Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
ICSC Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing

Generative AI for cosmic ray background data augmentation for LiteBIRD experiment

Giovanni Cavallotto (INFN MiB), Stefano Della Torre (INFN MiB)

Spoke 3 General Meeting, Elba 5-9 / 05, 2024

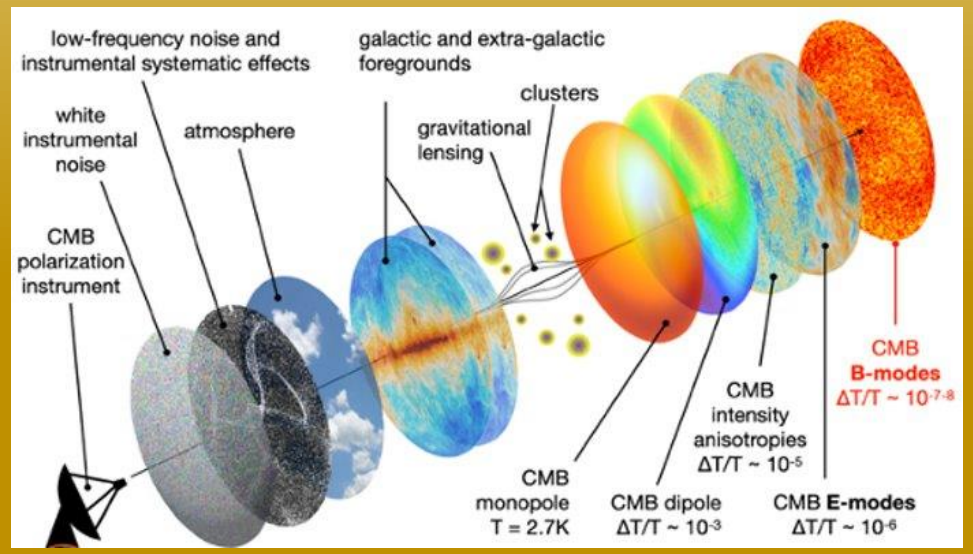
Scientific Rationale

LiteBIRD study *B*-mode polarization and Inflation from Cosmic Background Radiation:

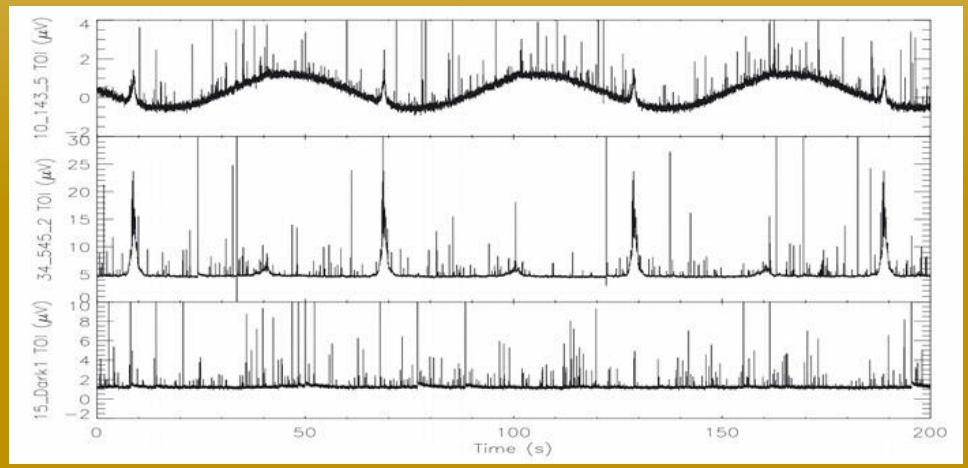
- Making a discovery or ruling out well-motivated inflationary models
- Insight into the quantum nature of gravity

- **B – modes ≈ 10^{-3} CMB signal & sensitivity ≈ 30x of Plank**
- **90% of Plank data affected by CR background**
- **Sensible to CR energy deposit & direct hits (Cryogenic bolometers + TES electronic)**
- **19Hz final sampling frequency + long exposion (no filtering)**

CMB noise superimposion

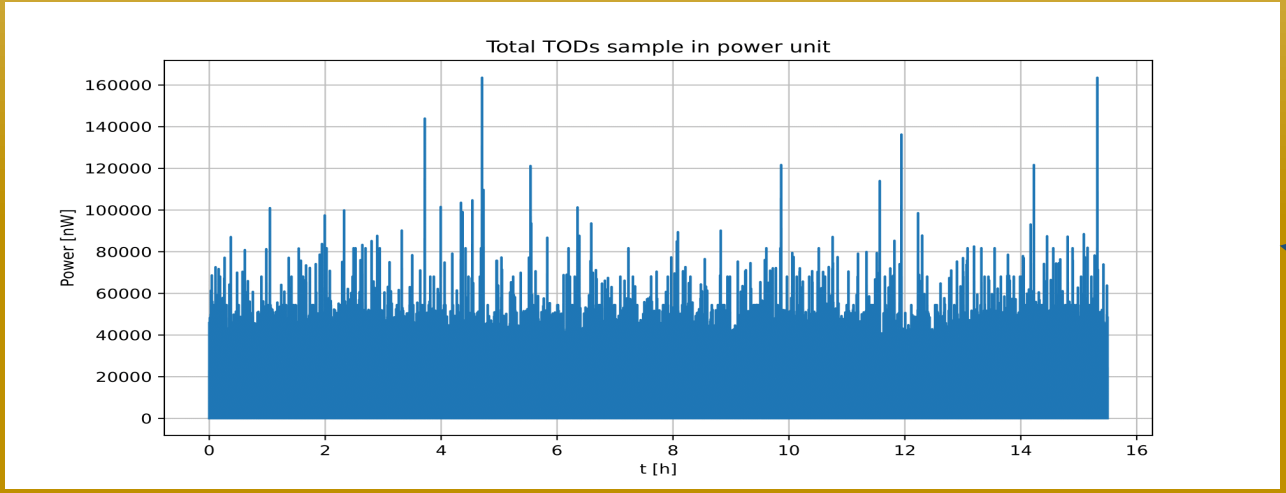


Plank CR glitches



Technical Objectives

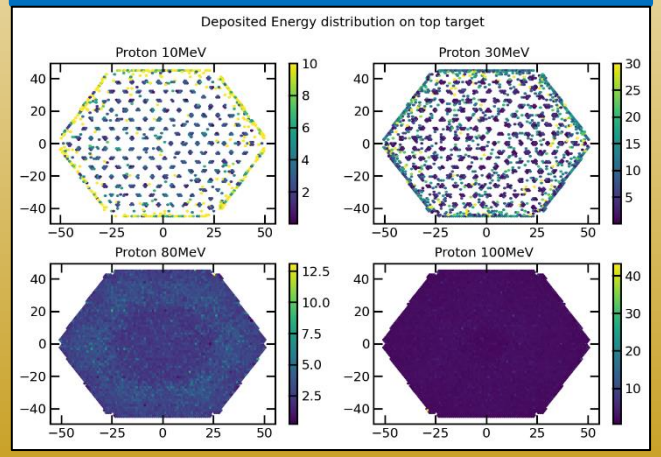
- **Synthetically** generate the time series covering the **whole mission**
- Achieve a reasonable generation **computational time (no ML ≈ 30x TOD length)**
- Genuine **statistically independent** AI generation
- Mimic MC data sample **peculiar features**
- Study different mission space environment & periods **(CR flux evolution)**



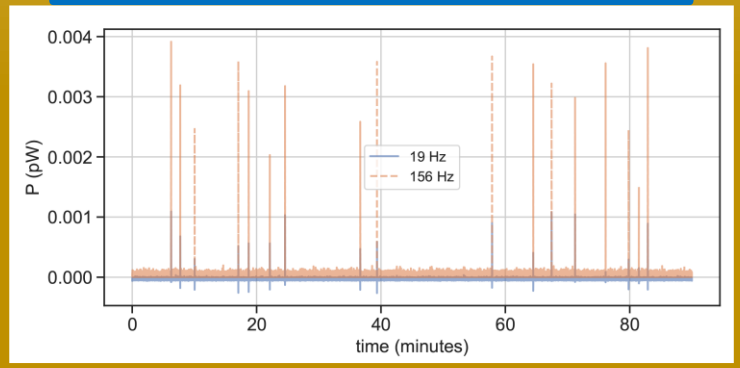
Iron

Methodologies

Synthetically generate CR energy deposit map



MC generate TOD output



Modulated CR at L2

Detector structure

Energy deposit in HFT

Semi analitical

- Bolometer (therma)
- TES (current) response

Analitical

CR TOD Background

Data Augmentation

Data post analysis

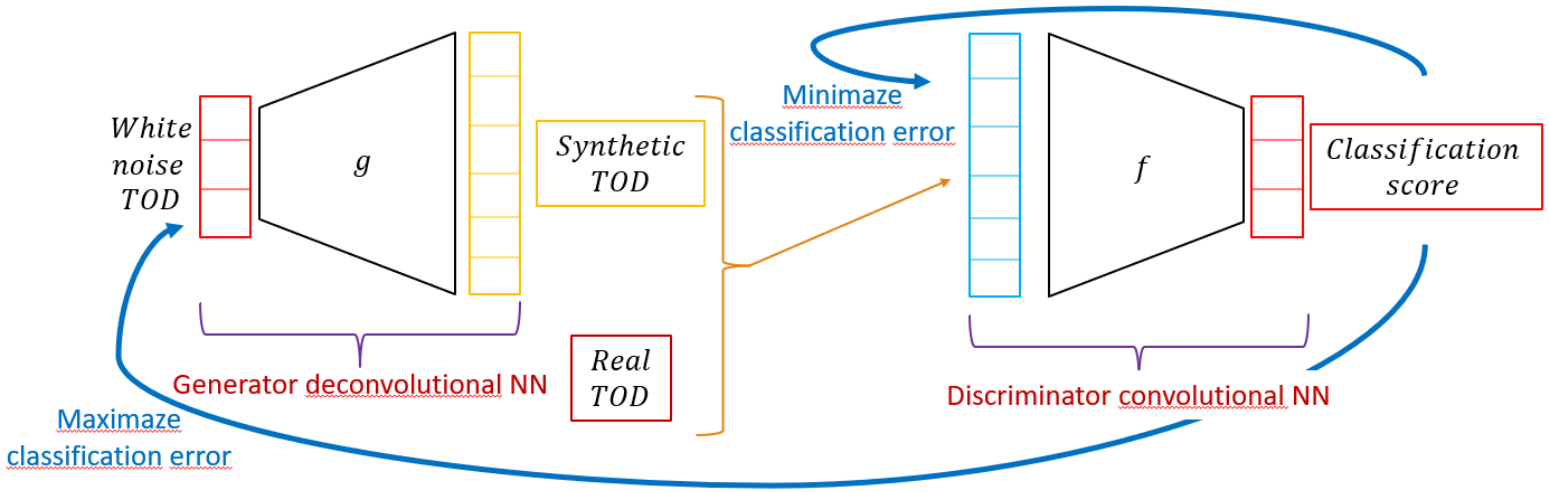
Our project: CRAB

Methodologies

Python & TensorFlow library

- Sequatial Convolutional & Deconvolutional NN
- Custom combined training (discriminator & generator)

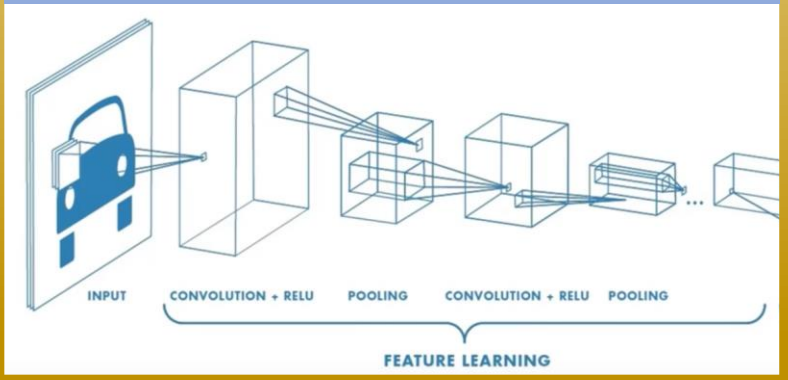
GAN algorithm (NN couple)



$Loss_G = Wess[f(g(x_i))]$

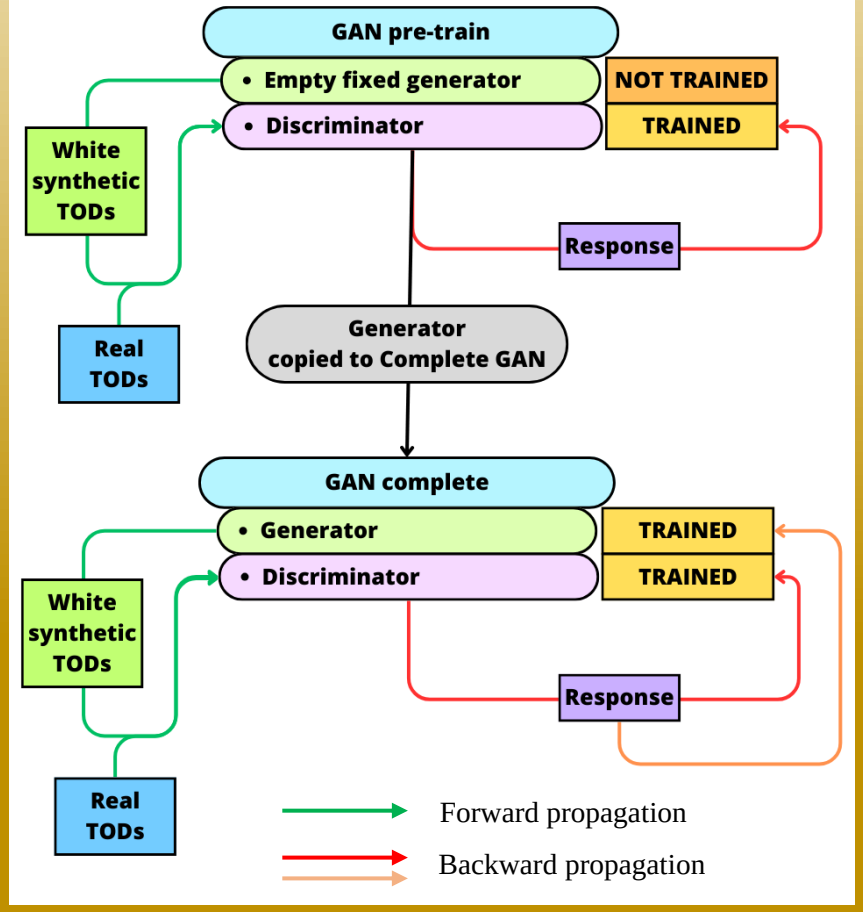
$Loss_D = Wess[f(x_i), f(g(x_i))]$

Convolutional Neural Networks



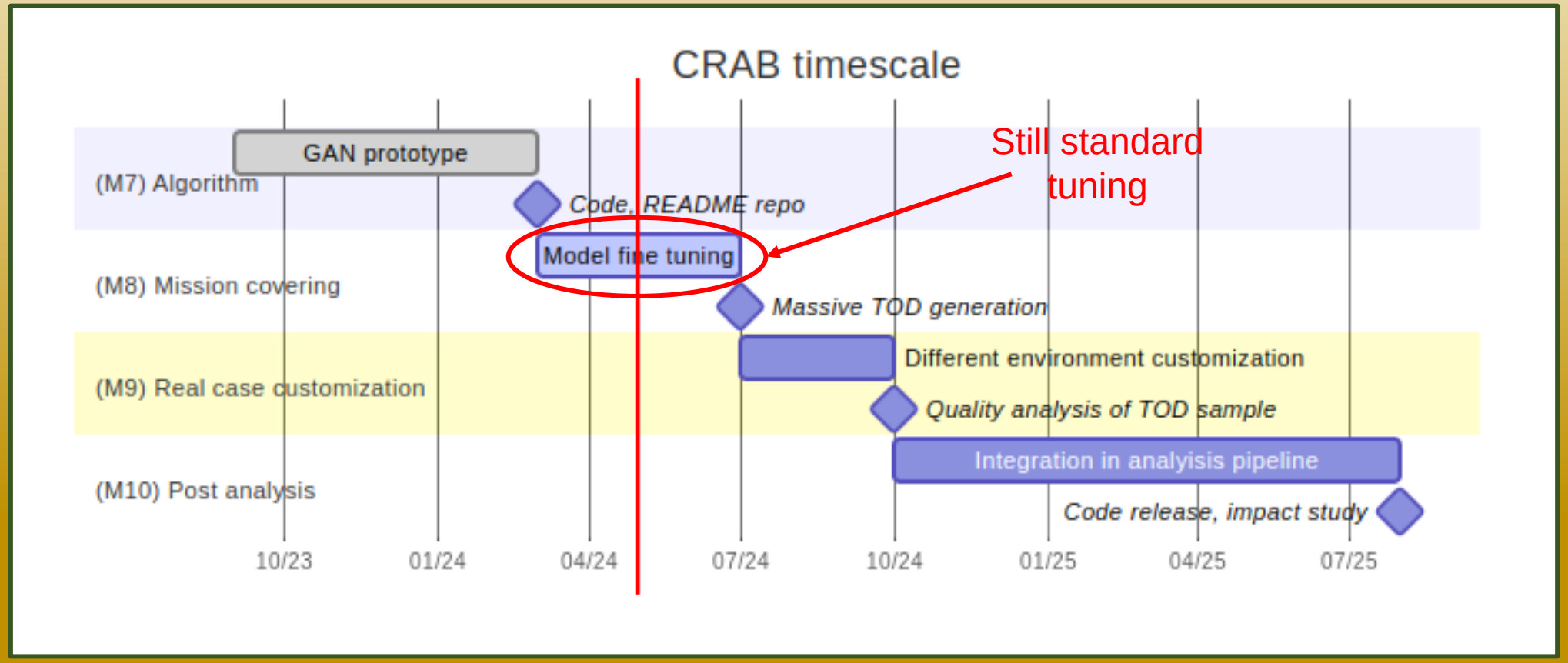
Solutions: Cosmic Ray Artifitial Background (CRAB)

Training scheme



- Pre-training of the discriminator only for limited epochs → Learn the real TODs classification (starting from a stable point)
- GAN building with pre-trained discriminator
- Complete GAN training
- (Eventially extra discriminator training steps) → Avoid discriminator domianted
- Synthetic generation (generator predict) → Final TOD outputs

Timescale, Milestones, KPIs



From Trieste meeting (October 2023): Expected Results

Identified optimization points:

- Exploit GPU tensorflow integrated API for deeper training
- Increase generated TODs complexity
- Overcome the discriminator dominated model and mode collapse issues
- Insert noise in the training input
- Test VAE algorithm

Expected results:

- Find the best (and fast) ML architecture, training strategy
- Produce first realistic TODs samples

Accomplished Work, Results

Different loss function and training algorithm

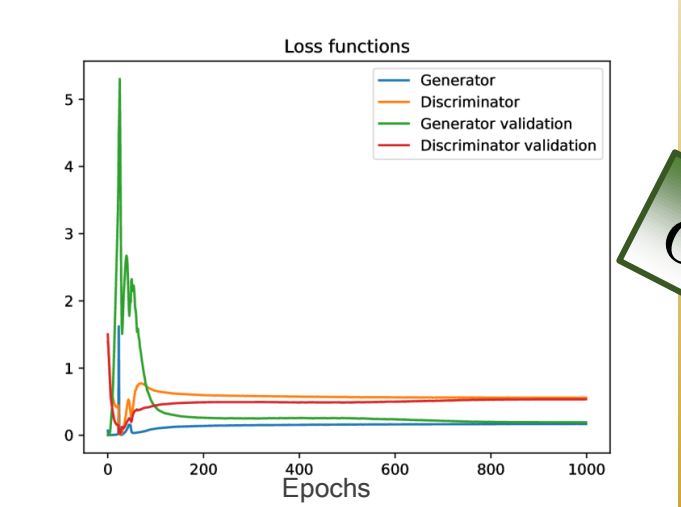
Wasserstein GAN

- Wasserstein loss metric
- Easily stucked in local minima
- Not trivial minimization of the loss (not bounded and negative for the generator)
- Accuracy not normalized
- General distribution separation

Cross entropy GAN

- Binary cross entropy loss metric
- Limited and always positive
- Included in the TensorFlow frame
- Trivial implementation of validation metrics
- Not completely stable

Accomplished Work, Results

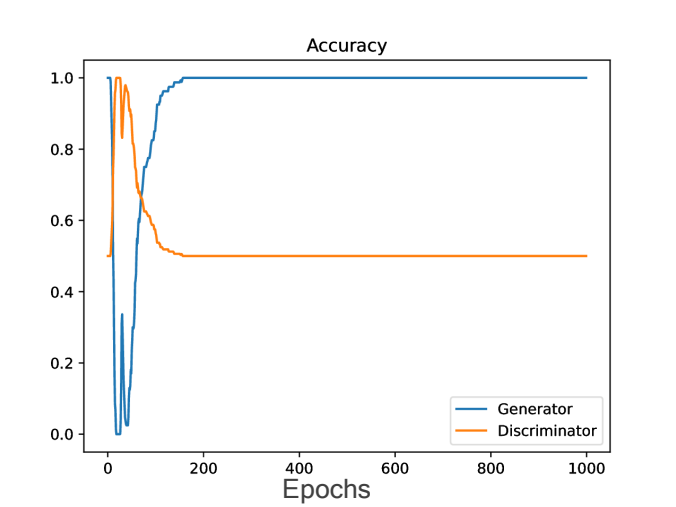


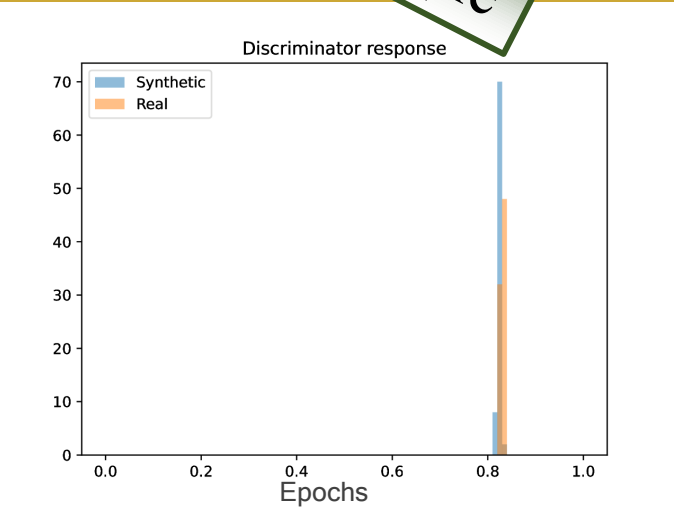
Different loss function and training algorithm

Output metrics example

Cross entropy GAN

- Binary cross entropy loss metric
- Limited and always positive
- Included in the TensorFlow frame
- Trivial implementation of validation metrics
- Not completely stable



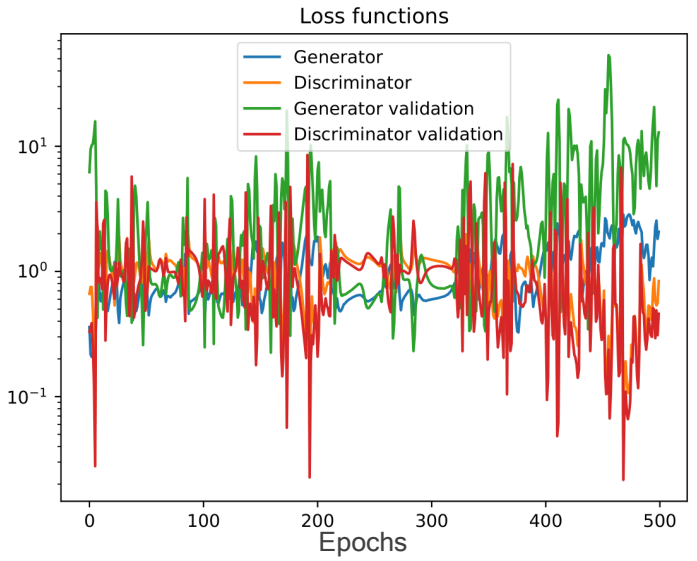


Accomplished Work, Results

Known issues

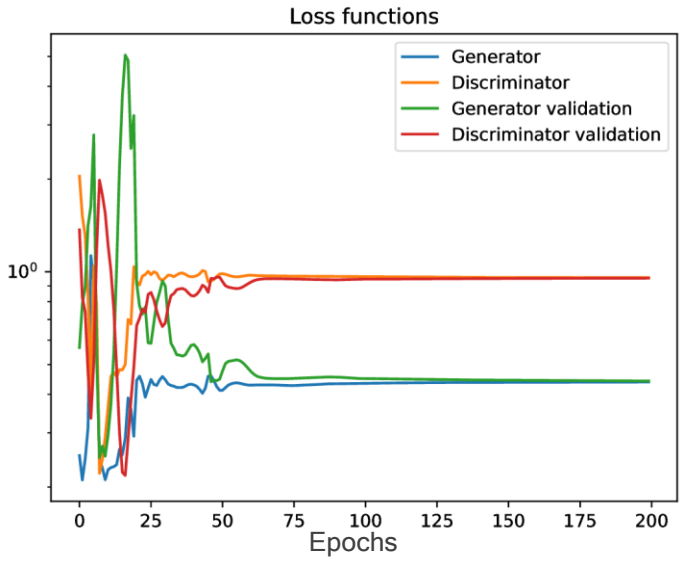
Mode collapse

Switching win-lose performances in unstable loop



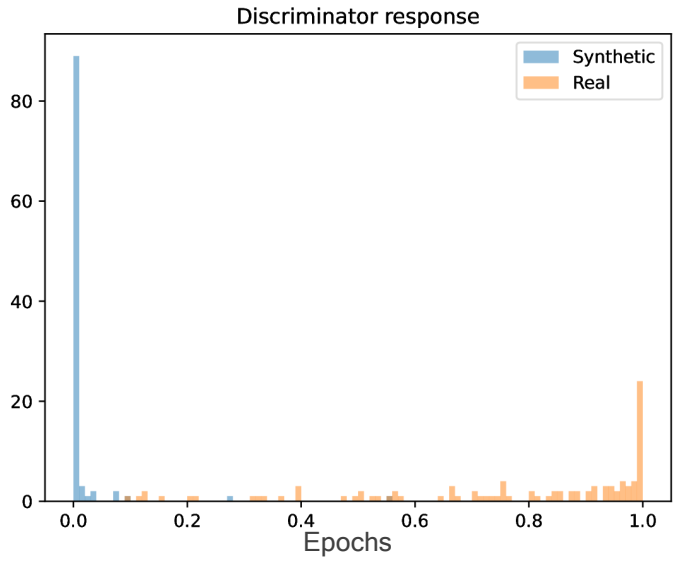
Generator dominated

The discriminator classify all the TODs as real



Real data miss classification

Loss of generator ability to classify real TODs



Next Steps and Expected Results

NEXT STEPS:

- Overcome the mode collapse issues
- Test the Variational Auto Encoders as alternative
- Synthetic TODs post analysis

Expected results:

- Stable synthetic TOD generator and fine tuning
- Benchmarks of training sample dimension and TOD production consumption

Thanks for your attention

Accomplished Work, Results

NN architecture

Model: "Generator"

Layer (type)	Output Shape	Param #
Dense (LeakyReLU)	(None, 265)	26765
Conv1D Transpose (LeakyReLU)	(None, 265, 8)	88
Batch Normalization	(None, 265, 8)	32
Conv1D Transpose (LeakyReLU)	(None, 530, 16)	1296
Batch Normalization	(None, 530, 16)	64
Conv1D Transpose (LeakyReLU)	(None, 1060, 32)	5152
Batch Normalization	(None, 1060, 32)	128
Separabl eConv1D (LeakyReLU)	(None, 1060, 1)	353
lambda_1 (Normalization)	(None, 1060, 1)	0

Trainable params: 33766 (131.90 KB)

Model: "Discriminator"

Layer (type)	Output Shape	Param #
Conv1D (LeakyReLU)	(None, 1060, 8)	168
LayerNormalization	(None, 1060, 8)	16
Conv1D (LeakyReLU)	(None, 1060, 8)	648
LayerNormalization	(None, 1060, 8)	16
Conv1D (LeakyReLU)	(None, 530, 16)	1296
LayerNormalization	(None, 530, 16)	32
Conv1D (LeakyReLU)	(None, 106, 32)	5152
LayerNormalization	(None, 106, 32)	64
Dense	(None, 1)	3393

Trainable params: 10785 (42.13 KB)

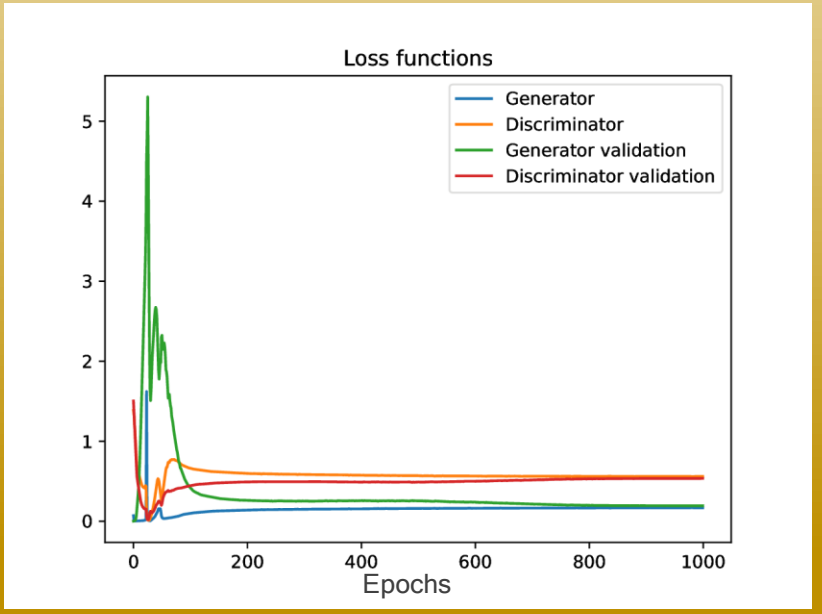
Solutions

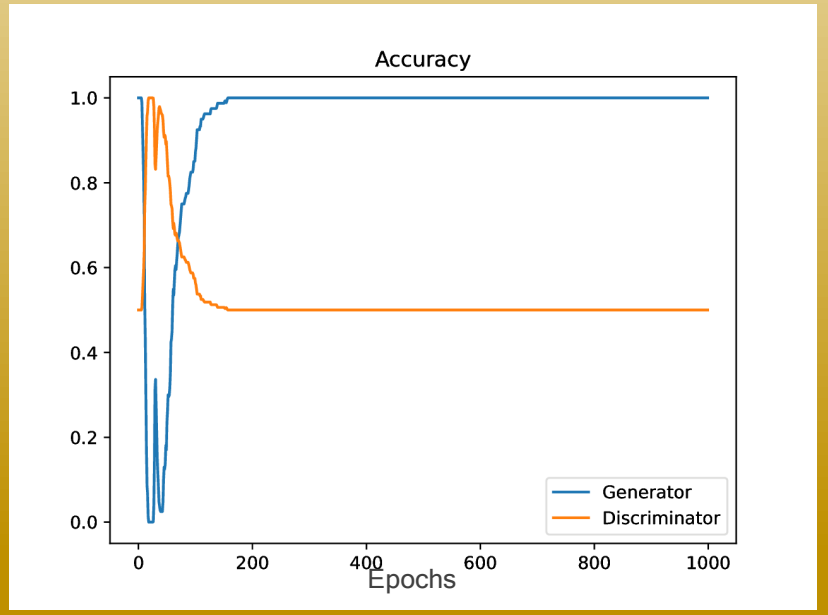
Parameter tuning

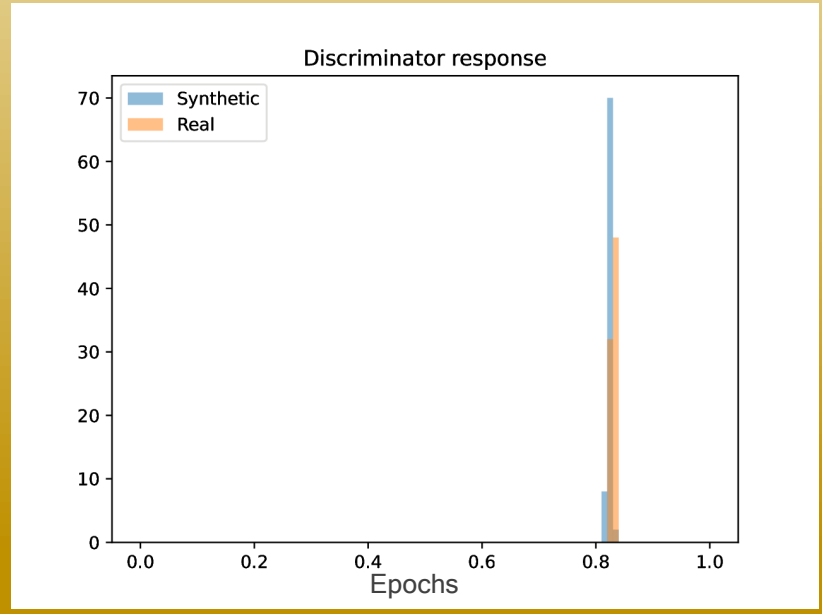
- Normalization (strongly influencing the classification accuracy)
- Layers (Convolution 1D, Number, Windows (Lenght of deposit energy relaxation */ n)
- Activation (LeakyReLU, Tanh)
- Optimizers (Adam, SGD and combinations)
- LR (larger for generator, history)
- Synthetic weight (to keep the discriminator focused on real TODs)
- Labem smoothing

Accomplished Work, Results

- **Best GAN version (model-18)**

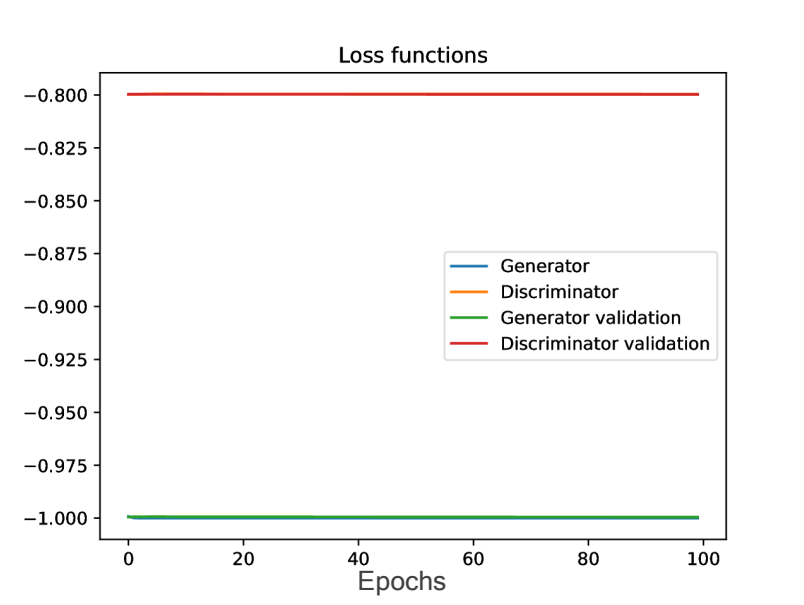


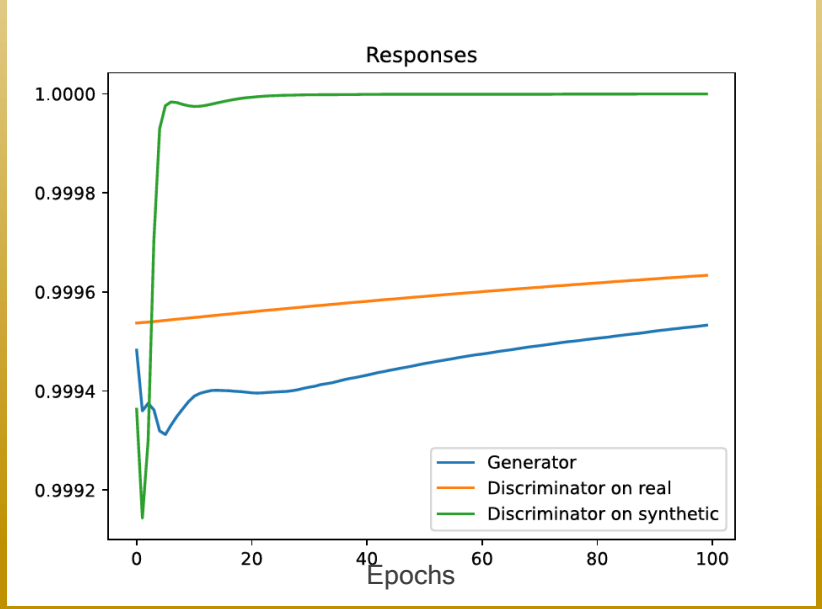


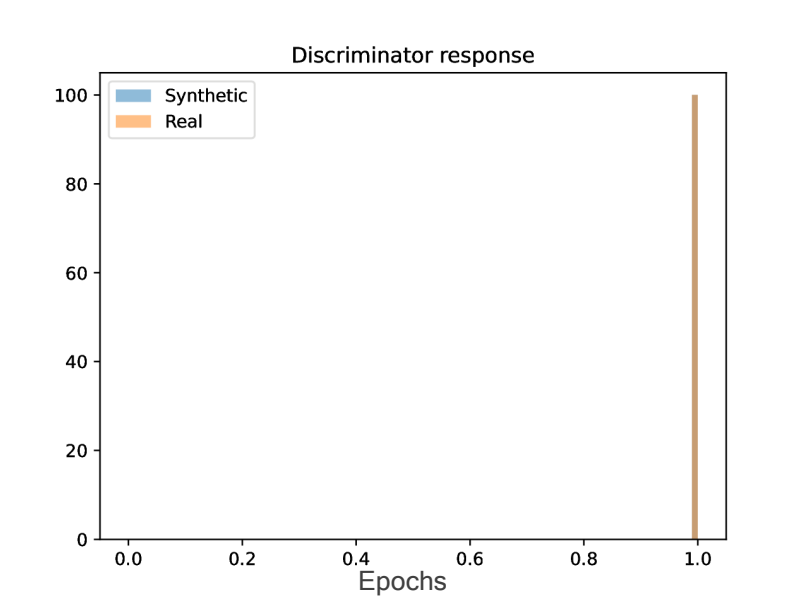


Accomplished Work, Results

- **WGAN modello 21**

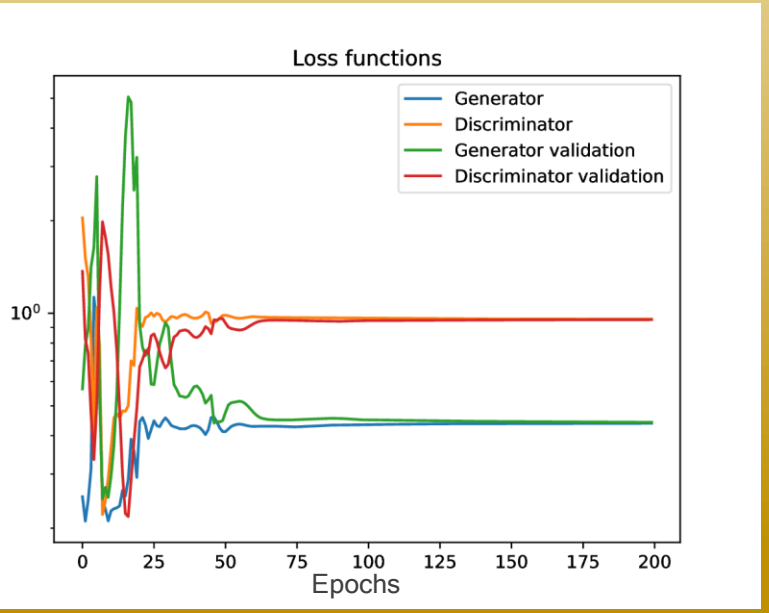


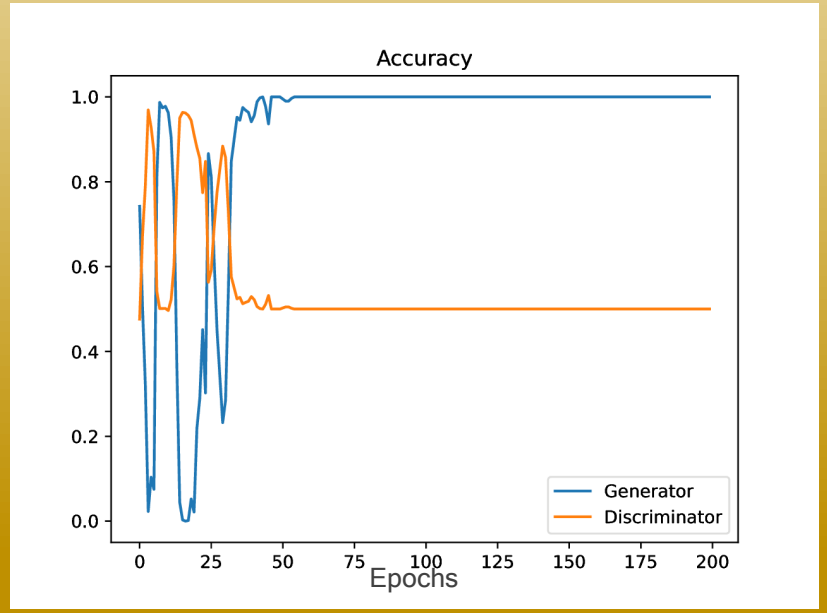


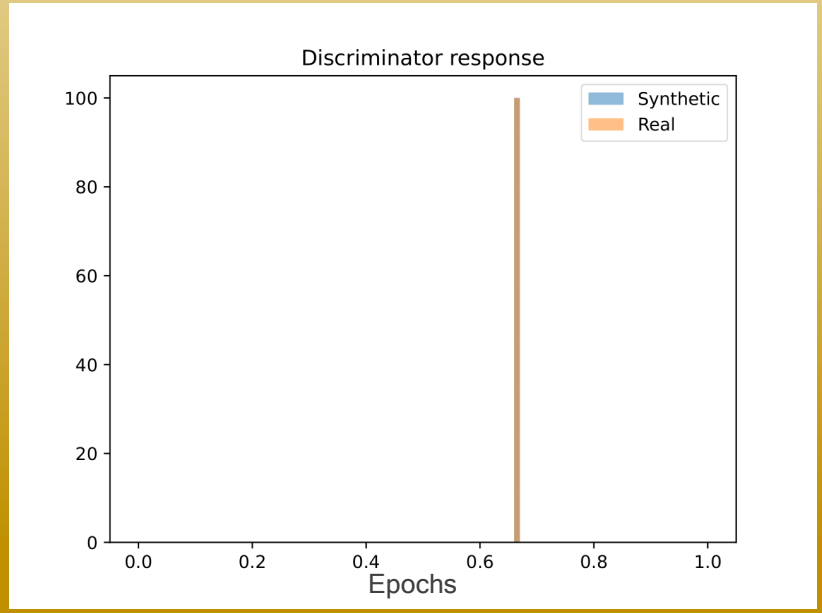


Accomplished Work, Results

- **GAN modello 28 Guillermo digital**







Accomplished Work, Results

- **GAN modello 28 Guillermo basic**