Exhaustive Neural Importance Sampling applied to Monte Carlo event generation

Sebastian Pina-Otey [1 2] **Federico Sánchez** [3] **Thorsten Lux** [2] **Vicens Gaitan** [1]

Abstract

The generation of accurate neutrino-nucleus cross section models needed for neutrino oscillation experiments requires simultaneously the description of many degrees of freedom and precise calculations to model nuclear responses. The detailed calculation of complete models makes the Monte Carlo generators slow and impractical. We present Exhaustive Neural Importance Sampling (ENIS), a method based on normalizing flows to find a suitable proposal density for rejection sampling automatically and efficiently, and discuss how this technique solves common issues of the rejection algorithm.

1. Motivation

In modern science and engineering disciplines, the generation of random samples from a theoretical model, the target probability density function $p(\mathbf{x})$, has become essential. A standard algorithm to obtain Monte Carlo (MC) samples is the rejection sampling (Casella et al., 2004), which produces i.i.d. samples from the target density via an auxiliary proposal function. The proposal has to satisfy being a density which can both be sampled from and evaluated efficiently, as well as being as close to the target density as possible. The main disadvantages of the method are as follows (Robert & Casella, 2004): i) Designing the proposal function close to a particular target density can be very costly in human time. ii) If taken a generic proposal function, such as a uniform distribution over the domain, the algorithm is usually very inefficient. iii) The inefficiency grows rapidly with the number of dimensions.

[1]Aplicaciones en Informática Avanzada (AIA), Sant Cugat del Vallès (Barcelona), Spain [2]Institut de Física d´Altes Energies (IFAE) - Barcelona Institute of Science and Technology (BIST), Bellaterra (Barcelona), Spain [3]University of Geneva, Section de Physique, DPNC, Geneva, Switzerland. Correspondence to: Sebastian Pina-Otey <pinas@aia.es>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 108, 2020.

An approach of the usage of normalizing flows to find a suitable proposal which solves these drawbacks for a given target density has been suggested previously as Neural Importance Sampling (NIS) (Müller et al., 2018), focused on the integration of functions via importance sampling (Kahn & Marshall, 1953). Normalizing flows provide an expressive family of parametrized density functions $q_\phi(\mathbf{x})$ through neural networks, by defining a differentiable and invertible transformation from a base distribution to a target distribution, allowing to evaluate and sample from complex distributions by transforming simpler ones. The concept of integrating via importance sampling with normalizing flows for High Energy Physics (HEP) has been explored in other works to obtain top-quark pair production and gluon-induced multijet production (Bothmann et al., 2020) or to simulate collider experimental observables for the Large Hadron Collider (Gao et al., 2020).

In this work we further explore the possibility of utilizing normalizing flows to find a proposal function for a given target density to perform rejection sampling for MC samples, and analyze its viability through the following points: i) We discuss the importance of adding a background density to the target one to assure the coverage of the whole phase-space when performing rejection sampling. ii) We define a two-phase training scheme for the normalizing flow to boost initial inefficiency in the optimization when adjusting the initialized random density toward the target one. iii) We measure the performance of the method and argue for relaxing the rejection sampling constant factor k to improve largely the efficiency of acceptance while quantifying the error committed in doing this approximation via the concept of coverage. Considering the proposed algorithm covers the whole phase-space by modifying NIS with the background density, we denote this method by Exhaustive Neural Importance Sampling (ENIS).

We apply the above algorithm to an HEP problem, in the form of the charged current quasielastic (CCQE) cross section for antineutrinos interactions with nuclei, performing in-depth analysis and discussion of the efficiency of the method. We will show that ENIS opens the possibility to incorporate efficiently complex theoretical models in the existing MC models enhancing the physics reach of running

and future neutrino oscillation experiments.

2. Methodology

2.1. Rejection sampling

Rejection sampling is a well-known technique (Casella et al., 2004; Robert & Casella, 2004) to obtain MC samples from a target density $p(\mathbf{x})$ which can be evaluated (up to a constant), but cannot be sampled through the inverse transform. It relies on an auxiliary proposal function $q(\mathbf{x})$, from which one should be able to sample from and evaluate efficiently. A constant $k > 0$ is introduced which has to satisfy that

$$k \cdot q(\mathbf{x}) \geq p(\mathbf{x}) \quad \forall\, \mathbf{x} : p(\mathbf{x}) > 0. \tag{1}$$

The resulting function $k \cdot q(\mathbf{x})$ is called the comparison function.

The procedure to sample from the target density is then as follows: i) A sample $\mathbf{x}$ is generated following $q(\mathbf{x})$, $\mathbf{x} \sim q(\mathbf{x})$. ii) A random number u is generated uniformly in the range $[0, k \cdot q(\mathbf{x})]$, $u \sim \text{Unif}(0, k \cdot q(\mathbf{x}))$. iii) If u fulfills the condition $u \leq p(\mathbf{x})$, the sample is accepted; otherwise, it is rejected.

2.2. Optimizing the parameters of the normalizing flow

Consider a target probability density function $p(\mathbf{x})$, approximated by the normalizing flow (Papamakarios et al., 2019) $q_\phi(\mathbf{x})$. The parameters ϕ of $q_\phi(\mathbf{x})$ are obtained by minimizing the Kullback-Leibler divergence (KL divergence) (Kullback & Leibler, 1951) between both distributions. When minimizing with respect to ϕ, the KL divergence is simplified to

$$\begin{aligned}\arg\min_\phi D_{\text{KL}}\big(p(\mathbf{x}) \,\|q_\phi(\mathbf{x})\big) \\ = \arg\max_\phi \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}\left[\log q_\phi(\mathbf{x})\right].\end{aligned}$$

Müller *et al.* (Müller et al., 2018) propose a solution for computing the gradient with respect to ϕ when $p(\mathbf{x})$ can be evaluated but not sampled from, via importance sampling (Kahn & Marshall, 1953),

$$\begin{aligned}\nabla_\phi \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}\left[\log q_\phi(\mathbf{x})\right] \\ \approx \frac{1}{N}\sum_{i=1}^{N} w(\mathbf{x}_i)\nabla_\phi \log q_\phi(\mathbf{x}_i),\end{aligned} \tag{2}$$

with $\mathbf{x}_i \sim q_\phi(\mathbf{x})$ and the weights defined as $w(\mathbf{x}) = p(\mathbf{x})/q_\phi(\mathbf{x})$. Notice how we only need to be able to evaluate $p(\mathbf{x})$ to compute this quantity. With this gradient, we are able to minimize the KL divergence if the support of $q_\phi(\mathbf{x})$ (i.e., the domain where the function is nonzero) contains the support of $p(\mathbf{x})$ to perform the importance sampling of Eq. (2).

In this work we will focus on the implementation of normalizing flows, the Neural Spline Flow (NSF) (Durkan et al., 2019b).

2.3. Relevance of background noise

To ensure the proper $p(\mathbf{x})$ support for the importance sampling in Eq. (2) and for the rejection sampling, we introduce the concept of a background density function, $p_{\text{bg}}(\mathbf{x})$. In HEP, as in many other scientific areas, the density is restricted to a certain domain of $\mathbf{x} \in \mathbb{R}^D$, e.g., the cosine has to be in $[-1, 1]$, etc. Hence $p_{\text{bg}}(\mathbf{x})$ should be a density that has a support beyond these phase-space boundaries. In what follows, a uniform distribution will be considered, with limits in each dimension according to the phase-space of that coordinate.

The background density $p_{\text{bg}}(\mathbf{x})$ will be used for two tasks:

(i) Improve initial training: At the beginning of the training, we cannot assure that the support of $q_\phi(\mathbf{x})$ contains the one of $p(\mathbf{x})$. Hence, instead of using $q_\phi(\mathbf{x})$ for the importance sampling of Eq. (2), $p_{\text{bg}}(\mathbf{x})$ will be used during the warm-up phase of the training, as will be shown in Sec. 2.4.

(ii) Ensure exhaustive coverage of the phase-space: The target density $p_{\text{target}}(\mathbf{x})$ that the neural network will learn will be constructed as a linear combination of the true target density $p(\mathbf{x})$ and the background $p_{\text{bg}}(\mathbf{x})$:

$$p_{\text{target}}(\mathbf{x}) = (1-\alpha)\cdot p(\mathbf{x}) + \alpha \cdot p_{\text{bg}}(\mathbf{x}), \tag{3}$$

with $\alpha \in (0, 1)$. This implementation adds a certain percentage α of background noise to the target density, spreading it over all the domain of the background density, allowing to properly apply the methods rejection and importance sampling with $q_\phi(\mathbf{x})$ as the proposal function, covering exhaustively the phase-space. Experimentally we have found good compromise with $\alpha = 0.05$.

2.4. ENIS training scheme of the proposal function

The training procedure to obtain $q_\phi(\mathbf{x})$ from $p(\mathbf{x})$ following ENIS consists of two phases:

1. Warm-up phase:
   (i) Sample $\mathbf{x}_p \sim p_{\text{bg}}(\mathbf{x})$ and compute their weights $w_p(\mathbf{x}_p) = p(\mathbf{x}_p)/p_{\text{bg}}(\mathbf{x}_p)$.
   (ii) Sample background $\mathbf{x}_{\text{bg}} \sim p_{\text{bg}}(\mathbf{x})$ with associated weights $w_{\text{bg}}(\mathbf{x}_{\text{bg}}) = C_{w_{\text{bg}}} \cdot p_{\text{bg}}(\mathbf{x}_{\text{bg}})$, where $C_{w_{\text{bg}}} = \frac{\alpha}{1-\alpha}\frac{\langle w_p(\mathbf{x}_p)\rangle}{\langle p_{\text{bg}}(\mathbf{x}_{\text{bg}})\rangle}$.
   (iii) Optimize the parameters of $q_\phi(\mathbf{x})$ via Eq. (2) using $\mathbf{x} = \{\mathbf{x}_p, \mathbf{x}_{\text{bg}}\}$ with weights $w(\mathbf{x}) = \{w_p(\mathbf{x}_p), w_{\text{bg}}(\mathbf{x}_{\text{bg}})\}$.

2. Iterative phase:

   (i) Sample $\mathbf{x}_q \sim q_\phi(\mathbf{x})$ and compute their weights $w_q(\mathbf{x}_q) = p(\mathbf{x}_q)/q_\phi(\mathbf{x}_q)$.

   (ii) Sample background $\mathbf{x}_{\text{bg}} \sim p_{\text{bg}}(\mathbf{x})$ with associated weights $w_{\text{bg}}(\mathbf{x}_{\text{bg}}) = C'_{w_{\text{bg}}} p_{\text{bg}}(\mathbf{x}_{\text{bg}})$, where $C'_{w_{\text{bg}}} = \frac{\alpha}{1-\alpha}\frac{\langle w_q(\mathbf{x}_q)\rangle}{\langle p_{\text{bg}}(\mathbf{x}_{\text{bg}})\rangle}$.

   (iii) Optimize the parameters of $q_\phi(\mathbf{x})$ via Eq. (2) using $\mathbf{x} = \{\mathbf{x}_q, \mathbf{x}_{\text{bg}}\}$ with weights $w(\mathbf{x}) = \{w_q(\mathbf{x}_q), w_{\text{bg}}(\mathbf{x}_{\text{bg}})\}$.

Fig. 1 depicts a flow diagram of the training method for ENIS, showing on the left block the warm-up phase, while on the right block the iterative phase.

Steps 1. (ii) and 2. (ii) allow the method to add background following Eq. (3) to construct $p_{\text{target}}(\mathbf{x})$ even if $p(\mathbf{x})$ is not normalized.

2.5. Measuring the performance of the proposal function

Consider n samples $\{\mathbf{x}_i\}_{i=1}^n$ generated with the proposal function $\mathbf{x} \sim q(\mathbf{x})$, with weights $w(\mathbf{x}_i) = p(\mathbf{x})/q(\mathbf{x})$, satisfying that k_{max}, the smallest constant $k > 0$ such that the inequality of Eq. (1) holds, is equal to $(\max w(\mathbf{x}_i))^{-1}$.

In real conditions, the parameter k can be relaxed, being taken as the inverse of the Q-quantile of these weights, w_{Q}, denoted by $k_{\text{Q}} = (Q\text{-quantile}(w))^{-1} = w_{\text{Q}}^{-1}$. This is equivalent of clipping the weights' maximum value to the Q-quantile of w, defining weights $w'(\mathbf{x})$. The ratio of volume with respect to the original density $p(\mathbf{x})$ we are maintaining by clipping the weights this way defines the coverage we have of the rejection sampling,

$$\text{Coverage} = \frac{\sum_{i=1}^N w'(\mathbf{x}_i)}{\sum_{i=1}^N w(\mathbf{x}_i)}. \tag{4}$$

This allows us to quantify the error we are committing when choosing a quantile over the maximum of weights when defining a constant k for rejection sampling.

The idea behind relaxing this constant k is that we will approximate wrongly only a small region of $p(\mathbf{x})$ with $q(\mathbf{x})$. In that small region, the ratio $p(\mathbf{x})/q(\mathbf{x})$ is large compared to the rest of the domain but still it is occupying a small volume of the density $p(\mathbf{x})$. This region can be ignored by relaxing k, making the overall ratio of $p(\mathbf{x})/(k \cdot q(\mathbf{x}))$ closer to 1 and improving drastically the rejection sampling at the cost of this small discrepancy which we are committing, quantified in Eq. (4).

3. Monte Carlo generation of the CCQE antineutrino cross section performance

In this Section, we will focus on analyzing the performance of the proposal density obtained by the NSF, depicted in Figs. 3 and 4. Additionally we also compare it to a uniform proposal density, $p_{\text{Unif}}(\mathbf{x})$, defined over the limits of the phase-space. We start by generating ten million samples from each proposal density and compute their associated weights. The proportion of samples with weight equal to zero is 5.56 % for the NSF proposal, compared to the 98.03 % for the uniform one. To understand the distribution of such weights, Fig. 2 shows the logarithmic scale of them (for the weights > 0), assuming the average of the weights is equal to 1. For the NSF $q_\phi(\mathbf{x})$ (left), all weights are concentrated around $\log_{10} w = 0$ with a small dispersion around it. Notice that there are only three weights in ten million barely greater than hundred. This shape justifies using not the maximum value of w to perform rejection sampling, but some quantile of it, as we will discuss below. Contrary, for the uniform distribution $p_{\text{Unif}}(\mathbf{x})$ (right), we can see that the spectrum of weights goes over 9 orders of magnitude. The mean for $\log_{10} w_{q_\phi}$ is 0.023 ± 0.040, while for $\log_{10} w_{p_{\text{Unif}}}$ we obtain an average of 0.85 ± 0.88, indicating a huge fluctuation in the magnitude of the weights.

The results of the performance test for rejection sampling are summarized in Tab. 1, where we compare two quantities for the NSF $q_\phi(\mathbf{x})$ and uniform $p_{\text{Unif}}(\mathbf{x})$ proposal functions. For this, different quantiles for the constant k for the rejection method are used, relaxing its restriction as discussed in Sec. 2.5. The quantiles for k were chosen using the ten million weights computed for the previous discussion of the weight magnitudes. The values in Tab. 1, for each quantile value and a proposal function, are the following:

- p_{accept}: probability of accepting a single event, given by the average of $p(\mathbf{x})/(k \cdot q_\phi(\mathbf{x}))$. If $p(\mathbf{x})/(k \cdot q_\phi(\mathbf{x})) > 1$, it is taken as 1 for the computation.
- Coverage: volume of the original density covering when taking k with a certain quantile, following Eq. (4).

The probability of acceptance, p_{accept}, for the NSF is at least 1 order of magnitude higher than the one obtained from uniform sampling. Additionally, NSF grows rapidly toward $\sim$ 70 % acceptance while also covering $>$ 99 % of the original density volume, as shown in the Coverage column. This is not the case for the uniform distribution, which, while being only 1 order of magnitude behind NSF with regards to acceptance, is missing a large volume of coverage of the original density.

Regarding the coverage, for all the chosen quantiles, the NSF drops a volume $<$ 1 %, while for the uniform distri-

bution the loss is of > 5 % for quantile 0.9999, > 38 % for 0.999, and > 91 % for 0.99, which is unacceptable when trying to produce samples from the original distribution. For the NSF this level of performance when taking the above quantiles is expected, as in Fig. 2 we have seen that the upper tail of weights with large magnitudes is barely a percentage over the whole distribution. However, for the uniform distribution, the loss of coverage is caused by two facts: i) 98.03 % of the weights are zero, hence placing the whole distribution on a 1.97 % of the weights. (ii) These weights, as seen in Fig. 2, span over many orders of magnitude, making a cut on the quantile of their distribution more noticeable, as will be discussed below.

To visualize the coverage and the regions missing by choosing a quantile $k_{\text{Q}0.999}$ and different proposal functions, Fig. 5 shows 2-dimensional histogram representation of the marginalized coverage bin-to-bin, taking the variables in pairs, where each bin quantifies the coverage of that bin (i.e., the sum of weights in that bin after choosing a certain quantile over the sum of weights of those weights without clipping). On the left, the coverage for the NSF is presented and shows that only few regions of the phase-space have values smaller than 1, and even in those regions the coverage has no noticeable discrepancies. On the right, the coverage of the uniform proposal is shown for the same quantile 0.999, marking clear regions where the coverage drops drastically to values close to zero.

When comparing both coverage regions in Fig. 5, a clear pattern can be seen for the uniform one, while it looks quite random for the NSF. This is because the coverage is related to the ratio $p(\mathbf{x})/q(\mathbf{x})$, with $q(\mathbf{x})$ the corresponding proposal density. For the NSF, $q_\phi(\mathbf{x})$ has a shape very closely related to $p(\mathbf{x})$, as shown in Fig. 3 and Fig. 4, so the coverage would correspond to regions where the discrepancy is large, which has a chaotic behavior. Contrary, for the uniform proposal, $p_{\text{Unif}}(\mathbf{x})$, this ratio is proportional to $p(\mathbf{x})$; hence, by clipping, we are doing so according to that particular shape, making the coverage less chaotic and more structured. This translates into making highly probable areas equally likely than others with less probability, affecting this exact group of regions as we will now analyze.

Fig. 5 gives us an overall picture of where the densities are wrongly estimated by choosing certain quantile, but it does not quantify or indicate the amount of error, that is, it is not telling us whether the coverage is poor in areas of small or high density. To answer this question, a multidimensional histogram overall four dimensions was performed, with a binning of 20 in each dimension. Then, for each bin, we compute the percentage of weight for a proposal q,

$$\%\ w_q \text{ of bin } = \sum_{\mathbf{x}\in\text{bin}} w_q(\mathbf{x}) / \sum_{\mathbf{x}} w_q(\mathbf{x}),$$

which is equivalent to the percentage of density $p(\mathbf{x})$ in that bin, and the coverage for the quantile $k_{\text{Q}0.999}$. Fig. 6 shows a histogram of the number of bins according to their % w_q of bin vs their coverage. Notice how % w_q is taken on the logarithmic scale. For the NSF (Fig. 6 left), the regions of coverage visibly smaller than one are 2 to 4 orders of magnitude smaller in % w_q than the denser high % w_q region on the top right. This means that the areas being misrepresented by taking the quantile $k_{\text{Q}0.999}$ are relatively small. Also, most of the area with coverage < 1 are close to full coverage. Contrary, the uniform proposal (Fig. 6 right) shows the coverage dropping for high values of % w_q, indicating that important regions of the original density are being trimmed down by choosing $k_{\text{Q}0.999}$. This observation is in agreement with the total coverage we are seeing in Tab. 1.

To summarize the analysis performed on the results of Tab. 1, in the case of the NSF, by lowering the quantile, the probability of acceptance grows until reaching almost 80 %, while maintaining coverage of over 0.99 %. These scores allow us to justify using a smaller quantile for rejection sampling to improve significantly the performance, while also quantifying the misrepresentation we are doing by lowering the constant k. Contrary, for the uniform proposal, the analysis showed the weakness when trying to utilize smaller quantiles, lowering the coverage by over 38 % when using a quantile of only 0.999.

4. Conclusion

In this paper we have presented Exhaustive Neural Importance Sampling (ENIS), a framework to find accurate proposal density functions in the form of normalizing flows. This proposal density is subsequently used to perform rejection sampling on a target density to obtain MC data sets or compute expected values. We argue that ENIS solves the main issues associated with rejection sampling algorithms as described in the introduction, finding a proposal automatically while reducing the inefficiency of the method.

As of usage, ENIS brings the possibility of applying the same normalizing flow for rejection sampling of similar densities, e.g., densities coming from a model where the parameters are changed slightly, altering the overall density smoothly. The weight distribution of the ratio between target and proposal will be altered, but no additional training of the neural network would be needed regarding the theoretical model remains similar to the original one. This is a huge advantage compared to methods like MCMC, where one would have to rerun the complete algorithm to obtain samples from each of the different densities.

Acknowledgements

The authors have received funding from the Spanish Ministerio de Economía y Competitividad (SEIDI-MINECO) under Grants No. FPA2016-77347-C2-2-P and SEV-2016-0588, from the Swiss National Foundation Grant No. 200021_85012 and acknowledge the support of the Industrial Doctorates Plan of the Secretariat of Universities and Research of the Department of Business and Knowledge of the Generalitat of Catalonia. The authors also are indebted to the Servei d'Estadística Aplicada from the Universitat Autònoma de Barcelona for their continuous support. The authors would also like to thank C. Durkan et al. for the code provided in their paper (Durkan et al., 2019a) which served as a basis to build ENIS, and Tobias Golling for the valuable discussion he brought to the table.

References

Bothmann, E., Janßen, T., Knobbe, M., Schmale, T., and Schumann, S. Exploring phase space with Neural Importance Sampling. *SciPost Phys.*, 8(4):069, 2020. doi: 10.21468/SciPostPhys.8.4.069.

Casella, G., Robert, C. P., and Wells, M. T. *Generalized Accept-Reject sampling schemes*, volume Volume 45 of *Lecture Notes–Monograph Series*, pp. 342–347. Institute of Mathematical Statistics, Beachwood, Ohio, USA, 2004. doi: 10.1214/lnms/1196285403. URL https://doi.org/10.1214/lnms/1196285403.

Durkan, C., Bekasov, A., Murray, I., and Papamakarios, G. Cubic-spline flows. *ArXiv*, abs/1906.02145, 2019a.

Durkan, C., Bekasov, A., Murray, I., and Papamakarios, G. Neural spline flows. In *NeurIPS*, 2019b.

Gao, C., Höche, S., Isaacson, J., Krause, C., and Schulz, H. Event generation with normalizing flows. *Phys. Rev. D*, 101:076002, Apr 2020. doi: 10.1103/PhysRevD.101.076002. URL https://link.aps.org/doi/10.1103/PhysRevD.101.076002.

Kahn, H. and Marshall, A. W. Methods of reducing sample size in monte carlo computations. *Journal of the Operations Research Society of America*, 1(5):263–278, 1953. ISSN 00963984. URL http://www.jstor.org/stable/166789.

Kullback, S. and Leibler, R. A. On information and sufficiency. *Ann. Math. Statist.*, 22(1):79–86, 03 1951. doi: 10.1214/aoms/1177729694. URL https://doi.org/10.1214/aoms/1177729694.

Müller, T., McWilliams, B., Rousselle, F., Gross, M., and Novák, J. Neural importance sampling. *ACM Trans. Graph.*, 38:145:1–145:19, 2018.

Papamakarios, G., Nalisnick, E. T., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. Normalizing flows for probabilistic modeling and inference. *ArXiv*, abs/1912.02762, 2019.

Robert, C. P. and Casella, G. *Monte Carlo Statistical Methods*. Springer New York, 2004. doi: 10.1007/978-1-4757-4145-2. URL https://doi.org/10.1007/978-1-4757-4145-2.

A. Figures and Tables

In the next pages the Figures and Tables of the paper can be found.

Table 1. Performance values for different quantile choices of k for rejection sampling, as discussed in Sec. 2.5, comparing both NSF $q_\phi(\mathbf{x})$ and uniform $p_{\text{Unif}}(\mathbf{x})$ proposal functions. For this exercise, one million samples were generated. The quantities are the probability of accepting a single sample p_{accept} and the coverage of the target density for that particular quantile [Eq. (4)].

Quantile	Prop.	p_{accept}	Coverage
1.00000	NSF	0.0051	1.0000
	Unif.	0.0002	1.0000
0.99999	NSF	0.0633	0.9999
	Unif.	0.0004	0.9947
0.99990	NSF	0.1623	0.9996
	Unif.	0.0008	0.9480
0.99900	NSF	0.3590	0.9984
	Unif.	0.0027	0.6185
0.99000	NSF	0.7187	0.9939
	Unif.	0.0137	0.0818
0.98500	NSF	0.7730	0.9927
	Unif.	0.0171	0.0325
0.98100	NSF	0.7968	0.9920
	Unif.	0.0193	0.0039

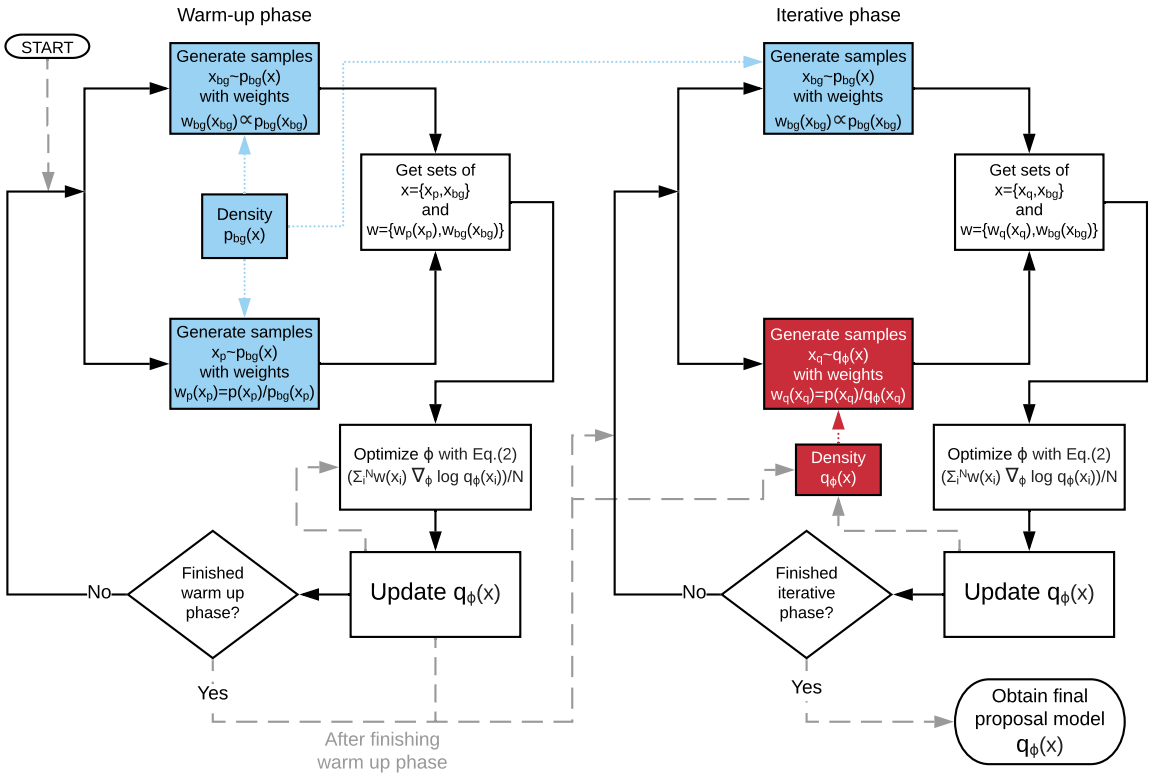

Figure 1. Exhaustive Neural Importance Sampling flow diagram. The warm-up phase is depicted on the left block and the iterative phase on the right block. See text for a detail description.

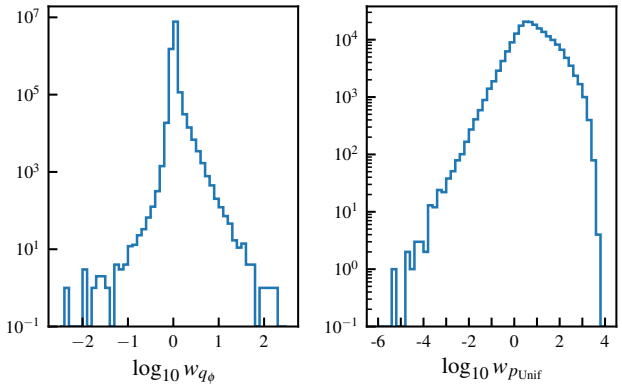

Figure 2. Logarithmic weight distribution for ten million samples from the NSF proposal $q_\phi(\mathbf{x})$ (left) vs the same number of samples from the uniform proposal $p_{\text{Unif}}(\mathbf{x})$ (right). Notice that we are only computing the logarithm for weights > 0. For the NSF, all weights are concentrated around $\log_{10} w = 0$ with a small dispersion around it, while for the uniform distribution the spectrum of weights goes over 9 orders of magnitude.

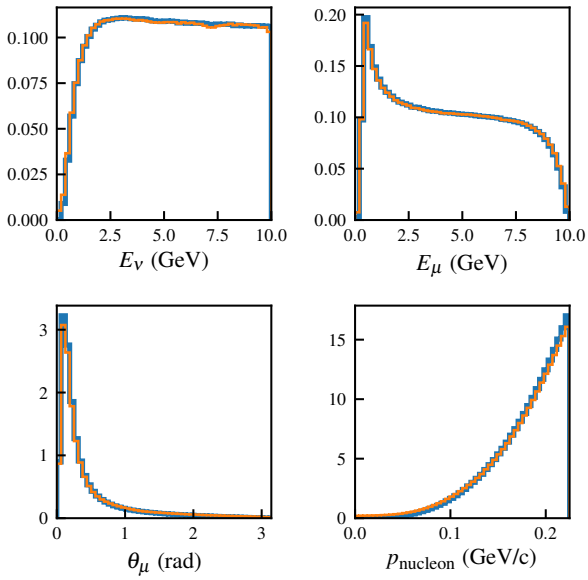

Figure 3. $p(\mathbf{x})$ (blue) vs $q_\phi(\mathbf{x})$ (orange) 1-dimensional normalized histograms of the marginalized CCQE cross section density for each of the variables. The plots show light discrepancy in each variable, but an overall agreement between the NSF proposal $q_\phi(\mathbf{x})$ and the CCQE cross section density $p(\mathbf{x})$.

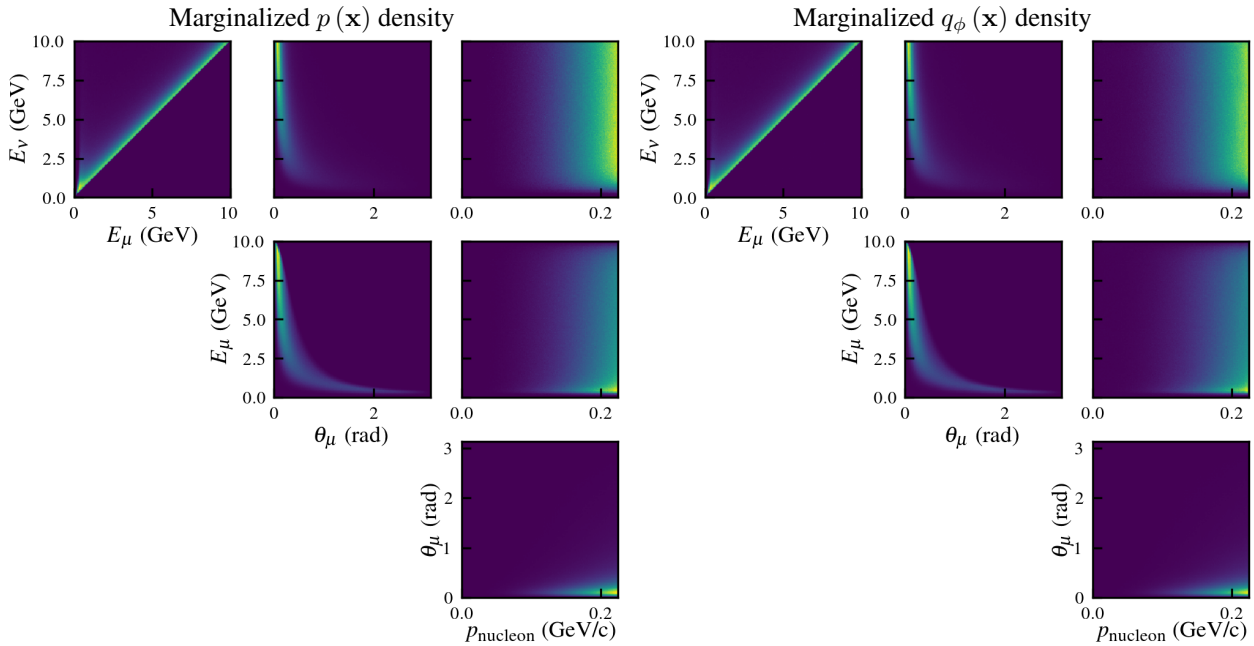

Figure 4. 2D histograms comparison of cross section density for the real cross section $p(\mathbf{x})$ (left) and the proposal density $q_\phi(\mathbf{x})$(right). Visually, an overall agreement can be seen.

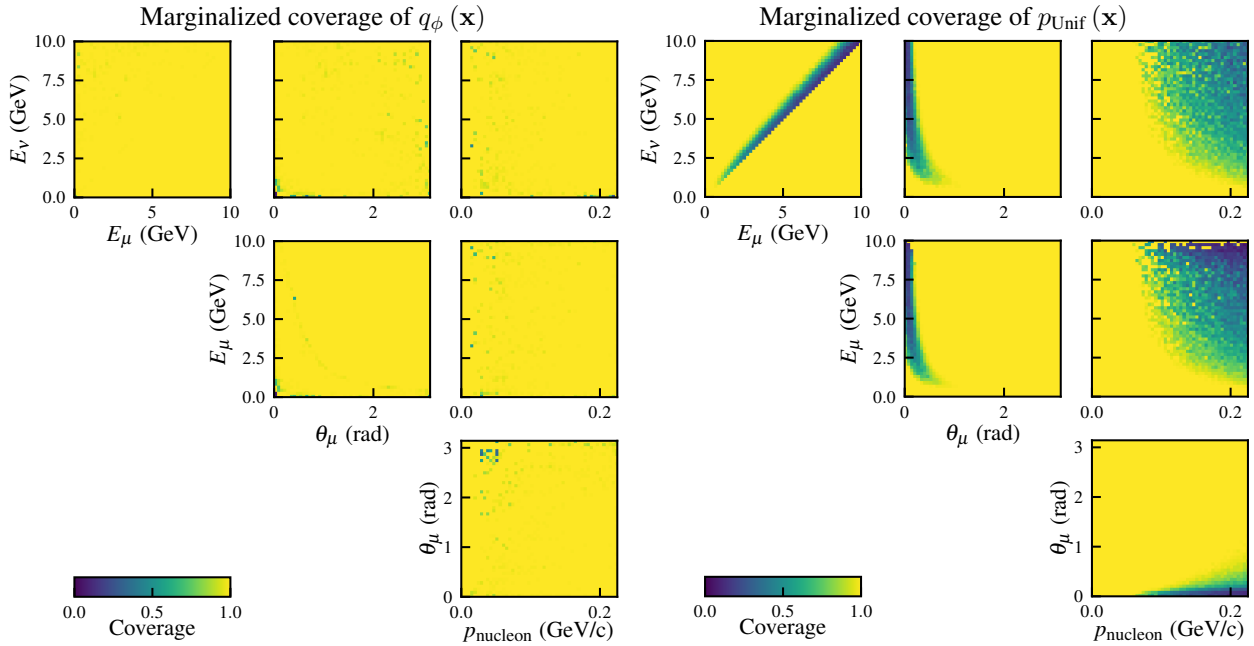

Figure 5. 2D histogram representation of the marginalized coverage bin-to-bin for NSF proposal (left) and for the uniform proposal (right), for k-quantile$= 0.999$. The coverage of the NSF presents barely discrepancies in small areas, justifying the use of a quantile for k to improve acceptance and time, as shown in Tab. 1. For the uniform proposal, the coverage presents an important size of the total area with significant low coverage, which is unacceptable when trying to perform rejection sampling from it.

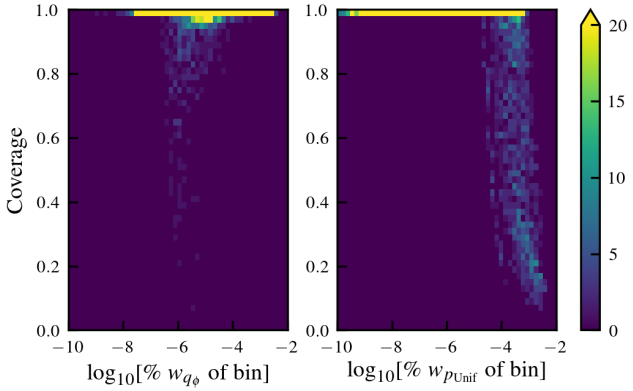

Figure 6. Representation of the number of bins according to their $\%w_q$ of bin vs their coverage for four-dimensional bins in $p\left(\mathbf{x}\right)$, taking $k_{\text{Q0.999}}$. For the NSF (left), the coverage is barely lower than one in most of the bins, and when it drops it is for low $\%w_q$. Contrary, for the uniform proposal (right), the coverage drops for high $\%w_q$, making the overall coverage way smaller than the one for NSF, as shown in Tab. 1.