Generative models uncertainty estimation

Lucio Anderlini[1], Constantine Chimpoesh[2], Nikita Kazeev[2,3], Agata Shishigina[2] on behalf of the LHCb collaboration

[1]Universita e INFN, Firenze [2]HSE University, Moscow [3]Yandex School of Data Analysis, Moscow

20th International Workshop on Advanced Computing and Analysis Techniques in Physics Research (ACAT-2021)

Abstract

In recent years fully-parametric fast simulation methods based on generative models have been proposed for a variety of high-energy physics detectors. By their nature, the quality of data-driven models degrades in the regions of the phase space where the data are sparse. Since machine-learning models are hard to analyze from the physical principles, the commonly used testing procedures are performed in a data-driven way and can't be reliably used in such regions. In our work we propose three methods to estimate the uncertainty of generative models inside and outside of the training phase space region, along with data-driven calibration techniques. Test of the proposed methods on the LHCb RICH fast simulation is also presented.

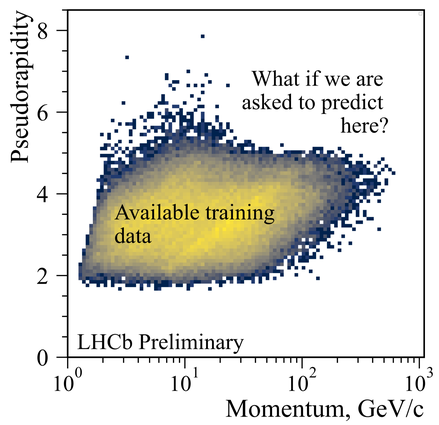
Fast data-driven simulation

See the talk on LHCb RICH fast simulation
Towards Reliable Neural Generative Modeling of Detectors, 30 Nov 8:20 UTC

1. The computational costs of detailed simulation based on Geant4 will be unsustainable for the upcoming runs at the major LHC experiments [1]
2. Fast data-driven simulation works by training a machine learning model to approximate the detector response [4]
3. The relationship between the detector readout y and the parameters of the particles x is not bijective, instead of $y = f(x)$ we need to learn $p(y|x)$
4. Machine learning introduces another source of uncertainty and possible mistakes. **We propose ML models that are aware how wrong they are.**

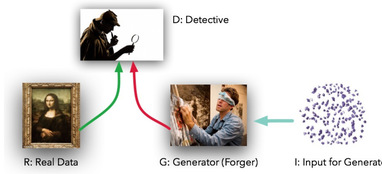
Challenges of the data-driven approach

1. An ML model is imperfect and training sample is finite
2. Training data are available only for a part of the phase space



Kinematic distribution of pions in the LHCb Run 2 calibration sample used in this study

Generative adversarial networks (GAN)



[5]

Cramer (Energy) distance [3]:

$$C(\mathfrak{P}||\mathfrak{Q}) = 2\mathbb{E}\|\mathfrak{X} - \mathfrak{Y}\|_2 - \mathbb{E}\|\mathfrak{X} - \mathfrak{X}'\|_2 - \mathbb{E}\|\mathfrak{Y} - \mathfrak{Y}'\|_2,$$

$\mathfrak{X}, \mathfrak{X}' \sim \mathfrak{P}$ – real data distribution, $\mathfrak{Y}, \mathfrak{Y}' \sim \mathfrak{Q}$ – model

Cramer GAN:

Let $G(Z): Z \to Y$ be the generator, $D(Y): Y \to R^d$ be the discriminator, $\boldsymbol{y}_r \sim P$ be a sample from the data and $\boldsymbol{y}_g, \boldsymbol{y}'_g \sim G(Z)$ be two independent samples from the generator. The loss functions:

$L_G = f(\boldsymbol{y}_r) - f(\boldsymbol{y}_g)$,

$L_D = -L_G + \lambda(\| \nabla_{\hat{y}} f(\hat{y}) \|_2 - 1)^2$,

where $f(\boldsymbol{y}) = \left\|D(\boldsymbol{y}) - D(\boldsymbol{y}'_g)\right\|_2 - \|D(\boldsymbol{y})\|_2$ and

$\hat{\boldsymbol{y}} = \epsilon \boldsymbol{y}_r + (1 - \epsilon)\boldsymbol{y}_g$, $\epsilon \sim U(0,1)$ are the interpolation points for the gradient penalty

Adversarial ensembles for uncertainty

An ensemble of several GANs with the following loss functions and training procedure:

$$f(\boldsymbol{y}) = \left\|D(\boldsymbol{y}) - D(\boldsymbol{y}'_g)\right\|_2 - \|D(\boldsymbol{y})\|_2$$

$$L_G = f(\boldsymbol{y}_r) - f(\boldsymbol{y}_g) - \alpha \left\|D(\boldsymbol{y}_g) - D(\boldsymbol{y}_{\cup_g})\right\|_2$$

$\boldsymbol{y}_{\cup_g}$ is a concatenation of the predictions of the ensemble, corresponding to a model with averaged probability density

Training schedule:
1. Train Cramer GANs with the classic loss ($\alpha = 0$)
2. Reinitialize the generators with random weights, retain the discriminators weights, set ($\alpha = 10$)
3. Train both the generators and the discriminators with our loss, decrease α according to a schedule

MC Dropout

Applying dropout at inference time provides a virtual ensemble [6]

- Bernoulli dropout: neuron zeroed with probability **p**
- Structured Bernoulli dropout: neuron with neighborhood of size *k* zeroed with probability **p**

Ensemble summarization

Ensembles use proportionally more resources for prediction than a single model, which is undesirable for fast simulation. We address this by approximating the ensemble with a single model as following. Assume that for given track parameters distribution of each output variable y_i is close to a Gaussian, then

$$\sigma_{\text{total}} = \sqrt{\sigma^2_{\text{intrinsic}} + \sigma^2_{\text{systematic}}}$$

- $\sigma^2_{\text{intrinsic}}$ - variance of distribution of y_i for the reference model
- $\sigma^2_{\text{systematic}}$ - systematic uncertainty in the training procedure

Then $\sigma_{\text{systematic}} =$

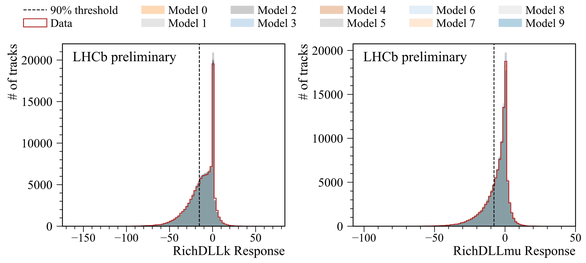
$$\sqrt{\frac{1}{2}\left(\mathbb{E}_{\text{ens}}\left[\left(y_i^{(1)} - y_i^{(2)}\right)^2\right] - \mathbb{E}_{\text{ref}}\left[\left(y_i^{(1)} - y_i^{(2)}\right)^2\right]\right)},$$

where $\mathbb{E}_{\text{ens}}$ and $\mathbb{E}_{\text{ref}}$ are the average operators computed across data produced by reference model and ensemble model respectively, and $y_i^{(1)}$ and $y_i^{(2)}$ are two examples independently sampled from the corresponding model.

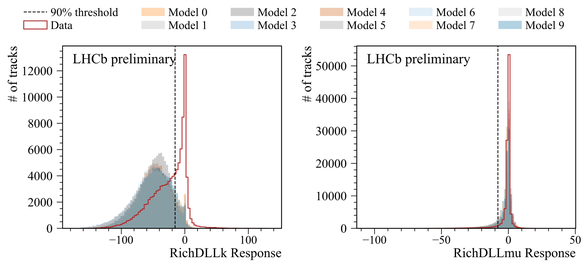
We train a neural network regression to approximate $\sigma_{\text{systematic}}$ from ensemble, thus allowing uncertainty computation with just a single model

Test case: LHCb RICH

1. Input variables: track momentum (P), pseudorapidity (η) and the number of tracks in the event
2. Output: 5 particle class likelihoods, expressed as differential logarithmic likelihoods (DLL)
3. Quantile transformation is applied to make all input and output features normal
4. Model: Cramer GAN with 5 fully-connected layers N. B. The production version of the LHCb RICH GAN uses 10 layers [2]

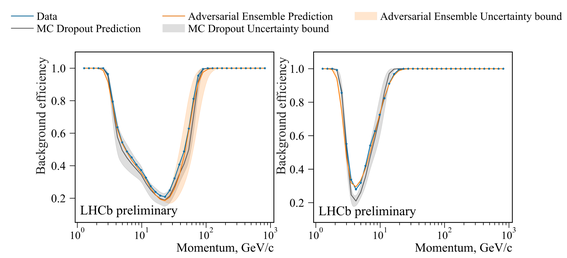


Distribution of the RichDLLs in a region with training data (the training region of the extrapolation scan). Models are consistent with the data and each other



Distribution of the RichDLLs in a region without training data (the testing region #9 of the extrapolation scan). Models differ from each other and the data.

Results: uniform train/test split



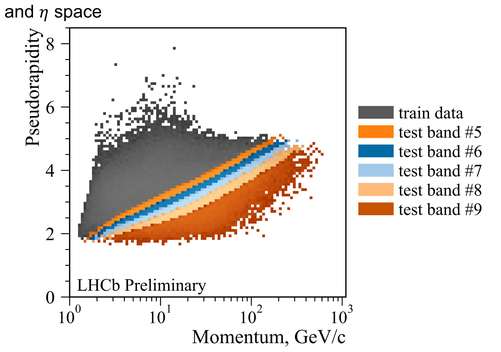
Background efficiency at 90% overall signal efficiency as a function of momentum. The data are uniformly split into training and testing parts

Acknowledgements

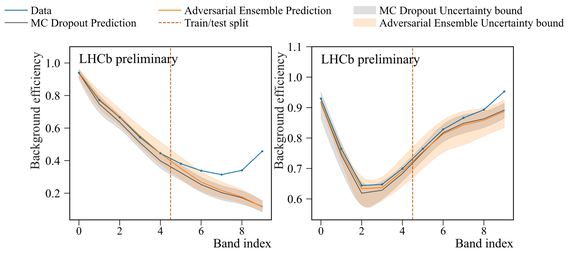
We are grateful to Denis Derkach for sharing his knowledge of LHCb fast simulation and review.

Results: extrapolation scan

The aim of the test is to access the performance of the models in the regions of the phase space where there are no data. We emulate this situation by splitting the data into train and test parts in P and η space



Training and testing datasets used for the extrapolation scan. The regions are separated by straight lines in the normalized space. Pions from LHCb Run 2 calibration sample. Each test band contains the same number of examples



Background efficiency at 90% overall signal efficiency as a function of the extrapolation scan test band index

Conclusion

- We present
  - Methods for estimating uncertainty of GANs with adversarial ensembles and MC dropout
    - Although in this work we only use Cramer GAN, both methods are applicable to any GAN
    - The ensembles have a desirable theoretical property: each model converges to local minimum of the unperturbed problem
  - A method for summarizing ensemble-based uncertainty estimation algorithms into a single model
- The methods are evaluated on the LHCb RICH dataset
  - For most of the bins, efficiency on the test data lies inside the error bounds of the efficiency of the model
  - In the extrapolation case, the uncertainly increases while getting further from the training region. However, the uncertainty does not increase sufficiently to account for the discrepancy in the furthest test regions.

References

[1] Davis, Adam. *Fast Simulations at LHCb*. No. LHCb-TALK-2019-404. 2019.
[2] Towards Reliable Neural Generative Modeling of Detectors, ACAT 2021
[3] Bellemare, Marc G., et al. "The cramer distance as a solution to biased wasserstein gradients." arXiv preprint arXiv:1705.10743 (2017).
[4] Maevskiy, Artem, et al. "Fast data-driven simulation of Cherenkov detectors using generative adversarial networks." Journal of Physics: Conference Series. Vol. 1525. No. 1. IOP Publishing, 2020.
[5] Generative Adversarial Networks (GANs) in 50 lines of code (PyTorch)
[6] Gal, Yarin, and Zoubin Ghahramani. "Dropout as a bayesian approximation: Representing model uncertainty in deep learning." international conference on machine learning. PMLR, 2016.