Generative models uncertainty estimation

L Anderlini[1], C Chimpoesh[2], N Kazeev[2] and A Shishigina[2] on behalf of the LHCb collaboration

[1] Universita e INFN, Firenze (IT), via Sansone 1, 50019 Sesto Fiorentino (FI), Italia
[2] HSE University, 20 Myasnitskaya st., Moscow 101000, Russia

E-mail: nikita.kazeev@cern.ch

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 analyse 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. A test of the proposed methods on the LHCb RICH fast simulation is also presented.

1. Introduction

In the last years, deep learning has become a common tool in natural science. Generative models, such as generative adversarial networks (GANs) [1], variational autoencoder (VAE) [2], normalising flows [3], and diffusion models [4], can learn to sample from a distribution efficiently. They are used for fully-parametric simulation of detectors – in place of computationally-intensive simulation from the physical principles, usually with Geant4 [5, 6, 7, 8].

Neural networks (NN) are black-box models that don't provide theoretical guarantees on the uncertainty of the prediction. This makes it difficult to use them in rigorous scientific reasoning. Uncertainty of machine learning models is an active area of research, but almost all works deal with classification and regression tasks, not generative modelling [9]. A recent work [10] shows how Bayesian normalising flows capture uncertainties.

Our work extends uncertainty estimation research by introducing new methods for estimating the uncertainty of GANs. Comparing with [10], in practice GANs are usually faster in training and inference, and more accurate than normalising flows, and thus are more widely used for fast simulations in high-energy physics. The contributions of this work are summarised as follows:

- We propose methods for estimating uncertainty of GANs
- We propose an approach to distillate the ensemble into a single model for efficient uncertainty computation

2. LHCb RICH fast simulation

In the LHCb experiment, the new fully-parametric simulation of Ring-Imaging Cherenkov detectors (RICH) is based on training a fully-connected Cramer GAN [11, 12] to approximate the reconstructed detector response. It is trained using the real data calibration samples [13].

RICH particle identification works as follows. First, the likelihoods for each particle type hypothesis are computed for each track. Second, the delta log-likelihoods are computed as the difference between the given hypothesis and the pion hypothesis. The variables are named `RichDLL*`, where `*` can be `k` (kaon), `p` (proton), `mu` (muon), `e` (electron) and `bt` (below the threshold of emitting Cherenkov light).

For the generator, input $x \in X \subset \mathbb{R}^{3+d_{\text{noise}}}$ consists of kinematic characteristics of particles (pseudorapidity η, momentum P, number of tracks) and random noise. The output $y \in Y \subset \mathbb{R}^5$ corresponds to the delta log-likelihoods.

3. Uncertainty estimation methods

3.1. MC dropout

Common dropout [14] acts as a regularisation to avoid overfitting when training an NN. The dropout is applied at both training and inference for Monte Carlo dropout (MC dropout) [15]. The prediction is no longer deterministic but depends on which NN nodes are randomly chosen to be kept. The MC dropout generates random predictions, and the latter has the interpretation of samples from a probabilistic distribution.

In our work, for MC dropout experiments, we add a dropout layer after each fully connected one and train with the same configuration as before. In the beginning, we used Bernoulli dropout, and then we experimented with Gaussian and Variational dropouts [16]. Finally, we found that the "structured" dropout modification (neuron with the neighborhood of arbitrary size k zeroed with probability $\mathbf{p}$) improves uncertainty quality.

During inference, for each batch we generate a fixed set of dropout masks as a way to have a virtual ensemble.

3.2. Adversarial deep ensembles

Ensemble methods are a widely-used heuristic uncertainty estimation method [17]. The core idea of ensembles is to introduce perturbations to the training procedure that shouldn't affect the outputs. Thus, the observed deviation in outputs is considered as uncertainty.

These perturbations can be implemented using randomisation techniques such as bagging and random initialisation of the NN parameters. Bagging on average uses 63% unique data points which leads to a biased performance estimate [17]. The diversity of the ensemble also tends to zero with the increase of training dataset size. While a reasonable outcome for in-domain uncertainty, it renders bagging unsuitable for out-domain uncertainty estimation.

In our method we start with the idea of diversity through NN weights. In addition to random initialisation, we add a component to the loss function that rewards the models for being different. For Cramer GAN [12] the loss function is modified as follows:

$$f(y) = ||D(y) - D(y'_g)||_2 - ||D(y)||_2, \tag{1}$$

$$L_G = f(y_r) - f(y_g) - \boldsymbol{\alpha||D(y_g) - D(y_{\bigcup g})||_2}, \tag{2}$$

where y_r are real data, y_g are generated data, and $y_{\bigcup g}$ is a concatenation of the predictions of the ensemble, corresponding to a model with averaged probability density; $\alpha \geq 0$ is a hyperparameter. The method is not specific to Cramer GAN and can be used with any GAN.

We reduce the influence of the adversarial component as the training progresses. Using pre-trained discriminators leads to more variety among the models. The overall training scheme is summarised in Algorithm 1. As α tends to zero, each ensemble member is trained without additional bias. This provide a principled advantage to adversarial ensembles: instead of heuristically perturbing the training objective as common in other methods [9], we take advantage of existence of many equally good local minima and search for a maximally diverse set of solutions to the unbiased problem of learning the distribution.

```
Algorithm 1: Adversarial ensembles training scheme
  1. Train several GANs with the classic loss ($\alpha = 0$);
  2. Reinitialize the generators with random weights; keep the discriminators weights; set
     $\alpha > 0$. We used $\alpha = 10$;
  3. Train both the generators and the discriminators with our loss Eq. (2), decrease $\alpha$
     gradually to 0;
```

3.3. Distillation

Running multiple generators for each simulated particle is excessively expensive for a fast simulation. Methods for distilling ensemble models are discussed in the literature [18], but they do not deal with generative models. Let us indicate the variance of the underlying PDF as $\mathrm{Var}^{(\mathrm{pdf})}$, and the variance due to differences among the trained generators, representing the uncertainty on the implicit model of the underlying PDF, as $\mathrm{Var}^{(\mathrm{train})}$. Evaluating the ensemble multiple times, at fixed conditions, we expect a distribution of outputs with a variance $\mathrm{Var}^{(\mathrm{tot})} = \mathrm{Var}^{(\mathrm{pdf})} + \mathrm{Var}^{(\mathrm{train})}$ as long as we accept the very reasonable hypothesis that the random component of each generator in the ensemble is not correlated to the random differences between generators of the ensemble. Let Y be a random variable – output of the generative model; $Y_{r,1}$ and $Y_{r,2}$ be the results of two independent inferences of a generator with fixed input conditions X (η, P, number of tracks in the case of RICH GAN), then

$$\mathrm{Var}^{(\mathrm{pdf})}(Y|X) = \frac{1}{2}\mathbb{E}_{\mathrm{reference}}\left[(Y_{r,2} - Y_{r,1})^2\right], \tag{3}$$

where $\mathbb{E}_{\mathrm{reference}}$ indicates the average over several samples obtained from the same, reference model. Instead, when sampling $Y_{e,1}$ and $Y_{e,2}$ from independent predictors in the ensemble, we expect their variance to be

$$\mathrm{Var}^{(\mathrm{tot})}(Y|X) = \frac{1}{2}\mathbb{E}_{\mathrm{ensemble}}\left[(Y_{e,2} - Y_{e,1})^2\right] \tag{4}$$

By training a regressor to predict $(Y_{r,1} - Y_{r,2})^2$ and $(Y_{e,1} - Y_{e,2})^2$ as a function of X optimised to minimise the Mean Squared Error, we obtain explicit models for $2 \cdot \mathrm{Var}^{(\mathrm{pdf})}(Y|X)$ and $2 \cdot \mathrm{Var}^{(\mathrm{tot})}(Y|X)$, which can be combined to assess $\mathrm{Var}^{(\mathrm{train})}$ as a function of the conditions X as

$$\mathrm{Var}^{(\mathrm{train})}(Y|X) = \mathrm{Var}^{(\mathrm{tot})}(Y|X) - \mathrm{Var}^{(\mathrm{pdf})}(Y|X) \tag{5}$$

In summary, naming $f_r(X)$ and $f_e(X)$ the trained predictors for $(Y_{r,1}-Y_{r,2})^2$ and $(Y_{e,1}-Y_{e,2})^2$, respectively, and assuming a normal distribution for the training error, the uncertainty on the generated samples, at a given condition X, is estimated as

$$\sigma_{\mathrm{syst}}(X) = \sqrt{\frac{1}{2}f_e\left(X\right) - \frac{1}{2}f_r\left(X\right)} \tag{6}$$

4. Results

4.1. Figure of merit

The quality of a fast simulation model in a particular phase space region is measured by the difference between the distributions of real and generated data. The objective of our uncertainty estimation methods is to predict this discrepancy.

To evaluate our methods, we compare background efficiency on real and generated data, computed as following. `RichDLL` values are commonly used for filtering tracks by a condition `RichDLLx > threshold`. We choose a threshold for `RichDLLx` so that 90% of all tracks with type `x` in the training dataset are accepted. Since the detector is not perfect, not only `x`-particles pass the selection, but there are also false positives. We plot the efficiency of the requirement `RichDLLx > threshold` on pions, as an approximation of the background selection efficiency. For a good uncertainty estimate real efficiencies should lie inside the uncertainty bounds for 68% of the bins.

4.2. Uniform split

We uniformly split the dataset into training and testing parts, containing 2 and 1 million examples, correspondingly. The results are presented in figure 1; for most of the bins efficiency on the test data lies inside the error bounds of the efficiency of the model. The side effect of MC dropout is dropout layers themselves. The dropout acts as a regularizer: it reduces variance but may increase the bias in the resulting model. Thus, the MC dropout model could be overregularized (more significant RichDLLmu efficiency error compared to ensembles).

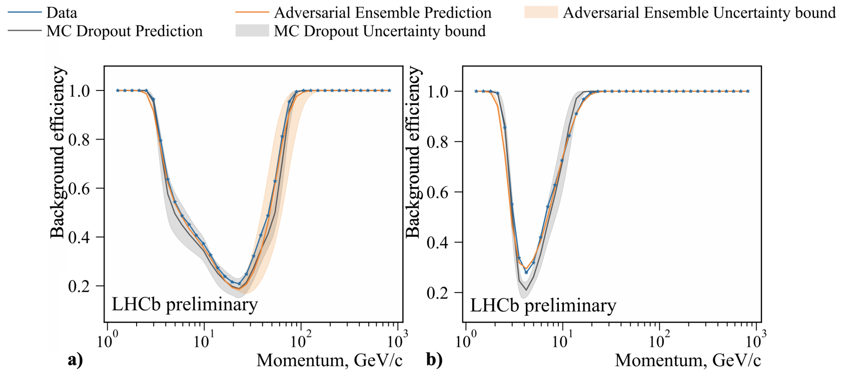

Figure 1. Fraction of pions in the test dataset selected by applying requirements on `RichDLLK` (a) and `RichDLLmu` (b) with thresholds corresponding to 90% selection efficiency on kaons and muons, respectively.

4.3. Extrapolation scan

This test aims to assess 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 as shown in figure 2. The train part contains 947933 examples, and the test part contains 523917 examples. The models are trained on the train part. For evaluation, 101917 examples are samples from each training and test band. The high number of tracks makes the statistical uncertainty of the efficiency estimation negligible, both for real and generated data.

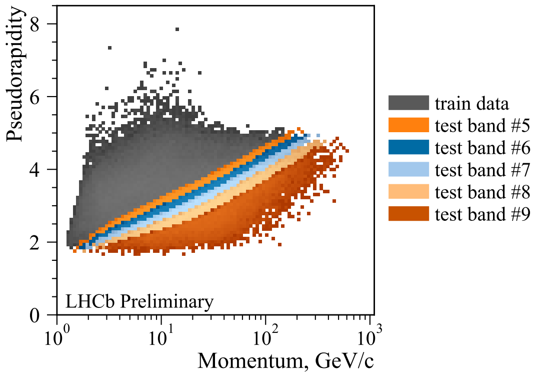

Figure 2. Training and testing datasets used for the extrapolation scan. The regions are separated by straight lines in the normalised space. Pions data correspond to the ones present in the LHCb Run 2 calibration sample [13]. Each test band contains the same number of examples.

The results are present in figure 3. The adversarial ensembles show wider uncertainty bounds; nevertheless, both methods underestimate the uncertainty in the last bands. For kaons, figure 3 (a), real background efficiency decreases for the first half of the test part, then starts to increase. This demonstrates a great obstacle for extrapolating with purely machine learning methods, as the qualitative change can't be reasonably predicted from the training data without incorporating additional knowledge. For muons, figure 3 (b), efficiency in the test part continues the trend observed in the training part, resulting in uncertainty bands that contain the real efficiencies for the most bands.

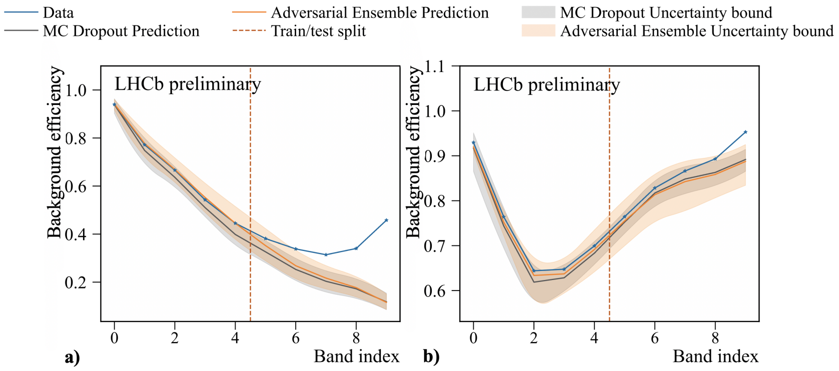

Figure 3. Fraction of pions selected by applying requirements on `RichDLLK` (a) and `RichDLLmu` (b) with thresholds corresponding to 90% selection efficiency on kaons and muons, respectively as a function of the extrapolation scan test band index.

5. 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 a local minimum of the unperturbed problem. We propose a method for distilling ensemble-based uncertainty estimation into a single model for fast inference.

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 uncertainty 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 where the detector operational conditions are much different from those corresponding to the training sample. Our code is available online [19, 20].

This work is a first step towards incorporating GAN uncertainty into high-energy physics fast simulation. We see the future directions for the research as following. Better correspondence between uncertainty and the real/simulated data difference could be achieved, along with more robust uncertainty growth outside the training region. We use background efficiency as a proxy metric, evaluating GAN uncertainty impact on the uncertainty of the final measurement would be more instructive.

Acknowledgments

We are grateful to Denis Derkach for sharing his knowledge of LHCb fast simulation and review. The publication was prepared within the framework of the Academic Fund Program at the HSE University in 2022 (grant №22-00-02). This research was supported in part through computational resources of HPC facilities at HSE University [21].

References

[1] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A and Bengio Y 2014 *Advances in neural information processing systems* **27**

[2] Kingma D P and Welling M 2013 *arXiv preprint arXiv:1312.6114*

[3] Rezende D and Mohamed S 2015 Variational inference with normalizing flows *International conference on machine learning* (PMLR) pp 1530–1538

[4] Sohl-Dickstein J, Weiss E, Maheswaranathan N and Ganguli S 2015 Deep unsupervised learning using nonequilibrium thermodynamics *International Conference on Machine Learning* (PMLR) pp 2256–2265

[5] Derkach D, Kazeev N, Ratnikov F, Ustyuzhanin A and Volokhova A 2020 *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **952** 161804

[6] Vallecorsa S 2018 *Journal of Physics: Conference Series* **1085** 022005 URL `https://doi.org/10.1088/1742-6596/1085/2/022005`

[7] Chekalina V, Orlova E, Ratnikov F, Ulyanov D, Ustyuzhanin A and Zakharov E 2019 *EPJ Web Conf.* **214** 02034 URL `https://doi.org/10.1051/epjconf/201921402034`

[8] Agostinelli S *et al.* 2003 *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **506** 250–303 ISSN 0168-9002

[9] Gawlikowski J and et al 2022 A survey of uncertainty in deep neural networks (*Preprint* `2107.03342`)

[10] Bellagente M, Haußmann M, Luchmann M and Plehn T 2021 Understanding event-generation networks via uncertainties (*Preprint* `2104.04543`)

[11] Maevskiy A, Derkach D, Kazeev N, Ustyuzhanin A, Artemev M and Anderlini L 2020 *Journal of Physics: Conference Series* **1525** 012097 URL `https://doi.org/10.1088/1742-6596/1525/1/012097`

[12] Bellemare M G, Danihelka I, Dabney W, Mohamed S, Lakshminarayanan B, Hoyer S and Munos R 2017 The Cramer Distance as a Solution to Biased Wasserstein Gradients (*Preprint* `1705.10743`)

[13] Aaij R, Anderlini L, Benson S, Cattaneo M, Charpentier P, Clemencic M, Falabella A, Ferrari F, Fontana M, Gligorov V V *et al.* 2019 *EPJ Techniques and Instrumentation* **6** 1

[14] Srivastava N, Hinton G, Krizhevsky A, Sutskever I and Salakhutdinov R 2014 *Journal of Machine Learning Research* **15** 1929–1958 URL `http://jmlr.org/papers/v15/srivastava14a.html`

[15] Gal Y and Ghahramani Z 2016 Dropout as a bayesian approximation: Representing model uncertainty in deep learning (*Preprint* `1506.02142`)

[16] Molchanov D, Ashukha A and Vetrov D 2017 Variational dropout sparsifies deep neural networks *Proceedings of the 34th International Conference on Machine Learning* (*Proceedings of Machine Learning Research* vol 70) (PMLR) pp 2498–2507 URL `https://proceedings.mlr.press/v70/molchanov17a.html`

[17] Lakshminarayanan B, Pritzel A and Blundell C 2017 Simple and scalable predictive uncertainty estimation using deep ensembles

[18] Malinin A, Mlodozeniec B and Gales M 2019 Ensemble distribution distillation (*Preprint* `1905.00076`)

[19] URL `https://gitlab.com/lambda-hse/lhcb-rich-gan-uncertainty`

[20] URL `https://gitlab.com/lambda-hse/gan-uncertainty-ensembles`

[21] Kostenetskiy P S, Chulkevich R A and Kozyrev V I 2021 *Journal of Physics: Conference Series* **1740** 012050