

This is a self-archived version of an original article. This version may differ from the original in pagination and typographic details.

Author(s): Hirvonen, Henry; Eskola, Kari J.; Niemi, Harri

Title: Deep learning for flow observables in high energy heavy-ion collisions

Year: 2024

Version: Published version

Copyright: © 2024 the Authors

Rights: CC BY 4.0

Rights url: <https://creativecommons.org/licenses/by/4.0/>

Please cite the original version:

Hirvonen, H., Eskola, K. J., & Niemi, H. (2024). Deep learning for flow observables in high energy heavy-ion collisions. In R. Bellwied, F. Geurts, R. Rapp, C. Ratti, A. Timmins, & I. Vitev (Eds.), 30th International Conference on Ultra-Relativistic Nucleus-Nucleus Collisions (Quark Matter 2023) (Article 02002). EDP Sciences. EPJ Web of Conferences, 296.
<https://doi.org/10.1051/epjconf/202429602002>

Deep learning for flow observables in high energy heavy-ion collisions

Henry Hirvonen^{1,2,*}, Kari J. Eskola^{1,2,**}, and Harri Niemi^{1,2,***}

¹University of Jyväskylä, Department of Physics, P.O.B. 35, FI-40014 University of Jyväskylä, Finland

²Helsinki Institute of Physics, P.O.B. 64, FI-00014 University of Helsinki, Finland

Abstract. We demonstrate how deep convolutional neural networks can be trained to predict 2+1 D hydrodynamic simulation results for flow coefficients, mean- p_T and charged particle multiplicity from the initial energy density profile. We show that this method provides results that are accurate enough, so that one can use neural networks to reliably estimate multi-particle flow correlators. Additionally, we train networks that can take any model parameter as an additional input and demonstrate with a few examples that the accuracy remains good. The usage of neural networks can reduce the computation time needed in performing Bayesian analyses with multi-particle flow correlators by many orders of magnitude.

1 Introduction

Neural networks have proven to be an effective tool for a variety of applications in heavy-ion physics. These range from performing pre-processing or selection of large data flows in experiments to emulating computationally expensive simulations [1–4]. The rising popularity of neural networks is driven by their accuracy and fast inference speed when dealing with complex multi-dimensional data. These aspects can be crucial when performing real-time data selection or heavy numerical simulations that need to be repeated a large number of times.

The reduced computation time is especially needed when trying to extract the matter properties of the quark-gluon plasma (QGP) from the experimental data through hydrodynamic simulations using a Bayesian analysis. This is due to the fact that one Bayesian analysis will need $\sim 10^6 - 10^9$ simulated collision events depending on which measured observables one includes in the analysis. Performing this many hydrodynamic simulations will take $\sim 10^5 - 10^8$ CPU hours, which makes the inclusion of some multi-particle correlations impractical, even though they could provide additional information to constrain the QCD matter properties.

In principle, all the final state information in hydrodynamic simulation is encoded into the initial state and the matter properties of QGP. However, extracting the final state information directly from the initial state is a highly nontrivial task since relativistic hydrodynamics is a nonlinear theory. The convolutional networks are particularly good at detecting patterns in structured 2-dimensional data, like images, which is why they are excellent tools when trying to estimate the final state observables from an initial state event by event.

*Speaker, e-mail: hevivahi@jyu.fi

**e-mail: kari.eskola@jyu.fi

***e-mail: harri.m.niemi@jyu.fi

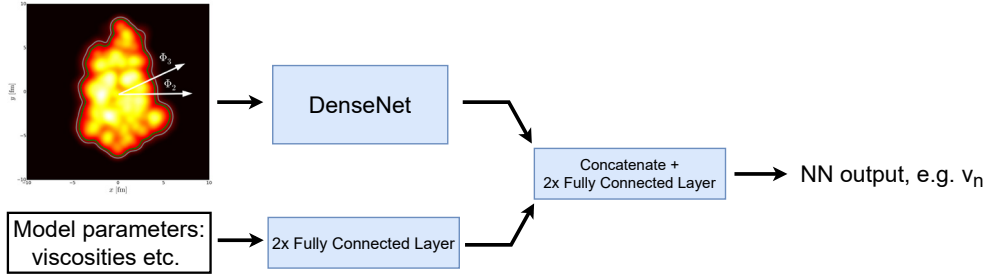


Figure 1. Schematic presentation of neural network structure with multiple inputs.

2 Neural network

The convolutional neural networks, one for each observable, are trained to produce p_T -integrated flow coefficients v_n , mean transverse momentum $[p_T]$ and charged particle multiplicities $dN_{ch}/d\eta$. This was originally done in Ref. [4], from where one can find a detailed description of the setup. Here we just go through the main points.

As the training data, we used 20 k EKRT (Eskola-Kajantie-Ruuskanen-Tuominen) model [5–8] initial energy density profiles in the transverse plane and the corresponding final state observables in these events at midrapidity. The final state observables are obtained from the 2+1 D hydrodynamic simulations done in Ref. [9]. The training events are distributed evenly between four different collision systems: 200 GeV Au+Au, 2.76 TeV Pb+Pb, 5.023 TeV Pb+Pb, and 5.44 TeV Xe+Xe. Even though we use one specific setup of an initial state model combined with a hydrodynamics code to produce the training data, the methods introduced here are expected to be applicable also with training data obtained from any other setup of a similar type.

The neural network architecture is the DenseNet architecture [10] with slight modifications that will make it suitable for regression tasks. For a more complete description of the architecture, see Ref. [4]. It took ~ 1 hour to train a network that can produce one observable. With a set of trained networks, one can generate 1 M events in ~ 20 hours.

2.1 Model parameters as an input

A neural network that can predict a final state observable from an initial state is already a lot faster than doing full hydrodynamic simulations, but it has a drawback: every time one wants to change QCD matter properties or model parameters that affect time evolution of the system, one would need to generate a new set of training data and retrain the networks. This issue can be solved by adding all the parameters of interest as additional input to the neural networks. Here we refer to this type of network with additional input parameters as NN_p . The architecture of NN_p is demonstrated in Fig. 1. The energy density input is treated the same way as without additional inputs and all the additional inputs are put through two fully connected layers and then combined with the output of the DenseNet layer structure. After this, we have included two fully connected layers from which we then obtain the final output. The training of NN_p was done using in total of 160 k training events distributed evenly between 4 collision systems and 2 k parameter points sampled from a Latin hypercube. This makes only 80 events of training data for one parameter point, which is 250 times more efficient than the training in the previous case.

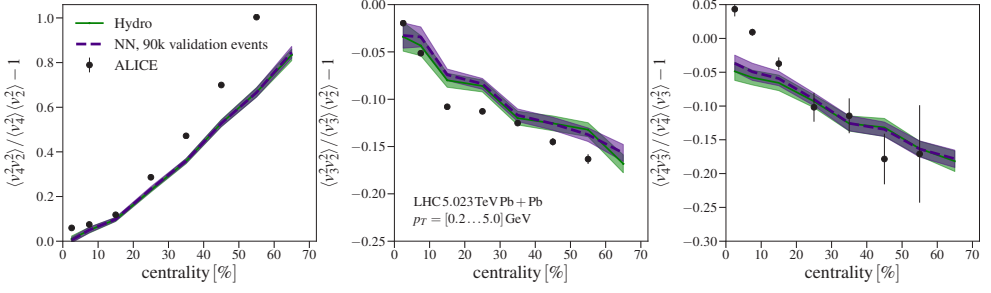


Figure 2. Validation test of the neural networks for $NSC(m,n)$ with 90 k validation events. The experimental data are from the ALICE Collaboration [11]. Figure from Ref. [4].

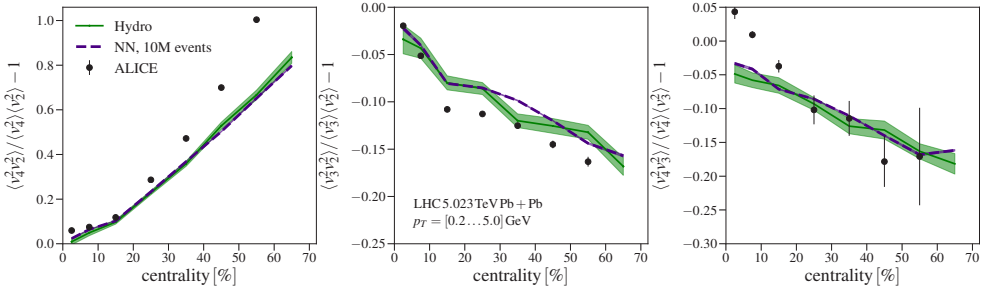


Figure 3. Neural network prediction for $NSC(m,n)$ with 10 M events. The experimental data are from the ALICE Collaboration [11]. Figure from Ref. [4].

3 Results and Conclusions

To test the accuracy of the neural networks that were trained with one set of model parameters, we generated 90 k independent EKRT initial energy density profiles and compared the results of hydrodynamic simulations against the neural network predictions. In Fig. 2 we show a comparison of these two for normalized symmetric cumulants $NSC(m,n)$ as a function of centrality. We can see that the neural network can reproduce the cumulants well, even though the size of the training data for one collision system was only 5 k events. In Fig. 3, we demonstrate how one can then generate 10 M events with the neural networks to see how these cumulants would look when statistical errors became insignificant. Here we can see quite noticeable deviations from the result that used 90 k events, especially for $NSC(2,3)$, for which the centrality dependence clearly changes, matching the shape of the ALICE measurements better. This illustrates the importance of the number of events used when trying to constrain the QCD matter properties with multi-particle flow correlations.

In the case of NN_p networks, we are mostly interested in the network accuracy for generating new events with the same parameter values as in the training data. This is because the most efficient way to do Bayesian analysis is to first generate a high number of events using neural networks in a set of parameter points, compute all of the observables in these parameter points, and then train Gaussian process emulators for these observables. Here the accuracy of NN_p networks was tested by taking two sets of model parameter points from the training data which correspond to drastically different values of viscosities, and then generating 20 k new independent initial state profiles for both points and doing a similar validation comparison between NN_p and hydrodynamic simulations as before. The results are shown in

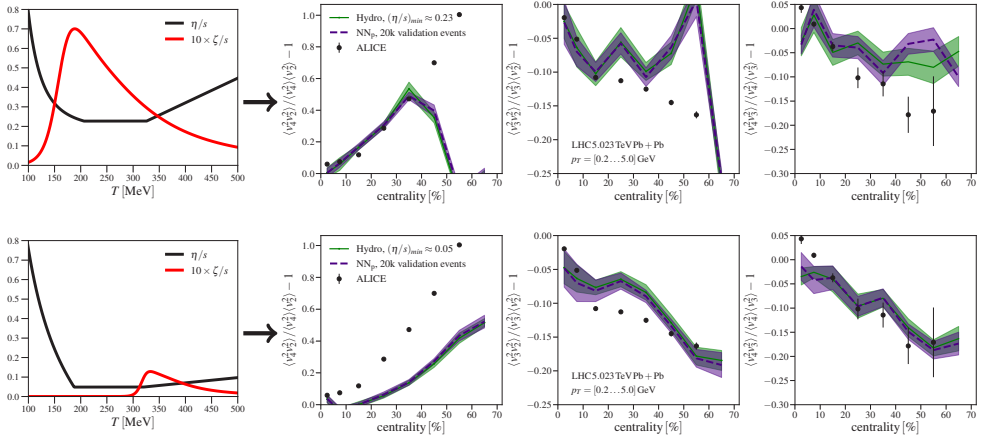


Figure 4. Validation tests of NN_p networks for $NSC(m, n)$ with 20 k validation events in each case. The upper (lower) panels show result with high (low) values of specific viscosities. The experimental data are from the ALICE Collaboration [11].

Fig. 4, from where one can see that NN_p can still very accurately reproduce the results from hydrodynamic simulations. The only exception is the peripheral region of $NSC(3, 4)$ in the extremely high viscosity case, where the numerical errors of hydrodynamical simulations of themselves might be very significant.

The goal of introducing neural networks in this work was to replace the slow hydrodynamic simulations and make it possible to add multi-particle flow correlators to Bayesian analysis. We have demonstrated that this is indeed possible and has the potential to cut the computational time needed for these analyses by many orders of magnitude.

We acknowledge the financial support from the Jenny and Antti Wihuri Foundation, and the Academy of Finland project 330448. This research was funded as a part of the Center of Excellence in Quark Matter of the Academy of Finland (project 346325), the European Research Council project ERC-2018-ADG-835105 YoctoLHC, and the European Union's Horizon 2020 research and innovation program under grant agreement No 824093 (STRONG-2020). The Finnish IT Center for Science (CSC) is acknowledged for the computing time through the Project jyy2580.

References

- [1] G. Aad *et al.* [ATLAS], *Eur. Phys. J. C* **79**, no.11, 970 (2019)
- [2] L. Liu, J. Velkovska, Y. Wu and M. Verweij, *JHEP* **04**, 140 (2023)
- [3] H. S. Wang, S. Guo, K. Zhou and G. L. Ma, [arXiv:2305.09937 [nucl-th]]
- [4] H. Hirvonen, K. J. Eskola and H. Niemi, *Phys. Rev. C* **108**, no.3, 034905 (2023)
- [5] K.J. Eskola, K. Kajantie, P.V. Ruuskanen, K. Tuominen, *Nucl. Phys. B* **570**, 379 (2000)
- [6] R. Paatelainen *et al.*, *Phys. Rev. C* **87**, no.4, 044904 (2013)
- [7] R. Paatelainen, K.J. Eskola, H. Niemi, K. Tuominen, *Phys. Lett. B* **731**, 126 (2014)
- [8] H. Niemi, K. J. Eskola and R. Paatelainen, *Phys. Rev. C* **93**, no.2, 024907 (2016)
- [9] H. Hirvonen, K. J. Eskola and H. Niemi, *Phys. Rev. C* **106**, no.4, 044913 (2022)
- [10] G. Huang, Z. Liu, L. van der Maaten, K. Q. Weinberger, *CVPR*, 4700-4708 (2017)
- [11] S. Acharya *et al.* [ALICE], *Phys. Lett. B* **818**, 136354 (2021)