CERN openlab

Ensemble GAN for Simulation

Kristina Jaruskova (CERN, CTU), Sofia Vallecorsa (CERN)

CERN openlab Technical Workshop 2024

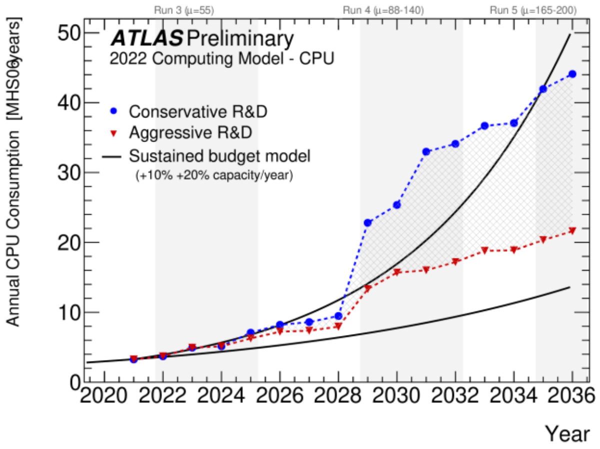
Introduction

Detector simulations

- Monte Carlo-based toolkits (Geant4) – particles interacting with matter
- HEP experiments – specific software frameworks using Geant4
- Computationally intensive
  - 50 % WLCG resources used for simulations[1]
  - E.g. ATLAS – aggressive R&D approach required for HL-LHC

Faster alternatives

- Deep learning models of different types
  - GANs, VAEs, NFs, GNNs, …
  - Development in experiment groups, Geant4, Openlab (IT)
- Focusing on electromagnetic calorimeters (ECAL)
  - High granularity -> most time demanding step in simulation (> 50 %[2])



ATLAS (2022)

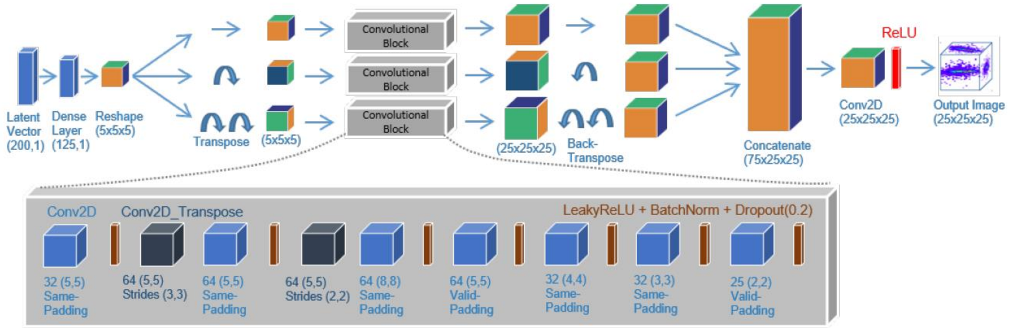
[1] The HEP Software Foundation. A Roadmap for HEP Software and Computing R&D for the 2020s. Comput. Softw. Big. Sci 2019.

[2] M. Rama. Fast Calorimeter Simulation in the LHCb Gauss Framework.CHEP 2018.

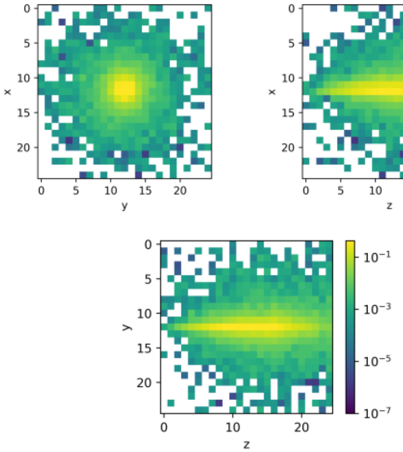
2DGAN and ECAL dataset

- **2DGAN model** [3]
  - 3-branch architecture with 2DConv layers
  - Generator output conditioned by primary energy E_p
  - 3-component loss:
    discriminator loss (true/fake) + E_{tot} loss + E_p prediction loss

- **Training dataset** (MC samples)
  - 3D image of a shower from a single e^- entering ECAL
    - 25x25x25 cells (~ 15,6k)
  - Primary energy E_p: 2 to 500 GeV



2DGAN generator architecture



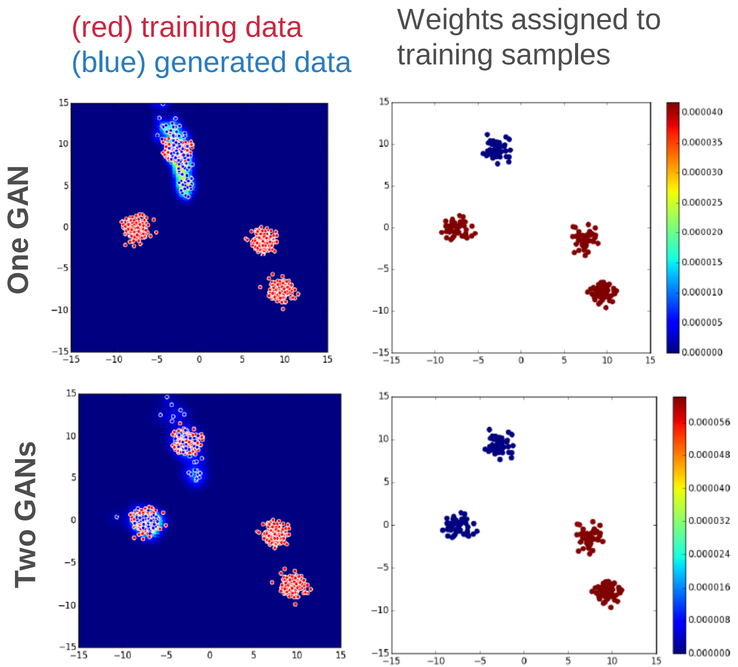
[3] Rehm F. Physics Validation of Novel Convolutional 2D Architectures for Speeding Up High Energy Physics Simulations. 2021

Multi-generator ensemble

- **AdaGAN**[4]
  - multi-generator ensemble model
  - Sequentially trained generators
  - Addressing the issue of **missing modes –** the next generator focuses on the weak spots of the previously trained generators

- **Principle – weighting training data**
  - Training samples are re-weighted before training the next GAN
  - Training samples poorly represented in the generated data
    → discriminator is confident in its classification
    → large weights assigned
  - Training samples well represented in the generated data
    → discriminator is confused
    → small weights assigned

- **Toy example – mixture of Gauss clusters**



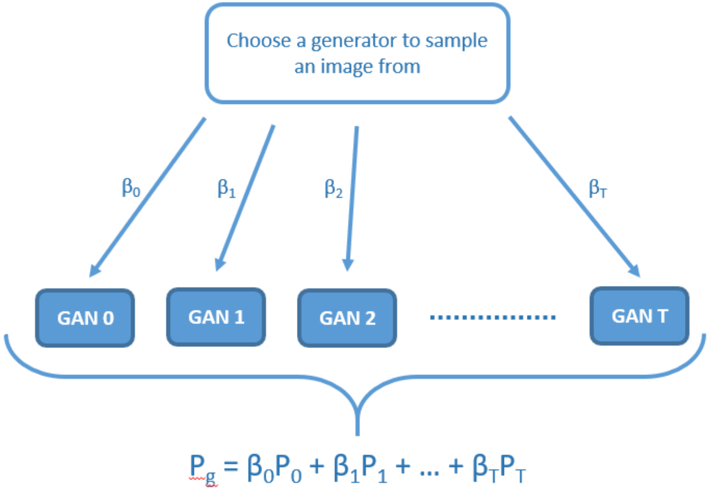
[4] I. Tolstikhin et al. AdaGAN: Boosting generative models 2017

Multi-generator ensemble

- **Distribution of the ensemble**
  - Generator distributions: $P_0, P_1, P_2, \ldots, P_T$
  - Component weights: $\boldsymbol{\beta} = (\beta_0, \beta_1, \beta_2, \ldots, \beta_T)$
  - Final distribution P_g: linear mixture of GAN distributions

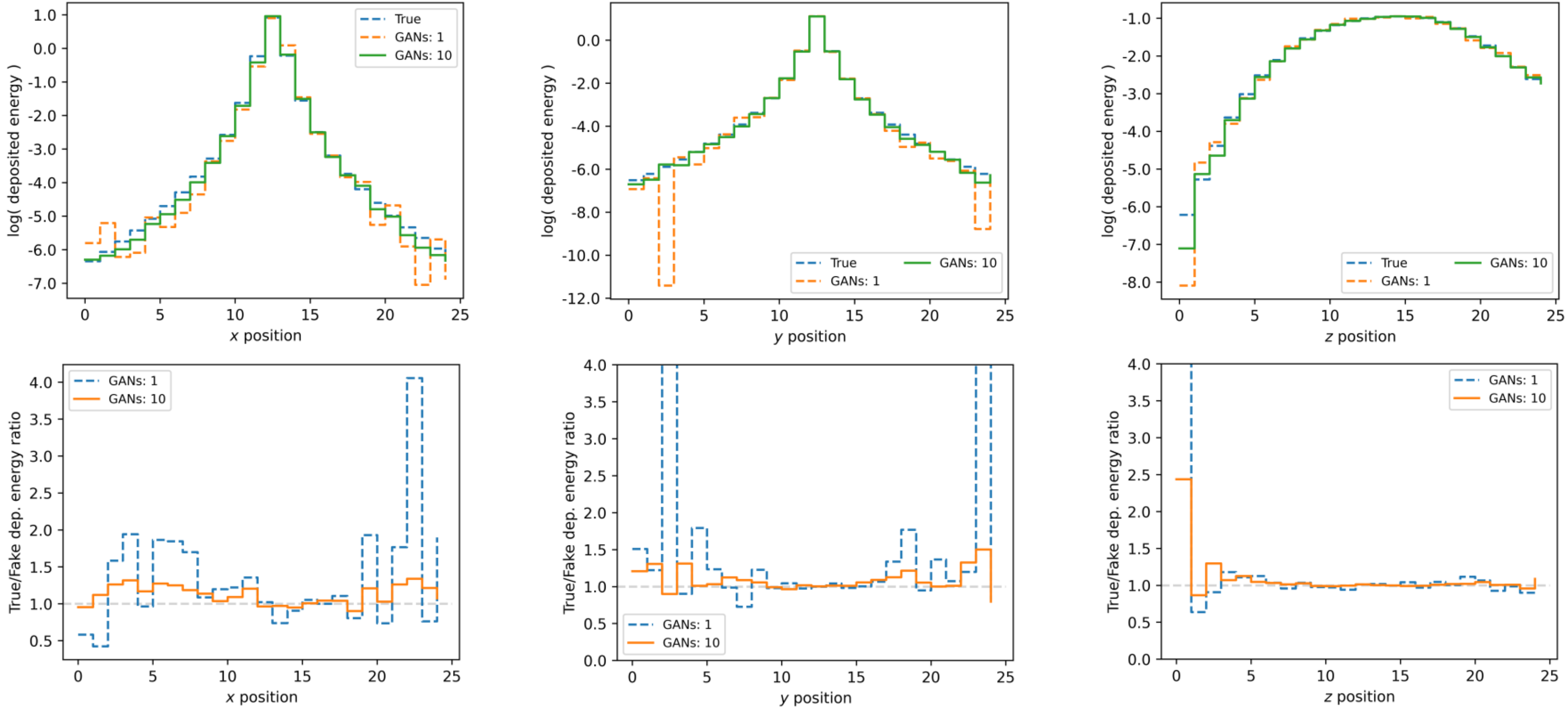
- **Generating from the ensemble**
  1. Draw generator index *i* from Cat(T, $(\beta_0, \beta_1, \beta_2, \ldots, \beta_T)$)
  2. Generate an image from GAN-*i* generator

- **Trained ensemble**
  - 2DGAN as a building block
  - **T = 10** GANs trained on MC data
  - $\boldsymbol{\beta}$ = (1/10, 1/10, …, 1/10)



Shower shapes

- Improvement at the edges

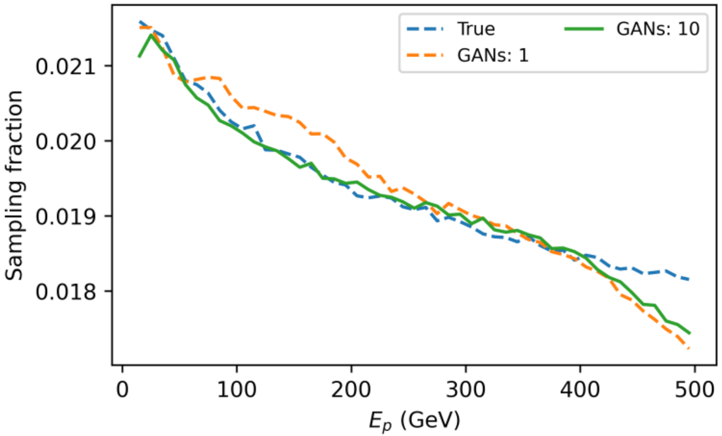


Sampling fraction

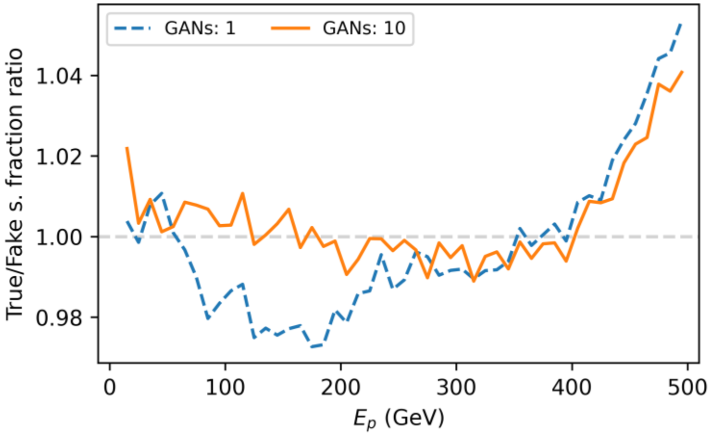
- Visible improvement in sampling fraction – ratio of the total deposited energy to the primary energy of a particle

Sampling fraction comparison

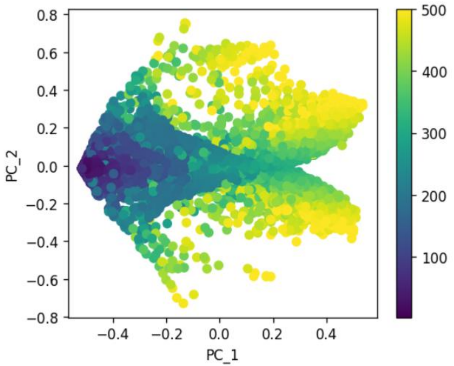
- Data split up into energy bins of 10 GeV



Ratio of the true SF to the SF of the generated data



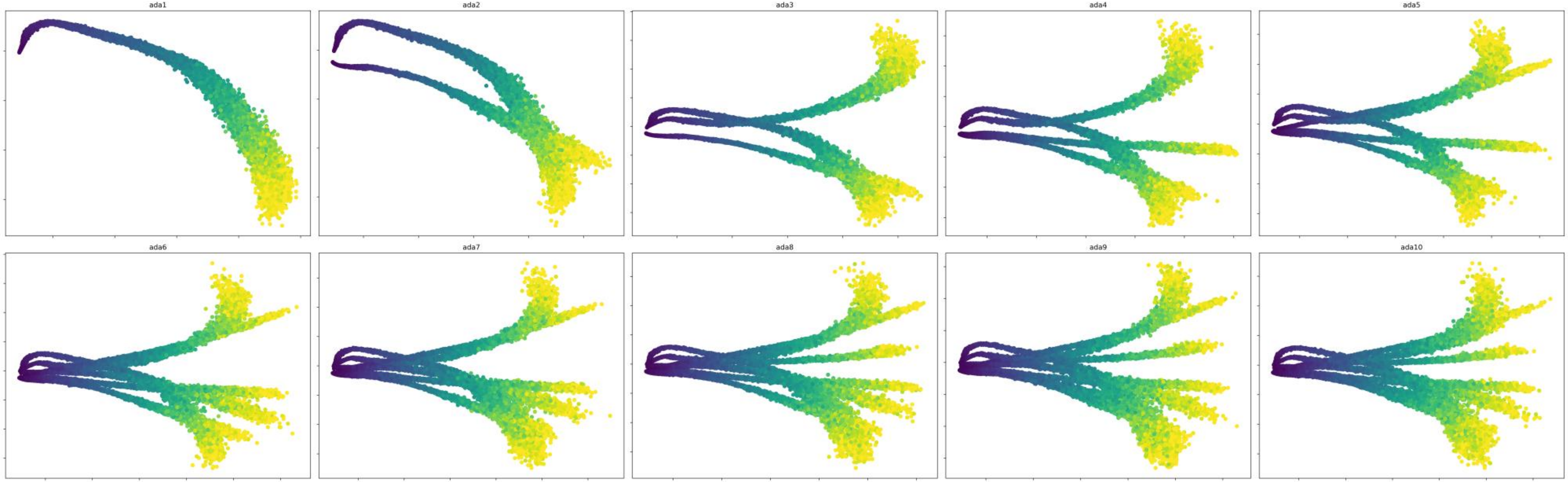
PCA on images



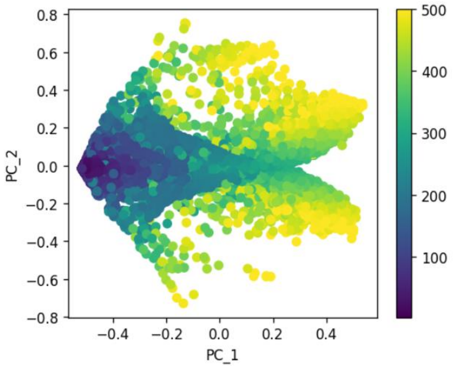
PCA on MC samples

Colorbar represents primary energy E_p: 2 to 500 GeV

PCA on generated samples



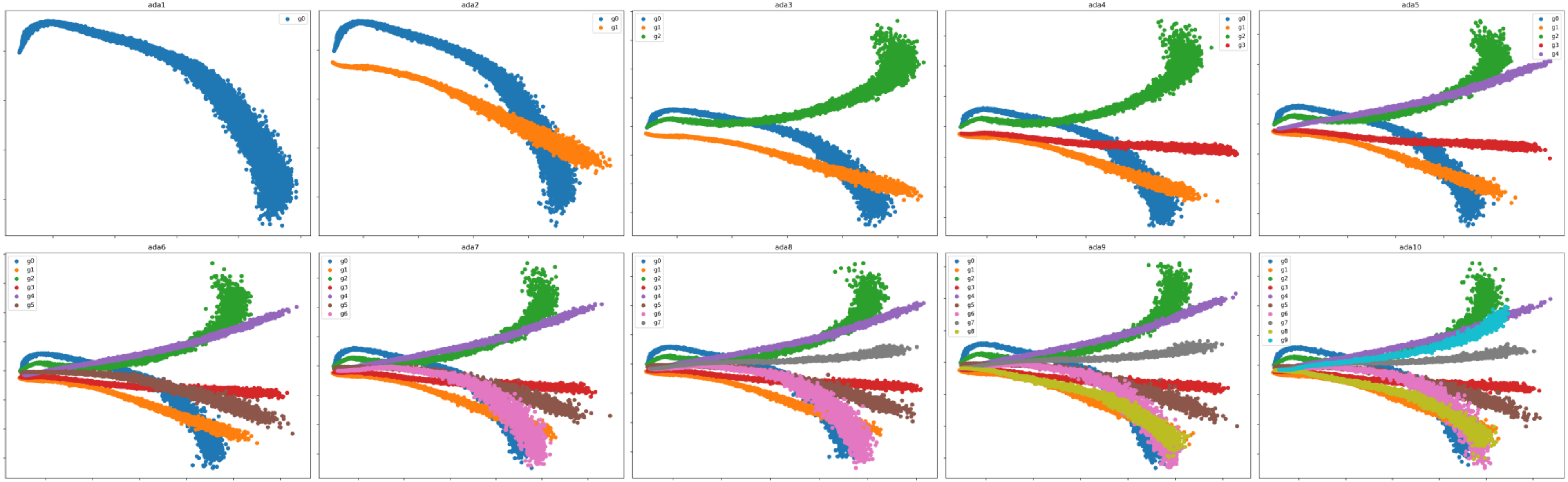
PCA on images



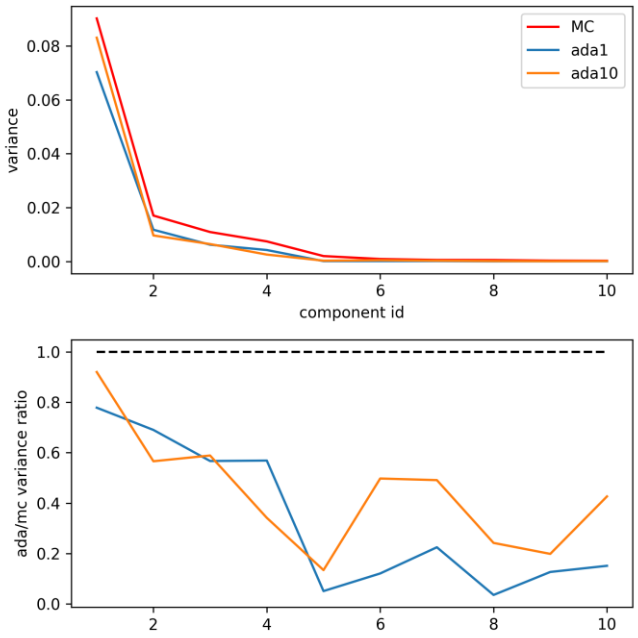
PCA on MC samples

Colorbar represents primary energy E_p: 2 to 500 GeV

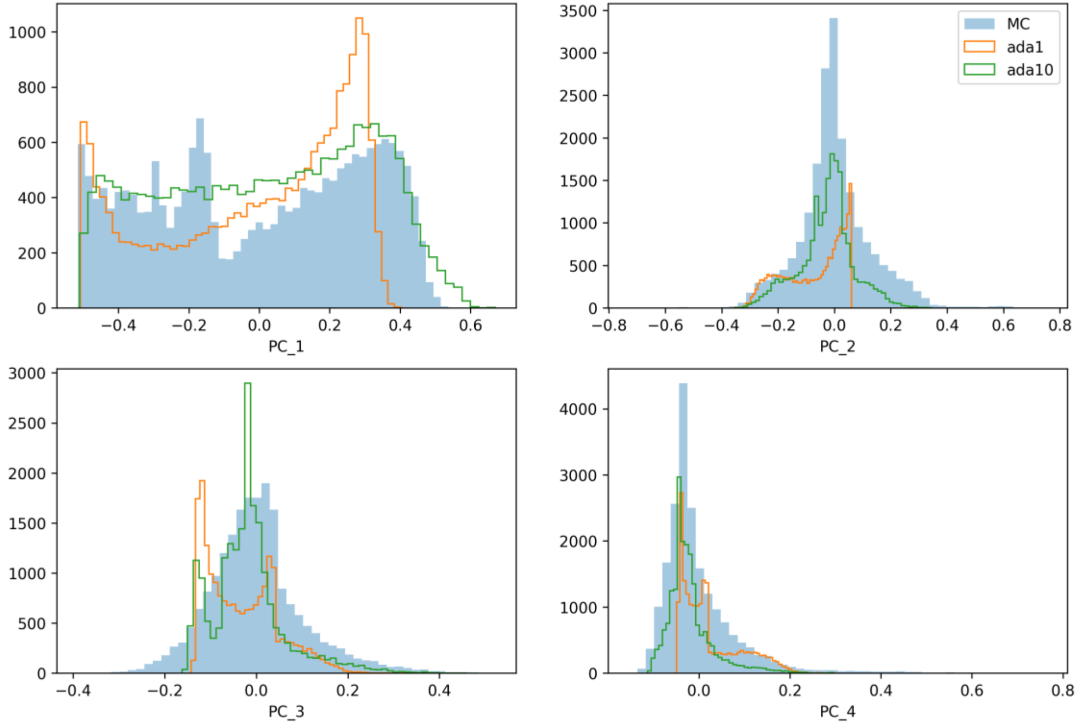
PCA on generated samples



Variance of PCs



Histograms of PCs

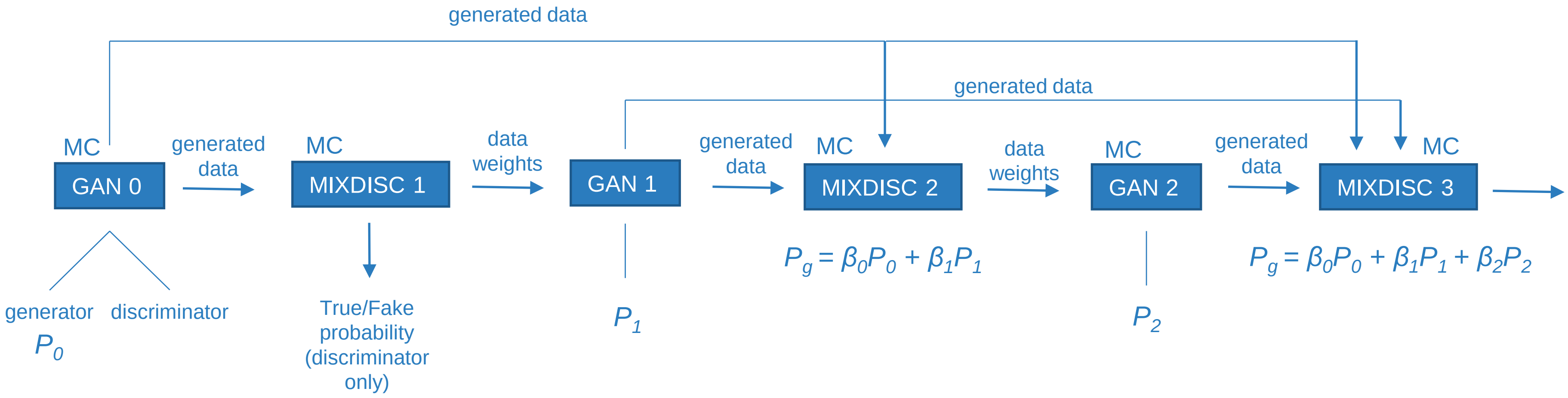


CERN
openlab

Ensemble training details

- MC = Monte Carlo training data
- GAN 0: trained on MC training data
- MIXDISC: trained on MC data and data from GAN 0 (1 : 1 ratio)
- GAN 1: trained on weighted MC data (weights from MIXDISC 1)
- MIXDISC 2: trained on MC data (no weights) and data from GAN 0 + GAN 1 (MC : generated = 1 : 1)
- GAN 2: trained on weighted MC data (weights from MIXDISC 2)

etc.



Data weights

- Main idea: minimize Jensen-Shannon divergence between data distribution P_d and the ensemble distribution P_g with the next GAN distribution Q

$$\min_{Q \in \mathbb{P}} D_{JS}((1-\beta)P_g + \beta Q \parallel P_d)$$

- In practice: any improvement on the J-S div. is enough
- Formula:

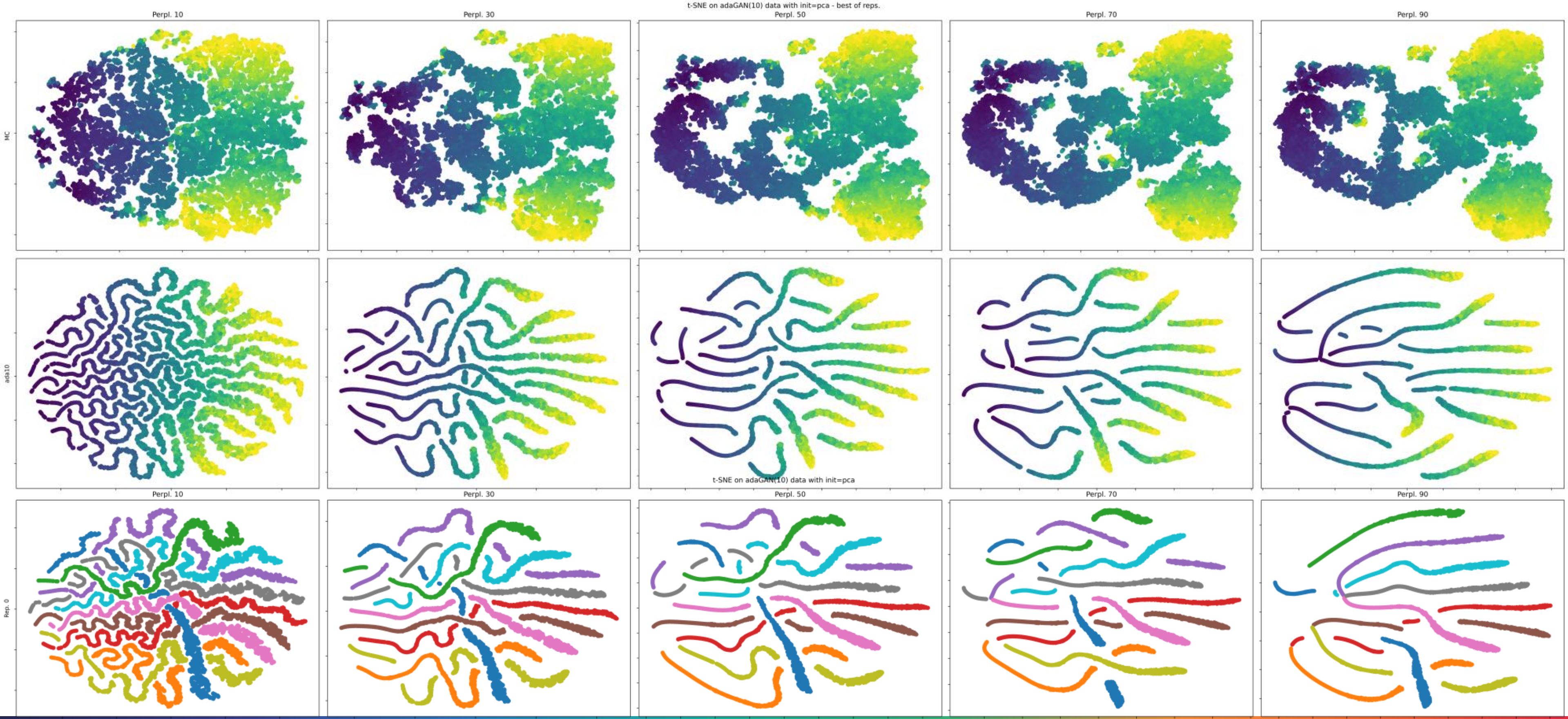
$$w_i = \frac{p_i}{\beta}\left(\lambda^* - (1-\beta)\frac{1-D(X_i)}{D(X_i)}\right)_+$$

$p_i = 1/N$ … empirical distribution of the training data

λ^* … normalization factor

- If $D(X_i) \sim 1$ → MIXDISC is certain it is training sample → high weight
- If $D(X_i) \sim 0{,}5$ → MIXDISC is confused → well represented in generated dataset → low weight

t-SNE



t-SNE on adaGAN(10) data with init=pca - best of reps.
Perpl. 10
Perpl. 30
Perpl. 50
Perpl. 70
Perpl. 90
MC
ada10
t-SNE on adaGAN(10) data with init=pca
Perpl. 10
Perpl. 30
Perpl. 50
Perpl. 70
Perpl. 90
Rep. 0