

EVStabilityNet: predicting the stability of star clusters in general relativity

Christopher Straub*  and Sebastian Wolfschmidt* 

Department of Mathematics, University of Bayreuth, Bayreuth, Germany

E-mail: christopher.straub@uni-bayreuth.de and
sebastian.wolfschmidt@uni-bayreuth.de

Received 12 October 2023; revised 15 January 2024

Accepted for publication 25 January 2024

Published 9 February 2024



CrossMark

Abstract

We present a deep neural network which predicts the stability of isotropic steady states of the asymptotically flat, spherically symmetric Einstein–Vlasov system in Schwarzschild coordinates. The network takes as input the energy profile and the redshift of the steady state. Its architecture consists of a U-Net with a dense bridge. The network was trained on more than ten thousand steady states using an active learning scheme and has high accuracy on test data. As first applications, we analyze the validity of physical hypotheses regarding the stability of the steady states.

Keywords: Einstein–Vlasov, general relativity, stability, deep learning, neural network, numerics

1. Introduction

The Einstein–Vlasov system describes the dynamics of self-gravitating, collisionless matter by coupling Albert Einstein’s general relativity with the Vlasov (or collisionless Boltzmann) equation. This system holds strong significance in astrophysics, from modeling the evolution of star clusters and galaxy clusters to understanding the formation of black holes [4–7, 10, 21, 26, 36]. An ensemble of particles or gas—interacting solely via their self-generated gravitation—is treated as a continuous distribution of mass represented by a phase-space density function f . We are only interested in the asymptotically flat case with a vanishing cosmological constant.

* Authors to whom any correspondence should be addressed.



Original Content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](https://creativecommons.org/licenses/by/4.0/). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

The speed of light and the gravitational constant are normalized to unity. We always assume spherical symmetry. We state the system in this setting in section 2.1. For an overview of what is known about the Einstein–Vlasov system, we refer the interested reader to [4, 26, 36] and the references therein.

The Einstein–Vlasov system possesses a plethora of steady state solutions. Such a solution corresponds to a configuration with constant-in-time density. Hence, steady states are common models for star clusters which have settled in an equilibrium state. In this work we investigate isotropic steady states of the Einstein–Vlasov system. This means that the phase space density of the steady state depends only on the particle energy. The density of particles with a certain energy is determined by the prescribed *energy profile function* $\Phi: [0, 1[\rightarrow [0, \infty[$; see section 2.2 for a concrete description. A physically motivated [18, 37] assumption is that Φ is strictly increasing. This results in the concentration of ever more energetic particles to be decreasing within the equilibrium configuration. Another parameter of the steady state is the redshift factor $\kappa > 0$. Loosely speaking, large values of κ correspond to highly relativistic configurations, while choosing smaller values of κ lead to less relativistic steady states. Under suitable assumptions on Φ , any combination of Φ and $\kappa > 0$ yields a compactly supported steady state of the Einstein–Vlasov system. We review the steady state construction in more detail in section 2.2.

It should be mentioned that there exist steady states which are anisotropic, i.e. the distribution function not only depends on E , see [24]. Extending the present analysis to this more general case is a natural next step—however, this would require much more computational resources.

Stability theory analyzes the response of steady states to small perturbations. For instance, a star cluster in equilibrium might be perturbed by the gravitational force of other star clusters passing by. Loosely speaking, a *stable* steady state remains close to the equilibrium configuration after perturbation, while an *unstable* steady state evolves away from the equilibrium. In the present setting, slightly perturbing an unstable steady state either carries (parts of) the configuration to a new equilibrium or leads to the collapse into a black hole [12]. Obviously, we cannot expect to meet in nature an unstable steady state [9, chapter 5]. We must therefore first determine the stability of a steady state that is to be used as a model in reality.

Unfortunately, analyzing the stability of the aforementioned steady states turns out to be very challenging. In fact, neither the stability nor the instability of any such steady state has yet been mathematically proven. We refer to [27] for a recent review of the mathematical results towards this question. Let us explicitly mention here the following two results proven on the linearized level, i.e. in a simplified yet physically well motivated setting: For a fixed energy profile function Φ (satisfying suitable regularity assumptions), any not too relativistic steady state ($0 < \kappa \ll 1$) is stable [15], while any highly relativistic steady state ($\kappa \gg 1$) is unstable [14]. It is conjectured that this also holds for the actual non-linear stability behavior. Furthermore, in the astrophysics literature, some hypotheses regarding the precise location of the onset of instability, i.e. the κ -value where the steady states change from being stable to being unstable, have been developed.

Since the 1960’s, the stability of steady states of the Einstein–Vlasov system has also been analyzed numerically. We refer to [13, section 1.2] and [27, section 8] for an overview over past numerical investigations. The motivation is that a thorough understanding of the stability of steady states, derived numerically, ought to be useful for proving the stability properties mathematically. For instance, a recent numerical investigation [13] falsified a long-standing stability hypothesis from the astrophysics literature. In numerical stability investigations, one first computes a steady state, perturbs it slightly, and then evolves the resulting configuration over time using a suitable simulation of the Einstein–Vlasov system. We describe our explicit

numerical algorithm in section 2.3. Although the numerical simulations have become very accurate and reliable over the years, they are still computationally expensive. Determining the stability of a large amount of steady states, e.g. to test the validity of a physical hypothesis in depth, is usually not possible due to limited computational resources.

For this reason we have developed a deep neural network which can quickly and accurately predict the stability of steady states: the EVStabilityNet. The architecture of the EVStabilityNet consists of a U-Net with a dense bridge. These two parts mostly work in parallel and are unified only at late layers of the network. This architecture is motivated by mathematical aspects of the problem. We discuss the architecture in more detail in section 3.1.

The EVStabilityNet was trained on over 10^4 steady states, which were labeled as ‘stable’ or ‘unstable’ using the numerical simulations outlined above. In order to select the training data, we employed an active learning scheme [23, 33], i.e. after starting with a randomly chosen set of training data, cf section 3.2.1, we added more training data iteratively. We explain in detail the entire training process in section 3.2.

In section 3.3 we analyze the speed and accuracy of the EVStabilityNet. On randomly selected test data, the EVStabilityNet reaches an accuracy of 99%. We also test the accuracy on isotropic polytropic steady states as well as on the steady states from [13, figure 4]. The former are particularly natural from a mathematics point of view, whereas the latter exhibit highly diverse stability properties. The EVStabilityNet completes all tests with convincing accuracy.

We have made the EVStabilityNet publicly available so that it can be tested and applied by everyone. The download link and some notes on how to use it are provided in section 3.4.

In section 4, we apply the EVStabilityNet to test the validity of several hypotheses from the astrophysics literature regarding the stability of steady states. All these hypotheses are about linking (in)stability to a criterion that can be checked by examining specific quantities of the steady state, such as its total mass or redshift. To investigate the validity of such a hypothesis, we compare its statements about the stability of steady states to the predictions of the EVStabilityNet. Due to the EVStabilityNet’s low computational costs, we can perform this comparison on larger data sets than would otherwise be possible. If the stability predictions of the hypothesis and the EVStabilityNet consistently agree, it provides a strong indication of the hypothesis’s validity. Otherwise, the EVStabilityNet provides promising examples of steady states for which the hypothesis does not hold. These examples can then be double-checked with the conventional particle-in-cell method.

The first hypothesis we examine in this way is the so-called *weak binding energy hypothesis*. It states that if the redshift κ is increased for steady states with a fixed energy profile function Φ , the steady states are stable at least up to the first maximum of the binding energy E_b , see (2.12) for a definition of this quantity. This hypothesis has been tested and confirmed repeatedly in the literature [8, 12, 13, 19, 20, 25, 34, 35]. We can confirm it again here after testing it on 2000 randomly generated energy profile functions. The second hypothesis, the *strong binding energy criterion*, is more rigid than the first, claiming that the stability changes precisely at the first maximum of E_b . As noted above, this hypothesis has been proven to be not valid for general steady states in [13]. We come to the same conclusion here. It is notable that the hypothesis, nonetheless, seems to hold true for a significant proportion of the randomly selected steady states we tested it on. The third hypothesis originates from [8, 12, 35] and states that every steady state with negative binding energy has to be unstable. The predictions of the EVStabilityNet are in agreement with this hypothesis on a large number of randomly generated steady states. Lastly, we outline a future application concerning the presence of oscillating and damped star clusters at the end of section 4.

2. Mathematical and numerical background

2.1. The Einstein–Vlasov system

We describe the evolution of a star cluster by its phase-space density function f . Due to the assumption of spherical symmetry, $f = f(t, r, w, L)$ can be written as a function of the time variable $t \in \mathbb{R}$, the radial space variable $r \geq 0$, the radial momentum variable $w \in \mathbb{R}$, and the variable $L \geq 0$ describing the squared modulus of the angular momentum. The mass density and pressure associated to f are given by

$$\rho_f(t, r) = \frac{\pi}{r^2} \int_0^\infty \int_{\mathbb{R}} \varepsilon f(t, r, w, L) dw dL, \quad (2.1)$$

$$p_f(t, r) = \frac{\pi}{r^2} \int_0^\infty \int_{\mathbb{R}} \frac{w^2}{\varepsilon} f(t, r, w, L) dw dL, \quad (2.2)$$

respectively, where we introduce the abbreviation

$$\varepsilon := \sqrt{1 + w^2 + \frac{L}{r^2}}. \quad (2.3)$$

We consider Schwarzschild coordinates

$$ds^2 = -e^{2\mu(t,r)} dt^2 + e^{2\lambda(t,r)} dr^2 + r^2 (d\theta^2 + \sin^2(\theta) d\psi^2), \quad (2.4)$$

in which the metric parameterizing the geometry of spacetime is determined by the two functions $\mu_f = \mu_f(t, r)$ and $\lambda_f = \lambda_f(t, r)$ given by ρ_f and p_f via the differential equations

$$e^{-2\lambda_f} (2r\partial_r \lambda_f - 1) + 1 = 8\pi r^2 \rho_f, \quad e^{-2\lambda_f} (2r\partial_r \mu_f + 1) - 1 = 8\pi r^2 p_f. \quad (2.5)$$

Equation (2.5) correspond to Einstein's equations in Schwarzschild coordinates. The assumption that we consider an isolated system with a regular center (at $r = 0$) is implemented mathematically by incorporating the following boundary conditions on μ_f and λ_f :

$$\lim_{r \rightarrow \infty} \lambda_f(t, r) = \lim_{r \rightarrow \infty} \mu_f(t, r) = 0 = \lambda_f(t, 0). \quad (2.6)$$

We refer to [4, 26] for more background, but state here the following nice feature of Schwarzschild coordinates: The variable t can indeed be interpreted as the proper time of an observer looking onto the configuration from spatial infinity. The reason we choose these coordinates is because they simplify Einstein's equations sufficiently and because they are the most commonly used in the literature. However, past numerical investigations suggest that choosing other coordinates does not affect the stability properties which we will analyze later [12, 13]. The metric coefficients μ_f and λ_f determine the evolution of f via the partial differential equation

$$\partial_t f + e^{\mu_f - \lambda_f} \frac{w}{\varepsilon} \partial_r f - \left(w \partial_t \lambda_f + \varepsilon e^{\mu_f - \lambda_f} \mu_f' - \frac{L}{r^3 \varepsilon} e^{\mu_f - \lambda_f} \right) \partial_w f = 0. \quad (2.7)$$

This equation is known as the *Vlasov equation*. Accordingly, the system (2.1)–(2.7) is the *asymptotically flat Einstein–Vlasov system in Schwarzschild coordinates*. For brevity, we shall refer to it simply as the *Einstein–Vlasov system* throughout this article. It is a closed system for the phase-space density function f . A derivation of the system in the above form can be found in [26]. The existence theory is reviewed in [4].

2.2. Steady states

A steady state is a time-independent solution $f_0 = f_0(r, w, L)$ of the Einstein–Vlasov system (2.1)–(2.7). As motivated earlier, we seek such steady states by assuming that they only depend on the particle energy. More precisely,

$$f_0(r, w, L) = \Phi \left(1 - \frac{E(r, w, L)}{E_0} \right), \quad (2.8)$$

where the energy $E = E(r, w, L)$ of a particle with phase-space coordinates (r, w, L) is given by

$$E(r, w, L) = e^{\mu_{f_0}(r)} \varepsilon, \quad (2.9)$$

and $E_0 \in]0, 1[$ is the *cut-off energy* which we determine below. For the *energy profile function* $\Phi: [0, 1[\rightarrow [0, \infty[$, we make the following assumptions:

- ($\Phi 1$) Φ is continuous and increasing with $\Phi(0) = 0$.
- ($\Phi 2$) There exist constants $\eta_0 > 0$ and $0 < k \leq 2$ such that

$$\Phi(\eta) = \eta^k, \quad 0 < \eta < \eta_0. \quad (2.10)$$

The assumption ($\Phi 2$) looks rather restrictive at first glance, but we discuss it below. We always extend Φ continuously onto $] - \infty, 1[$ by setting $\Phi(\eta) := 0$ for $\eta < 0$, which implies $f_0(r, w, L) = 0$ if $E(r, w, L) \geq E_0$.

Making the ansatz (2.8) reduces the Einstein–Vlasov system for f_0 to a scalar integro-differential equation for the auxiliary quantity $y := \ln(E_0) - \mu_{f_0}$, see [28, section 1]. It is proven in [24] that for any initial value $y(0) = \kappa > 0$ and any energy profile function Φ satisfying ($\Phi 1$) and ($\Phi 2$) with $0 < k < \frac{3}{2}$, the integro-differential equation has a unique solution leading to a steady state f_0 with cut-off energy E_0 determined by $E_0 = \exp(\lim_{r \rightarrow \infty} y(r))$. This steady state is compactly supported and has finite (ADM-)mass, i.e. $\{f_0 > 0\}$ is bounded and $M := \int \varepsilon f_0 < \infty$, which shows that f_0 indeed serves as a realistic model for a star cluster in equilibrium. Our numerical simulations show that the same properties also hold for $\frac{3}{2} \leq k \leq 2$, which is why we include these cases here too.

By the scaling law from [13, section 2.3], multiplying the energy profile function Φ by a positive factor results in a steady state with the same stability behavior. This means that, after rescaling, our investigation also covers energy profile functions satisfying

$$\Phi(\eta) = c \eta^k, \quad 0 < \eta \ll 1, \quad (2.11)$$

for some $c > 0$ and k as above. Requiring equality in (2.11) is of mere technical nature—in numerical practice, one could also consider energy profile functions which only asymptotically satisfy (2.11), like the King model $e^\eta - 1$. Anyway, the steady states satisfying (2.11) constitute a sufficiently extensive and diverse class for the numerical analysis. Mathematically, even more general energy profile functions could be considered [24].

In conclusion, we obtain for every energy profile Φ a family of stationary solutions which we denote as $(f_\kappa)_{\kappa > 0}$. We only consider $\kappa < 1$ since we have never encountered a steady state which is stable for $\kappa > 0.8$.

A physically important quantity associated to a steady state f_0 is its (fractional) *binding energy*

$$E_b = \frac{N - M}{M}, \quad (2.12)$$

where $N = \int e^{\lambda_{f_0}} f_0$ is the particle number of f_0 and M is its mass.

2.3. Numerical stability analysis

In order to determine the stability of a steady state f_0 , we analyze the solution f of the Einstein–Vlasov system launched by the initial distribution $\mathring{f} := \alpha f_0$. One should think of \mathring{f} as a perturbation of the steady state f_0 . In fact, the observed stability behaviour is independent of the way of generating the perturbation, e.g. one could equivalently use dynamically accessible perturbations [8, 12]. The strength of the perturbation f is determined by the difference of the amplitude α and 1. As observed in [8, 12], choosing $\alpha > 1$ in the case of an unstable steady state f_0 leads to the collapse of the matter and the formation of a black hole. One manifestation of this behavior is that $e^{\mu_r(t,0)}$ gets close to 0. On the other hand, choosing $\alpha < 1$ leads to dispersion or a heteroclinic orbit [8, 12]. Although instability can always be observed for $\alpha > 1$ and $\alpha < 1$, in practice, it is more convenient to only consider $\alpha > 1$ since the resulting behavior can be distinguished more clearly from the stable case. In the case of a stable steady state, any slight perturbation, i.e. $|1 - \alpha| \ll 1$, results in the solution f remaining close to f_0 for all times.

We evolve the initial condition \mathring{f} numerically using a *particle-in-cell scheme*. This is the state-of-the-art method for simulating the Einstein–Vlasov system and is also the most commonly used in the literature [2, 3, 8, 12, 13, 30]. The basic idea is to split the (r, w, L) -support of \mathring{f} into a finite number of cells. In the center of each cell we place a numerical particle representing the contribution of its cell. These particles are propagated according to the characteristic system associated to the Vlasov equation (2.7); the arising ODE is solved using the fourth-order Runge–Kutta method. Based on the new positions of the numerical particles, the metric coefficients are updated after each time step. We refer the interested reader to the literature cited above for more details on the particle-in-cell scheme. The code realizing this numerical method is written in C++. Based on [22], it is parallelized using the Pthreads API. For the choice of all numerical parameters we have followed [13]. For instance, we use roughly $7 \cdot 10^7 \kappa^{\frac{3}{2}}$ numerical particles to represent the distribution function. The κ -dependency is included here in order to handle the more peaked metric coefficients occurring at more relativistic configurations. As described in [13, section 3], this method simulates the Einstein–Vlasov system with high accuracy.

The amplitude α , which determines the strength of the perturbation, is fixed at $\alpha = 1.0001$. In order to automatically detect a stable or unstable steady state, we distinguish between two situations: If $e^{\mu_r(t,0)}$ eventually converges to zero due to the perturbation, in which case we stop the simulation, we define the steady state as unstable. Otherwise, $e^{\mu_r(t,0)}$ stays roughly at $e^{\mu_r(0,0)}$ until $t = 500M$, and we deem the steady state stable unless we observe unusual behavior after manually inspecting $e^{\mu_r(t,0)}$. Note that this dichotomy has been found in all the numerical studies mentioned earlier.

3. The EVStabilityNet

3.1. Architecture

The neural network architecture for predicting the stability of steady states of the Einstein–Vlasov system is based on a U-Net [32] with a dense bridge bypassing the U-Net. The architecture is shown in detail in figure 1. For a general overview on the different types of layers and for background on deep learning as a whole, we refer to [1, 11].

U-Nets have their origins in image segmentation tasks [32] and are commonly used in that field. Their fully convolutional architecture enables them to effectively capture local features within the input data. In the present context, this could, e.g. be the slope of the energy profile

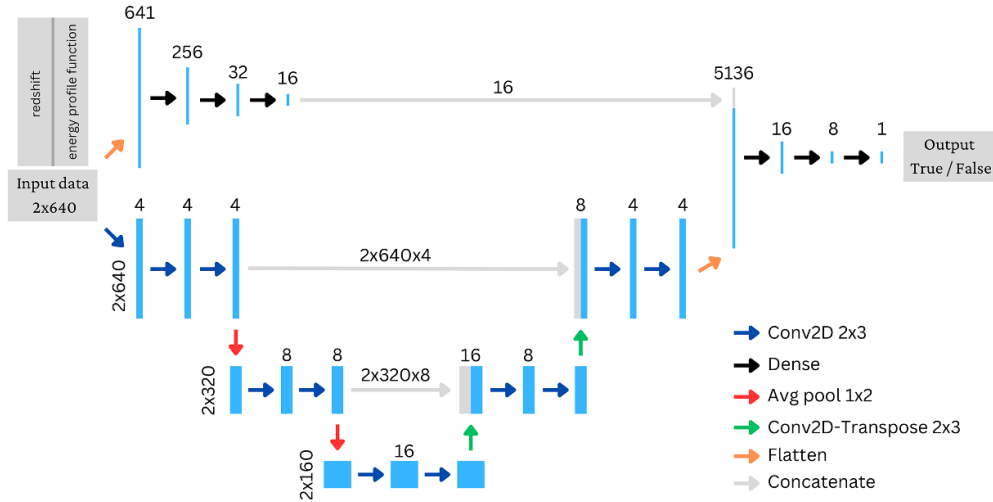


Figure 1. Architecture of the EVStabilityNet.

Table 1. Structure of the input data for the EVStabilityNet for the steady state associated to the energy profile $\Phi(\eta) = \eta$ and redshift $\kappa = 0.2$; note that $1 - e^{-0.2} \in]0.181, 0.182[$.

Index Nr.	0	1	2	...	180	181	182	183	...	639
κ	0.2	0.2	0.2	...	0.2	0.2	0.2	0.2	...	0.2
$\Phi(\eta)$	0	0.001	0.002	...	0.18	0.181	0	0	...	0

function Φ . The U-Net consists of an encoder and a decoder, both using convolutional and pooling layers. To avert loss of information in the encoding stage, skip connections transfer the information from the encoding path to the decoding path.

In addition to the fully convolutional U-Net, we integrate a dense bridge into the EVStabilityNet. This improves the network's capacity for capturing global features in the input data. This is useful in our context because global information about the steady state, such as its total mass or particle number, is connected to its stability in certain cases [12, 13].

As an aside, we have alternatively tried using a pure dense network, a pure U-Net, and different convolutional networks instead of the U-Net, but this has led to worse results at similar training costs.

The network takes as input a steady state as a two-dimensional tensor of size 2×640 , i.e. two channels of length 640. The first channel contains the redshift value $\kappa > 0$ repeated 640 times; this is done to ensure access to the value of κ in the convolutional layers. The second input channel stores the values of the energy profile function Φ , discretized with a step size of $\Delta\eta = 0.001$. We further set the values of Φ outside $[0, 1 - e^{-\kappa}]$ to 0 as these values have no significance in the computation of the steady state, cf section 2.2. Because we always choose $\kappa < 1$, this also shows that we generally only need the values of Φ on $[0, 1 - e^{-1}] \subset [0, 0.639]$, leading to the input length of 640. An example of the input data is given in table 1. For the dense path, we flatten the input tensor and delete the copies of the value of κ .

Finally, the outputs from the dense bridge and U-Net path are concatenated, and the architecture concludes with several additional dense layers to combine the information from both

paths. Overall, the network has 263937 trainable parameters. The network outputs a number $p \in]0, 1[$. If $p \leq 0.5$, the network predicts the input steady state as unstable, otherwise it predicts the steady state to be stable.

Throughout all layers, we use L^2 -regularization to prevent overfitting to the training data. We always use the ReLU activation function apart from the last layer where a sigmoid function is employed for binary classification. In order to arrive at the final architecture of the EVStabilityNet, we have implemented several stages of hyperparameter-tuning by, e.g. changing the number of filters in the convolutional layers, the number of neurons in the dense layers, the regularization parameters, etc.

3.2. Training

For the training of the EVStabilityNet, we have first developed an algorithm to randomly generate steady states of the Einstein–Vlasov system satisfying the conditions from section 2.2. From a large set of steady states generated this way, we then select interesting ones, label them, and use them as training data. We describe these steps in more detail in the following subsections.

3.2.1. Random steady states. The key step to create a random steady state as in section 2.2 is to randomly choose an energy profile function Φ satisfying $(\Phi 1)$ – $(\Phi 2)$. Let us describe the process we use for this; some examples of the resulting energy profile functions are depicted in figure 2:

- (i) Choose random numbers $k \in [\frac{1}{4}, 2]$ and $\eta_0 \in]0, 1[$.
- (ii) Choose a random integer $N \in \{0, \dots, 50\}$ which determines from how many parts the energy profile is generated.
- (iii) Generate the energy profile iteratively by adding up piecewise linear functions: We start with

$$\Phi_0(\eta) := \begin{cases} \eta, & 0 \leq \eta \leq \eta_0, \\ 0, & \text{else.} \end{cases}$$

For each $i \in \{1, \dots, N\}$, we choose random numbers $s \in [0, 10]$, $l \in [\eta_0, 1]$, and $r \in [l, 1]$ corresponding to the incline, the left boundary, and the right boundary of the piecewise function, respectively. We then define

$$\Phi_i(\eta) := \begin{cases} s(\eta - l), & l \leq \eta \leq r, \\ s(r - l), & \eta > r, \\ 0, & \text{else.} \end{cases}$$

- (iv) The final energy profile is given by

$$\Phi(\eta) := \left(\sum_{i=0}^N \Phi_i(\eta) \right)^k, \quad \eta \in [0, 1].$$

Finally, we choose a random redshift $0.005 < \kappa < 1$ where the lower limit is to prevent numerically expensive outliers; we never observe unstable steady states for such small values of κ anyways.

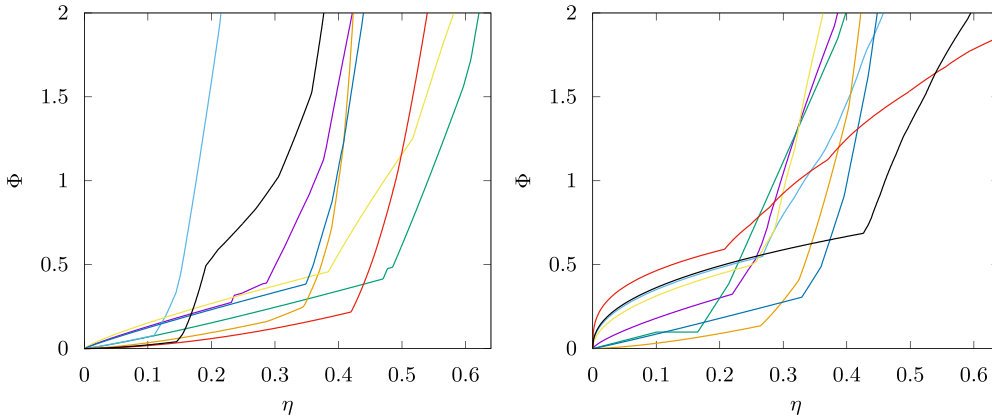


Figure 2. An illustration of sixteen randomly generated energy profile functions following the process described in (i)–(iv).

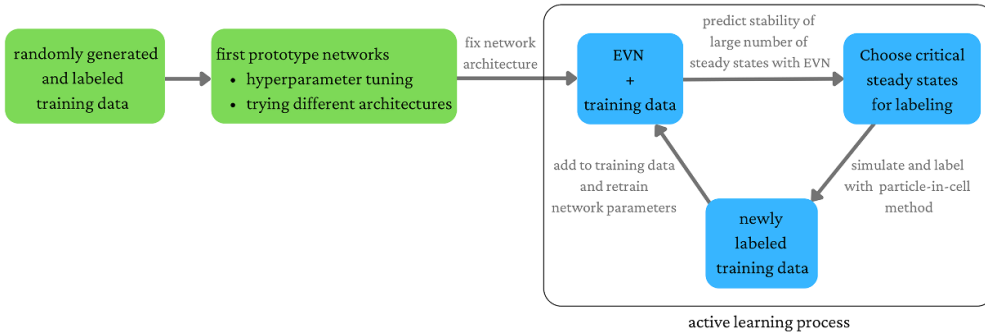


Figure 3. The training process for the EVStabilityNet using active learning.

Let us briefly comment on the range of k in (i): The precise bounds on k are chosen somewhat arbitrarily, and it is in principle possible to enlarge the range $[\frac{1}{4}, 2]$. However, including smaller or larger values of k requires more and more expensive numerical simulations.

A pair of the energy profile function and the redshift (Φ, κ) can be labeled as stable (1) or unstable (0) using the particle-in-cell method described in section 2.3.

3.2.2. Training process. The training process is illustrated in figure 3. The first part of the training process consisted of gathering a large basis of ~ 3000 randomly generated, labeled data, according to section 3.2.1. From this, we have implemented various versions of the final network in order to optimize accuracy, reducing variance and bias, training duration, and for fixing the hyperparameters suitably. Throughout the entire training process, we have split all labeled data into the actual training set (80%) and a cross-validation set (20%). The latter was used for hyperparameter tuning and to monitor possible overfitting to the training data.

From this point onwards, the architecture of EVStabilityNet is fixed and only the parameters are subject to further learning through an active learning exploration; see [23, 33] for more background on active learning. Concretely, we generate a large number of random, unlabeled steady states, as pairs of (Φ, κ) , and predict stability of the corresponding steady states via the

current version of the EVStabilityNet. We then choose those pairs for further labeling which are in some sense critical cases according to one of the following criteria:

- (i) The predicted value p is close to $\frac{1}{2}$, i.e. the EVStabilityNet is unsure about the stability behavior for the particular pair (Φ, κ) .
- (ii) The predicted values p_κ corresponding to (Φ, κ) probed along κ vary largely in a small range of values of κ .

The first case (i) is commonly known as the *least confidence criterion*, corresponding to data where the network is unsure or inaccurate in its prediction of the stability behavior. The second case (ii) is more adapted to the present problem and detects steady states which are in some sense close to the boundary where stability changes. Choosing to label these critical examples via the particle-in-cell method should yield much more valuable information compared to adding more randomly picked, labeled data. After adding a suitable amount of new training data through this active learning process, we retrain the EVStabilityNet parameters starting from the previous parameters of the network. For the actual training process, we train in parallel multiple networks employing different number of epochs as well as various learning rate schedules. Consequently, we choose the resulting network with the lowest cross-validation error as the next version of the EVStabilityNet. This whole process was repeated several times. Throughout the training process, the cross-validation error decreased, although we added edge case steady states to the training and cross-validation sets iteratively.

The training set's final size is 8163, the cross-validation set's 2145. The network's final version achieves a cross-validation accuracy of 96.83% and a training accuracy of 99.00%. This difference of over two percentage points can be justified by the recurrent retraining involved in the active learning approach, which is rather sensitive to overfitting. Nonetheless, we shall see in the following section that overfitting is not an issue on test data.

3.3. Performance

We now assess the EVStabilityNet's accuracy on three different test sets. Before doing so, let us briefly comment on its computational costs. The EVStabilityNet can predict the stability of 10^3 steady states within a few seconds when run on an ordinary laptop. Compared to the particle-in-cell program, which takes up to a day to determine the stability of a single steady state when run on a 40-core supercomputer, this is rather fast.

Performance on random test data: The first test set consists of 500 randomly generated, labeled data, i.e. we have determined the stability of these steady states as described in section 2.3; recall section 3.2.1 for the generation of random steady states. On this test set, the EVStabilityNet achieves an accuracy of 99.00%, making incorrect predictions in only five cases.

It should be noted that this error is comparable to the expected error rate resulting from numerical inaccuracies in the particle-in-cell scheme described in section 2.3. Since the latter is used to label the data, we conclude that the performance of the EVStabilityNet on this test set is as accurate as can be reasonably hoped for.

Performance on polytropes: An important class of steady states is obtained by the polytropes where the energy profile function is given by $\Phi(\eta) = \eta^k$ for $\eta > 0$ and some suitable $k > 0$. In the literature, it has been confirmed repeatedly [8, 12, 13] that—for a fixed polytropic exponent k —stability along the redshift κ changes at the first local maximum of the binding energy (2.12). We have compared this criterion with the predictions made by the

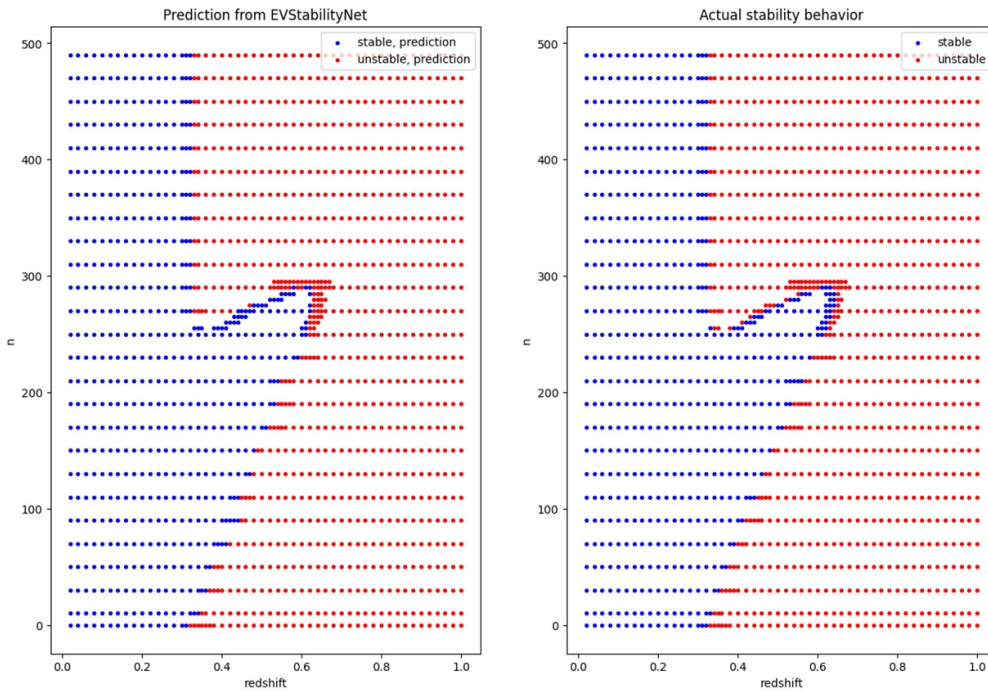


Figure 4. Performance of the EVStabilityNet on data from [13, figure 4].

EVStabilityNet for the polytropic energy profiles with $\frac{1}{4} \leq k \leq 2$ and $0.005 \leq \kappa \leq 1$. The result shows agreement in 99.06% of the cases. The errors are due to edge cases close to the point where stability changes along κ .

We note that, in contrast to the training set and the first test set, the polytropes are not derived from the random steady state generation from section 3.2.1. Despite the differing data distribution of the polytropes, the EVStabilityNet’s accuracy is similar as on the training set and the first test set. This shows that the EVStabilityNet is not overfitted to the distribution chosen in section 3.2.1.

Performance on data from [13]: In [13], a class of piecewise energy profiles has been introduced which boasts unique stability features, i.e. energy profiles which contradict the so-called strong binding energy hypothesis and ones that comprise multiple stability changes along the redshift κ . On the same set of data as illustrated in [13, figure 4], our network correctly predicts the stability behavior in 97.43% of the cases. The error here is larger than the test set error due to the focus on edge cases in [13, figure 4] which are inherently hard to predict correctly. The stability predictions of the EVStabilityNet on this data set alongside the actual stability behavior is depicted in figure 4. It is remarkable that the EVStabilityNet can reproduce these unique stability behavior results with good accuracy, as a large amount of computational resources went into finding the results published in [13].

3.4. How to use the neural network

The EVStabilityNet is available in the repository

<https://github.com/Sebastian-D-G/EVStabilityNet>

for public use. The network has been implemented and trained in Python 3.7.7 using Tensorflow 2.11.0. Besides the trained network, we also provide a simple working example of the network as well as the test set from above consisting of 500 random steady states.

We have described earlier, in particular in section 3.1, the input that can be used for the EVStabilityNet. Let us summarize this here once more.

- (i) The energy profile function Φ must satisfy $(\Phi 1)$ and $(\Phi 2)$. It should have neither discontinuities nor too large slopes.
- (ii) The redshift κ must satisfy $0.005 \leq \kappa \leq 1$.
- (iii) The input data must be of the format described in section 3.1, see table 1.

4. First applications

The reason we developed the EVStabilityNet is that we can investigate the (predicted) stability of a large number of steady state with little computational costs. Let us discuss some applications relying on this feature.

The weak binding energy hypothesis: A long standing hypothesis, which has been confirmed over and over again numerically [8, 12, 13, 19, 25, 34, 35] and made plausible by physical reasoning [20], is the weak binding energy hypothesis. It claims that steady states with the same energy profile function parameterized by the redshift κ are stable at least up to the first local maximum of the binding energy (2.12) as a function in κ . We test the weak binding energy hypothesis by generating 2000 random energy profiles as in section 3.2, determining the location of the first local binding energy maximum, and consequently predicting stability for the corresponding family of steady states with the EVStabilityNet. The results show that the weak binding energy hypothesis holds for 1984 families of steady states. In 16 cases the EVStabilityNet predicts instability closely before the first local maximum of the binding energy. However, we have checked these handful of steady states manually via the particle-in-cell method, and ascertained that these steady states are in fact stable, i.e. the prediction of the EVStabilityNet was slightly incorrect; this is to be expected with the accuracy of the EVStabilityNet at roughly 99%. Overall, we take our results as strong evidence for the validity of the weak binding energy hypothesis, as this hypothesis was never tested on such a large dataset before.

The (strong) binding energy criterion: The first local maximum of the binding energy in κ along a family of steady states often coincides with the onset of instability. For example, the polytropes seem to follow this rule, recall section 3.3. With the same techniques as with the weak binding energy hypothesis, we have checked how often along a family of steady states, this ‘(strong) binding energy criterion’ holds. We find that it is satisfied for 1220 of the 2000 energy profile functions, i.e. in 61.0% of the cases. Given that it was shown in [13] that the binding energy criterion does not hold in general, it is remarkable that it is nonetheless valid for such a large number of energy profiles. Further research is required to grasp the relation between the binding energy and stability analytically.

Negative binding energy: In [8, 35], it was observed numerically and discussed formally that suitable perturbations of steady states with negative binding energy E_b always result in complete dispersion. This was put into perspective in [12], where it was found that steady states with negative binding energy, which do not completely disperse, exist. Nonetheless, it is still commonly believed that every steady state with $E_b < 0$ is unstable. We conducted an analysis of $\sim 188\,000$ randomly generated steady states with negative binding energy to check the validity of this hypothesis. The EVStabilityNet predicts that all of these steady states are indeed unstable, thereby further strengthening the hypothesis.

Damping vs. oscillation for stable steady states: A future application, which we only sketch here, is to further investigate the dynamical behavior of slightly perturbed stable steady states. In a related context, it has recently been shown that, depending on the underlying steady state, such solutions either oscillate around the original equilibrium or converge towards it in a suitable sense ('damping') [16], see also [17, 31]. Similar behaviors have also been observed numerically for the Einstein–Vlasov system [8, 12, 13, 36]. First steps towards proving the existence of oscillations around certain stable steady states in this context are made in [36, chapter 6]. Nevertheless, the connection between stable steady states and the presence of damped or oscillating behavior (as well as the frequency of the oscillation) is elusive for the Einstein–Vlasov system. To advance further research, a detailed numerical study of the behavior of many slightly perturbed stable steady state will certainly be helpful. Since one has to know which steady states are stable in the first place, the fast stability predictions of the EVStabilityNet can be used to significantly reduce the computational costs of such investigation.

We hope that the EVStabilityNet can be used for a larger range of applications beyond our initial scope, making it a valuable tool for researchers exploring the diverse field of stability of steady states of the Einstein–Vlasov system.

Data availability statement

The data cannot be made publicly available upon publication because they are not available in a format that is sufficiently accessible or reusable by other researchers. The data that support the findings of this study are available upon reasonable request from the authors.

Acknowledgments

We are grateful to Gerhard Rein for his guidance and for the freedom he granted us to conduct independent research, which led to this paper.

ORCID iDs

Christopher Straub  <https://orcid.org/0000-0003-2194-0256>

Sebastian Wolfschmidt  <https://orcid.org/0000-0002-5403-8616>

References

- [1] Aggarwal C C 2018 *Neural Networks and Deep Learning, A Textbook* (Springer)
- [2] Ames E, Andréasson H and Rinne O 2021 Dynamics of gravitational collapse in the axisymmetric Einstein–Vlasov system *Class. Quantum Grav.* **38** 105003
- [3] Ames E, Andréasson H and Rinne O 2023 Hoop and weak cosmic censorship conjectures for the axisymmetric Einstein–Vlasov system *Phys. Rev. D* **108** 064054
- [4] Andréasson H 2011 The Einstein–Vlasov system/kinetic theory *Living Rev. Relativ.* **14** 4
- [5] Andréasson H 2012 Black hole formation from a complete regular past for collisionless matter *Ann. Henri Poincaré* **13** 1511–36
- [6] Andréasson H, Kunze M and Rein G 2010 Gravitational collapse and the formation of black holes for the spherically symmetric Einstein–Vlasov system *Q. Appl. Math.* **68** 17–42

- [7] Andréasson H, Kunze M and Rein G 2011 The formation of black holes in spherically symmetric gravitational collapse *Math. Ann.* **350** 683–705
- [8] Andréasson H and Rein G 2006 A numerical investigation of the stability of steady states and critical phenomena for the spherically symmetric Einstein-Vlasov system *Class. Quantum Grav.* **23** 3659–77
- [9] Binney J and Tremaine S 2008 *Galactic Dynamics (Princeton Series in Astrophysics vol 13)* 2nd edn (Princeton University Press)
- [10] Ehlers J 1973 Survey of general relativity theory *Relativity, Astrophysics and Cosmology* (D. Reidel Publishing Company)
- [11] Goodfellow I, Bengio Y and Courville A 2016 *Deep Learning* (MIT Press)
- [12] Günther S, Körner J, Lebeda T, Pötzl B, Rein G, Straub C and Weber J 2021 A numerical stability analysis for the Einstein-Vlasov system *Class. Quantum Grav.* **38** 035003
- [13] Günther S, Straub C and Rein G 2021 Collisionless equilibria in general relativity: stable configurations beyond the first binding energy maximum *Astrophys. J.* **918** 48
- [14] Hadžić M, Lin Z and Rein G 2021 Stability and instability of self-gravitating relativistic matter distributions *Arch. Ration. Mech. Anal.* **241** 1–89
- [15] Hadžić M and Rein G 2015 On the small redshift limit of steady states of the spherically symmetric Einstein-Vlasov system and their stability *Math. Proc. Camb. Phil. Soc.* **159** 529–46
- [16] Hadžić M, Rein G, Schrecker M and Straub C 2023 Damping versus oscillations for a gravitational Vlasov-Poisson system (arXiv:2301.07662) p 49
- [17] Hadžić M, Rein G and Straub C 2022 On the existence of linearly oscillating galaxies *Arch. Ration. Mech. Anal.* **243** 611–96
- [18] Ipser J R 1969 Relativistic, spherically symmetric star clusters. II. Sufficient conditions for stability against radial perturbations *Astrophys. J.* **156** 509–27
- [19] Ipser J R 1969 Relativistic, spherically symmetric star clusters. III. Stability of compact isotropic models *Astrophys. J.* **158** 17–43
- [20] Ipser J R 1980 A binding-energy criterion for the dynamical stability of spherical stellar systems in general relativity *Astrophys. J.* **238** 1101–10
- [21] Ipser J R and Thorne K S 1968 Relativistic, spherically symmetric star clusters. I. Stability theory for radial perturbations *Astrophys. J.* **154** 251–70
- [22] Korch M, Ramming R and Rein G 2013 Parallelization of particle-in-cell codes for nonlinear kinetic models from mathematical physics *2013 42nd Int. Conf. on Parallel Processing (ICPP)* (IEEE Computer Society) pp 523–9
- [23] Kumar P and Gupta A 2020 Active learning query strategies for classification, regression and clustering: a survey *J. Comput. Sci. Technol.* **35** 913–45
- [24] Ramming T and Rein G 2013 Spherically symmetric equilibria for self-gravitating kinetic or fluid models in the nonrelativistic and relativistic case—a simple proof for finite extension *SIAM J. Math. Anal.* **45** 900–14
- [25] Rasio F A, Shapiro S L and Teukolsky S A 1989 On the existence of stable relativistic star clusters with arbitrarily large central redshifts *Astrophys. J.* **336** 63–66
- [26] Rein G 1995 *The Vlasov-Einstein System With Surface Symmetry* (Habilitationsschrift)
- [27] Rein G 2023 Stability and instability results for equilibria of a (relativistic) self-gravitating collisionless gas—a review *Class. Quantum Grav.* **40** 193001
- [28] Rein G and Rendall A D 1993 Smooth static solutions of the spherically symmetric Vlasov-Einstein system *Ann. Inst. Henri Poincaré Phys. Théor.* **59** 383–97
- [29] Rein G and Rendall A D 1992 Global existence of solutions of the spherically symmetric Vlasov-Einstein system with small initial data *Commun. Math. Phys.* **150** 561–83 Rein G and Rendall A D 1996 *Commun. Math. Phys.* **176** 475–8 (erratum)
- [30] Rein G, Rendall A D and Schaeffer J 1998 Critical collapse of collisionless matter: a numerical investigation *Phys. Rev. D* **58** 044007
- [31] Rioseco P and Sarbach O 2020 Phase space mixing in an external gravitational central potential *Class. Quantum Grav.* **37** 195027
- [32] Ronneberger O, Fischer P and Brox T 2015 U-Net: convolutional networks for biomedical image segmentation *MICCAI 2015* vol III pp 234–41
- [33] Settles B 2009 Active learning literature survey *Computer Sciences Technical Report 1648* (University of Wisconsin-Madison)

- [34] Shapiro S L and Teukolsky S A 1985 Relativistic stellar dynamics on the computer. I. Motivation and numerical method *Astrophys. J.* **298** 34–57
- [35] Shapiro S L and Teukolsky S A 1985 Relativistic stellar dynamics on the computer. II. Physical applications *Astrophys. J.* **298** 58–79
- [36] Wolfschmidt S 2023 Stability and oscillations of star clusters in general relativity *Doctoral Thesis* Universität Bayreuth
- [37] Zel'dovich Y B and Novikov I D 1971 *Relativistic Astrophysics. Volume 1: Stars and Relativity* (University of Chicago Press)