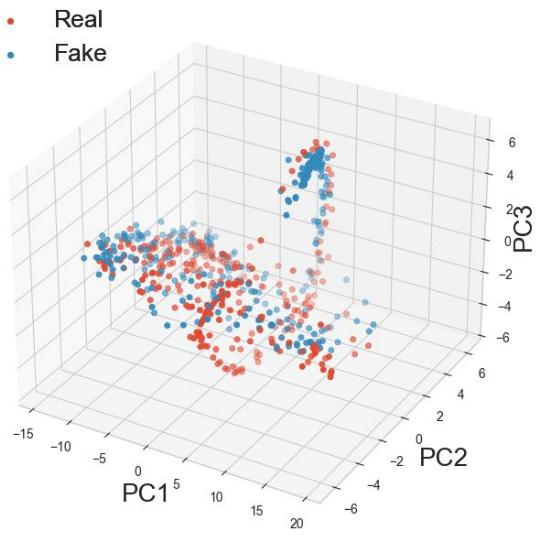
- Sample each point on U[0,1]

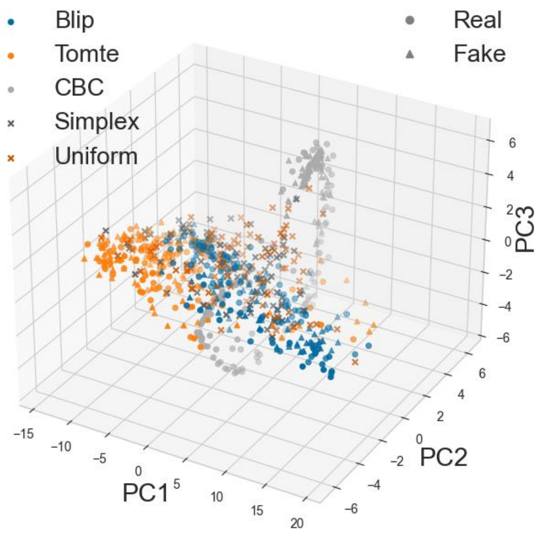


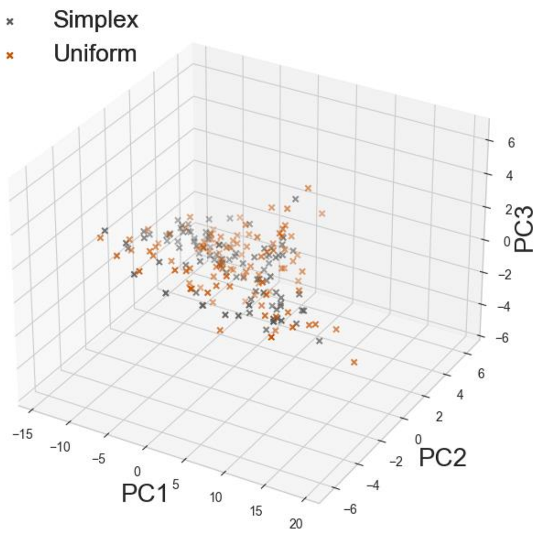
Class interpolation



PCA – Real vs Generated Data







Evaluation of GAN data

- Experiment – search for transients in background noise using convolutional neural networks (CNNs)
    - Binary classification: Noise only / Noise + sample
    - Inject training and testing samples into LIGO detector background
    - Injected at SNR in [1,16]
    - Train CNN on fake, test on real holdout set

- Test set: 15,000 samples
    - 7,500 Noise only, 7,500 samples + Noise
    - Samples from GAN training data distribution (2,500 each class)

- We train 3 CNNs on 3 datasets from each GAN
    - Vertex
    - Simplex
    - Uniform
    - 100,000 samples in each training set, 20 epochs

- We want to investigate 2 things
    1. Effectiveness of derivative discriminators
    2. Effectiveness of hybrid datasets for training CNNs

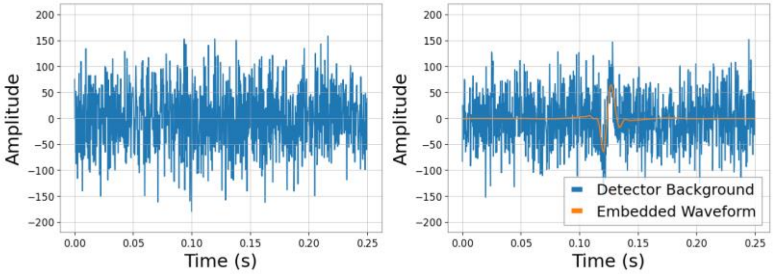


Fig: Two classes to predict
Noise only (0) and Signal+Noise (1)

Evaluation of GAN data

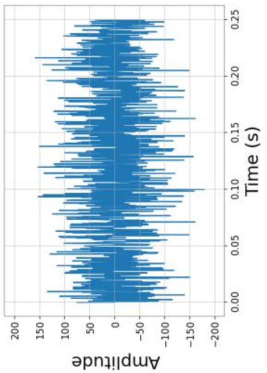
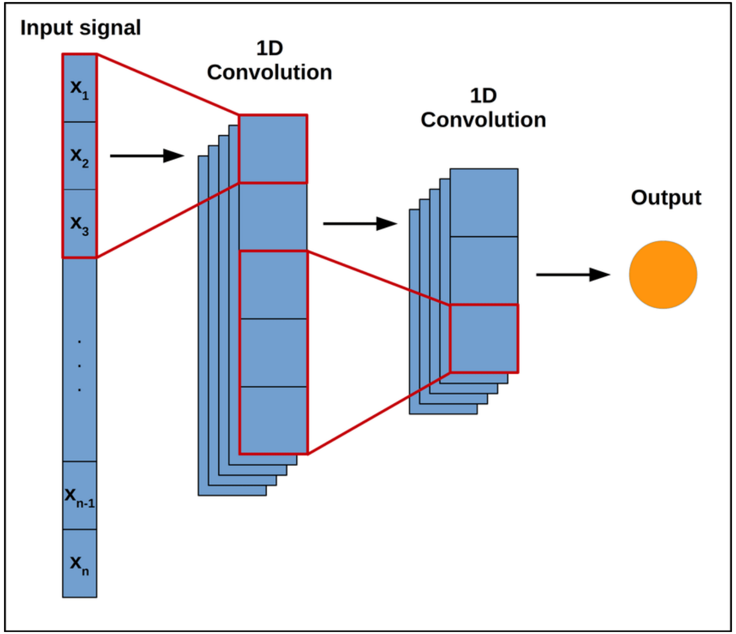


Fig: Noise only input



= 0

Fig: CNN detector

Evaluation of GAN data

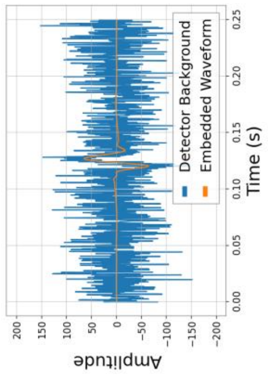
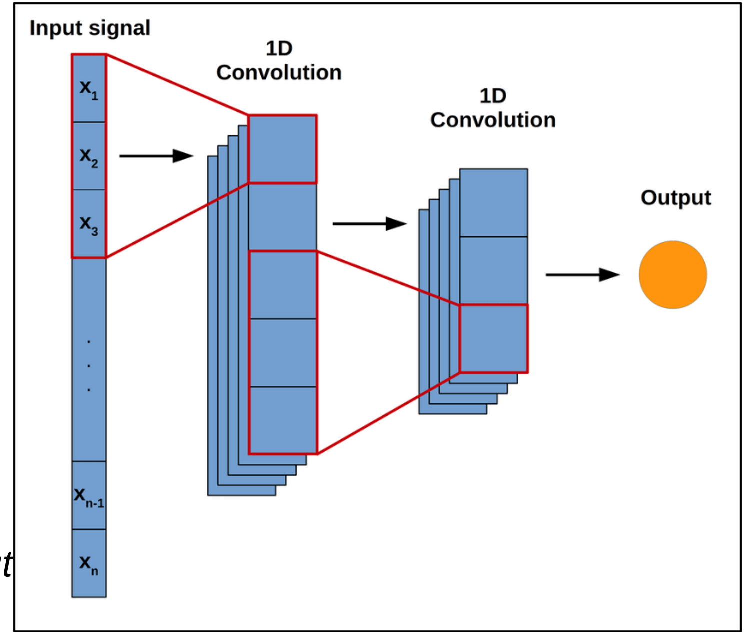


Fig: Sample + Noise input



= 1

Fig: CNN detector

Evaluation of GAN data

- We present the effect of derivative discriminators through ablation studies

- Ablation:
    - Selectively turning on/off a model component to understand its effect
    - Disable auxiliary discriminators

- Five GAN models
    1. cDVGAN
    2. cDVGAN2
    3. cWGAN
    4. McGANn
    5. McDVGANn

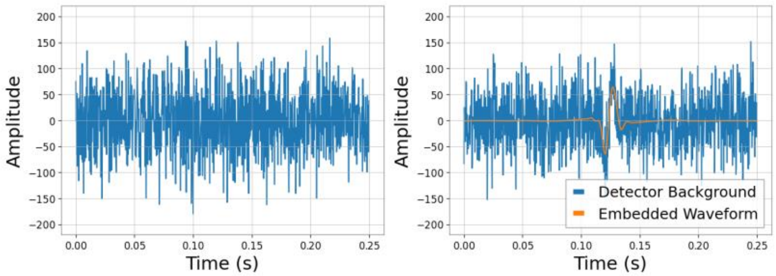
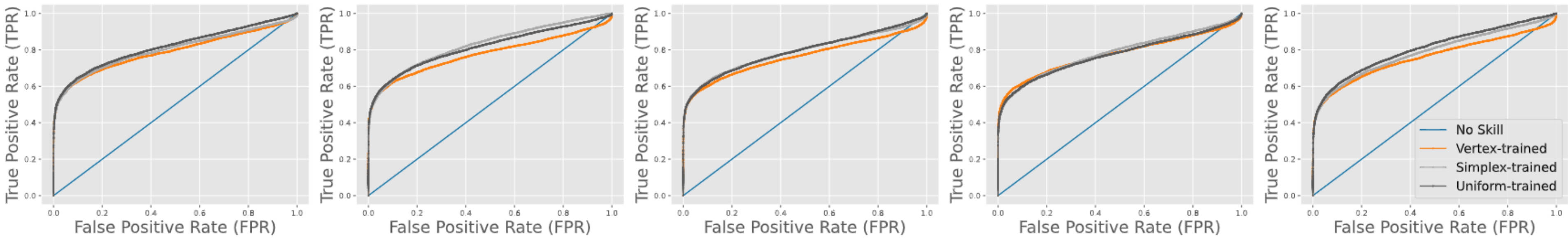


Fig: Two classes to predict
Noise only (0) and Sample+Noise (1)

Tom Dooney, td1@ou.nl

ROC Curves



(a) cDVGAN (b) cDVGAN2 (c) cWGAN (d) McGANn (e) McDVGANn

Fig: ROC curves for three datasets from each GAN

Area-Under-Curve (AUC) results

Dataset	cDVGAN (ours)	cDVGAN2 (ours)	cWGAN	McGANn	McDVGANn (ours)
Vertex-Trained	0.771 ± 0.012	0.758 ± 0.008	0.762 ± 0.018	0.768 ± 0.016	0.768 ± 0.022
Simplex-Trained	***0.802*** ± 0.019	0.789 ± 0.010	*0.788* ± 0.009	0.759 ± 0.010	*0.786* ± 0.012
Uniform-Trained	0.797 ± 0.022	*0.791* ± 0.009	0.777 ± 0.014	*0.770* ± 0.014	0.778 ± 0.012

Tab: μ±σ AUC values on a real holdout test set (5 iterations).

Area-Under-Curve (AUC) results

Dataset	cDVGAN (ours)	cDVGAN2 (ours)	cWGAN	McGANn	McDVGANn (ours)
Vertex-Trained	0.771 ± 0.012	0.758 ± 0.008	0.762 ± 0.018	0.768 ± 0.016	0.768 ± 0.022
Simplex-Trained	***0.802*** ± 0.019	0.789 ± 0.010	*0.788* ± 0.009	0.759 ± 0.010	*0.786* ± 0.012
Uniform-Trained	0.797 ± 0.022	*0.791* ± 0.009	0.777 ± 0.014	*0.770* ± 0.014	0.778 ± 0.012

Tab: μ±σ AUC values on a real holdout test set (5 iterations).

Conclusions

- GANs can be useful for learning underlying distributions of transient signals
    - Data augmentation
    - Validate detection schemes
    - Mock Data Challenges
    - Next generation glitch generator
- Conditional GANs allow us to explore the variation between sample classes
    - Inter-class and intra-class variation
    - Generate data outside the training data distribution
- Derivative discriminators can be effective at improving the features across multiple GAN architectures
    - cWGAN and traditional cGAN
    - Improves SNR in context of signal searches
- Hybrid samples useful for searching for multi-model distributions in background

Future Work

- Combine synthetic datasets for searches
- Expand cDVGAN to more signal (glitch) classes
  - Improved data augmentation
- Use BayesWave for more faithful training data
- Optimize architecture
  - Discriminator contributions to generator loss
  - Consitency term

Train your own cDVGAN!

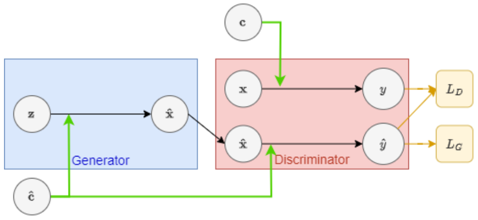
- https://git.ligo.org/tom.dooney/cdvgan_paper.git

Tom Dooney, td1@ou.nl

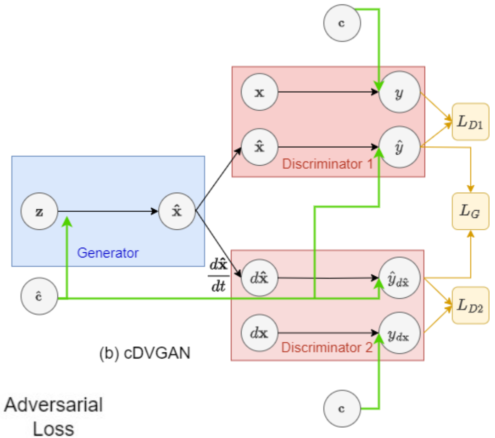
cDVGAN architecture

		Discriminator			(3.5M param.)
Operation	Output shape	Kernel size	Stride	Dropout	Activation
Signal input	(1024)	-	-	0	-
Reshape	(64,16)	-	-	0	-
Convolutional	(64,128)	14	2	0.5	Leaky ReLU
Convolutional	(32,128)	14	2	0.5	Leaky ReLU
Convolutional	(16, 256)	14	2	0.5	Leaky ReLU
Convolutional	(8, 256)	14	2	0.5	Leaky ReLU
Convolutional	(4, 512)	14	2	0.5	Leaky ReLU
Global Avg. Pooling	(512)	-	-	0.5	-
Avg. Pooling Dense	(128)	-	-	0.2	Leaky ReLU
Dense	(1)	-	-	0	Linear
Class Input	(3)	-	-	-	-
Class Dense	(128)	-	-	0	Linear
Scalar Product	(1)	-	-	-	-
Dense + Scalar Product	(1)	-	-	-	-
		DV Discriminator			(1.1M param.)
Operation	Output shape	Kernel size	Stride	Dropout	Activation
Signal input	(1023)	-	-	0	-
Dense	(512)	-	-	0	Leaky ReLU
Reshape	(32,16)	-	-	0	-
Convolutional	(32, 64)	5	2	0.5	Leaky ReLU
Convolutional	(16,128)	5	2	0.5	Leaky ReLU
Convolutional	(8,256)	5	2	0.5	Leaky ReLU
Convolutional	(4,256)	5	2	0.5	Leaky ReLU
Global Avg. Pooling	(256)	-	-	0.5	-
Avg. Pooling Dense	(128)	-	-	0.2	Leaky ReLU
Dense	(1)	-	-	0	Linear
Class Input	(3)	-	-	-	-
Class Dense	(128)	-	-	0	Linear
Scalar Product	(1)	-	-	-	-
Dense + Scalar Product	(1)	-	-	-	-
		Generator			3.5M param.
Operation	Output shape	Kernel size	Stride	BN	Activation
Latent input	(100)	-	-	✗	-
Class Input	(3)	-	-	✗	-
Class Dense	(32)	-	-	✗	-
Concatenate	(132)	-	-	✗	-
Dense	(1024)	-	-	✗	ReLU
Reshape	(32,32)	-	-	✗	-
Transposed conv.	(64,512)	18	2	✓	ReLU
Transposed conv.	(128,256)	18	2	✓	ReLU
Transposed conv.	(256,128)	18	2	✓	ReLU
Transposed conv.	(512, 64)	18	2	✓	ReLU
Transposed conv.	(1024,1)	18	2	✗	Linear
Flatten	(1024)	-	-	✗	-
Optimizer	RMSprop($\alpha = 0.0001$)				
Batch size	512				
Epochs	500				
Loss	Wasserstein				

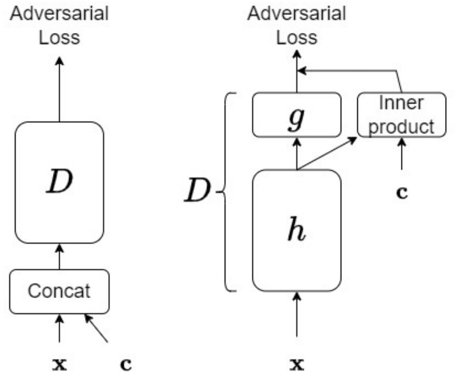
Conditioning Approaches



(a) Typical cGAN



(b) cDVGAN

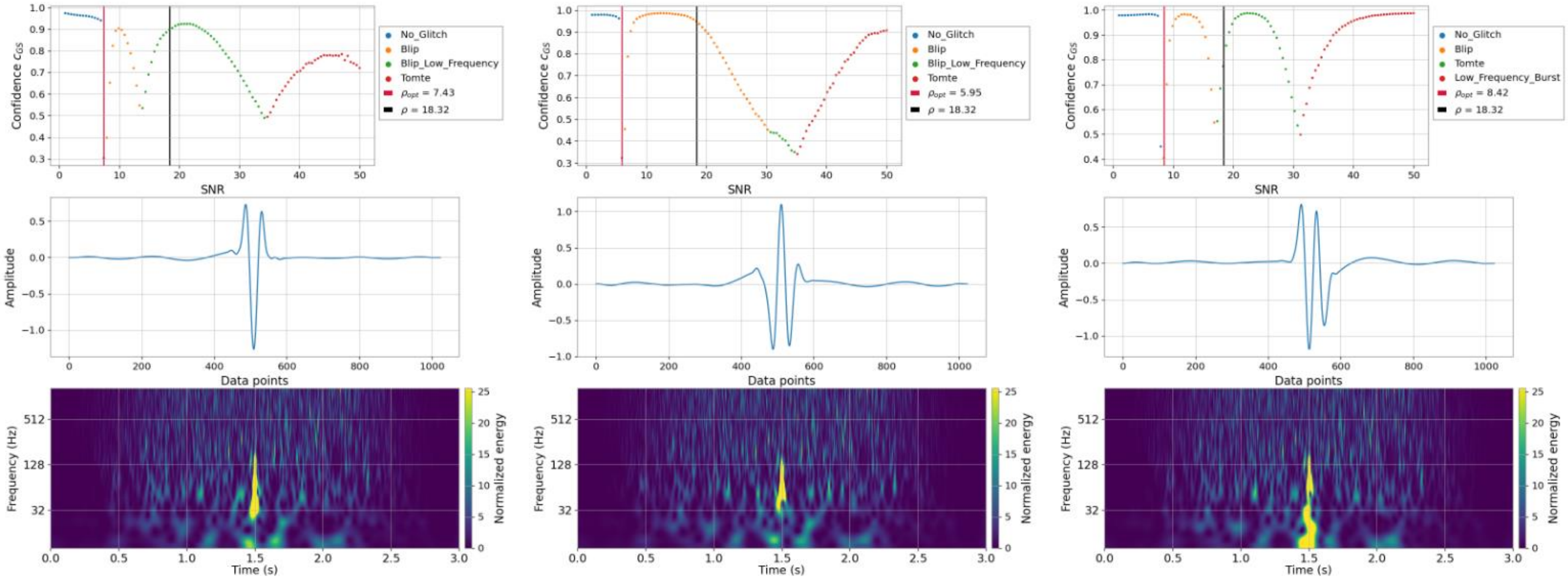


(a) Original CGAN (b) Projection Discriminator

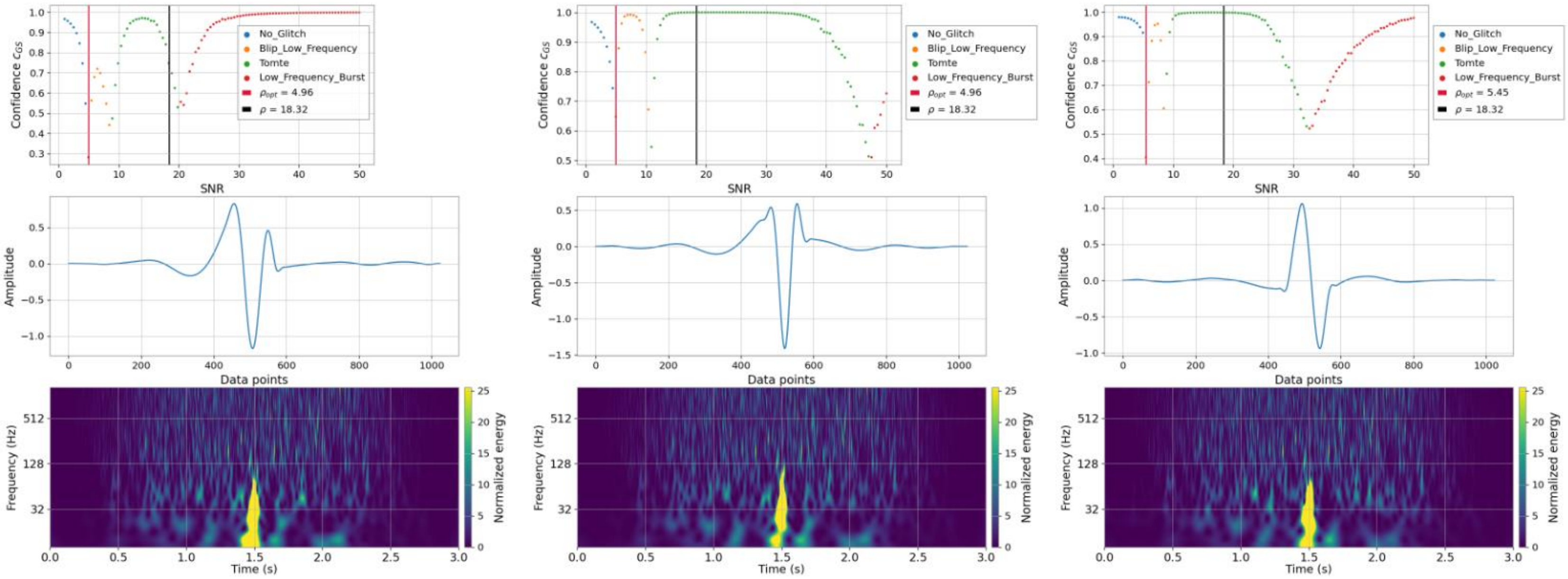
Questions?

Appendix

Training Data - Blips



Training Data - Tomte



AUC results

Dataset	cDVGAN (ours)	cDVGAN2 (ours)	cWGAN	McGANn	McDVGANn (ours)
Vertex-Trained	0.689 ± 0.009	0.680 ± 0.006	0.685 ± 0.017	*0.687* $\pm$ 0.010	0.668 ± 0.020
Simplex-Trained	0.698 ± 0.013	0.686 ± 0.007	*0.695* $\pm$ 0.010	0.673 ± 0.008	*0.676* $\pm$ 0.010
Uniform-Trained	***0.702*** $\pm$ 0.014	*0.693* $\pm$ 0.008	0.692 ± 0.009	0.680 ± 0.013	0.671 ± 0.006

Tab: AUC results for SNR < 8

Dataset	cDVGAN (ours)	cDVGAN2 (ours)	cWGAN	McGANn	McDVGANn (ours)
Vertex-Trained	0.844 ± 0.015	0.827 ± 0.010	0.830 ± 0.022	0.841 ± 0.021	0.856 ± 0.027
Simplex-Trained	***0.893*** $\pm$ 0.029	*0.881* $\pm$ 0.013	*0.867* $\pm$ 0.010	0.836 ± 0.012	*0.885* $\pm$ 0.013
Uniform-Trained	0.883 ± 0.031	0.878 ± 0.011	0.851 ± 0.020	*0.852* $\pm$ 0.020	0.873 ± 0.014

Tab: AUC results for SNR > 8

AUC results

SNR	Real:Fake 100:0	Real:Fake 75:25	Real:Fake 50:50	Real:Fake 25:75	Real:Fake 0:100
1-16	0.900 ± 0.001	0.898 ± 0.002	0.893 ± 0.002	0.887 ± 0.002	0.802 ± 0.019
1-8	0.799 ± 0.001	0.792 ± 0.005	0.787 ± 0.003	0.777 ± 0.003	0.698 ± 0.013
8-16	0.988 ± 0.001	0.987 ± 0.001	0.987 ± 0.001	0.985 ± 0.001	0.893 ± 0.029

Tab: AUC values over three SNR ranges for different proportions of real:synthetic data for a training set fixed at 100,000 samples.