Refining Fast Simulation Using Machine Learning



Acelya Deniz Güngördü¹ Dorukhan Boncukcu¹ Kevin Pedro² Samuel Louis Bein³ Moritz Jonas Wolf⁴ Mirac Ekim Vural¹ Altan Cakir¹ on behalf of the CMS Collaboration

> ³UCLouvain ¹Istanbul Technical University ²Fermilab

> > ⁴University of Hamburg



Introduction

- CMS physics analysis rely on large quantities of simulated data
- LHC Phase 2: higher luminosity, complex new detectors, more data
- CMS experiment uses two simulation chains: 1. FullSim: Based on Geant4, high accuracy but slower 2. FastSim: Approximate techniques, faster but less accurate



FastSim

Speed



Network Architecture

- FastSim is a rapid Monte Carlo application for detector simulation and event reconstruction, approximately 10 times faster than FullSim.
- FastSim's speed advantage comes with reduced accuracy in some observables.
- R&D: Refine FastSim output with ML
- based on methodology: Fast Perfekt[1][2]

Data Sample and Method

Training sample: SUSY simplified model "T1tttt" simulated:

- 1. Gen \rightarrow FastSim + PU
- 2. Same Gen \rightarrow FullSim + PU

The aim is to establish a refined version of the FastSim data sample, which is more similar to the FullSim output, i.e., more accurate.

Matching jets using ΔR angular criterion

Network Inputs and Targets:

Input: FastSim variables $\mathbf{x}^{\text{Fast}} = 4$ DeepJet discriminators and p_{T} , $\vec{x} = (p_{\text{T}} \ b \ \text{CvB} \ \text{CvL} \ \text{QvG})^{\mathsf{T}}$ **Parameters:** $\mathbf{y} = p_{\mathrm{T}}^{\mathrm{GEN}}, \eta^{\mathrm{GEN}}, \text{ true hadron flavor (b, c, or light quark/gluon)}$ **Output:** Refined variables $\mathbf{x}^{\text{Refined}} = 4$ DeepJet discriminators and p_{T} **Target:** FullSim variables $\mathbf{x}^{Full} = 4$ DeepJet discriminators and p_T

Figure 2. Network Architecture of Refinement

Primary loss: Maximum Mean Discrepancy (MMD)

- Comparing ensembles of jets
- To cope with independent stochasticity in both simulation chains

Given two samples from P(X) and Q(Y):

$$\widehat{\mathsf{MMD}}(P,Q) = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n k(x_i, x_j) + \frac{1}{m^2} \sum_{i=1}^m \sum_{j=1}^m k(y_i, y_j) - \frac{2}{nm} \sum_{i=1}^n \sum_{j=1}^m k(x_i, y_j)$$

where n = m = batch size = 2048 and k : Gaussian kernel (adaptive σ)

Combine loss terms via MDMM (modified differential method of multipliers):

• Allows us to account for multiple loss terms such as MSE and loss terms that enforce boundary conditions or unitarity

Results



Training Framework



(a) Grid Search Mechanism

SB5_LP3_NHL1024_LR1eM06_PTtanhlogit_tanhNorm1200_newdataset Real-Time Loss Monitor ×	Loss Analyses of Trainings of tanhNorm_nSB_nLP_nNHL_LR_CvB_tanhlogit_log					
Train vs Validation Losses 🕀 🔿 🔍 🖱 🍙 =			All Train Validation			
1.3280	Loss Name	Loss Source	Min Value	Min Train Name	Max Value	Max Train Name
1.3260	mmdfixsigma_output_target	train	0.005971414283849299		0.20732822465896605	
1.3240	mmdfixsigma_output_target_hadflavSum	train	0.013946287022903561		0.3656244630217552	A
1.3220	mmd_output_target	train	0.002999835108872503		0.04803402514010668	A
	mmd_output_target_hadflav0	train	0.0068960847184062	A	0.1419171801507473	A 1
	mmd_output_target_hadflav4	train	0.019462304476648568		0.13744606597721576	
	mmd_output_target_hadflav5	train	0.0055282217329368		0.09179326069355011	&
	mmd_output_target_hadflavSum	train	0.008488135800696909		0.11944161854684353	
	mse_output_target	train	1.3828126349449157	A state of the	9.748155460357665	&
丶```` \ ଦ ଦ ଦ ଦ ଦ ଦ ଦ ଦ ଦ ଦ ବ ବ ବ ବ ବ ବ ବ ବ ବ	mse_input_output	train	0.029513584420084952		8.274864023208618	
 Train Loss Validation Loss Train Loss Trend Validation Loss Trend 	huber_output_target	train	0.06854319548606873		0.15068345448374748	B
						_
(b) Real-Time Loss Monitor	(c) Comparing of Losses in a Grid Search					

Figure 1. Training Framework

- A grid search system was integrated into the monitoring system.
- All models in the grid are displayed in the grouped trainings tab as a grouped training.

(c) Correlation matrix for FullSim, FastSim, FastSim Refined





Summary table lists best models and all loss values

Conclusion

Refinement of FastSim leads to significantly improved agreement with FullSim.

Training monitoring system implemented to track progress across various configurations. Refinement of jets, electrons, photons, and muons ongoing.

References

[1] S. Bein, P. Connor, K. Pedro, P. Schleper, and M. Wolf. Refining fast simulation using machine learning. In EPJ Web of Conferences, volume 295, page 09032. EDP Sciences, 2024.

[2] Moritz Wolf, Lars O. Stietz, Patrick L. S. Connor, Peter Schleper, and Samuel Bein. Fast Perfekt: Regression-based refinement of fast simulation. 10 2024.



Stu 2500

∂ 2000

1500

E 1000

of



(d) Histograms comparing Fast and Refined to Full (e) 1D discrepancy Fast-Full and Refined-Full



(f) Profiles of discrepancy from Full (mean and RMS)

FERMILAB-POSTER-24-0302-CMS-CSAID-PPD

Conference on Computing in High Energy and Nuclear Physics (CHEP) 2024