

# Machine learning approach to QCD kinetic theory

Sergio Barrera Cabodevila<sup>1,\*</sup>, Aleksi Kurkela<sup>2</sup>, and Florian Lindenbauer<sup>3,4</sup>

<sup>1</sup>Instituto Galego de Física de Altas Enerxías IGFAE, Universidade de Santiago de Compostela, E-15782 Galicia-Spain

<sup>2</sup>Faculty of Science and Technology, University of Stavanger, 4036 Stavanger, Norway

<sup>3</sup>Institute for Theoretical Physics, TU Wien, Wiedner Hauptstrasse 8-10, 1040 Vienna, Austria

<sup>4</sup>MIT Center for Theoretical Physics - a Leinweber Institute, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

**Abstract.** The effective kinetic theory (EKT) of QCD provides a possible picture of various non-equilibrium processes in heavy- and light-ion collisions. While there have been substantial advances in simulating the EKT in simple systems with enhanced symmetry, eventually, event-by-event simulations will be required for a comprehensive phenomenological modeling. As of now, these simulations are prohibitively expensive due to the numerical complexity of the Monte Carlo evaluation of the collision kernels. In this talk, we show how the evaluation of the collision kernels can be performed using neural networks paving the way to full event-by-event simulations.

## 1 Introduction

Heavy-ion collisions at experiments at the LHC have been shown to produce one of the most extreme states of matter: the Quark-Gluon Plasma. This extremely short-lived fluid is believed to thermalize very quickly. A state-of-the-art description of this thermalization process is given by the so-called Effective Kinetic Theory (EKT), which captures the leading-order QCD scattering and splitting/merging processes [1].

In its most general form, the microscopic behaviour of the system is governed by the leading order QCD Boltzmann equation,

$$(\partial_t + \mathbf{v} \cdot \nabla_{\mathbf{x}}) f(\mathbf{x}; \mathbf{p}; \mathbf{t}) = C_{1 \leftrightarrow 2}[f] + C_{2 \leftrightarrow 2}[f]. \quad (1)$$

In this talk, we focus on the evolution of a pure gluon system; therefore,  $f$  is the gluonic distribution function. Eq. (1) describes the time evolution of an out-of-equilibrium system, where  $1 \leftrightarrow 2$  and  $2 \leftrightarrow 2$  processes are the underlying interactions of the partons that form the plasma.

Solving Eq. (1) involves a numerical implementation in 3+3+1D, which is impractical from the computational point of view due to the large time required to compute the collision kernels. For this reason, different approximations have been used to study the thermalization of the Quark-Gluon Plasma. For instance, a longitudinal boost-invariant system with

\*e-mail: [sergio.barrera.cabodevila@usc.es](mailto:sergio.barrera.cabodevila@usc.es)

infinite transverse size [2, 3], the diffusion approximation of the  $2 \leftrightarrow 2$  kernel [4], or Relaxation Time Approximation (RTA) [5] are some examples that have been used to perform phenomenological studies of the hydrodynamization process.

We propose a novel approach based on Artificial Neural Networks (ANNs) to overcome the bottleneck of computing the collision kernels in less symmetric systems. Our approach exploits the fact that the QCD Boltzmann equation is local in space and, therefore, obtaining the collision kernels in each spatial cell requires doing a very similar calculation in each of them. Thus, this problem is suitable to be dealt with by training a neural network that fits the collision kernel for a given distribution function at every given spatial point. The locality will allow the application of the same ANN to each spatial cell. A more detailed explanation of the method we present here can be found in our recent paper [6].

## 2 Training data

The first thing we need to take care of is which data we use to train our neural network. Since we want to create a map of a distribution function to its corresponding collision kernel, it is clear that both of them must be the input and output of the ANN, respectively. The calculation of the collision kernels is done by the well-known Monte Carlo solver of the EKT. We should restrict ourselves to physically sensible distributions, relevant for the thermalization process. Thus, we generate pairs of data with distribution functions corresponding to the kinetic evolution starting from initial conditions inspired by the CGC framework [3] and their respective collision kernels. Additionally, we will also include perturbations over the equilibrium distribution, which helps the network to approximate the collision kernels around thermal equilibrium.

To reduce the size of the data needed, we also exploit symmetries preserved by the Boltzmann equation. First, we take advantage of the conformal symmetry to fix the energy density of the distribution functions used in the training dataset. If we want to input a distribution with a different energy density, we can apply the corresponding conformal transformation such that the distribution has the required energy density. Then, after calculating the collision kernel, we perform the inverse transformation. Similarly, in the case of a 3D distribution function in momentum space, we establish a hierarchy for the anisotropies, such that the pressures are ordered  $P_z > P_y > P_x$ . Then, as before, if this is not true for the input distribution function, we perform a rotation before passing it to the ANN and rotate back the output.

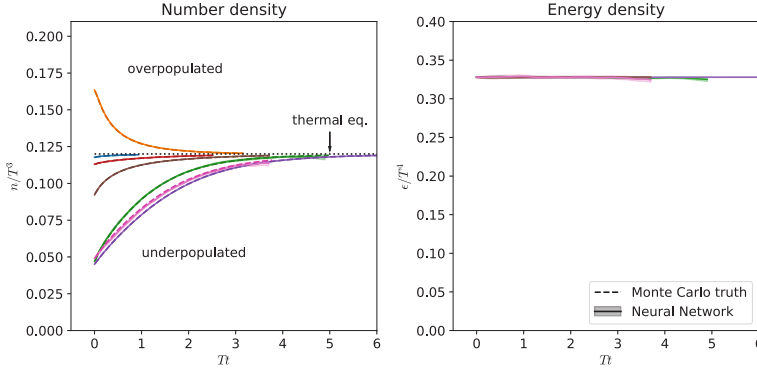
This dataset needs to be preprocessed to improve the convergence of the subsequent training. Here, we enumerate the transformations we apply to the dataset:

- Instead of using  $f$  and  $C$ , we use the energy distributions,  $p^3 f$  and  $p^3 C$ . This helps the network to have a good energy density conservation, which is a feature implemented in the Monte Carlo solver.
- We subtract the thermal equilibrium from the input since we observe that then the network reproduces better the thermal fixed point. Then, the mapping we fit is  $p^3 f - p^3 f_{eq} \rightarrow p^3 C$ .
- We standardize the training data, that is, we normalize the data such that each feature has a standard deviation equal to unity.

In total, we generate of the order of 100000 training pairs for both 1D system ( $\sim 100$  MB) and 3D system ( $\sim 50$  GB).

## 3 Choosing the ANN architecture

The choice of the architecture of the neural network is, in principle, arbitrary. In our case, we restrict to neural networks with linear hidden layers and an arbitrary number of nodes per



**Figure 1.** Number (left) and energy (right) densities time evolutions for different initial conditions in the isotropic/1D case. Solid lines with error bars correspond to the neural network predictions, while the dashed lines are computed with the Monte Carlo algorithm. Figure obtained from [6].

layer that use ReLU as activation function. To choose the optimal number of both internal layers and nodes, as well as the learning rate, we use RayTune [7]. This tool trains several networks in parallel and compares their performance to identify the better-performing ones.

We perform this procedure for  $C_{1\leftrightarrow 2}$  and  $C_{2\leftrightarrow 2}$  independently, such that we have two different networks to compute the collision kernels separately. Besides, we keep the 10 best-performing networks and not just the best one, so we can compare their output when presenting the results.

## 4 Results

The results we show here are obtained with the ten best fitted neural networks for each collision kernel. To compute the time evolution of a given initial distribution function, we produce ten independent evolutions with a fourth-order Runge-Kutta algorithm, one for each of the networks. Then, we show the mean value of the ten independent evolutions and estimate the error bands with the Jackknife method,

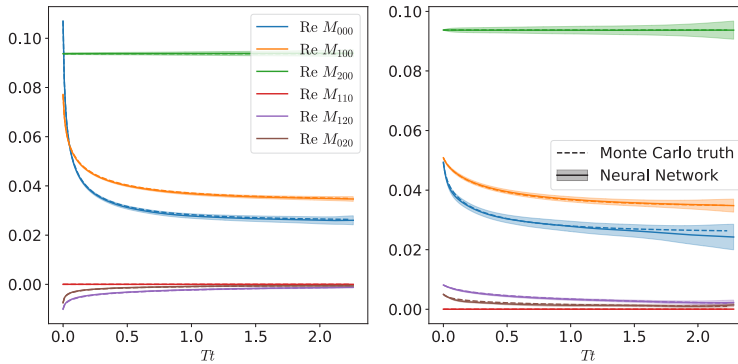
$$\delta f(t_n) = \sqrt{\frac{M-1}{M} \sum_m (f_m(t_n) - \langle f(t_n) \rangle)^2}. \quad (2)$$

The trained neural networks, as well as the training dataset used for the 1D case, are publicly available in [8]. In the following, let us briefly comment on the results we have obtained.

First, let us focus on the results for the 1D scenario. In Fig. 1 we display the number and energy density for different evolutions. In all cases, the energy density is nicely conserved, and the number density approaches its thermal equilibrium value following the same trend as the Monte Carlo calculation. It is relevant to mention that when the system is close to equilibrium, the error bars start to grow and the evolution is not fully stable.

Regarding the 3D scenario, we show moments of the distribution, defined as

$$M_{nlm} = \frac{1}{T^{n+2}} \int \frac{d^3 \mathbf{p}}{(2\pi)^3} p^{n-1} Y_l^{m*}(\theta, \phi) f(\mathbf{p}), \quad (3)$$



**Figure 2.** Different moments of the distribution defined in Eq. (3) as a function of time for two different initial conditions. Figure obtained from [6].

in Fig. 2. In this expression,  $Y_l^m(\theta, \phi)$  are the spherical harmonics. As in the previous case, the curves follow the same trend as the results obtained in the Monte Carlo approach. The energy density, given by the  $M_{200}$  moment, is conserved, and at later times, the error bars grow, indicating that the network has challenges in capturing the thermal fixed point.

A detailed benchmark comparison between the Monte Carlo and the novel neural network approaches is subtle and depends a lot on the chosen grid and the desired accuracy. However, as an estimate, we observe a systematic speed-up of roughly three orders of magnitude in the calculation of the full evolution, making it an extremely attractive method despite its challenges near equilibrium.

## Acknowledgements

FL is a recipient of a DOC Fellowship of the Austrian Academy of Sciences at TU Wien (project 27203). This work is funded in part by the Austrian Science Fund (FWF) under Grant DOI 10.55776/P34455, and Grant DOI 10.55776/J4902. For the purpose of open access, the authors have applied a CC BY public copyright license to any Author Accepted Manuscript (AAM) version arising from this submission. The results in this paper have been achieved in part using the Austrian Scientific Computing (ASC) infrastructure, project 71444. SBC is supported by the European Research Council project ERC-2018-ADG-835105 YoctoLHC; by María de Maeztu grant CEX2023-001318-M and by project PID2023-152762NB-I00, both funded by MCIN/AEI/10.13039/501100011033; from the Xunta de Galicia (CIGUS Network of Research Centres) and the European Union.

## References

- [1] P.B. Arnold, G.D. Moore, L.G. Yaffe, Effective kinetic theory for high temperature gauge theories, *JHEP* **01**, 030 (2003), [hep-ph/0209353](https://arxiv.org/abs/hep-ph/0209353). [10.1088/1126-6708/2003/01/030](https://doi.org/10.1088/1126-6708/2003/01/030)
- [2] J.D. Bjorken, Highly relativistic nucleus-nucleus collisions: The central rapidity region, *Phys. Rev. D* **27**, 140 (1983). [10.1103/PhysRevD.27.140](https://doi.org/10.1103/PhysRevD.27.140)
- [3] A. Kurkela, Y. Zhu, Isotropization and hydrodynamization in weakly coupled heavy-ion collisions, *Phys. Rev. Lett.* **115**, 182301 (2015), [1506.06647](https://arxiv.org/abs/1506.06647). [10.1103/PhysRevLett.115.182301](https://doi.org/10.1103/PhysRevLett.115.182301)

- [4] A.H. Mueller, The Boltzmann equation for gluons at early times after a heavy ion collision, *Phys. Lett. B* **475**, 220 (2000), [hep-ph/9909388](https://arxiv.org/abs/hep-ph/9909388). [10.1016/S0370-2693\(00\)00084-8](https://doi.org/10.1016/S0370-2693(00)00084-8)
- [5] A. Kurkela, S.F. Taghavi, U.A. Wiedemann, B. Wu, Hydrodynamization in systems with detailed transverse profiles, *Physics Letters B* **811**, 135901 (2020). [10.1016/j.physletb.2020.135901](https://doi.org/10.1016/j.physletb.2020.135901)
- [6] S. Barrera Cabodevila, A. Kurkela, F. Lindenbauer, Solving the QCD effective kinetic theory with neural networks (2025), [2506.19632](https://arxiv.org/abs/2506.19632).
- [7] R. Liaw, E. Liang, R. Nishihara, P. Moritz, J.E. Gonzalez, I. Stoica, Tune: A research platform for distributed model selection and training, *arXiv preprint arXiv:1807.05118* (2018).
- [8] S. Barrera Cabodevila, A. Kurkela, F. Lindenbauer, EKT-NN (2025), <https://doi.org/10.5281/zenodo.15701887>, <https://doi.org/10.5281/zenodo.15701887>