Evidential Reasoning with Expert-Guided Machine Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

Evidential reasoning aims to infer hidden causes from observed effects. In the context of fault detection, it is possible to trace the cause of anomalies by combining evidential reasoning with physical knowledge. However, experts may not have full knowledge of the physical systems, and data may not be large enough to learn from scratch. We develop an online evidential reasoning algorithm that blends machine learning with expert-provided physical knowledge about causal structure and functional form. The expert first represents possible causes and effects in the structure of a Bayesian Network, then provides physics-informed priors about the model relating failure modes to effects, allowing inference in the absence of strong supervision. As data are sampled from the physical system, predictions are generated using a quasi-Bayesian mixture of the expert's judgment and a data-driven estimate. With simulated datasets, we evaluate the conditions under which the system converges to correct causal inferences under weak supervision, and small amounts of strong supervision from the expert. We find that the approach is able to make accurate inferences with little or no data unless the expert's physical model is very incorrect or the signal to noise ratio across error modes is small.

1 Introduction

Evidential reasoning can be used to understand physical systems, such as predicting the state of a machine, or identifying the root cause of machine failure (i.e, root cause analysis or RCA). Measurable quantities (e.g., sound, vibration, force) captured during operation of many machines will change over time due to component degradation, corrosion, or overheating, as shown in Figure 1. The maintenance technician or engineer usually has an expectation that operation will degrade after a specified number of cycles, and will take notice if the pattern differs from expectation. Before the system fails, a critical task of a maintenance technician or engineer is to study sensor readings (signals) to determine whether the system is healthy, and if not, identify and remedy the cause of the problem. In order to minimize the impact of unanticipated downtime, maintenance is often performed on a fixed schedule. This approach, however, often has the side-effect of performing maintenance before it is necessary, resulting in excess cost of operation. In other situations, anomalous behavior may indicate imminent failure, resulting in loss of productivity depending on the task the machine is performing. Expert knowledge of the machine dynamics, and expert labeling of data when failure is anticipated, may be used to build fault detection algorithms to improve situational awareness and optimize machine operation [1, 2, 3].

Since machine failure is costly, algorithms would ideally be able to detect faults before they occur, without supervision. An online RCA system should actively monitor system health without significant involvement from the operator, maintenance technician, or engineer. It is also possible that the operator, technician, or engineer does not have a complete understanding of the system's workings for

Submitted to 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

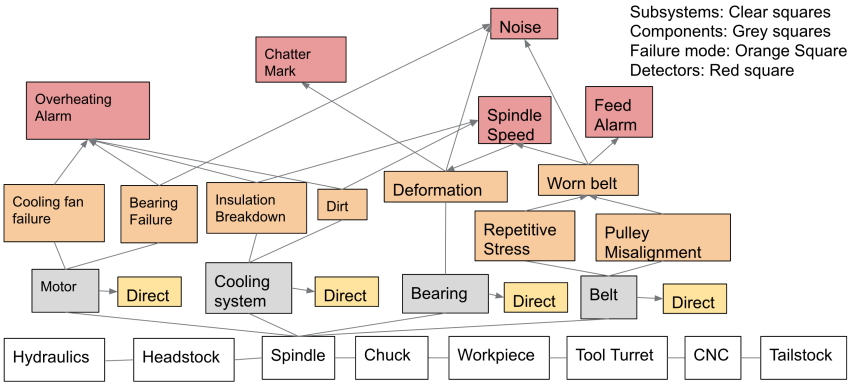


Figure 1: An expert-provided Bayesian Network showing failure modes and possible effects for a spindle in a computer numerical control (CNC) turning machine. Based on an analysis from [4].

complex machinery. Nevertheless, an expert still explicitly or tacitly uses physical and engineering knowledge about the system to detect problems before they occur. That physical knowledge might be represented in the form of an ordinary differential equation (ODE) or partial differential equation (PDE). Recent research has used neural networks to leverage this kind of prior physical knowledge [5, 6, 7].

Leveraging expert knowledge about the potential causes and physical functional relationships underlying machine operation, we propose an approach that: 1) Incorporates expert-provided physical knowledge into the modeling of machine failure modes, 2) uses an expert-provided Bayesian Network to simplify the evidential reasoning inference problem, and 3) combines the expert's judgment with a data-driven model to solve the zero-shot root cause analysis problem without experiencing machine failure.

2 Related Work

Inclusion of experts in machine learning has gained interest in recent years. Abdollahi *et al.* [8] use expert guidance to optimize the selection of parameter spaces and hyper-parameters in the challenging task of 3D printing soft materials. Gennatas *et al.* [9] use expert decision rules as priors in a machine learning model. Xu *et al.* [10] use misspecified expert functions in the form of pairwise comparisons for optimization.

Application of machine learning models for RCA is typically preceded by established physics-based approaches for feature selection and engineering, such as use of a signal's frequency spectrum from the Fourier transform. There are many features in the time and frequency domain that have proved useful, including the energy ratio (ER), side-band index (SI), sideband level factor (SLF), root mean squared error, energy operator, skewness, kurtosis, crest, energy ratios of wavelet coefficients, ensemble empirical mode decomposition, wavelet kernel local fisher discriminant analysis, among others [11, 12].

Another important tool for RCA is the use of Bayesian Networks. Kaufman *et al.* [13] propose a method for estimating the Markov blanket for some variables in large networks, which improves subsequent Bayesian inference using Gibbs sampling, and Louizos et al. [14] use variational auto-encoders to represent latent confounders as latent variables to estimate causal effects. For RCA, Feng *et al.* [11] use a local connection network with normalized sparse auto-encoders, and Hong *et al.* [15] use Gaussian Process regression to model bearing degradation, where the rMSE of the GP fit on normal data and tested on new data is used to classify failure one day in advance.

3 Using expert-provided Bayesian Networks for root cause analysis

Here we describe a general framework for expert-provided Bayesian Networks necessary for evidential reasoning. Assume there are S sensors $s_1, s_2, \dots s_S$. For any time interval $[0, t]$, we are interested in the quantity $p(\mathbf{m}|s_{10}, s_{20}, ...s_{1t}, s_{2t}, \dots s_{St})$ where $\mathbf{m}$ is a set of causes (normal or failure modes) with categorical input space $\mathcal{M}$, and s_{it} denotes the measurements of sensor i at time t. If both the sensors and failure modes are observed, one can train physics-based neural network models $\{{\psi_i}^{\mathbf{m}}\}$(see next section for ψ) to fit the sensor i's readings to a pattern, according to specific failure modes.

The Bayesian Network captures the fact that it is unlikely that all failure modes affect all sensors. The causal relationships between the hidden states $\mathbf{m}$ and the observed variables s are given by the experts with a Bayesian Network $\mathcal{G}$. Using the Bayes Net we can factorize the joint distribution $p(\mathbf{m}, s_{10}, s_{20}, ...s_{1t}, s_{2t}, \dots s_{St})$ into a product of conditional distributions based on the parents (Pa) of each sensor $p(\mathbf{m}, s_{10}, s_{20}, ...s_{1t}, s_{2t}, \dots s_{St}) = \prod_{i=1}^{S} p(s_{i0}, s_{i1}, ...s_{it}, |Pa(s_i) \subseteq \mathbf{m})) \cdot p(\mathbf{m})$.

With this factorization it is possible to do likelihood-based inference if robust likelihoods are available for the conditional distributions of $p(s_{i0}, s_{i1}, ...s_{it}, |Pa(s_i) \subseteq \mathbf{m}))$. In our model we impose the assumption that the distribution of sensors with respect to the fault vector $\mathbf{m}$ satisfies a Gaussian distribution, such that $p(s_{i0}, s_{i1}, ...s_{it}|\mathbf{m}) \sim \prod_{j=0}^{t} \mathcal{N}({\psi_i}^{\mathbf{m}}(j), \sigma_i^2)$. A sequence is predicted based on the neural network ${\psi_i}^{\mathbf{m}}$ with input $[0, t]$ and compared with each $s_{ij}, j \in [0, t]$ to see how similar two functions behave in terms of mean squared error. Our goal is to: 1) Estimate $\tilde{\mathbf{m}}$ by the conditional probabilities, and 2) update ${\psi_i}^{\mathbf{m}}$ based on streaming data.

The approach works as follows:

1. Construct Bayesian Network $\mathcal{G}$ from the expert's engineering analysis of the system
2. For S sensors $s_{10}, s_{20}, ...s_{1t}, s_{2t}, \dots s_{St}$ and hidden state $\mathbf{m}$, factorize their joint distribution according to $\mathcal{G}$

$$p(\mathbf{m}, s_{10}, s_{20}...s_{1t}, s_{2t}, \dots s_{St}) \propto \prod_{i=1}^{S} p(s_{i0}, s_{i1}, ...s_{it}|Pa(s_i) \subseteq \mathbf{m})p(\mathbf{m}) \tag{1}$$

   where $Pa(s_i) \subseteq \mathbf{m}$ are the parents of sensor i in $\mathcal{G}$
3. Initialize ${\psi_i}^{\mathbf{m}}$ with an expert's assessment of the functional causal model for each sensor i and each combination of its parents
4. For each of the sensors and failure mode combinations, label the sensor data $\mathbf{s}_t$ observation with:

$$\tilde{\mathbf{m}} = \underset{\mathbf{m} \in \mathcal{M}}{\arg\max} \Big[\prod_{i=1}^{S} p(s_{i0}, s_{i1}, ...s_{it}|Pa(s_i) \subseteq \mathbf{m})p(\mathbf{m}) \Big] \tag{2}$$

$$= \underset{\mathbf{m} \in \mathcal{M}}{\arg\max} \Big[\prod_{i=1}^{S} \prod_{j=0}^{t} \mathcal{N}\big(s_{ij}; {\psi_i}^{\mathbf{m}}(j), \sigma_i^2\big)p(\mathbf{m}) \Big] \tag{3}$$

5. Label the data vector $\mathbf{s}_t$ with the label $\tilde{\mathbf{m}}$, which indicates the failure modes that cause the machine to produce this data. (Optional strong expert supervision: Ask the expert to verify the accuracy of the label by inspecting the machine.)
6. With $n^{\tilde{\mathbf{m}}}$ as the number of labels for class $\tilde{\mathbf{m}}$, use learning rate $0 \leq \alpha(n^{\tilde{\mathbf{m}}}) \leq \alpha(n^{\tilde{\mathbf{m}}}+1) \leq 1$ to update ${\psi_i}^{\tilde{\mathbf{m}}}$ using backpropagation:

$$\psi_{i,\text{updated}}^{\tilde{\mathbf{m}}} = \alpha(n^{\tilde{\mathbf{m}}})\psi_i^{\tilde{\mathbf{m}}} + (1 - \alpha(n^{\tilde{\mathbf{m}}}))\psi_{i,\text{expert}}^{\tilde{\mathbf{m}}} \tag{4}$$

4 Integrating expert-provided physics into learning time-series functions

Frequently, the sensor readings and signals follow some ODEs or PDEs that represent the underlying physics. Thus, physicists and machinists are essentially solving the parameters of the equations that

govern the patterns of the data, under various anomaly scenarios. In the current application, sensors may pick up harmonic oscillations or follow ODE equations that result from normal and abnormal machine operation and return a physics-based time series data. We want to aid the experts by training the neural network ψ on a series of time-dependent readings for each sensor and failure mode.

We leverage recent advances in physics-based modeling to model those time series with neural networks that require little or no data for accurate estimation. Specifically, we use *physics-constrained loss functions* to approximate the solution to an ODE/PDE using a neural network [16, 17, 18]. The approach learns a hidden function $u(t, x)$ such that [18]:

$$\frac{\partial u}{\partial t} + \mathcal{N}[u; \lambda] = 0 \tag{5}$$

where $\mathcal{N}[u; \lambda]$ is an arbitrary non-linear differential operator parameterized by λ and $x \in \mathbb{R}^D$. For example, the 1D Berger's equation is $\mathcal{N}[u; \lambda] = \lambda_1 u \frac{\partial u}{\partial x} - \lambda_2 \frac{\partial^2 u}{\partial^2 x}$. To enforce this constraint, the approach defines $f(t, x) = \frac{\partial u}{\partial t} + \mathcal{N}[u; \lambda]$ and adds $\sum_{t=1}^{T} f(t, x)^2$ as a regularizer, where $\frac{\partial u}{\partial t}$ and $\mathcal{N}[u; \lambda]$ are calculated using auto-differentiation on the fitted neural network.

In our approach, we treat this differential equation information as imperfect prior knowledge provided by the expert. Model predictions are then based partly on a data-driven model, and partly on the expert's guess about the system's physical dynamics ψ_{expert} (given by the expert in the beginning). To capture the data-driven component, we use a neural network with two hidden layers and sine activation functions [19] to predict physical quantities of interest at each timestep. Prior research [17] finds that such an architecture can learn to reproduce not only the positions in a fluid flow problem, but also the governing PDE equations. The use of sinusoidal activation functions in physics-informed neural networks was rigorously studied in the SIREN architecture to deal with the infinite local minima problem [19], with the ability to solve boundary value problems including Poisson, Helmholtz, and wave equations.

5 Summary of the role of the expert

The expert plays a key role in several stages of the proposed approach. Here we provide an overview of the expert's role in each part of the algorithm.

Bayes Net. The expert provides a Bayesian network $\mathcal{G}$ for the algorithm. For failure mode analysis, this often takes the form of a bipartite graph, where the first part of the graph consists of the failure modes F, and the second part includes the observable sensors S. The directed edges E link failure modes to some subset of the sensors. Experts should be able to construct such a graph based on their engineering knowledge of the system, particularly if they are experienced with the machine.

Assessment of the causal model. The expert provides an assessment of how the sensor observations behave given different configurations of failure modes. For example, in the case of a turning or cutting machine, there is usually a fundamental frequency at which the system vibrates, and chattering or other problems with the system add new frequencies. This additivity of signals can be used to simplify the combinatorial problem of specifying the conditional distribution of each sensor for all combinations of its parents. For the algorithm to be useful, the expert needs to be able to give precise enough estimates such that the difference between the signals across normal and failure modes is larger than the noise variance in the system.

Estimation of failure modes distribution. Although not used in our current approach, the expert can also provide base rate (unconditional) guesses about the probability of different failure modes. As the system is continuously fitting data-driven models and weakly classifying hidden states as failure modes at each timestep, it is possible for these estimates to also be data driven.

Strong supervision. We assume the expert can inspect the machine to determine whether the algorithm gives correct classification of the failure modes. This might be visual inspection, diagnostic testing, or other relevant engineering analyses. Strong supervision is particularly important when the expert is inaccurate in the initial functional causal model estimates (or is unable to provide estimates). If a previously unobserved malfunctioning cause is detected, the expert can add additional nodes (F, S) and edges (E) to refine the Bayesian network $\mathcal{G}$. The ideal scenario is that expert can give strong supervision with the system; that is, the expert can detect all misclassified labels and provide correct feedback.

6 Experiments

We examine the ability of the proposed approach to recover time series functions and accurately classify error modes with little or no data, focusing on a) the amount of data required, b) noise in the data generating process, c) bias in the expert's judgments, and d) strong supervision. In a real machine deployed in a manufacturing environment, the Bayesian Network and functional relationships can be quite complex. For the purpose of initial testing, we use a simple Bayesian network with two sensors $S = \{s_1, s_2\}$ and three failure modes $F = \{f_1, f_2, f_3\}$. The directed edges depict the failure modes that are parents of each sensor, as shown by Figure 2. The probability of each failure mode occurring is characterized by independent Bernoulli random variables with probability p of being active during any time window $T = [T_1, T_2]$.

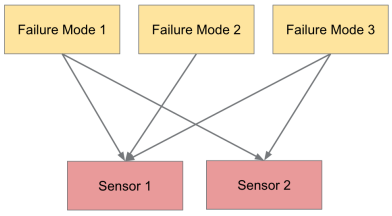


Figure 2: Bayesian Network represented as Bipartite Graph with sensors $S = \{S1, S2\}$ and Failure Modes $F = \{F1, F2, F3\}$.

Each sensor outputs either the normal operation signal (no failure modes are active) or a sum of the signals produced by the active failure modes. This means that each element $m \in \mathbf{m}$ is also $m \in F$. Let s_{iT} denote the signal observed by sensor i during the time interval T, then we have

$$s_{iT} = \begin{cases} \psi_i^0(T), & \text{if no fault} \\ \sum_{m \in F} \mathbb{1}(m = 1)\psi_i^m(T), & \text{if fault occurs} \end{cases} \tag{6}$$

where ψ_i^m is the signal produced when fault m occurs, ψ_i^0 denotes the normal functioning signal.

Before data are collected from the machine, experts provide an assessment of the functional causal model for each sensor for all combinations of its parent failure modes, with a maximum of $2^{|F|}$ assessments. Ideally, given an arbitrary time window t, the estimate for sensor i follows $Pr(|\hat{s}_{iT} - s_{iT}|) > \epsilon) < \delta$, and as we have more data collected and with correct expert's intervention, we want our estimated $\hat{s}_i$ to get closer to the true s_i.

6.1 Experiment 1: Time Windows

First we consider the case where the expert provides nearly accurate estimates of the signal for each failure mode in the presence of small amounts of noise. We examine the amount of data required for the algorithm to accurately learn the sum of periodic functions. Our key variable manipulated in the experiment is the time window size, representing the number of observations collected after the failure occurs. Ideally this number will be small so that faults can be identified early and machine down-time will be small.

Data Generation. Our assumption is that the expert already knows how s_1 and s_2 behaves during normal operation ($\sin(t)$ and $\cos(t)$, respectively). When a failure occurs, the two sensors s_1 and s_2 are generated with the same periodic functions, according to Table 1. This simplification does not affect the generality of the approach, because different neural networks are still required to learn the periodic functions for each combination of failure modes.

We assume all combinations of errors occur with equal probability. Sensor measurements are collected for 1 observation per 0.2 seconds for a time window of $T \times 0.2$ seconds. The experiment varies the length of time T for collection of sensor data. Our primary outcomes are the accuracy of failure mode classification and resulting average MSE's between the modeled functions and data.

Table 1: Parameters for Data Generation in Experiment 1

Fault Mode	Expert's Assumption	Actual Equation
1	$2\cos(5t)$	$2.2\cos(5t) + \epsilon_t \sim N(0, 0.01)$
2	$-2.4\cos(1.9t)$	$-0.7\cos(2.2t) + \epsilon_t \sim N(0, 0.01)$
3	$0.6\cos(0.5t)$	$0.7\cos(0.5t) + \epsilon_t \sim N(0, 0.01)$

Training. We train independent SIREN and LSTM models for each combination of sensors and fault modes. For each of the SIREN models, we initialize 2 internal layers of 256 nodes with sine activation and 1 output sine layer. For LSTM we apply 2 internal layers of 256 nodes with Relu activation and 1 output linear layer. We also use expert's interventions when the model misclassifies the labels. We repeat this experiment for 25 iterations for each of the failure modes.

Results. Figure 3 and 4 show the MSE for s_1 and Table 2 reports the accuracy of root cause classifications. We find that a small time window causes the classifier to slowly converge, as shown in Figure 3 and 4. The accuracy improves as the time window expands, but the small time window does not affect the classification accuracy too much, reflecting the expert's initial accuracy. Function recovery is possible with around 10 to 15 iterations, regardless of the time window sizes.

Table 2: Accuracy of model classification for failure modes, with varying the window size during each time the mode fails.

Accuracy	F1	F2	F3	F1,F2	F1,F3	F2,F3	F1,F2,F3
WS = 5, LSTM	1.00	0.96	1.00	0.88	1.00	0.88	0.60
WS = 15, LSTM	1.00	0.96	1.00	0.88	1.00	0.92	0.72
WS = 25, LSTM	1.00	0.96	1.00	0.76	1.00	0.88	0.84
WS = 35, LSTM	1.00	0.96	1.00	0.92	1.00	0.80	0.76
WS = 45, LSTM	1.00	0.96	1.00	0.92	0.96	1.0	0.84
WS = 5, SIREN	1.00	0.88	0.96	0.96	0.88	0.96	0.72
WS = 15, SIREN	0.96	0.96	1.00	0.84	0.92	1.00	0.92
WS = 25, SIREN	1.0	1.00	1.00	1.00	1.00	0.96	0.92
WS = 35, SIREN	0.96	0.96	1.00	0.92	1.00	1.00	1.00
WS = 45, SIREN	1.00	1.00	1.00	1.00	1.00	0.96	0.96

6.2 Experiment 2: Noise variance

Here we examine how noise in sensor measurements affects function recovery and failure mode classification. As shown in Table 3, the approach is the same as Experiment 1 but with a fixed window size of $T = 25$ and different noise variances $\sigma_t^2 \in \{0.1, 0.2, 0.4, 0.6, 0.8, 1\}$.

Table 3: Parameters for Data Generation in Experiment 2.

Fault Mode	Expert's Assumption	Actual Equation
1	$2\cos(5t)$	$2.2\cos(5t) + \epsilon_t \sim N(0, \sigma_t^2)$
2	$-2.4\cos(1.9t)$	$-0.7\cos(2.2t) + \epsilon_t \sim N(0, \sigma_t^2)$
3	$0.6\cos(0.5t)$	$0.7\cos(0.5t) + \epsilon_t \sim N(0, \sigma_t^2)$

Results. We plot the MSE in fitting s_1 between the predicted and actual data as shown by Figures 5 and 7. We also give the failure mode classification accuracy of the model in Table 4. We see that the classification accuracy drops as the sensor observations become more noisy. For larger variances, the MSEs vacillate instead of converging to their lower bound, meaning that the model fails to recover the underlying mean function. Note that this happens with a σ_t^2 of approximately 0.4, where accuracy drops significantly across both LSTM and SIREN models.

6.3 Experiment 3: Expert accuracy and weak supervision

There are two cases where the RCA model's performance is potentially affected by the experts: 1) the experts may not have full knowledge of the underlying physical equations for the failure modes,

Table 4: Accuracy of model classification for failure modes, with varying the variance of the observations.

Accuracy	**F1**	**F2**	**F3**	**F1,F2**	**F1,F3**	**F2,F3**	**F1,F2,F3**
$\sigma_t = 0.1$, LSTM	1.00	0.96	1.00	0.96	1.00	1.00	0.96
$\sigma_t = 0.2$, LSTM	1.00	0.96	1.00	0.92	1.00	0.96	0.92
$\sigma_t = 0.4$, LSTM	0.72	0.92	0.92	0.40	1.00	1.00	0.44
$\sigma_t = 0.8$, LSTM	0.44	0.36	0.20	0.40	0.28	0.04	0.76
$\sigma_t = 1.0$, LSTM	0.24	0.28	0.12	0.28	0.72	0.0	0.56
$\sigma_t = 0.1$, SIREN	1.00	1.00	1.00	0.96	0.92	0.96	0.96
$\sigma_t = 0.2$, SIREN	0.92	1.00	1.00	0.84	1.00	0.96	0.92
$\sigma_t = 0.4$, SIREN	0.96	0.96	0.8	0.64	0.88	0.88	0.40
$\sigma_t = 0.8$, SIREN	0.32	0.40	0.16	0.52	0.52	0.16	0.44
$\sigma_t = 1.0$, SIREN	0.64	0.08	0.20	0.28	0.24	0.12	0.24

and 2) the experts may not provide strong supervision to correct misclassifications. We examine the model's performance in each scenario. To illustrate model performance, we focus only on the classification of failure mode F_2.

Expert's Prior Knowledge. We vary two components of the expert's assessment: 1) the amplitude of the sine or cosine functions and the frequency of the sine or cosine functions. We use a training strategy similar to the previous experiments, see Table 5. Scale parameters θ_1 are generated from $\{-10, -5, -2, 0, 2, 5, 10\}$ and frequency parameters θ_2 are generated from $\{0.1, 1, 1.5, 2, 2.5, 3, 3.5\}$. When we alter θ_1, θ_2 is fixed to be the same as the frequency parameter in actual equation (2.2). When we alter θ_2, θ_1 is taken to be the same as the scale parameter in the actual equation (-0.7). We use 30 iterations instead of 25, to fully observe $F2$. We assume the experts are presented to give correct classification labels when the model misclassifies.

Table 5: Parameters for Data Generation in Experiment 3.

Fault Mode	**Expert's Assumption**	**Actual Equation**
1	$2\cos(5t)$	$2.2\cos(5t) + \epsilon_t \sim N(0, 0.01)$
2	$\theta_1 \cos(\theta_2 t)$	$-0.7\cos(2.2t) + \epsilon_t \sim N(0, 0.01)$
3	$0.6\cos(0.5t)$	$0.7\cos(0.5t) + \epsilon_t \sim N(0, 0.01)$

Results. We show the performance of the model model under different prior knowledge, with its accuracy in detecting $F2$ and the MSE convergence for $F2$. For simple periodic functions, if enough data is given to the SIRENs for each of the iterations and an expert is presented to correct misclassified labels, the model is able to give relatively accurate predictions. In both cases, the MSE between the true function and predicted function converges in Figure 6. We further compare the accuracy results with strong expert's supervision and weak expert's supervision, provided in Table 6.

Weak Supervision. We conduct the the same experiments as shown in Table 5 and varying the scale θ_1 and frequency θ_2 with two same ranges of parameters, $\theta_1 \in \{-5, -2, 0, 2, 5\}$, and $\theta_2 \in \{1, 1.5, 2, 2.5, 3\}$. However, in this experiment we assume that the experts cannot give correct labels to the classification results (no labels are provided).

Results. The MSE for convergence is shown in Figure 8 and the accuracy in comparison with strong supervision is shown in Table 6. Without expert's correction of the wrongly classified labels, the model does not modify the internal physical neural network associated with the failure mode. Thus, the classifier will never recover the true underlying physical model and MSE never converges. Our experiment shows that the expert's strong supervision is important, especially in the event of limited data, to initialize the physical models.

7 Conclusion

We provide an initial test of an online evidential reasoning machine learning model under various window sizes at each time of query, different levels of noise, inaccurate prior knowledge, and expert supervision. Overall, our results suggest that 1) function recovery and accuracy are high when the

Table 6: The accuracy of the model under different scales and frequencies comparing the expert's strong supervision and weak supervision.

Accuracy	**Strong Expert**	**Weak Expert**
$\theta_1 = -5, \theta_2 = 2.2$	1.0	0.86
$\theta_1 = -2, \theta_2 = 2.2$	1.0	1.0
$\theta_1 = 0, \theta_2 = 2.2$	1.0	1.0
$\theta_1 = 2, \theta_2 = 2.2$	1.0	0.0
$\theta_1 = 5, \theta_2 = 2.2$	1.0	0.0
Accuracy	**Strong Expert**	**Weak Expert**
$\theta_2 = 1.0, \theta_1 = -0.7$	1.0	0.0
$\theta_2 = 1.5, \theta_1 = -0.7$	1.0	0.0
$\theta_2 = 2.0, \theta_1 = -0.7$	1.0	1.0
$\theta_2 = 2.5, \theta_1 = -0.7$	1.0	1.0
$\theta_2 = 3.5, \theta_1 = -0.7$	1.0	0.0

expert proposes a reasonably accurate model of the physical process and is able to supervise the model when it makes errors, even when the number of observations are small; 2) function recovery and accuracy may degrade when the expert is either very wrong or there is too much noise in the system; 3) a small amount of strong supervision from the expert can realign the algorithm in the presence of inaccurate initial estimates or noise.

8 Broader Impact and Future Work

Human experts can improve upon purely data-driven models in terms of accuracy, interpretability, and reliability, while reducing economic costs associated with computation. Our model combines expert knowledge of physics and model parameters with algorithms to improve evidential reasoning in data-constrained settings. In future work we hope to deploy the proposed system to a complex mechanical systems as described in Figure 1.

Modern machine learning models used for research have increased in complexity and size leading to significant energy consumption and resource use. The implications of such have not been fully explored and is still in nascent stages. García-Martín *et al*. [20] describe methods to estimate energy consumption of machine learning models. In future work, we aim to estimate and study the effects of experts-guidance in learning systems to reduce energy consumption and model complexity.

Another avenue of interest is the joint learning of experts and machines. Interesting questions include "Can machines guide experts in learning?" and "Can machine can teach experts new physics?". Feedback loops though which machines can communicate to experts with information and suggestions can improve learning and teaching. In the current framework, this would be advising the experts on the effectiveness of the priors provided or the causal relations.

Finally, inclusion of neural differential equations (ODEs, PDEs) is another area where experts may be able to provide stronger functional form assumptions to the network. Chen *et al*. [21] represent the transformations of inputs through a neural network as a continuous ordinary differential equation rather than discrete layers. Maulik *et al*. [22] apply the approach to the viscous Berger's equation. Recently, Kidger *et al*. [23] use controlled differential equations to allow neural differential equation time series models with missing data and irregularly spaced temporal observations.

References

[1] N. Laptev, S. Amizadeh, and I. Flint, "Generic and scalable framework for automated time-series anomaly detection," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, (New York, NY, USA), p. 1939–1947, Association for Computing Machinery, 2015.

[2] T. Ahmed, B. Oreshkin, and M. Coates, "Machine learning approaches to network anomaly detection," in *Proceedings of the 2nd USENIX workshop on Tackling computer systems problems with machine learning techniques*, pp. 1–6, USENIX Association, 2007.

[3] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "Lstm-based encoder-decoder for multi-sensor anomaly detection," 2016.

[4] R. Patil and B. Kothavale, "Failure modes and effects analysis (fmea) of computerized numerical control (cnc) turning center," *International Review of Mechanical Engineering*, vol. 12, no. 1, pp. 78–87, 2018.

[5] L. Ruthotto and E. Haber, "Deep neural networks motivated by partial differential equations," *Journal of Mathematical Imaging and Vision*, pp. 1–13, 2019.

[6] E. Dupont, A. Doucet, and Y. W. Teh, "Augmented neural odes," in *Advances in Neural Information Processing Systems*, pp. 3140–3150, 2019.

[7] Z. Long, Y. Lu, and B. Dong, "Pde-net 2.0: Learning pdes from data with a numeric-symbolic hybrid deep network," *Journal of Computational Physics*, vol. 399, p. 108925, 2019.

[8] S. Abdollahi, A. Davis, J. H. Miller, and A. W. Feinberg, "Expert-guided optimization for 3d printing of soft and liquid materials," *PloS one*, vol. 13, no. 4, p. e0194890, 2018.

[9] E. D. Gennatas, J. H. Friedman, L. H. Ungar, R. Pirracchio, E. Eaton, L. Reichman, Y. Interian, C. B. Simone, A. Auerbach, E. Delgado, M. J. V. der Laan, T. D. Solberg, and G. Valdes, "Expert-augmented machine learning," 2019.

[10] Y. Xu, A. Joshi, A. Singh, and A. Dubrawski, "Zeroth order non-convex optimization with dueling-choice bandits," in *Conference on Uncertainty in Artificial Intelligence*, pp. 899–908, PMLR, 2020.

[11] F. Jia, Y. Lei, L. Guo, J. Lin, and S. Xing, "A neural network constructed by deep learning technique and its application to intelligent fault diagnosis of machines," *Neurocomputing*, vol. 272, pp. 619–628, 2018.

[12] S. R. Saufi, Z. A. B. Ahmad, M. S. Leong, and M. H. Lim, "Challenges and opportunities of deep learning models for machinery fault detection and diagnosis: A review," *IEEE Access*, vol. 7, pp. 122644–122662, 2019.

[13] D. Kaufmann, S. Parbhoo, A. Wieczorek, S. Keller, D. Adametz, and V. Roth, "Bayesian markov blanket estimation," in *Artificial Intelligence and Statistics*, pp. 333–341, 2016.

[14] C. Louizos, U. Shalit, J. M. Mooij, D. Sontag, R. Zemel, and M. Welling, "Causal effect inference with deep latent-variable models," in *Advances in Neural Information Processing Systems*, pp. 6446–6456, 2017.

[15] S. Hong, Z. Zhou, C. Lu, B. Wang, and T. Zhao, "Bearing remaining life prediction using gaussian process regression with composite kernel functions," *Journal of Vibroengineering*, vol. 17, no. 2, pp. 695–704, 2015.

[16] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations," *arXiv preprint arXiv:1711.10561*, 2017.

[17] M. Raissi, A. Yazdani, and G. E. Karniadakis, "Hidden fluid mechanics: A navier-stokes informed deep learning framework for assimilating flow visualization data," *arXiv preprint arXiv:1808.04327*, 2018.

[18] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.

[19] V. Sitzmann, J. N. Martel, A. W. Bergman, D. B. Lindell, and G. Wetzstein, "Implicit neural representations with periodic activation functions," *arXiv preprint arXiv:2006.09661*, 2020.

[20] E. García-Martín, C. F. Rodrigues, G. Riley, and H. Grahn, "Estimation of energy consumption in machine learning," *Journal of Parallel and Distributed Computing*, vol. 134, pp. 75–88, 2019.

[21] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," in *Advances in neural information processing systems*, pp. 6571–6583, 2018.

[22] R. Maulik, A. Mohan, B. Lusch, S. Madireddy, P. Balaprakash, and D. Livescu, "Time-series learning of latent-space dynamics for reduced-order model closure," *Physica D: Nonlinear Phenomena*, vol. 405, p. 132368, 2020.

[23] P. Kidger, J. Morrill, J. Foster, and T. Lyons, "Neural controlled differential equations for irregular time series," *arXiv preprint arXiv:2005.08926*, 2020.

9 Appendix

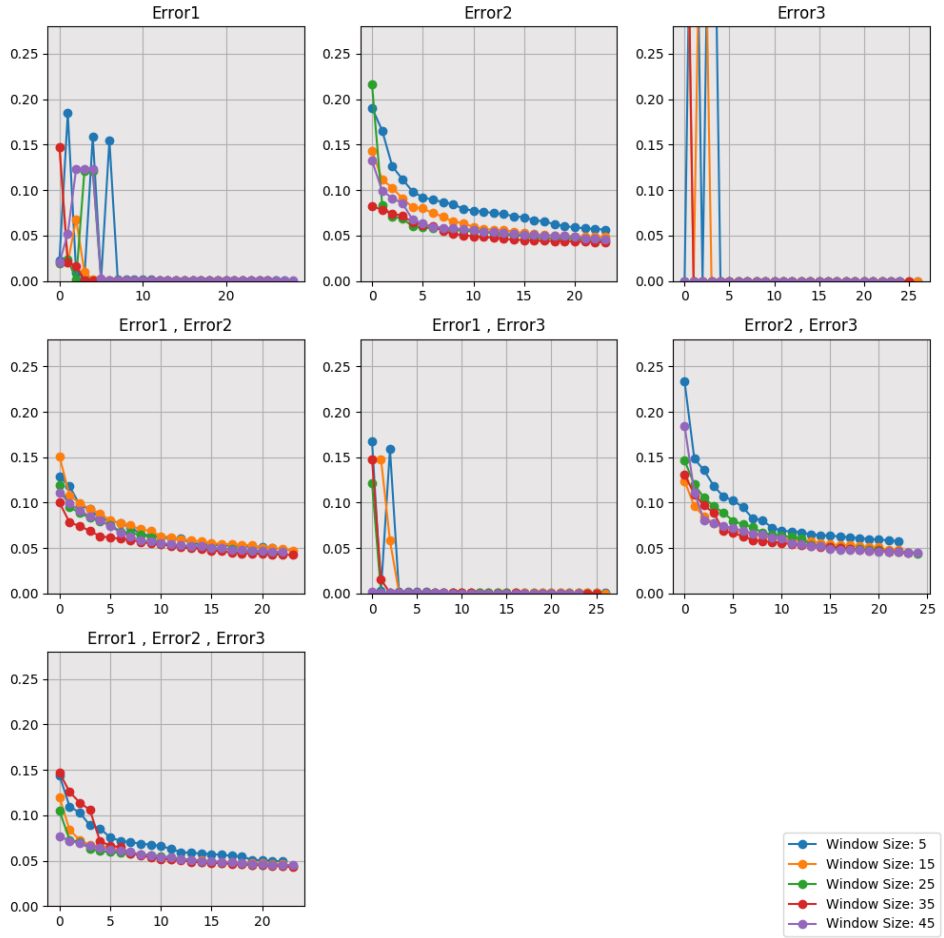


Figure 3: Mean square error over 25 iterations by error mode for different time windows(LSTM).

Figure 4: Mean square error over 25 iterations by error mode for different time windows(SIREN).

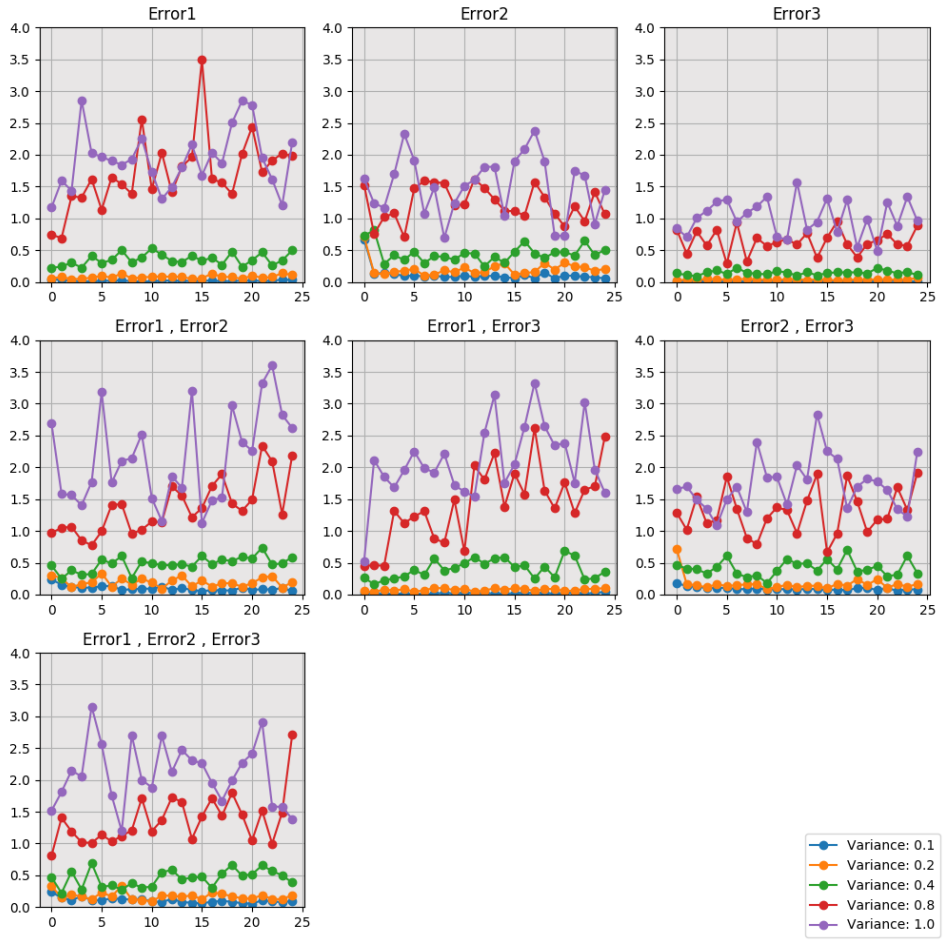
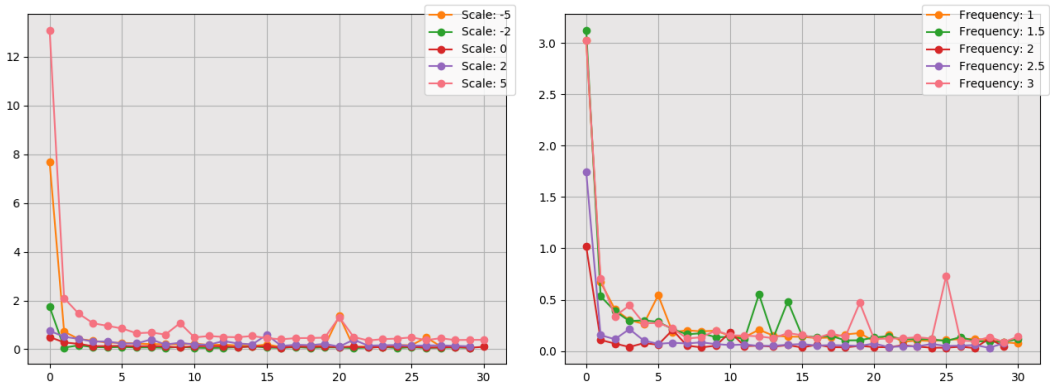


Figure 5: Mean square error over 25 iterations by error mode for different variance of data(LSTM).



(a) Change the expert's guess on scales.

(b) Change the expert's guess on frequencies.

Figure 6: MSE convergence between true and predicted functions(SIREN)

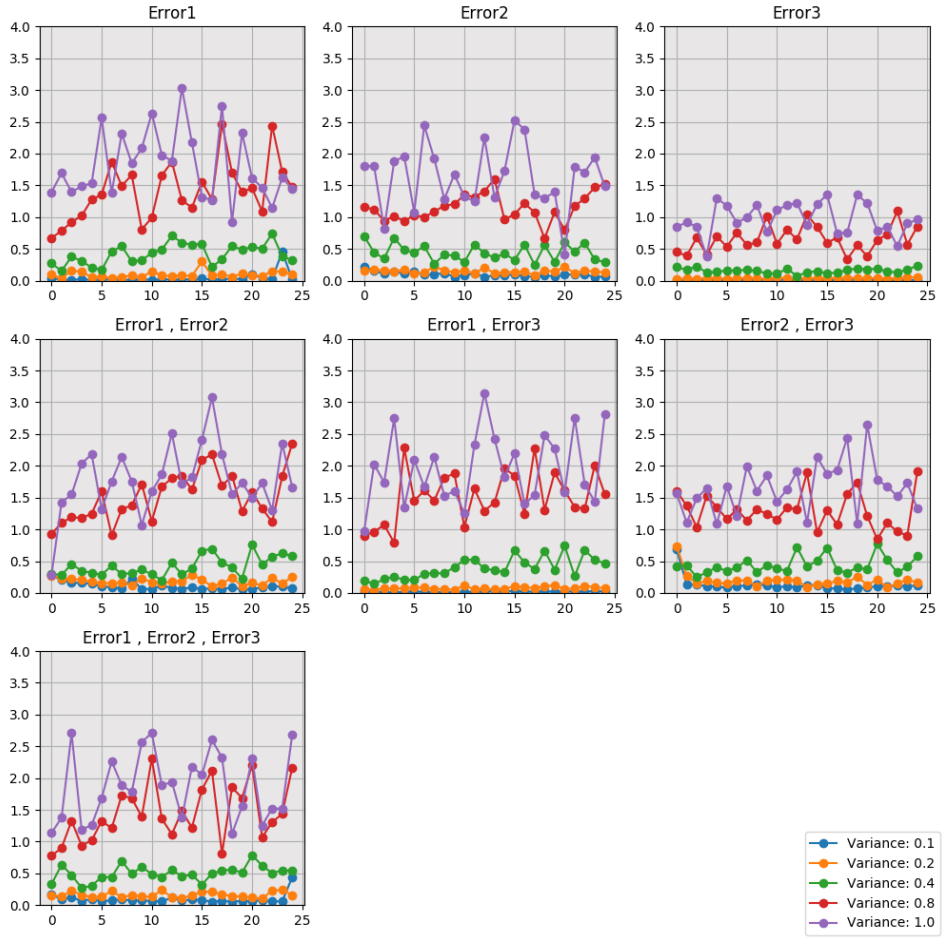
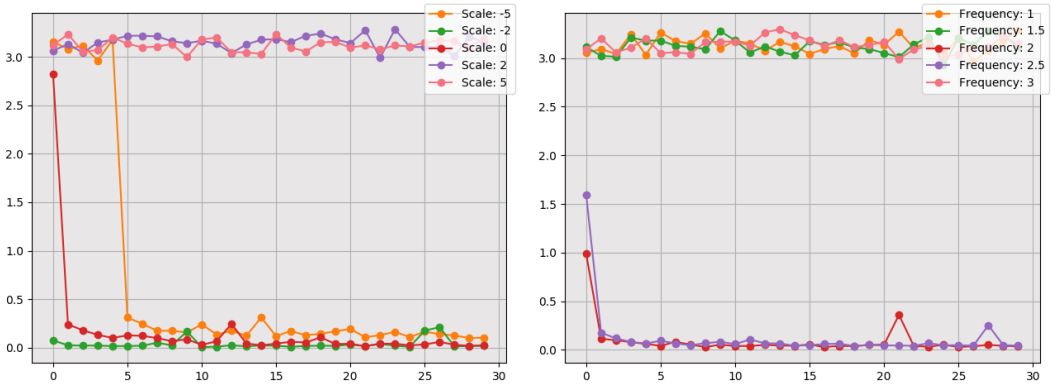


Figure 7: Mean square error over 25 iterations by error mode for different variance of data(SIREN).



(a) Change the expert's guess on scales.

(b) Change the expert's guess on frequencies.

Figure 8: With weak expert's intervention, MSE convergence between true and predicted functions(SIREN)