Discovering Unexpected Local Nonlinear Interactions in Scientific Black-box Models

Michael Doron
michael.doron@mail.huji.ac.il
The Hebrew University of Jerusalem
Jerusalem, Israel

Idan Segev
idan@lobster.ls.huji.ac.il
The Hebrew University of Jerusalem
Jerusalem, Israel

Dafna Shahaf
dshahaf@cs.huji.ac.il
The Hebrew University of Jerusalem
Jerusalem, Israel

ABSTRACT

Scientific computational models are crucial for analyzing and understanding complex real-life systems that are otherwise difficult for experimentation. However, the complex behavior and the vast input-output space of these models often make them opaque, slowing the discovery of novel phenomena. In this work, we present Hint (Hessian INTerestingness) – a new algorithm that can automatically and systematically explore black-box models and highlight local nonlinear interactions in the input-output space of the model. This tool aims to facilitate the discovery of interesting model behaviors that are unknown to the researchers. Using this simple yet powerful tool, we were able to correctly rank all pairwise interactions in known benchmark models and do so faster and with greater accuracy than state-of-the-art methods. We further applied Hint to existing computational neuroscience models, and were able to reproduce important scientific discoveries that were published years after the creation of those models. Finally, we ran Hint on two real-world models (in neuroscience and earth science) and found new behaviors of the model that were of value to domain experts.

CCS CONCEPTS

• **Information systems → Data mining**; • **Computing methodologies → Simulation evaluation**; *Unsupervised learning*.

KEYWORDS

Nonlinear interactions; interestingness; simulation; computational models; neuroscience

ACM Reference Format:
Michael Doron, Idan Segev, and Dafna Shahaf. 2019. Discovering Unexpected Local Nonlinear Interactions in Scientific Black-box Models. In *The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '19), August 4–8, 2019, Anchorage, AK, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3292500.3330886

1 INTRODUCTION

Over the decades, the scientific community developed an abundance of computational models that range from abstract to highly realistic. Despite the fact that many models have predicted phenomena decades before they were confirmed experimentally [21, 40], the complexity of models and the size of their input-output space makes serendipitous discoveries difficult. This reliance on manual exploration limits scientists' ability to discover novel phenomena in computer models that are otherwise not bound by experimental limitations such as technological accessibility or animal lives.

In this work, we present Hint (Hessian INTerestingness), a tool to **automatically and systematically explore and highlight potential interesting phenomena in computational models**. This tool enables an exploration of models' input-output spaces similar to how one would explore a real system via experimentation; it thus better utilizes the highly realistic computational models at our disposal. By highlighting potential emergent phenomena in computational models, we are able to both produce a list of suggested future experiments, as well as easily find the parameter regions where the model fails to reproduce desired outcomes, which can aid in model fine-tuning and debugging.

Our work is informed by considerable advances in the field of scientific automation, such as active learning [43] and automated experimentation [11, 26]. The emergence of accurate yet opaque black-box models, such as deep neural networks, inspired the study of model interpretation to better explore those systems [32, 35]. This automation of exploration requires a theoretical basis to rigorously define what is interesting to explore. The works of [25, 38] suggest that an observed sample is interesting if the observers update their understanding of the system after viewing the sample.

Using Hint, we wish to promote the discovery of interesting phenomena by highlighting regions in the feature space where the model behaves not as expected. Thus, we need to formally define "expected". We focus on the case where the researchers have full knowledge of the effect each feature has on the output while having little knowledge of how these features interact with one another with respect to the output. This is the case, for example, in the common situation where each feature is modular and can be modeled separately (such as individual ion channels in neuron models [3], or various physical constraints in climate models [36]) and later combined to a more complex realistic model [18].

Thus, when constructing the researcher's prior of the model, we heuristically define interestingness as the amount of local nonlinear interaction between features. Building on this heuristic, we create nonlinear interaction maps for each feature pair across the parameter space. This allows us to accurately identify and rank the interacting pairs, locate nonlinear phenomena, and be robust with a very small sample size.

Other methods have worked on the problem of detecting nonlinear interactions in the past [13, 14, 19, 20, 30, 50], although their

KDD '19, August 4–8, 2019, Anchorage, AK, USA

ACM ISBN 978-1-4503-6201-6/19/08…$15.00
https://doi.org/10.1145/3292500.3330886

Algorithm 1 HINT

procedure CREATE_HESSIAN
  Require: $f(\overline{x}) \leftarrow$ A black-box model, $f(\overline{x}) : \mathbb{R}^m \rightarrow \mathbb{R}$
  Require: $N_c \leftarrow$ Number of core samples
  Require: *featureLimits* $\leftarrow$ (min, max) per feature m
  for $s \in N_c$ **do**
    // Sample uniformly between lower and upper bounds
    $\overline{x}_s \leftarrow Uniform(featureLimits)$
    // Run simulator on sample and perturbations
    $\overline{y}_s \leftarrow f(\overline{x}_s)$
    for $(i, j) \in \binom{m}{2}$ **do**
      $\overline{y}_{s_{perturb_i}} \leftarrow f(\overline{x}_s + \Delta x_i)$
      $\overline{y}_{s_{perturbj}} \leftarrow f(\overline{x}_s + \Delta x_j)$
      $\overline{y}_{s_{perturbij}} \leftarrow f(\overline{x}_s + \Delta x_i + \Delta x_j)$
      // Normalize data between 0 and 1
      $\overline{y} \leftarrow normalize(\overline{y})$
      // Calculate Hessian using forward difference
      $H(\overline{x}_s)_{ij} \leftarrow \frac{\overline{y}_s - \overline{y}_{s_{perturbi}} - \overline{y}_{s_{perturbj}} + \overline{y}_{s_{perturbij}}}{\Delta x_i \cdot \Delta x_j}$
procedure RANK_LOCAL_INTERACTIONS
  Require: $H(\overline{x}_s) \leftarrow$ created with *CREATE_HESSIAN*
  Require: *threshold* $\leftarrow$ Threshold in SD units
  // Denoise Hessians by rectifying top and bottom 0.1%
  $H \leftarrow denoise(H)$
  // Normalize Hessians by their standard deviation
  $H_{SD} \leftarrow \frac{H}{SD(H)}$
  // return filter whose activation crosses the threshold
  for *fSize* $\in [\sqrt{N}, \sqrt{N} - 1, ..., 2, 1]$ **do**
    if $max(\frac{1}{fSize^2} \sum_{s \in fSize} |H_{SD_{ij}}(s)|) > threshold$ **then**
      return *fSize*
  return -1

approach was either data-driven [14, 30] or not local [19], unlike our local model-driven approach.

Our main contributions are:

- We develop HINT, a tool that facilitates the discovery of local nonlinear interactions between parameters in complex black-box simulators.
- Despite the algorithm's simplicity, we demonstrate that HINT is able to outperform state-of-the-art methods on known benchmarks with running time far smaller than all other methods. In addition, HINT is easy to parallelize and can scale with computational resources.
- We demonstrate how to use HINT to explore computational models in neuroscience. Importantly, we **reproduce scientific phenomena** by running HINT on models that existed years prior to the original discoveries. In both of the cases we tried, the phenomena ranked at the top of HINT's results.
- Finally, we use HINT in an exploratory setting for two realistic and complex models in **computational neuroscience** and **earth science**. We demonstrate how we are able to find complex behaviors that are of value to the domain experts.

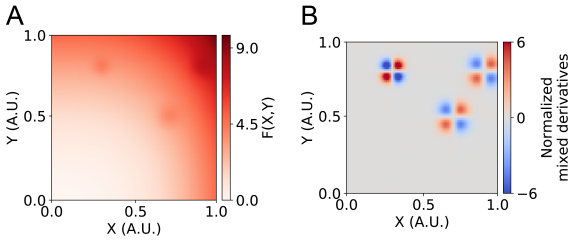

Figure 1: HINT highlights local nonlinear interactions in a black-box simulator. (A) In this generated data example, there are three local nonlinear interactions, and two single feature effects nearly obfuscating them. (B) When running HINT on the model, we highlight the local nonlinear interactions, presenting their feature regions to the researcher.

2 THE ALGORITHM

2.1 Problem definition

Let $f(\overline{x}) : \mathbb{R}^m \rightarrow \mathbb{R}$ be a black-box simulator that receives a feature vector $\overline{x} \in \mathbb{R}^m$ and outputs a scalar result y (or raw output r from which scalar feature y is extracted). For example, in a conductance-based model of a neuron [18], features can include ion channel densities and the output could be the number of somatic spikes.

Let f^* denote the researchers' prior beliefs of the model. We assume that researchers know the independent effect each feature has on the output. However, based on the modular structure of computational models, non-linear interactions between the features are not necessarily easily predicted. For example, a researcher might know the response to voltage of each voltage-dependent ion channel, but still not predict the emergent behavior of a neuron with several interacting ion channels. In this work, we set the researchers' prior to be such that the combined effect of any two features is equal to the linear summation of the effects of each individual feature. Let $f_j(\overline{x}, \Delta x_j) = f(\overline{x} + \Delta x_j) - f(\overline{x})$. We assume the researcher believes $f^*(\overline{x} + \Delta x_i + \Delta x_j) = f(\overline{x}) + f_i(\overline{x}, \Delta x_i) + f_j(\overline{x}, \Delta x_j)$ for every $i \neq j$. Thus, when calculating the mixed derivative $\frac{\delta f^*(\overline{x})}{\delta x_i \delta x_j}$, we get $\frac{\delta f^*(\overline{x})}{\delta x_i \delta x_j} = \frac{f^*(\overline{x}+\delta x_i+\delta x_j) - f(\overline{x}+\delta x_i) - f(\overline{x}+\delta x_j) + f(\overline{x})}{\delta x_i \cdot \delta x_j} = \frac{f(\overline{x}) + f_i(\overline{x}, \delta x_i) + f_j(\overline{x}, \delta x_j) - f(\overline{x}+\delta x_i) - f(\overline{x}+\delta x_j) + f(\overline{x})}{\delta x_i \cdot \delta x_j} = \frac{f(\overline{x}) + f(\overline{x}+\delta x_i) - f(\overline{x}) + f(\overline{x}+\delta x_j) - f(\overline{x}) - f(\overline{x}+\delta x_i) - f(\overline{x}+\delta x_j) + f(\overline{x})}{\delta x_i \cdot \delta x_j} = 0.$

2.2 Our approach

To calculate where a sample does not fit the researchers' prior, we approximate the Hessian (the matrix of second-order partial derivatives) of the output of the simulator with respect to every feature pair $(x_i, x_j) \in \binom{m}{2}$ $(H(\bar{x})_{ij} = \frac{\delta^2 f}{\delta x_i \delta x_j}$, see Algorithm 1). As shown in Section 2.1, we assume the researcher believes the mixed derivative $H(f^*(\overline{x}))_{ij} = 0$ for every $i \neq j$. To find the input-output regions where the model does not fit the researchers' prior, we define the interestingness of a sample by the absolute value of $H(\bar{x})_{ij}$ - the more nonlinear is the interaction between features i and j the more interesting it is to the researcher. In our previous example, the researcher who studies the effects of ion channels might be surprised and interested to find that certain regions in the density space

cause ion channels to increase each other's effect on the number of spikes. Because we do not have f as an analytic function, but only as a black-box simulator, we cannot derive the Hessian analytically. However, with the model, we use our ability to sample as we wish (Alg. lines 3-8) and to compute the mixed derivative using forward difference method ($\frac{f(x)-f(x+\Delta x_i)-f(x+\Delta x_j)+f(x+\Delta x_i+\Delta x_j)}{\Delta x_i \cdot \Delta x_j}$, Alg. line 10). The result is used to rank the samples according to their and their surroundings nonlinearity, leading to the discovery of novel emergent phenomena (see Figure 1).

In order to highlight the samples where the model does not fit the researchers' prior beliefs, we employ two methods of ranking: **global** pairwise interaction rank and **local** interaction rank. For the global ranking, we calculate the mean across core samples of the absolute mixed derivative for every feature pair $rank_{ij} = \frac{1}{N_c}\sum_{s=1}^{N_c}|H_{ij}(\bar{x}_s)|$. However, the global ranking could not distinguish between feature pairs that interact strongly only at one confined region or feature pairs that interact weakly across the entire input-output space. To highlight the features that interact strongly at a specific region, we first denoise the Hessian from outliers by rectifying the top and bottom 0.1% (Alg. line 12), then normalize it by its standard deviation for all samples and all feature pairs. For each feature pair, we then iteratively run an average spatial filter of decreasing size (Alg. lines 14-16). We take the maximum activation for each filter size and receive a monotonically increasing function. The local phenomena ranking is the first filter size that crossed a predefined threshold that controls the sensitivity of the local ranking measurement (see below for discussion on threshold choice). This simple yet powerful method allows us to correctly rank all global pairwise interactions in existing benchmarks (Section 3.2, Figure 2A-D), as well as find the local nonlinear interactions with high degrees of precision and recall (Figure 2E-F).

Hyperparameters. Our algorithm includes several hyper-parameters that a researcher can adjust. N_c, or the **core sample number**, is the number of uniformly sampled feature vectors around which we look for nonlinearities. For global interaction ranking in systems where the nonlinearity does not change locally, a single core sample $N_c = 1$ is sufficient to correctly classify all feature pairs. For local interactions, the larger is N_c, the higher is the accuracy.

m, the **number of features**, controls which features will be perturbed and which feature pairs will be ranked. Excluding features from the algorithm will make it run faster, but prevent the discovery of new local or global interactions using these features.

Δx_i, the **step size**, determines the distance of the perturbation from the core sample, written as a fraction of the feature range. This step should be small, as the approximation error is $\frac{\delta^2 f}{\delta x_i \delta x_j} - \frac{f(x)-f(x+\Delta x_i)-f(x+\Delta x_j)+f(x+\Delta x_i+\Delta x_j)}{\Delta x_i \cdot \Delta x_j} \approx O(\Delta x_i \Delta x_j)$. In this work we chose Δx_i to be 1% in the data generation phase to keep the approximation error small. We present the results using $\Delta x_i = 10\%$ to better illustrate the effect of perturbing the parameters.

Finally, the **threshold** in the local interaction ranking determines the value required after averaging the normalized Hessian of all samples within a certain filter size. The lower the value, the more sensitive the ranking is to small derivatives and the higher are the scores of the local interaction pairs. We used a threshold of three standard deviations to reduce the chance of spurious interactions.

Robustness. Hint retains its accuracy for various nonlinear interaction types (Figure 2A) and for various sample sizes (Figure 2B). See details in the next section. It is, however, susceptible to intrinsic noise in the model (i.e., when sampling the same feature input vector returns different results each time). This reduction in Hint performance can be handled under the assumption that the noise has zero mean, by averaging over several samples from the same feature vector while using a larger step size Δx_i (see Figure 2C).

Complexity analysis. Hint samples N_c core samples (Alg. line 3), in addition to N_c local perturbations for each of the m features and N_c perturbations for each feature pair, s.t. $n = N_c \cdot (m^2 + m + 1)$ (Alg. lines 4-9). Then, it numerically computes N_c local Hessian matrices, each taking $O(m^2)$ (Alg. line 9). Thus, the time complexity is $O(N_c \cdot m^2)$, or $O(n)$. The space used is the sampled dataset of $N_c \cdot (m^2 + m + 1)$ samples, each consisting of a feature vector of size m and a result scalar, making it $O(N_c \cdot m^3)$. The m^2 local interaction maps use N_c samples each, making the space complexity $O(N_c \cdot m^3)$ or $O(n \cdot m)$. Note that the algorithm is "embarrassingly parallel" and scales linearly with the number of machines. In practice, Hint is very fast (see Section 3.1.2 for details).

3 RESULTS

3.1 Synthetic data

To test the accuracy of Hint, we ran experiments on four synthetic models: three discussed in the literature and one of our own design. We compared our results on the known models to previous benchmarks and report the results here.

3.1.1 Benchmarks. To the best of our knowledge, no other method exist that ranks local nonlinear interactions directly in black-box models. However, several data-driven works exist who rank interactions on given data. We use them here as a baseline for comparison.

- ANOVA fits a Generalized Additive Model (GAM) [17] to the data with all single parameters and pairwise interactions. It then calculates each interaction strength by its p-value [50].
- PDP [14] calculates the H-Statistic, which can be defined as ($H^2_{jk} = \frac{\sum_{i=1}^{n}\left[\hat{F}_{jk}(x_{ij},x_{ik})-\hat{F}_j(x_{ij})-\hat{F}_k(x_{ik})\right]^2}{\sum_{i=1}^{n}\hat{F}^2_{jk}(x_{ij},x_{ik})}$), $\hat{F}$ being the partial dependency plot. In practice, PDP often uses gradient boosting. It is possible to run PDP directly on the model, while finding only global interactions (worse but comparable to Hint's performance in global interaction detection).
- GA2M [30] fits a GAM to the data using shallow trees and iteratively search for areas where the additive model does not capture the data, ranking these parameter pairs as nonlinear.

For fair comparison with the methods that do not benefit from data that is structured as Hint requires, we uniformly sampled an equal size dataset of $n = N_c \cdot (m^2 + m + 1)$ scattered samples for the benchmarks to rank (running these benchmarks on the original structured data reduced their performance, as the mixed effects were far smaller than the total signal).

3.1.2 Global pairwise interaction. We used three synthetic models in our experiments. The first, **"Complex function"** [19], is

$$F(x) = \pi^{x_1 x_2}\sqrt{2x_3} - sin^{-1}(x_4) + log(x_3 + x_5) - \frac{x_9}{x_{10}} - x_2 x_7$$

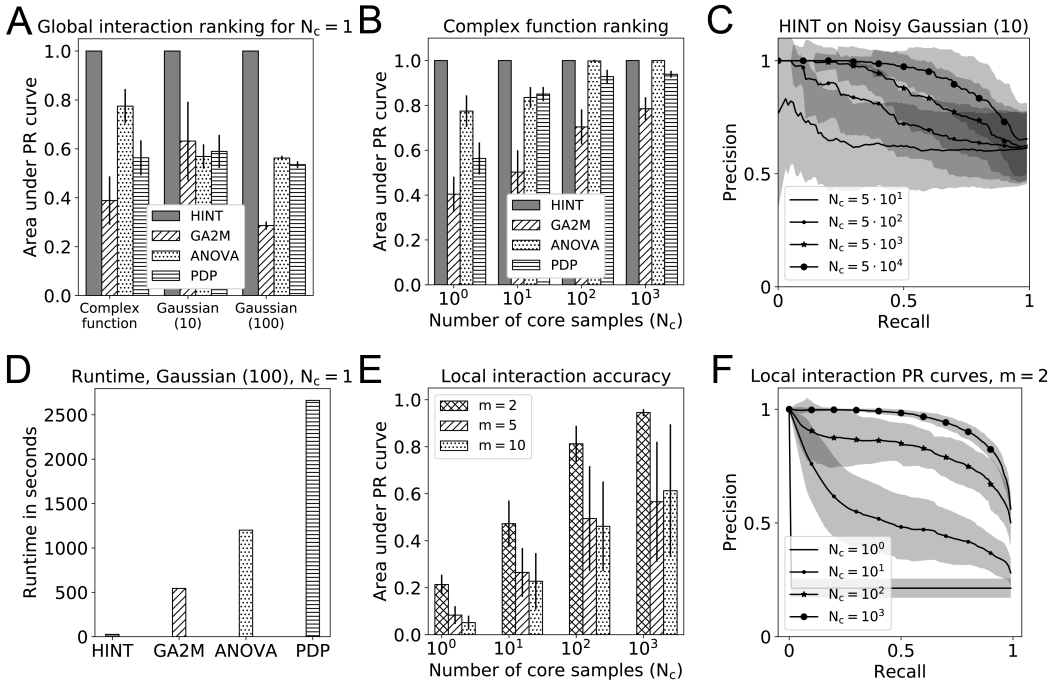

Figure 2: Evaluation on synthetic data. A. Average area under the precision-recall curve on global interaction models, using $N_c = 1$, for the complex function, Gaussian (10) and Gaussian (100) from Section 3.1.2. HINT reached 100% accuracy using a single core sample for all models. B. Average area under the PR curve on the Complex function for different number of N_c samples. The benchmarks required orders of magnitude more samples to reach the same accuracy as HINT. C. Precision-Recall curves for the performance of HINT on Gaussian (10) with added noise. A single feature vector and its perturbations were sampled N_c times, and the Hessian approximation was done on their averages. D. Runtime for ranking the Gaussian (100) model using $N_c = 1$. E. HINT accuracy in detecting the exact location of nonlinear phenomena in a randomly generated model defined in Section 3.1.3. HINT performance increases with the number of core samples. F. The Precision-Recall curves of the performance of HINT in detecting local nonlinearities for $m = 2$.

Parameters x_4, x_5, x_8, x_{10} were sampled from $U(0.6, 1.0)$ and the rest were sampled from $U(0.0, 1.0)$.

The second synthetic model, **Gaussian (10)**, was a summation of randomly generated functions, as described in [13]. For each iteration, we generated a summation of 25 interactions between subsets of 10 parameters. The third synthetic model, **Gaussian (100)**, was created similarly to Gaussian (10) but using 1000 interactions between subsets of 100 parameters.

In Figure 2A-C we show HINT's performance in detecting all feature pairs with nonlinear interactions. Using $N_c = 1$ and $m^2 + m$ perturbations around it, HINT was able to correctly classify all interacting pairs in the models described above, with an average area under the precision-recall curve of 1. As exhibited in Figure 2A, HINT outperforms all other compared methods.

In Figure 2B, we show that HINT retains its advantage in larger sample sizes. When classifying pairs in the Complex function model, the next most accurate method required $n = 11100$ samples to detect all interacting feature pairs, while HINT required only $n = 111$ (a single N_c and $m^2 + m$ perturbations).

To test HINT's vulnerability to noise in Figure 2C, we used the previously defined Gaussian (10) model with an added Gaussian noise with mean 0 and variance of $\frac{1}{20}$ of the signal variance. We set $\Delta x_i = 0.1$, and ran the simulator on a single randomly sampled feature vector $N_c \in \{5 \cdot 10^1, 5 \cdot 10^2, 5 \cdot 10^3, 5 \cdot 10^4\}$ times. We averaged over the values of the core sample and approximated the Hessian with the perturbations around it, which also averaged over N_c iterations. As displayed in Figure 2C, the accuracy of HINT increased with the number of iterations in the case of noise.

Running time. We compared HINT's running time to benchmarks on the Gaussian (100) model using $N_c = 1$ ($n = 10101$) samples. As exhibited in Figure 2D, HINT required substantially less running time than the next fastest method, 26 seconds compared to 544 seconds and had much greater accuracy. Note that we added the sampling time for HINT while removing it from the other methods' time as they work on pre-existing datasets and do not require sampling structure. Given data, HINT required 0.5 seconds to correctly rank all 4950 putative interacting pairs in the Gaussian (100) model.

3.1.3 Local interaction. Unlike the previous existing models that tested global ranking, here we suggest a new benchmark created to test local interaction detection, similar to the global interaction model created by [13]. In this model, several local multivariate

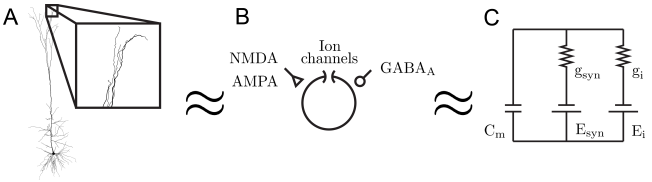

Figure 3: Model of a neuron used in Section 3.2. A. Full reconstruction of a layer-5 pyramidal cell. Dendrite is shown in the inset. B. The dendrite can be approximated as a single compartment with ion channels and excitatory and inhibitory synaptic inputs. C. This approximation can be modeled as a single compartment RC circuit.

Gaussian functions are randomly sampled with random magnitudes and centers. Unlike the global interaction model, the features in this model interact only in a small portion of the feature space, making the discovery of nonlinear interactions more difficult. For modeling details, see Reproducability Appendix A.2.5.

In (Figure 2E-F we show Hint's performance in detecting the local interactions for each sample and for each feature pair. To allow Hint to detect interactions in the entire feature space despite sampling a finite amount, we used linear interpolation of the absolute mixed derivatives for each feature pair. We ran 30 random iterations of this model, each with number of features $m \in \{2, 5, 10\}$, and number of core samples $N_c \in \{1, 10, 100, 1000\}$. The number of phenomena was set to be three and the magnitudes were randomly sampled from $U(10, 20)$. Our performance increased with the number of samples, reaching near 95% (Figure 2E-F).

3.2 Reproductions of Scientific Discoveries

Our goal is to design a method for researchers to easily identify interesting unknown behaviors of their models, thus facilitating the discovery of putative real phenomena. We applied Hint on realistic neuron models used in computational neuroscience to test if it finds phenomena that are of interest to the scientific community. We chose two cases of models commonly used for between seven years [18] and *two decades* [23], and demonstrate how applying Hint to them finds two discoveries published in 2017. For the first example, we looked for a simple well-known system that still hid an interesting nonlinear interaction. For the second example, we chose a more complex system, testing Hint's ability to detect an interesting behavior among many possible interactions. In both cases, we wished to see if Hint could replicate a discovery that was made experimentally in a highly nonlinear system using only a computational model of that system.

3.2.1 Realistic neuron modelling framework. In this section, we search for highly nonlinear interactions between various cellular mechanisms in computational models that simulate the behavior of single neurons. A common approach in computational neuroscience is to approximate the full multi-compartmental neuron (Figure 3A) as a single compartmental model, consisting of membrane ion channels and excitatory and inhibitory synapses (Figure 3B):

$$C_m \cdot \frac{dV}{dt} = \sum_i g_i \cdot (V(t) - E_i)$$

This compartment can be modeled as an RC circuit (Figure 3C), with the ion channel/synaptic conductance g_i representing the membrane resistivity, the ion channel/synaptic reversal potential E_i modeled as batteries, and C_m representing the membrane capacitance (see Appendix A.4 for a more in-depth description of the models). As g_i is a function that can be dependent on voltage, time and/or other ion channel currents, this model can be highly nonlinear and generate complex emergent phenomena.

3.2.2 Reproduction of the effect of timed synaptic inhibition on the NMDA spike.

Background. The first system we studied was the interaction between excitatory and inhibitory synaptic inputs to the dendrites, where the neuron receives inputs (See inset in Figure 3A). Pyramidal cells receive a complex spatio-temporal pattern of inputs that increase (depolarize) and decrease (hyperpolarize) the membrane voltage [28]. These dendritic inputs are integrated at the soma to generate action potentials and communicate with other neurons. While some of these inputs have a relatively linear effect on the membrane voltage, like the excitatory *AMPA* and the inhibitory $GABA_A$ synapses, some are highly nonlinear, like the *NMDA* synapse [23]. The current flowing through the *NMDA* synapse depends on both the *NMDA* activation time and the local voltage, allowing it to create a regenerative phenomenon called the "NMDA spike" [37]. The NMDA spike is involved in learning and computation and has been studied for more than a decade [1, 31, 37]. Although the model for the NMDA current existed for over two decades [23], the nonlinear interactions between the NMDA synapse and other cellular mechanisms still allow new discoveries to be made [8, 9, 22].

Phenomenon. A recent paper [8] found that inhibitory input has a highly nonlinear and surprising interaction with the NMDA spike: while the NMDA spike was resistant to inhibition arriving at its onset, it became more vulnerable the later inhibition arrived. To see if Hint could automatically capture this finding, we used a simple model simulating a dendrite receiving inputs (see Appendix A.4.1). The features were the strength (i.e. the conductance) of the excitatory *NMDA* synapse, the strength of the inhibitory $GABA_A$ synapse, and the time difference between the two, making our feature dimension $m = 3$. The feature limits were based on realistic synaptic conductance ranges and time difference between local inputs [18]. For the output scalar function, we chose the time integral of the membrane voltage, which is a good indicator of the strength of the NMDA spike [37]. We randomly sampled $N_c = 3000$ feature vectors and $N_c \cdot (\binom{m}{2} + m) = 18000$ permutation input vectors, resulting in $n = 21000$ samples. We simulated each input vector using the NEURON simulation software [7] and extracted the membrane voltage time integral as described in Appendix A.4.1. Next, we approximated the Hessian of the core N_c samples using forward difference method.

Discoveries. The most nonlinear pair Hint found was the *NMDA synaptic strength* and the *time difference between the NMDA activation and the $GABA_A$ activation* (Figure 4A). Out of the entire feature space, the most nonlinear sample Hint found was the one that exhibited the phenomena reported in [8]: In Figure 4B the *NMDA* spike is on the threshold between regeneration and termination. An increase in the time difference caused the inhibition to terminate

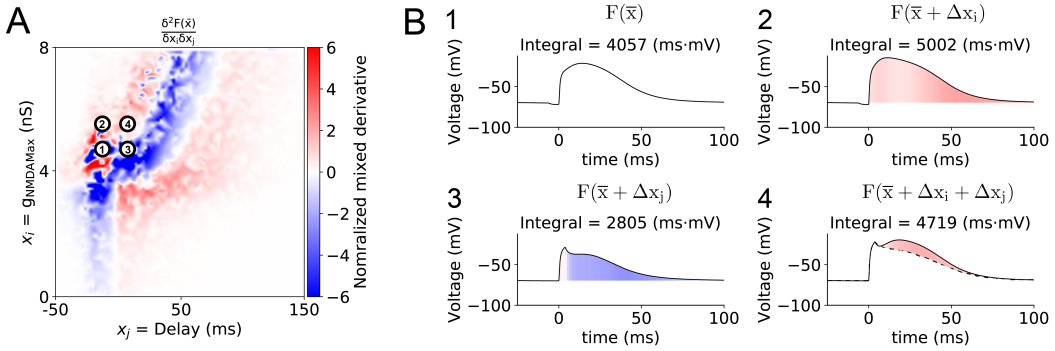

Figure 4: Top nonlinear interaction automatically found in the NMDA/GABA model (Section 3.2). A. Mixed derivative map of the integral of the membrane voltage with respect to the time difference between excitation and inhibition (x-axis) and the conductance of the *AMPA/NMDA* receptors (y axis). Red/blue areas represent samples that had a supra-linear/sub-linear interaction. Colorbar is normalized to mixed derivative SDs, and rectified between [-6, 6] SDs for clarity. Dot 1 represents the most nonlinear sample. Dots 2-4 are the perturbations around it. B. Raw simulator output on dots 1-4. The dendritic voltage trace returned by running the model on the most nonlinear sample (upper left), after perturbing *NMDA* (upper right), *Delay* (lower left), and both (lower right, solid line for double perturbation result, dashed line for linear summation of the effects of the two single perturbations). Red/blue areas under the trace show where the voltage increased/decreased, respectively, compared to the original trace (for single perturbation) or compared to the linear summation (for double perturbation).

the NMDA spike, while an increase in the NMDA conductance allowed the NMDA spike to regenerate after inhibition. The increase in both features caused the NMDA spike to stay around the threshold, barely regenerating from the inhibition.

This vulnerability of the NMDA spike, here found automatically, was **the core finding** of [8], and was found experimentally by [9].

3.2.3 *Reproduction of the effect of Backpropagating action potentials on subsequent EPSP.*

Background. The second system we chose to test Hint on was the Backpropagating action potential (BAP) [42] and its influence on dendritic voltage. After an action potential is initiated at the axon, it backpropagates to the dendrites, depolarizing the membrane voltage. This change in voltage evokes ion channels that further impact dendritic computation and plasticity [41, 48]. However, due to the interplay between voltage-dependent ion channels, it is difficult to predict the effect the BAP will have on this system.

Phenomenon. In [24], the researchers found that an excitatory postsynaptic potential (EPSP) arriving after a BAP is smaller than an EPSP arriving without a BAP proceeding it, due to the activation of potassium channels that hyperpolarize the local voltage. We wished to test whether Hint could highlight this behavior when faced with a complex system of 12 interacting features. As before, we simulated a single compartment approximating a dendrite with leak channels, AMPA / NMDA synapses and $GABA_A$ synapses (see Appendix A.4.2). We added voltage- and ion-dependent ion channels found in an apical dendrite [18], making the model of the dendrite more realistic. Finally, we added a simulation of a BAP arriving to the dendrite, representing the distance of the dendrite from the soma by a feature controlling the attenuation of the BAP's height. Unlike the previous model, here we tried to simulate the ongoing *in-vivo* input arriving to the dendrite. To do that, we randomly sampled the activation times of the *AMPA* / *NMDA* and the $GABA_A$ synapses from Poisson distributions and set the rates to be features, making our feature dimension $m = 12$ (see Appendix A.4.2). As before, the raw output of the simulator was the membrane voltage and the chosen output scalar was the voltage time integral, which gave a unified indicator of the voltage throughout the simulation duration. We randomly sampled $N_c = 3000$ feature vectors and $N_c \cdot (\binom{m}{2} + m) = 234000$ permutation vectors, resulting in $n = 237000$ samples. We simulated each sample using the NEURON simulation software [7] and extracted the membrane voltage time integral.

Discoveries. When ranking the most nonlinear feature pairs, the top three were the *excitatory rate* and *inhibitory rate*, the *Leak channel conductance* and *Inhibitory input rate*, and the *BAP height* and *Exitatory input rate* (Figure 5A). While the first two were known [22], the interaction between the height of the BAP and excitatory inputs was found experimentally in biological neurons in 2017 [24]. However, using Hint, we were able to find it using models created in 2011 [18], showing that additional excitatory input arriving after a larger BAP has a much smaller impact compared to the same input arriving after a more attenuated BAP (Figure 5B). Most other interactions found were known in the literature and addressed the sublinear effect of multiple inhibitory currents.

4 EXPLORATION

In this section we set out to find putative *new* interesting discoveries. We start by exploring single neuron firing patterns (Section 4.1). We then test the applicability of Hint to domains other than neuroscience, and study a model from climate sciences (Section 4.2).

4.1 Single neuron firing patterns

We tested Hint on a long-standing question in neuroscience: How do ion channels contribute to the firing properties of neurons? [33]. Our goal was to search the feature space of ion channel densities

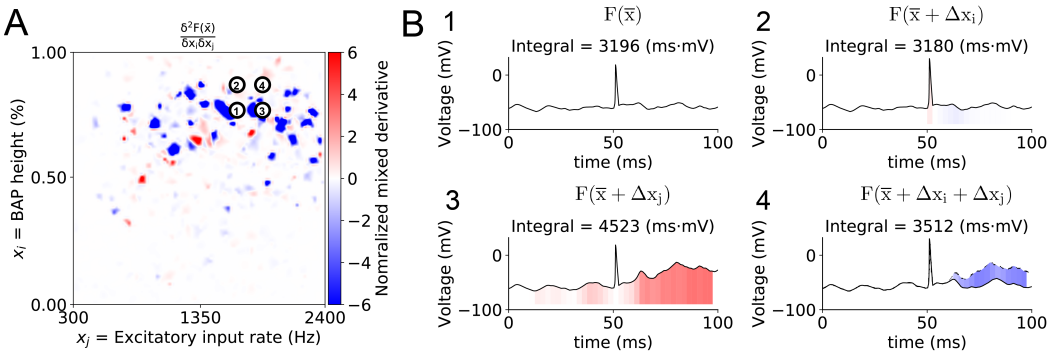

Figure 5: Nonlinear interaction automatically found in the BAP model (Section 3.2). A. Same as Figure 4A, with x-axis: excitatory input rate, and y-axis: height of the backpropagating action potential (BAP). B. Same as in Figure 4B. The somatic voltage trace returned on the most nonlinear sample (upper left), after perturbing BAP height (upper right), the excitatory input rate (lower left), and both (lower right). Red/blue areas under the trace show where the voltage increased/decreased, respectively, compared to the original trace (for single perturbation) or compared to the linear summation (for double perturbation).

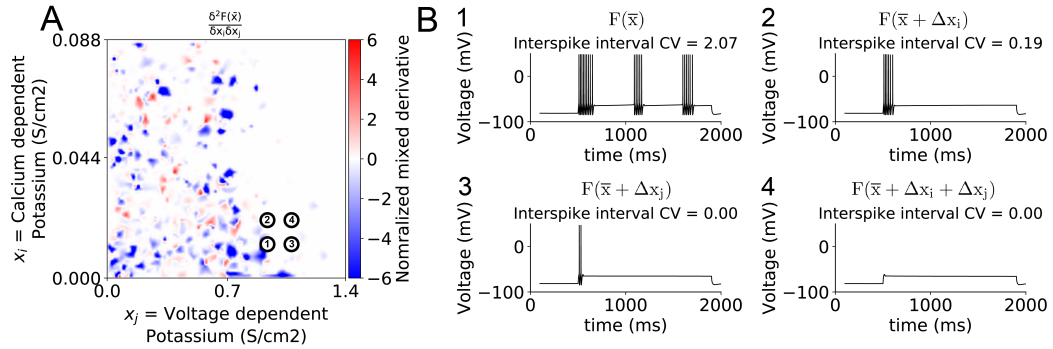

Figure 6: The top 3rd nonlinear interaction automatically found in the burst model (see Section 4.1). A. Same as in Figure 4A, with the two parameters being the voltage dependent potassium (x-axis) and the calcium dependent potassium (y-axis). B. Same as in Figure 4B. The somatic voltage trace returned by running the model on the most nonlinear sample parameter vector (upper left), after perturbing the voltage dependent potassium (upper right), the calcium dependent potassium (lower left), and both (lower right).

and find interesting firing behaviors that can be verified experimentally. As this system has highly nonlinear interplay between ion channels, it was ideal for testing HINT's ability to find interesting behaviors in a scientific computational system. In order to study the spiking behavior of the neuron, we simulated a model of a soma using the ion channels and passive properties described in [18] (see Appendix A.4.3). We set the features of this model to be the parameters controlling somatic ion channels densities. For each sample, we simulated an injection of 0.5 nano-Ampre of current to the soma, evoking a spiking activity. We chose the scalar output function to be the inter-spike interval coefficient of variance and looked for interesting patterns of activity using HINT.

The first two interactions ranked highest by HINT were already known (controlling long duration action potentials and depolarization block [4]). The third interaction, between *calcium dependent potassium* and *voltage dependent potassium*, however, led to an interesting observation: given the current injection, the neuron responded by a set of several bursts (Figure 6B1). This behavior, called **tonic bursting**, has been studied in simplified models specifically designed to create such behavior [21] as well as in experiments [15]. It is believed to be involved in oscillations in the cortex [15]. The ability of voltage-dependent potassium and calcium-dependent potassium to switch between non-bursting and bursting behavior was found experimentally [45, 49]; however, using HINT, we were able to find a region of ion channel densities where the voltage- and calcium-dependent potassium channels had a finer control over the behavior of the model than previously described.

Figure 6 demonstrates our finding. Increasing the density of calcium dependent potassium channels caused the cell to only fire a single burst of spikes (Figure 6B2). Increasing the density of voltage-dependent potassium caused the cell to fire a smaller number of spikes in its burst (Figure 6B3). Increasing both caused the cell to not fire at all (Figure 6B4). Further exploration of the input-output region automatically found by HINT showed that the voltage-dependent potassium channel controlled the inter-burst interval while the calcium-dependent potassium channel controlled

the number of spikes in each burst. Notably (and unlike Section 3.2), we were able to find this unique interaction between the features **without prior knowledge** and using a model which was not specifically designed for any tonic bursting related behavior [18].

4.2 Earth sciences

So far, we have demonstrated HINT on neuroscience models. However, it can be used to reveal local nonlinearities in any black-box simulator. We now test the tool on another domain that is rich in highly nonlinear, realistic, data-driven models: *Earth sciences*. We collaborated with a lab that studies an NPZD (Nutrient-Phytoplankton-Zooplankton-Detritus) model, commonly used as representation of the ecological system in the ocean [10]. This model contains four coupled equations describing the concentration evolution of these four variables in time, set in a physical model simulating ocean depth and forced by insolation [5].

To explore the input-output space of this model, we perturbed 15 features (see Appendix A.4.4) and ran the simulation. Our output scalar function was the timing of the phytoplankton population peak, signifying the phytoplankton bloom. The mechanism for this phenomenon is controversial [2, 44] and has been examined using both ecological models and observations.

Using HINT, we found that the ranking for *growth rate* and *mortality rate* was higher than the other pairs by a large margin (Score of 75 compared to 48 for the next-highest feature pair). Inspecting the results, we found that HINT was able to highlight a feature region that limited the behavior of the model: when the growth rate/mortality rate ratio was below a certain threshold, the phytoplankton population never rose above the starting value. This undesired behavior was new and intriguing to the researchers and its discovery can assist in constraining the feature space for this model. This further demonstrates the ability of HINT to find informative input regions in scientific models.

5 RELATED WORK

To the best of our knowledge, this work is the first attempt to automatically highlight local interesting phenomena in scientific models, where interestingness is defined as updating users' priors. However, our work builds on many other studies before us:

Our work is greatly influenced by attempts to mathematically define interestingness. By defining learning as the compression of models, Schmidhuber [38] defined interestingness as the derivative of model compression introduced by observing a new sample. Similarly, the subjective interestingness theory [25] defines interestingness as the update in users' beliefs after observing the data. In his work on SICA [25], De Bie proposes a rigorous mathematical framework for reducing the dimensionality of data such that it will organize datasets with respect to their update of users' priors.

Computer scientists have been trying for decades to automate the process of scientific discovery, starting from rule discovery by ABACUS [12] and BACON [6], with recent developments by [26, 39, 51] and others. HINT differs from those works mainly by searching for local phenomena, instead of global rules, and by assuming a black-box simulator instead of unstructured data.

The use of neuroscientific models as targets for scientific analysis was introduced and advanced by Eve Marder [34]. Marder revolutionized the use of neuroscientific models, exploring their input-output spaces and studying how their behavior changes with feature perturbation. Following her work, Uncertainpy [46] performs sensitivity analysis on neuroscientific models, showing how sensitive are the model parameters to individual perturbation.

Finally, other tools were built to detect nonlinear pairwise interactions: ANOVA fits a GAM to the data [17], then calculates the interaction strength of each feature by its p-value [50]. RuleFit [13, 14] fits sparse linear models using binary decision rules, and tests the interaction between features by their decomposition into two additive partial dependence plots. VIN [19] created a graph of interactions using additive decomposition of the black-box model. Similarly, GA2M [30] finds interacting pairs by calculating the error of a GAM, thus finding the relevant pairwise interaction terms to add to the model. Our method differs from these tools by exploring black-box simulators, while they explore unstructured data.

6 DISCUSSION AND CONCLUSIONS

In this work we presented HINT, a tool that highlights interesting and potentially novel phenomena by ranking local interactions between features in scientific models. We tested HINT on existing and new synthetic models, outperforming all state-of-the-art methods in a smaller running time. In addition, we reproduced the discovery of experimentally confirmed phenomena using scientific models that were created prior to the original theoretical and experimental discovery. Finally, we tested HINT on two currently studied models (in neuroscience and earth science) and discovered phenomena that are of interest to the scientists studying those fields.

Compared to existing tools available, HINT has several advantages: It is model-agnostic and can accept any model defined as $f(\bar{x}) : \mathbb{R}^m \to \mathbb{R}$. It requires a very small number of samples (in the order m^2 samples when m is the number of features). It is capable of accurately highlighting local nonlinear interactions, allowing researchers to easily identify bifurcations, borders between different states, and unknown local properties of models.

However, HINT has several limitations that should be addressed. First, as HINT aims to find interactions in scientific computational models, it is model-based rather than data-based. While this allows us to sample structured data and to numerically calculate the Hessian using forward differences, this also prevents us from using HINT on real-life unstructured data. Likewise, the forward difference approach requires accurate sampling of data from the underlying model. In case of noise, HINT's performance reduces, and more samples are necessary to correctly detect the interactions.

In addition, HINT requires the definition of a scalar output function. This means that any interesting behaviors that manifest in the raw data but are not captured in its transformation to scalar form may not be revealed in HINT (e.g., phase shift of a periodic function may not manifest when the output function is the integral). This restriction of the model output limits the abilities of HINT and should be improved in the future, e.g., by performing informed dimensionality reduction of the raw data to an interpretable feature.

Finally, HINT highlights input-output regions that contrast a single heuristic assumption, namely that all features do not interact with one another with respect to the output. While this was sufficient to find previously unknown behaviors in models explored in

this work, the actual priors modelers have while studying models might be different than our chosen heuristic. Similar to the work of De Bie [25], we would want users of Hint to be able to flexibly define priors that fit their beliefs, such that the highlighted samples will be as effective in updating said priors as possible. We believe that such automated tools to explore computational models can serve as a vehicle to accelerate scientific discoveries in many fields.

Acknowledgements

We thank the anonymous reviewers for their helpful comments. Dafna Shahaf is a Harry & Abe Sherman assistant professor. This work was supported by ISF grant 1764/15, the Drahi family foundation to IS, the Gatsby Charitable Foundation and the EPFL-Hebrew University Collaborative Grant and the EU Horizon 2020 program (720270, Human Brain Project).

REFERENCES

[1] Srdjan D. Antic, Wen Liang Zhou, Anna R. Moore, Shaina M. Short, and Katerina D. Ikonomu. 2010. The decade of the dendritic NMDA spike. *Journal of Neuroscience Research* (2010).
[2] Michael J Behrenfeld and Michael J Behrenfeld. 2017. Abandoning Sverdrup' s Critical Depth Hypothesis on phytoplankton blooms. *Ecology* (2017).
[3] Francisco Bezanilla. 2008. Ion Channels: From Conductance to Structure. *Neuron* (2008).
[4] Daniela Bianchi, Addolorata Marasco, Alessandro Limongiello, Cristina Marchetti, Helene Marie, Brunello Tirozzi, and Michele Migliore. 2012. On the mechanisms underlying the depolarization block in the spiking dynamics of CA1 pyramidal neurons. *Journal of Computational Neuroscience* (2012).
[5] E. Biton and H. Gildor. 2011. Stepwise seasonal restratification and the evolution of salinity minimum in the Gulf of Aqaba (Gulf of Eilat). *Journal of Geophysical Research* (2011).
[6] Gary F Bradshaw, P. W. Langley, and H. A. Simon. 1983. Studying Scientific Discovery by Computer Simulation. *Science* (1983).
[7] Nicholas T. Carnevale and Michael L. Hines. 2006. *The NEURON Book*. Cambridge University Press, Cambridge.
[8] Michael Doron, Giuseppe Chindemi, Eilif Muller, Henry Markram, and Idan Segev. 2017. Timed Synaptic Inhibition Shapes NMDA Spikes, Influencing Local Dendritic Processing and Global I/O Properties of Cortical Neurons. *Cell Reports* (2017).
[9] Kai Du, Yu-Wei Wu, Robert Lindroos, Yu Liu, Balázs Rózsa, Gergely Katona, Jun B. Ding, and Jeanette Hellgren Kotaleski. 2017. Cell-type-specific inhibition of the dendritic plateau potential in striatal spiny projection neurons. (2017).
[10] A Edwards. 1999. Zooplankton Mortality and the Dynamical Behaviour of Plankton Population Models. *Bulletin of Mathematical Biology* (1999).
[11] J. Evans and A. Rzhetsky. 2010. Machine Science. *Science* (2010).
[12] Brian C. Falkenhainer and Ryszard S Michalski. 1986. Integrating quantitative and qualitative discovery: The ABACUS system. *Machine Learning* (1986).
[13] Jerome H. Friedman. 2001. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics* (2001).
[14] Jerome H. Friedman and Bogdan E. Popescu. 2008. Predictive learning via rule ensembles. *The Annals of Applied Statistics* (2008).
[15] C. M. Gray and D. A. McCormick. 1996. Chattering Cells: Superficial Pyramidal Neurons Contributing to the Generation of Synchronous Oscillations in the Visual Cortex. *Science* (1996).
[16] Brandon Greenwell, Bradley Boehmke, Jay Cunningham, and GBM Developers. 2019. *gbm: Generalized Boosted Regression Models*. R package version 2.1.5.
[17] T.J. Hastie and R.J. Tibshirani. 1990. *Generalized additive models*. Chapman & Hall/CRC.
[18] Etay Hay, Sean Hill, Felix Schürmann, Henry Markram, and Idan Segev. 2011. Models of neocortical layer 5b pyramidal cells capturing a wide range of dendritic and perisomatic active properties. *PLoS Computational Biology* (2011).
[19] Giles Hooker. 2004. Discovering additive structure in black box functions. In *Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '04*. ACM Press, New York, New York, USA.
[20] Giles Hooker. 2007. Generalized Functional ANOVA Diagnostics for High-Dimensional Functions of Dependent Variables. *Journal of Computational and Graphical Statistics* (2007).
[21] Eugene M. Izhikevich. 2000. neural Excitability, Spiking And Bursting. *International Journal of Bifurcation and Chaos* (2000).
[22] Monika Jadi, Alon Polsky, Jackie Schiller, and Bartlett W. Mel. 2012. Location-Dependent Effects of Inhibition on Local Spiking in Pyramidal Neuron Dendrites. *PLoS Computational Biology* (2012).
[23] C E Jahr and C F Stevens. 1990. Voltage dependence of NMDA-activated macroscopic conductances predicted by single-channel kinetics. *The Journal of neuroscience* (1990).
[24] Scott L. Jones, Minh-Son To, and Greg J. Stuart. 2017. Dendritic small conductance calcium-activated potassium channels activated by action potentials suppress EPSPs and gate spike-timing dependent synaptic plasticity. *eLife* (2017).
[25] Bo Kang, Jefrey Lijffijt, Raúl Santos-Rodríguez, and Tijl De Bie. 2018. SICA: subjectively interesting component analysis. *Data Mining and Knowledge Discovery* (2018).
[26] Ross D. King, Jem Rowland, Stephen G. Oliver, Michael Young, Wayne Aubrey, Emma Byrne, Maria Liakata, Magdalena Markham, Pinar Pir, Larisa N. Soldatova, Andrew Sparkes, Kenneth E. Whelan, and Amanda Clare. 2009. The Automation of Science. *Science* (2009).
[27] Angela M. Kuhn, Katja Fennel, and Jann Paul Mattern. 2015. Model investigations of the North Atlantic spring bloom initiation. *Progress in Oceanography* (2015).
[28] Alan U. Larkman. 1991. Dendritic morphology of pyramidal neurones of the visual cortex of the rat: III. Spine distributions. *The Journal of Comparative Neurology* (1991).
[29] Yin Lou. 2018. *mltk: Machine Learning Tool Kit*. https://github.com/yinlou/mltk
[30] Yin Lou, Rich Caruana, Johannes Gehrke, and Giles Hooker. 2013. Accurate intelligible models with pairwise interactions. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '13*. ACM Press, New York, New York, USA.
[31] Guy Major, Matthew E. Larkum, and Jackie Schiller. 2013. Active Properties of Neocortical Pyramidal Neuron Dendrites. *Annual Review of Neuroscience* (2013).
[32] Christoph Molnar. 2019. *Interpretable Machine Learning*. https://christophm.github.io/interpretable-ml-book/.
[33] William F. Podlaski, Alexander Seeholzer, Lukas N. Groschner, Gero Miesenböck, Rajnish Ranjan, and Tim P. Vogels. 2017. Mapping the function of neuronal ion channels in model and experiment. *eLife* (2017).
[34] Astrid A. Prinz, Cyrus P. Billimoria, and Eve Marder. 2003. Alternative to Hand-Tuning Conductance-Based Models: Construction and Analysis of Databases of Model Neurons. *Journal of Neurophysiology* (2003).
[35] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. (2016).
[36] Markus Schartau and Andreas Oschlies. 2003. Simultaneous data-based optimization of a 1D-ecosystem model at three locations in the North Atlantic: Part II-Standing stocks and nitrogen fluxes. *Journal of Marine Research* (2003).
[37] Jackie Schiller, Guy Major, Helmut J Koester, and Yitzhak Schiller. 2000. NMDA spikes in basal dendrites of cortical pyramidal neurons. *Nature* (2000).
[38] Jürgen Schmidhuber. 2010. Formal Theory of Creativity, Fun, and Intrinsic Motivation (1990-2010). *IEEE Transactions on Autonomous Mental Development* (2010).
[39] Michael Schmidt and Hod Lipson. 2009. Distilling Free-Form Natural Laws from Experimental Data. *Science* (2009).
[40] I Segev and W Rall. 1988. Computational study of an excitable dendritic spine. *Journal of Neurophysiology* (1988).
[41] Greg J. Stuart and Michael Häusser. 2001. Dendritic coincidence detection of EPSPs and action potentials. *Nature Neuroscience* (2001).
[42] Greg J. Stuart and Bert Sakmann. 1994. Active propagation of somatic action potentials into neocortical pyramidal cell dendrites. *Nature* (1994).
[43] Yuriy Sverchkov, Mark Craven, FM Jones, PGK Reiser, CH Bryant, and SH Mugggleton. 2017. A review of active learning approaches to experimental design for uncovering biological networks. *PLOS Computational Biology* (2017).
[44] H. U. Sverdrup. 1953. On Conditions for the Vernal Blooming of Phytoplankton. *ICES Journal of Marine Science* (1953).
[45] J. Tabak, M. Tomaiuolo, A. E. Gonzalez-Iglesias, L. S. Milescu, and R. Bertram. 2011. Fast-Activating Voltage- and Calcium-Dependent Potassium (BK) Conductance Promotes Bursting in Pituitary Cells: A Dynamic Clamp Study. *Journal of Neuroscience* (2011).
[46] Simen Tennøe, Geir Halnes, and Gaute T. Einevoll. 2018. Uncertainpy: A Python Toolbox for Uncertainty Quantification and Sensitivity Analysis in Computational Neuroscience. *Frontiers in Neuroinformatics* (2018).
[47] Werner Van Geit, Michael Gevaert, Giuseppe Chindemi, Christian Rössert, Jean-Denis Courcol, Eilif B. Muller, Felix Schürmann, Idan Segev, and Henry Markram. 2016. BluePyOpt: Leveraging Open Source Software and Cloud Infrastructure to Optimise Model Parameters in Neuroscience. *Frontiers in Neuroinformatics* (2016).
[48] Jack Waters, Andreas Schaefer, and Bert Sakmann. 2005. Backpropagating action potentials in neurones: Measurement, mechanisms and potential functions. *Progress in Biophysics and Molecular Biology* (2005).
[49] Mary D Womack and Kamran Khodakhah. 2003. Somatic and Dendritic Small-Conductance Calcium-Activated Potassium Channels Regulate the Output of Cerebellar Purkinje Neurons. *The Journal of Neuroscience* (2003).
[50] Simon N Wood. 2017. *Generalized Additive Models: An Introduction with R*. Chapman and Hall/CRC.
[51] Tailin Wu and Max Tegmark. 2018. Toward an AI Physicist for Unsupervised Learning. (2018). http://arxiv.org/abs/1810.10525

A REPRODUCIBILITY

A.1 Code and data availability

The implementation of HINT in Python can be accessed via this link (https://github.com/MichaelDoron/HINT). The same git repository holds the code running the benchmarks Anova, GA2M and PDP, as well as the synthetic and computational models in Sections 3 and 4 and compressed files containing the data simulated from the computational models.

A.2 Synthetic black-box models

A.2.1 Complex function. Following the work of [19], we defined the first synthetic model as the function

$$F(x) = \pi^{x_1 x_2}\sqrt{2x_3} - sin^{-1}(x_4) + log(x_3 + x_5) - \frac{x_9}{x_{10}} - x_2 x_7$$

Features x_4, x_5, x_8, x_{10} were uniformly distributed between the limits $[0.6, 1.0]$ and the rest were uniformly distributed between $[0.0, 1.0]$. The ground truth for this function were the pairs $\{ (x_1, x_2), (x_1, x_3), (x_2, x_3), (x_2, x_7), (x_3, x_5), (x_7, x_8), (x_7, x_9), (x_7, x_{10}), (x_8, x_9), (x_8, x_{10}), (x_9, x_{10}) \}$

A.2.2 Gaussian (10). Following the work in [13], we generated a random model that sums 25 multivariate Gaussian functions, each receiving a subset of 10 variables, thus receiving 45 possible interacting pairs. Ground truth was all pairs within the 25 subsets. Detailed explanation is found in [13].

A.2.3 Gaussian (100). Following the work in [13], we generated a random model that sums 1000 multivariate Gaussian functions, each receiving a subset of 100 variables, thus receiving 4590 possible interacting pairs. Ground truth was all pairs within the 1000 subsets. Detailed explanation is found in [13].

A.2.4 Noisy Gaussian (10). To generate the Noisy Gaussian, we sampled 1000 samples from the randomly generated Gaussian (10) model, and calculated their standard deviation σ_{signal}. Then, ran the Gaussian (10) model on new samples, adding Gaussian noise $\mathcal{N}(0, \frac{\sigma^2_{signal}}{20})$ to the result.

A.2.5 Local interactions. Given several local phenomena N_p, we construct a function

$$F(x) = \sum_{p=1}^{N_p} L(z_p) + \sum_{i=1}^{m} k_i \cdot {x_i}^{r_i}$$

with k_i being a coefficient for the non-interacting effect of x_i and r_i being the power to which the x_i is raised. k_i is an integer randomly sampled from $U(-5, 5)$, and r_i is an integer randomly sampled from $U(1, 3)$. z_p is defined to be a function of x, returning a permutation of a subset of the input dimensions of x

$$z_p = \{x_{s_p(j)}\}_{j=1}^{n_p}$$

Each perturbation set s_p is a random perturbation of the integer set {1,..., m} with a random length $n_p = \lfloor 2 + r \rfloor$, r being drawn from an exponential distribution with mean 1. Finally, let $L(z_p)$ be an n_p-dimensional local Gaussian function

$$L(z_p) = e^{\left(-\sum_{i=1}^{n_p}\left(g_i \cdot \left(z_{p\,i} - c_i\right)\right)^2\right)}$$

with $\bar{g}$ and $\bar{c}$ being vectors of magnitudes and centers (respectively) of length n_p.

For ground truth, we took all points for which the function value was not negligible (the function is between 0 and e, and we set the threshold of contribution at $\frac{e}{1000}$, which we receive when the values are $e^{\left(-\sum_{i=1}^{n_p}\left(g_i \cdot \left(z_{p\,i} - c_i\right)\right)^2\right)} = e^{log(\frac{e}{1000})} = e^{-5.9}$). Thus,

$$y(z_p) = \begin{cases} 1, & \text{if } \sum_{i=1}^{n_p}\left(g_i \cdot \left(z_{p\,i} - c_i\right)\right)^2 < 5.9 \\ 0, & \text{otherwise} \end{cases}$$

A.3 Benchmarks

A.3.1 GA2M. We used the MLTK Java package [29] to run GA2M on the data. We discretized the data using 256 bins, trained a boosted tree ensemble with 1000 trees, and ran FAST on the residuals using 8 bins.

A.3.2 ANOVA. We used the mgcv R package [50] to train a GAM on the data, and calculated the p-values of all possible feature pairs.

A.3.3 PDP. We used the gbm R package [16] to train a gradient boosting machine with 1000 trees and to calculate the H-statistic for each feature pair.

A.4 Computational models

Single neuron models were modelled as a single compartment RC circuit

$$C_m \cdot \frac{dV}{dt} = \sum_i g_i \cdot (V(t) - E_i)$$

Where the membrane capacitance is represented by C_m, the ion reversal potential E_i is modelled as batteries, and the conductance g_i represents the membrane resistivity. The g_i functions can be dependent on voltage, time, and other ion currents, often making this system highly nonlinear.

In the models below, we set the reversal potentials E_i as constants, and changed parameters within the g_i functions as features of the model. The raw output of the model was membrane voltage recorded from the center of the single compartment representing the dendrite, and the scalar output was a function of that voltage trace.

A.4.1 Dendritic model - NMDA. This model, used in Section 3.2.2 reproduces the membrane voltage in the dendrite of a pyramidal cell [8]. The dendrite was modelled as an isopotential passive cell 20 μm in diameter, with an $R_m = 20,000\ \Omega cm^2$, $C_m = 1\ \mu F/cm^2$ and a resting potential of $-70mV$ [8]. It consists of a leak channel and two synapses - an excitatory AMPA/NMDA synapse and an inhibitory $GABA_A$ synapse. Reversal potentials were $E_{leak} = -70\ mV$, $E_{AMPA} = 0\ mV$, $E_{NMDA} = 0\ mV$ and $E_{GABA_A} = -80\ mV$. $AMPA$ and $GABA_A$ synapses were modelled as

$$g_{syn}(t) = {g_{syn}}_{Max} \cdot \left(e^{\frac{t_{0\,syn} - t}{\tau_d}} - e^{\frac{t_{0\,syn} - t}{\tau_r}}\right) \quad (1)$$

with the *AMPA* and $GABA_A$ rise / decay times being 0.18 / 5 ms and 0.2 / 1.7 ms, respectively. t_{0syn} represents the time of synaptic activation. The NMDA synapse was modelled [23] as

$$g_{NMDA} = g_{NMDAMax} \cdot \left(\frac{e^{\frac{t_{0NMDA}-t}{\tau_d}} - e^{\frac{t_{0NMDA}-t}{\tau_r}}}{1 + e^{-0.08 \cdot V(t)} \cdot \frac{1}{3.57}} \right)$$

with the rise / decay times being 2 / 75 ms. In the model used in Section 3.2.2, we set the features to be the maximum synaptic conductances $g_{NMDAMax}$ and g_{GABA_AMax}, and the time difference between the activation of the two synapses $Delay = t_{0GABA_A} - t_{0NMDA}$. In all models in this work, the AMPA and NMDA maximum conductances were equal and represented by the same feature $g_{NMDAMax}$, as well as the *AMPA* and *NMDA* activation times.

The scalar output used was the time integral of the dendritic membrane voltage $\int_{t=0}^{300} V(t) + |\mathrm{v_{init}}|\, dt$, where $\mathrm{v_{init}}$ is the resting potential (v_{init} = -70), and the model was simulated for 300 milliseconds (from $t = -100ms$ to $t = 200ms$)

A.4.2 Dendritic model - Backpropagating action potential. This model which was used in Section 3.2.3 simulates the dendrite of a layer 5 pyramidal cell [18], with additional ion channels and a dynamic clamp injection of a Backpropagating action potential (BAP). The ion channels used in this model are the leak channel, high- and low- voltage dependent calcium channel, fast- and persistent-type sodium channels, M-type Potassium channel, and calcium- and voltage- dependent potassium channels, as described in the dendritic part of the L5PC model of [18]. The features of the model were 9 features controlling these ion channels and the leak current, as well as three other: synaptic rates and BAP height.

The activation times of the excitatory and inhibitory synapses were sampled from a Poisson distribution, with the rates being features of the model. In addition, this model had was injected with current simulating a BAP arriving to the dendrite from the soma. The shape of the BAP was recorded from the soma of the full model of a pyramidal cell [18], and the attenuation of its height was set to be a feature representing the distance of the dendrite from the soma.

The feature limits of the ion channel features were set to be between 90% and 110% of the values specified in [18]. The limits of the excitatory and inhibitory rates were (300, 2400) ms, and the limits of the BAP height were (0%, 100%) of the height of the action potential measured at the soma.

As before, the scalar output used was the time integral of the dendritic membrane voltage, and the model was simulated for 100 milliseconds.

A.4.3 Somatic model. In the model used in Section 4.1 we simulated the soma of a layer 5 pyramidal cell under current injection. The model had no synaptic input, and instead was injected with 0.5 *nA* for 1400 milliseconds.

The ion channels in this model were: Leak channel, high- and low- voltage dependent calcium channel, fast- and persistent- type sodium channels, calcium- and voltage- dependent potassium channels, and persistent- and transient- potassium channels, as described in the somatic part of the L5PC model of [18]. The features of the model were 11 features controlling these ion channels, and their limits were between 0 and twice the values specified in [18].

The raw output of the model was membrane voltage recorded from the center of the single compartment representing the soma, and the scalar output was the interspike interval coefficient of variance, calculated using the python library BluePyOpt [47]. The model was simulated for 2000 milliseconds.

A.4.4 NPDZ model. The NPDZ (Nutrient - Phytoplankton - Zooplankton - Detritus) model is commonly used as a simple representation of the ecological system in the ocean. The model we used is based on the equations by [10], set in a 1D model (single column) of Eilat bay and forced by insolation (measurements by the national monitoring program) and mixing features (from a 3D MITgcm specified for the Gulf of Eilat [5]).

The 15 features used for this model were: Growth rate, PON remineralization rate, Nitrate half saturation coefficient, Zooplankton grazing rate, Grazing half saturation, Mortality rate, Sinking rate of phytoplankton, Zooplankton mortality rate, Grazing efficiency, PON sinking rate, light saturation coefficient, light inhibition coefficient, and two features controlling the chlorophyll to carbon ratio. We set the feature limits to be in the range of values found in literature (e.g., [27, 36]).

The raw output of the model was the size of the phytoplankton population, and the scalar output was the timing of the peak of the population.