Detecting fast neutrino flavor conversions with machine learning

Sajad Abbar[1] and Hiroki Nagakura[2]

[1]*Max-Planck-Institut für Physik (Werner-Heisenberg-Institut),*
Föhringer Ring 6, 80805 München, Germany
[2]*Division of Science, National Astronomical Observatory of Japan, 2-21-1 Osawa,*
Mitaka, Tokyo 181-8588, Japan

(Received 28 September 2023; accepted 17 January 2024; published 31 January 2024)

Neutrinos in dense environments like core-collapse supernovae (CCSNe) and neutron star mergers (NSMs) can undergo fast flavor conversions (FFCs) once the angular distribution of neutrino lepton number crosses zero along a certain direction. Recent advancements have demonstrated the effectiveness of machine learning (ML) in detecting these crossings. In this study, we enhance prior research in two significant ways. First, we utilize realistic data from CCSN simulations, where neutrino transport is solved using the full Boltzmann equation. We evaluate the ML methods' adaptability in a real-world context, enhancing their robustness. In particular, we demonstrate that when working with artificial data, simpler models outperform their more complex counterparts, a noteworthy illustration of the bias-variance tradeoff in the context of ML. We also explore methods to improve artificial datasets for ML training. In addition, we extend our ML techniques to detect the crossings in the heavy-leptonic channels, accommodating scenarios where ν_x and $\bar{\nu}_x$ may differ. Our research highlights the extensive versatility and effectiveness of ML techniques, presenting an unparalleled opportunity to evaluate the occurrence of FFCs in CCSN and NSM simulations.

DOI: 10.1103/PhysRevD.109.023033

I. INTRODUCTION

Core-collapse supernovae (CCSNe) and neutron star mergers (NSMs) are cataclysmic stellar events that represent the dramatic culmination of massive stars' life cycles and the collision and coalescence of incredibly dense remnants, respectively [1–4]. These events not only mark the end of massive stars and dense objects, but also unveil some of the most energetic and enigmatic phenomena in the universe. In the heart of these cosmic fireworks, one of the most fascinating processes at play is the neutrino emission, which are released in vast quantities during CCSNe and NSMs.

As they journey through the extraordinarily dense and extreme conditions within these events, neutrinos undergo an intriguing phenomenon known as collective neutrino oscillations. This fascinating behavior arises from their interactions with the dense background neutrino gas, where coherent forward scatterings play a pivotal role. This phenomenon occurs in a nonlinear and collective manner, creating a rich tapestry of flavor transformations [5–11] (for a recent review see Ref. [12]).

Of particular interest are the so-called *fast* flavor conversions (FFCs), which occur on scales characterized by $\sim G_\mathrm{F}^{-1} n_\nu^{-1}$ (see, e.g., Refs. [13–60]). Here, G_F represents the Fermi coupling constant, and n_ν denotes the neutrino number density. These FFCs can take place on timescales much shorter than what would be expected in the vacuum.

FFCs occur *iff* the angular distribution of the neutrino lepton number, defined as,

$$G(\mathbf{v}) = \sqrt{2} G_\mathrm{F} \int_0^\infty \frac{E_\nu^2 \mathrm{d}E_\nu}{(2\pi)^3} [(f_{\nu_e}(\mathbf{p}) - f_{\nu_x}(\mathbf{p})) - (f_{\bar{\nu}_e}(\mathbf{p}) - f_{\bar{\nu}_x}(\mathbf{p}))], \tag{1}$$

crosses zero at some $\mathbf{v} = \mathbf{v}(\mu, \phi_\nu)$, with $\mu = \cos\theta_\nu$ [30]. Here, E_ν, θ_ν, and ϕ_ν are the neutrino energy, the zenith, and azimuthal angles of the neutrino velocity, respectively, and f_ν's are the neutrino occupation numbers. When ν_x and $\bar{\nu}_x$ have similar angular distributions, a scenario commonly observed in state-of-the-art CCSN simulations, this expression transforms into the conventional νELN (neutrino electron lepton number).

Exploring νELN crossings necessitates access to the complete angular distributions of neutrinos. However, obtaining such detailed angular information poses a significant challenge in most cutting-edge CCSN and NSM simulations due to the prohibitive computational demands involved.

As a practical alternative, many simulations simplify neutrino transport by relying on a limited set of angular distribution moments [61–63]. In our specific investigation, we focus on radial moments, defined as,

$$I_n = \int_{-1}^{1} \mathrm{d}\mu\, \mu^n \int_0^\infty \int_0^{2\pi} \frac{E_\nu^2 \mathrm{d}E_\nu \mathrm{d}\phi_\nu}{(2\pi)^3} f_\nu(\mathbf{p}). \tag{2}$$

These moments capture the key aspects of the neutrino angular distribution, at the same time allowing for its more computationally manageable treatment.

Note that our current emphasis lies on axisymmetric crossings, directing our attention specifically to radial moments where the angular distribution integrates over ϕ_ν. It is important to note that nonaxisymmetric crossings fall outside the scope of our present study, a matter to be explored in future works.

Despite the inherent loss of a significant amount of information when considering only a select few neutrino angular moments, ingenious methods can still be devised to harness these limited information for assessing FFCs in CCSN and NSM simulations. In the initial stages of research in this field, the primary focus was on analytical or semianalytical techniques [25,31,64–68]. While these methods have demonstrated their ability to capture ELN crossings and have found relative success in the literature, they are constrained in their performance. This limitation arises from their either sluggish computational speed or their inefficiency in identifying ELN crossings which impacts their ability to efficiently detect FFCs in real-time simulations. Specifically, the most efficient techniques tend to be noticeably sluggish, and their development can be relatively intricate when starting from scratch.[1]

Recent research has demonstrated the effectiveness of machine learning (ML) techniques in identifying FFC in CCSN and NSM simulations [69]. While ML methods are data-intensive and require an initial training phase with data, it's important to note that once trained, they exhibit exceptional speed and efficiency. This presents a promising avenue for real-time detection of FFI's within the context of CCSN and NSM simulations. Moreover, integrating pretrained ML models is a straightforward procedure, significantly reducing the requirement for extensive coding work, even when analyzing the occurrence of FFCs in a postprocessing phase. In fact, ML techniques offer the fastest and most precise approach to detect FFCs, and their performance in more complex environments can be further enhanced as and if one can train them on the data obtained from those environments.

In this paper, we advance the prior study in two pivotal directions. First, in earlier work, ML models were trained using artificial data generated from specific parametric angular distributions. While these models showed promise, they were only partially validated against a limited amount of realistic data from NSM remnant simulations. It is essential for ML techniques to be trained and tested on data that closely resembles real-world simulations.[2] In our study, we take a significant step forward by utilizing authentic data from a CCSN simulation, where neutrino transport was modeled using the full Boltzmann equation. This allows us to assess the adaptability of ML methods in a real-world context and examine their limitations as well as their optimal performance range. Furthermore, we acknowledge that artificial data are more readily available and can offer broader distributions that are expected in CCSN and NSM environments. Consequently, we also investigate methods to enhance artificial datasets for ML training purposes.

In addition, in our previous study, we assumed that the distributions of ν_x and $\bar{\nu}_x$ were identical. From a ML perspective, this simplification facilitated the development of our ML module by requiring a smaller number of features and a more efficient classification. In this work, we develop ML techniques to detect the crossings regarding the neutrino heavy-leptnic channel distribution (νXLN) addressing scenarios where ν_x and $\bar{\nu}_x$ may exhibit differences, a *previously unexplored* area in the literature. While our results may be slightly less accurate than those in the previous scenario, ML methods still prove remarkably effective in identifying the occurrence of FFC in this scenario..

In the upcoming section, we delve into our CCSN model, the source of our data. We then assess the performance of ML methods in detecting FFCs in our CCSN model, specifically focusing on the detection of νELN crossings. Finally and before presenting our conclusions, we analyze

[1] As discussed in the text, the traditional methodologies can be broadly categorized into two classes: analytical methods, which concentrate on the instability of a specific mode, and semi-analytical models, which focus on identifying descriptive polynomials or fitting functions to angular distributions. While analytical approaches boast speed and ease of implementation, their efficiency in the CCSN environment is generally questioned due to specific conditions required for their usefulness. Conversely, semi-analytical methods, while being more efficient in capturing crossings, pose a significant computational burden. In contrast, ML, by focusing on simulation-derived moments (it can only focus on I_0 and I_1), presents a promising alternative that mitigates the influence of variations and artefacts introduced by closure relations. This should be compared with the results, e.g., in Ref. [65], that showed using the semianalytical methods could only capture $\lesssim$50% of the ELN crossings (although more angular information were provided, namely I_2 and I_3). Therein, it was also shown that the simple method developed in Ref. [64] could not capture any ELN crossings.

[2] Henceforth in this paper, whenever the term *realistic* or *real-world data* is mentioned, it specifically refers to the data acquired from our axisymmetric CCSN simulation.

the performance of ML methods in detecting the crossings in the heavy-leptnic channel.

II. CCSN MODEL

Here, we construct a ML technique to detect νELN- and XLN angular crossings based on an axisymmetric CCSN model with full Boltzmann neutrino transport [70]. Before entering into the detail of our ML, we briefly describe our CCSN model providing neutrino dataset used for ML training and its testing.

The numerical simulation was carried out by a Boltzmann-neutrino-radiation hydrodynamic code [71] with some special treatments to handle proper motions of proto-neutron star (PNS) [72,73]. In this model, a table of multinuclear variational method equation-of-state [74] was used, and the nuclear abundance in the table was also used to compute neutrino-matter interactions for the consistent treatment between EOS and weak rates [75]. We used a 11.2 solar mass progenitor model in Ref. [76].

One of the noticeable features in the CCSN model is that large-scale asymmetric neutrino emission emerges > 150 ms after bounce, that corresponds to the timing of asymmetric shock expansion and the onset of PNS proper motion (see Fig. 1 in Ref. [70]). We note that the asymmetric emission is clearly anticorrelated between ν_e and $\bar{\nu}_e$, which is attributed to the distribution of electron fraction (Y_e) in the vicinity of PNS, and $\bar{\nu}_e$ tends to be more abundant in low Y_e environments. As shown in Ref. [77], the increase of $\bar{\nu}_e$ reduces the disparity between ν_e and $\bar{\nu}_e$ angular distributions, leading to enhance the possibility of νELN crossings. In the region with higher Y_e environments, on the other hand, ν_e becomes much higher than $\bar{\nu}_e$, indicating that FFI is unlikely to occur.

In the CCSN model, νELN angular crossings are observed rather stably at > 200 ms (see Fig. 2 in Ref. [77]); hence, we employ three different time snapshots for our ML training (200, 250, and 300 ms after bounce), in which there are both spatial regions with and without νELN angular crossings. It should also be mentioned that νELN crossings are also observed in PNS convective layer and in preshock regions, which are consistent with previous studies [78–80]. For more detailed discussion about neutrino angular distributions associated with arguments of νELN crossings, we refer readers to Ref. [77].

III. ML ALGORITHMS

ML, at the crossroads of computer science and artificial intelligence, is transforming how computers learn and make decisions from data. By unraveling intricate patterns, ML drives progress across diverse domains, from image and speech recognition to healthcare, finance, and autonomous vehicles.

Recent advancements have demonstrated the effective utilization of ML algorithms for detecting νELN crossings in CCSN and NSM simulations [69]. In this context, we commence with a brief overview of the data preparation and ML techniques employed in Ref. [69]. Subsequently, we provide a comprehensive discussion of our research findings.

To effectively train and evaluate our ML algorithms, it is imperative to possess a substantial dataset comprising labeled values for I_0 and I_1 associated with ν_e and $\bar{\nu}_e$. These labels are instrumental in discerning the presence or absence of νELN crossing. It is worth highlighting that our current emphasis is primarily on the first two moments. These moments are of particular interest as they are the ones typically tracked directly in the simulation processes.

In order to train our ML algorithms, we partially employ two parametric neutrino angular distributions which have been widely used in the literature [46,67,81], namely, the maximum entropy distribution defined as,

$$f_\nu^{\max-\mathrm{ent}}(\mu) = \exp[\eta + a\mu], \tag{3}$$

and the Gaussian distribution,

$$f_\nu^{\mathrm{Gauss}}(\mu) = A \exp\left[-\frac{(1-\mu)^2}{a}\right], \tag{4}$$

with,

$$f_\nu(\mu) = \int_0^\infty \int_0^{2\pi} \frac{E_\nu^2 \mathrm{d}E_\nu \mathrm{d}\phi_\nu}{(2\pi)^3} f_\nu(\mathbf{p}). \tag{5}$$

Here the parameters a, η, and A determines the overall neutrino number density and the shape of the neutrino distributions.

In addition, in order to improve the efficiency of our ML algorithms, we do feature engineering and instead of I_0 and I_1 of ν_e and $\bar{\nu}_e$ which are the provided information, we use,

$$\alpha = I_0^{\bar{\nu}_e}/I_0^{\nu_e}, \qquad F_{\nu_e} = I_1^{\nu_e}/I_0^{\nu_e}, \quad \text{and} \quad F_{\bar{\nu}_e} = I_1^{\bar{\nu}_e}/I_0^{\bar{\nu}_e}, \tag{6}$$

as the relevant features to be considered in the ML algorithms. This is justified by bearing in mind that an overall normalization factor does not affect the occurrence of νELN crossings.

To facilitate the training and evaluation of ML algorithms, it is essential to partition the dataset into three distinct sets: (i) The training set: This subset is employed to train the ML algorithm, allowing it to learn patterns and relationships within the data, (ii) The development set: This set serves as a tool for fine-tuning the algorithm's hyperparameters, ensuring optimal performance, and (iii) The test set: This portion is dedicated to assessing the ML method's performance on previously unseen data, providing a reliable measure of its effectiveness. In practice, one should randomly distribute the data among these different datasets to avoid any sort of bias. In our analysis, we have randomly distributed the data given the $0.8:0.1:0.1$ fraction for the training, development, and the test sets, respectively.

To comprehensively assess the effectiveness of our ML algorithm, we go beyond mere accuracy, which can be a somewhat simplistic measure. Instead, we consider also more detailed evaluations using precision and recall metrics, defined as,

$$\begin{aligned} \text{accuracy} &= \frac{T_p + T_n}{T_p + T_n + F_p + F_n} \\ \text{precision} &= \frac{T_p}{T_p + F_p} \\ \text{recall} &= \frac{T_p}{T_p + F_n} \\ F_1 &= 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}, \end{aligned} \tag{7}$$

with $T(F)_{p(n)}$ denoting true (false) positive (negative) classifications. A discerning reader will notice that the precision/recall metric informs us about the reliability/detectability of classifications, while F_1 is their harmonic mean. In this study, we opt for accuracy as the suitable metric because we aim for equal sensitivity to the presence or absence of νELN crossings.

In this study we consider the following ML algorithms: (i) Logistic regression (LR): a statistical classification algorithm that models the probability of a binary outcome. LR turns out to be one of the most promising ML algorithm to be used in detecting νELN crossings [69], (ii) k-nearest neighbors (KNN): an intuitive learning algorithm that classifies data points based on the majority class of their k-nearest neighbors in the feature space, (iii) Support vector machine (SVM): a powerful algorithm that separates data into classes by finding the hyperplane that maximizes the margin between them in a high-dimensional space, and (iv) Decision tree (DT): a treelike model used for both classification and regression tasks, where the data is split into subsets based on feature conditions, ultimately leading to a decision or prediction.

TABLE I. A summary of the metric scores of the previously-trained ML algorithms (using artificial data) tested on the realistic dataset. This is to be compared with Table I. in Ref. [69]. Alongside each algorithm, one can find its corresponding accuracy score.

LR ($n = 9$) (68%)			
	Precision	Recall	F_1-score
No crossing	72%	82%	77%
Crossing	59%	43%	50%
KNN ($n = 3$) (77%)			
	Precision	Recall	F_1-score
No crossing	77%	89%	83%
Crossing	75%	55%	63%
SVM (87%)			
	Precision	Recall	F_1-score
No crossing	98%	81%	89%
Crossing	75%	98%	85%
Decision tree (71%)			
	Precision	Recall	F_1-score
No crossing	74%	84%	79%
Crossing	63%	48%	55%

There are two final aspect of the LR and SVM algorithms that requires a bit of clarification. While LR incorporates the nonlinear logistic function, it fundamentally operates as a linear classifier. Consequently, it cannot be directly applied to the detection of νELN crossings, which inherently represents a nonlinear problem [69]. To overcome this limitation, it becomes necessary to undertake a preprocessing step involving nonlinear transformations and the creation of new features based on the original three features involved in the problem. The degree of polynomial transformation, being a hyperparameter of this algorithm, plays a crucial role in this process.[3]

Regarding the SVM algorithm, we should mention that we employ the radial basis function (RBF) kernel, defined as, $\mathcal{K}(x, x') = \exp(-\gamma \|x - x'\|^2)$. Here γ is a hyper-parameter which is set to be $\gamma = 100$ [69].

In the next part, we present our results. In addition, to promote transparency and collaboration, we have made our ML methodologies available on GitHub.

A. ML-based detection of νELN crossings using the SN data

We initially assess the performance of the pre-trained ML models of Ref. [69] (using artificial data) on the realistic dataset obtained from the simulation. The metric scores for the performance of these pretrained ML algorithms are presented in Table I. It becomes evident that the performance of ML models, which were initially trained on artificial data, degrades when applied to realistic data.

[3] The polynomial transformation method produces an augmented feature matrix by considering all possible polynomial combinations of the original features up to a specified degree. To illustrate, if an input sample has two dimensions denoted as (a, b), the resulting degree-2 polynomial features would include (a, b, a^2, ab, b^2) (assuming no bias term is taken into account).

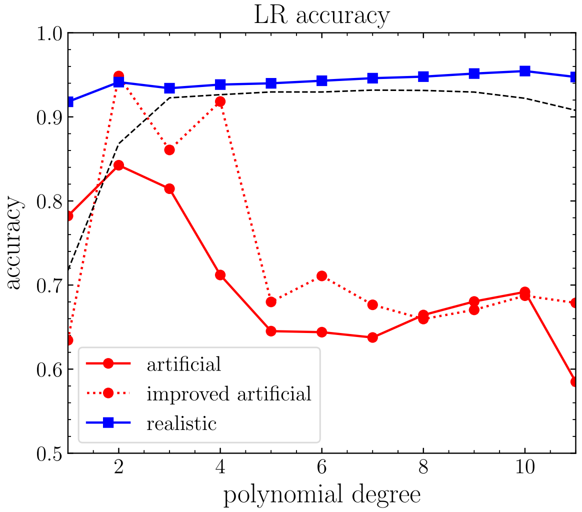

FIG. 1. The accuracy of the LR algorithms evaluated on the realistic dataset for models trained on various training sets, with a focus on the impact of polynomial degree of the nonlinear transformations. It is noteworthy that LR models trained on the improved artificial training sets can achieve comparable accuracies to those trained on realistic data at lower polynomial degrees, implying simpler model structures. The black dashed line represents the performance of the old LR model trained on artificial data when tested on its dedicated test set.

Notably, the accuracies of LR and SVM are of paramount concern. Specifically, the LR model (here was considered with a polynomial degree of 9) exhibits a significant deviation from its accuracy on artificial data (as indicated in Table I of Ref. [69]). This indicates poor generalization of the old LR algorithm with a polynomial degree $n = 9$, a known issue referred to as high variance in the context of ML.

Another enlightening aspect centers on the SVM model's relatively good performance, consistently achieving high accuracy even when confronted with previously unseen data. This serves as yet another compelling example of the effectiveness of maximizing margins to separate classes, underscoring its conceptual soundness and its ability to enhance the model's overall generalizability in ML.

However and as depicted in Fig. 1, the former LR model (red-curve) trained on synthetic data, attains its peak performance around $n \simeq 2$ achieving an accuracy of approximately 85%. In simpler terms, the more straightforward models outperform their counterparts when it comes to νELN detection on previously unseen data. This serves as a prime illustration of the bias-variance tradeoff within the realm of ML. As we see later on, it seems to be a general observation that when considering LR, opting for nonlinear transformations with lower polynomial degrees tends to yield better results, at least as long as a variation between the training and test sets is expected.

While achieving an accuracy of ~85% is already acceptable, it's important to note that the performance of models developed using artificial datasets can still be enhanced. The initial ML models were trained on data with α values ranging from 0.03 to 2.5, alongside random selections of F_{ν_e} and $F_{\bar{\nu}_e}$ in the (0, 1) range. However, this approach, while suitable for an initial step, does not align with realistic conditions. In realistic simulations of CCSNe, it is anticipated that $F_{\nu_e} \lesssim F_{\bar{\nu}_e}$ (note that this could be a bit distorted in the case of NSM where the radiation fields can be qualitatively different).

To address this issue, we enhance our ML model by training it with artificial data while considering F_{ν_e} within the range of $(0.6F_{\bar{\nu}_e}, F_{\bar{\nu}_e})$. The performance of such a ML model on realistic data is illustrated in Fig. 1 (red-dotted curve), revealing two significant insights. Firstly, high variance is observed at large polynomial degrees, indicating poor generalizability of the ML performance at those polynomial degrees. Second, the ML model, trained on this improved artificial dataset, achieves notably higher accuracy compared to the previous version. This reaffirms the significance of a well-representative training dataset for accurate testing.

Hence, variations in the distributions between training and test datasets can impact accuracy. It is essential to emphasize that this scenario differs entirely from encountering unexplored regions of the parameter space in the test set. To better understand this, let us consider an ML classifier trained on a dataset comprising 500 cat images, 490 horse images, and only 10 dog images, achieving an overall accuracy of 99%. However, the accuracy drops to 30% when classifying dog images. This discrepancy arises because most of the training focus was directed toward enhancing the classification of more abundant cat and horse images. Now, envision a test set consisting of 500 cat images and 500 dog images. Despite some dog images being present in the training set, a subpar performance on the test set is anticipated due to differing distributions compared to training set.

While this approach notably improves the ML model's performance in CCSN environments, the constrained parameter space within the training set could also partially restricts the applicability of the ML model to that specific parameter range.

In the final step, we enhance our ML model by incorporating real-world data into the training set. It is important to highlight that we do not exclusively rely on real data for training. This approach allows us to maintain variability in the angular distributions of neutrinos. The performance of this ML model is depicted by the blue curve in Fig. 1. Notably, it exhibits exceptionally high accuracy across all polynomial degrees. However, drawing from our previous experiences, we favor selecting $n = 2$ due to its strong potential for effective generalization to unseen data. Beyond its generalization capabilities, opting for $n = 2$ also brings the advantage of reduced computational intensity when implementing the LR model in CCSN and NSM

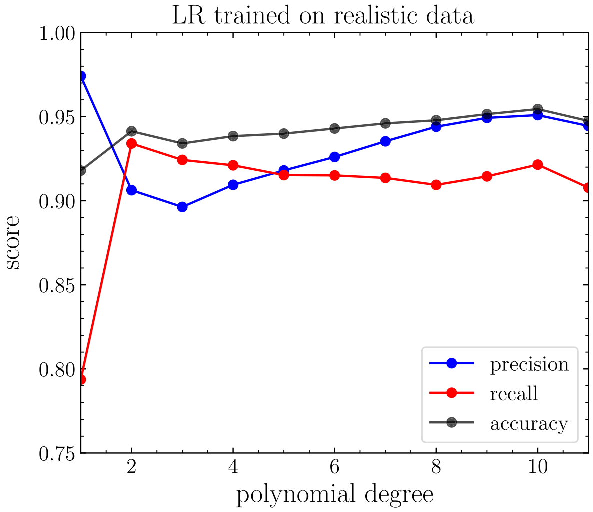

FIG. 2. The metric scores of the LR algorithm trained on a combination of the artificial and realistic datasets, as a function of the polynomial degree of the nonlinear transformations. It is worth noting that the precision and recall scores typically exhibit opposing trends, a phenomenon commonly referred to as the precision-recall tradeoff within the field of ML.

simulations on the fly. In addition, In Fig. 2, we present the comprehensive set of metric scores obtained from our improved LR model. It is worth highlighting that all these scores attain satisfactory values when the polynomial degree is set to 2.

TABLE II. A summary of the metric scores of ML algorithms trained on the combination of the realistic and artificial datasets, and then tested with the realistic data. Alongside each algorithm, one can find its corresponding accuracy score.

LR ($n = 2$) (94%)			
	Precision	Recall	F_1-score
No crossing	96%	95%	95%
Crossing	91%	93%	92%
KNN ($n = 3$) (98%)			
	Precision	Recall	F_1-score
No crossing	98%	99%	99%
Crossing	98%	97%	98%
SVM (97%)			
	Precision	Recall	F_1-score
No crossing	98%	98%	98%
Crossing	96%	97%	97%
Decision tree (99%)			
	Precision	Recall	F_1-score
No crossing	99%	99%	99%
Crossing	98%	98%	98%

In Table II, we present the performance results of our enhanced ML models, which were trained using a combination of both realistic and artificial datasets. The outcomes demonstrate that our ML models achieve very high performance metrics. Notably, these scores surpass those obtained by the ML model trained exclusively on artificial data, as shown in Table I of Ref. [69]. This improvement can be attributed to the fact that the artificial data includes noisy labels, as discussed in Ref. [69], which was identified as a primary source of inaccuracies once the ML model is tested on artificial data.

In all our calculations, we have primarily employed random data distribution between training and test datasets. While this is crucial to ensure having a robust ML model, we conducted also additional calculations where the training and testing processes were executed on datasets belonging to distinct time snapshots. We observed that these variations yielded similar results, confirming the robustness of our methods.

B. Performance of ML in the neutrino decoupling region

In the previous section, we assessed the overall performance of ML in the SN environment. It is important to note that FFCs are expected to be most impactful when they occur closer to the surface of the PNS, well within the SN post-shock zone. This can be attributed to two primary reasons. First, when FFCs happen in deeper SN regions, the νELN crossings can potentially be wider/deeper, which may result in more pronounced flavor conversions. Additionally, any flavor conversion occurring above the SN shock is not expected to have a noticeable impact on the CCSN dynamics [52], though can still impact the neutrino signal.

Considering this, in addition to evaluating our ML methods' overall performance, we specifically examined their performance in SN regions located well below the shock. To do this, we assessed the performance of our ML algorithms in SN zones where the radial distance is $\lesssim$100 km. As illustrated in Table III, our ML models effectively do a good job in capturing νELN crossings. The only exception is the recall score of the LR method, which is relatively low at around 50%, despite its good overall performance.

Fixing this issue involves lowering the threshold probability (denoted as p_c) for the LR algorithm, as discussed in Ref. [69]. As depicted in Fig. 3, significant improvement of the recall score can be achieved by reducing the LR threshold probability (here to $p_c = 0.3$). However, this comes at the cost of a reduction in the precision score, showcasing the traditional precision-recall tradeoff. While other ML algorithms exhibit very high scores, rendering them ideal for detecting the post-processing detection of

TABLE III. A summary of the metric scores of ML algorithms trained on the combination of the realistic and artificial datasets, in the SN post-shock region, namely at radii $\lesssim$100 km. Alongside each algorithm, one can find its corresponding accuracy score.

LR ($n = 2$) (96%)			
	Precision	Recall	F_1-score
No crossing	96%	100%	98%
Crossing	94%	49%	64%

KNN ($n = 3$) (100%)			
	Precision	Recall	F_1-score
No crossing	100%	100%	100%
Crossing	100%	99%	99%

SVM (99%)			
	Precision	Recall	F_1-score
No crossing	99%	99%	99%
Crossing	92%	90%	91%

Decision tree (100%)			
	Precision	Recall	F_1-score
No crossing	100%	100%	100%
Crossing	99%	99%	99%

νELN crossings, the LR algorithm remains valuable due to its ease of implementation in CCSN and NSM simulations. Therefore, we believe it is still worth considering the LR method and adjusting its threshold probability, despite the inherent tradeoff between precision and recall.

C. Detection of νELN-XLN crossings

So far, our discussion has focused exclusively on detecting νELN crossings. However, in a broader context, it is important to acknowledge that the angular distributions of ν_x and $\bar{\nu}_x$ can be different in CCSN and NSM environments. This difference becomes particularly pronounced when we account for the potential creation of muons at the core of these extreme astrophysical objects [82–84]. Consequently, to accurately identify the occurrence of FFCs under the most realistic conditions, we must shift our attention toward detecting νELN-XLN crossings rather than confining ourselves to the νELN ones.

The distinction between the detection of νELN-XLN crossings and νELN crossings presents several key differences. One of the most significant distinctions lies in the increased complexity of required information. In the context of our ML methods, this translates to an expansion in the number of essential features. Specifically, we now necessitate seven features instead of the previous three, namely α_{ν_e}, α_{ν_x}, $\alpha_{\bar{\nu}_x}$, F_{ν_e}, $F_{\bar{\nu}_e}$, F_{ν_x}, and $F_{\bar{\nu}_x}$, with $\alpha_{\nu_\beta} = n_{\nu_\beta}/n_{\nu_e}$.

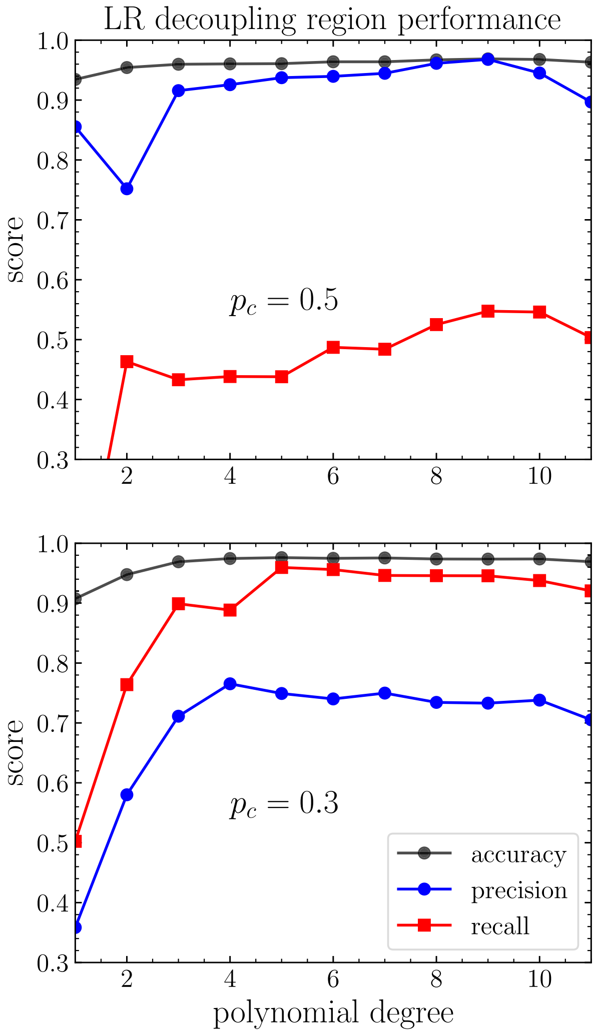

FIG. 3. The performance of the LR algorithm in the neutrino decoupling region, as a function of the polynomial degree of the nonlinear transformations. The LR is trained on a combination of the artificial and realistic datasets. As mentioned previously, it's worth noting that the precision and recall scores typically exhibit opposing trends due to precision-recall tradeoff. As one can see in the lower panel, one can enhance the recall score significantly by decreasing the threshold probability of LR, p_c, though with the price of a reduction in the precision score.

The increase of the number of features and the greater demand for information significantly contribute to an elevated classification error in this context. Notably, when ν_x and $\bar{\nu}_x$ exhibit disparities, the νELN-XLN profile can exhibit more intricate characteristics, e.g., it is conceivable that even multiple crossings occur.

In order to train our ML models, we use artificial distributions for neutrinos, given the fact that labeled data regarding the existence of νELN-XLN crossings are not available. In order to prepare our data, we consider α_{ν_e} $\alpha_{\nu_x(\bar{\nu}_x)}$ to be in the range of (0., 2.5) and (0., 3.), respectively. Also we assume an allowed maximum 40% difference between ν_x

TABLE IV. A summary of the metric scores of the ML algorithms for νELN-XLN crossing detection. Alongside each algorithm, one can find its corresponding accuracy score.

LR ($n = 2$) (88%)			
	Precision	Recall	F_1-score
No crossing	87%	89%	88%
Crossing	88%	86%	87%

KNN ($n = 3$) (88%)			
	Precision	Recall	F_1-score
No crossing	89%	88%	89%
Crossing	88%	89%	88%

SVM (88%)			
	Precision	Recall	F_1-score
No crossing	92%	84%	88%
Crossing	85%	93%	89%

Decision tree (87%)			
	Precision	Recall	F_1-score
No crossing	87%	87%	87%
Crossing	87%	87%	87%

and $\bar{\nu}_x$ quantities. This is consistent with the observation that the difference between ν_x and $\bar{\nu}_x$ should be subdominant in realistic simulations [82]. In addition, we keep in mind what we learned previously that the data should be enough representative of the realistic data. Thus, we also respect the hierarchy $F_{\nu_e} \lesssim F_{\bar{\nu}_e} \lesssim F_{\nu_x(\bar{\nu}_x)}$.

The performance of our ML models in detecting νELN-XLN crossings is presented in Table IV. Notably, the overall performance lags behind that of νELN crossing detection. This disparity can be attributed to the presence of intricate patterns governing the crossings and an increase in label noise.

IV. DISCUSSION AND OUTLOOK

Recent advancements have showcased the remarkable capabilities of ML in identifying the νELN crossings in the CCSN and NSM simulations [69]. In this study, we have propelled prior research in two pivotal and distinctive directions. Firstly, we have subjected ML models to the rigorous test of real-world data acquired from CCSN simulations, where the intricate problem of neutrino transport is addressed through the comprehensive Boltzmann equation. Secondly, we have expanded our ML techniques to encompass the detection of νELN-XLN crossings, accommodating situations where there may exist distinctions between ν_x and $\bar{\nu}_x$.

Using realistic CCSN data, we have demonstrated that the simpler models consistently outperform their more complex counterparts in the context of νELN detection when applied to previously unseen data. This provides a clear and compelling example of the bias-variance tradeoff within the domain of ML. Specifically, it was observed that the LR model performs most effectively when a polynomial transformation of degree $n = 2$ is applied, as opposed to the previously suggested degree of $n = 9$ which was based on artificial data. This underscores the importance of considering the complexity of models and their suitability for real-world data, highlighting that sometimes, a simpler approach can yield superior results.

We demonstrate a significant enhancement in model performance when utilizing artificial datasets by aligning the parameter space of the synthetic data with the realism expected in CCSN and NSM simulations. Specifically, we adhere to the hierarchy $F_{\nu_e} \lesssim F_{\bar{\nu}_e} \lesssim F_{\nu_x(\bar{\nu}_x)}$, which mirrors the conditions anticipated in these astrophysical events. This deliberate consideration in the preparation of artificial data results in model performance that rivals that of ML models trained on realistic data, at least within a certain parameter range.

We have further fortified our ML models by integrating real-world data into the training set, and this enhancement has yielded remarkable generic accuracy in out ML models. Based on our observations, we are inclined to assert that the LR model with a polynomial degree of $n = 2$ stands out as the optimal choice for detecting FFCs in CCSN and NSM simulations. This choice not only exhibits effective generalization capabilities for unseen data but also offers the distinct advantage of reducing computational overhead when deploying the LR model in real-time CCSN and NSM simulations.

We have also developed ML models to identify neutrino flavor crossings in the νELN-XLN distributions. This is particularly relevant because the angular distributions of ν_x and $\bar{\nu}_x$ can exhibit variations in CCSN and NSM environments. Unlike the simpler task of detecting νELN crossings, detecting νELN-XLN crossings introduces a higher level of complexity due to the need for more information. In the context of our ML methods, this complexity manifests as an increase in the number of essential features. This augmented feature set substantially contribute to an elevated classification error in this specific context.

In summary, our study significantly expands upon prior research, allowing for more confident utilization of ML methods in detecting FFCs. However, there remain crucial avenues for exploration. Specifically, our current analysis focuses on crossings occurring in the zenith angle (μ), when the angular distribution is integrated over ϕ_ν. Nevertheless, as demonstrated in previous references, a substantial fraction of νELN crossings may be exclusively in ϕ_ν, exhibiting nonaxisymmetric behavior [77]. Thus, it is imperative to develop ML techniques capable of capturing these nonaxisymmetric crossings. Closely related to this issue is the exploration of FFCs in rotating CCSN models [37], as the prominence of nonaxisymmetric features in such models could influence the performance of ML algorithms. Given the proven and remarkable versatility and effectiveness of ML in this context, implementing these measures will further improve the detection of FFCs in CCSN and NSM simulations.

ACKNOWLEDGMENTS

We would like to thank Georg Raffelt for useful discussions. We would also like to express our sincere gratitude to the Institute of Physics of Academia Sinica for their warm hospitality and support during the *Focus Workshop on Collective Oscillations and Chiral Transport of Neutrinos*, where the inception of this project took place. Their gracious hosting and collaborative environment played a pivotal role in shaping the foundation of our work. S. A. was supported by the German Research Foundation (DFG) through the Collaborative Research Centre “Neutrinos and Dark Matter in Astro- and Particle Physics (NDM),” Grant No. SFB-1258, and under Germany’s Excellence Strategy through the Cluster of Excellence ORIGINS EXC-2094-390783311. H. N. was supported by Grant-in-Aid for Scientific Research (23K03468) and also by the NINS International Research Exchange Support Program. The numerical simulations for CCSNe were carried out by using “K”, “’Fugaku”, and the high performance computing resources of “’Flow” at Nagoya University ICTS through the HPCI System Research Project (Project ID: 220173, 220047, 220223, 230033, 230204, 230270). We would also like to acknowledge the use of the following softwares: SCIKIT-LEARN [85], MATPLOTLIB [86], NumPy [87], SciPy [88], and IPython [89].

[1] A. Burrows and D. Vartanyan, Core-collapse supernova explosion theory, Nature (London) **589**, 29 (2021).

[2] H.-T. Janka, Explosion mechanisms of core-collapse supernovae, Annu. Rev. Nucl. Part. Sci. **62**, 407 (2012).

[3] F. Foucart, Neutrino transport in general relativistic neutron star merger simulations, arXiv:2209.02538.

[4] K. Kyutoku, M. Shibata, and K. Taniguchi, Coalescence of black hole–neutron star binaries, Living Rev. Relativity **24**, 5 (2021).

[5] J. T. Pantaleone, Neutrino oscillations at high densities, Phys. Lett. B **287**, 128 (1992).

[6] G. Sigl and G. Raffelt, General kinetic description of relativistic mixed neutrinos, Nucl. Phys. **B406**, 423 (1993).

[7] S. Pastor and G. Raffelt, Flavor oscillations in the supernova hot bubble region: Nonlinear effects of neutrino background, Phys. Rev. Lett. **89**, 191101 (2002).

[8] H. Duan, G. M. Fuller, J. Carlson, and Y.-Z. Qian, Simulation of coherent non-linear neutrino flavor transformation in the supernova environment. 1. Correlated neutrino trajectories, Phys. Rev. D **74**, 105014 (2006).

[9] H. Duan, G. M. Fuller, J. Carlson, and Y.-Z. Qian, Coherent development of neutrino flavor in the supernova environment, Phys. Rev. Lett. **97**, 241101 (2006).

[10] H. Duan, G. M. Fuller, and Y.-Z. Qian, Collective neutrino oscillations, Annu. Rev. Nucl. Part. Sci. **60**, 569 (2010).

[11] A. Mirizzi, I. Tamborra, H.-T. Janka, N. Saviano, K. Scholberg, R. Bollig, L. Hüdepohl, and S. Chakraborty, Supernova neutrinos: Production, oscillations and detection, Riv. Nuovo Cimento **39**, 1 (2016).

[12] M. C. Volpe, Neutrinos from dense: Flavor mechanisms, theoretical approaches, observations, new directions, arXiv:2301.11814.

[13] R. F. Sawyer, Speed-up of neutrino transformations in a supernova environment, Phys. Rev. D **72**, 045003 (2005).

[14] R. F. Sawyer, Neutrino cloud instabilities just above the neutrino sphere of a supernova, Phys. Rev. Lett. **116**, 081101 (2016).

[15] S. Chakraborty, R. S. Hansen, I. Izaguirre, and G. Raffelt, Self-induced neutrino flavor conversion without flavor mixing, J. Cosmol. Astropart. Phys. 03 (2016) 042.

[16] I. Izaguirre, G. Raffelt, and I. Tamborra, Fast pairwise conversion of supernova neutrinos: A dispersion-relation approach, Phys. Rev. Lett. **118**, 021101 (2017).

[17] F. Capozzi, B. Dasgupta, E. Lisi, A. Marrone, and A. Mirizzi, Fast flavor conversions of supernova neutrinos: Classifying instabilities via dispersion relations, Phys. Rev. D **96**, 043016 (2017).

[18] S. Abbar and H. Duan, Fast neutrino flavor conversion: Roles of dense matter and spectrum crossing, Phys. Rev. D **98**, 043014 (2018).

[19] S. Abbar and M. C. Volpe, On fast neutrino flavor conversion modes in the nonlinear regime, Phys. Lett. B **790**, 545 (2019).

[20] F. Capozzi, B. Dasgupta, A. Mirizzi, M. Sen, and G. Sigl, Collisional triggering of fast flavor conversions of supernova neutrinos, Phys. Rev. Lett. **122**, 091101 (2019).

[21] J. D. Martin, C. Yi, and H. Duan, Dynamic fast flavor oscillation waves in dense neutrino gases, Phys. Lett. B **800**, 135088 (2020).

[22] S. Abbar, H. Duan, K. Sumiyoshi, T. Takiwaki, and M. C. Volpe, On the occurrence of fast neutrino flavor conversions in multidimensional supernova models, Phys. Rev. D **100**, 043004 (2019).

[23] S. Abbar, H. Duan, K. Sumiyoshi, T. Takiwaki, and M. C. Volpe, Fast neutrino flavor conversion modes in

multidimensional core-collapse supernova models: The role of the asymmetric neutrino distributions, Phys. Rev. D **101**, 043016 (2020).

[24] F. Capozzi, G. Raffelt, and T. Stirner, Fast neutrino flavor conversion: Collective motion vs decoherence, J. Cosmol. Astropart. Phys. 09 (2019) 002.

[25] L. Johns, H. Nagakura, G. M. Fuller, and A. Burrows, Neutrino oscillations in supernovae: Angular moments and fast instabilities, Phys. Rev. D **101**, 043009 (2020).

[26] J. D. Martin, J. Carlson, V. Cirigliano, and H. Duan, Fast flavor oscillations in dense neutrino media with collisions, Phys. Rev. D **103**, 063001 (2021).

[27] I. Tamborra and S. Shalgar, New developments in flavor evolution of a dense neutrino gas, Annu. Rev. Nucl. Part. Sci. **71**, 165 (2021).

[28] G. Sigl, Simulations of fast neutrino flavor conversions with interactions in inhomogeneous media, Phys. Rev. D **105**, 043005 (2022).

[29] C. Kato, H. Nagakura, and T. Morinaga, Neutrino transport with the Monte Carlo method. II. Quantum kinetic equations, Astrophys. J. Suppl. Ser. **257** (2021) 55.

[30] T. Morinaga, Fast neutrino flavor instability and neutrino flavor lepton number crossings, Phys. Rev. D **105**, L101301 (2022).

[31] H. Nagakura, L. Johns, A. Burrows, and G. M. Fuller, Where, when, and why: Occurrence of fast-pairwise collective neutrino oscillation in three-dimensional core-collapse supernova models, Phys. Rev. D **104**, 083025 (2021).

[32] H. Sasaki and T. Takiwaki, A detailed analysis of the dynamics of fast neutrino flavor conversions with scattering effects, Prog. Theor. Exp. Phys. **2022**, 073E01 (2022).

[33] I. Padilla-Gay, I. Tamborra, and G. G. Raffelt, Neutrino flavor pendulum reloaded: The case of fast pairwise conversion, Phys. Rev. Lett. **128**, 121102 (2022).

[34] S. Abbar, F. Capozzi, R. Glas, H.-T. Janka, and I. Tamborra, On the characteristics of fast neutrino flavor instabilities in three-dimensional core-collapse supernova models, Phys. Rev. D **103**, 063033 (2021).

[35] F. Capozzi, S. Abbar, R. Bollig, and H. T. Janka, Fast neutrino flavor conversions in one-dimensional core-collapse supernova models with and without muon creation, Phys. Rev. D **103**, 063013 (2021).

[36] M. Delfan Azari, S. Yamada, T. Morinaga, W. Iwakami, H. Okawa, H. Nagakura, and K. Sumiyoshi, Linear analysis of fast-pairwise collective neutrino oscillations in core-collapse supernovae based on the results of Boltzmann simulations, Phys. Rev. D **99**, 103011 (2019).

[37] A. Harada and H. Nagakura, Prospects of fast flavor neutrino conversion in rotating core-collapse supernovae, Astrophys. J. **924**, 109 (2022).

[38] S. Abbar and F. Capozzi, Suppression of fast neutrino flavor conversions occurring at large distances in core-collapse supernovae, J. Cosmol. Astropart. Phys. 03 (2022) 051.

[39] O. Just, S. Abbar, M.-R. Wu, I. Tamborra, H.-T. Janka, and F. Capozzi, Fast neutrino conversion in hydrodynamic simulations of neutrino-cooled accretion disks, Phys. Rev. D **105**, 083024 (2022).

[40] I. Padilla-Gay, I. Tamborra, and G. G. Raffelt, Neutrino fast flavor pendulum. II. Collisional damping, Phys. Rev. D **106**, 103031 (2022).

[41] F. Capozzi, M. Chakraborty, S. Chakraborty, and M. Sen, Supernova fast flavor conversions in $1 + 1D$: Influence of mu-tau neutrinos, Phys. Rev. D **106**, 083011 (2022).

[42] M. Zaizen and H. Nagakura, Simple method for determining asymptotic states of fast neutrino-flavor conversion, Phys. Rev. D **107**, 103022 (2023).

[43] S. Shalgar and I. Tamborra, Supernova neutrino decoupling is altered by flavor conversion, Phys. Rev. D **108**, 043006 (2023).

[44] C. Kato and H. Nagakura, Effects of energy-dependent scatterings on fast neutrino flavor conversions, Phys. Rev. D **106**, 123013 (2022).

[45] S. Bhattacharyya and B. Dasgupta, Fast flavor depolarization of supernova neutrinos, Phys. Rev. Lett. **126**, 061302 (2021).

[46] M.-R. Wu, M. George, C.-Y. Lin, and Z. Xiong, Collective fast neutrino flavor conversions in a 1D box: Initial conditions and long-term evolution, Phys. Rev. D **104**, 103003 (2021).

[47] S. Richers, D. E. Willcox, N. M. Ford, and A. Myers, Particle-in-cell simulation of the neutrino fast flavor instability, Phys. Rev. D **103**, 083013 (2021).

[48] S. Richers, D. Willcox, and N. Ford, Neutrino fast flavor instability in three dimensions, Phys. Rev. D **104**, 103023 (2021).

[49] B. Dasgupta, Collective neutrino flavor instability requires a crossing, Phys. Rev. Lett. **128**, 081102 (2022).

[50] H. Nagakura and M. Zaizen, Time-dependent and quasisteady features of fast neutrino-flavor conversion, Phys. Rev. Lett. **129**, 261101 (2022).

[51] J. Ehring, S. Abbar, H.-T. Janka, and G. Raffelt, Fast neutrino flavor conversion in core-collapse supernovae: A parametric study in 1D models, Phys. Rev. D **107**, 103034 (2023).

[52] J. Ehring, S. Abbar, H.-T. Janka, G. Raffelt, and I. Tamborra, Fast neutrino flavor conversions can help and hinder neutrino-driven explosions, Phys. Rev. Lett. **131**, 061401 (2023).

[53] Z. Xiong, M.-R. Wu, S. Abbar, S. Bhattacharyya, M. George, and C.-Y. Lin, Evaluating approximate asymptotic distributions for fast neutrino flavor conversions in a periodic 1D box, Phys. Rev. D **108**, 063003 (2023).

[54] M. Zaizen and H. Nagakura, Characterizing quasisteady states of fast neutrino-flavor conversion by stability and conservation laws, Phys. Rev. D **107**, 123021 (2023).

[55] Z. Xiong, M.-R. Wu, and Y.-Z. Qian, Symmetry and bipolar motion in collective neutrino flavor oscillations, Phys. Rev. D **108**, 043007 (2023).

[56] D. F. G. Fiorillo and G. G. Raffelt, Flavor solitons in dense neutrino gases, Phys. Rev. D **107**, 123024 (2023).

[57] H. Nagakura, Global features of fast neutrino-flavor conversion in binary neutron star merger, Phys. Rev. D **108**, 103014 (2023).

[58] J. D. Martin, D. Neill, A. Roggero, H. Duan, and J. Carlson, Equilibration of quantum many-body fast neutrino flavor oscillations, Phys. Rev. D **108**, 123010 (2023).

[59] D. F. G. Fiorillo and G. G. Raffelt, Slow and fast collective neutrino oscillations: Invariants and reciprocity, Phys. Rev. D **107**, 043024 (2023).

[60] E. Grohs, S. Richers, S. M. Couch, F. Foucart, J. Froustey, J. Kneller, and G. McLaughlin, Two-moment neutrino flavor

transformation with applications to the fast flavor instability in neutron star mergers, arXiv:2309.00972.

[61] M. Shibata, K. Kiuchi, Y.-i. Sekiguchi, and Y. Suwa, Truncated moment formalism for radiation hydrodynamics in numerical relativity, Prog. Theor. Phys. **125**, 1255 (2011).

[62] C. Y. Cardall, E. Endeve, and A. Mezzacappa, Conservative $3+1$ general relativistic variable Eddington tensor radiation transport equations, Phys. Rev. D **87**, 103004 (2013).

[63] K. S. Thorne, Relativistic radiative transfer: Moment formalisms, Mon. Not. R. Astron. Soc. **194**, 439 (1981).

[64] B. Dasgupta, A. Mirizzi, and M. Sen, Simple method of diagnosing fast flavor conversions of supernova neutrinos, Phys. Rev. D **98**, 103001 (2018).

[65] S. Abbar, Searching for fast neutrino flavor conversion modes in core-collapse supernova simulations, J. Cosmol. Astropart. Phys. 05 (2020) 027.

[66] L. Johns and H. Nagakura, Fast flavor instabilities and the search for neutrino angular crossings, Phys. Rev. D **103**, 123012 (2021).

[67] S. Richers, Evaluating approximate flavor instability metrics in neutron star mergers, Phys. Rev. D **106**, 083005 (2022).

[68] H. Nagakura and L. Johns, New method for detecting fast neutrino flavor conversions in core-collapse supernova models with two-moment neutrino transport, Phys. Rev. D **104**, 063014 (2021).

[69] S. Abbar, Applications of machine learning to detecting fast neutrino flavor instabilities in core-collapse supernova and neutron star merger models, Phys. Rev. D **107**, 103006 (2023).

[70] H. Nagakura, K. Sumiyoshi, and S. Yamada, Possible early linear acceleration of proto-neutron stars via asymmetric neutrino emission in core-collapse supernovae, Astrophys. J. Lett. **880**, L28 (2019).

[71] H. Nagakura, K. Sumiyoshi, and S. Yamada, Three-dimensional Boltzmann-Hydro code for core-collapse in massive stars I. Special relativistic treatments, Astrophys. J. Suppl. Ser. **214**, 16 (2014).

[72] H. Nagakura, W. Iwakami, S. Furusawa, K. Sumiyoshi, S. Yamada, H. Matsufuru, and A. Imakura, Three-dimensional Boltzmann-Hydro code for core-collapse in massive stars II. The implementation of moving-mesh for neutron star kicks, Astrophys. J. Suppl. Ser. **229**, 42 (2017).

[73] H. Nagakura, K. Sumiyoshi, and S. Yamada, Three-dimensional Boltzmann-Hydro code for core-collapse in massive stars III. A new method for momentum feedback from neutrino to matter, Astrophys. J. **878**, 160 (2019).

[74] S. Furusawa, H. Togashi, H. Nagakura, K. Sumiyoshi, S. Yamada, H. Suzuki, and M. Takano, A new equation of state for core-collapse supernovae based on realistic nuclear forces and including a full nuclear ensemble, J. Phys. G **44**, 094001 (2017).

[75] H. Nagakura, S. Furusawa, H. Togashi, S. Richers, K. Sumiyoshi, and S. Yamada, Comparing treatments of weak reactions with nuclei in simulations of core-collapse supernovae, Astrophys. J. Suppl. Ser. **240**, 38 (2019).

[76] S. E. Woosley, A. Heger, and T. A. Weaver, The evolution and explosion of massive stars, Rev. Mod. Phys. **74**, 1015 (2002).

[77] H. Nagakura, T. Morinaga, C. Kato, and S. Yamada, Fast-pairwise collective neutrino oscillations associated with asymmetric neutrino emissions in core-collapse supernova, arXiv:1910.04288.

[78] R. Glas, H. T. Janka, F. Capozzi, M. Sen, B. Dasgupta, A. Mirizzi, and G. Sigl, Fast neutrino flavor instability in the neutron-star convection layer of three-dimensional supernova models, Phys. Rev. D **101**, 063001 (2020).

[79] M. Delfan Azari, S. Yamada, T. Morinaga, H. Nagakura, S. Furusawa, A. Harada, H. Okawa, W. Iwakami, and K. Sumiyoshi, Fast collective neutrino oscillations inside the neutrino sphere in core-collapse supernovae, Phys. Rev. D **101**, 023018 (2020).

[80] T. Morinaga, H. Nagakura, C. Kato, and S. Yamada, Fast neutrino-flavor conversion in the preshock region of core-collapse supernovae, Phys. Rev. Res. **2**, 012046 (2020).

[81] C. Yi, L. Ma, J. D. Martin, and H. Duan, Dispersion relation of the fast neutrino oscillation wave, Phys. Rev. D **99**, 063005 (2019).

[82] R. Bollig, H. T. Janka, A. Lohs, G. Martinez-Pinedo, C. J. Horowitz, and T. Melson, Muon creation in supernova matter facilitates neutrino-driven explosions, Phys. Rev. Lett. **119**, 242702 (2017).

[83] T. Fischer, G. Guo, G. Martínez-Pinedo, M. Liebendörfer, and A. Mezzacappa, Muonization of supernova matter, Phys. Rev. D **102**, 123001 (2020).

[84] G. Guo, G. Martínez-Pinedo, A. Lohs, and T. Fischer, Charged-current muonic reactions in core-collapse supernovae, Phys. Rev. D **102**, 023037 (2020).

[85] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, SCIKIT-LEARN: Machine learning in PYTHON, J. Mach. Learn. Res. **12**, 2825 (2011).

[86] J. D. Hunter, MATPLOTLIB: A 2D graphics environment, Comput. Sci. Eng. **9**, 90 (2007).

[87] C. R. Harris *et al.*, Array programming with NumPy, Nature (London) **585**, 357 (2020).

[88] P. Virtanen *et al.* (SciPy1.0 Contributors), SciPy1.0: Fundamental algorithms for scientific computing in Python, Nat. Methods **17**, 261 (2020).

[89] F. Pérez and B. E. Granger, IPython: A system for interactive scientific computing, Comput. Sci. Eng. **9**, 21 (2007).