Mapping general inference to physics

J. Shepard Bryan IV and Pressé Lab

Department of Physics, Arizona State University

April 26, 2024

Abstract

Here we attempt to map general inference to physics.

DISCLAIMER: This is a work in progress. Please excuse any typos or errors. Please also excuse the lack of citations.

1 Introduction

It is well known that there is a fundamental connection between physics and learning. For example, recording information requires energy [**Landauer**, **Bechhoeffer**, 1], the act of measurement unavoidably alters the measured object [**quantumtextbook**, 1], and physical objects are known to be capable of learning [1]. Moreover, the whole field of statistical mechanics is based on the concept of "incomplete information" [1], which requires the existence of an observer capable of learning [**maxwellsdemon**, 1]. Recently, there has been interested in exploiting the connection between physics and learning to better understand deep learning [1] and explore new models [**hopfield**, 1]. In this work, we expand upon this work by creating a direct mapping from general inference to physics.

The heart of what we would like to explore is as follows: Imagine actively measuring some unlabeled variables over a long series of time steps, with the goal of predicting the value of each variable at the next time step. For example, the measurements could be readouts from an experiment, values of nodes in a neural network, or states (on or off) of neurons in a brain at a time step. How would one best simplify and make sense of this data? What general inference algorithm would allow one to best predict future measurements, regardless of the form of the object being measured?

Here, we create a mapping from this general inference problem to physics. In this paradigm, physical quantities such as mass, charge, space, and spin are defined as statistical quantities of measured variables, and natural constants such as the vacuum permittivity emerge as averaged error. Additionally, machine learning concepts such as gradient descent optimizers are replaced with physics inspired dynamics laws. We show that this paradigm can stably minimize prediction error. Moreover, our mapping has potential advantages over traditional physics based inference models [**hopfield**, 1] because it allows for long range interactions between variables while minimizing the number of model parameters.

2 Methods

Here we lay out the mathematical framework for our work mapping general inference to physics. We start by presenting a graphical model for the general inference problem, starting with the most general case of fully connected binary random variables, then making approximations to simplify the network. We then derive the loss function for a prediction model on the network, decomposing the loss term into two parts: one governing the continuous variables, and another governing the binary variables. We show that when the model is optimized (has minimal error) this loss function effectively acts as a concerned quantity. We then show that careful choices of model parameters lead to model parameter dynamics, i.e. the way in which model parameters are updated after each new measurement, which are identical to electrodynamics and quantum mechanics.

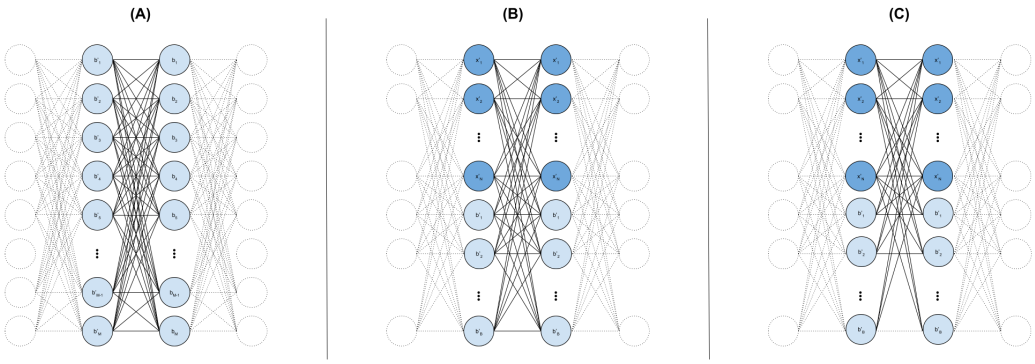

Figure 1: Here, we present a graphical model for the general inference problem under three different cases. (A) A fully connected network of binary random variables. (B) A fully connected mixture network of both binary and continuous variables. (C) A network of fully connected continuous variables, but sparsely connected binary variables.

2.1 Graphical model

Our first goal is to rigorously define the problem. Loosely speaking, we want to predict future measurements of a system given a stream of past measurements. Let us start by defining the variables we are measuring.

Some of the variables, we will call "external variables", in our system will directly correspond to measurements of external observables. For example, think of a sensory neuron in a human body that fires in response to heat [**trp**, 1]. The firing state of this neuron is directly linked to the temperature of the external environment. Conversely, some of the variables in our system will be "internal variables". For example, think of an interneuron, which fires in response to other neurons in the body, but not in response to external stimuli [**interneuron**, 1]. These internal variables are necessary for predicting the external variables, thus they are included in our system. We will make no distinction between internal and external variables in our system.

Let us now assume that all the variables in the system are binary random variables. Let there be M binary variables, labelled $b_1, b_2, \ldots, b_M$, and let $\boldsymbol{b}'_{1:M}$ be the grouping of all variables. At first, it may seem that by considering only binary random variables, we limit the systems that we can look at, however, when we remember that all measurements will have finite precision we can safely assume that all measurements can be converted into sets of binary random variables. For example, a variable measuring temperature to 32 bit precision, can be decomposed into 32 separate binary variables.

Lastly, regarding the binary variables, we will use a convention where the possible outcomes of the binary variables are -1 and 1, like those we experience in quantum spin mechanics, instead of 0 and 1 like we use in traditional probability theory. This choice does not lose generality, as we can always convert back to the probability convention via $b \to (b+1)/2$.

Next, let us introduce memory into our system. Let the prime symbol ($'$) indicate past measurement. So dividing the measurement process into discrete time steps, b_n represents the measurement of the nth variable at some time level and b'_n represents the measurement of the same variable at the previous time step. We will only consider two time steps at a time, with the hope that the states of internal variables will be able to retain information for longer periods.

We now need to define the network of relationships between variables. In the most general case, a fully connected network, every variable $\boldsymbol{b}_{1:M}$ is conditioned on the variables at the previous time step, $\boldsymbol{b}'_{1:M}$. Figure 1 panel A shows a graphical representation of this network. Notice that every variable at the first time step, b'_i, is connected to every variable at the subsequent time step, b_j. Notice that this fully connected pattern continues in the future and past, represented by greyed out nodes.

We can write out the probability of the future states, $\boldsymbol{b}_{1:M}$, given the past states, $\boldsymbol{b}'_{1:M}$ according to a Boltzmann distribution

$$\mathcal{P}\left(b_i|\boldsymbol{b}'_{1:M}\right) = \frac{1}{Z_i}\exp\left(-\beta U(b_i, \boldsymbol{b}'_{1:M})\right) \tag{1}$$

$$Z_i = \exp\left(-\beta U(+1, \boldsymbol{b}'_{1:M})\right) + \exp\left(-\beta U(-1, \boldsymbol{b}'_{1:M})\right) \tag{2}$$

where U is some energy of the state. Taylor expanding the energy to first order, we get

$$U(b_i, \boldsymbol{b'}_{1:M}) \approx b_i(J_{0i} + \sum_j J_{ij} {b'}_j) \tag{3}$$

$$\mathcal{P}\left(b_i | \boldsymbol{b'}_{1:M}\right) \approx \frac{1}{Z_i} \exp\left(-\beta b_i (J_{0i} + \sum_j J_{ij} {b'}_j)\right) \tag{4}$$

where β represents an inverse temperature, J_{0i} represents some base energy for the ith variable, J_{ij} represents the interaction energies.

Note that the connection strengths between nodes may change over time, however, we will assume that if the connection strengths change over time, they do so slowly enough that to some certainty, the connection strength between ${b'}_i$ and b_j will be approximately equal to the connection strength between ${b''}_i$ and ${b'}_j$.

Equation (4) is effectively a stochastic fully connected binary neural network [1], which has $M^2 + M$ connections. When M is large, it is prohibitively costly to model this system, thus our next goal is to simplify the graphical model. We can simplify this graphical model by clustering variables with large coefficients of correlation (see supplementary section S0.1). In other words, we can replace large clusters of correlated our binary variables in our system with variables that only encode the sum of the cluster. For example, if $\boldsymbol{b}_{1:K}$ are highly correlated, we may replace them with

$$x = \sum_{k=1}^{K} b_k. \tag{5}$$

This cluster variable, x, will be distributed around some mean, with variance related to the error of our clustering process times the number of components in the cluster, which we will discuss in later sections. Note that when the number of binary variables that make up the cluster is large, then we can approximate cluster variables as continuous variables. Moving forward, we will use this approximation and treat x as continuously distributed.

Let us proceed by replacing every highly correlated subset of $\boldsymbol{b}_{1:M}$ with such cluster variables. For concreteness, let us say that there are N such clusters, labelled $x_1, x_2, \ldots, x_N$, and B remaining binary variables, which are not highly correlated with any of the subsets. Let us use similar notation allowing $\boldsymbol{x}_{1:N}$ to represent the grouping of the cluster variables and let $\boldsymbol{x'}_{1:N}$ represent the cluster variables at the previous time step.

After the substitution process laid out above, our new graphical model will look like the one presented in figure 1 panel B. Here we have N cluster variables and B binary variables. The network is fully connected, with $(N + B)^2 + (N + B)$ parameters. Since $M > N + B$ this new system has exponentially fewer parameters.

We can simplify this graphical model further by considering that highly correlated binary variables have already been clustered. By definition, this means that the remaining correlations between variables are weak. Thus, we can ignore connections between binary variables. However, we will keep the connections between clusters, since weak correlations may be balanced out by the large variance of each x_n. This final graphical model is shown in figure 1 panel C, where we see that the clusters are fully connected, but each binary variable, b_i is connected only to the clusters.

We are now ready to construct a model to predict the system. In the next section, we construct a loss function for the system presented in figure 1 panel C.

2.2 Loss function

In the previous section, we built out a graphical model representing a system of fully connected binary variables as a set of clusters and sparsely connected binary variables. In this section, we introduce the concept of a model, which is used to predict the values of the variables in the system. We then construct a loss function for the model.

Consider a single time step. Let us define the ground truth distribution of variables as

$$\mathcal{P}^*\left(\boldsymbol{x}_{1:N}, \boldsymbol{b}_{1:B}\right)$$

where the star indicates that this is the "real" distribution. Next, we define a model that predicts the values of the system variables,

$$\mathcal{P}\left(\boldsymbol{x}_{1:N}, \boldsymbol{b_{1:B}} | \boldsymbol{\Theta}\right) \mathcal{P}\left(\boldsymbol{\Theta}\right)$$

which has two parts: 1) a likelihood, $\mathcal{P}\left(\boldsymbol{x}_{1:N}, \boldsymbol{b}_{\boldsymbol{1:B}}|\boldsymbol{\Theta}\right)$, which gives the probability of the measurements according to some model parameters, $\boldsymbol{\Theta}$, and 2) a prior, $\mathcal{P}\left(\boldsymbol{\Theta}\right)$,

$$\mathcal{P}\left(\boldsymbol{x}_{1:N}, \boldsymbol{b}_{\boldsymbol{1:B}}|\boldsymbol{\Theta}\right)$$

and a prior over model parameters which acts as a regularizer over model parameters. The hope is that the model will assign probabilities to measurements that are as close as possible to the ground truth distribution. We can quantify the error of the model using the Kullback-Leibler divergence [1]

$$\begin{aligned}
\text{KLD} &= -\int d\boldsymbol{x}_{1:N} \sum_{\boldsymbol{b}_{1:B}} \mathcal{P}^*\left(\boldsymbol{x}_{1:M}, \boldsymbol{b}_{1:B}\right) \log\left(\frac{\mathcal{P}\left(\boldsymbol{x}_{1:M}, \boldsymbol{b}_{1:B}|\boldsymbol{\Theta}\right)\mathcal{P}\left(\boldsymbol{\Theta}\right)}{\mathcal{P}^*\left(\boldsymbol{x}_{1:M}, \boldsymbol{b}_{1:B}\right)}\right) && (6)\\
&= -\int d\boldsymbol{x}_{1:N} \sum_{\boldsymbol{b}_{1:B}} \mathcal{P}^*\left(\boldsymbol{x}_{1:M}, \boldsymbol{b}_{1:B}\right) \left(\log\left(\mathcal{P}\left(\boldsymbol{x}_{1:M}, \boldsymbol{b}_{1:B}|\boldsymbol{\Theta}\right)\mathcal{P}\left(\boldsymbol{\Theta}\right)\right) - \log\left(\mathcal{P}^*\left(\boldsymbol{x}_{1:M}, \boldsymbol{b}_{1:B}\right)\right)\right) && (7)\\
&= -\mathcal{H}_0 - \int d\boldsymbol{x}_{1:N} \sum_{\boldsymbol{b}_{1:B}} \mathcal{P}^*\left(\boldsymbol{x}_{1:M}, \boldsymbol{b}_{1:B}\right) \log\left(\mathcal{P}\left(\boldsymbol{x}_{1:M}, \boldsymbol{b}_{1:B}|\boldsymbol{\Theta}\right)\mathcal{P}\left(\boldsymbol{\Theta}\right)\right) && (8)\\
\mathcal{H}_0 + \text{KLD} &= -\int d\boldsymbol{x}_{1:N} \sum_{\boldsymbol{b}_{1:B}} \mathcal{P}^*\left(\boldsymbol{x}_{1:M}, \boldsymbol{b}_{1:B}\right) \log\left(\mathcal{P}\left(\boldsymbol{x}_{1:M}, \boldsymbol{b}_{1:B}|\boldsymbol{\Theta}\right)\mathcal{P}\left(\boldsymbol{\Theta}\right)\right) && (9)
\end{aligned}$$

where $\mathcal{H}_0$ is the intrinsic entropy of the system. Equation (9) states that the average prediction error (defined as the log probability of the model, which is often called the "surprise" [1]) of our model is equal to the intrinsic entropy of the system, $\mathcal{H}_0$, plus the error (KL Divergence) between our model and the ground truth. Defining loss as

$$\begin{aligned}
\mathcal{L} &= \mathcal{H}_0 + \text{KLD} && (10)\\
&= -\int d\boldsymbol{x}_{1:N} \sum_{\boldsymbol{b}_{1:B}} \mathcal{P}^*\left(\boldsymbol{x}_{1:M}, \boldsymbol{b}_{1:B}\right) \log\left(\mathcal{P}\left(\boldsymbol{x}_{1:M}, \boldsymbol{b}_{1:B}|\boldsymbol{\Theta}\right)\mathcal{P}\left(\boldsymbol{\Theta}\right)\right) && (11)
\end{aligned}$$

we see that minimizing the loss minimizes the prediction error.

One key difference between our loss function, equation (11), and traditional loss terms defined elsewhere [1] is that we include the prior term as an inseparable part of the model. In later sections we will show that prior will act as both a regularizer and a momenta term during training, which may be advantageous over other paradigms in which loss functions, regularizers, and optimizers are defined separately.

We can simplify this loss function further, decomposing it into a loss from the clusters and a loss from the binary variables,

$$\begin{aligned}
\mathcal{L} &= -\int d\boldsymbol{x}_{1:N} \sum_{\boldsymbol{b}_{1:B}} \mathcal{P}^*\left(\boldsymbol{x}_{1:M}, \boldsymbol{b}_{1:B}\right) \log\left(\mathcal{P}\left(\boldsymbol{x}_{1:M}, \boldsymbol{b}_{1:B}|\boldsymbol{\Theta}\right)\mathcal{P}\left(\boldsymbol{\Theta}\right)\right) && (12)\\
&= -\int d\boldsymbol{x}_{1:N} \sum_{\boldsymbol{b}_{1:B}} \mathcal{P}^*\left(\boldsymbol{x}_{1:M}, \boldsymbol{b}_{1:B}\right) \log\left(\mathcal{P}\left(\boldsymbol{b}_{1:B}|\boldsymbol{x}_{1:M}, \boldsymbol{\Theta}\right)\mathcal{P}\left(\boldsymbol{x}_{1:M}|\boldsymbol{\Theta}\right)\mathcal{P}\left(\boldsymbol{\Theta}\right)\right) && (13)\\
&= -\int d\boldsymbol{x}_{1:N} \sum_{\boldsymbol{b}_{1:B}} \mathcal{P}^*\left(\boldsymbol{x}_{1:M}, \boldsymbol{b}_{1:B}\right) \left(\log\left(\mathcal{P}\left(\boldsymbol{b}_{1:B}|\boldsymbol{x}_{1:M}, \boldsymbol{\Theta}\right)\right) + \log\left(\mathcal{P}\left(\boldsymbol{x}_{1:M}|\boldsymbol{\Theta}\right)\right) + \log\left(\mathcal{P}\left(\boldsymbol{\Theta}_x\right)\right)\right) && (14)\\
&= -\int d\boldsymbol{x}_{1:N} \sum_{\boldsymbol{b}_{1:B}} \mathcal{P}^*\left(\boldsymbol{x}_{1:M}, \boldsymbol{b}_{1:B}\right) \log\left(\mathcal{P}\left(\boldsymbol{b}_{1:B}|\boldsymbol{x}_{1:M}, \boldsymbol{\Theta}\right)\right) - \int d\boldsymbol{x}_{1:N}\mathcal{P}^*\left(\boldsymbol{x}_{1:M}\right) \log\left(\mathcal{P}\left(\boldsymbol{x}_{1:M}|\boldsymbol{\Theta}\right)\right) - \log\left(\mathcal{P}\left(\boldsymbol{\Theta}\right)\right) && (15)\\
&= -\int d\boldsymbol{x}_{1:N} \sum_{\boldsymbol{b}_{1:B}} \mathcal{P}^*\left(\boldsymbol{x}_{1:M}, \boldsymbol{b}_{1:B}\right) \log\left(\prod_{i=1}^{B} \mathcal{P}\left(b_i|\boldsymbol{x}_{1:M}, \boldsymbol{\Theta}\right)\right) - \int d\boldsymbol{x}_{1:N}\mathcal{P}^*\left(\boldsymbol{x}_{1:M}\right) \log\left(\mathcal{P}\left(\boldsymbol{x}_{1:M}|\boldsymbol{\Theta}\right)\right) - \log\left(\mathcal{P}\left(\boldsymbol{\Theta}\right)\right) && (16)\\
&= -\sum_{i=1}^{B}\int d\boldsymbol{x}_{1:N} \sum_{\boldsymbol{b}_{1:B}} \mathcal{P}^*\left(\boldsymbol{x}_{1:M}, \boldsymbol{b}_{1:B}\right) \log\left(\mathcal{P}\left(b_i|\boldsymbol{x}_{1:M}, \boldsymbol{\Theta}\right)\right) - \int d\boldsymbol{x}_{1:N}\mathcal{P}^*\left(\boldsymbol{x}_{1:M}\right) \log\left(\mathcal{P}\left(\boldsymbol{x}_{1:M}|\boldsymbol{\Theta}\right)\right) - \log\left(\mathcal{P}\left(\boldsymbol{\Theta}\right)\right) && (17)\\
&= -\sum_{i=1}^{B}\int d\boldsymbol{x}_{1:N} \sum_{b_i} \mathcal{P}^*\left(b_i, \boldsymbol{x}_{1:M}\right) \log\left(\mathcal{P}\left(b_i|\boldsymbol{x}_{1:M}, \boldsymbol{\Theta}\right)\right) - \int d\boldsymbol{x}_{1:N}\mathcal{P}^*\left(\boldsymbol{x}_{1:M}\right) \log\left(\mathcal{P}\left(\boldsymbol{x}_{1:M}|\boldsymbol{\Theta}\right)\right) - \log\left(\mathcal{P}\left(\boldsymbol{\Theta}\right)\right) && (18)
\end{aligned}$$

Where we have used the fact that binary variables are independent in the model to separate the binary variable terms. Finally by splitting $\boldsymbol{\Theta}$ into parameters which affect only the binary variables, $\boldsymbol{\Theta}_b$, and parameters which only affect the clusters, $\boldsymbol{\Theta}_x$, we get

$$\mathcal{L} = \mathcal{L}_b + \mathcal{L}_x \tag{19}$$

$$\mathcal{L}_b = -\log\left(\mathcal{P}\left(\boldsymbol{\Theta}_b\right)\right) - \sum_{i=1}^{B} \int d\boldsymbol{x}_{1:N} \mathcal{P}^*\left(\boldsymbol{x}_{1:M}\right) \sum_{b_i} \mathcal{P}^*\left(b_i|\boldsymbol{x}_{1:M}\right) \log\left(\mathcal{P}\left(b_i|\boldsymbol{x}_{1:M}, \boldsymbol{\Theta}_b\right)\right) \tag{20}$$

$$\mathcal{L}_x = -\log\left(\mathcal{P}\left(\boldsymbol{\Theta}_x\right)\right) - \int d\boldsymbol{x}_{1:N} \mathcal{P}^*\left(\boldsymbol{x}_{1:M}\right) \log\left(\mathcal{P}\left(\boldsymbol{x}_{1:M}|\boldsymbol{\Theta}\right)\right) \tag{21}$$

where we have successfully split the loss into two parts, one governing binary variables and one governing cluster variables.

Here we have defined probability distributions over the system variables and our model then constructed a loss function governing prediction error. We have not yet defined the architecture of the model, nor what the parameters of the model represent. In the next sections, we will choose a model architecture that is explicitly designed to minimize this loss. We will start with the loss over cluster variables, then move onto binary variables.

2.3 Model over cluster variables

Our goal now is to start designing the architecture of our model. In this section, we will design the model architecture for the distribution over cluster variables, with the goal of minimizing the error defined in equation (21). Because a Gaussian distribution can be thought of as a second order approximation to arbitrary distributions in log space [1], we will use it as the functional form for our model distribution

$$\mathcal{P}\left(\boldsymbol{x}_{1:M}|\boldsymbol{\Theta}_x\right) \approx \mathcal{N}\left(\boldsymbol{x}_{1:M}; \boldsymbol{0}, \boldsymbol{K}\right) \tag{22}$$

$$= \left((2\pi)^M |\boldsymbol{K}|\right)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\boldsymbol{x}_{1:M}\boldsymbol{K}^{-1}\boldsymbol{x}_{1:M}\right) \tag{23}$$

where $\boldsymbol{K}$ is the covariance of the distribution. Notice that for simplicity we have set the mean of the distribution to zero, which can be interpreted as implicitly shifting the measurements by $\boldsymbol{x}_{1:M} \to \boldsymbol{x}_{1:M} - \boldsymbol{\mu}_{1:M}$ where $\boldsymbol{\mu}_{1:M}$ is the mean of the raw model distribution. We will eventually marginalize over measurements, meaning that this shift will have no effect on the final result, but simplifies the derivation.

Plugging in equation (23) into the loss function, equation (21), we get

$$\mathcal{L}_x = -\log\left(\mathcal{P}\left(\boldsymbol{\Theta}_x\right)\right) - \int d\boldsymbol{x}_{1:N} \mathcal{P}^*\left(\boldsymbol{x}_{1:M}\right) \log\left(\left((2\pi)^M |\boldsymbol{K}|\right)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\boldsymbol{x}_{1:M}\boldsymbol{K}^{-1}\boldsymbol{x}_{1:M}\right)\right) \tag{24}$$

$$= -\log\left(\mathcal{P}\left(\boldsymbol{\Theta}_x\right)\right) - \int d\boldsymbol{x}_{1:N} \mathcal{P}^*\left(\boldsymbol{x}_{1:M}\right) \left(-\frac{1}{2}\log\left((2\pi)^M |\boldsymbol{K}|\right) - \frac{1}{2}\boldsymbol{x}_{1:M}\boldsymbol{K}^{-1}\boldsymbol{x}_{1:M}\right) \tag{25}$$

$$= -\log\left(\mathcal{P}\left(\boldsymbol{\Theta}_x\right)\right) + \frac{M}{2}\log\left(2\pi\right) + \frac{1}{2}\log\left(|\boldsymbol{K}|\right) + \frac{1}{2}\int d\boldsymbol{x}_{1:N} \mathcal{P}^*\left(\boldsymbol{x}_{1:M}\right) \boldsymbol{x}_{1:M}\boldsymbol{K}^{-1}\boldsymbol{x}_{1:M}. \tag{26}$$

Let us now define exactly what the model parameters, $\boldsymbol{\Theta}_x$, are and how they relate to the covariance matrix, $\boldsymbol{K}$. A covariance matrix has two parts: 1) the variance of each variable along the diagonal, and 2) the covariance between variables on the off diagonals [1]. Because our goal is to map inference to physics, we will assign physics inspired labels to each part of the covariance.

Remember that we already know from section 2.1 that each cluster, x_i, is the sum of binary variables, meaning that each x_i will be binomial distributed, which has variance proportional to the number of components in the cluster. Therefore, we expect that the variance of x_i will be proportional to the size of the cluster. Let us define,

$$\text{Var}\left[x_i\right] = m_i \tag{27}$$

which we will call the "mass" of the cluster, which has the dual purpose of quantifying both the variance of the variable and the number of binary variables that make up the cluster.

Moving onto the covariance, we notice that covariance encodes two things: the magnitude of correlation and the sign of the correlation. Taking inspiration from physics, we will treat the sign of the covariance as a product of charges, and the strenth of the correlation as an inverse distance

$$\text{Cov}\left[x_i, x_j\right] = \frac{q_i q_j}{|\vec{r}_i - \vec{r}_j|} \tag{28}$$

where q_i and q_j are the charges of variables and $\vec{r}_i$ and $\vec{r}_j$ are the positions of the variables. Through thought experiments like "if A is correlated with B and B is correlated with C" and "if A is anti-correlated with B and B is anti-correlated with C then A is correlated with C" we can conclude that charges can be used to keep track of the sign of correlation. As for positions, inverse distance makes sense because typically we expect objects that are close together to be more correlated than objects that are far apart, an assumption already used in many regression methods [1]. Let us start by assuming that the positions of each variable span a space with D dimensions where D is large enough that we can effectively tune charges and positions of each variable to match any arbitrary set of covariances. We will later show that we we can constrain the number of dimensions to 3 when we define a covariance between cluster variables and binary variables.

All together, plugging our definitions, equations 27 and 28, into the covariance matrix, $\boldsymbol{K}$, we get

$$\boldsymbol{K} = \begin{bmatrix} m_1 & \frac{q_1 q_2}{|\vec{r}_1 - \vec{r}_2|} & \frac{q_1 q_3}{|\vec{r}_1 - \vec{r}_3|} & \cdots \\ \frac{q_1 q_2}{|\vec{r}_1 - \vec{r}_2|} & m_2 & \frac{q_2 q_3}{|\vec{r}_2 - \vec{r}_3|} & \cdots \\ \frac{q_1 q_3}{|\vec{r}_1 - \vec{r}_3|} & \frac{q_2 q_3}{|\vec{r}_2 - \vec{r}_3|} & m_3 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \tag{29}$$

where the pattern continues along the whole N by N matrix. The inverse covariance matrix, $\boldsymbol{K}^{-1}$, can be calculated using a Taylor expansion where we separate the covariance into diagonal and off-diagonal components

$$\begin{aligned} \boldsymbol{K} =& \boldsymbol{K}_{\text{diag}} + \boldsymbol{K}_{\text{off}} && (30) \\ \boldsymbol{K}^{-1} =& (\boldsymbol{K}_{\text{diag}} + \boldsymbol{K}_{\text{off}})^{-1} && (31) \\ \approx& \boldsymbol{K}_{\text{diag}}^{-1} - \boldsymbol{K}_{\text{diag}}^{-1} \boldsymbol{K}_{\text{off}} \boldsymbol{K}_{\text{diag}}^{-1} && (32) \\ =& \begin{bmatrix} \frac{1}{m_1} & \frac{-q_1 q_2}{m_1 m_2 |\vec{r}_1 - \vec{r}_2|} & \frac{-q_1 q_3}{m_1 m_3 |\vec{r}_1 - \vec{r}_3|} & \cdots \\ \frac{-q_1 q_2}{m_1 m_2 |\vec{r}_1 - \vec{r}_2|} & \frac{1}{m_2} & \frac{-q_2 q_3}{m_2 m_3 |\vec{r}_2 - \vec{r}_3|} & \cdots \\ \frac{-q_1 q_3}{m_1 m_3 |\vec{r}_1 - \vec{r}_3|} & \frac{-q_2 q_3}{m_2 m_3 |\vec{r}_2 - \vec{r}_3|} & \frac{1}{m_3} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} && (33) \end{aligned}$$

where we have used the fact that magnitude of covariance between clusters is defined to be small in our approximation, in other words, we drop all $\frac{1}{r^2}$ terms. We can use our inverse covariance matrix to simplify the expression $\boldsymbol{x}_{1:M} \boldsymbol{K}^{-1} \boldsymbol{x}_{1:M}$ which shows up in the loss function, equation (26),

$$\begin{aligned} \int d\boldsymbol{x}_{1:N} \mathcal{P}^*(\boldsymbol{x}_{1:M}) \, \boldsymbol{x}_{1:M} \boldsymbol{K}^{-1} \boldsymbol{x}_{1:M} =& \int d\boldsymbol{x}_{1:N} \mathcal{P}^*(\boldsymbol{x}_{1:M}) \sum_i \sum_j x_i x_j (\boldsymbol{K}^{-1})_{ij} && (34) \\ =& \int d\boldsymbol{x}_{1:N} \mathcal{P}^*(\boldsymbol{x}_{1:M}) \sum_i \left(x_i^2 (\boldsymbol{K}^{-1})_{ii} + \sum_{j \neq i} x_i x_j (\boldsymbol{K}^{-1})_{ij} \right) && (35) \\ =& \int d\boldsymbol{x}_{1:N} \mathcal{P}^*(\boldsymbol{x}_{1:M}) \sum_i \left(\frac{x_i^2}{m_i} - \sum_{j \neq i} \frac{x_i x_j}{m_i m_j} \frac{q_i q_j}{|\vec{r}_i - \vec{r}_j|} \right) && (36) \\ =& \sum_i \left(\frac{\mathrm{E}\left[x_i^2\right]}{m_i} - \sum_{j \neq i} \frac{\mathrm{E}\left[x_i x_j\right]}{m_i m_j} \frac{q_i q_j}{|\vec{r}_i - \vec{r}_j|} \right) && (37) \end{aligned}$$

where $\mathrm{E}[\dots]$ is the expectation value taken over the ground truth distribution. Finally, we can calculate the

determinant by again separating the covariance matrix into diagonal and off diagonal components,

$$
\begin{aligned}
|\boldsymbol{K}| =& |\boldsymbol{K}_{\text{diag}} + \boldsymbol{K}_{\text{off}}| && (38)\\
=& |\boldsymbol{K}_{\text{diag}}||\boldsymbol{I} + \boldsymbol{K}_{\text{diag}}^{-1}\boldsymbol{K}_{\text{off}}| && (39)\\
=& |\boldsymbol{K}_{\text{diag}}| \exp(\text{Trace}(\log(\boldsymbol{I} + \boldsymbol{K}_{\text{diag}}^{-1}\boldsymbol{K}_{\text{off}}))) && (40)\\
\approx& |\boldsymbol{K}_{\text{diag}}| \exp(\text{Trace}(\boldsymbol{K}_{\text{diag}}^{-1}\boldsymbol{K}_{\text{off}})) && (41)\\
=& |\boldsymbol{K}_{\text{diag}}| \exp(0) && (42)\\
=& |\boldsymbol{K}_{\text{diag}}| && (43)\\
=& \prod_{i=1}^{N} m_i && (44)
\end{aligned}
$$

where we have used the identity $\log(\boldsymbol{I} + \boldsymbol{A}) \approx \boldsymbol{A}$.

It is now clear that the parameters of our model, $\boldsymbol{\Theta}_x$, are the positions of the variables. Let $\vec{\boldsymbol{r}}_{1:M}$ be the grouping of positions defined in equation (28) and let $\boldsymbol{r}_{1:M}^{(d)}$ be the grouping of all positions along the dth dimension. We must choose a prior over these positions. Let us choose

$$
\begin{aligned}
\mathcal{P}\left(\boldsymbol{\Theta}_x\right) =& \mathcal{P}\left(\vec{\boldsymbol{r}}_{1:M}\right) && (45)\\
=& \prod_{d=1}^{D} \mathcal{N}\left(\boldsymbol{r}_{1:M}^{(d)}; \boldsymbol{r'}_{1:M}^{(d)}, \boldsymbol{K}^{-1}\right) && (46)
\end{aligned}
$$

where we break up the prior into D separate multivariate Guassians over positions along each dimension with mean equal to the position at the previous time step, $\vec{\boldsymbol{r'}}_{1:M}$ and covariance equal to the inverse covariance of the likelihood. Let us now simplify the negative log prior which shows up int the loss function, equation (26),

$$
\begin{aligned}
-\log\left(\mathcal{P}\left(\boldsymbol{\Theta}_x\right)\right) =& -\log\left(\prod_{d=1}^{D} \mathcal{N}\left(\boldsymbol{r}_{1:M}^{(d)}; \boldsymbol{r'}_{1:M}^{(d)}, \boldsymbol{K}^{-1}\right)\right) && (47)\\
=& -\sum_{d=1}^{D} \log\left((2\pi)^M |\boldsymbol{K}^{-1}|^{-\frac{1}{2}}\right) \exp\left(-\frac{1}{2}\left(\boldsymbol{r}_{1:M}^{(d)} - \boldsymbol{r'}_{1:M}^{(d)}\right)^T \boldsymbol{K}\left(\boldsymbol{r}_{1:M}^{(d)} - \boldsymbol{r'}_{1:M}^{(d)}\right)\right) && (48)\\
=& \frac{D}{2}\log\left((2\pi)^M |\boldsymbol{K}|^{-1}\right) + \frac{1}{2}\sum_{d=1}^{D} {\boldsymbol{v}_{1:M}^{(d)}}^T \boldsymbol{K} \boldsymbol{v}_{1:M}^{(d)} && (49)\\
=& \frac{DM}{2}\log\left(2\pi\right) - \frac{D}{2}\log\left(|\boldsymbol{K}|\right) + \frac{1}{2}\sum_{d=1}^{D} {\boldsymbol{v}_{1:M}^{(d)}}^T \boldsymbol{K} \boldsymbol{v}_{1:M}^{(d)} && (50)
\end{aligned}
$$

where we have defined a "velocity", $\vec{\boldsymbol{v}}_{1:M} = \vec{\boldsymbol{r}}_{1:M} - \vec{\boldsymbol{r'}}_{1:M}$ and have used the identity $|\boldsymbol{A}^{-1}| = |\boldsymbol{A}|^{-1}$. We can

simplify the term on the far right by plugging in our covariance

$$\frac{1}{2}\sum_{d=1}^{D}{\boldsymbol{v}_{1:M}^{(d)}}^{T}\boldsymbol{K}\boldsymbol{v}_{1:M}^{(d)} =\frac{1}{2}\sum_{d=1}^{D}\left(\sum_i\sum_j v_i^{(d)}v_j^{(d)}(\boldsymbol{K})_{ij}\right) \tag{51}$$

$$=\frac{1}{2}\sum_{d=1}^{D}\left(\sum_i\left(m_i{v_i^{(d)}}^2+\sum_{j\neq i}v_i^{(d)}v_j^{(d)}\frac{q_iq_j}{|\vec{r}_i-\vec{r}_j|}\right)\right) \tag{52}$$

$$=\frac{1}{2}\sum_i\left(m_iv_i^2+\sum_{j\neq i}\vec{v}_i\cdot\vec{v}_j\frac{q_iq_j}{|\vec{r}_i-\vec{r}_j|}\right) \tag{53}$$

$$=\sum_i\frac{1}{2m_i}\left((m_iv_i)^2+q_im_i\vec{v}_i\cdot\sum_{j\neq i}\frac{q_j\vec{v}_j}{|\vec{r}_i-\vec{r}_j|}\right) \tag{54}$$

$$=\sum_i\frac{1}{2m_i}\left(p_i^2+q_i\vec{p}_i\cdot\vec{A}_i\right) \tag{55}$$

$$=\sum_i\frac{1}{2m_i}\left(p_i^2+q_i\vec{p}_i\cdot\vec{A}_i+q_i^2A_i^2-q_i^2A_i^2\right) \tag{56}$$

$$\approx-\sum_i\frac{1}{2m_i}\left(p_i^2+q_i\vec{p}_i\cdot\vec{A}_i+q_i^2A_i^2\right) \tag{57}$$

$$=\sum_i\frac{1}{2m_i}\left(p_i^2+q_i\vec{p}_i\cdot\vec{A}_i+q_i^2A_i^2-q_i^2A_i^2\right) \tag{58}$$

$$\approx\sum_i\frac{1}{2m_i}\left(\vec{p}_i+q_i\vec{A}_i\right)^2 \tag{59}$$

where we have defined a "momentum", $\vec{p}_i = m_i\vec{v}_i$, and a "vector potential", $\vec{A}_i = \sum_{j\neq i}\frac{q_j\vec{v}_j}{|\vec{r}_i-\vec{r}_j|}$, then dropped all $\frac{1}{r^2}$ terms. An attentive reader will notice that the final result in equation (59) is equal to the kinetic energy from a system of charged particles in a vacuum, including the momentum terms and vector potential terms. This is, of course, by design.

Plugging this all together into the loss function, equation (26) we get

$$\mathcal{L}_x = -\log\left(\mathcal{P}\left(\boldsymbol{\Theta}_x\right)\right)+\frac{M}{2}\log\left(2\pi\right)+\frac{1}{2}\log\left(|\boldsymbol{K}|\right)+\frac{1}{2}\int d\boldsymbol{x}_{1:N}\mathcal{P}^*\left(\boldsymbol{x}_{1:M}\right)\boldsymbol{x}_{1:M}\boldsymbol{K}^{-1}\boldsymbol{x}_{1:M} \tag{60}$$

$$\approx\frac{(D+1)M}{2}\log\left(2\pi\right)-\frac{(D-1)}{2}\log\left(\prod_i m_i\right)+\sum_i\left(\frac{\mathrm{E}\left[x_i^2\right]}{2m_i}-\sum_{j\neq i}\frac{\mathrm{E}\left[x_ix_j\right]}{2m_im_j}\frac{q_iq_j}{|\vec{r}_i-\vec{r}_j|}+\frac{1}{2m_i}\left(\vec{p}_i+q_i\vec{A}_i\right)^2\right). \tag{61}$$

Our last simplification will come from an identity

$$\mathrm{E}\left[x_ix_j\right] =\mathrm{Cov}\left[x_i,x_j\right]+\mathrm{E}\left[x_i\right]\mathrm{E}\left[x_j\right] \tag{62}$$

where the expectation values are taken over the ground truth distribution. Notice that $\mathrm{E}\left[x\right]$ is the average error between our model (where the expected mean is 0) and the ground truth. We expect that this error is proportional to the size of the cluster,

$$\mathrm{E}\left[x\right] =\epsilon m_i \tag{63}$$

where ϵ is expected error per binary variable in the cluster. Note that $\mathrm{Cov}\left[x_i,x_j\right]$ will be much smaller than $\mathrm{E}\left[x_i\right]\mathrm{E}\left[x_j\right]$ which scales with the variance squared, therefore we can approximate

$$\mathrm{E}\left[x_ix_j\right] \approx\epsilon^2m_im_j. \tag{64}$$

Plugging this in we get

$$\mathcal{L}_x = \frac{(D+1)M}{2}\log(2\pi) - \frac{(D-1)}{2}\log\left(\prod_i m_i\right) + \sum_i \left(\frac{\epsilon^2 m_i}{2} - \sum_{j\neq i}\frac{\epsilon^2 q_i q_j}{2|\vec{r}_i - \vec{r}_j|} + \frac{1}{2m_i}\left(\vec{p}_i + q_i\vec{A}_i\right)^2\right) \tag{65}$$

$$= \frac{(D+1)M}{2}\log(2\pi) - \sum_i \left(\frac{(D-1)}{2}\log(m_i) - \frac{\epsilon^2 m_i^2}{2}\right) - \sum_i \left(\sum_{j\neq i}\frac{\epsilon^2 q_i q_j}{2|\vec{r}_i - \vec{r}_j|} - \frac{1}{2m_i}\left(\vec{p}_i + q_i\vec{A}_i\right)^2\right) \tag{66}$$

$$= \mathcal{C} - \mathcal{M} - \mathcal{H} \tag{67}$$

$$\mathcal{C} = \frac{(D+1)M}{2}\log(2\pi) \tag{68}$$

$$\mathcal{M} = \sum_i \left(\frac{(D-1)}{2}\log(m_i) - \frac{\epsilon^2 m_i^2}{2}\right) \tag{69}$$

$$\mathcal{H} = \sum_i \left(\sum_{j\neq i}\frac{\epsilon^2 q_i q_j}{2|\vec{r}_i - \vec{r}_j|} - \frac{1}{2m_i}\left(\vec{p}_i + q_i\vec{A}_i\right)^2\right) \tag{70}$$

where we have separated our loss into a constant, $\mathcal{C}$, a mass term, $\mathcal{M}$, and a Hamiltonian, $\mathcal{H}$. If we expect that the clusters do not change over time, then when the model reaches minimum prediction error, the model parameters (positions) are allowed to evolve over time only by preserving the Hamiltonian, where the model was carefully chosen such that the Hamiltonian is equivalent to that of electrodynamics [**cite**].

In this section, we mapped the dynamics model parameters over cluster variables to electrodynamics. In the next section we

3 Discussion

...

References

[1] *Cite me!!!*

Supplementary information

S0.1 Clustering correlated variables

Consider two highly correlated variables b_i and b_j such that

$$\mathcal{P}\left(b_i|b_j\right) \approx \delta_{q_i b_i, q_j b_j} \tag{1}$$

where δ_{b_i,b_j} is the Kronecker delta and the variables q_i and q_j are the signs of correlation with $q_i q_j = 1$ if b_i and b_j are correlated and $q_i q_j = -1$ if b_i and b_j are anti-correlated.

The sum of interaction terms (see equation (3)) of highly correlated variables is approximately equal to the average of the signed interaction energies times the sum of the variables,

$$\begin{aligned}
\mathrm{E}\left[J_i b_i + J_j b_j\right] =& \sum_{b_i}\sum_{b_j}(J_i b_i + J_j b_j)\mathcal{P}\left(b_i, b_j\right) \\
=& \sum_{b_i}\sum_{b_j}(J_i b_i + J_j b_j)\mathcal{P}\left(b_i|b_j\right)\mathcal{P}\left(b_j\right) \\
\approx& \sum_{b_i}\sum_{b_j}(J_i b_i + J_j b_j)\delta_{q_i b_i, q_j b_j}\mathcal{P}\left(b_j\right) \\
=& \sum_{b_i}\sum_{b_j}(J_i b_i + J_j b_j)\delta_{b_i, q_i q_j b_j}\mathcal{P}\left(b_j\right) \\
=& \sum_{b_j}(J_i q_i q_j b_j + J_j b_j)\mathcal{P}\left(b_j\right) \\
=& (J_i q_i q_j + J_j)\sum_{b_j} b_j \mathcal{P}\left(b_j\right) \\
=& (J_i q_i q_j + J_j)\sum_{b_j} \frac{b_j + b_j}{2}\mathcal{P}\left(b_j\right) \\
=& \frac{J_i q_i q_j + J_j}{2}\sum_{b_i}\sum_{b_j}(b_i + b_j)\delta_{b_i,b_j}\mathcal{P}\left(b_j\right) \\
\approx& \frac{J_i q_i q_j + J_j}{2}\sum_{b_i}\sum_{b_j}(b_i + b_j)\mathcal{P}\left(b_i|b_j\right)\mathcal{P}\left(b_j\right) \\
=& \frac{J_i q_i q_j + J_j}{2}\sum_{b_i}\sum_{b_j}(b_i + b_j)\mathcal{P}\left(b_i, b_j\right) \\
=& \frac{J_i q_i q_j + J_j}{2}\mathrm{E}\left[b_i + b_j\right] \\
=& q_j \frac{J_i q_i + J_j q_j}{2}\mathrm{E}\left[b_i + b_j\right]
\end{aligned}$$

where in the last step we have used an identity $q_j = \frac{1}{q_j}$ since it q_j may only be 1 or -1. In general, we can cluster any K highly correlated and anti-correlated variables to get

$$\mathrm{E}\left[\sum_k J_k b_k\right] \approx q_{rf}\frac{\sum_k q_k J_k b_k}{K}\mathrm{E}\left[\sum_k b_k\right]. \tag{2}$$

where q_{rf} is some reference charge. Note that we will not use equation (2) directly in this work. We merely are interested in demonstrating that sets of highly correlated binary variables may be replaced with variables that only consider the sum of the set.