Functional Transparency for Structured Data: a Game-Theoretic Approach

Guang-He Lee [1] Wengong Jin [1] David Alvarez-Melis [1] Tommi S. Jaakkola [1]

Abstract

We provide a new approach to training neural models to exhibit transparency in a well-defined, functional manner. Our approach naturally operates over structured data and tailors the predictor, functionally, towards a chosen family of (local) witnesses. The estimation problem is setup as a co-operative game between an unrestricted *predictor* such as a neural network, and a set of *witnesses* chosen from the desired transparent family. The goal of the witnesses is to highlight, locally, how well the predictor conforms to the chosen family of functions, while the predictor is trained to minimize the highlighted discrepancy. We emphasize that the predictor remains globally powerful as it is only encouraged to agree locally with locally adapted witnesses. We analyze the effect of the proposed approach, provide example formulations in the context of deep graph and sequence models, and empirically illustrate the idea in chemical property prediction, temporal modeling, and molecule representation learning.

1. Introduction

Modern machine learning tasks are increasingly complex, requiring flexible models with large numbers of parameters such as deep networks (Silver et al., 2016; Vaswani et al., 2017; Huang et al., 2017). Such modeling gains often come at the cost of transparency or interpretability. This is particularly problematic when predictions are fed into decision-critical applications such as medicine where the ability to verify predictions may be just as important as the raw predictive power.

It seems plausible to guide a flexible neural network towards a complex yet well-understood (i.e., transparent) functional class. For example, in realizing Wasserstein-1 distance (Arjovsky et al., 2017), the discriminator should be limited to 1-Lipschitz functions. A strict adherence to a complex, global functional class is not the only way to achieve transparency. For example, linearity is a desirable characteristic for transparency but is sensible to enforce only locally. We offer therefore a new notion of transparency – functional transparency – where the goal is to guide models to adopt a desirable local behavior yet allowing them to be more flexible globally. Note that functional transparency should be established only approximately in many cases since, e.g., strict local linearity implies global linearity.

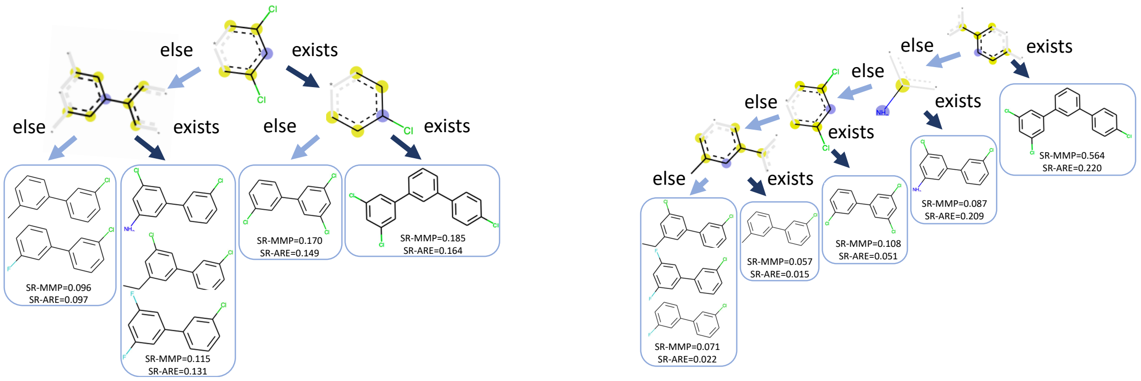
Previous approaches to interpretability have mainly focused on models that operate on fixed-size data, such as scalar-features (Lakkaraju et al., 2016) or images (Selvaraju et al., 2016; Mahendran & Vedaldi, 2015). The emphasis has been on feature relevance or selection (Ribeiro et al., 2016). Recent methods do address some of the challenges in sequential data (Lei et al., 2016; Arras et al., 2017; Alvarez-Melis & Jaakkola, 2017), primarily in NLP tasks where the input sequence is discrete. Interpretability for continuous temporal data (Al-Shedivat et al., 2017; Wu et al., 2018a) or graph structures remains largely unexplored.

We develop a novel approach to transparency that is naturally suited for structured data. At the core of our approach is a game-theoretic definition of transparency. This is set up as a two-player co-operative game between a *predictor* and a *witness*. The predictor remains a complex model whereas the witness is chosen from a simple transparent family. Transparency arises from the fact that the predictor is encouraged to exemplify simple behavior as captured by the witness in each local region while remaining globally powerful. The approach differs from global regularization of models towards interpretability (Wu et al., 2018a), models that are constructed a priori to be interpretable, either architecturally or in terms of the function class (Al-Shedivat et al., 2017; Lei et al., 2016), or from post-hoc explanations of black-box methods via local perturbations (Ribeiro et al., 2016; Alvarez-Melis & Jaakkola, 2017). Our models are guided towards functional transparency during learning.

As an illustration, we contrast our approach with methods that seek to obtain interpretable explanations after the fact (e.g., (Ribeiro et al., 2016)). Derived explanation after training can be misleading in some cases if the explanation does not match the functional behavior of the model. For exam-

[1] MIT Computer Science and Artificial Intelligence Laboratory. Correspondence to: Guang-He Lee <guanghe@csail.mit.edu>.

Proceedings of the 36^{th} *International Conference on Machine Learning*, Long Beach, California, PMLR 97, 2019. Copyright 2019 by the author(s).



(a) The explanation from our model (trained for transparency).

(b) The explanation from a normally trained model.

Figure 1. During testing, we fit decision trees to our model and an unregularized model on molecule property prediction at the same local neighborhood such that the functional approximations are comparable in AUC (because the scale is not crucial). The split criterion on each node is based on the existence of a *complete chemical substructure* in Morgan fingerprints (Rogers & Hahn, 2010). The color of each Morgan fingerprint simply reflects the radius of the fingerprint.

ple, Figure 1 shows local decision tree approximations for two models: our model trained with such local witnesses (a, left), and an unregularized model (b, right). The trees are constructed to achieve the same level of approximation. The tree for the unregularized model only filters one sample in each split, lacking generality to explain the (local) behavior. This phenomenon is related to unstable explanations that arise with already trained models (Alvarez-Melis & Jaakkola, 2018a; Ghorbani et al., 2019).

The game theoretic approach is very flexible in terms of models and scenarios. We therefore illustrate the approach across a few novel scenarios: explaining graph convolutional models using decision trees, revealing local functional variation of a deep sequence model, and exemplifying decision rules for the encoder in unsupervised graph representation learning. Our main contributions are:

- A novel game-theoretic approach to transparency, applicable to a wide range of prediction models, architectures, and local transparency classes, without requiring differentiability.
- Analysis on the effective size of the local regions and establishing equilibria pertaining to different game formulations.
- Illustration of deep models across several tasks, from chemical property prediction, physical component modeling, to molecule representation learning.

2. Related Work

The role of transparency is to expose the inner-workings of an algorithm (Citron & Pasquale, 2014; Pasquale, 2015), such as decision making systems. This is timely for state-of-the-art machine learning models that are typically overparameterized (Silver et al., 2016; He et al., 2016) and therefore effectively black-box models. An uncontrolled model is also liable to various attacks (Goodfellow et al., 2015).

Our goal is to regularize a complex deep model so that it exhibits a desired local behavior. The approach confers an approximate operational guarantee rather than directly interpretability. In contrast, examples of archetypal interpretable models include linear classifiers, decision trees (Quinlan, 2014), and decision sets (Lakkaraju et al., 2016); recent approaches also guide complex models towards highlighting pieces of input used for prediction (Lei et al., 2016), learning representations that can be decomposed among training examples (Yeh et al., 2018), or generalizing linear models while maintaining interpretability (Alvarez-Melis & Jaakkola, 2018b). A model conforming to a known functional behavior, at least locally, as in our approach, is not necessarily itself human-interpretable. The approximate guarantee we offer is that the complex model indeed follows such a behavior and we also quantify to what extent this guarantee is achieved.

Previous work on approximating a functional class via neural networks can be roughly divided into two types: parametrization-based and regularization-based methods. Works in the first category seek self-evident adherence to a functional class, which include maintaining Lipschitz continuity via weight clipping (Arjovsky et al., 2017), orthogonal transformation via scaled Cayley transform of skew-symmetric matrices (Helfrich et al., 2018), and "stable" recurrent networks via spectral norm projection on the transition matrix (Miller & Hardt, 2018).

A softer approach is to introduce a regularization problem that encourages neural networks to match properties of the functional class. Such regularization problem might come in the form of a gradient penalty as used in several variants of GAN (Gulrajani et al., 2017; Bellemare et al., 2017; Mroueh et al., 2018) under the framework of integral probability metrics (Müller, 1997), adversarial approaches to learn representations for certain independence statements (Ganin et al., 2016; Zhao et al., 2017), and expanding locally linear regions of piecewise linear networks for stable gradients (Lee et al., 2019). Typically, a tailored regularization problem is introduced for each functional class. Our work follows this general theme in the sense of casting the overall problem as a regularization problem. However, we focus on transparency and our approach – a general co-operative game – is quite different. Our methodology is applicable to any choice of (local) functional class without any architectural restrictions on the deep model whose behavior is sculpted. The optimization of functional deviation in the game must remain tractable, of course.

3. Methodology

In this work, given a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N \subset \mathcal{X} \times \mathcal{Y}$, we learn an (unrestricted) predictive function $f \in \mathcal{F} : \mathcal{X} \to \mathcal{Y}$ together with a transparent – and usually simpler – function $g \in \mathcal{G} : \mathcal{X} \to \mathcal{Y}$ defined over a functional class $\mathcal{G}$. We refer to functions f and g as the *predictor* and the *witness*, respectively, throughout the paper. Note that we need not make any assumptions on the functional class $\mathcal{F}$, instead allowing a flexible class of predictors. In contrast, the family of witnesses $\mathcal{G}$ is strictly constrained to be a *transparent* functional set, such as the set of linear functions or decision trees. We assume to have a deviation function $d : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$ such that $d(y, y') = 0 \iff y = y'$, which measures discrepancy between two elements in $\mathcal{Y}$ and can be used to optimize f and g. To simplify the notation, we define $\mathcal{D}_x := \{x_i : (x_i, y_i) \in \mathcal{D}\}$. We introduce our game-theoretic framework in §3.1, analyze it in §4, and instantiate the framework with concrete models in §5.

3.1. Game-Theoretic Transparency

There are many ways to use a witness function $g \in \mathcal{G}$ to guide the predictor f by means of discrepancy measures. However, since the witness functions can be weak such as linear functions, we cannot expect that a reasonable predictor would agree to it globally. Instead, we make a slight generalization to enforce this criterion only locally, over different sets of neighborhoods. To this end, we define *local* transparency by measuring how close f is to the family $\mathcal{G}$ over a local neighborhood $\mathcal{B}(x_i) \subset \mathcal{X}$ around an observed point x_i. One straightforward instantiation of such a neighborhood $\mathcal{B}_\epsilon(x_i)$ in temporal domain will be simply a local window of points $\{x_{i-\epsilon}, \ldots, x_{i+\epsilon}\}$. Our resulting local discrepancy measure is

$$\min_{g \in \mathcal{G}} \frac{1}{|\mathcal{B}(x_i)|} \sum_{x_j \in \mathcal{B}(x_i)} d(f(x_j), g(x_j)). \tag{1}$$

The summation can be replaced by an integral when a continuous neighborhood is used. The minimizing witness function, $\hat{g}_{x_i}$, is indexed by the point x_i around which it is estimated; depending on the function f, the minimizing witness can change from one neighborhood to another. If we view the minimization problem game-theoretically, $\hat{g}_{x_i}$ is the *best response strategy* of the local witness around x_i.

The local discrepancy measure can be incorporated into an overall estimation criterion in many ways so as to guide the predictor towards the desired functional form. This guidance can be offered as a *uniform* constraint with a permissible δ-margin, as an additive *symmetric* penalty, or defined *asymmetrically* as a game theoretic penalty where the information sets for the predictor and the witness are no longer identical. We consider each of these in turn.

Uniform criterion. A straightforward formulation is to confine f to remain within a margin δ of the best fitting witness for every local neighborhood. Assume that a primal loss $\mathcal{L}(\cdot, \cdot)$ is given for a learning task. The criterion imposes the δ-margin constraint uniformly as

$$\sum_{(x_i, y_i) \in \mathcal{D}} \mathcal{L}(f(x_i), y_i) \tag{2}$$

$$s.t. \min_{g \in \mathcal{G}} \frac{1}{|\mathcal{B}(x_i)|} \sum_{x_j \in \mathcal{B}(x_i)} d(f(x_j), g(x_j)) \leq \delta, \forall x_i \in \mathcal{D}_x.$$

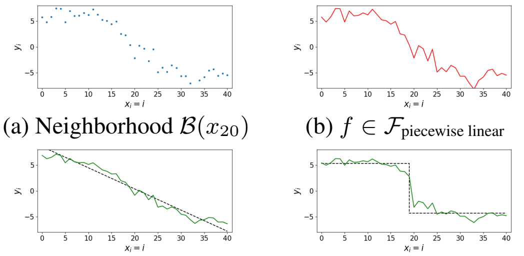
We assume that the optimal g with respect to each constraint may be efficiently found due to the simplicity of $\mathcal{G}$ and the regularity of $d(\cdot, \cdot)$. We also assume that the partial derivatives with respect to f, for fixed witnesses, can be computed straightforwardly under sufficiently regular $\mathcal{L}(\cdot, \cdot)$ in a Lagrangian form. In this case, we can solve for f, local witnesses, and the Lagrange multipliers using the mirror-prox algorithm (Nemirovski, 2004).

The hard constraints in the uniform criterion will lead to strict transparency guarantees. However, the effect may be undesirable in some cases where the observed data (hence the predictor) do not agree with the witness in all places. The resulting loss of performance may be too severe. As an alternative, we can enforce the agreement with local witnesses to be small in aggregate across neighborhoods.

Symmetric game. We define an additive, unconstrained, symmetric criterion to smoothly trade off between performance and transparency. The resulting objective is

$$\sum_{(x_i, y_i) \in \mathcal{D}} \Bigg[\mathcal{L}(f(x_i), y_i) + \min_{g \in \mathcal{G}} \frac{\lambda}{|\mathcal{B}(x_i)|} \sum_{x_j \in \mathcal{B}(x_i)} d(f(x_j), g(x_j)) \Bigg] \tag{3}$$

To illustrate the above idea, we generate a synthetic dataset to show a *neighborhood* in Figure 2a with an unconstrained piecewise linear predictor $f \in \mathcal{F}_{\text{piecewise linear}}$ in Figure 2b.



(a) Neighborhood $\mathcal{B}(x_{20})$ (b) $f \in \mathcal{F}_{\text{piecewise linear}}$

(c) $g_{x_{i=1}} \in \mathcal{G}_{\text{linear}}$ (d) $g_{x_{i=1}} \in \mathcal{G}_{\text{decision stump}}$

Figure 2. Examples of fitting a neighborhood $\mathcal{B}(x_{20})$ (2a) with a piecewise linear predictor (2b). Using different witness families (Figs. 2c&2d, dashed lines) leads to predictors (solid green) with different behaviors, despite yielding the same error (MSE=1.026).

Clearly, f does not agree with a linear witness within this neighborhood. However, when we solve for f together with a linear witness $g_{x_i} \in \mathcal{G}_{\text{linear}}$ as in Figure 2c, the resulting function has a small residual deviation from $\mathcal{G}_{\text{linear}}$, more strongly adhering to the linear functional class while still closely tracking the observed data. Figure 2d shows the flexibility of our framework where a very different functional behavior can be induced by changing the functional class for the witness.

Asymmetric game. Solving the symmetric criterion can be computationally inefficient since the predictor is guided by its deviation from each of the local witness on all points within each of the local neighborhoods. Moreover, the predictor value at any point x_i is subject to potentially conflicting regularization terms across the neighborhoods, which is undesirable. The inner summation in Eq. (3) may involve different sizes of neighborhoods $\mathcal{B}(x_i)$ (e.g., endpoint boundary cases) and this makes it more challenging to parallelize the computation.

We would like to impose even functional regularization at every $f(x_i)$ based on how much the value deviates from the witness associated with the local region $\mathcal{B}(x_i)$. This approach leads to an *asymmetric* co-operative formulation, where the information sets for the predictor f and local witnesses g_{x_i} differ. Specifically, the local best-response witness $\hat{g}_{x_i}$ is chosen to minimize the local discrepancy as in Eq. (1), and thus depends on f values within the whole region; in contrast, the predictor f only receives feedback in terms of the resulting deviation at x_i, only seeing $\hat{g}_{x_i}(x_i)$. From the point of view of the predictor f, the best response strategy is obtained by minimizing

$$\sum_{(x_i,y_i)\in\mathcal{D}} \Big[\mathcal{L}(f(x_i), y_i) + \lambda\, d(f(x_i), \hat{g}_{x_i}(x_i))\Big] \tag{4}$$

To train the proposed method, we perform alternating updates for $f(\cdot)$ and $\hat{g}_{x_i}(\cdot)$ on their respective criteria. Note that in this case the objective cannot be written as a single minimization problem (different information sets) but can be still interpreted as a game. The deviation between the predictor and the witness, $d(\cdot,\cdot)$, can also be defined asymmetrically as we do in §5.3.

4. Analysis

We consider here the effectiveness of regularization in relation to the neighborhood size and establish fixed point equations for the predictor under the three estimation criteria. For simplicity, we assume $\mathcal{X} = \mathbb{R}^D$ and $\mathcal{Y} = \mathbb{R}$, but the results are generalizable to our examples in §5. All the proofs are in Appendix A.

Neighborhood size. The formulation involves a key tradeoff between the size of the region where the function should be simple and the overall accuracy achieved by the predictor. When the neighborhood is too small, local witnesses become perfect, inducing no regularization on f. Thus the size of the region is a key parameter. A neighborhood size is sufficient if the witness class $\mathcal{G}$ cannot readily overfit f values within the neighborhood. Formally,

Definition 1. *We say that a neighborhood size m is* ***effective*** *for $\mathcal{G}$ if for any $f \notin \mathcal{G}$ we can find $\mathcal{B} \subset \mathcal{X} : |\mathcal{B}| = m$ s.t.*

$$\min_{g\in\mathcal{G}} \frac{1}{m} \sum_{x\in\mathcal{B}} d(f(x), g(x)) > 0. \tag{5}$$

A trivial example is when $\mathcal{G}$ is the constant class, a neighborhood size m is effective if $m > 1$. Note that the neighborhood $\mathcal{B}$ in the above definition can be any finite collection of points $\mathcal{B}(\cdot)$. For example, the points in the neighborhood induced by a temporal window $\{x_{i-\epsilon}, \ldots, x_{i+\epsilon}\}$ need not remain in a small ℓ_p-norm ball.

For linear models and decision trees, we have

- $D + 1$ is the tight lower bound on the effective neighborhood size for the linear class.
- $2^k + 1$ is the tight lower bound on the effective neighborhood size for decision trees with depth bounded by k.

When the sample sizes within a neighborhoods fall below the bounds, regularization can still be useful if the witness class is not uniformly flexible or if the algorithm for finding the witness is limited (e.g., greedy algorithm for decision trees).

Equilibrium solutions. The symmetric game constitutes a standard minimization problem, but the existence or uniqueness of equilibria under the asymmetric game are not obvious. Our main results in this section make the following assumptions.

(A1) the predictor f is unconstrained.

(A2) both the loss and deviation are squared errors.

(A3) $|\mathcal{B}(x_i)| = m, \forall x_i \in \mathcal{D}_x$.

(A4) $x_j \in \mathcal{B}(x_i) \implies x_i \in \mathcal{B}(x_j), \forall x_i, x_j \in \mathcal{D}_x$.

(A5) $\cup_{x_i \in \mathcal{D}_x} \mathcal{B}(x_i) = \mathcal{D}_x$.

We note that **(A3)** and **(A4)** are not technically necessary but simplify the presentation. We denote the predictor in the uniform criterion (Eq. (2)), the symmetric game (Eq. (3)), and the asymmetric game (Eq. (4)) as f_U, f_S, and f_A, respectively. We use $X_i \in \mathbb{R}^{m \times D}$ to denote the neighborhood $\mathcal{B}(x_i) = \{x'_1, \dots, x'_m\}$ ($X_i = [x'_1, \dots, x'_m]^\top$), and $f(X_i) \in \mathbb{R}^m$ to denote the vector $[f(x'_1), \dots, f(x'_m)]^\top$. $X_j^\dagger$ denotes the pseudo-inverse of X_j. Then we have

Theorem 2. *If* **(A1-5)** *hold and the witness is in the linear family, the optimal f_S satisfies*

$$f_S^*(x_i) = \frac{1}{1+\lambda}\left[y_i + \frac{\lambda}{m}\Big(\sum_{x_j \in \mathcal{B}(x_i)} X_j^\dagger f_S^*(X_j)\Big)^\top x_i\right],$$

and the optimal f_A, at every equilibrium, is the fixed point

$$f_A^*(x_i) = \frac{1}{1+\lambda}\left[y_i + \lambda(X_i^\dagger f_A^*(X_i))^\top x_i\right], \forall x_i \in \mathcal{D}_x.$$

The equilibrium in the linear class is not unique when the witness is not fully determined in a neighborhood due to degeneracy. To avoid these cases, we can use Ridge regression to obtain a stable equilibrium (proved also in Appendix).

A special case of Theorem 2 is when $x_i = [1], \forall x_i \in \mathcal{D}_x$, which effectively yields the equilibrium result for the constant class; we found it particularly useful to understand the similarity between the two games in this scenario. Concretely, each $X_j^\dagger f(X_j) x_i$ becomes equivalent to $\frac{1}{m}\sum_{x_k \in \mathcal{B}(x_j)} f(x_k)$. As a result, the solution for both the symmetric and asymmetric game induce the optimal predictors as recursive convolutional averaging of neighboring points with the same decay rate $\lambda/(1+\lambda)$, while the convolutional kernel evolves twice as fast in the symmetric game than in the asymmetric game.

Next, we show that the hard uniform constraint criterion yields a very different equilibrium.

Theorem 3. *If* **(A1-5)** *hold and the witness is in the linear family, the optimal f_U satisfies*

$$f_U^*(x_i) = \begin{cases} \alpha(x_i, f_U^*), & \textit{if } \alpha(x_i, f_U^*) > y_i, \\ \beta(x_i, f_U^*), & \textit{if } \beta(x_i, f_U^*) < y_i, \\ y_i, & \textit{otherwise,} \end{cases}$$

for $x_i \in \mathcal{D}_x$, where

$$\alpha(x_i, f_U^*) = \max_{x_j \in \mathcal{B}(x_i)} \Bigg[(X_j^\dagger f_U^*(X_j))^\top x_i - \sqrt{\delta m - \sum_{x_k \in \mathcal{B}(x_j)\setminus\{x_i\}} (f_U^*(x_k) - (X_j^\dagger f_U^*(X_j))^\top x_k)^2}\Bigg];$$

$$\beta(x_i, f_U^*) = \min_{x_j \in \mathcal{B}(x_i)} \Bigg[(X_j^\dagger f_U^*(X_j))^\top x_i + \sqrt{\delta m - \sum_{x_k \in \mathcal{B}(x_j)\setminus\{x_i\}} (f_U^*(x_k) - (X_j^\dagger f_U^*(X_j))^\top x_k)^2}\Bigg].$$

A noticeable difference from the games is that, under uniform criterion, the optimal predictor $f_U^*(x_i)$ may faithfully output the actual label y_i if the functional constraint is satisfied, while the functional constraints are translated into a "convolutional" operator in the games.

Efficient computation. We also analyze ways of accelerating the computation required for solving the symmetric game. An equivalent criterion is given by

Lemma 4. *If $d(\cdot,\cdot)$ is squared error, $\mathcal{L}(\cdot,\cdot)$ is differentiable, f is sub-differentiable, and* **A(4-5)** *hold, then*

$$\sum_{(x_i,y_i) \in \mathcal{D}} \mathcal{L}(f(x_i), y_i) + \frac{\lambda}{\bar{N}_i}\left[\bar{N}_i f(x_i) - \sum_{x_t \in \mathcal{B}(x_i)} \frac{\hat{g}_{x_t}(x_i)}{|\mathcal{B}(x_t)|}\right]^2,$$

where $\bar{N}_i := \sum_{x_t \in \mathcal{B}(x_i)} \frac{1}{|\mathcal{B}(x_t)|}$, induces the same equilibrium as the symmetric game.

The result is useful when training f on GPU and $\hat{g}_{x_i}$ is solved analytically on CPU. Compared to a for-loop to handle different neighborhood sizes for Eq. (3) on the GPU, computing a summarized feedback as in Lemma 4 on CPU is more efficient (and easier to implement).

Discussion We investigated here discrete neighborhoods and they are suitable also for structured data as in the experiments. The method itself can be generalized to continuous neighborhoods with an additional difficulty: the exact computation and minimization of functional deviation between the predictor and the witness in such neighborhood is in general intractable. We may apply results from learning theory (e.g., (Shamir, 2015)) to bound the (generalization) gap between the deviation computed by finite samples from the continuous neighborhood and the actual deviation under a uniform probability measure.

5. Examples

5.1. Conditional Sequence Generation

The basic idea of co-operative modeling extends naturally to conditional sequence generation over longer periods. Broadly, the mechanism allows us to inspect the temporal progression of sequences on a longer term basis.

Given an observation sequence $x_1, \dots, x_t \in \mathbb{R}^c$, the goal is to estimate probability $p(x_{t+1:T}|x_{1:t})$ over future events $x_{t+1}, \dots, x_T \in \mathbb{R}^c$, typically done via maximum likelihood. For brevity, we use $x_{1:i}$ to denote $x_1, \dots, x_i$. We model the conditional distribution of x_{i+1} given $x_{1:i}$ as a multivariate Gaussian distribution with mean $\mu(x_{1:i})$ and covariance $\Sigma(x_{1:i})$, both parametrized as recurrent neural networks. Each local witness model $g_{x_{1:i}}(\cdot)$ is estimated based on the neighborhood $\mathcal{B}(x_{1:i}) := \{x_{1:i-\epsilon}, \dots, x_{1:i+\epsilon}\}$ with respect to the mean function $\mu(\cdot)$. A natural choice would be a

K-order Markov autoregressive (AR) model with an ℓ_2 deviation loss as:

$$\min_{\theta} \sum_{x_{1:t}\in\mathcal{B}(x_{1:i})} \| \sum_{k=0}^{K-1} \theta_{k+1} \cdot x_{t-k} + \theta_0 - \mu(x_{1:t}) \|_2^2,$$

where $\theta_k \in \mathbb{R}^{c\times c}, \forall k > 0$ and $\theta_0 \in \mathbb{R}^c$. The AR model admits an analytical solution similar to linear regression.

5.2. Chemical Property Prediction

The models discussed in §3 can be instantiated on highly-structured data, such as molecules, too. These can be represented as a graph $\mathcal{M} = (\mathcal{V}, \mathcal{E})$ whose nodes encode the atom types and edges encode the chemical bonds. Such representation enables the usage of recent graph convolutional networks (GCNs) (Dai et al., 2016; Lei et al., 2017) as the predictor f. As it is hard to realize a simple explanation on the raw graph representation, we exploit an alternative data representation for the witness model; we leverage depth-bounded decision trees that take as input Morgan fingerprints (Rogers & Hahn, 2010) $x(\mathcal{M})$, which are binary vector features indicating the presence of various chemical substructures (e.g., the nodes in Fig. 1).

The neighborhood $\mathcal{B}(\mathcal{M})$ includes molecules $\{\mathcal{M}'\}$ with Tanimoto similarity greater than 0.6, automatically constructed through matching molecular pair analysis (Griffen et al., 2011). Here we use a multi-label binary classification task as an example, and adopt a cross-entropy loss for each label axis for simplicity. At each neighborhood $\mathcal{B}(\mathcal{M})$, we construct a witness decision tree g that minimizes the total variation (TV) from the predictor as

$$\min_{g\in\mathcal{G}_{\text{tree}}} \frac{1}{|\mathcal{B}(\mathcal{M})|} \sum_{\mathcal{M}'\in\mathcal{B}(\mathcal{M})} \sum_{i=1}^{\dim(\mathcal{Y})} |f(\mathcal{M}')_i - g(x(\mathcal{M}'))_i|. \quad (6)$$

Note that Eq. (6) is an upper bound and efficient alternative to fitting a tree for each label axis independently.

5.3. Molecule Representation Learning

Our approach can be further applied to learn transparent latent graph representations by variational autoencoders (VAEs) (Kingma & Welling, 2014; Jin et al., 2018). Concretely, given a molecular graph $\mathcal{M} = (\mathcal{V}, \mathcal{E})$, the VAE encoder q outputs the approximated posterior $z_{\mathcal{M}} \sim \mathcal{N}(\mu_{\mathcal{M}}, \Sigma_{\mathcal{M}})$ over the latent space, where $z_{\mathcal{M}}$ is the continuous representation of molecule $\mathcal{M}$. Following common practice, $\Sigma_{\mathcal{M}}$ is restricted to be diagonal. The VAE decoder then reconstructs the molecule $\mathcal{M}$ from its probabilistic encoding $z_{\mathcal{M}}$. Our goal here is to guide the behavior of the neural encoder q such that the derivation of (probabilistic) $z_{\mathcal{M}}$ can be locally explained by a decision tree.

We adopt the same setting for the witness function and neighborhoods as in §5.2, except that the local decision tree g now outputs a joint normal distribution with parameters $[\widehat{\mu}_{\mathcal{M}}, \widehat{\Sigma}_{\mathcal{M}}]$. To train the encoder, we extend the original VAE objective $\mathcal{L}^{\text{VAE}}$ with a local deviation loss $\mathcal{L}^{\mathcal{G}_{\text{tree}}}$ defined on the KL divergence between the VAE posterior $q(\mathcal{M}) = \mathcal{N}(\mu_{\mathcal{M}}, \Sigma_{\mathcal{M}})$ and witness posterior $g(x(\mathcal{M})) = \mathcal{N}(\widehat{\mu}_{\mathcal{M}}, \widehat{\Sigma}_{\mathcal{M}})$ at each neighborhood as

Table 1. Performance on the Tox-21 dataset. $\text{AUC}_{\mathcal{D}}(\hat{g}_{\mathcal{M}}, f)$ and $\text{AUC}_{\mathcal{B}}(\hat{g}_{\mathcal{M}}, f)$ generalize the AUC score to use f values as labels, computed on the testing data and their neighborhoods, respectively.

Aspect	Measure	GAME$_{\text{unif}}$	GAME$_{\text{sym}}$	DEEP
Performance	$\text{AUC}(f, y)$	0.744	**0.826**	0.815
(the higher the better)	$\text{AUC}(\hat{g}_{\mathcal{M}}, y)$	0.742	**0.824**	0.818
Transparency	$\text{AUC}_{\mathcal{B}}(\hat{g}_{\mathcal{M}}, f)$	**0.764**	0.759	0.735
(the higher the better)	$\text{AUC}_{\mathcal{D}}(\hat{g}_{\mathcal{M}}, f)$	0.959	**0.967**	0.922

$$\mathcal{L}^{\mathcal{G}_{\text{tree}}} := \frac{1}{|\mathcal{D}|} \sum_{\mathcal{M}\in\mathcal{D}} \min_{g\in\mathcal{G}_{\text{tree}}} \sum_{\mathcal{M}'\in\mathcal{B}(\mathcal{M})} \frac{\text{KL}(g(x(\mathcal{M}'))||q(\mathcal{M}'))}{|\mathcal{B}(\mathcal{M})|}$$

The VAE is trained to maximize $\mathcal{L}^{\text{VAE}} + \lambda \cdot \mathcal{L}^{\mathcal{G}_{\text{tree}}}$. For ease of optimization, we asymmetrically estimate each decision tree g with mean squared error between the vectors $[\mu_{\mathcal{M}}, \Sigma_{\mathcal{M}}]$ and $[\widehat{\mu}_{\mathcal{M}}, \widehat{\Sigma}_{\mathcal{M}}]$.

6. Experiments

We conduct experiments on chemical and time-series datasets. Due to the lack of existing works for explaining structured data, we adopt an ablation setting – comparing our approach (GAME) versus an unregularized model (DEEP) – and focus on measuring the transparency. We use subscripts to denote specific versions of the GAME models. Note that we only fit the local witnesses to the DEEP model during testing for evaluation. Unless otherwise noted, the reported results are based on the testing set.

6.1. Molecule Property Prediction

We conduct experiments on molecular toxicity prediction on the Tox21 dataset from MoleculeNet benchmark (Wu et al., 2018b), which contains 12 binary labels and $7,831$ molecules. The labels are very unbalanced; the fraction of the positive label is between 16.15% and 3.51% among the 12 labels. We use GCN as the predictor and decision trees as the witnesses as in §5.2. The neighborhood sizes m of about 60% of the molecules are larger than 2, whose median and maximum are 59 and 300, respectively. Since each neighborhood has a different size m, we set the maximum tree depth as $\max\{\lceil \log_2(m) \rceil - 1, 1\}$ for each neighborhood, which ensures that the corresponding size m is effective for $m > 2$ (see Definition 1). More details are in Appendix B.

Evaluation Measures: A more detailed discussion of how these measures are computed can be found in Appendix B.

(1) Performance: For the predictor, we compare its pre-

dictions with respect to the labels in AUC, denoted as AUC(f, y). As each local witness $\hat{g}_{\mathcal{M}}(x(\mathcal{M}))$ also realizes a function of $\mathcal{M}$, it is also evaluated against the labels in AUC, denoted as AUC$(\hat{g}_{\mathcal{M}}, y)$.

(2) Transparency: As labels are unavailable for testing data in practice, it is more realistic to measure the similarity between the predictor and the local witnesses to understand the validity of the explanations derived from the decision trees $\mathcal{G}$. To this end[1], we generalize the AUC criterion for continuous labels for N references y and predictions y' as

$$\sum_{i=1}^{N}\sum_{j=1}^{N}\mathbb{I}(y_i > y_j)\mathbb{I}(y'_i > y'_j)/\sum_{i=1}^{N}\sum_{j=1}^{N}\mathbb{I}(y_i > y_j).$$

The proposed score has the same pairwise interpretation as AUC, recovers AUC when y is binary, and is normalized to $[0, 1]$. Locally, we measure the criterion for the local witnesses with respect to the predictor in each testing neighborhood as the local deviation, where the average result is denoted as AUC$_{\mathcal{B}}(\hat{g}_{\mathcal{M}}, f)$. Globally, the criterion is also validated among the testing data, denoted as AUC$_{\mathcal{D}}(\hat{g}_{\mathcal{M}}, f)$.

The results with the uniform and symmetric criteria are shown in Table 1. A baseline vanilla decision tree, with depth tuned between 2 and 30, yields 0.617 in AUC(f, y). Compared to GAME$_{\text{sym}}$, the local deviation in GAME$_{\text{unif}}$ is marginally improved due to the strict constraint at the cost of severe performance loss. We investigate the behaviors in training neighborhoods and find that GAME$_{\text{sym}}$ exhibits a tiny fraction of high deviation losses, allowing the model to behave more flexibly than the strictly constrained GAME$_{\text{unif}}$ (see Figure 5 in Appendix B). In terms of performance, our GAME$_{\text{sym}}$ model is superior to the DEEP model in both the predictor and local witnesses. When comparing the witnesses to the predictor, locally and globally, the GAME models significantly improve the transparency from the DEEP model. The local deviation should be interpreted relatively since the tree depth inherently prevents local overfitting.

We visualize the resulting witness trees in Figure 1 under the same transparency constraint: for a local neighborhood, we grow the witness tree for the DEEP model until the local transparency in AUC$_{\mathcal{B}}$ is comparable to the GAME$_{\text{sym}}$ model. For explaining the same molecule, the tree for the DEEP model is deeper and extremely unbalanced. Since a Morgan fingerprint encodes the existence of a substructure of molecule graphs, an unbalanced tree focusing on the left branch (non-existence of a substructure) does not capture much generality. Hence, the explanation of the DEEP model does not provide as much insight as our GAME$_{\text{sym}}$ model.

Here we do an analysis on the tree depth constraint for the witness model, as a shallower tree is easier to interpret, but more challenging to establish transparency due to the restricted complexity. To this end, we revise the depth constraint to $\max\{\lceil\log_2(m)\rceil - 1 + \Delta, 1\}$ during training and testing, and vary $\Delta \in \{-3, \dots, 0\}$. All the resulting GAME models outperform the DEEP models in AUC(f, y), and we report the transparency score in terms of AUC$_{\mathcal{D}}(\hat{g}_{\mathcal{M}}, f)$ in Table 2. Even when $\Delta = -3$, the witness trees in our GAME model still represent the predictor more faithfully than those in the DEEP model with $\Delta = 0$.

Table 2. AUC$_{\mathcal{D}}(\hat{g}_{\mathcal{M}}, f)$ score on different Δ in the Tox-21 dataset (lower Δ implies shallower trees).

Model	$\Delta = 0$	$\Delta = -1$	$\Delta = -2$	$\Delta = -3$
GAME	0.967	0.967	0.964	0.958
DEEP	0.922	0.916	0.915	0.914

Table 3. Performance of the symmetric and asymmetric setting of the GAME model with $\epsilon = 9$.

$(\times 10^{-2})$	λ	0	0.1	1	10	100	AR
GAME$_{\text{asym}}$	Error	8.136	8.057	8.309	9.284	9.794	9.832
	Dev.	4.197	4.178	3.431	1.127	0.186	0.000
	TV	7.341	7.197	5.706	1.177	0.144	0.000
GAME$_{\text{sym}}$	Error	8.136	8.089	8.315	9.314	9.807	9.832
	Dev.	4.197	4.169	3.426	1.116	0.182	0.000
	TV	7.341	7.292	5.621	1.068	0.132	0.000

6.2. Physical Component Modeling

We next validate our approach on a physical component modeling task with the bearing dataset from NASA (Lee et al., 2016), which records 4-channel acceleration data on 4 co-located bearings. We divide the sequence into disjoint subsequences, resulting in $200,736$ subsequences. Since the dataset exhibits high frequency periods of 5 points and low frequency periods of 20 points, we use the first 80 points in an sequence to forecast the next 20. We parametrize $\mu(\cdot)$ and $\Lambda(\cdot)$ jointly by stacking 1 layer of CNN, LSTM, and 2 fully connected layers. We set the neighborhood radius ϵ to 9 such that the witnesses are fit with completely different data for the beginning and the end of the sequence. The Markov order K is set to 2 to ensure the effectiveness of the neighborhood sizes. More details are in Appendix C.

Evaluation involves three different types of errors: 1) 'error' is the root mean squared error (RMSE) between greedy autoregressive generation and the ground truth, 2) 'deviation' is RMSE between the predictor $\mu(x_{1:i})$ and the witness $\hat{g}_{x_{1:i}}(x_{1:i})$, and 3) 'TV' is the average total variation of witness $\hat{g}_{x_{1:i}}$ parameters $[\theta, \theta_0]$ between every two consecutive time points. Since the deviation and error are both computed on the same space in RMSE, the two measures are readily comparable. For testing, the witnesses are estimated based on the autoregressive generative trajectories.

[1] Since the predictor probability can be scaled arbitrarily to minimize the TV from decision trees without affecting performance, using TV to measure transparency as used in training is not ideal.

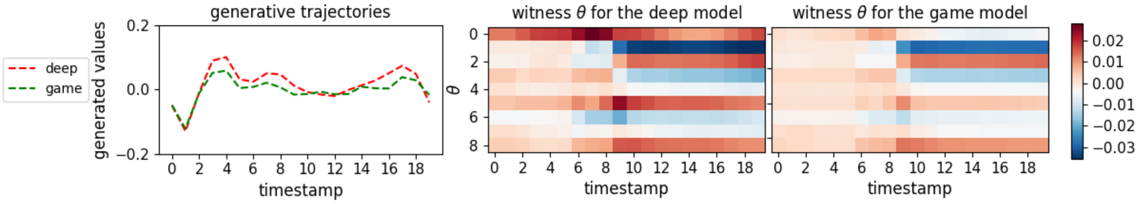


Figure 3. Visualizing the linear witnesses (middle and right plots) on the first channel (left plot) along the autoregressive generative trajectory (x-axis) on the bearing dataset. The y-axis of the parameters from 0 to 8 denotes the bias $(\theta_0)_1$ and weights $(\theta_1)_{1,1:4}, (\theta_2)_{1,1:4}$.

We present the results in Table 3 to study the impact of the game coefficient λ and the symmetry of the games. The trends in the measures are quite monotonic on λ: with an increasing λ, the model gradually operates toward the AR family with lower deviation and TV but higher error. When $\lambda = 0.1$, the GAME models are more accurate than the DEEP model ($\lambda = 0$) due to the regularization effect. Given the same hyper-parameters, marginally lower deviation in the symmetric game than in the asymmetric game confirms our analysis about the similarity between the two. In practice, the asymmetric game is more efficient and substantially easier to implement than the symmetric game. Indeed, the training time is 20.6 sequences/second for the asymmetric game, and 14.6 sequences/second for the symmetric game. If we use the formula in Lemma 4, the symmetric game can be accelerated to 20.4 sequences/second, but the formula does not generalize to other deviation losses.

We visualize the witnesses with their parameters $[\theta_0, \theta]$ along the autoregressive generative trajectories in Figure 3. The stable functional patterns of the GAME model as reflected by θ, before and after the 9th point, highlight not only close local alignments of the predictor and the AR family (being constant vectors across columns) but also flexible variation of functional properties on the predictor across regions. In contrast, the DEEP model yields unstable linear coefficients, and relies more on offsets/biases θ_0 than the GAME model, while the linear weights are more useful for grounding the coordinate relevance for interpretability. Finally, we remark that despite the uninterpretable nature of temporal signals, the functional pattern reflected by the linear weights as shown here yields a simple medium to understand its behavior. Due to space limitation, the additional analysis and visualization are included in Appendix C.

6.3. Molecule Representation Learning

Finally, we validate our approach on learning representations for molecules with VAEs, where we use the junction tree VAE (Jin et al., 2018) as an example. Here the encoders of VAEs, with and without the guidance of local decision trees as in §5.3, are denoted as DEEP and GAME, respectively. The models are trained on the ZINC dataset (Sterling & Irwin, 2015) containing 1.5M molecules, and evaluated on a test set with 20K molecules. We measure the performance in terms of the evidence lower bound (ELBO) over the test set. Here we consider two scenarios: the ELBO using the raw latent representations from the original neural encoder, and using the interpreted latent representations generated by locally fitted decision trees. The average deviation loss in KL divergence $\mathcal{L}^{\mathcal{G}_{\text{tree}}}$, defined in §5.3, over the testing neighborhoods is also evaluated.

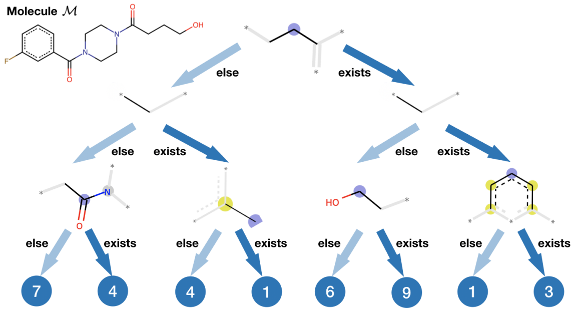


Figure 4. The local decision tree explains the latent representation for a molecule (upper left) by identifying locally discriminative chemical substructures. The leaf nodes are annotated with their sizes (number of molecules belonging to that cluster).

Table 4. The performance in ELBO for the raw neural encoders and locally adapted decision trees. The deviation is defined in §5.3.

Model	ELBO$_{\text{neural encoder}}$	ELBO$_{\text{decision tree}}$	deviation ($\mathcal{L}^{\mathcal{G}_{\text{tree}}}$)
DEEP	-21.6	-25.4	4.64
GAME	**-21.5**	**-25.1**	**3.98**

The results are shown in Table 4. Our GAME model performs consistently better under all the metrics. Figure 4 shows an example of how our decision tree explains the local neighborhood of a molecule. We found most of the substructures selected by the decision tree occur in the side chains outside of Bemis-Murcko scaffold (Bemis & Murcko, 1996). This shows the variation in the latent representation mostly reflects the local changes in the molecules, which is expected since changes in the scaffold typically lead to global changes such as chemical property changes.

7. Conclusion

We propose a novel game-theoretic approach to learning transparent models on structured data. The game articulates how the predictor model's fitting can be traded off against agreeing locally with a transparent witness. This work opens up many avenues for future work, from theoretical analysis of the games to a multi-player setting.

Acknowledgement

The work was funded in part by a grant from Siemens Corporation and in part by an MIT-IBM grant on deep rationalization.

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pp. 265–283, 2016.

Al-Shedivat, M., Dubey, A., and Xing, E. P. Contextual explanation networks. *arXiv preprint arXiv:1705.10301*, 2017.

Alvarez-Melis, D. and Jaakkola, T. S. A causal framework for explaining the predictions of black-box sequence-to-sequence models. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017.

Alvarez-Melis, D. and Jaakkola, T. S. On the robustness of interpretability methods. *Workshop on Human Interpretability in Machine Learning at ICML 2018; arXiv preprint arXiv:1806.08049*, 2018a.

Alvarez-Melis, D. and Jaakkola, T. S. Towards robust interpretability with self-explaining neural networks. In *Advances in Neural Information Processing Systems*, pp. 7786–7795, 2018b.

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 214–223, 2017.

Arras, L., Horn, F., Montavon, G., Müller, K.-R., and Samek, W. " What is relevant in a text document?": An interpretable machine learning approach. *PloS one*, 12(8): e0181142, 2017.

Bellemare, M. G., Danihelka, I., Dabney, W., Mohamed, S., Lakshminarayanan, B., Hoyer, S., and Munos, R. The cramer distance as a solution to biased wasserstein gradients. *arXiv preprint arXiv:1705.10743*, 2017.

Bemis, G. W. and Murcko, M. A. The properties of known drugs. 1. molecular frameworks. *Journal of medicinal chemistry*, 39(15):2887–2893, 1996.

Citron, D. K. and Pasquale, F. The scored society: due process for automated predictions. *Wash. L. Rev.*, 89:1, 2014.

Dai, H., Dai, B., and Song, L. Discriminative embeddings of latent variable models for structured data. In *Proceedings of the 33rd International Conference on Machine Learning*, pp. 2702–2711, 2016.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

Ghorbani, A., Abid, A., and Zou, J. Interpretation of neural networks is fragile. *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, 2019.

Goodfellow, I., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015.

Griffen, E., Leach, A. G., Robb, G. R., and Warner, D. J. Matched molecular pairs as a medicinal chemistry tool: miniperspective. *Journal of medicinal chemistry*, 54(22): 7739–7750, 2011.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pp. 5767–5777, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Helfrich, K., Willmott, D., and Ye, Q. Orthogonal recurrent neural networks with scaled cayley transform. *Proceedings of the 35th International Conference on Machine Learning*, 2018.

Huang, G., Liu, Z., v. d. Maaten, L., and Weinberger, K. Q. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2261–2269, 2017.

Jin, W., Barzilay, R., and Jaakkola, T. Junction tree variational autoencoder for molecular graph generation. *Proceedings of the 35th International Conference on Machine Learning*, 2018.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *International Conference on Learning Representations*, 2014.

Lakkaraju, H., Bach, S. H., and Leskovec, J. Interpretable decision sets: A joint framework for description and prediction. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1675–1684. ACM, 2016.

Lee, G.-H., Alvarez-Melis, D., and Jaakkola, T. S. Towards robust, locally linear deep networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SylCrnCcFX.

Lee, J., Qiu, H., Yu, G., Lin, J., and Rexnord Technical Services (2007). IMS, U. o. C. Bearing data set. *NASA Ames Prognostics Data Repository (http://ti.arc.nasa.gov/project/prognostic-data-repository), NASA Ames Research Center, Moffett Field, CA*, 7(8), 2016.

Lei, T., Barzilay, R., and Jaakkola, T. Rationalizing Neural Predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 107–117, 2016. URL http://arxiv.org/abs/1606.04155.

Lei, T., Jin, W., Barzilay, R., and Jaakkola, T. Deriving neural architectures from sequence and graph kernels. *Proceedings of the 34th International Conference on Machine Learning*, 2017.

Mahendran, A. and Vedaldi, A. Understanding deep image representations by inverting them. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.

Miller, J. and Hardt, M. When recurrent models don't need to be recurrent. *arXiv preprint arXiv:1805.10369*, 2018.

Mroueh, Y., Li, C.-L., Sercu, T., Raj, A., and Cheng, Y. Sobolev gan. *International Conference on Learning Representations*, 2018.

Müller, A. Integral probability metrics and their generating classes of functions. *Advances in Applied Probability*, 29 (2):429–443, 1997.

Nemirovski, A. Prox-method with rate of convergence o (1/t) for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15(1): 229–251, 2004.

Pasquale, F. *The black box society: The secret algorithms that control money and information*. Harvard University Press, 2015.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

Quinlan, J. R. *C4. 5: programs for machine learning*. Elsevier, 2014.

Ribeiro, M. T., Singh, S., and Guestrin, C. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4232-2. doi: 10.1145/2939672.2939778. URL http://arxiv.org/abs/1602.04938http://doi.acm.org/10.1145/2939672.2939778.

Rogers, D. and Hahn, M. Extended-connectivity fingerprints. *Journal of chemical information and modeling*, 50 (5):742–754, 2010.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. *https://arxiv. org/abs/1610.02391 v3*, 7(8), 2016.

Shamir, O. The sample complexity of learning linear predictors with the squared loss. *The Journal of Machine Learning Research*, 16(1):3475–3486, 2015.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

Sterling, T. and Irwin, J. J. Zinc 15–ligand discovery for everyone. *Journal of chemical information and modeling*, 55(11):2324–2337, 2015.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.

Wu, M., Hughes, M. C., Parbhoo, S., Zazzi, M., Roth, V., and Doshi-Velez, F. Beyond sparsity: Tree regularization of deep models for interpretability. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018*, 2018a. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16285.

Wu, Z., Ramsundar, B., Feinberg, E. N., Gomes, J., Geniesse, C., Pappu, A. S., Leswing, K., and Pande, V. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018b.

Yeh, C.-K., Kim, J., Yen, I. E.-H., and Ravikumar, P. K. Representer point selection for explaining deep neural networks. In *Advances in Neural Information Processing Systems*, pp. 9291–9301, 2018.

Zhao, M., Yue, S., Katabi, D., Jaakkola, T. S., and Bianchi, M. T. Learning sleep stages from radio signals: a conditional adversarial architecture. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 4100–4109, 2017.

A. Proofs

Our main results in this section make the following assumptions.

(A1) the predictor f is unconstrained.

(A2) both the loss and deviation are squared errors.

(A3) $|\mathcal{B}(x_i)| = m, \forall x_i \in \mathcal{D}_x$.

(A4) $x_j \in \mathcal{B}(x_i) \implies x_i \in \mathcal{B}(x_j), \forall x_i, x_j \in \mathcal{D}_x$.

(A5) $\cup_{x_i \in \mathcal{D}_x} \mathcal{B}(x_i) = \mathcal{D}_x$.

We note that **(A3)** and **(A4)** are not technically necessary but simplify the presentation. We denote the predictor in the uniform criterion (Eq. (2)), the symmetric game (Eq. (3)), and the asymmetric (Eq. (4)) game as f_U, f_S, and f_A, respectively. We use $X_i \in \mathbb{R}^{m \times d}$ to denote the neighborhood $\mathcal{B}(x_i) = \{x'_1, \dots, x'_m\}$ ($X_i = [x'_1, \dots, x'_m]^\top$), and $f(X_i) \in \mathbb{R}^m$ to denote the vector $[f(x'_1), \dots, f(x'_m)]^\top$. $X_j^\dagger$ denotes the pseudo-inverse of X_j. Then we have

Theorem 2. *If* **(A1-5)** *hold and the witness is in the linear family, the optimal f_S satisfies*

$$f^*_S(x_i) = \frac{1}{1+\lambda}\Big[y_i + \frac{\lambda}{m}\big(\sum_{x_j \in \mathcal{B}(x_i)} X_j^\dagger f^*_S(X_j)\big)^\top x_i\Big],$$

and the optimal f_A, at every equilibrium, is the fixed point

$$f^*_A(x_i) = \frac{1}{1+\lambda}\Big[y_i + \lambda(X_i^\dagger f^*_A(X_i))^\top x_i\Big], \forall x_i \in \mathcal{D}_x.$$

Proof. We first re-write the symmetric criterion explicitly as a game:

$$\min_f \sum_i (f(x_i) - y_i)^2 + \frac{\lambda}{m} \sum_{x_j \in \mathcal{B}(x_i)} (f(x_j) - \hat{g}_{x_i}(x_j))^2,$$

where $\hat{g}_{x_i}$ is the best response strategy from the local witness.

Since f is unconstrained and the objective in convex in it, we can treat each $f(x_i)$ as a distinct variable, and use the derivative to find its optimum:

$$\begin{aligned} f^*_S(x_i) &= \frac{1}{1+\lambda}\Big[y_i + \frac{\lambda}{m} \sum_{x_j \in \mathcal{B}^{-1}(x_i)} \hat{g}_{x_j}(x_i)\Big] \\ &= \frac{1}{1+\lambda}\Big[y_i + \frac{\lambda}{m} \sum_{x_j \in \mathcal{B}(x_i)} \hat{g}_{x_j}(x_i)\Big], \qquad (7) \end{aligned}$$

where $\mathcal{B}^{-1}(x_i) = \{x_j \in \mathcal{D}_x : x_i \in \mathcal{B}(x_j)\}$. Note that we only have to collect witnesses $\hat{g}_{x_j}$ that are relevant to $f(x_i)$ for the first equality, and the second equality is due to **(A4)**. On the other hand, the objective for f in the asymmetric game is:

$$\min_f \sum_i (f(x_i) - y_i)^2 + \lambda(f(x_i) - \hat{g}_{x_i}(x_i))^2,$$

The corresponding optimum is:

$$f^*_A(x_i) = \frac{1}{1+\lambda}\Big[y_i + \lambda \hat{g}_{x_i}(x_i)\Big] \qquad (8)$$

For both games, the objective for g_{x_i} can be described as:

$$\begin{aligned} &\min_{g_{x_i}} \frac{\lambda}{m} \sum_{x_j \in \mathcal{B}(x_i)} (f(x_j) - g_{x_i}(x_j))^2 \\ &= \min_{\theta_i} \frac{\lambda}{m} \|f(X_i) - X_i\theta_i\|_2^2, \qquad (9) \end{aligned}$$

Then Eq. (10) is an optimal witness $g^*_{x_i}$ at x_i.

$$g^*_{x_i}(x_j) = \theta_i^\top x_j = (X_i^\dagger f(X_i))^\top x_j, \forall x_j \in \mathcal{X}, \qquad (10)$$

and we note that every optimal witness $g^*_{x_i}$ has the same values on $\mathcal{B}(x_i)$

Since the optimal $g^*_{x_i}$ is functionally dependent to f. we put Eq. (10) back to Eq. (7) to obtain the optimal condition for f^*_S (at equilibrium) as

$$f^*_S(x_i) = \frac{1}{1+\lambda}\Big[y_i + \frac{\lambda}{m}\big(\sum_{x_j \in \mathcal{B}(x_i)} X_j^\dagger f^*_S(X_j)\big)^\top x_i\Big].$$

Again, putting Eq. (10) back to Eq. (8), we obtain the optimal condition for f^*_A at equilibrium as

$$f^*_A(x_i) = \frac{1}{1+\lambda}\Big[y_i + \lambda(X_i^\dagger f^*_A(X_i))^\top x_i\Big].$$

□

Note that the equilibrium for the linear class is not unique when the solution of Eq. (9) is not unique: there may be infinitely many optimal solution to the witness in a neighborhood due to degeneracy. In this case, Theorem 2 adopts the minimum norm solution as used in the pseudo-inverse in Eq. (10). In this case, one may use Ridge regression instead to establish a strongly convex objective for the witness to ensure a unique solution, where the objective for the witness is rewritten as

$$\min_{\theta_i} \frac{\lambda}{m} \|f(X_i) - X_i\theta_i\|_2^2 + \alpha\|\theta_i\|_2^2, \qquad (11)$$

with a positive α.

Theorem 3. *If* **(A1-5)** *hold and the witness is in the linear family, the optimal f_U satisfies*

$$f^*_U(x_i) = \begin{cases} \alpha(x_i, f^*_U), & \textit{if } \alpha(x_i, f^*_U) > y_i, \\ \beta(x_i, f^*_U), & \textit{if } \beta(x_i, f^*_U) < y_i, \\ y_i, & \textit{otherwise,} \end{cases}$$

for $x_i \in \mathcal{D}_x$, where

$$\alpha(x_i, f_U^*) = \max_{x_j \in \mathcal{B}(x_i)} \Bigg[(X_j^\dagger f_U^*(X_j))^\top x_i - \sqrt{\delta m - \sum_{x_k \in \mathcal{B}(x_j) \backslash \{x_i\}} (f_U^*(x_k) - (X_j^\dagger f_U^*(X_j))^\top x_k)^2} \Bigg];$$

$$\beta(x_i, f_U^*) = \min_{x_j \in \mathcal{B}(x_i)} \Bigg[(X_j^\dagger f_U^*(X_j))^\top x_i + \sqrt{\delta m - \sum_{x_k \in \mathcal{B}(x_j) \backslash \{x_i\}} (f_U^*(x_k) - (X_j^\dagger f_U^*(X_j))^\top x_k)^2} \Bigg].$$

Proof. The objective for the uniform criterion is:

$$\min_f \sum_{i=1}^N (f(x_i) - y_i)^2 \tag{12}$$
$$s.t. \ \min_{g \in \mathcal{G}} \frac{1}{m} \sum_{x_j \in \mathcal{B}(x_i)} (f(x_j) - g(x_j))^2 \le \delta, \forall x_i \in \mathcal{D}_x.$$

Our strategy is to temporarily treat each g as a fixed function, and then replace it with its best response strategy.

Since f is unconstrained (in capacity), we can treat each $f(x_i)$ as a distinct variable for optimization. For each $f(x_i)$, we first filter its relevant criteria:

$$\min_{f(x_i)} (f(x_i) - y_i)^2$$
$$s.t. \ (f(x_i) - g_{x_j}(x_i))^2, \le \delta m - \sum_{x_k \in \mathcal{B}(x_j) \backslash \{x_i\}} (f(x_k) - g_{x_j}(x_k))^2, \forall x_j \in \mathcal{B}(x_i).$$

For any feasible f, we can further rewrite the constraint of $f(x_i)$ with respect to each x_j as:

$$g_{x_j}(x_i) - \sqrt{\delta m - \sum_{x_k \in \mathcal{B}(x_j) \backslash \{x_i\}} (f(x_k) - g_{x_j}(x_k))^2} \le f(x_i) \le g_{x_j}(x_i) + \sqrt{\delta m - \sum_{x_k \in \mathcal{B}(x_j) \backslash \{x_i\}} (f(x_k) - g_{x_j}(x_k))^2}.$$

Collectively, we can fold all the upper bounds of $f(x_i)$ as

$$f(x_i) \le \min_{x_j \in \mathcal{B}(x_i)} \Bigg[g_{x_j}(x_i) + \sqrt{\delta m - \sum_{x_k \in \mathcal{B}(x_j) \backslash \{x_i\}} (f(x_k) - g_{x_j}(x_k))^2} \Bigg].$$

All the lower bounds can be folded similarly.

Finally, since the objective for $f(x_i)$ is simply a squared error with an interval constraint, evidently if y_i satisfies the lower bounds and upper bounds, then $f_U^*(x_i) = y_i$. If

$$y_i > \min_{x_j \in \mathcal{B}(x_i)} \Bigg[g_{x_j}(x_i) + \sqrt{\delta m - \sum_{x_k \in \mathcal{B}(x_j) \backslash \{x_i\}} (f(x_k) - g_{x_j}(x_k))^2} \Bigg],$$

then we have

$$f_U^*(x_i) = \min_{x_j \in \mathcal{B}(x_i)} \Bigg[g_{x_j}(x_i) + \sqrt{\delta m - \sum_{x_k \in \mathcal{B}(x_j) \backslash \{x_i\}} (f(x_k) - g_{x_j}(x_k))^2} \Bigg].$$

Otherwise, we have

$$f_U^*(x_i) = \max_{x_j \in \mathcal{B}(x_i)} \Bigg[g_{x_j}(x_i) - \sqrt{\delta m - \sum_{x_k \in \mathcal{B}(x_j) \backslash \{x_i\}} (f(x_k) - g_{x_j}(x_k))^2} \Bigg].$$

For each g_{x_i} is in the linear class, Eq. (13) is an optimal solution.

$$g_{x_j}^*(x_i) = (X_j^\dagger f(X_j))^\top x_i, \forall x_i \in \mathcal{X}, \tag{13}$$

and we note that every optimal witness $g_{x_j}^*$ has the same values on $\mathcal{B}(x_j)$.

Since the optimal $g_{x_i}^*$ is functionally dependent to f, to obtain the optimal f_U^*, we combine our previous result with $g_{x_i}^*$ such that the optimality conditions for f and g_{x_i} are both satisfied. Finally, we have

$$f_U^*(x_i) = \begin{cases} \alpha(x_i, f_U^*), & \text{if } \alpha(x_i, f_U^*) > y_i, \\ \beta(x_i, f_U^*), & \text{if } \beta(x_i, f_U^*) < y_i, \\ y_i, & \text{otherwise,} \end{cases}$$

for $x_i \in \mathcal{D}_x$, where

$$\alpha(x_i, f_U^*) = \max_{x_j \in \mathcal{B}(x_i)} \Bigg[(X_j^\dagger f_U^*(X_j))^\top x_i - \sqrt{\delta m - \sum_{x_k \in \mathcal{B}(x_j) \backslash \{x_i\}} (f_U^*(x_k) - (X_j^\dagger f_U^*(X_j))^\top x_k)^2} \Bigg];$$

$$\beta(x_i, f_U^*) = \min_{x_j \in \mathcal{B}(x_i)} \Bigg[(X_j^\dagger f_U^*(X_j))^\top x_i + \sqrt{\delta m - \sum_{x_k \in \mathcal{B}(x_j) \backslash \{x_i\}} (f_U^*(x_k) - (X_j^\dagger f_U^*(X_j))^\top x_k)^2} \Bigg].$$

□

Lemma 4. *If $d(\cdot,\cdot)$ is squared error, $\mathcal{L}(\cdot,\cdot)$ is differentiable, f is sub-differentiable, and* **A(4-5)** *hold, then*

$$\sum_{(x_i,y_i)\in\mathcal{D}} \mathcal{L}(f(x_i),y_i) + \frac{\lambda}{\bar{N}_i}\left[\bar{N}_i f(x_i) - \sum_{x_t\in\mathcal{B}(x_i)} \frac{\hat{g}_{x_t}(x_i)}{|\mathcal{B}(x_t)|}\right]^2, \tag{14}$$

where $\bar{N}_i := \sum_{x_t\in\mathcal{B}(x_i)} \frac{1}{|\mathcal{B}(x_t)|}$, induces the same equilibrium as the symmetric game.

Proof. Since the criteria for the witness g_{x_i} are the same in the symmetric game and the proposed asymmetric criterion here, we only have to check for the optimality condition for the predictor f. Here we use $\nabla_\theta f(x)$ to denote the subgradient of f at x with respect to the underlying parameter θ, the optimality condition for Eq. (14) is

$$\begin{aligned}
0 \in &\sum_{(x_i,y_i)\in\mathcal{D}} \Bigg[\frac{\partial}{\partial f(x_i)}\mathcal{L}(f(x_i),y_i) \\
&\quad + 2\lambda\Big(\sum_{x_t\in\mathcal{B}(x_i)} \frac{f(x_i)}{|\mathcal{B}(x_t)|} - \sum_{x_t\in\mathcal{B}(x_i)} \frac{\hat{g}_{x_t}(x_i)}{|\mathcal{B}(x_t)|}\Big)\Bigg]\nabla_\theta f(x_i) \\
= &\sum_{(x_i,y_i)\in\mathcal{D}} \Bigg[\frac{\partial}{\partial f(x_i)}\mathcal{L}(f(x_i),y_i)\nabla_\theta f(x_i) \\
&\quad + \sum_{x_t\in\mathcal{B}(x_i)} \frac{2\lambda}{|\mathcal{B}(x_t)|}(f(x_i) - \hat{g}_{x_t}(x_i))\nabla_\theta f(x_i)\Bigg]
\end{aligned}$$

For the symmetric game, the optimality condition is

$$\begin{aligned}
0 \in &\sum_{(x_i,y_i)\in\mathcal{D}} \Bigg[\frac{\partial}{\partial f(x_i)}\mathcal{L}(f(x_i),y_i)\nabla_\theta f(x_i) \\
&\quad + \sum_{x_t\in\mathcal{B}(x_i)} \frac{2\lambda}{|\mathcal{B}(x_i)|}(f(x_t) - \hat{g}_{x_i}(x_t))\nabla_\theta f(x_t)\Bigg]
\end{aligned}$$

It is evident that the two conditions coincide if Eq. (15) is equal to Eq. (16).

$$\sum_{(x_i,y_i)\in\mathcal{D}} \sum_{x_t\in\mathcal{B}(x_i)} \frac{1}{|\mathcal{B}(x_i)|}(f(x_t) - \hat{g}_{x_i}(x_t))\nabla_\theta f(x_t) \tag{15}$$

$$\begin{aligned}
= &\sum_{x_t\in\cup_{x_i\in\mathcal{D}_x}\mathcal{B}(x_i)} \sum_{x_i\in\mathcal{B}^{-1}(x_t)} \\
&\qquad \frac{1}{|\mathcal{B}(x_i)|}(f(x_t) - \hat{g}_{x_i}(x_t))\nabla_\theta f(x_t) \\
= &\sum_{x_t\in\mathcal{D}_x} \sum_{x_i\in\mathcal{B}(x_t)} \frac{1}{|\mathcal{B}(x_i)|}(f(x_t) - \hat{g}_{x_i}(x_t))\nabla_\theta f(x_t) \\
= &\sum_{(x_i,y_i)\in\mathcal{D}} \sum_{x_t\in\mathcal{B}(x_i)} \frac{1}{|\mathcal{B}(x_t)|}(f(x_i) - \hat{g}_{x_t}(x_i))\nabla_\theta f(x_i),
\end{aligned} \tag{16}$$

where the first equality is simply re-ordering of the two summations, and the second equality is due to $x_t \in \mathcal{B}(x_i) \iff x_i \in \mathcal{B}(x_t)$ and $\cup_{x_i\in\mathcal{D}_x}\mathcal{B}(x_i) = \mathcal{D}_x$. □

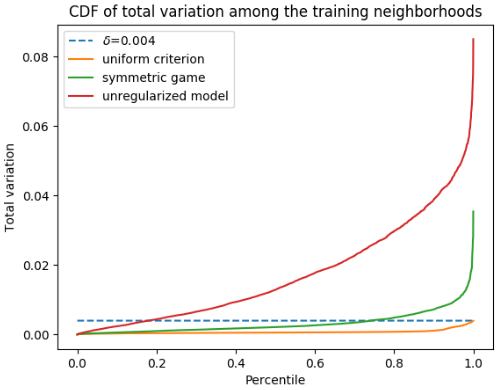


Figure 5. The cumulative distribution function of the total variation loss between the predictor f and the local witness g in each training neighborhood.

B. Supplementary Materials for Molecule Property Prediction

Implementation. To conduct training, we use GCNs as the predictor with 6 layers of graph convolution with 1800 hidden dimension. We use a $80\%/10\%/10\%$ split for training / validation / testing.

Evaluation Measures. We use the `roc_auc_score` in `scikit-learn` (Pedregosa et al., 2011) to compute the AUC score. Note that for each criterion, we evaluate the model with respect to each label, and then report the average score across the 12 labels. Here N denotes the number of testing data.

- AUC(f, y): we compare $f(\mathcal{M}_i)$ with the labels y_i among the testing data $\{(\mathcal{M}_i, y_i)\}_{i=1}^N$ in AUC.
- AUC$(\hat{g}_\mathcal{M}, y)$: we compare $\hat{g}_{\mathcal{M}_i}(x(\mathcal{M}_i))$ with the labels y_i among the testing data $\{(\mathcal{M}_i, y_i)\}_{i=1}^N$ in AUC.
- AUC$_\mathcal{B}(\hat{g}_\mathcal{M}, f)$: for each testing data $(\mathcal{M}, y)$, we evaluate the following score among the neighborhood $\mathcal{B}(\mathcal{M}) = \{\mathcal{M}_1, \ldots, \mathcal{M}_{N_\mathcal{M}}\}$, where $N_\mathcal{M} := |\mathcal{B}(\mathcal{M})|$, around $\mathcal{M}$:

  $$\frac{\sum_{i=1}^{N_\mathcal{M}}\sum_{j=1}^{N_\mathcal{M}} \mathbb{I}(f(\mathcal{M}_i) > f(\mathcal{M}_j))\mathbb{I}(\hat{g}_\mathcal{M}(\mathcal{M}_i) > \hat{g}_\mathcal{M}(\mathcal{M}_j))}{\sum_{i=1}^{N_\mathcal{M}}\sum_{j=1}^{N_\mathcal{M}} \mathbb{I}(f(\mathcal{M}_i) > f(\mathcal{M}_j))}.$$

  The average score across all the testing neighborhood is then reported.

- AUC$_\mathcal{D}(\hat{g}_\mathcal{M}, f)$: we evaluate the following score among the testing data $\{(\mathcal{M}_i, y_i)\}_{i=1}^N$:

  $$\frac{\sum_{i=1}^{N}\sum_{j=1}^{N} \mathbb{I}(f(\mathcal{M}_i) > f(\mathcal{M}_j))\mathbb{I}(\hat{g}_{\mathcal{M}_i}(\mathcal{M}_i) > \hat{g}_{\mathcal{M}_j}(\mathcal{M}_j))}{\sum_{i=1}^{N}\sum_{j=1}^{N} \mathbb{I}(f(\mathcal{M}_i) > f(\mathcal{M}_j))}.$$

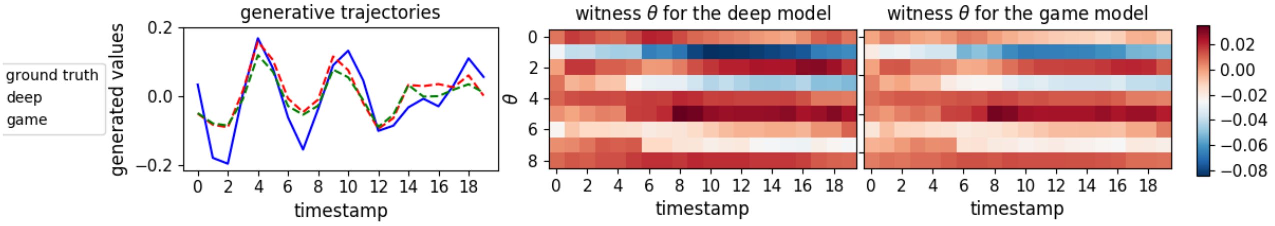


Figure 6. Visualization of the witnesses with the their parameters (middle and right plots) for *teacher-forced* predictions on the first channel (left plot) along each timestamp (x-axis) on the bearing dataset. The y-axis of the parameters from 0 to 8 denotes the bias $(\theta_0)_1$ and weights $(\theta_1)_{1,1:4}$, $(\theta_2)_{1,1:4}$.

Visualization. To investigate the behavior of the models, we plot their total variation loss from the local witness among the training neighborhoods in Figure 5. The uniform criterion imposes a strict functional constraint, while the symmetric game allows a more flexible model, exhibiting a tiny fraction of high deviation among the training neighborhoods.

C. Supplementary Materials for Physical Component Modeling

Implementation. We randomly sample 85%, 5%, and 10% of the data for training, validation, and testing. We set the learning rate as 10^{-5} with the Adam optimizer (Kingma & Ba, 2015). The batch size is set to 128. All the hidden dimensions are set to 128. We use the `MultivariateNormalTriL` function in Tensorflow (Abadi et al., 2016) to parametrize the multivariate Gaussian distribution. Specifically, we let the network output a $N + \frac{(N+1)(N)}{2}$ dimensional vector. The first N dimensions are treated as the mean. The second part is transformed to a lower triangular matrix, where the diagonal is further processed with a softplus nonlinearity. Such representation satisfies the Cholesky decomposition for covariance matrix.

For fitting the linear witness, we use Ridge regression in `scikit-learn` (Pedregosa et al., 2011) with the default hyperparameter. The usage of Ridge regression instead of vanilla linear regression is justified by our analysis of the equilibrium for linear witnesses.

Visualization. The visualization for the teacher-forced generative trajectory is in Figure 6.

Neighborhood size analysis

Here we investigate the effect of neighborhood radius ϵ. The results are shown in Figure 7. The impact of the neighborhood size is quite monotonic to deviation and TV, but in a reverse way. As ϵ increases, the weight of the witness on fitting the current point x_i among the neighborhood $\mathcal{B}(x_i)$ decreases, so the deviation of the witness $\hat{g}_{x_i}(x_i)$ from $f(x_i)$ increases. In contrast, as more points are overlapped between the neighborhoods of consecutive points, the resulting witnesses are more similar and thus yield smaller TV. In terms of prediction error, as the neighborhood radius ϵ determines the region to impose coherency, a larger region leads to greater restriction on the predictive model. All the arguments are well supported by the empirical results. We suggest users to trade off faithfulness (deviation) and smooth transition of functional properties (TV) based on the application at hand. We note that, however, smooth transition of functional properties is not equivalent to smoothness of f.

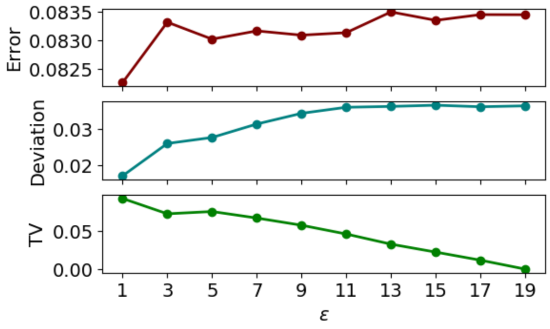


Figure 7. Parameter analysis of ϵ on the GAME model with $\lambda = 1$.

Finally, we remark that our sample complexity analysis for the linear class suggests that the neighborhood size is guaranteed to be effective for $2\epsilon + 1 > d = 2c + 1 = 9$. However, since the result is an sufficient condition, the regularization may still happens for $\epsilon < 5$ if the matrix rank of each neighborhood $X_i = [x_{i-\epsilon}, \ldots, x_{i+\epsilon}]^\top$ is less than $\min\{d, m\} = \min\{2c + 1, 2\epsilon + 1\}$.