Machine Learning Flags Fast Neutrino-Flavor Instabilities

John McGuigan & Sherwood Richers

University of Tennessee Knoxville, Department of Physics and Astronomy

Background on Neutrinos

- Core-collapse Supernovae Release on the order of 10^58 neutrinos, which is ~99% of the total gravitational energy released.[1]
- Neutrino Flavors Mix (swap); This mixing—in extreme environments can happen fast [5]— instabilities act on cm / ns [7]
- Missing this physics = wrong explosion + wrong nucleosynthesis [1]

SN 1987A light curve – photons arrived ≈ 3 h after 24 detected neutrinos, proving neutrinos escape first and carry the energy.[3] Image captured by[4]

Background on Flavor Dynamics

Equations of Motion (EOMs)

$$i\partial_t \rho = [H_{vac} + H_{matt} + H_{\nu\nu}]$$

Mean-field density-matrix evolution[5]

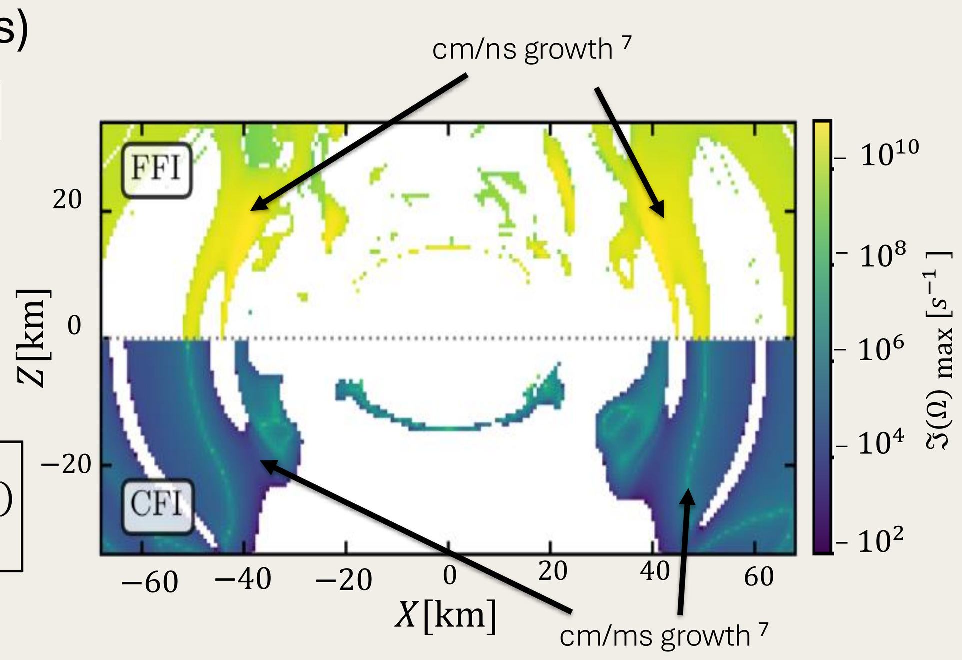
$$G(\vartheta) = n_{\nu_e}(\vartheta) - n_{\bar{\nu}_e}(\vartheta)$$

Electron Lepton Number (ELN) Distribution[6]

$$H_{\nu\nu}(\vartheta) = \mu \int d\Omega' (1 - \cos\theta_{\vartheta\vartheta'}) G(\vartheta)$$

Self-interaction potential from ELN angular spectrum[6] with the constant:

$$\mu = \sqrt{2} G_F n_\nu$$



Fast-flavor (yellow ~ cm/ns) vs collisional flavor (blue) growth rates in a 15 M☉ core-collapse model, 200 ms post-bounce (log scale). Figure adapted from[8]

Machine Learning Motivation

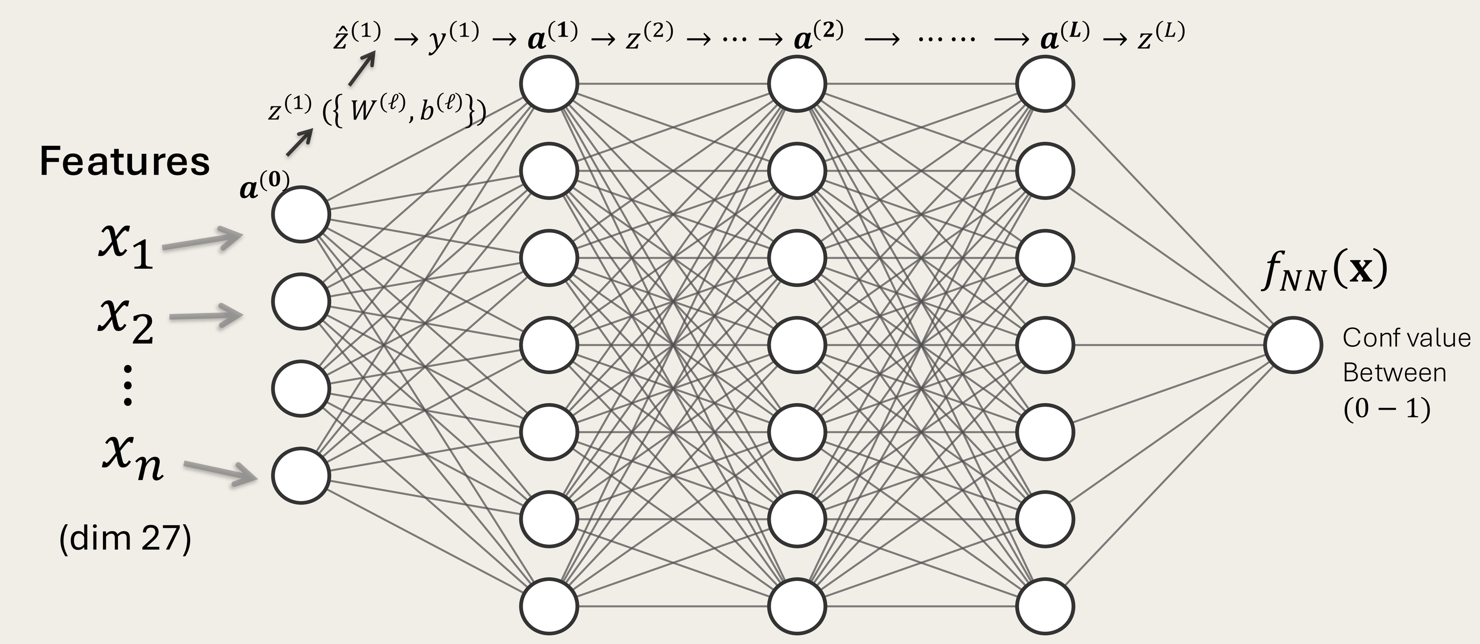
- A realistic 3-D core-collapse grid solved deterministically is computationally very difficult [11]
- Orders of magnitude beyond feasible simulation time [11]

Solution: Replace the solver with Machine Learning [9,10]

- Feed the network a compact feature vector for each cell[9]
- Network outputs a confidence P(Instability|x) much faster than solving the dispersion equation.
- We then set a cutoff confidence and use a simple rule: [9]

$$\boxed{P > P_{\text{cut}} \Rightarrow \text{Flag Instability}(1)}$$

Neural Network Architecture



Forward Pass

$$z^{(\ell)} = b^{(\ell)} + W^{(\ell)} a^{(\ell-1)}$$

$\ell + 1$

$$a^{(\ell)} = \text{ReLU}(y^{(\ell)})$$

$$y^{(\ell)} = \gamma^{(\ell)} \hat{z}^{(\ell)} + \beta^{(\ell)}$$

Justification for ReLU found at[14]

Batch Normalization

Batch Normalization equations from[12]

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} z_i^{(\ell)} \longrightarrow \frac{1}{m}\sum_{i=1}^{m}(z_i^{(\ell)} - \mu_B)^2 = \sigma_B^2$$

$$\hat{z}^{(\ell)} = \frac{z^{(\ell)} - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}}$$

Binary Cross Entropy (BCE) Loss

BCE Loss equation and weight decay[13]

$$\mathcal{L}(\theta) = -[y \ln \hat{y} + (1-y)\ln(1-\hat{y})] + \lambda \sum_{\ell} ||W^{(\ell)}||_2^2$$

$$\theta = \{W^{(\ell)}, b^{\ell}, \gamma^{(\ell)}, \beta^{(\ell)}\}_{\ell=1}^{L}$$

$$\theta_{t+1} = \theta_t - \eta\left(\frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon} + \lambda\theta_t\right)$$

Loss Optimizer (AdamW)

AdamW Optimizer Equations[13]

Mini-batch gradient

$$g_t = \eta \nabla_\theta \frac{1}{m} \sum_{i=1}^{m} \mathcal{L}(\theta; x_i, y_i)$$

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t$$
$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^{\odot 2}$$

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \qquad \hat{v}_t = \frac{v_t}{1-\beta_2^t}$$

Model Eval

Due to the large class imbalance that will be discussed, we use f1, precision, and recall for model performance evaluation.

$$\text{Precision} = \frac{TP}{TP + FP}$$

The fraction of cells flagged as fast-flavor-unstable that are indeed unstable.

$$\text{Recall} = \frac{TP}{TP + FN}$$

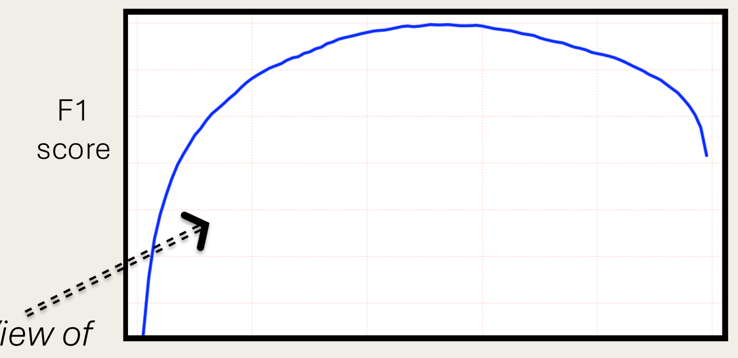
The fraction of all truly unstable cells that the model successfully identifies.

$$\text{F1} = \frac{2(\text{Precision})(\text{Recall})}{\text{Precision} + \text{Recall}}$$

Harmonic mean between precision and recall.[9]

How to choose the right cutoff?

These statistical performance metrics are "cutoff" dependent, which means the confidence threshold chosen, for instability assignment changes the performance.



Qualitative View of F1 vs Threshold

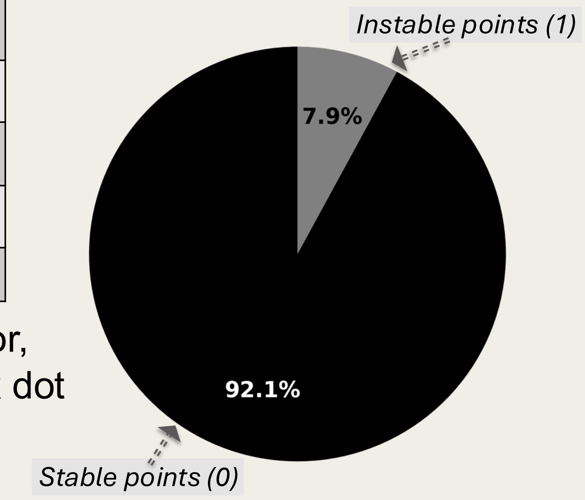
Training Data and Procedure

Features(x)	Neutrino Flux
Train/Test	80/20 (train/ Val)
Batch Size	4,096-16,384
Learning Rate	.003-.007
Hidden Layers	4-6
Hardware	1xV100 (16 GB)

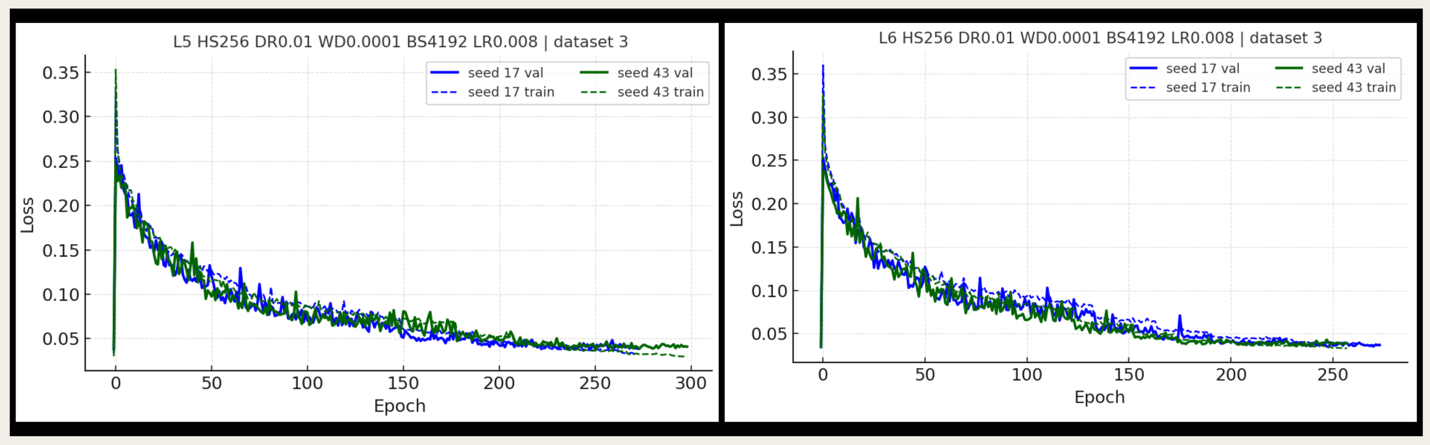
Training Data: 4 snapshots (CC-SN, one-flavor, random, NSM) of local and global neutrino-flux dot products.

Split: 80/20 Split stratified by data set then randomly split into batches.

Data Class Imbalance



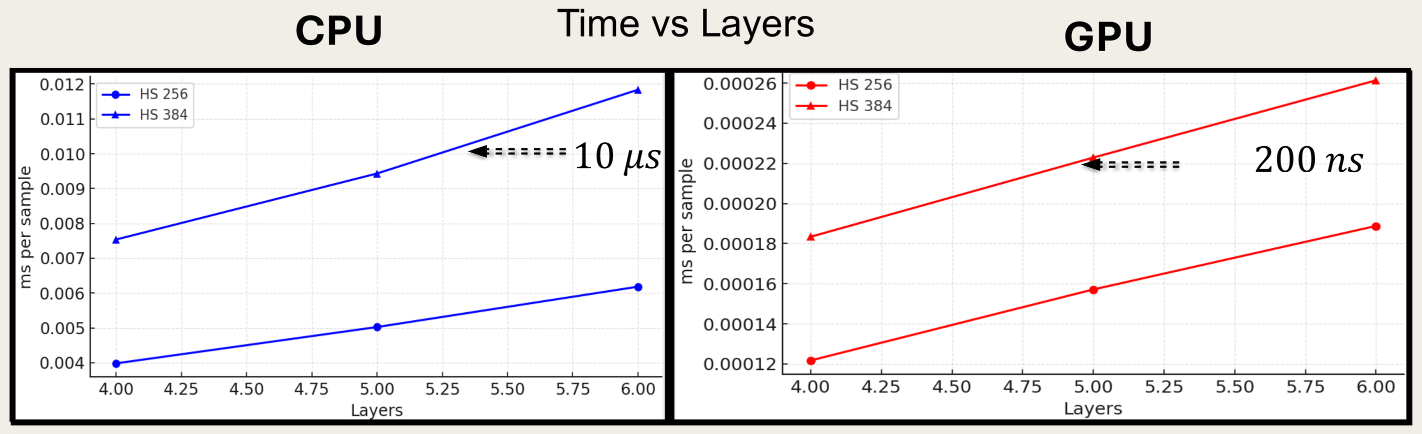
Model Convergence



Each plot shows clean model loss convergence for 2 separate random seeds (for generalization). This is to show effective regularization and optimization practices and no clear signs of overfitting.

Results and Conclusions

Latency time normalized by batch Time vs Layers



Current Evaluation Results

Layers	Hidden Size	Precision	Recall	F1
4	256	0.9722	0.9617	0.9669
4	384	0.9757	0.9498	0.9626
5	256	0.9787	0.9588	0.9687
5	384	0.9764	0.9586	0.9674
6	256	0.973	0.9625	0.9677
6	384	0.967	0.9634	0.9652

Comments:

- **Speed-aware design:** Model width and depth were deliberately capped; we prioritized real-time deployability over marginal gains from larger architectures.

Future and Current Work:

- **Faster Architecture:** the focus right now is continuing to improve inference for implementation in actual Simulations

THE UNIVERSITY OF TENNESSEE KNOXVILLE

References

1. Bethe, H. A., & Wilson, J. R. (1985). *Revival of a stalled super-nova shock by neutrino heating*. **ApJ**, 295, 14.
2. Volpe, M. C. (2022). *Neutrinos: from the r-process to the diffuse super-nova neutrino background*. **J. Phys. Conf. Ser.**, 2156, 012126.
3. Hirata, K. et al. (Kamiokande-II Coll.) (1987). *Observation of a neutrino burst from the super-nova SN 1987A*. **Phys. Rev. Lett.**, 58, 1490–1493.
4. European Southern Observatory. (2007, February 24). *SN 1987A in the Large Magellanic Cloud.*
5. Sigl, G., & Raffelt, G. (1993). *General kinetic description of relativistic mixed neutrinos*. **Nucl. Phys. B**, 406, 423–451.
6. Yi, C., Ma, L., Martin, J. D., & Duan, H. (2019). *Dispersion relation of the fast neutrino oscillation wave*. **Phys. Rev. D**, 99, 063005.
7. Ehring, J. et al. (2023). *Fast neutrino flavor conversion in core-collapse super-novae: a parametric 1-D study*. **Phys. Rev. D**, 107, 103034.
8. Johns, L., Richers, S., & Wu, M.-R. (2025). *Neutrino Oscillations in Core-Collapse Supernovae and Neutron-Star Mergers*. **Annu. Rev. Nucl. Part. Sci.**, in press, arXiv:2503.05959.
9. Abbar, S. (2023). *Applications of machine learning to detecting fast neutrino flavor instabilities in CCSNe and NS mergers*. **Phys. Rev. D**, 107, 103006.
10. Abbar, S., Harada, A., & Nagakura, H. (2024). *Machine-learning-based detection of non-axisymmetric fast neutrino flavor instabilities in CCSNe*. **Phys. Rev. D**, 111, 063077.
11. Mezzacappa, A. et al. (2020). *Physical, numerical, and computational challenges of modeling neutrino transport in core-collapse super-novae*. **Living Rev. Comput. Astrophys.**, 6, 4. § 6 pp. 108–115.
12. Ioffe, S., & Szegedy, C. (2015). *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. arXiv:1502.03167.
13. Loshchilov, I., & Hutter, F. (2019). *Decoupled Weight Decay Regularization (AdamW)*. arXiv:1711.05101.
14. Nair, V., & Hinton, G. E. (2010). *Rectified Linear Units Improve Restricted Boltzmann Machines*. Proc. ICML.