Energy Reconstruction with Semi-Supervised Autoencoders for Dual-Phase Time Projection Chambers

Ivy Li[1*], *Aarón* Higuera[1], *Shixiao* Liang[1], *Juehang* Qin[1], and *Christopher* Tunnell[1]

[1]Department of Physics and Astronomy, William Marsh Rice University, Houston, Texas, USA.

Abstract. This paper presents a proof-of-concept semi-supervised autoencoder for the energy reconstruction of scattering particle interactions inside dual-phase time projection chambers (TPCs), such as XENONnT. This autoencoder model is trained on simulated XENONnT data and is able to simultaneously reconstruct photosensor array hit patterns and infer the number of electrons in the gas gap, which is proportional to the energy of ionization signals in the TPC. Development plans for this autoencoder model are discussed, including future work in developing a faster simulation technique for dual-phase TPCs.

1 Introduction

Over the past decade, machine learning has played an increasingly important role in particle physics [1]. More machine-learning methods are being applied to high energy physics, with notable long-term success in collider experiments at the Large Hadron Collider [2–4] and burgeoning developments in noble element time projection chambers (TPCs) [5–8]. For example, machine-learning classification methods have sped up data quality selection and regression methods have increased robustness in the reconstruction of parameters such as interaction position or energy.

In particular, machine learning can enable more avenues in data analysis for liquid xenon (LXe) TPC experiments. Thus far, LXe TPCs have set leading sensitivities for weakly interacting massive particle (WIMP) dark matter [9–11] but are versatile in their applications including the study of neutrino production, rare nuclear decays, and other processes [12]. LXe TPCs have collected petabytes of information-rich, spatiotemporal data where machine learning can make a potentially significant impact. Machine learning can allow us to glean physically meaningful information from this data while reducing the computational resources necessary for such a feat. The next generation LXe experiments have and will continue to experience an increase in detector sensitivity and thereby the amount of data collected, which will require the software analysis tools to meet greater challenges.

For this work, we propose a proof-of-concept neural network to be applied on simulated XENONnT data. XENONnT is a direct detection dark matter experiment based on a dual-phase TPC located in the Laboratori Nazionali del Gran Sasso in Italy [13]. The TPC contains an ultra-pure LXe target of 6 tonnes which is designed to measure both scintillation and ionization signals from the particle interaction. The rare-event search conducted on these data signals has enabled XENONnT to set one of the leading sensitivities for the low-keV

*e-mail: il11@rice.edu

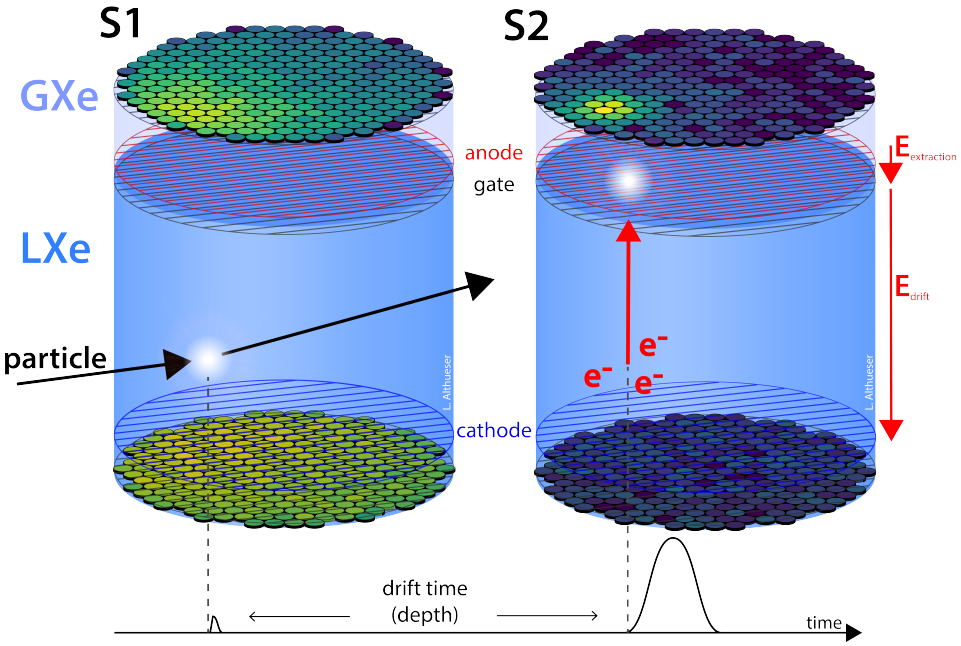


Figure 1: An incoming particle interacts with liquid xenon and causes a scintillation signal (S1) with energy which is proportional to the initial signal's number of photons. The electric field drifts electrons to the liquid-gas interface where an extraction electric field moves the electrons through to the gas gap. The proportion of electrons which reach the gas gap cause a brighter ionization signal (S2) with energy which is proportional to the number of electrons at the interaction point. Figure Credit: Lutz Althüser [15].

energy regime for WIMP dark matter. The XENON Collaboration reported the WIMP dark matter [11] and electronic recoil [14] analyses results on Science Run 0 data.

1.1 Energy Reconstruction

The dual-phase TPC experimental set up of XENONnT is shown in Fig. 1 where a typical scattering interaction is described. An incoming particle interacts with the LXe target and causes a scintillation signal (S1) with some number of photons n_γ. The electric field drifts electrons to the liquid-gas interface where an extraction electric field moves the electrons through to the gaseous xenon (GXe) layer. However, not all of the electrons reach this layer, as some electrons are captured by impurities in the LXe. The number of electrons that successfully reach the gas gap is described by the equation

$$N_e = N_{e_{int}} exp\left(-\frac{t}{\tau}\right) \tag{1}$$

where $N_{e_{int}}$ is the original number of electrons at the initial interaction point, N_e is the number of electrons that reached the gas gap, τ is the mean electron life time, and and t is the time since the initial interaction. The proportion of electrons which reach the gas gap cause a brighter ionization signal (S2) with some number of electrons at the interaction point $n_{e_{int}}$.

The light generated from the scintillation and ionization signals, along with any other light that is generated by other interactions, is measured by an array of photosensors above and below the detector. Putting all of this together, the total expected energy measurement of an event as described by a matching S1 and S2 pair, is described as

$$E_{total} = W\left(n_\gamma + n_{e_{int}}\right) = W\left(n_\gamma + \frac{n_e}{exp(-\frac{t}{\tau})}\right)$$

where W is the mean energy required to produce either a photon or an electron. A more detailed treatment of the scintillation and ionization signal modeling can be found in [16].

Accurate energy reconstruction of a particle interaction is crucial for a rare event search where there are not many event signals expected above background events. The information provided by multichannel spatiotemporal data from these photosensors enables the reconstruction of energy, position, and particle type (e.g. potential dark matter). Unlike the interactions in tracking experiments, the interactions in XENONnT do not have resolvable tracks. This reduces the amount of information that be can be used for energy reconstruction. Moreover, simulations are becoming more computationally expensive because of increasingly complex experimental setups. There are ongoing efforts to reduce this computational load through machine learning.

1.2 Simulated Dataset

The autoencoder model is trained on only spatial data of simulated ionization signals. We use the term hit pattern to describe the spatial data of each ionization signal, the amount of light seen by each photosensor in an array, summed over time for the duration of a given signal. Each hit pattern includes light measurements from each photosensor as measured in photoelectrons, totaling 494 photosensors across the top and bottom arrays. Simplifying the energy reconstruction problem, we task the autoencoder to infer the number of electrons in the gas gap from a given hit pattern. This number of electrons should be proportional to the S2 component of the particle interaction's total energy. Notably, inferring the number of electrons for low energy S2 signals of a few keV is particularly challenging, so any potential improvement we can attain through autoencoders is helpful.

To train this proof-of-concept autoencoder that can also infer the number of electrons, we simulated hit patterns of ionization signals with certain assumptions. We generated a dataset from simulated shallow S2 signals located just under the liquid-gas interface to avoid diffusion effects. In the simulation, we also assumed all electrons that reach the liquid-gas interface are successfully extracted by the electric field into the gas gap. For each hit pattern, there is an associated number of electrons (ground truth) used for hit pattern generation which we will use to evaluate the performance of our autoencoder.

From the generated hit patterns, the training and validation sets had 447,500 hit patterns each, while 100,000 hit patterns are set aside for testing performance as discussed in Sec. 3. Given a uniform distribution for the number of electrons in the gas gap, hit patterns were simulated ranging from 0 to 2,000 electrons for 494 photosensors. Simulations were generated using `wfsim` [17] v0.5.11 and `nestpy` v1.4.8 which include the Python bindings of the Noble Element Simulation Technique (`NEST`) package [16] library, NEST v2.2.1 [18].

2 Semi-Supervised Autoencoder

One of the most popular unsupervised neural network models is the autoencoder [19] which is able to encode higher-dimensional, complex input data into a lower-dimensional latent space

representation. Its encoder is followed by a decoder that is trained to reconstruct the original input data from this latent space representation. Autoencoders have been successfully applied to anomaly detection [20, 21] and data compression [22] problems in high energy physics.

We will be applying an autoencoder to infer the size of the ionization signal while simultaneously creating a potentially faster simulation technique. The traditional autoencoder is unsupervised but semi-supervised training has proven to be more effective during training when including labeled data [23, 24]. The current literature generally uses the term semi-supervised machine learning to refer to models trained on partially labeled datasets [25]. For this work, we refer to this autoencoder as a hybrid semi-supervised autoencoder in which part of its latent space is compared to labeled data (supervised) while the rest of the latent space is free to evolve during training without labeled data (unsupervised). This is to distinguish the model presented here from the future work discussed in Sec.4 in which this autoencoder will eventually be trained on partially labeled datasets with a mixture of simulated (labeled) and taken (unlabeled) data.

2.1 Architecture

The architecture of our hybrid semi-supervised autoencoder is shown in Fig. 2 in which we have one value of the latent space constrained through the loss function as described in Eq. 2 while the rest of the latent values can evolve freely during training. The model's loss function is a weighted sum of the mean square error (MSE) of the inferred number of electrons and the MSE of the reconstructed hit pattern:

$$\text{Loss} = (1-\beta)\ \text{MSE}(n_{e,\text{inferred}}, n_{e,\text{truth}}) + \beta\ \text{MSE}(p_{\text{reconstructed}}, p_{\text{truth}}) \tag{2}$$

The weights of the loss function are β and $(1-\beta)$ which allow for varying contribution from each component at different phases of the training. The value of β varies gradually by 0.04 per epoch (totaling 25 epochs of training and validation) starting from 0 until it reaches 1, i.e. focusing on inferring the number of electrons before focusing on reconstructing the input hit pattern. To ensure greater reconstruction accuracy, we trained for an additional set of 10 epochs where $\beta = 0.96$.

For the encoder and decoder, we adopt residual blocks as described from traditional residual networks [26]. For our model, each residual block implements a shortcut (skip) connection of one layer. The skip connections allow for the reconstruction of the hit patterns without scaling the photosensor measurements to reduce the dynamic range of the data. For both training and validation, the batch size is 500 hit patterns. The Adam optimizer was used, with a starting learning rate of 5×10^{-4} and in which the learning rate was reduced by a factor of 0.1 if the loss does not decrease after 5 epochs while training.

3 Reconstruction Results

The size of the latent space affects the resolution of the number of electrons inferred and the reconstruction accuracy of the hit patterns. A latent space that is too small will result in worse hit pattern reconstruction error due to the loss of information passing through the network, but too large latent spaces result in less interpretability for the non-constrained values. If the latent space is too small, the autoencoder will struggle to faithfully reconstruct some hit patterns but it is still possible to infer the number of electrons. In such a case, the autoencoder will reconstruct artifacts in the hit pattern that were not originally present, and in particular, the autoencoder can severely under- or overestimate the amount of light measured for hits occurring near the detector walls. We did not apply an exhaustive hyper-parameter search for

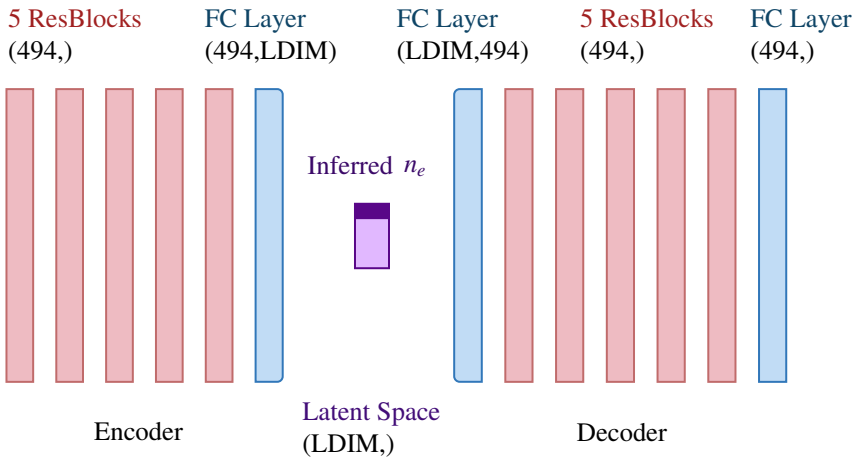
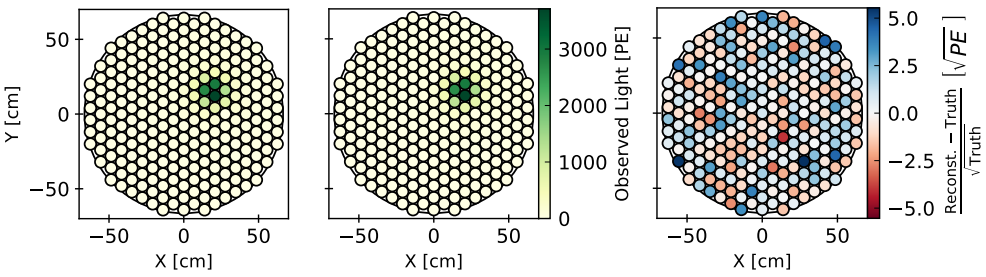


Figure 2: The autoencoder includes an encoder and a decoder containing 5 residual blocks (ResBlock) [26] as based on the one layer skip connection in residual networks and fully connected (FC) layers. The latent space is of size LDIM = 10 where the dimensions of the latent space are one of the hyperparameters tested. One value of the latent space is "constrained" as the number of electrons by the loss function while other 9 values of the latent space are allowed to evolve freely. The input is the original hit pattern with 494 photosensor measurements and the output is the reconstructed hit pattern.



(a) Reconstruction (b) Ground Truth (c) Reconst. Difference

Figure 3: This is an example of a well reconstructed event at the center of the top photosensor array. The left figure is the reconstructed hit pattern from the autoencoder. The middle figure is the original input top array hit pattern of a simulated signal. The right figure is the reconstruction difference between the original input and the reconstructed hit pattern.

the optimal size of the latent space, but for the purposes of this proceeding, we found that setting the size of the latent space to be 10 values avoided artifact reconstruction while also not being so large that the data was not meaningfully compressed. Further optimization of the latent space will be explored in the future.

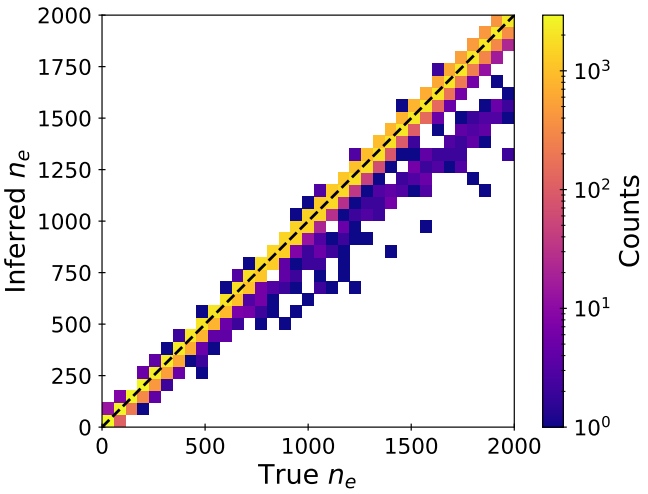


Figure 4: A 2-dimensional histogram showing the results of the inference on the test dataset after the autoencoder has been trained. The dotted line is an exact inference of the number of electrons in the gas gap.

From the test dataset, the autoencoder can simultaneously infer the number of electrons as shown in Fig.4 and reconstruct the hit pattern. One example of a well reconstructed event is shown in Fig. 3. Events that occur near the edge of the photosensor array have poorer reconstruction accuracy than the events that occur near the center of the photosensor array.

4 Discussion

In this work we demonstrated a proof-of-concept semi-supervised autoencoder for reconstructing the number of electrons while reconstructing the input photosensor data. Further work to optimize this autoencoder is ongoing, including better understanding the latent space size and the possibility of having the latent space be physically meaningful. The varied applications of autoencoders include simultaneous reconstruction of physical parameters and data compresssion. Outside of the reconstruction problem, we were able to reduce the latent space to be of size 10, demonstrating that the autoencoder is promising as a lossy data compression method.

We plan to test whether we could expand the variables constrained in the latent space to include the inference of the interaction position. This could provide an alternate method to correct our detector's light collection efficiency map which models the expected proportion of the total light observed by each photosensor for a given interaction in a given position. To ensure robustness, we have plans to train this autoencoder on a combination of simulated data and taken calibration data from XENONnT to test whether the inference of the number of electrons still works well outside of simulation.

Future efforts towards variational autoencoders [27] for fast simulation techniques [28, 29] will enable new avenues of data analysis for dual-phase TPCs. This may allow us to encode the input hit pattern into a latent space of distributions and use the decoder for the fast simulation of hit patterns.

5 Acknowledgments

We acknowledge support from the National Science Foundation under Award No. 2046549. We also acknowledge the use of these `python` [30] packages in this work: `matplotlib` [31], `numpy` [32], `pandas` [33], and `pytorch` [34].

References

[1] P. Shanahan et al. (2022), 2209.07559

[2] G. Karagiorgi, G. Kasieczka, S. Kravitz, B. Nachman, D. Shih, Nature Rev. Phys. **4**, 399 (2022)

[3] A. Radovic, M. Williams, D. Rousseau, M. Kagan, D. Bonacorsi, A. Himmel, A. Aurisano, K. Terao, T. Wongjirad, Nature **560**, 41 (2018)

[4] C. Grojean, A. Paul, Z. Qian, I. Strümke, Nature Rev. Phys. **4**, 284 (2022), 2203.08021

[5] E. Aprile et al. (XENON), Phys. Rev. D **108**, 012016 (2023), 2304.05428

[6] N. Shaheed, X. Chen, M. Wang, JINST **18**, T06002 (2023), 2303.05088

[7] P. Brás, F. Neves, A. Lindote, A. Cottle, R. Cabrita, E. Lopez Asamar, G. Pereira, C. Silva, V. Solovov, M.I. Lopes, Eur. Phys. J. C **82**, 553 (2022), 2201.05659

[8] M. Biassoni, A. Giachero, M. Grossi, D. Guffanti, D. Labranca, R. Moretti, M. Rossi, F. Terranova, S. Vallecorsa (2023), 2305.09744

[9] J. Aalbers et al. (LZ), Phys. Rev. Lett. **131**, 041002 (2023), 2207.03764

[10] Y. Meng et al. (PandaX-4T), Phys. Rev. Lett. **127**, 261802 (2021), 2107.13438

[11] E. Aprile et al. (XENON), Phys. Rev. Lett. **131**, 041003 (2023), 2303.14729

[12] E. Aprile et al. (XENON), Phys. Rev. C **106**, 024328 (2022), 2205.04158

[13] E. Aprile et al. (XENON), JCAP **11**, 031 (2020), 2007.08796

[14] E. Aprile et al. (XENON), Phys. Rev. Lett. **129**, 161805 (2022), 2207.11330

[15] L. Althüser, *Light collection efficiency simulations of the xenon1t experiment and comparison to data* (2017)

[16] M. Szydagis, N. Barry, K. Kazkaz, J. Mock, D. Stolp, M. Sweany, M. Tripathi, S. Uvarov, N. Walsh, M. Woods, JINST **6**, P10002 (2011), 1106.1613

[17] P. Gaemers, T. Zhu, J.R. Angevaare, D.R. García, K. Mizukoshi, J. Aalbers, A. Terliuk, H.S. Eißing, L. Althüser, E. Shockley et al., *Xenonnt/wfsim: v0.6.1* (2022), https://doi.org/10.5281/zenodo.6452974

[18] M. Szydagis, S. Andaloro, J. Balajthy, G. Block, J. Brodsky, J. Cutter, J. Huang, E. Kozlova, B. Lenardo, A. Manalaysay et al., *Noble element simulation technique* (2021), https://doi.org/10.5281/zenodo.4569211

[19] M.A. Kramer, AIChE Journal **37**, 233 (1991), https://aiche.onlinelibrary.wiley.com/doi/pdf/10.1002/aic.690370209

[20] O. Atkinson, A. Bhardwaj, C. Englert, V.S. Ngairangbam, M. Spannowsky, JHEP **08**, 080 (2021), 2105.07988

[21] L. Anzalone, S.S. Chhibra, B. Maier, N. Chernyavskaya, M. Pierini (2023), 2306.12955

[22] G.D. Guglielmo, F. Fahim, C. Herwig, M.B. Valentin, J. Duarte, C. Gingu, P. Harris, J. Hirschauer, M. Kwok, V. Loncar et al., IEEE Transactions on Nuclear Science **68**, 2179 (2021)

[23] J. Herrero-Garcia, R. Patrick, A. Scaffidi, JCAP **02**, 039 (2022), 2110.12248

[24] M. Kuusela, T. Vatanen, E. Malmi, T. Raiko, T. Aaltonen, Y. Nagai, J. Phys. Conf. Ser. **368**, 012032 (2012), 1112.3329

[25] W. Xu, H. Sun, C. Deng, Y. Tan, Proceedings of the AAAI Conference on Artificial Intelligence **31** (2017)

[26] K. He, X. Zhang, S. Ren, J. Sun, *Deep residual learning for image recognition* (2015), 1512.03385

[27] D.P. Kingma, M. Welling, *Auto-encoding variational bayes* (2022), 1312.6114

[28] A. Hariri, D. Dyachkova, S. Gleyzer (2021), 2104.01725

[29] J.C. Cresswell, B.L. Ross, G. Loaiza-Ganem, H. Reyes-Gonzalez, M. Letizia, A.L. Caterini, *CaloMan: Fast generation of calorimeter showers with density estimation on learned manifolds*, in *36th Conference on Neural Information Processing Systems* (2022), 2211.15380

[30] Python Software Foundation, *Python language reference, version 3.8.18*, http://www.python.org

[31] J.D. Hunter, Computing in Science & Engineering **9**, 90 (2007)

[32] C.R. Harris et al., Nature **585**, 357 (2020)

[33] The pandas development team, *pandas-dev/pandas: Pandas* (2023), https://doi.org/10.5281/zenodo.8364959

[34] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga et al., *Pytorch: An imperative style, high-performance deep learning library* (2019), 1912.01703