Classifying the toxicity of pesticides to honey bees via support vector machines with random walk graph kernels

Ping Yang[1], E. Adrian Henle[1], Xiaoli Z. Fern[2], and Cory M. Simon[1*]

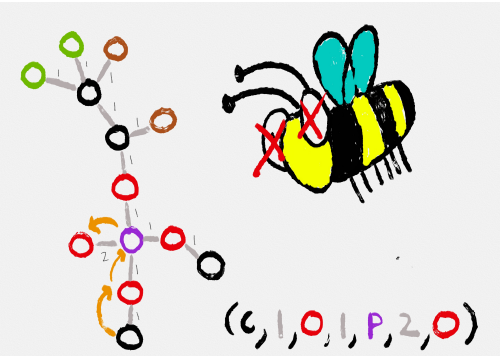
[1]School of Chemical, Biological, and Environmental Engineering. Oregon State University. Corvallis, OR. USA.
[2]School of Electrical Engineering and Computer Science. Oregon State University. Corvallis, OR. USA.
[*]Cory.Simon@oregonstate.edu

May 21, 2022

Abstract

Pesticides benefit agriculture by increasing crop yield, quality, and security. However, pesticides may inadvertently harm bees, which are valuable as pollinators. Thus, candidate pesticides in development pipelines must be assessed for toxicity to bees.

Leveraging a data set of 382 molecules with toxicity labels from honey bee exposure experiments, we train a support vector machine (SVM) to predict the toxicity of pesticides to honey bees. We compare two representations of the pesticide molecules: (i) a random walk feature vector listing counts of length-L walks on the molecular graph with each vertex- and edge-label sequence and (ii) the MACCS structural key fingerprint (FP), a bit vector indicating the presence/absence of a list of pre-defined subgraph patterns in the molecular graph. We explicitly construct the MACCS FPs, but rely on the fixed-length-L random walk graph kernel (RWGK) in place of the dot product for the random walk representation. The L-RWGK-SVM achieves an accuracy, precision, recall, and F1 score (mean over 2000 runs) of 0.81, 0.68, 0.71, and 0.69 on the test data set—with $L = 4$ the mode optimal walk length. The MACCS-FP-SVM performs on par/marginally better than the L-RWGK-SVM, lends more interpretability, but varies more in performance. We interpret the MACCS-FP-SVM by illuminating which subgraph patterns in the molecules tend to strongly push them towards the toxic/non-toxic side of the separating hyperplane.



1 Introduction

1.1 Pesticide toxicity to bees

Pesticides (incl. insecticides, fungicides, and herbicides) are used in agriculture as an economical means to control weeds, pests, and pathogens. Thereby, pesticides increase expected crop yield and quality and contribute to food security [1–4]. However, widespread pesticide use has negative externalities on both aquatic and terrestrial ecosystems and human health [5–9]. For example, pesticides can harm agriculturally beneficial species not deliberately targeted, such as earthworms and bees [10, 11].

Though under debate, extensive pesticide use in agriculture may play a role [11–16] in the widespread decline [17–20] of bee populations (see Ref. [21] for a synopsis) via both lethal and sublethal toxicity [11, 21]. Harms to bee populations are especially concerning because bees are valuable for agricultural production [22]: (1, primary value) bees serve as pollen vectors[1] for many crops, including fruits, vegetables, nuts, oilseed, spices, and coffee. [24, 25]; (2, secondary value [26]) honey bees produce honey and beeswax. In addition, bees are ecologically valuable as pollen vectors for plants in natural habitats [27].

Because insect [28], weed [29], and fungi [30] populations can develop resistance to an insecticide, herbicide, and fungicide, respectively, new pesticides must be continually discovered and deployed [1]. New pesticide development is also driven by the aim to reduce negative environmental impacts of incumbent pesticides [31].

Virtual screenings can accelerate the discovery of new pesticides operating under a known mechanism. For example, suppose an insect protein is a known target for insecticides. Then, computational protein-ligand docking [32] can score candidate compounds for insecticide activity, informing experimental campaigns. [33–39] However, newly proposed pesticides must also be assessed for toxicity to honey bees [40] (see US EPA website [41]).

A computational model that accurately predicts the toxicity of pesticides to bees would be useful [42] (i) as a toxicity filter in virtual and experimental screenings of compounds for pesticide activity; (ii) in emergency situations where an immediate assessment of toxicity risk is needed; and (iii) to focus scrutiny and motivate more thorough toxicity assessments on existing and new pesticides predicted to be toxic. Generally, training machine learning models to predict the toxicity of compounds to biological organisms is an active area of research [43, 44]. And, indeed, open data from bee toxicity experiments [45–57] have been leveraged to train machine learning models to computationally predict the toxicity of pesticides to bees [58–64].

1.2 Representing molecules for supervised machine learning tasks

A flurry of research activity is devoted to the data-driven prediction of the properties of molecules via supervised machine learning [65]. A starting point is to design a machine-readable represen-

[1] Bees visit the flowers of plants (angiosperms) to collect pollen or nectar as a food source. In the process, bees [inadvertently] transfer pollen from the anther of one flower to the stigma of another flower, a necessary step in the production of seeds and fruit for many plants. [23]

tation of the molecule for input to the machine learning model [66–68].

A vertex- and edge-labeled graph (vertices = atoms, edges = bonds, vertex label = element, edge label = bond order) is a fundamental representation of the concept of a small molecule[2]. However, because classical machine learning algorithms operate in a Euclidean vector space, much research is devoted to the design of fixed-size, information-rich *vector* representations of molecules that encode their salient features [66, 70–72]. Many molecular fingerprinting methods [72] extract topological features from the molecular graph [72] to produce a "bag of fragments" bit vector representation of the molecule [73]. For example, Molecular ACCess System (MACCS) structural key fingerprints [74] of a molecular graph are bit vectors indicating the presence or absence of a pre-defined list of subgraph patterns. Other hand-crafted molecular feature vectors include chemical, electronic, and structural/shape (3D) properties of the molecule as well [66, 73].

Two advanced supervised machine learning approaches circumvent explicit hand-crafting of vector representations of molecular graphs:

1. graph representation learning [75], such as message passing neural networks (MPNNs) [76, 77] that *learn* task-specific vector representations of molecular graphs for prediction tasks in an end-to-end manner

2. graph kernels [78–83], which (loosely speaking) measure the similarity between any two input graphs, allowing for the use of kernel methods [84], such as support vector machines [85], kernel regression/classification [84], and Gaussian processes [86], for prediction tasks.

I.e., MPNNs and kernel methods operate directly on the molecular graph representation, bypassing engineering and explicit construction, respectively, of molecular feature vectors for machine learning tasks.

MPNNs are powerful models for molecular machine learning tasks [76] but require large training data sets. In contrast, kernel methods with graph kernels are likely more appropriate when training data are limited, as they are easier to train, possess fewer hyper-parameters, and are less susceptible to overfitting [87]. Empirically, graph kernels give performance on par with MPNNs on a variety of molecular prediction tasks [88].

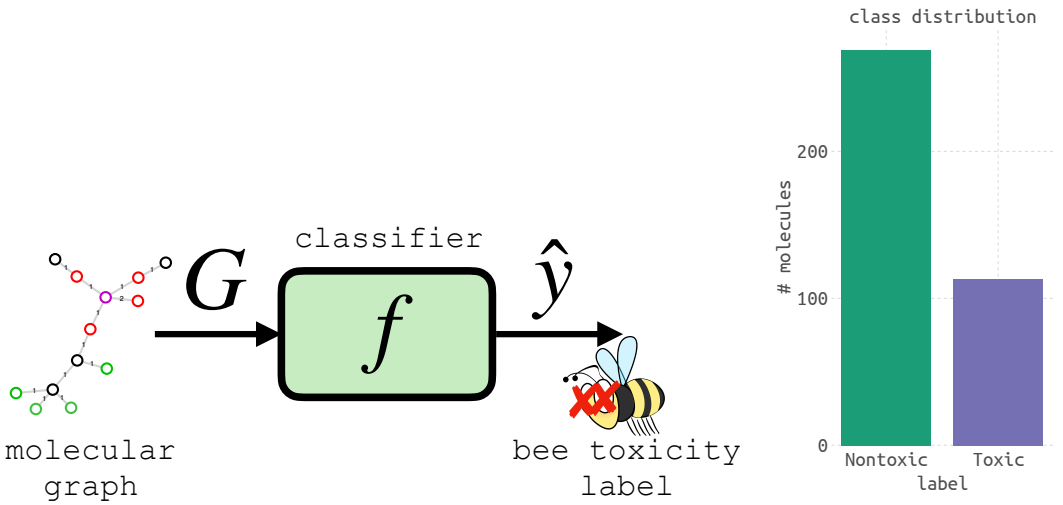
1.3 Our contribution: building a bee toxicity classifier of pesticides via the random walk graph kernel

We train and evaluate a support vector machine (SVM) classifier for predicting the toxicity of pesticide molecules to bees. Enabling a machine learning approach, the BeeToxAI project [58] compiled labeled data from bee toxicity experiments, composed of 382 (pesticide molecule, bee toxicity outcome) pairs. We compare two constructions of a molecular vector space for the

[2]For many classes of molecules, the mapping of the concept of a molecule to a molecular graph is one-to-one. If we wished to communicate a small molecule to an intelligent, extraterrestrial life form that has just arrived on earth and does not know our language, we would likely sketch a vertex- and edge-labeled, undirected graph. Though, molecular graph representations break down for certain classes of molecules [66] and are invariant to 3D structure and stereoisomerism [69].

SVM: (1) a random walk feature space, describing pesticides by the set of vertex- and edge-label sequences along length-L walks on their molecular graphs and (2) the MACCS fingerprint (FP) space, describing pesticides by the presence/absence of a list of pre-defined subgraph patterns in their molecular graphs. We explicitly construct the MACCS FPs but instead rely on the kernel trick and fixed-length-L random walk graph kernel (RWGK) for dot products in the random walk feature space. The L-RWGK-SVM achieves an F1 score (mean over 2000 runs) of 0.69 on the test data set, and $L = 4$ is the mode optimal walk length to describe the molecular graphs. The MACCS-FP-SVM performs on par/marginally better but exhibits more variance in its performance. Finally, we illuminate subgraphs in the pesticide molecules that tend to most strongly push molecules in the MACCS FP space towards the toxic/non-toxic side of the separating hyperplane of the MACCS-FP-SVM.

2 Problem setup: classifying the toxicity of a pesticide to honey bees



(a) The prediction task.

(b) Class distribution.

Figure 1: Problem setup. (a) Our objective is to train a classifier f that maps a molecular graph G to a binary prediction $\hat{y}$ of the bee toxicity of the pesticide molecule it represents. (b) The label distribution in the BeeToxAI [58] data set.

The pesticide toxicity classification task. We wish to construct a classifier $f : \mathcal{G} \to \{-1, 1\}$ that maps any molecular graph G (see Sec. 3.1) representing a pesticide molecule to a predicted binary label $\hat{y} = f(G)$, where $\hat{y} = 1$ is toxic to honey bees (*Apis mellifera*) and $\hat{y} = -1$ is nontoxic. The classifier f is valuable as a cheap-to-evaluate "surrogate model" of an expensive bee toxicity experiment. See Fig. 1a.

The labeled bee toxicity data set [58]. From the BeeToxAI project [58], we took labeled data $\{(G_n, y_n)\}_{n=1}^{N}$ composed of $N = 382$ examples of (i) a molecular graph $G_n \in \mathcal{G}$ representing

a pesticide or pesticide-like molecule and (ii) its experimentally-determined acute contact bee toxicity label $y_n \in \{-1, 1\}$ (1: toxic, -1: nontoxic).

Fig. 1b shows the class (im)balance; 113 of the molecules are labeled toxic, 269 nontoxic.

The outcome of a bee exposure experiment was mapped to a toxicity label on the pesticide following US EPA guidelines [89]: the pesticide was labeled as toxic if the median lethal dose (LD_{50}) after 48 hr to an adult honey bee was greater than 11 μg/bee—and nontoxic otherwise.

The data set includes neonicotinoid, pyrethroid, organophosphate, carbamate, pyridine azomethine, phenylpyrazole, and organochlorine insecticides [45–47, 49–54, 56, 57], herbicides [56], miticides [50–52], and fungicides [47, 48, 52, 53, 56]. Any molecules bearing tetrahedral chiral centers are not labeled with stereochemical configuration.

The machine learning approach: data-driven prediction of bee toxicity. Our objective is to leverage the labeled bee toxicity data set to train an SVM as the toxicity classifier $f(G)$. An SVM is a versatile supervised machine learning model that aims to find the maximum-margin separator (a hyperplane) between the positive and negative training examples in a mapped feature [vector] space. The mapped feature space does not need to be explicitly constructed. Instead, kernel functions can be used to [implicitly] perform the needed operation (dot product) in the mapped feature space. We compare two constructions of a molecular vector space by representing pesticide molecules with (1) the fixed-length random walk feature vector and (2) the MACCS fingerprint. The latter we explicitly construct, while for the former we rely on the fixed-length random walk graph kernel for the dot product.

3 Methods

3.1 The vertex- and edge-labeled graph representation of a molecule

A fundamental representation of a molecule is as a vertex- and edge-labeled, undirected graph $G = (\mathcal{V}, \mathcal{E}, \ell_v, \ell_e)$, with:

- $\mathcal{V} = \{v_1, ..., v_N\}$ the set of vertices representing its N atoms, excluding hydrogen atoms[3].
- $\mathcal{E}$ the set of edges representing chemical bonds; $\{v_i, v_j\} \in \mathcal{E}$ iff the atoms represented by vertices $v_i \in \mathcal{V}$ and $v_j \in \mathcal{V}$ are bonded.
- $\ell_v : \mathcal{V} \to \{\mathrm{C}, \mathrm{N}, \mathrm{O}, \mathrm{S}, \mathrm{P}, \mathrm{F}, \mathrm{Cl}, \mathrm{I}, \mathrm{Br}, \mathrm{Si}, \mathrm{As}\}$ the vertex-labeling function that provides the chemical element of each vertex (atom).
- $\ell_e : \mathcal{E} \to \{1, 2, 3, a\}$ ("a" for aromatic) the edge-labeling function that provides the bond order of each edge (bond).

[3]We exclude H atoms in the molecular graph to avoid redundancy. E.g., the hydrogen-excluding molecular graphs of ethane, ethylene, and acetylene can be distinguished by the order (an edge label) of the C-C bond (edge).

For example, see Fig. 2. This *molecular graph* representation of a molecule describes its topology and is invariant to translations and rotations of the molecule and to bond stretching, bending, and rotation.

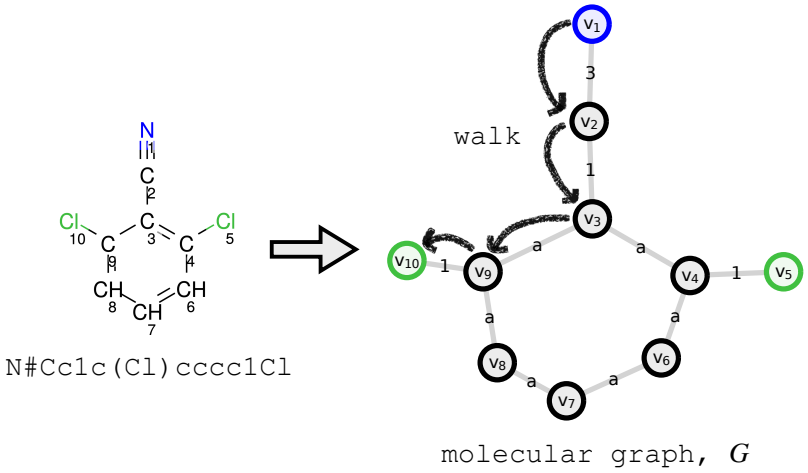


Figure 2: The molecular graph representation of 2,6-dichlorobenzonitrile (SMILES string shown). Nodes are labeled by atomic species (indicated by color). Edges are labeled by bond order. A length $L = 4$ walk $(v_1, v_2, v_3, v_9, v_{10})$ on the molecular graph is indicated by the arrows. The label sequence of this walk is $(N, 3, C, 1, C, a, C, 1, Cl)$.

Let $\mathcal{G}$ be the set of possible molecular graphs, so $G \in \mathcal{G}$.

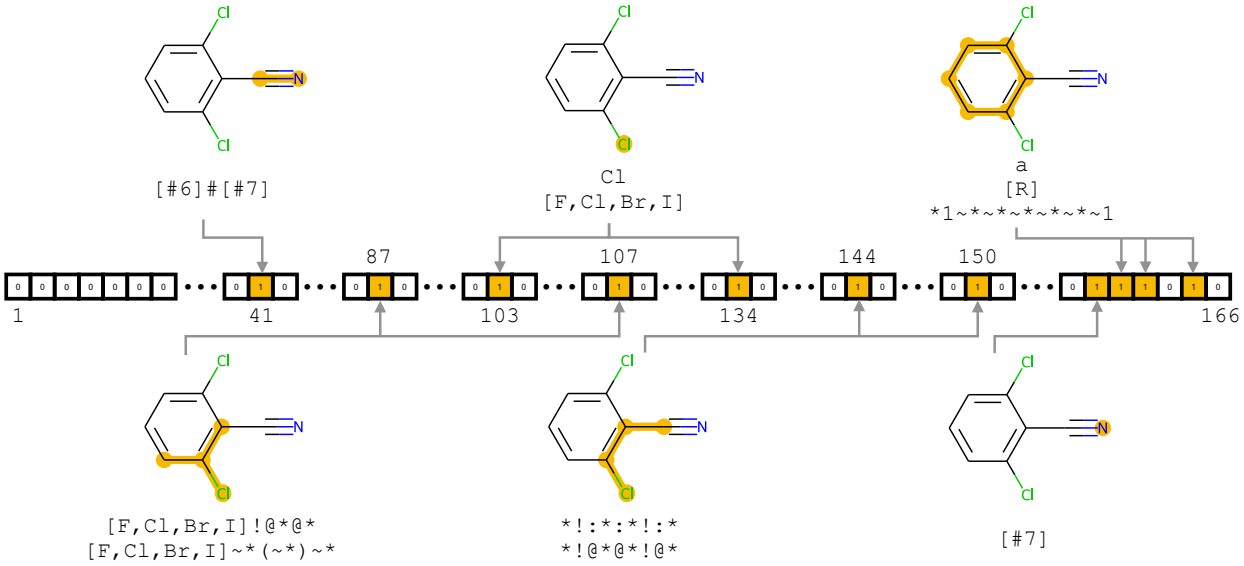
3.2 Two molecular vector spaces

We explore two *feature maps* $\phi : \mathcal{G} \to \mathbb{R}^F$ that map a molecular graph $G \in \mathcal{G}$ to a feature vector $\phi(G) \in \mathbb{R}^F$, with $\mathbb{R}^F$ the molecular vector space in which the SVM operates: (1) the MACCS structural key fingerprint and (2) the fixed-length random walk feature vector.

3.2.1 MACCS structural key fingerprint

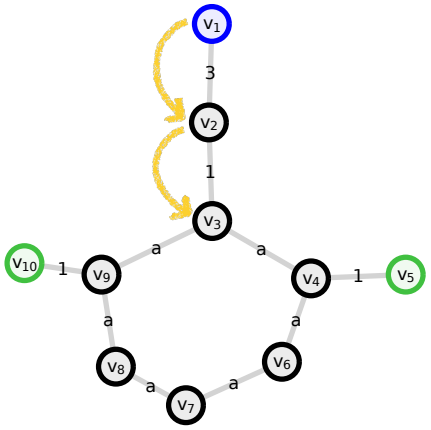
The Molecular ACCess System (MACCS) structural key fingerprint (FP) of a molecular graph is a bit vector whose entries indicate the presence (1) or absence (0) of a list of $F = 166$ pre-defined subgraph patterns (molecular substructures/fragments) [72, 74]. The number of "on" (1) bits in the MACCS FP is equal to the number of these subgraphs with presence in the molecule. The list of molecular patterns defining the MACCS feature map $\phi^{MACCS}(G) : \mathcal{G} \to \mathbb{R}^{166}$ were curated by a company, Molecular Design Limited, Inc. (MDL), for drug discovery tasks[4] [74]. We use the MACCS fingerprint implementation in `RDKit` [94], whose source code lists the subgraph patterns (described by SMARTS strings) corresponding to each keybit [95]. For example, keybits 29, 134, 125, and 154 indicate the presence of phosphorus, a halogen, an aromatic ring, and a carbonyl group, respectively. Fig. 3a illustrates further.

[4]Thus, we hypothesize the MACCS fingerprint encodes biologically-relevant information about pesticide molecules for predicting their toxicity to bees. Indeed, MACCS fingerprints have been found to be predictive of toxicity in other studies [64, 90–93].



(a)

label sequence, s	# L=2 walks
(N, 3, C, 1, C)	1
(N, 3, C, 3, N)	1
(Cl, 1, C, 1, Cl)	2
(Cl, 1, C, a, C)	4
(C, 3, N, 3, C)	1
(C, 1, Cl, 1, C)	2
(C, a, C, 1, Cl)	4
(C, 1, C, 3, N)	1
(C, a, C, a, C)	24
(C, a, C, 1, C)	2
(C, 1, C, a, C)	2
(C, 1, C, 1, C)	2



(b)

Figure 3: Illustrating the two molecular vector representations we employ for pesticides. (a) The MACCS fingerprint (FP) is a length-166 bit vector indicating the presence/absence of a predefined list of 166 subgraphs in a molecular graph. Shown here is the MACCS FP ("on" bits orange) of 2,6-dichlorobenzonitrile. Subgraphs of the molecular graph that activate bits of the MACCS FP are highlighted orange. The pattern(s) each subgraph matches is/are indicated in the SMARTS language. (b) The fixed-length random walk feature vector contains counts of label sequences encountered along all fixed-length walks on a molecular graph. The table lists all label sequences encountered along length $L = 2$ walks on the molecular graph of 2,6-dichlorobenzonitrile. The numbers in the second column, giving the number of walks having each label sequence, comprise the nonzero entries of the random walk feature vector of 2,6-dichlorobenzonitrile.

3.2.2 The fixed-length random walk feature vector

3.2.2.1 Walks on a molecular graph and label sequences along them. The random walk feature map describes a molecular graph by the set of label sequences along walks on it.

A walk. A *walk* w of length L on a molecular graph G is a sequence of vertices such that consecutive vertices are joined by an edge:

$$w = (v_1, ..., v_{L+1}) \text{ s.t. } \{v_i, v_{i+1}\} \in \mathcal{E} \text{ for } i \in \{1, ..., L\} \tag{1}$$

The length L refers to the number of edges (not necessarily unique) traversed along the walk. E.g., see Fig. 2 and caption.

Let $\mathcal{W}_L(G)$ be the set of all possible walks of length L on a graph G.

The label sequence of a walk. The *label sequence* $s = \ell_w(w)$ of a walk $w = (v_1, ..., v_{L+1})$ gives the progression of vertex and edge labels along the walk:

$$\ell_w(w) = [\ell_v(v_1), \ell_e(\{v_1, v_2\}), ..., \ell_e(\{v_L, v_{L+1}\}), \ell_v(v_{L+1})] =: s. \tag{2}$$

E.g., see Fig. 2 and caption again.

Let $\mathcal{S}_L = \{s_1, ..., s_{S_L}\}$ be the set of all possible label sequences among length-L walks on all molecular graphs $G \in \mathcal{G}$—so $|\mathcal{S}_L| = S_L$.

3.2.2.2 The fixed-length random walk feature map.

The fixed-length random walk feature vector of a molecular graph lists the number of fixed-length walks on the graph with each possible vertex- and edge-label sequence. Thereby, the molecular graph is described by the distribution of label sequences along fixed-length (equipoise [96]) random walks on it [97–99].

Precisely, the fixed-length-L feature map $\phi^{(L)} : \mathcal{G} \to \mathbb{R}^{S_L}$ constructs a vector representation of a graph $G \in \mathcal{G}$ whose element i is a count of length-L walks on G with label sequence s_i:

$$\phi^{(L)}(G) := [\phi_1^L(G), ..., \phi_{S_L}^{(L)}(G)] \tag{3}$$

$$\text{where } \phi_i^{(L)}(G) := |\{w \in \mathcal{W}_L(G) : \ell_w(w) = s_i\}|. \tag{4}$$

As a length $L = 0$ walk constitutes an atom, $\phi^{(0)}(G)$ lists counts of atom types in the molecule. As a length $L = 1$ walk constitutes two (ordered) atoms joined by a bond, $\phi^{(1)}(G)$ lists counts of each particular (ordered) pairing of atoms joined by a particular bond type in the molecule.

Fig. 3b illustrates by listing (in an arbitrary order) the nonzero elements of $\phi^{(L=2)}(G)$ for an example molecular graph G. In both eqn. 4 and Fig. 3b, we dodge the task of explicitly imposing an ordering of the set $\mathcal{S}_L$, since we never explicitly construct $\phi^{(L)}(G)$.

3.2.3 Comparing and contrasting the MACCS FP and random walk feature vector

Both the MACCS FP and random walk feature vector characterize a molecular graph by looking for a list of "patterns" in it—subgraph patterns for the MACCS FP and label sequences along walks for the random walk feature vector. Distinctions are: (1) the MACCS FP looks for *variable-size* subgraphs, whereas the random walk feature vector looks at *fixed-size* (length) label sequences along walks; (2) the random walk feature vector *counts* patterns, whereas the MACCS FP only

indicates the presence of patterns; (3) the random walk feature vector exhaustively counts *all possible* walk patterns, while the MACCS FP non-exhaustively looks for a pre-defined, curated *subset* of the possible subgraph patterns; (4) owing to the variably-sized list of subgraph patterns including wildcard atoms/bonds, in the MACCS FP, multiple subgraphs can activate the same bit, and a single subgraph can activate multiple bits; (5) the MACCS FPs $\phi^{MACCS}(G) \in \mathbb{R}^{166}$ are feasible to explicitly construct and store in memory, while the fixed-length random walk feature vectors $\phi^{(L)}(G) \in \mathbb{R}^{S_L}$ are not for large L, owing to the large number of possible label sequences S_L present in length-L walks on molecular graphs[5] [98].

3.3 The fixed-length random walk graph kernel

The fixed-length random walk kernel [98] $k^{(L)}(G, G') = \phi^{(L)}(G) \cdot \phi^{(L)}(G')$ allows us to circumvent explicit construction of $\phi^{(L)}(G)$ when employing a kernel method of machine learning, which can be cast to rely only on dot products $\phi^{(L)}(G) \cdot \phi^{(L)}(G')$ of pairs of vector representations of molecular graphs G, G'.

3.3.1 Definition and explanation of the L-RWGK

The fixed length-L random walk graph kernel [97, 100] (L-RWGK) $k^{(L)} : \mathcal{G} \times \mathcal{G} \to \mathbb{R}$ is a (symmetric, positive semidefinite) function such that evaluating $k(G, G')$ is implicitly equivalent to (i) mapping the two input graphs G and G' into the random walk vector space $\mathbb{R}^{S_L}$ via the feature map $\phi^{(L)}$ then (ii) taking the inner product of these two vectors:

$$k^{(L)}(G, G') = \phi^{(L)}(G) \cdot \phi^{(L)}(G'). \tag{5}$$

Seen from eqn. 4, term i of $k^{(L)}(G, G')$ in eqn. 5 is the number of pairs of length-L walks—one in graph G, the other in graph G'—with label sequence $s_i \in \mathcal{S}_L$. So, $k^{(L)}(G, G')$ sums counts of pairs of length-L walks on the two graphs G, G' sharing a label sequence:

$$k^{(L)}(G, G') = \sum_{s \in \mathcal{S}_L} |\{w \in \mathcal{W}_L(G) : \ell_w(w) = s\}||\{w' \in \mathcal{W}_L(G') : \ell'_w(w') = s\}|. \tag{6}$$

As the term associated with a label sequence s is nonzero iff *both* graphs G and G' possess a length-L walk with label sequence s, this sum may be restricted to be over the subset of label sequences in common between length-L walks on the two graphs, $\ell_w(\mathcal{W}_L(G)) \cap \ell'_w(\mathcal{W}_L(G'))$.

Intuitively, the 0-RWGK $k^{(0)}(G, G')$ sums counts of pairs of atoms of a particular type between the two graphs G, G'. The 1-RWGK $k^{(1)}(G, G')$ sums counts of pairs of two particular (ordered) atoms joined by a particular bond.

To evaluate the L-RWGK $k^{(L)}(G, G')$ without explicitly constructing the random walk feature vectors $\phi^{(L)}(G), \phi^{(L)}(G')$, we leverage the direct product graph to count pairs of label sequences in common between walks on two graphs G, G'.

[5] Given V possible vertex labels and E possible edge labels, theoretically $|\mathcal{S}_L| = V^{L+1}E^L$ label sequences are possible in length-L walks, although many of these will not be observed in any plausible chemical system.

3.3.2 The direct product graph to compute RWGKs

Given two input graphs $G, G' \in \mathcal{G}$, we construct a new graph, the direct product graph $G_\times = G \times G' = (\mathcal{V}_\times, \mathcal{E}_\times, \ell_{v,\times}, \ell_{e,\times})$, to evaluate the L-RWGK $k^{(L)}(G, G')$ between G and G'. The direct product graph $G_\times$ is constructed to give a one-to-one mapping between (i) walks in $G_\times$ and (ii) pairs of walks—one on G and one on G'—with the same label sequence.

Definition of the direct product graph. Each vertex of the direct product graph $G_\times = G \times G'$ is an ordered pair of vertices—the first in G, the second in G'. The vertices of the direct product graph are constituted by the subset of pairs of vertices between G and G' with the same vertex label:

$$\mathcal{V}_\times := \{(v, v') \in \mathcal{V} \times \mathcal{V}' \mid \ell_v(v) = \ell'_v(v')\}. \tag{7}$$

An undirected edge joins two vertices of the direct product graph $G_\times = G \times G'$ iff (i) the two involved vertices of G are joined by an edge in $\mathcal{E}$ and (ii) the two involved vertices of G' are joined by an edge in $\mathcal{E}'$ and (iii) these two edges in $\mathcal{E}$ and $\mathcal{E}'$ have the same label:

$$\begin{aligned}\mathcal{E}_\times := \{\{(u, u'), (v, v')\} \mid\ & (u, u') \in \mathcal{V}_\times \ \wedge\ (v, v') \in \mathcal{V}_\times \wedge \\ & \{u, v\} \in \mathcal{E} \ \wedge\ \{u', v'\} \in \mathcal{E}' \wedge \\ & \ell_e(\{u, v\}) = \ell'_e(\{u', v'\})\}.\end{aligned} \tag{8}$$

We equip the direct product graph $G_\times = G \times G'$ with vertex- and edge-labeling functions that give the (same) label of the involved vertices and edges in G and G':

$$\ell_{v,\times}((v, v')) := \ell_v(v) = \ell'_v(v') \tag{9}$$

$$\ell_{e,\times}(\{(u, u'), (v, v')\}) := \ell_e(\{u, v\}) = \ell'_e(\{u', v'\}). \tag{10}$$

Fig. 4 shows the direct product graph of two molecular graphs as an example.

Utility of the direct product graph for evaluating the L-RWGK. By construction, any given length-L walk $w_\times$ on the direct product graph $G_\times = G \times G'$ with label sequence $\ell_{w,\times}(w_\times)$ corresponds to a unique pair of walks $\{w, w'\}$, with $w \in \mathcal{W}_L(G), w' \in \mathcal{W}_L(G')$, possessing the label sequence $\ell_w(w) = \ell'_w(w') = \ell_{w,\times}(w_\times)$, and vice versa (giving a bijection). Fig. 4 illustrates. Therefore, all three of the following quantities are equivalent:

- the number of length-L walks on the direct product graph $G_\times = G \times G'$
- the number of pairs of length-L walks on G and G' with the same label sequence
- via eqn. 6, the value of the L-RWGK $k^{(L)}(G, G')$.

The key to counting length-L walks on $G_x = G \times G'$—and thus to evaluating $k^{(L)}(G, G')$—lies in its $|\mathcal{V}_x| \times |\mathcal{V}_x|$ adjacency matrix A_x whose entry (i, j) is one if vertices $v_{x,i}, v_{x,j} \in \mathcal{V}_x$ are joined

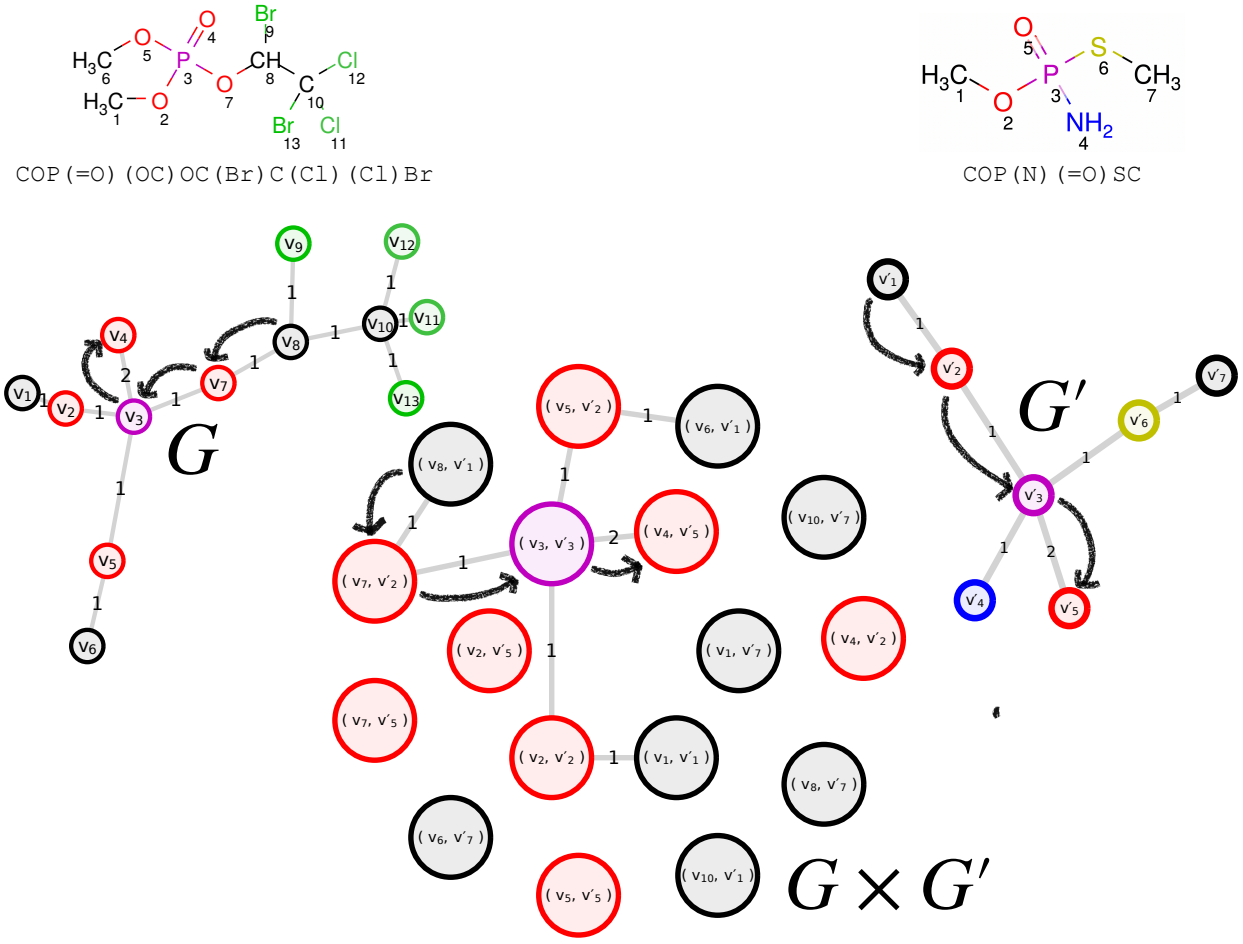


Figure 4: Illustrating the direct product graph $G_\times = G \times G'$ of two molecular graphs G and G' representing two molecules (shown above their SMILES strings) in the BeeToxAI data set. Vertex labels in the graphs are indicated by color. Note the one-to-one correspondence between (i) a walk on $G_\times$ and (ii) two walks on G and G' with the same label sequence. We indicate one such correspondence with the black arrows.

by an edge and zero otherwise. The number of walks of length L from vertex $v_{x,i}$ to vertex $v_{x,j}$ is given by element (i, j) of A_x^L. Summing over all possible starting and end vertices of walks:

$$k^{(L)}(G, G') = \sum_{i=1}^{|\mathcal{V}_\times|} \sum_{j=1}^{|\mathcal{V}_\times|} [A_\times^L]_{i,j}. \tag{11}$$

Summary of evaluating the L-RWGK. Computing the L-RWGK $k^{(L)}(G, G')$, therefore, involves (i) constructing the direct product graph $G_\times = G \times G'$, (ii) building the adjacency matrix $A_\times$ of $G_\times$, (iii) computing the L-th power of $A_\times$, $A_\times^L$, then (iv) summing its entries.

3.4 The linear kernel between two MACCS structural key fingerprints

For comparison to the L-RWGK, note the (linear) kernel applied to the MACCS FPs of a pair of molecular graphs:

$$k^{MACCS}(G, G') := \phi^{MACCS}(G) \cdot \phi^{MACCS}(G') \tag{12}$$

gives the number of subgraph patterns in the MACCS library that are exhibited by *both* graphs G and G'.

3.5 Support vector machines (SVMs) as classifiers

A support vector machine (SVM) [84, 85, 101, 102] is a supervised machine learning model for binary classification. To train an SVM using a labeled training data set $\{(G_n, y_n)\}_{n=1}^N$, with $G_n \in \mathcal{G}$ and $y_n \in \{-1, 1\}$, we rely on a feature map $\phi : \mathcal{G} \to \mathbb{R}^F$ to represent graphs in a vector space. Such feature maps can be constructed explicitly (the case with the MACCS FP), or implicitly via the use of a kernel function $k(G, G') = \phi(G) \cdot \phi(G')$ between pairs of data (the case with the random walk feature vector). We briefly explain the SVM here. For more details, consult Refs. [84, 101].

The decision boundary. Ultimately, an SVM classifier $f(G)$ employs a hyperplane $w \cdot \phi(G) + b = 0$ in the feature space $\mathbb{R}^F$ as the decision boundary:

$$\hat{y} = f(G) = \text{sign}(w \cdot \phi(G) + b), \tag{13}$$

with $w \in \mathbb{R}^F$ normal to the hyperplane, pointing in the direction of (most) of the positive examples, and $b \in \mathbb{R}$ specifying the offset of the hyperplane from the origin. Training an SVM constitutes using the training data to find the "optimal" hyperplane described by parameters w, b.

The primal optimization problem. The (soft margin) SVM seeks a hyperplane that separates most of the training data with a large margin defined by the thickness of the region $|w \cdot \phi(G) + b| \leq 1$. The primal optimization problem associated with training an SVM is:

$$\min_{w,b} \left(\frac{1}{2} ||w||^2 + C \sum_{n=1}^{N} \xi_n \right) \tag{14}$$

$$\text{s.t. } \xi_n \geq 0 \text{ for } n \in \{1, ..., N\} \tag{15}$$

$$y_n(w \cdot \phi(G_n) + b) \geq 1 - \xi_n \text{ for } n \in \{1, ..., N\}. \tag{16}$$

The slack variable ξ_n associated with data vector $\phi(G_n)$ allows, if it is nonzero, violation of the constraint $y_n(w \cdot \phi(G_n) + b) \geq 1$ that it lies (i) on the correct side of the decision boundary and (ii) outside of or on the boundary of the margin. The first term in the objective function describes the size of the margin; the second term penalizes constraint violations. The hyperparameter $C \geq 0$ trades a large margin for constraint violations.

The dual optimization problem. The Lagrangian dual of the primal optimization problem is in N Lagrange multipliers $\{\alpha_1, ..., \alpha_N\}$:

$$\max_{\alpha} \left(\sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j \phi(G_i) \cdot \phi(G_j) \right) \tag{17}$$

$$\text{s.t. } 0 \leq \alpha_n \leq C \text{ for } n \in \{1, ..., N\} \tag{18}$$

$$\sum_{n=1}^{N} \alpha_n y_n = 0, \tag{19}$$

where the solution to the dual problem α and the solution to the primal problem w satisfy:

$$w = \sum_{n=1}^{N} \alpha_n y_n \phi(G_n). \tag{20}$$

The kernel trick. The objective of the dual problem in eqn. 17 depends only on the dot products $\phi(G_i) \cdot \phi(G_j)$ of the training data. The kernel trick is to replace $\phi(G_i) \cdot \phi(G_j)$ with a kernel function $k(G_i, G_j) = \phi(G_i) \cdot \phi(G_j)$ to bypass the explicit mapping of the graphs G_i and G_j into the vector space $\mathbb{R}^F$ to compute the dot product $\phi(G_i) \cdot \phi(G_j)$. Indeed, we use the L-RWGK $k^{(L)}(G_i, G_j)$ in eqn. 5 in place of constructing $\phi^{(L)}(G)$ and $\phi^{(L)}(G')$ and taking their dot product.

Using eqn. 20, we can also rewrite the decision rule in eqn. 13 for a new graph G in terms of the kernel between it and the graphs in the training data set:

$$f(G) = \text{sign} \left(\sum_{n=1}^{N} \alpha_n y_n k(G_n, G) + b \right), \tag{21}$$

with α_n the solution to the dual problem. Eqn. 21 allows us to also bypass mapping new molecular graphs G into the feature space $\mathbb{R}^F$ via ϕ when classifying them with the trained SVM.

The support vectors. An SVM is a *sparse* kernel machine [101]; the decision rule in eqn. 21 will depend on a *subset* of the training data, the *support vectors* $\phi^{(L)}(G_n)$ with $\alpha_n > 0$ that lie inside or on the boundary of the margin or outside the margin but on the wrong side of the decision boundary.

The Gram matrix. When in practice invoking the kernel trick, we store the inner products between all pairs of molecular graphs in a $N \times N$ Gram matrix K, whose element (i, j) gives the kernel $k(G_i, G_j)$ between molecular graphs G_i and G_j.

Centering. SVMs tend to perform better if the feature vectors $\{\phi(G_1), ..., \phi(G_N)\}$ are first centered [103]. Again to avoid explicit construction of them, the double-centering trick [84] allows us to obtain the inner products of the centered feature vectors from the inner products of the uncentered feature vectors in the Gram matrix K. Particularly, the centered Gram matrix $\tilde{K} := CKC$ with centering matrix $C = I - \frac{1}{N} oo^\intercal$ (I the identity matrix, o a vector of ones) [104].

3.6 Classification performance metrics

Performance metrics of a classifier $\hat{y} = f(G)$ include, measured over a labeled test data set:

- accuracy: fraction of examples classified correctly.
- precision: among the examples classified as toxic ($\hat{y}_n = 1$), the fraction that are truly toxic ($y_n = 1$)
- recall: among the examples that are truly toxic ($y_n = 1$), the fraction that are correctly predicted as toxic ($\hat{y}_n = 1$).

The F1 score is the harmonic mean of precision and recall. Owing to class imbalance (see Fig. 1b), the F1 score is a better performance metric of the classifier than accuracy [105].

4 Results

We now train and evaluate the performance of a support vector machine (SVM) to classify the toxicity of pesticide molecules to honey bees using two different molecular representations:

- the fixed-length-L random walk feature vector
- the MACCS structural key fingerprint (FP).

We explicitly construct the MACCS FPs, but invoke the kernel trick and rely on the fixed-length-L random walk graph kernel (L-RWGK) in place of a dot product in the random walk feature space.

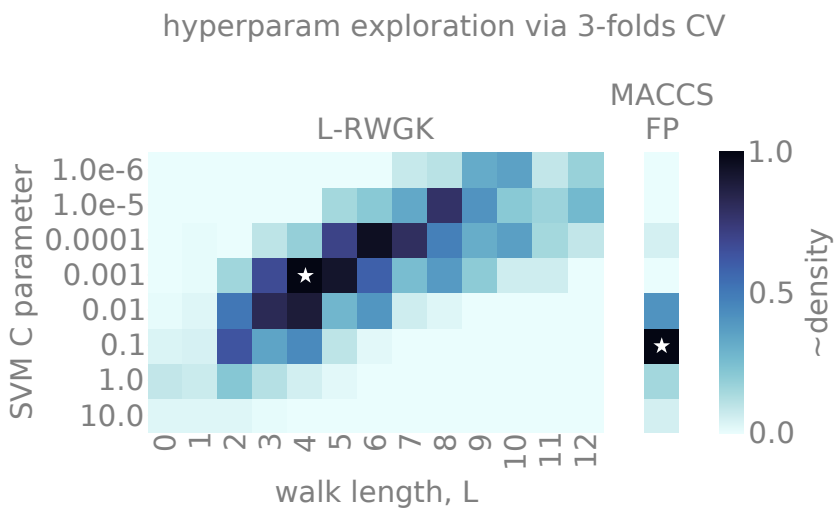
4.1 Machine learning procedures

Data preparation. From the SMILES strings representing the pesticide molecules in the BeeToxAI data set [58]...

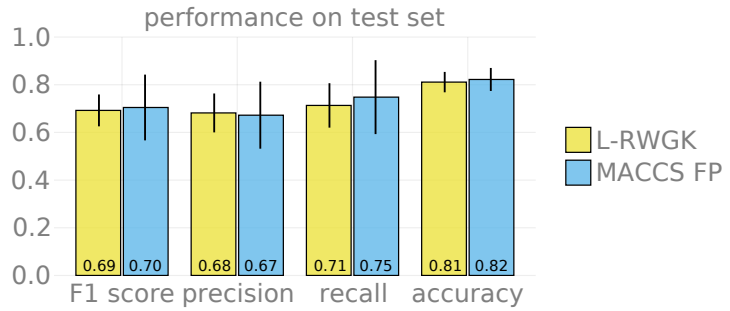
MACCS fingerprints. We used `RDKit` [94] to explicitly construct the MACCS FPs of the pesticide molecules, $\{\phi^{MACCS}(G_1), ..., \phi^{MACCS}(G_N)\}$. Then, we computed the dot product $\phi^{MACCS}(G_i) \cdot \phi^{MACCS}(G_j)$ between each pair of MACCS FPs and stored them in the Gram matrix K^{MACCS}.

Fixed-length random walk graph kernel. We used `MolecularGraph.jl` to obtain the molecular graphs $\{G_1, ..., G_N\}$ representing the pesticide molecules. For each pair of graphs (G_i, G_j), we constructed their direct product graph $G_i \times G_j$, evaluated the L-RWGK $k^{(L)}(G_i, G_j)$ via eqn. 11 (using our own code), then stored it in a Gram matrix $K^{(L)}$, for $L \in \{0, ..., 12\}$.

A train-test run. For both the MACCS FP and fixed-length random walk representations, a "train-test run" of an SVM comprises the following procedure. First, we randomly shuffle then split the examples into a 80%/20% train/test split. We stratify the split to preserve the distribution of class labels in the two splits. Second, using only the training split, we use stratified $K = 3$-fold cross-validation to determine the optimal hyperparameter(s). For the MACCS fingerprint, the hyperparameter is the C parameter of the SVM. For the fixed-length random walk representation, the hyperparameters are both C and L, the length of the random walks. Via grid search, we



(a)



(b)

Figure 5: Average results of the SVM toxicity classifier over 2000 (stochastic) runs of test/train splits using the (i) L-RWGK and (ii) MACCS FP with a linear kernel. (a) The empirical joint distribution of the optimal hyperparameters during the 3-fold cross-validation procedure to determine the optimal SVM C parameter and also, in the case of the L-RWGK, length L of the walks. The $\star$ marks the mode of the optimal hyperparameters. (b) Toxicity classification performance of the deployment SVM (with the optimal C, L from cross-validation) on hold-out test data. Bars show standard deviation.

choose the optimal hyperparameter(s) as the one(s) providing the K SVMs (each trained on $K-1$ folds) with the maximal mean F1 score on the validation sets (one fold each). The hyperparameter grid comprises (i) $\log_{10} C \in \{-6, \ldots, 1\}$ and (ii) $L \in \{0, \ldots, 12\}$. Finally, we train a deployment SVM with the optimal hyperparameter(s) on all training data and evaluate its performance (precision, recall, accuracy, and F1 score) on the hold-out test set.

N.b., for each SVM trained, we center the Gram matrix K pertaining only to the training graphs via the double-centering trick [84]. We adopt a similar centering trick [104] for the Gram matrix giving the similarity of the test graphs with the training graphs when we feed it as input to the SVM for predictions on the test split.

We used the SVC implementation and Gram matrix centerer in scikit-learn [106]. We scaled the C parameter in eqn. 14 seen by the slack variables pertaining to each class to balance penalization of constraint violations for each class.

Overall procedure. For both the MACCS fingerprint and fixed-length random walk representations, we conducted 2000 (stochastic, owing to the random train/test and K-folds splits) train-test runs; for each run, we evaluated the performance of a hyperparameter-optimized, trained SVM classifier on the hold-out test set. Conducting multiple train-test runs allows us to report both expected performance and variance in the performance.

4.2 Cross-validation results

Fig. 5a shows the empirical distribution of optimal hyperparameters during the $K = 3$-folds cross-validation routine. The mode of the distribution of the optimal C parameter for the MACCS-FP-SVM is 0.1. The mode of the joint distribution for the optimal walk length L and SVM C parameter for the L-RWGK-SVM is $L = 4$ and $C = 0.001$. In conclusion, the pesticide molecules were best described by random walks of length $L = 4$ for bee toxicity prediction. The optimal C parameter tended to decrease with the walk length, consistent with the view of the inverse of C as a regularization parameter expected to increase when the representation of the examples is more complex.

4.3 Classification performance on the test set

Fig. 5b shows the mean and standard deviation of the accuracy, precision, recall, and F1 score of the L-RWGK-SVM and MACCS-FP-SVM on the hold-out test set of pesticide molecules. The performance of the L-RWGK-SVM is on par with/slightly lower than that of the MACCS-FP-SVM, but has the advantage of a lower variance (see error bars). The L-RWGK-SVM achieves, on average, an F1 score, precision, recall, and accuracy of 0.69, 0.68, 0.71, and 0.81.

4.4 Interpreting the MACCS-FP-SVM

Explaining the predictions of and interpreting a molecular machine learning model can give chemical insights and foster trust—or distrust, by uncovering "Clever Hans" predictions—in the model [107, 108].

We interpret a MACCS-FP-SVM[6] toxicity classifier by inspecting the vector $w \in \mathbb{R}^{166}$ normal to its separating hyperplane. Weight $w_i \in \mathbb{R}$ in w is associated with MACCS keybit i, which looks for a particular subgraph pattern in the molecular graph. For ease of interpretability, here we do not standardize the input MACCS FPs and instead retain them as bit vectors; consequently, a positive (negative) coefficient w_i implies the presence of the subgraph pattern described by MACCS keybit i tends to produce a prediction of toxicity (non-toxicity).

[6] In contrast to the L-RWGK-SVM that leverages the kernel trick, the MACCS-FP-SVM lends interpretability because we may construct w via eqn. 20.

Fig. 6 visualizes the w vector of our interpretable MACCS-FP-SVM, trained on all of the data and with the optimal C hyperparameter found in Fig. 5a ($C = 0.1$). We inspect the molecular patterns corresponding with the four most positive (top) and four most negative (bottom) coefficients (bars decorated with $*$)—associated with predictions of toxicity and non-toxicity, respectively. Their MACCS keybits and SMARTS strings specifying the molecular pattern they look for are shown. We also show an example molecule in the data set exhibiting that pattern (see highlight); the molecules on the top (bottom) were correctly predicted to be toxic (non-toxic).

We caution against mistaking *association* for *causality* in our interpretation of the SVM, as (i) the two random variables indicating the presence of two subgraphs (described by two MACCS keybits) in a molecule are generally not independent and (ii) anthropogenic biases [109–111] could be involved in the generation and curation of the training data set.

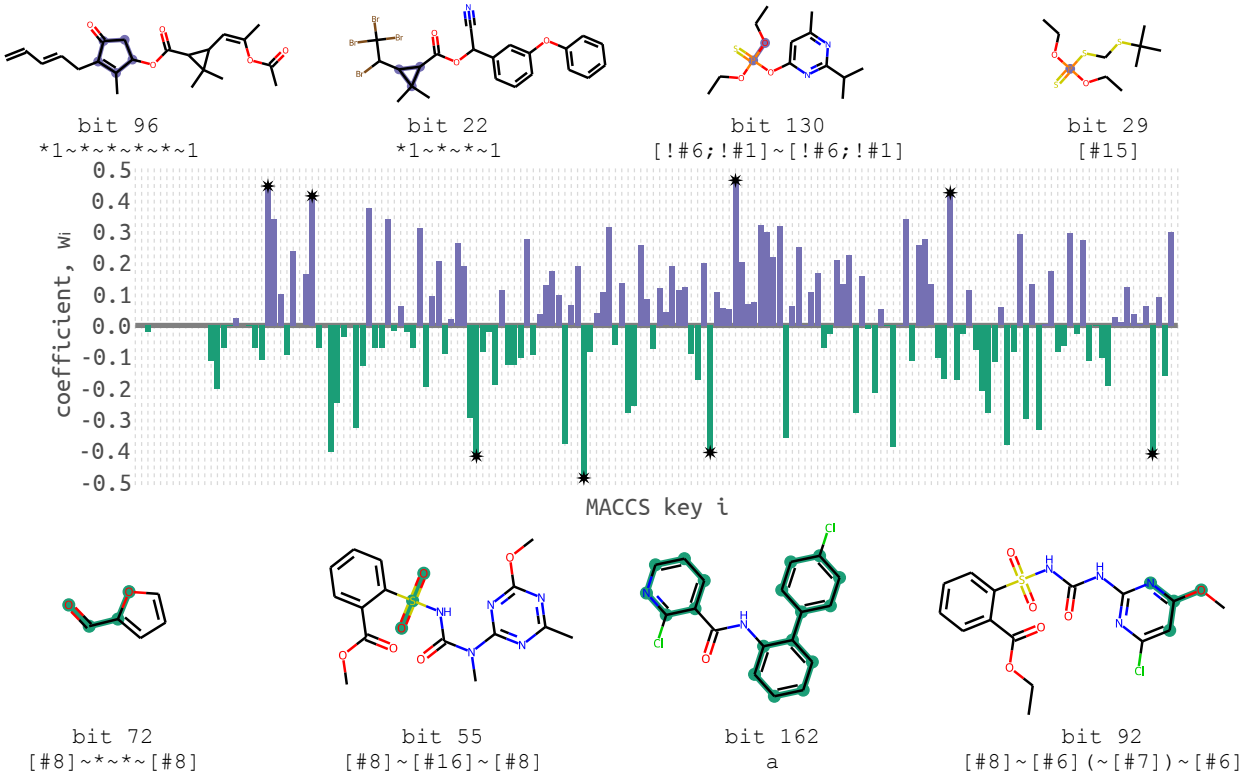


Figure 6: Interperting the SVM toxicity classifier operating on un-standardized MACCS fingerprints. The bar plot visualizes the $w \in \mathbb{R}^{166}$ vector normal to the separating hyperplane and pointing in the direction of the toxic examples (see eqn. 13). If a coefficient w_i of w is positive (negative), the presence of the corresponding molecular subgraph pattern (indicated by MACCS key i) correlates to a prediction of toxicity (non-toxicity). The MACCS keys and examples of molecules in the BeeToxAI data set exhibiting those patterns (highlighted) for the four most positive and four most negative w_i are shown in the top and bottom, respectively (top: all toxic, bottom: all non-toxic).

4.5 Run times.

The majority of the computational run time for generating our results was in computing the 382×382 Gram matrices $K^{(L)}$ involving the L-RWGK. Using four cores, the run time ranged from less than five minutes ($L = 0, 1$) to ~20-25 minutes for $L \geq 7$ (see SI).

5 Discussion

We trained and evaluated a support vector machine classifier that predicts the toxicity of pesticides to honey bees. We compared two molecular vector representations: (1) MACCS fingerprints listing the presence/absence of a set of pre-defined subgraph patterns in the molecular graph and (2) a random walk feature vector listing counts of label sequences along all fixed-length walks on the molecular graph. While we explicitly construct the fingerprints, we relied on the fixed-length random walk graph kernel for dot products in the random walk vector space. The classifier using the MACCS fingerprints (a) gave a slightly higher mean F1 score (0.70 vs. 0.69) than the classifier using the random walk feature, (b) grants a degree of interpretability, but (c) exhibits a higher variance in performance.

Graph kernels have been previously used with SVMs, Gaussian processes, and kernel regression for molecular machine learning tasks [79,112], such as to classify proteins [113], score protein-protein interactions [114], predict methane uptake in nanoporous materials [115], predict atomization energy of molecules [116,117], and predict thermodynamic properties of pure substances [118].

A Gaussian process model [86] using the L-RWGK would enable uncertainty quantification in the prediction.

As the BeeToxAI data set [58] was collected from many sources, including the scientific literature, anthropogenic biases could be present e.g., in the choices of which molecules to test for bee toxicity. As a result, the training and testing data distributions could differ, and the performance measure on the hold-out set (sampled from the training distribution) may not reflect the generalization error when the model is deployed. Ref. [111] articulates various types of this dataset shift. Anthropogenic biases in data sets in chemistry have been uncovered and shown to cause machine learning algorithms trained on them to exhibit poorer generalization performance [109,110].

Disadvantages of the L-RWGK include (i) its compute- and memory-intensity to evaluate, hence poor scalability to large molecules and large data sets [79] and (ii) tottering. Expanding on (ii): by definition, the vertices in a walk (see eqn. 1) may not be distinct (then, it would be a *path*). Thus, long walks that totter back and forth between the same few vertices—e.g., at the extreme: $w = (u, v, u, v, \ldots, u, v)$—are accounted for in the L-RWGK. These walks do not contribute extra information about the similarity of two graphs—e.g., for our extreme example, no more information beyond the length-2 walk (u, v). Tottering thus could lead to a "dilution" of the similarity metric expressed by the random walk kernel. [100] Modification of the random walk kernel can prevent tottering walks [119] from contributing to the similarity metric.

The L-RWGK can be generalized further by defining a kernel between two walks w and w' as a product of the kernel between the edges and vertices along the walk [79,98,113]. A (non-Dirac)

kernel between vertices could account for similarity of chemical elements.

In addition to the fixed-length-L random walk kernel we employed, the (i) max-length-L random walk kernel and (ii) geometric random walk kernel count pairs of length-ℓ walks with a shared label sequence (i) for $\ell \in \{0, ..., L\}$ and (ii) $\ell \in \{0, ...\}$ [98, 100].

In addition to random walk kernels, other graph kernels can be used to express the similarity of molecular graphs [79]: shortest-path [120], graphlet [121], tree- and cyclic-pattern [122, 123], and optimal assignment kernels [124].

Data Availability Statement

The data and code to fully reproduce this study are available at `github.com/SimonEnsemble/graph-kernel-SVM-for-toxicity-of-pesticides-to-bees`.

Acknowledgements

Thanks to Jana Doppa and Aryan Deshwal for stimulating Cory's interest in graph kernels. We acknowledge support from the National Science Foundation, award #1920945.

References

[1] Fernando P Carvalho. Agriculture, pesticides, food security and food safety. *Environ Sci Policy*, 9(7-8):685–692, 2006.

[2] E-C Oerke. Crop losses to pests. *J. Agric. Sci.*, 144(1):31–43, 2006.

[3] József Popp, Károly Pető, and János Nagy. Pesticide productivity and food security. a review. *Agron. Sustain. Dev.*, 33(1):243–255, 2013.

[4] Jerry Cooper and Hans Dobson. The benefits of pesticides to mankind and the environment. *J. Crop Prot.*, 26(9):1337–1348, 2007.

[5] Polyxeni Nicolopoulou-Stamati, Sotirios Maipas, Chrysanthi Kotampasi, Panagiotis Stamatis, and Luc Hens. Chemical pesticides and human health: the urgent need for a new concept in agriculture. *Front. Public Health*, 4:148, 2016.

[6] Clevo Wilson and Clem Tisdell. Why farmers continue to use pesticides despite environmental, health and sustainability costs. *Ecol Econ*, 39(3):449–462, 2001.

[7] Isra Mahmood, Sameen Ruqia Imadi, Kanwal Shazadi, Alvina Gul, and Khalid Rehman Hakeem. Effects of pesticides on environment. In *Plant, Soil and Microbes*, pages 253–269. Springer, 2016.

[8] David Tilman, Joseph Fargione, Brian Wolff, Carla D'antonio, Andrew Dobson, Robert Howarth, David Schindler, William H Schlesinger, Daniel Simberloff, and Deborah

Swackhamer. Forecasting agriculturally driven global environmental change. *science*, 292(5515):281–284, 2001.

[9] Sebastian Stehle and Ralf Schulz. Agricultural insecticides threaten surface waters at the global scale. *Proc. Natl. Acad. Sci. U.S.A.*, 112(18):5750–5755, 2015.

[10] Johnson Stanley and Gnanadhas Preetha. *Pesticide toxicity to non-target organisms*. Springer, 2016.

[11] Lennard W Pisa, Vanessa Amaral-Rogers, Luc P Belzunces, Jean-Marc Bonmatin, Craig A Downs, Dave Goulson, David P Kreutzweiser, Christian Krupke, Matthias Liess, Melanie McField, et al. Effects of neonicotinoids and fipronil on non-target invertebrates. *Environ. Sci. Pollut. Res.*, 22(1):68–102, 2015.

[12] Benjamin P Oldroyd. What's killing american honey bees? *PLoS Biology*, 5(6):e168, 2007.

[13] Ben A Woodcock, JM Bullock, RF Shore, MS Heard, MG Pereira, J Redhead, L Ridding, H Dean, D Sleep, P Henrys, et al. Country-specific effects of neonicotinoid pesticides on honey bees and wild bees. *Science*, 356(6345):1393–1395, 2017.

[14] Adam J Vanbergen and the Insect Pollinators Initiative. Threats to an ecosystem service: pressures on pollinators. *Front. Ecol. Environ.*, 11(5):251–259, 2013.

[15] Richard J Gill, Oscar Ramos-Rodriguez, and Nigel E Raine. Combined pesticide exposure severely affects individual-and colony-level traits in bees. *Nature*, 491(7422):105–108, 2012.

[16] Dave Goulson. An overview of the environmental risks posed by neonicotinoid insecticides. *J Appl Ecol*, 50(4):977–987, 2013.

[17] Dennis VanEngelsdorp, Jerry Hayes Jr, Robyn M Underwood, and Jeffery Pettis. A survey of honey bee colony losses in the us, fall 2007 to spring 2008. *PloS One*, 3(12):e4071, 2008.

[18] Insu Koh, Eric V Lonsdorf, Neal M Williams, Claire Brittain, Rufus Isaacs, Jason Gibbs, and Taylor H Ricketts. Modeling the status, trends, and impacts of wild bee abundance in the united states. *Proc. Natl. Acad. Sci. U.S.A.*, 113(1):140–145, 2016.

[19] Sydney A Cameron, Jeffrey D Lozier, James P Strange, Jonathan B Koch, Nils Cordes, Leellen F Solter, and Terry L Griswold. Patterns of widespread decline in north american bumble bees. *Proc. Natl. Acad. Sci. U.S.A.*, 108(2):662–667, 2011.

[20] Marla Spivak, Eric Mader, Mace Vaughan, and Ned H Euliss Jr. The plight of the bees, 2011.

[21] Dave Goulson, Elizabeth Nicholls, Cristina Botías, and Ellen L Rotheray. Bee declines driven by combined stress from parasites, pesticides, and lack of flowers. *Science*, 347(6229):1255957, 2015.

[22] Nicola Gallai, Jean-Michel Salles, Josef Settele, and Bernard E Vaissière. Economic valuation of the vulnerability of world agriculture confronted with pollinator decline. *Ecol. Econ.*, 68(3):810–821, 2009.

[23] P. H. Raven, G. B. Johnson, J. B. Losos, K. A. Mason, and S. R. Singer. *Biology*. McGraw-Hill higher education, 2008.

[24] Rachael Winfree, Neal M Williams, Hannah Gaines, John S Ascher, and Claire Kremen. Wild bee pollinators provide the majority of crop visitation across land-use gradients in new jersey and pennsylvania, usa. *J Appl Ecol*, 45(3):793–802, 2008.

[25] Alexandra-Maria Klein, Bernard E Vaissiere, James H Cane, Ingolf Steffan-Dewenter, Saul A Cunningham, Claire Kremen, and Teja Tscharntke. Importance of pollinators in changing landscapes for world crops. *Proc. R. Soc. B: Biol. Sci.*, 274(1608):303–313, 2007.

[26] MD Levin. Value of bee pollination to us agriculture. *Am. Entomol.*, 29(4):50–51, 1983.

[27] Keng-Lou James Hung, Jennifer M Kingston, Matthias Albrecht, David A Holway, and Joshua R Kohn. The worldwide importance of honey bees as pollinators in natural habitats. *Proc. R. Soc. B: Biol. Sci.*, 285(1870):20172140, 2018.

[28] Thomas C Sparks and Ralf Nauen. Irac: Mode of action classification and insecticide resistance management. *Pestic Biochem Physiol.*, 121:122–128, 2015.

[29] Micheal DK Owen and Ian A Zelaya. Herbicide-resistant crops and weed resistance to herbicides. *Pest Manag. Sci.*, 61(3):301–311, 2005.

[30] John A. Lucas, Nichola J. Hawkins, and Bart A. Fraaije. Chapter two - the evolution of fungicide resistance. In SIMA SARIASLANI and GEOFFREY MICHAEL GADD, editors, *Advances in Applied Microbiology*, volume 90 of *Advances in Applied Microbiology*, pages 29–92. Academic Press, 2015.

[31] Noriharu Umetsu and Yuichi Shirai. Development of novel pesticides in the 21st century. *J Pestic Sci*, 45(2):54–74, 2020.

[32] Sergio Filipe Sousa, Pedro Alexandrino Fernandes, and Maria Joao Ramos. Protein–ligand docking: current status and future challenges. *Proteins*, 65(1):15–26, 2006.

[33] Fangli Gang, Xiaoting Li, Chaofu Yang, Lijuan Han, Hao Qian, Shaopeng Wei, Wenjun Wu, and Jiwen Zhang. Synthesis and insecticidal activity evaluation of virtually screened phenylsulfonamides. *J. Agric. Food Chem*, 68(42):11665–11671, 2020.

[34] Toshiyuki Harada, Yoshiaki Nakagawa, Takehiko Ogura, Yutaka Yamada, Takehiro Ohe, and Hisashi Miyagawa. Virtual screening for ligands of the insect molting hormone receptor. *J. Chem. Inf. Model*, 51(2):296–305, 2011.

[35] Xueping Hu, Bin Yin, Kaat Cappelle, Luc Swevers, Guy Smagghe, Xinling Yang, and Li Zhang. Identification of novel agonists and antagonists of the ecdysone receptor by virtual screening. *J. Mol. Graph. Model.*, 81:77–85, 2018.

[36] Ting-Ting Yao, Shao-Wei Fang, Zhong-Shan Li, Dou-Xin Xiao, Jing-Li Cheng, Hua-Zhou Ying, Yong-Jun Du, Jin-Hao Zhao, and Xiao-Wu Dong. Discovery of novel succinate dehydrogenase inhibitors by the integration of in silico library design and pharmacophore mapping. *J. Agric. Food Chem*, 65(15):3204–3211, 2017.

[37] Shinri Horoiwa, Taiyo Yokoi, Satoru Masumoto, Saki Minami, Chiharu Ishizuka, Hidetoshi Kishikawa, Shunsuke Ozaki, Shigeki Kitsuda, Yoshiaki Nakagawa, and Hisashi Miyagawa. Structure-based virtual screening for insect ecdysone receptor ligands using MM/PBSA. *Bioorg. Med. Chem.*, 27(6):1065–1075, 2019.

[38] Galen J Correy, Daniel Zaidman, Alon Harmelin, Silvia Carvalho, Peter D Mabbitt, Viviane Calaora, Peter J James, Andrew C Kotze, Colin J Jackson, and Nir London. Overcoming insecticide resistance through computational inhibitor design. *Proc. Natl. Acad. Sci. U.S.A.*, 116(42):21012–21021, 2019.

[39] Kerem Teralı. An evaluation of neonicotinoids' potential to inhibit human cholinesterases: Protein–ligand docking and interaction profiling studies. *J. Mol. Graph. Model.*, 84:54–63, 2018.

[40] Axel Decourtye, Mickaël Henry, and Nicolas Desneux. Overhaul pesticide testing on bees. *Nature*, 497(7448):188–188, 2013.

[41] United States Environmental Protection Agency. Pollinator risk assessment guidance. `https://www.epa.gov/pollinator-protection/pollinator-risk-assessment-guidance`. Accessed: 2022-02-20.

[42] Glenn J. Myatt, Ernst Ahlberg, Yumi Akahori, David Allen, Alexander Amberg, Lennart T. Anger, Aynur Aptula, Scott Auerbach, Lisa Beilke, Phillip Bellion, Romualdo Benigni, Joel Bercu, Ewan D. Booth, Dave Bower, Alessandro Brigo, Natalie Burden, Zoryana Cammerer, Mark T.D. Cronin, Kevin P. Cross, Laura Custer, Magdalena Dettwiler, Krista Dobo, Kevin A. Ford, Marie C. Fortin, Samantha E. Gad-McDonald, Nichola Gellatly, Véronique Gervais, Kyle P. Glover, Susanne Glowienke, Jacky Van Gompel, Steve Gutsell, Barry Hardy, James S. Harvey, Jedd Hillegass, Masamitsu Honma, Jui-Hua Hsieh, Chia-Wen Hsu, Kathy Hughes, Candice Johnson, Robert Jolly, David Jones, Ray Kemper, Michelle O. Kenyon, Marlene T. Kim, Naomi L. Kruhlak, Sunil A. Kulkarni, Klaus Kümmerer, Penny Leavitt, Bernhard Majer, Scott Masten, Scott Miller, Janet Moser, Moiz Mumtaz, Wolfgang Muster, Louise Neilson, Tudor I. Oprea, Grace Patlewicz, Alexandre Paulino, Elena Lo Piparo, Mark Powley, Donald P. Quigley, M. Vijayaraj Reddy, Andrea-Nicole Richarz, Patricia Ruiz, Benoit Schilter, Rositsa Serafimova, Wendy Simpson, Lidiya Stavitskaya, Reinhard Stidl, Diana Suarez-Rodriguez, David T. Szabo, Andrew Teasdale, Alejandra Trejo-Martin, Jean-Pierre Valentin, Anna Vuorinen, Brian A. Wall, Pete Watts, Angela T. White, Joerg Wichard, Kristine L. Witt, Adam Woolley, David Woolley, Craig Zwickl, and Catrin Hasselgren. In silico toxicology protocols. *Regul. Toxicol. Pharmacol.*, 96:1–17, 2018.

[43] Arwa B Raies and Vladimir B Bajic. In silico toxicology: computational methods for the prediction of chemical toxicity. *Wiley Interdiscip. Rev. Comput. Mol. Sci.*, 6(2):147–172, 2016.

[44] Flor A Quintero, Suhani J Patel, Felipe Munoz, and M Sam Mannan. Review of existing qsar/qspr models developed for properties used in hazardous chemicals classification system. *Ind. Eng. Chem. Res.*, 51(49):16101–16115, 2012.

[45] Takao Iwasa, Naoki Motoyama, John T Ambrose, and R Michael Roe. Mechanism for the differential toxicity of neonicotinoid insecticides in the honey bee, apis mellifera. *J. Crop Prot.*, 23(5):371–378, 2004.

[46] Daniela Laurino, Marco Porporato, Augusto Patetta, Aulo Manino, et al. Toxicity of neonicotinoid insecticides to honey bees: laboratory tests. *Bull. Insectology*, 64(1):107–13, 2011.

[47] Francisco Sanchez-Bayo and Koichi Goka. Pesticide residues and bees–a risk assessment. *PLOS One*, 9(4):e94482, 2014.

[48] EL Atkins and D Kellum. Comparative morphogenic and toxicity studies on the effect of pesticides on honeybee brood. *J. Apic. Res.*, 25(4):242–255, 1986.

[49] Helen M Thompson. Assessing the exposure and toxicity of pesticides to bumblebees (bombus sp.). *Apidologie*, 32(4):305–321, 2001.

[50] Yee-Tung Hu, Tsai-Chin Wu, En-Cheng Yang, Pei-Chi Wu, Po-Tse Lin, and Yueh-Lung Wu. Regulation of genes related to immune signaling and detoxification in apis mellifera by an inhibitor of histone deacetylation. *Sci. Rep.*, 7(1):1–14, 2017.

[51] Krystyna Pohorecka, Teresa Szczęsna, Monika Witek, Artur Miszczak, and Piotr Sikorski. The exposure of honey bees to pesticide residues in the hive environment with regard to winter colony losses. *J Apic Sci*, 61(1):105–125, 2017.

[52] Christopher A Mullin, Maryann Frazier, James L Frazier, Sara Ashcraft, Roger Simonds, Dennis VanEngelsdorp, and Jeffery S Pettis. High levels of miticides and agrochemicals in north american apiaries: implications for honey bee health. *PLOS one*, 5(3):e9754, 2010.

[53] A Decourtye, James Devillers, Evelyne Genecque, K Le Menach, H Budzinski, S Cluzeau, and MH Pham-Delègue. Comparative sublethal toxicity of nine pesticides on olfactory learning performances of the honeybee apis mellifera. *Arch. Environ. Contam. Toxicol.*, 48(2):242–250, 2005.

[54] Thaís S Bovi, Rodrigo Zaluski, and Ricardo O Orsi. Toxicity and motor changes in africanized honey bees (apis mellifera l.) exposed to fipronil and imidacloprid. *An. Acad. Bras. Cienc.*, 90:239–245, 2018.

[55] Mohamed El Badawy, Hoda M Nasr, and Entsar I Rabea. Toxicity and biochemical changes in the honey bee apis mellifera exposed to four insecticides under laboratory conditions. *Apidologie*, 46(2):177–193, 2015.

[56] Nadejda Tsvetkov, Olivier Samson-Robert, Keshna Sood, HS Patel, DA Malena, PH Gajiwala, Philip Maciukiewicz, Valerie Fournier, and Amro Zayed. Chronic exposure to neonicotinoids reduces honey bee health near corn crops. *Science*, 356(6345):1395–1397, 2017.

[57] Mohamed E. I. Badawy, Hoda M. Nasr, and Entsar I. Rabea. Toxicity and biochemical changes in the honey bee apis mellifera exposed to four insecticides under laboratory conditions. *Apidologie*, 46(2):177–193, September 2014.

[58] José T. Moreira-Filho, Rodolpho C. Braga, Jade Milhomem Lemos, Vinicius M. Alves, Joyce V.V.B. Borba, Wesley S. Costa, Nicole Kleinstreuer, Eugene N. Muratov, Carolina Horta Andrade, and Bruno J. Neves. Beetoxai: An artificial intelligence-based web app to assess acute toxicity of chemicals to honey bees. *Artif. Intell. Life Sci.*, 1:100013, 2021.

[59] Fan Wang, Jing-Fang Yang, Meng-Yao Wang, Chen-Yang Jia, Xing-Xing Shi, Ge-Fei Hao, and Guang-Fu Yang. Graph attention convolutional neural network model for chemical poisoning of honey bees prediction. *Sci. Bull.*, 65(14):1184–1191, 2020.

[60] Edoardo Carnesecchi, Andrey A. Toropov, Alla P. Toropova, Nynke Kramer, Claus Svendsen, Jean Lou Dorne, and Emilio Benfenati. Predicting acute contact toxicity of organic binary mixtures in honey bees (a. mellifera) through innovative qsar models. *Sci. Total Environ.*, 704:135302, 2020.

[61] Mabrouk Hamadache, Othmane Benkortbi, Salah Hanini, and Abdeltif Amrane. QSAR modeling in ecotoxicological risk assessment: application to the prediction of acute contact toxicity of pesticides on bees (apis mellifera l.). *Environ. Sci. Pollut. Res.*, 25(1):896–907, October 2017.

[62] F. Como, E. Carnesecchi, S. Volani, J.L. Dorne, J. Richardson, A. Bassan, M. Pavan, and E. Benfenati. Predicting acute contact toxicity of pesticides in honeybees (apis mellifera) through a k-nearest neighbor model. *Chemosphere*, 166:438–444, 2017.

[63] Xuan Xu, Piaopiao Zhao, Zhiyuan Wang, Xiaoxiao Zhang, Zengrui Wu, Weihua Li, Yun Tang, and Guixia Liu. In silico prediction of chemical acute contact toxicity on honey bees via machine learning methods. *Toxicol In Vitro*, 72:105089, 2021.

[64] Xiao Li, Yuan Zhang, Hongna Chen, Huanhuan Li, and Yong Zhao. Insights into the molecular basis of the acute contact toxicity of diverse organic chemicals in the honey bee. *J. Chem. Inf. Model*, 57(12):2948–2957, 2017.

[65] Keith T Butler, Daniel W Davies, Hugh Cartwright, Olexandr Isayev, and Aron Walsh. Machine learning for molecular and materials science. *Nature*, 559(7715):547–555, 2018.

[66] Laurianne David, Amol Thakkar, Rocío Mercado, and Ola Engkvist. Molecular representations in AI-driven drug discovery: a review and practical guide. *J. Cheminformatics*, 12(1):1–22, 2020.

[67] Daniel S Wigh, Jonathan M Goodman, and Alexei A Lapkin. A review of molecular representation in the age of machine learning. *Wiley Interdiscip. Rev. Comput. Mol. Sci.*, page e1603, 2022.

[68] Lagnajit Pattanaik and Connor W Coley. Molecular representation: going long on fingerprints. *Chem*, 6(6):1204–1207, 2020.

[69] Lagnajit Pattanaik, Octavian-Eugen Ganea, Ian Coley, Klavs F Jensen, William H Green, and Connor W Coley. Message passing networks for molecules with tetrahedral chirality. *arXiv preprint arXiv:2012.00094*, 2020.

[70] Tu Le, V Chandana Epa, Frank R Burden, and David A Winkler. Quantitative structure–property relationship modeling of diverse materials properties. *Chem. Rev.*, 112(5):2889–2919, 2012.

[71] Simon Axelrod, Daniel Schwalbe-Koda, Somesh Mohapatra, James Damewood, Kevin P Greenman, and Rafael Gómez-Bombarelli. Learning matter: Materials design with machine learning and atomistic simulations. *Acc. Mater. Res.*, 2022.

[72] Adrià Cereto-Massagué, María José Ojeda, Cristina Valls, Miquel Mulero, Santiago Garcia-Vallvé, and Gerard Pujadas. Molecular fingerprint similarity search in virtual screening. *Methods*, 71:58–63, 2015.

[73] Steven Kearnes, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley. Molecular graph convolutions: moving beyond fingerprints. *J. Comput. Aided Mol. Des.*, 30(8):595–608, 2016.

[74] Joseph L Durant, Burton A Leland, Douglas R Henry, and James G Nourse. Reoptimization of MDL keys for use in drug discovery. *J Chem Inf Comput Sci*, 42(6):1273–1280, 2002.

[75] William L Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*, 2017.

[76] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

[77] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.*, 32(1):4–24, 2020.

[78] S Vichy N Vishwanathan, Nicol N Schraudolph, Risi Kondor, and Karsten M Borgwardt. Graph kernels. *J Mach Learn Res*, 11:1201–1242, 2010.

[79] Matthias Rupp and Gisbert Schneider. Graph kernels for molecular similarity. *Mol. Inform.*, 29(4):266–273, 2010.

[80] Karsten Borgwardt, Elisabetta Ghisu, Felipe Llinares-López, Leslie O'Bray, and Bastian Rieck. Graph kernels: State-of-the-art and future challenges. *arXiv preprint arXiv:2011.03854*, 2020.

[81] Jan Ramon and Thomas Gärtner. Expressivity versus efficiency of graph kernels. In *Proceedings of the first international workshop on mining graphs, trees and sequences*, pages 65–74, 2003.

[82] Liva Ralaivola, Sanjay J. Swamidass, Hiroto Saigo, and Pierre Baldi. Graph kernels for chemical informatics. *Neural Netw*, 18(8):1093–1110, 2005. Neural Networks and Kernel Methods for Structured Domains.

[83] Giannis Nikolentzos, Giannis Siglidis, and Michalis Vazirgiannis. Graph kernels: A survey. *J Artif Intell Res*, 72:943–1027, 2021.

[84] Kevin P. Murphy. *Probabilistic Machine Learning: An introduction*. MIT Press, 2022.

[85] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Mach. Learn.*, 20(3):273–297, 1995.

[86] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, 2006.

[87] Simon S Du, Kangcheng Hou, Russ R Salakhutdinov, Barnabas Poczos, Ruosong Wang, and Keyulu Xu. Graph neural tangent kernel: Fusing graph neural networks with graph kernels. *Advances in neural information processing systems*, 32, 2019.

[88] Yan Xiang, Yu-Hang Tang, Guang Lin, and Huai Sun. A comparative study of marginalized graph kernel and message-passing neural network. *J. Chem. Inf. Model*, 61(11):5414–5424, 2021.

[89] Office of Pesticide Programs; United States Environmental Protection Agency. Guidance for assessing pesticide risks to bees. `https://www.epa.gov/sites/default/files/2014-06/documents/pollinator_risk_assessment_guidance_06_19_14.pdf`, 2014. Accessed: 2022-05-03.

[90] Zimo Yin, Haixin Ai, Li Zhang, Guofei Ren, Yuming Wang, Qi Zhao, and Hongsheng Liu. Predicting the cytotoxicity of chemicals using ensemble learning methods and molecular fingerprints. *J. Appl. Toxicol.*, 39(10):1366–1377, 2019.

[91] Xiao Li, Lei Chen, Feixiong Cheng, Zengrui Wu, Hanping Bian, Congying Xu, Weihua Li, Guixia Liu, Xu Shen, and Yun Tang. In silico prediction of chemical acute oral toxicity using multi-classification methods. *J. Chem. Inf. Model*, 54(4):1061–1069, 2014.

[92] Dong-Sheng Cao, Yan-Ning Yang, Jian-Chao Zhao, Jun Yan, Shao Liu, Qian-Nan Hu, Qing-Song Xu, and Yi-Zend Liang. Computer-aided prediction of toxicity with substructure pattern and random forest. *J. Chemom.*, 26(1-2):7–15, 2012.

[93] Chen Zhang, Feixiong Cheng, Weihua Li, Guixia Liu, Philip W Lee, and Yun Tang. In silico prediction of drug induced liver toxicity using substructure pattern recognition method. *Mol. Inform.*, 35(3-4):136–144, 2016.

[94] Greg Landrum et al. RDKit: A software suite for cheminformatics, computational chemistry, and predictive modeling, 2013.

[95] RDKit. RDKit source code for MACCS fingerprint. `https://github.com/rdkit/rdkit/blob/master/rdkit/Chem/MACCSkeys.py`. Accessed: 2022-05-23.

[96] Douglas J Klein, José Luis Palacios, Milan Randić, and Nenad Trinajstić. Random walks and chemical graph theory. *J Chem Inf Comput Sci*, 44(5):1521–1525, 2004.

[97] Thomas Gärtner, Peter Flach, and Stefan Wrobel. On graph kernels: Hardness results and efficient alternatives. In *Learning theory and kernel machines*, pages 129–143. Springer, 2003.

[98] Nils M Kriege, Marion Neumann, Christopher Morris, Kristian Kersting, and Petra Mutzel. A unifying view of explicit and implicit feature maps of graph kernels. *Data Min Knowl Discov*, 33(6):1505–1547, 2019.

[99] Hisashi Kashima, Koji Tsuda, and Akihiro Inokuchi. Marginalized kernels between labeled graphs. In *Proceedings of the 20th international conference on machine learning (ICML-03)*, pages 321–328, 2003.

[100] Nils M Kriege, Fredrik D Johansson, and Christopher Morris. A survey on graph kernels. *Appl. Netw. Sci.*, 5(1):1–42, 2020.

[101] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

[102] Asa Ben-Hur and Jason Weston. A users guide to support vector machines. In *Data mining techniques for the life sciences*, pages 223–239. Springer, 2010.

[103] Marina Meilă. Data centering in feature space. In *International Workshop on Artificial Intelligence and Statistics*, pages 209–216. PMLR, 2003.

[104] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.*, 10(5):1299–1319, 1998.

[105] Takaya Saito and Marc Rehmsmeier. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PloS One*, 10(3):e0118432, 2015.

[106] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *J Mach Learn Res*, 12:2825–2830, 2011.

[107] Wojciech Samek and Klaus-Robert Müller. Towards explainable artificial intelligence. In *Explainable AI: interpreting, explaining and visualizing deep learning*, pages 5–22. Springer, 2019.

[108] Geemi P Wellawatte, Aditi Seshadri, and Andrew D White. Model agnostic generation of counterfactual explanations for molecules. *Chemical Science*, 13(13):3697–3705, 2022.

[109] Xiwen Jia, Allyson Lynch, Yuheng Huang, Matthew Danielson, Immaculate Langat, Alexander Milder, Aaron E Ruby, Hao Wang, Sorelle A Friedler, Alexander J Norquist, et al. Anthropogenic biases in chemical reaction data hinder exploratory inorganic synthesis. *Nature*, 573(7773):251–255, 2019.

[110] Dávid Péter Kovács, William McCorkindale, and Alpha A Lee. Quantitative interpretation explains machine learning models for chemical reaction prediction and uncovers bias. *Nat. Commun.*, 12(1):1–9, 2021.

[111] Amos Storkey. When training and test sets are different: characterizing learning transfer. *Dataset shift in machine learning*, 30:3–28, 2009.

[112] MP Preeja and KP Soman. Walk-based graph kernel for drug discovery: A review. *Int. J. Comput. Appl.*, 94:1–7, 2014.

[113] Karsten M Borgwardt, Cheng Soon Ong, Stefan Schönauer, SVN Vishwanathan, Alex J Smola, and Hans-Peter Kriegel. Protein function prediction via graph kernels. *Bioinformatics*, 21(suppl_1):i47–i56, 2005.

[114] Cunliang Geng, Yong Jung, Nicolas Renaud, Vasant Honavar, Alexandre M J J Bonvin, and Li C Xue. iScore: a novel graph kernel-based function for scoring protein-protein docking models. *Bioinformatics*, 36(1):112–121, 06 2019.

[115] Hiroshi Ohno and Yusuke Mukae. Machine learning approach for prediction and search: application to methane storage in a metal–organic framework. *J. Phys. Chem. C*, 120(42):23963–23968, 2016.

[116] Yu-Hang Tang and Wibe A de Jong. Prediction of atomization energy using graph kernel and active learning. *J. Chem. Phys*, 150(4):044107, 2019.

[117] Grégoire Ferré, Terry Haut, and Kipton Barros. Learning molecular energies using localized graph kernels. *J. Chem. Phys*, 146(11):114107, 2017.

[118] Yan Xiang, Yu-Hang Tang, Hongyi Liu, Guang Lin, and Huai Sun. Predicting single-substance phase diagrams: A kernel approach on graph representations of molecules. *J. Phys. Chem. A*, 125(20):4488–4497, 2021.

[119] Pierre Mahé, Nobuhisa Ueda, Tatsuya Akutsu, Jean-Luc Perret, and Jean-Philippe Vert. Extensions of marginalized graph kernels. In *Proceedings of the twenty-first international conference on Machine learning*, page 70, 2004.

[120] Karsten M Borgwardt and Hans-Peter Kriegel. Shortest-path kernels on graphs. In *Fifth IEEE international conference on data mining (ICDM'05)*, pages 8–pp. IEEE, 2005.

[121] Nino Shervashidze, SVN Vishwanathan, Tobias Petri, Kurt Mehlhorn, and Karsten Borgwardt. Efficient graphlet kernels for large graph comparison. In *Artificial intelligence and statistics*, pages 488–495. PMLR, 2009.

[122] Tamás Horváth, Thomas Gärtner, and Stefan Wrobel. Cyclic pattern kernels for predictive graph mining. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 158–167, 2004.

[123] Pierre Mahé and Jean-Philippe Vert. Graph kernels based on tree patterns for molecules. *Mach. Learn.*, 75(1):3–35, 2009.

[124] Holger Fröhlich, Jörg K Wegner, Florian Sieker, and Andreas Zell. Optimal assignment kernels for attributed molecular graphs. In *Proceedings of the 22nd international conference on Machine learning*, pages 225–232, 2005.