Understanding generative networks via weight distributions

based on arXiv:2305.16774

Luigi Favaro

in collaboration with R. Das, T. Heimel, C. Krause, T. Plehn, and D. Shih

November 9, 2023

UNIVERSITÄT HEIDELBERG
ZUKUNFT
SEIT 1386

Funded by
DFG Deutsche Forschungsgemeinschaft
German Research Foundation

Table of Contents

- Introduction
- Jet distortions
- Fast calorimeter simulations
- A CaloINN example

Evaluating generative networks

Huge interest in generative networks:

- Improve integration techniques (e.g. MadNIS);
- Event generation;
- Hadronization models;
- Fast Calorimeter Challenge 2022 [link]

Evaluating generative networks

1 Introduction

Huge interest in generative networks:

- Improve integration techniques (e.g. MadNIS);
- Event generation;
- Hadronization models;
- Fast Calorimeter Challenge 2022 [link]

Evaluating generative networks

- Classifiers are the best tools we have to test our generative networks;
- the output approximates the quantity:

$$C(x) = \frac{p_{true}(x)}{p_{true}(x) + p_{model}(x)} \qquad \longrightarrow \qquad \frac{p_{true}}{p_{model}}(x) = \frac{C(x)}{1 - C(x)}$$

- Classifiers are the best tools we have to test our generative networks;
- the output approximates the quantity:

$$C(x) = \frac{p_{true}(x)}{p_{true}(x) + p_{model}(x)} \quad \longrightarrow \quad \frac{p_{true}}{p_{model}}(x) = \frac{C(x)}{1 - C(x)}$$

- Optimal observable for a two hypothesis test according to the Neyman-Pearson lemma

Evaluating generative networks

- Classifiers are the best tools we have to test our generative networks;
- the output approximates the quantity:

$$C(x) = \frac{p_{true}(x)}{p_{true}(x) + p_{model}(x)} \qquad \longrightarrow \qquad \frac{p_{true}}{p_{model}}(x) = \frac{C(x)}{1 - C(x)}$$

- Optimal observable for a two hypothesis test according to the Neyman-Pearson lemma
- Proper training is essential: architecture, over-fitting, calibration, …

Table of Contents

2 Jet distortions

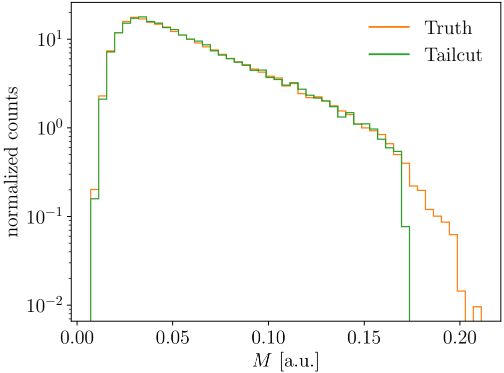
- Introduction
- Jet distortions
- Fast calorimeter simulations
- A CaloINN example

Toy example with `JetNet`

2 Jet distortions

`JetNet` example, apply distortions to the jet mass as in [1]:

- smear: smear with a Gaussian with $\mu = 1$ and $\sigma = 0.25$;
- shift: shift with a Gaussian with $\mu = 1.1$ and $\sigma = 0.05$.
- tailcut: remove the tail for $M > 0.17$;

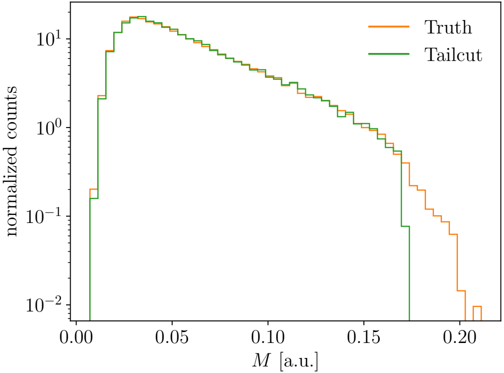


Toy example with `JetNet`

2 Jet distortions

`JetNet` example, apply distortions to the jet mass as in [1]:

- smear: smear with a Gaussian with $\mu = 1$ and $\sigma = 0.25$;
- shift: shift with a Gaussian with $\mu = 1.1$ and $\sigma = 0.05$.
- **tailcut**: remove the tail for $M < 0.17$;

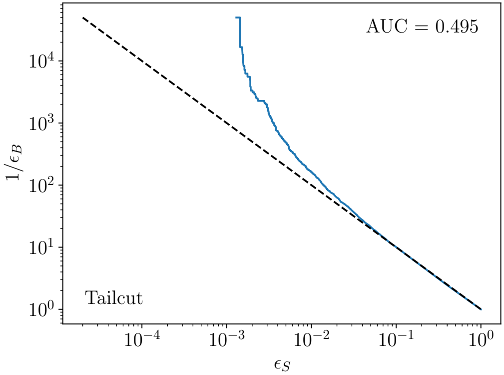


Train a `ParticleNet-Lite` classifier with 100k jets, validation on 50k jets.

[1] Kansal et al., arXiv:2211.10295

Toy example with `JetNet`
2 Jet distortions

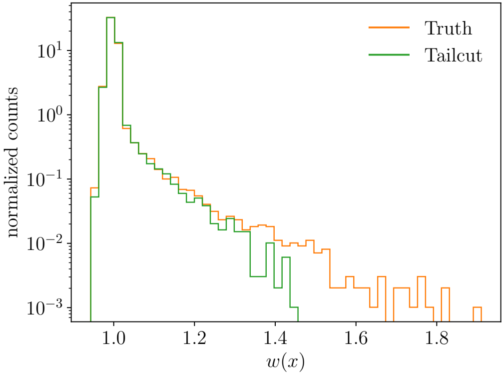
- Area Under the Curve (AUC) is not informative, AUC $= 0.495$;



Toy example with `JetNet`

2 Jet distortions

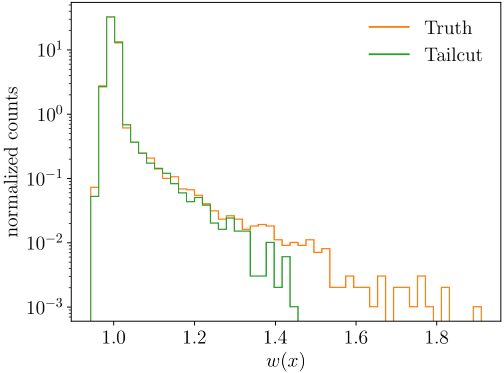
- Area Under the Curve (AUC) is not informative, AUC $= 0.495$;
- calculate and histogram the weight distribution;
- classifier builds an approx. $p_{true}/p_{tailcut}$;



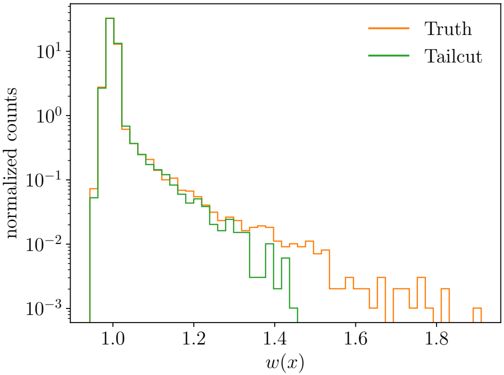
Toy example with `JetNet`

2 Jet distortions

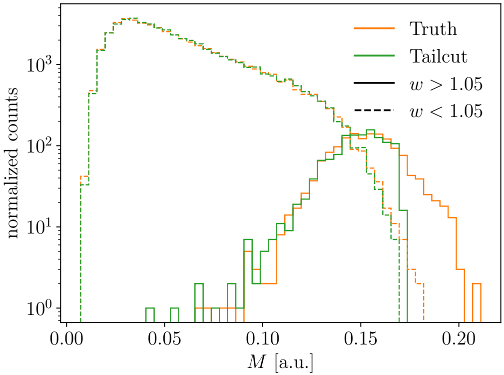
- Area Under the Curve (AUC) is not informative, AUC $= 0.495$;
- calculate and histogram the weight distribution;
- classifier builds an approx. $p_{true}/p_{tailcut}$;
- look at the w tail with clustering.



Toy example with `JetNet`



Weight distribution



Clustering plot

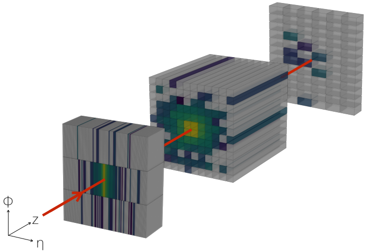
Table of Contents

3 Fast calorimeter simulations

- Introduction
- Jet distortions
- Fast calorimeter simulations
- A CaloINN example

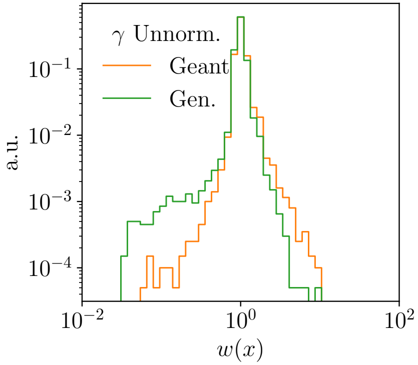
Geant4 vs GAN/VAE/NF/DM

- much harder problem, 100k samples/ $\mathcal{O}(500)$ voxels;
- high-dimensional density estimation;
- train on "low-level" features;
- based on caloGAN[2] data: e^+, γ, π showers;
- architecture $\to$ see Florian's talk.



[2] CaloGAN, Paganini et al.

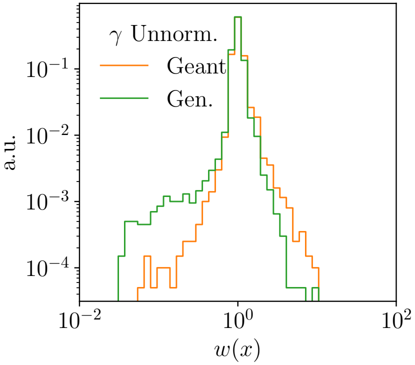
Geant4 vs GAN/VAE/NF/DM



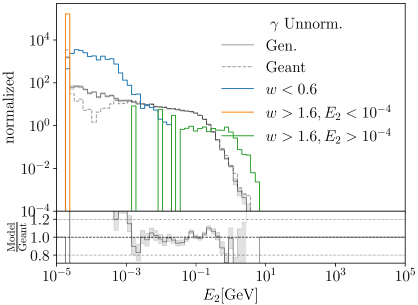
Weight distribution

Geant4 vs GAN/VAE/NF/DM

3 Fast calorimeter simulations

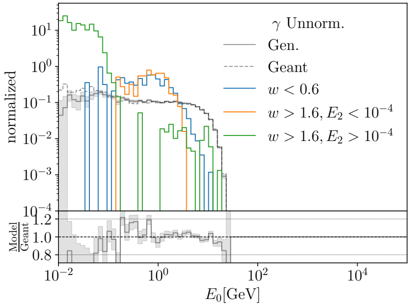


Weight distribution

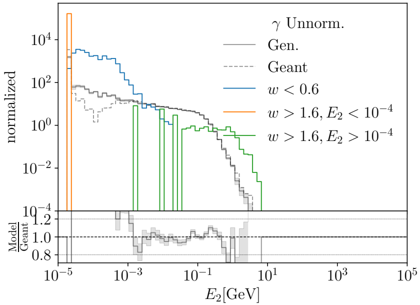


Energy dep. layer 2

Geant4 vs GAN/VAE/NF/DM
3 Fast calorimeter simulations



Energy dep. layer 0



Energy dep. layer 2

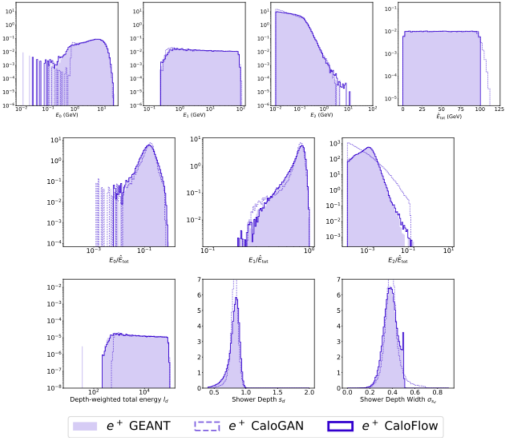
Table of Contents

- Introduction
- Jet distortions
- Fast calorimeter simulations
- A CaloINN example

Look back at CaloFlow

4 A CaloINN example

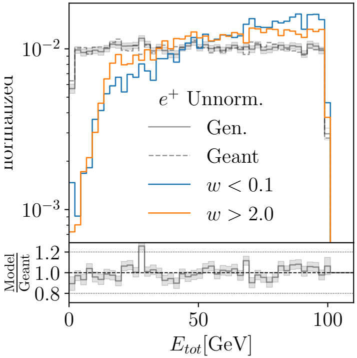
- Original CaloFlow [3] is less performing on e^+;
- physics of these showers should be simpler compared to π^+;
- understand the problem with a classifier $\rightarrow$ find a solution.



Krause C., Shih D., arXiv:2105.05285

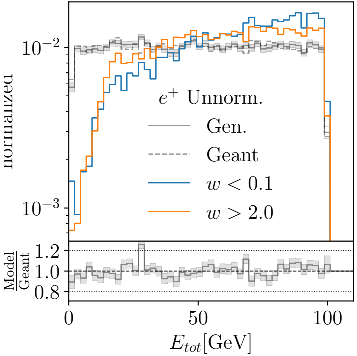
Look back at CaloFlow

4 A CaloINN example



Total energy deposition

Look back at CaloFlow
4 A CaloINN example



Total energy deposition

- Clear clustering for large energy depositions;
- those showers penetrate the entire detector;
- the last layer has the lowest fraction of energy deposited;
- is there something introducing a bias?

Noise injection

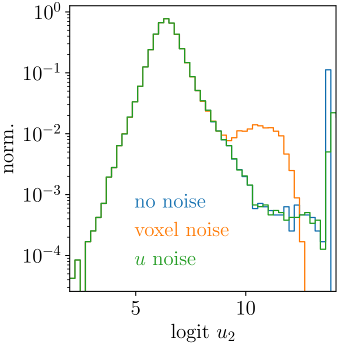
4 A CaloINN example

- Normalize by the energy:
- add some kind of noise;
- introduce energy encoding in new variables;
- ...

Noise injection

4 A CaloINN example

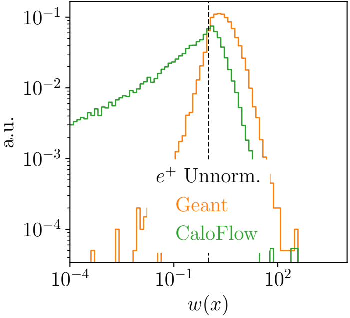
- Normalize by the energy:
- add some kind of noise;
- introduce energy encoding in new variables;
- ...
- order in which we apply them is important!



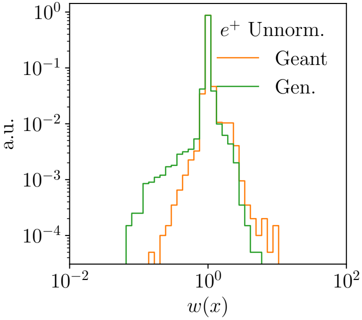
Energy variable u_2

Noise injection

4 A CaloINN example



CaloFlow weights



CaloINN weights

AUC: $0.86(1) \rightarrow 0.525(5)$

Conclusions

- Classifiers can systematically find all the failure modes of a generative model;
- we can easily extract weights from properly trained classifiers $\rightarrow p_{data}/p_{model}$;

Conclusions

- Classifiers can systematically find all the failure modes of a generative model;
- we can easily extract weights from properly trained classifiers $\rightarrow p_{data}/p_{model}$;
- I did not address *reweighting**;
- we can still learn from the weight distribution of a classifier;

*See. ELSA, arXiv:2305.07696

Conclusions

- Classifiers can systematically find all the failure modes of a generative model;
- we can easily extract weights from properly trained classifiers $\rightarrow p_{data}/p_{model}$;
- I did not address *reweighting**;
- we can still learn from the weight distribution of a classifier;
- the best tool to evaluate any surrogate ML model.

*See. ELSA, arXiv:2305.07696

Conclusions

- Classifiers can systematically find all the failure modes of a generative model;
- we can easily extract weights from properly trained classifiers $\rightarrow p_{data}/p_{model}$;
- I did not address *reweighting**;
- we can still learn from the weight distribution of a classifier;
- the best tool to evaluate any surrogate ML model.

*See. ELSA, arXiv:2305.07696

Outlook:

- Can we develop classifiers tailored for high-dimensional fast calorimeter simulations?

Conclusions

- Classifiers can systematically find all the failure modes of a generative model;
- we can easily extract weights from properly trained classifiers $\rightarrow p_{data}/p_{model}$;
- I did not address *reweighting**;
- we can still learn from the weight distribution of a classifier;
- the best tool to evaluate any surrogate ML model.

*See. ELSA, arXiv:2305.07696

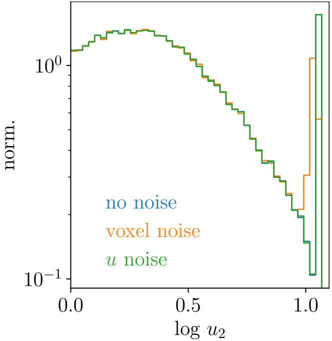
Outlook:

- Can we develop classifiers tailored for high-dimensional fast calorimeter simulations?

Thanks for your attention!

Backup

- Add uniform noise;
- separate energy generation from shape;
- order in which we apply them is important!



Energy variable u_2

Classifier params.

Parameter	Calorimeter
Optimizer	Adam
Learning rate	$2 \cdot 10^{-4}$
LR schedule	reduce on plateau
Decay factor	0.1
Decay patience (epochs)	10
Batch size	1000
Epochs	200
Number of layers	3
Hidden nodes	512
Dropout	30%
Activation function	leaky ReLU
Training samples	60k
Validation samples	20k
Testing samples	20k

Calibration curve