Predicting Redox Potentials by Graph-Based Machine Learning Methods

Linlin Jia[*]
The PRG Group, Institute of Computer Science, University of Bern

Éric Brémond[†]
Université Paris Cité, ITODYS, CNRS, F-75006 Paris, France

Larissa Zaida
Université Paris Cité, ITODYS, CNRS, F-75013 Paris, France

Benoit Gaüzere
*Univ Rouen Normandie, INSA Rouen Normandie, Université Le Havre Normandie,
Normandie Univ, LITIS UR 4108, 76000, Rouen, France*

Vincent Tognetti[‡] and Laurent Joubert
*Normandy Univ., COBRA UMR 6014 & FR 3038, Université de Rouen,
INSA Rouen, CNRS, 1 rue Tesnière 76821 Mont St Aignan Cedex, France*

The evaluation of oxidation and reduction potentials is a pivotal task in various chemical fields. However, their accurate prediction by theoretical computations, which is a complementary task and sometimes the only alternative to experimental measurement, may be often resource-intensive and time-consuming. This paper addresses this challenge through the application of machine learning techniques, with a particular focus on graph-based methods (such as graph edit distances, graph kernels, and graph neural networks) that are reviewed to enlighten their deep links with theoretical chemistry. To this aim, we establish the ORedOx159 database, a comprehensive, homogeneous (with reference values stemming from density functional theory calculations), and reliable resource containing 318 one-electron reduction and oxidation reactions and featuring 159 large organic compounds. Subsequently, we provide an instructive overview of the good practice in machine learning and of commonly utilized machine learning models. We then assess their predictive performances on the ORedOx159 dataset through extensive analyses. Our simulations using descriptors that are computed in an almost instantaneous way result in a notable improvement in prediction accuracy, with mean absolute error (MAE) values equal to 5.6 kcal mol^{-1} for reduction and 7.2 kcal mol^{-1} for oxidation potentials, which paves a way toward efficient *in silico* design of new electrochemical systems.

I. INTRODUCTION

The experimental optimization of chemical reagents is very often a time-consuming and financially expensive task as it requires numerous tries that can also involve hazardous compounds or complex synthetic strategies. As a consequence, the exploration of the chemical space for a given property frequently remains limited to a small number of variations, and the fine-tuning that is performed is then far from being optimal. It is thus highly desirable to have at disposal a fast screening tool that can efficiently guide the applied chemists for the selection of the best synthetic targets.

Numerical techniques are certainly suitable candidates for shortcut strategies that can be led at larger scales, provided the associated computations can treat the systems both in a reasonable time and with a sufficient accuracy. While this last point can be achieved using advanced quantum chemistry (QC) methods such as density functional theory (DFT) or post-Hartree-Fock methods, the first one precludes the use of these latter approaches for an extensive compound search since calculations for a single molecule can take several hours or days in the case of extended systems.

Such limitations can be alleviated by the use of machine learning (ML) methods, which usually provide predictive models that can be deployed at a large scale (then enhancing the exploration of the chemical space [1]) in a rather small amount of time when they are based on features than can be evaluated in a faster way than the related full QC calculation. The importance of ML in theoretical chemistry has tremendously increased in the last decade, and has now become ubiquitous in the field (see for instance the reviews by Keith *et al.* [2] and Mater *et al.* [3]), generating also a lot of individual tools and software [4]. Such booming is so exponential that it has become almost impossible to review the use of ML in chemistry in a comprehensive way, since it ranges from drug design to retro-synthesis analysis, encompassing material sciences and catalysis, or molecular dynamics and even

[*] A major part of this work was done when the author was at COBRA lab, France.; linlin.jia@unibe.ch

[†] eric.bremond@u-paris.fr

[‡] vincent.tognetti@univ-rouen.fr

QC itself, among other fields of application.

In this paper intended for the special issue of the Journal of Computational Chemistry devoted to ML and artificial intelligence in theoretical and computational Chemistry, we will illustrate in a didactic fashion to what extent the ML machinery can be efficiently implemented for the prediction of one of the most fundamental chemical properties, namely the redox potentials. Indeed, oxidation and reduction are elementary processes that occur in almost all chemical subfields [5]. This is obviously the case in biochemistry since many biological processes involve redox reactions (as epitomized by the cellular respiration [6]), but also in geology (formation of blast furnaces [7]), in chemical engineering and industry (we can mention cathodic protection in galvanized steels to fight against corrosion [8]), in new product synthesis through electrochemical catalysis [9], and so on.

Being able to accurately predict redox properties, both for known or new molecules, is thus of utmost importance in many areas of chemistry and also in daily life (if one thinks of batteries and solar cells for instance), and ML methods have already been applied to tackle this issue in the spirit of Quantitative Structure–Property Relationships (QSPR) approaches. A very recent publication by Fedorov and Gryn'ova reviewed in detail this topic [10], and we thus refer the interested reader to this remarkable paper for an up-to-date account on ML technics for redox prediction and an expanded panorama of reported models. We will restrict ourselves to only mention here few landmark papers (the choice being of course arbitrarily and too much reduced), covering various fields (from material science to biochemistry) by Kleinová *et al.* [11], Méndez-Hernández *et al.* [12], Ghule *et al.* [13], Galuzzi *et al.* [14], or Bhat *et al.* [15].

Our aim, here, is actually much less ambitious in this paper that adopts a hybrid format since it is both a short review and a research paper with new results. Hence, we will not look for a general and versatile (and even less universal) model, but, conversely, we will pick out some specific tools - mainly belonging to graph-based methods -, maybe less known in the chemical community, and we will discuss their relevant for the prediction of redox potentials. Graph theory has actually be efficiently used in chemistry for the prediction of many properties belonging to different fields of chemistry, ranging from mutagenicity or toxicity in medicinal chemistry to boiling points in physical chemistry (see for instance [16, 17]). Noteworthy, we will start from the very beginning by setting up from scratch a completely new database of organic molecules, with DFT reference values computed by ourselves.

This departs from the common practice based on preexisting databases. Indeed, one of the usual drawbacks of them is that they might gather reference values from various origins, sometimes without clear source, and they may thus suffer from a lack of homogeneity that is not without incidence on the "trustability" of the results. Conversely, we will generate here our reference values with a unique and perfectly defined, controlled, and - also an important point - reproducible computational protocol. This database will be described more in details in the next section.

Then, the computational details and the ML technics used in this study will be presented in a pedagogical way, so that it can build a bridge between the two communities (namely theoretical chemistry and data science). A particular emphasis will be put on graph-based methods since they are at the heart of our original ML method. Then, the various ML models obtained will be presented and discussed before final conclusions.

From a methodological point of view, any ML study should adhere to the FAIR (Findable, Accessible, Interoperable, Reusable) principles [18]. Artrith *et al.* have recently reviewed best practices in ML for chemistry [19] that serve as useful guidelines for our purposes. Their ckecklist consists in six main points: *(i)* data sources should be listed, publicly available with a clear identification numbers, and with possible biases reported; *(ii)* data should be cleaned using a well-defined and discussed protocol; *(iii)* the methods for data representations should be clearly articulated and compared with the literature; *(iv)* the implementation of the ML model should be provided, and the model should be compared to baseline methods and to state-of-the-art ones; *(v)* a clear data split between training, validation, and testing should be implemented and clearly described; *(vi)* the code and workflow should be made available and allow for reproducing the reported results. We will illustrate all these general rules throughout this paper.

II. THE OREDOX159 DATABASE

The ORedOx159 database is a collection of 318 one-electron reduction and oxidation reactions involving 159 large-size organic compounds routinely used as redox indicators, or involved in the development of molecular electrochemical storage and electrochemical sensors [20]. For instance, it counts viologen derivatives which are well-known from decades to reversibly change color between violet and deep blue through reduction and oxidation [21], or 2,2'-bipyridiniums which are used as 'electron reservoir' for electrochemical storage [22], or phenothiazine compounds which has been recently considered as efficient redox mediators in electrochemical sensors [23]. For each organic compound **A**, the database collects a one-electron reduction reaction

$$\mathbf{A} + \mathrm{e}^- \to \mathbf{A}^-, \qquad \Delta_r \mathrm{G}^0_{\mathrm{Red}} \tag{1}$$

where $\Delta_r \mathrm{G}^0_{\mathrm{Red}}$ denotes the (standard) Gibbs free energy variation of the reduction process at room temperature. $\Delta_r \mathrm{G}^0_{\mathrm{Red}}$ values are usually computed as negative except when the oxidant **A** is less stable than the reducer $\mathbf{A}^-$. This issue is related to that of negative electron affinities (in general defined for vertical processes) and can also be an artefact due to the neglecting of solvent effects (see for instance the discussion by De Proft and co-workers within a DFT context [24].

Complementarily, the oxidation reaction twin writes

$$\mathbf{A} \rightarrow \mathbf{A}^{+} + e^{-}, \qquad \Delta_r G^0_{\mathrm{Ox}} \tag{2}$$

with $\Delta_r G^0_{\mathrm{Ox}}$ being the Gibbs free energy variation during the oxidation process at the same temperature.

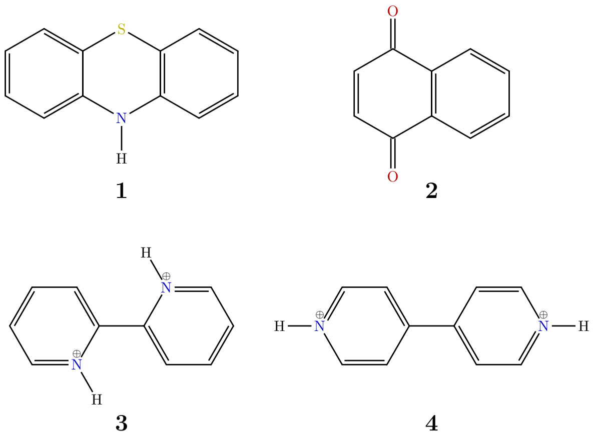

FIG. 1: Scheme of the main organic derivatives from which the compounds of the ORedOx159 database are derived.

Let us recall that the Gibbs free energy is directly related to the standard reduction potential E^0_{Red} encountered in electrochemistry. In the general case, an oxidizer is indeed reduced by z electrons to form a reduced species. These two quantities are linked by the well-known Nernst equation

$$\Delta_r G^0_{\mathrm{Red}} = -z\mathcal{F}E^0_{\mathrm{Red}}, \tag{3}$$

where $\mathcal{F}$ is the Faraday constant (*i.e.*, 96,500 C mol $^{-1}$). It is thus equivalent to build a predictive model on Gibbs reaction free energies or on redox potentials.

Regarding its chemical diversity, the database is composed by a large variety of compounds substituted by electro donor or acceptor groups such as aliphatic, etheroxy, keto, ester and halogen substituents, and derived from four different organic families that are clustered into the phenothiazine (**1**), quinone (**2**), 2,2'-bipyridinium (**3**) and viologen (**4**) subsets. Figure 1 provides a representation of the main core derivative of each of them, and Figure 2 depicts the distribution of their respective Gibbs free energy variations in reduction and oxidation computed in gas phase at PBE0/def2-SVP level.

More precisely, the phenothiazine subset counts 28 derivatives. They are 15 carbon- and 6 aza-substitued phenothiazines as well as 4 benzothiazines and 3 sulfoxided phenothiazines. Their reduction Gibbs free energy variations span between -26.31 and 10.07 kcal mol^{-1} while their oxidation ones varies between 140.00 and 181.55 kcal mol^{-1}.

The quinone subset composes of 37 compounds. They belong to naphto- and anthra-quinones (13 and 3 derivatives, respectively) as well as isoindole-4,7-diones (11 derivatives) and other types of quinones (10 derivatives). Their reduction (oxidation) Gibbs free energy variations cover a broader energy range than phenothiazine which spans between -82.39 (55.18) and 51.84 (213.77) kcal mol^{-1}.

The 2,2'-bipyridinium subset counts 45 derivatives. It is composed of unbridged 2,2'-bipyridiniums (3 derivatives), ethene (3 derivatives), ethane (19 derivatives), propane (6 derivatives) and butane 2,2'-bipyridiniums (3 derivatives) as well as other types of 2,2'-bipyridiniums (11 derivatives). Their reduction Gibbs free energy variations span between -228.58 and -5.67 kcal mol^{-1} while their oxidation ones varies between 182.06 and 402.04 kcal mol^{-1}.

Finally, the viologen subset is composed of 49 derivatives, belonging to core (20 derivatives), symmetric (12 derivatives) and asymetric (7 derivatives) substituted viologens as well as other types of viologens (10 derivatives). Their reduction (oxidation) Gibbs free energy variations cover a similar energy range as 2,2'-bipyridinium. It spans between -226.69 (185.29) and -10.59 (407.36) kcal mol^{-1}.

Please note that a more complete description of the database is provided in the Supporting Information file, and the structures and respective energies and Gibbs free energies are accessible through the GitHub platform [25].

III. COMPUTATIONAL DETAILS

With the collected database, we now perform the necessary computations and prediction of the redox potentials. FIG. 3 exhibits the workflow of the procedure, which breaks down into five steps. Three steps during the training procedure include: (I) Computing the redox potentials for each compound in training set using DFT. The corresponding computational method is detailed at the following part of this section. (II) Constructing the descriptor representation for each compound in the training set, which is detailed in Section IV A. (III) Training a ML model with the given descriptors and redox potentials. The list of models used in this paper is described in Section IV B. After that, a ready-to-use model is established. When a new chemical compound with unknown redox potentials is given, we (IV) construct the descriptors for this compound the same way as in step (II), and then (V) use these descriptors as the input for the trained model, and take the output of the model as the predicted redox potential.

We first present the QC methodology to generate the reference data (I). The Gibbs free energy variations in reduction and oxidation (see Eq. 1 and 2) are computed in gas phase with density-functional theory (DFT). The

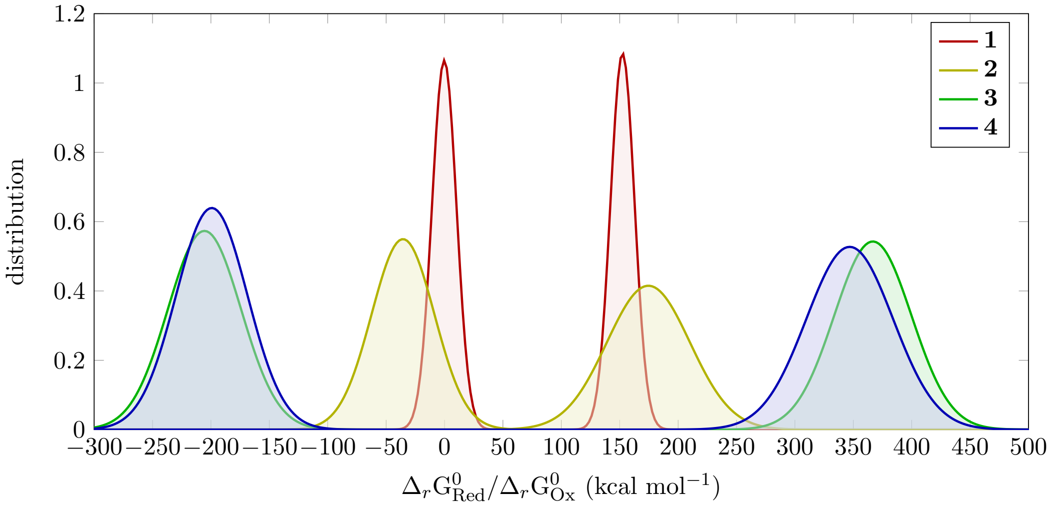

FIG. 2: Distribution of the reduction and oxidation Gibbs free energy variations (kcal mol^{-1}) covered by the ORedOx159 database. Indices **1**, **2**, **3** and **4** refer to the phenothiazine, quinone, 2,2'-bipyridinium and viologen subsets of the database.

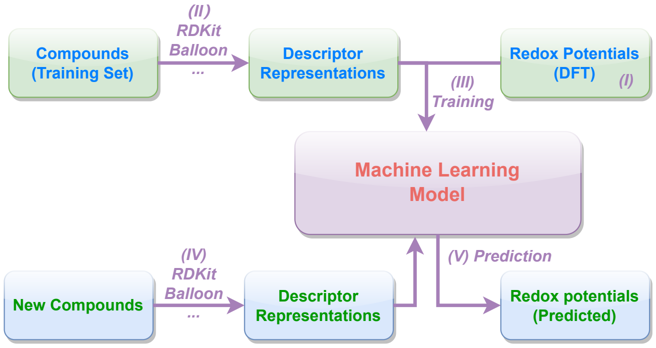

FIG. 3: The workflow for the redox potential prediction.

structures of the 159 organic derivatives are fully optimized in their neutral ground-state as well as in their reduced and oxidized states. Their geometries are thus fully relaxed. In other words, we do not consider vertical electron removal or attachment. Each of the 477 resulting minimum energy structures is then fully characterized by a frequency computation (within the harmonic approximation) from which the thermal, vibrational and entropy contributions are computed at room temperature using standard statistical physics formulae.

The PBE0 global-hybrid density functional approximation for the exchange-correlation energy [26] and the def2-SVP basis set [27] are chosen as a good accuracy/-cost trade-off to estimate the requested energy property for this large number of compounds. All the computations are performed with the release C.01 of the Gaussian'16 program [28] using a tight energy threshold criterion as well as an ultrafine integration grid.

It should be underlined that only gas phase values are here considered. Obviously, experimental data are in general related to measurements in solution. However, an accurate description of solvation effects in redox processes is far from being straightforward, and is clearly outside the scope of this paper. The same remark applies for a detailed assessment of the chosen QC methodology. We refer the interested reader to the valuable works, among others, by the groups of Gillmore [29] and Grimme [30].

IV. MACHINE LEARNING FRAMEWORK

With redox potential references generated, we now provide an in-depth elucidation of the ML framework utilized in this paper. We begin with introducing the descriptors utilized, and we subsequently delineate the ML models employed, corresponding respectively to steps (II) and (III) in FIG. 3.

A. Descriptor design

As stated in introduction, our models will be grounded in graph theory. In a nutshell, a graph G is an ordered pair, $G = (V, E)$, of vertices (also named points or nodes) and edges (also known as links or lines) that are unordered pairs of vertices. In a chemical context, we first construct a graph from any molecule by identifying atoms as nodes and by modeling chemical bonds by edges. It is thus obvious that the usual molecular representation of a chemical compound can be straightforwardly translated into the language of graph theory.

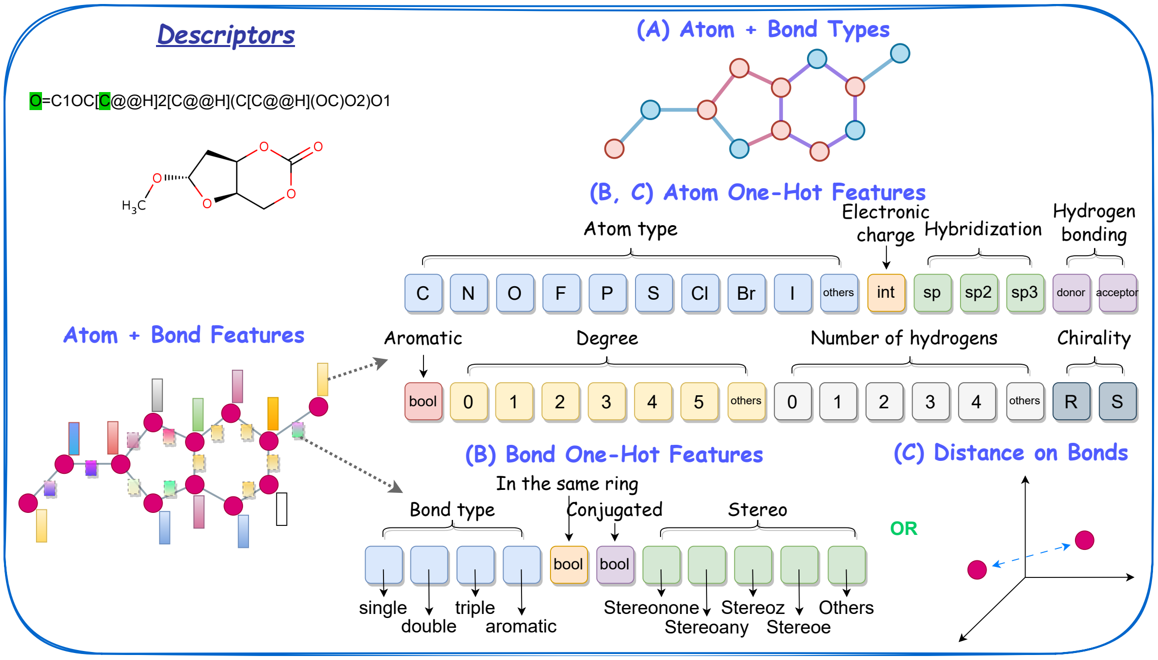

FIG. 4: Descriptors used for the machine learning models.

This does not mean that only interactions between bonded atoms exist. Actually, a full molecular graph will be in principle a complete graph (*i.e.* all pairs of nodes are linked - which is not a synonym of bonded - together), and one can for in principle evaluate the interaction energies between all nodes (for instance, using Pendás' Interacting Quantum Atoms (IQA) scheme [31]. However, such graphs rapidly become huge since they involve $n(n-1)/2$ edges (with n the number of atoms), so that their use would introduce a computing complexity that can be prohibitive for large databases of extended molecules. Conversely, a linear molecule (*i.e.* without rings or cages) exhibits (as a direct consequence of the Poincaré-Hopf relationship within the framework of Bader's Quantum Theory of Atoms-In-Molecules (QTAIM) [32]) only $n-1$ bonds, so that linear scaling may be possible with such graph assumption. An important point to notice is that, in principle, the node properties, if relevantly chosen, can reflect both bonded and non-bonded interactions.

Then, once the molecular graphs established, we consider in total 3 types of descriptors for graph-based models in this research. As we intend this paper to have some didactic content, we have decided here to describe them into more detail with no prerequisite on chemoinformatics. They are also represented in FIG. 4.

(A) Atom + Bond Types: This descriptor considers element types (for instance carbon, hydrogen...) as node features and chemical bond types (either single, double, triple, aromatic, see more details in type **B**) as edge features. Since these features are discrete, they are encoded in a one-hot manner. These descriptors, while affording a simplistic description of a molecule, are widely used in various databases for the evaluation of machine learning models, such as *TUDataset* [33]. However, this descriptor may lack the ability to capture sufficient information for effective learning.

(B) One-Hot Features: Such descriptors encompass common features for both atoms and chemical bonds, each of them represented as one-hot coded node and edge features, respectively. These descriptors are inspired by the featurizers in *DeepChem* [34], based on WeaveNet paper [35]. They can all be directly obtained from the *RDKit* [36] software without the need for computationally intensive quantum chemistry-informed descriptors.

In practice, they are actually very often directly generated from the Simplified Molecular Input Line Entry Specification (SMILES) [37] molecular code (see an example of the top left corner of FIG. 4) using the standard rules of chemistry based on Lewis representation , without resorting to any kind of advanced calculations. For relatively simple molecules in organic chemistry, the bond pattern predicted by SMILES code is fully consistent with ones from more elaborated approaches (for instance QTAIM already mentioned, which affords univocal definition of chemical bonds) based on QC analysis. This is one of the main strengths of SMILES code since it can be produced and processed in an almost instantaneous manner. However, as it does not encode any 3D information, it cannot distinguish between the different conformers (resulting for instance from free rotations around single bonds) of a given molecule.

More precisely, the node features include a 32-dimensional one-hot vector (see middle part in FIG. 4). The first 10 components encode the chemical symbol with the following possible categories: C, N, O, F, P, S, Cl, Br, I, and others. For instance, a carbon atom will be coded by (1 0 0 0 0 0 0 0 0 0 0) while an oxygen atom will be represented by (0 0 1 0 0 0 0 0 0 0 0). The hybridization type is encoded by three components depending on it is *sp*, sp^2 or sp^3 (a carbon engaged into a double bond will hence be coded as (0 1 0).

The other atom descriptors are the electronic charge (as its evaluation is not straightforward - ideally it requires a QC calculation -, we decided to disregard it by setting it to zero for all atoms), its aromaticity (represented by a boolean depending whether the atom belongs to an aromatic system or not), its so-called "*Degree*" (corresponding to the coordination number, the number of hydrogen atoms linked to the studied node (one-hot coded as a 6-dimension feature), and its possible chirality (R or S absolute configuration, once more in one-hot encoding) in the case of an asymmetric carbon atom. This representation is thus able to address stereochemistry issues.

The edge features consist of an 11-dimensional one-hot vector (see bottom part of FIG. 4) representing bond type (four categories: single, double, triple and aromatic) , whether the atoms at the end of the bond share the same ring (no matter the type) and whether the bond belongs to a conjugated system (both encoded as a boolean), and the possible stereoisomerism of the bond in a 5-dimension feature).

Notice that some structure information is implicitly embedded into the node features through "*Aromatic*", "*Degree*", and "*Chirality*". This may benefit the models that can not deal with edge features or rings. We emphasize that these features are easy-to-get and non-expensive, thus exhibiting how much ML models can benefit from a short amount of effort of descriptor engineering.

(C) One-Hot Features on Nodes + Distances on Bonds: This descriptor employs the same node features as the descriptor (B), but replaces edge features with the Euclidean distances (here expressed in Angström) between the nuclei of the corresponding bonded atoms (see bottom right corner in FIG. 4). We utilize distances to maintain a 3D equivariant system (that-is-to-say the values remain the same when the molecule is translated and rotated as a whole), making it applicable to any models with the ability to handle continuous edge features (at variance with the previous bond descriptions that used only discrete descriptors). It can be remarked that internuclear distances are also the main ingredient of the celebrated Coulomb matrices (CMs) [38] that enter many ML models in chemistry. However, it should be noticed that distances, here (at variance with that is currently done when using CMs) are not computed for non-bonded pair of atoms even if they can add additional information on the geometry adopted by the molecule.

To obtain these distances, we search for the most stable conformer for all compounds using the freely available *Balloon* [39] software with the MMFF94 molecular mechanics force field, starting from the SMILES code, with the following options: *–nconfs 20 –nGenerations 300 –rebuildGeometry*. We then compute the distance from the corresponding Cartesian coordinates of the nuclei. This computation, though probably less accurate than a QC geometry optimization, has however an almost negligible computational cost, fulfilling one of the targets of our ML modelling.

For the same reason, no QC descriptors are used at all. Obviously, some of them would be certainly useful for an accurate prediction of redox properties. For instance, the energy of the Highest Occupied Molecular Orbital (HOMO) energy (in Kohn-Sham DFT) is known to be exactly equal (this is due to the asymptotic form of the electron density in the exponential tail) to the opposite of the electronic component (thus without thermodynamic contributions) of the vertical (*i.e.* without geometry relaxation) ionization energy if the exact exchange-correlation functional is used (which is unfortunately not known in analytical form). It is also known that the energy of the lowest unoccupied molecular orbital (LUMO), while differing in principle from the opposite of the vertical electron affinity (as a consequence of the discontinuity of the exchange-correlation potential), is nevertheless linked to electron capture (see for instance a detailed discussion by Baerends [40]).

These descriptors, which have already been used in the literature to build predictive models, unfortunately require the computationally demanding task of a DFT calculation. This is exactly what we would like to avoid thanks to a well tailored ML model based on descriptors that can be computed at an almost zero cost.

B. Machine learning models

This section introduces three types of models, represented in FIG. 5, employed in this paper to address graph prediction problems: Graph Edit Distances (GEDs), Graph Kernels (GKs), and Graph Neural Networks (GNNs), baselined upon a set of vector-based machine learning models.

Graph Edit Distances: GEDs measure the dissimilarity between two graphs by evaluating the amount of distortion required to transform one graph into another. Various basic edit operations, each assigned with a cost, are used to measure this distortion, and the minimum total cost represents the GED between the two graphs. Commonly used edit operations include an insertion, removal, or substitution of vertices or edges. FIG. 5(III) shows a simple instance of an edit procedure. In this basic example, the first operation is the removal of the bottom vertex, followed by the removal of the bottom edge.

Then an edit path π, which in FIG. 5 goes from graph G to graph G', can be defined as the sequence of these operations, whose total cost c is the sum of the costs of all edit operations in the path, namely $c(\pi) = \sum_{e \epsilon \pi} c(e)$, where e denotes an elementary operation. Then, the

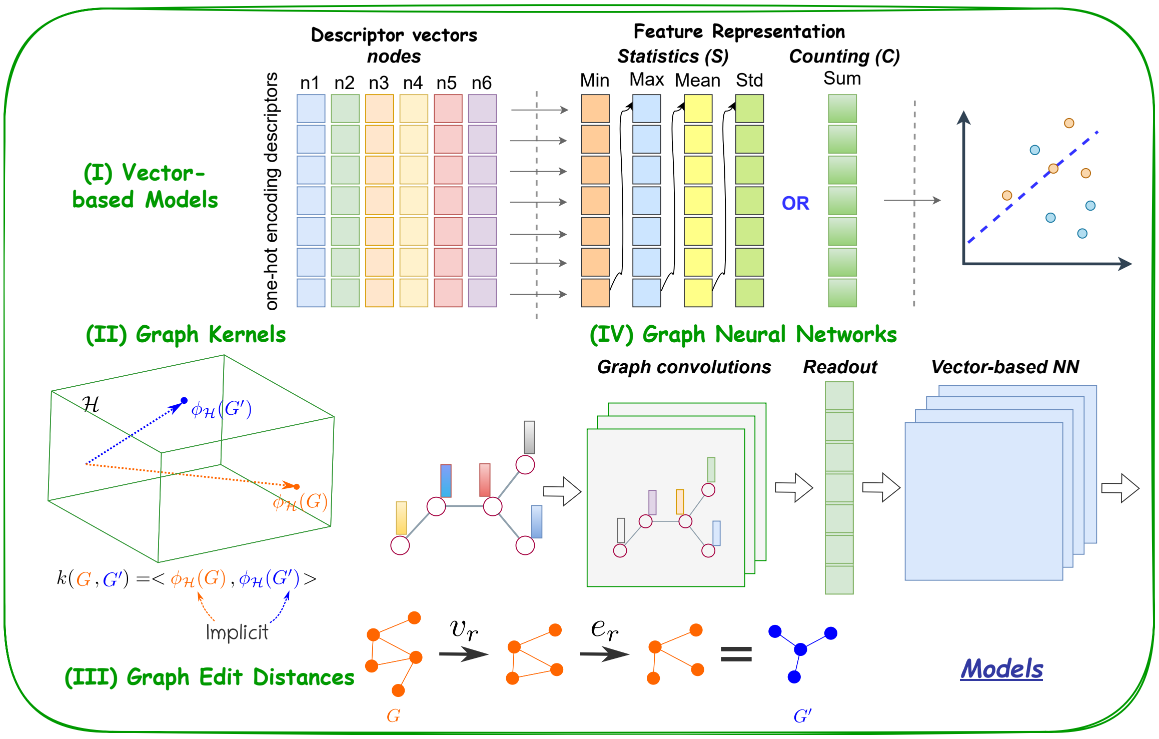

FIG. 5: Machine learning models used for the prediction.

GED between two graphs G_1 and G_2 is defined as the minimum cost associated with any possible edit path:

$$\texttt{ged}(G_1, G_2) = \min_{\pi \in \Pi(G_1, G_2)} c(\pi), \tag{4}$$

where $\Pi(G_1, G_2)$ denotes the set of all possible edit paths from G_1 to G_2. We mention here that we only consider in this paper a constant value for the cost (*i.e.* it is independent of the labels of nodes and edges) of every of the six elementary operations.

Since computing GED is an NP-hard problem [41], approximation methods have been proposed to estimate it. In this paper, we use a milestone method, `Bipartite` [42], where the edit costs are optimized using a method from [43]. This approach boils down the quadratic assignment problem (QAP) to a linear one by finding a suboptimal edit path only based on local structural information. Edit costs are optimized in a bi-alternate optimization scheme. This approach constitutes a good trade off between computational time and accuracy. More accurate methods based on the QAP have been proposed [44, 45], at the cost of an higher computational time. We recommend in-depth papers [46, 47] on GEDs to interested readers.

As GEDs are a distance measure, we combine them with the k-Nearest Neighbor regressor (KNN) [48] for the final prediction. To predict the output value of an example, this approach simply consists in averaging the target values associated to its k nearest neighbors according to the GED, with the hypothesis that the target values of similar examples are close to each other. The contribution of each neighbour can be weighted according to distance values.

The advantages of GEDs inherit from their flexible and explicit application of edit operations, which allows for capturing complex structural differences between graphs, as well as an explicit demonstration of the modification process of graphs. This latter merit helps establish the explainability of the model, which is conductive to their application on the generative tasks. However, the semantic meaning of graph elements may not be captured by GEDs. Moreover, as a NP-hard problem, calculating GEDs can be computationally expensive, especially for large or dense graphs. The lack of providing a continuous space to interpolate between graphs and invariance to graph size may also limit their applications to certain tasks.

Graph Kernels: Graph kernels compute a similarity measure between graphs by implicitly mapping them into high-dimensional spaces where the inner product is computed. Such a measure can be defined as a symmetric, positive semi-definite function $k : \mathcal{G} \times \mathcal{G} \to \mathbb{R}$ on a space of graphs $\mathcal{G}$, where a map $\phi : \mathcal{G} \to \mathcal{H}$ exists into a Hilbert space $\mathcal{H}$, so that

$$k(G_1, G_2) = \langle \phi(G_1), \phi(G_2) \rangle_{\mathcal{H}} \tag{5}$$

for all pairs of graphs $G_1, G_2 \in \mathcal{G}$. Here $\langle \cdot, \cdot \rangle_{\mathcal{H}}$ is the inner product in $\mathcal{H}$, as shown in FIG. 5(II). The results are then processed by a kernel machine for final prediction, specifically Kernel Ridge Regression (KRR) [49] in this paper. By altering the design of the mapping, various graph kernels can be constructed. A commonly design strategy inherits from R-convolution kernels, which measures similarity between two objects, by measuring the similarities between their substructures [50].

Normally, a series of sub-kernels are first constructed between pairs of sub-structures of two graphs, then a graph-level kernel is established upon the summation of these sub-kernels. In our paper, we examine a set of baseline graph kernels, including Shortest Path Kernel (SP) [51], Structural Shortest Path Kernel (SSP) [52], Path Kernel [16], Treelet Kernel [53], and Weisfeiler-Lehman Subtree Kernel (WLSubtree) [54], each of them named after the sub-structures from which it builds the kernel. These kernels are able to tackle linear and non-linear sub-structures. Detailed descriptions and comparisons of graph kernels can be found in [55–58].

Graph kernels are able to work directly with the graph structures, while tackling the similarity measure on high or infinite dimension spaces, thanks to the kernel trick [59]. This allows bypassing acquiring fixed embeddings, extends their expressiveness and flexibility of design and integration of prior-knowledge. The proper design can lead to the invariance to graph isomorphism as well. The other size of the coin minted their lack of the ability of automatic learning of the representation, especially compared to graph neural networks. The choice of the appropriate kernel is thus crucial for achieving good performance.

Graph Neural Networks (GNNs): GNNs are a class of neural networks that specialize in learning from graph-structured data. They often utilize a message-passing scheme, wherein information from nodes, edges, and the overall graph structure, are aggregated from the neighbors of each node through a series of graph convolutional operations, as shown in FIG. 5(IV). With denoting $\mathbf{x}_i^{(k-1)} \in \mathbb{R}^F$ node features of node i in layer $(k-1)$, and $\mathbf{e}_{j,i} \in \mathbb{R}^D$ denoting (optional) edge features from node j to node i, update of node features using message passing graph neural networks can be described as

$$\mathbf{x}_i^{(k)} = \gamma^{(k)}\left(\mathbf{x}_i^{(k-1)}, \bigoplus_{j\in\mathcal{N}(i)} \phi^{(k)}\left(\mathbf{x}_i^{(k-1)}, \mathbf{x}_j^{(k-1)}, \mathbf{e}_{j,i}\right)\right), \tag{6}$$

where $\bigoplus$ denotes a differentiable, permutation invariant function (*e.g.* sum, mean or max), and ϕ denote differentiable functions such as MLPs (Multi Layer Perceptrons). $\gamma^{(k)}$ can be thought of as the node's update or processing function at that particular layer, which can be as well implemented as a neural network layer. This formula captures the message-passing operation of GNNs. At each layer, information is propagated from neighboring nodes to the central node i [60]. Following this message-passing process, a pooling operation (such as sum, mean, or max) over node features, is often applied to obtain a graph-level representation; a MLP can then be used for graph-level predictions. We refer to [61–64] readers interested in GNNs.

Several representative GNN architectures have been considered for study. The Message Passing Neural Network (MPNN) [65] employs message passing and aggregation mechanisms in combination of an edge network to capture relational information. In contrast, the Graph Convolutional Network (GCN) [66] utilizes convolution-like operations with weight sharing for neighbor node information aggregation. The Deep Graph Convolutional Neural Network (DGCNN) [67] distinguishes itself through the use of edge features and sort pooling to learn graph embeddings invariant to permutation of the nodes. The Graph Isomorphism Network (GIN) [68] combines a graph isomorphism test with learnable functions to integrate global and local information. On the other hand, the Graph Attention Network (GAT) [69] employs attention mechanisms to selectively weight neighboring node contributions.

Finally, the Unified Message Passing Framework (UniMP) [70] integrates message passing and transformer-based architectures, harnessing both relational and self-attention mechanisms for comprehensive graph representation learning. These diverse models offer distinct approaches to address our problem.

One significant advantage of GNN models lies in their time complexity during inference, setting them apart from GEDs and graph kernels. In the case of GEDs and graph kernels, it is typically necessary to perform metric or Gram matrix computations, involving the calculation of similarity values between each data point in the training set and the data point to be inferred. This matrix serves as the inner product in kernel machines and aids in finding the K nearest neighbors in K-Nearest Neighbors algorithms. The catch is that the computational time associated with this matrix scales linearly with the size of the training set, rendering it prohibitively time-consuming for large datasets. As a result, the scalability of these models becomes constrained.

In contrast, GNN models usually feature a linear or nearly linear time complexity during inference. This efficiency is intricately linked to the model's design, particularly the scale of weight parameters. In practice, GNN computations predominantly involve basic matrix operations like multiplication and addition. Consequently, GNN models excel in efficiently handling graphs of varying sizes, making them exceptionally well-suited for real-world scenarios. These scenarios often involve graph data that ranges from small and simple to large and intricate, then underlining the substantial benefits of GNNs in real-world applications.

More advantages are offered by GNNs, notably in their ability to automatically learn representations, which alleviates the need for explicit feature engineering. By aggregating and propagating attributes associated with nodes and edges within the graph, the incorporation of both local and global information is facilitated for a comprehensive understanding of the graph's context. Additionally, GNNs allow for knowledge transfer through a pretraining and fine-tuning strategy. However, they come with certain limitations. GNNs often require substantial amounts of training data and memory resources, which can be restrictive in real-world applications.

The lack of proper interpretability and explainability poses challenges, potentially diminishing trust among domain experts. Furthermore, GNNs inherit unresolved fundamental issues, such as over-smoothing, where nodes tend to lose their distinctiveness after a certain number of GNN layers, limiting the feasibility of designing deep GNN architectures. These complexities need further investigation and innovation to fully harness GNNs' potential in practical settings.

In addition to graph-based models, we also consider **vector-based models** [49] as baselines, namely Linear Regression (LR), Gaussian Kernel Ridge Regression (GKRR), Support Vector Regression (SVR), Gaussian Process Regression (GPR), Random Forest (RF), Gradient-boosted Decision Trees (GBDT), and K-Nearest Neighbor Regression (KNN). Given the vector-focused nature of these models, a graph-level pooling step is required beforehand to derive fixed size vectors from varying size graphs. For this purpose, we adopted the strategy outlined in [71], wherein statistics are computed for each feature over all nodes within each graph (stats). Indeed, the number of nodes varies along the molecular dataset, so that the vector gathering the node features would have, in general a different size for one chemical system to another one, precluding the use of standard ML methods. To avoid this, for a given feature, we compute the minimal, maximal, mean and standard deviation values on all nodes, resulting in four descriptors that can be evaluated for all molecules.

Additionally, we introduced a novel pooling approach, where we quantified the frequency of each feature. As discussed in Section IV A, the node features, which encompass Atom and Bond Types (A) and One-Hot Features (B), are encoded using a one-hot representation in this paper. When two feature vectors are summed, it results in a vector where each element corresponds to the total count of that feature. To illustrate this concept, consider a simple two-node toy graph with feature vectors (1 0 0 1 0) and (0 1 0 1 0) assigned to the two nodes. After applying our pooling approach, the graph-level representation becomes (1 1 0 2 0), obtained through element-wise summation of the two node feature vectors.

This means that features at the first, second, and fourth positions occur once, once, and twice, respectively, in the entire graph, while the other features do not appear. For a visual representation, refer to FIG. 5. Our ML experiments show that this pooling strategy outperforms the one proposed in [71] in most cases, as illustrated in Section V. It is important to note that these models primarily emphasize node features and that, during the experiments, edge features were disregarded.

V. RESULTS AND DISCUSSION

Performance evaluations are carried out on ten different random splits. Each split is partitioned into 80%, 10%, and 10% for training, validation, and testing, respectively. We calculate the Average Mean Absolute Errors (MAEs) over the 10 splits as the final results, estimated on the respective test sets. For the sake of reproducibility, the code is available through the GitHub platform [72].

TABLE I presents the MAE values obtained for each ML experiment. The ± sign indicates the 95% confidence interval computed over the 10 repetitions. Unsupported descriptors for each model are denoted by "-". Notably, the best descriptor for each model is highlighted in bold, the optimum result for each descriptor across all models is marked in green, and the superior results across all experiments are underlined. In the case of vector-based models, we focus on the results obtained through count pooling as it consistently outperforms statistical pooling. This count pooling thus represents an interesting improvement over the methods previously used by us to deal with the non-constant number of nodes along the whole dataset. An exception is observed for the *GPR* model with the One-Hot descriptor, where statistical pooling yields better results, and thus we present both pooling methods.

When **comparing different descriptors**, a clear trend emerges with the One-Hot descriptor prevailing in most models: 16 out of 19 for reduction Gibbs free energies and 15 out of 19 for oxidation Gibbs free energies. In the context of graph-based models, this dominance continues as 8 out of 11 models for both targets. The following is The Atom Bond Types descriptor. Importantly, the performance gap among descriptors is significant on a model-wise basis, which often surpasses the differences in performance between models. This phenomenon is particularly pronounced for vector-based models, likely due to their limitation in handling only node features. The One-Hot descriptor, encoding implicit structural information, proves highly beneficial in this context.

A similar pattern is observed for most GNN models, emphasizing the crucial role of feature engineering in their performance. An exception is for reduction values *Red*, where the One-Hot descriptor excels. This suggests the intriguing potential of transformer-like models in capturing core information for graphs with limited features, even on smaller datasets. Conversely, the Distance descriptor is evaluated only for models supporting continuous edge features. While its performance closely aligns with that of the One-Hot descriptor, it never surpasses the latter. This can be attributed to the reduction in information when utilizing distances instead of 3D coordinates, and the fact that none of the models are specifically designed for a 3D equivariant system, especially with limited learning space.

Further insights emerge when **comparing different models**. Notably, all optimal results are obtained by

TABLE I: The prediction MAE (in kcal/mol).

Models	$\Delta_r \mathbf{G}^0_{\mathbf{Red}}$			$\Delta_r \mathbf{G}^0_{\mathbf{Ox}}$		
	AB Types	**One-Hot**	**Dis**	**AB Types**	**One-Hot**	**Dis**
LR	56.0±10.8	**29.4±2.7**		56.1±11.1	**31.0±2.5**	-
GKRR	28.9±7.5	**14.4±5.0**	-	30.5±10.4	**17.5±4.4**	-
SVR	26.0±9.2	**14.1±3.7**	-	31.4±10.3	**17.2±5.2**	-
GPR (stats)	42.3±6.5	**21.0±5.5**	-	43.2±8.8	22.0±3.0	-
GPR (count)	36.8±7.3	**26.2±4.6**	-	37.6±8.8	**29.6±5.1**	-
RF	19.7±5.0	**14.3±4.6**	-	22.3±5.4	**15.2±4.0**	-
GBDT	19.6±5.1	**11.9±4.1**	-	20.4±4.9	**13.7±3.2**	-
KNN	25.3±9.9	**10.2±3.9**	-	26.2±8.2	**11.3±5.4**	-
SP	29.6±14.4	**12.5±2.1**	-	31.9±13.9	**14.0±2.1**	-
SSP	**9.5±2.3**	14.7±2.8	14.4±3.2	**11.9±2.7**	16.6±2.8	16.9±3.4
Path	**9.7±2.1**	10.7±2.4	-	**12.8±2.7**	**12.8±2.3**	-
Treelet	**9.9±2.7**	25.1±15.9	-	**13.2±2.9**	23.0±11.3	-
WLSubtree	12.4±2.1	**12.0±1.8**	-	**12.0±1.8**	13.6±2.2	-
GED (fitted)	11.9±4.2	**8.0±1.8**	9.5±4.0	14.3±5.1	**12.1±2.7**	13.8±4.9
MPNN	8.2±3.7	5.8±1.9	**5.6±0.6**	8.7±3.2	**7.2±2.5**	10.7±3.3
GCN	21.5±7.6	**8.4±3.2**	-	24.5±8.4	**8.8±2.0**	-
DGCNN	17.7±6.2	**8.8±3.2**	-	22.1±6.2	**13.5±4.8**	-
GIN	26.9±9.0	**13.7±5.0**	-	24.7±9.7	**13.0±2.9**	-
GAT	11.5±4.9	**6.9±1.4**	8.0±2.3	11.1±6.6	**8.5±1.5**	9.9±2.2
UniMP	**7.3±1.5**	**7.4±3.0**	7.9±1.0	18.0±18.3	**9.1±2.6**	20.3±24.5

graph-based models, particularly GNNs when using the One-Hot descriptor. The best performances across all experiments for both targets are achieved by GAT on the One-Hot descriptor, underlining the flexible capabilities of graph-based models and the representational power of GNNs. However, when a better-engineered descriptor is applied, the performance of a "simpler" model tends to approach that of a more complex model.

For instance, with the One-Hot descriptor, the performance of KNN closely rivals and surpasses the best graph kernels, while GED outperforms 3 out of 5 GNNs for reduction and 2 out of 5 for oxidation. Comparing GKRR and graph kernels, both employing Kernel Ridge Regression for predictions, GKRR achieves similar performance to three graph kernels, including WLSubtree, and surpasses one of them. It is important to highlight that with the One-Hot descriptor, numerous graph kernels, GED, and even vector-based models outperform GNNs with the Atom Bond Types descriptor, underscoring the critical role of descriptor engineering.

On the whole, with the chosen low-level and computational-friendly descriptors, the best ML model achieves a prediction MAE of 5.8 kcal mol^{-1} for reduction and 7.2 kcal mol^{-1} for oxidation on the ORedOx159 database. It corresponds to a MAE ranging between 0.2 and 0.3 V on the potential, an error which is in line with the state-of-the-art approach recently developed in computational chemistry [30].

We collected the test data from all cross-validation splits, comprising a total of 180 compounds. Subsequently, we generated plots comparing the Gibbs free energies calculated by DFT with their predicted counterparts, as shown in FIG. 6. These comparisons were made using the best-performing prediction system, which includes the MPNN model, as well as the Dis descriptor and the One-Hot descriptor for reduction and oxidation tasks, respectively. The figure displays high correlations between the DFT-calculated and the predicted values. Specifically, the R^2 scores amount to 0.992 for reduction and 0.988 for oxidation. These substantial correlation coefficients validate the accuracy of our predictive model for these tasks and affirm its predictive abilities. This level of agreement reflects the model's suitability for practical applications in predicting reduction and oxidation potentials.

Lastly, we briefly discuss the computational effort associated to these ML procedures. Figure 7 illustrates the reference time required for each datapoint or compound. The values are presented in seconds and log-scaled with a base of 10. In this representation, faster reference times, characterized by smaller values, are indicated by the blue color, while slower times are depicted in red. Among the evaluated models, GNN models demonstrate superior efficiency in terms of reference times compared to graph kernel and GED models. For GNN models, reference times can be as low as 10^{-4} seconds, as seen in the case of GCN, DGCNN, and GIN. The most accurate model, MPNN, achieves reference times of approximately 0.001 seconds for reduction and 0.01 seconds for oxidation. These values represent, as expected, a dramatic improvement in efficiency compared to quantum chemistry-based methods, for which minutes or hours are needed for a molecule.

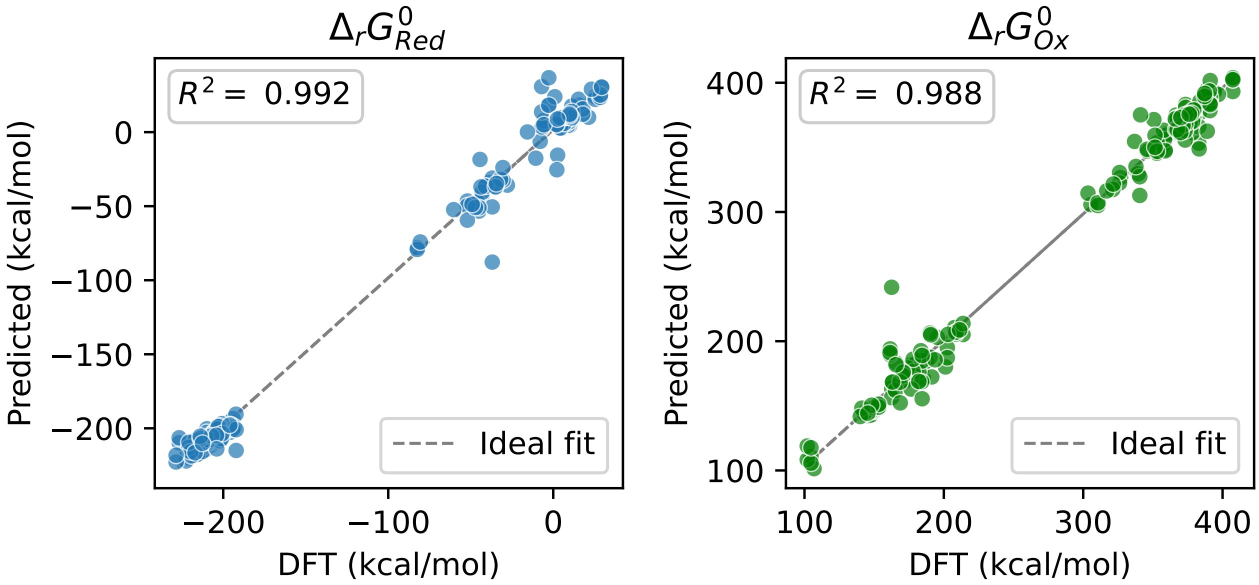

FIG. 6: Potentials computed by DFT vs the ones predicted via the best descriptor and model.

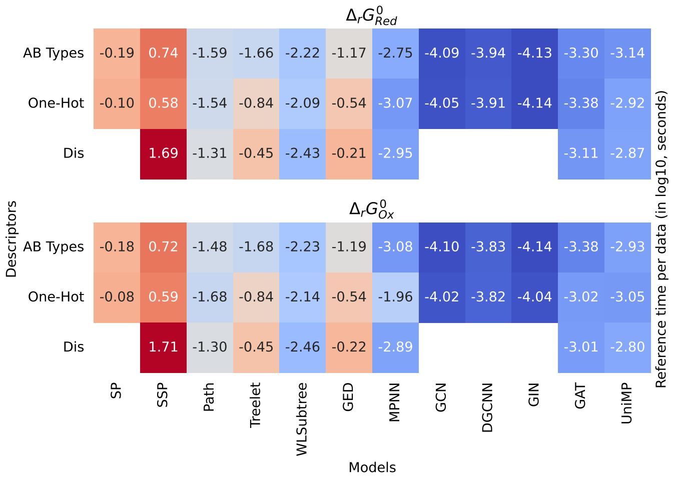

FIG. 7: Reference time per datapoint for graph-based models, in log-scaled with base 10, and in seconds.

VI. CONCLUSION AND FUTURE WORK

In this study, we proposed a comprehensive, homogeneous, and reliable database ORedOx159 (based on density functional theory calculations), which collects one-electron reduction and oxidation reactions involving large-size organic compounds of experimental interest. We then applied a series of machine learning models on this dataset to predict the reduction and oxidation Gibbs free energies using several descriptors computed from the structures of the compounds obtained at the cheap molecular mechanics level. The analyses of the results suggests that with the proper choice of descriptors and machine learning models, a relatively high prediction accuracy can be achieved, where the prediction time is considerably reduced.

Looking ahead, future research work will include several dimensions. Firstly, from the chemical point of view, the next important step will be to *(i)* include to our model solvation effects, and *(ii)* extend the training database to other chemical families of high interest in electrochemistry. Then, in comparison to tabulated reference potentials, our research output will become a robust built-in package to fast and accurately predict redox potentials of organic compounds in solutions, as it is for instance the case for NMR spectrum simulations.

Secondly, state-of-the-art machine learning models may serve as better prediction tools, especially the ones tailored for our problem. For example, Graph Neural Networks and transformer-like models, particularly focusing on leveraging 3D coordinates and leveraging state-of-the-art models may unlock new capabilities and insights. Other machine learning strategies, such as pretraining, may also unlock the potential of better prediction. Thirdly, we aim at exploring more suitable descriptors that can better balance the representation ability and the acquiring time complexity. Lastly, we plan to broaden our evaluation criteria to incorporate additional important metrics such as robustness, interpretability and explainability, and overall model trustworthiness.

Finally, from a more general perspective, we are convinced that significant progress can be made mainly if the two involved scientific communities (*i.e.* theoretical chemistry researchers and data scientists) enter a fruitful dialogue, and manage not only to share their tools but also to build a common language.

ACKNOWLEDGMENTS

E.B. and L.Z. thank Dr. P.P. Lainé (ITODYS lab, CNRS) for fruitful discussions regarding the construction of the ORedOx159 database.

E.B. gratefully acknowledges ANR (Agence Nationale de la Recherche) for the financial support of this work through the MoMoPlasm project (ANR-21-CE29-0003). He thanks also ANR and CGI (Commissariat à l'Investissement d'Avenir) for their financial support of this research through Labex SEAM (Science and Engineering for Advanced Materials and devices) ANR-10-LABX-096, ANR-18-IDEX-0001, and acknowledges TGGC (Très Grand Centre de Calcul du CEA) for computational resources allocation (Grant No. AD010810359R1).

This work has also been partially supported by University of Rouen Normandy, INSA Rouen Normandy, the "Centre National de la Recherche Scientifique" (CNRS), the European Regional Development Fund (ERDF), Labex SynOrg (ANR-11-LABX-0029), Carnot Institut I2C, the graduate school for research XL-Chem (ANR-18-EURE-0020 XL CHEM), and the "Région Normandie". The Centre Régional Informatique et d'Applications Numériques de Normandie (CRIANN) is acknowledged for providing access to computational resources to L.J., V.T. and L.J.

[1] C. W. Coley, Defining and exploring chemical spaces, Trends in Chemistry **3**, 133 (2021).

[2] J. A. Keith, V. Vassilev-Galindo, B. Cheng, S. Chmiela, M. Gastegger, K.-R. Müller, and A. Tkatchenko, Combining machine learning and computational chemistry for predictive insights into chemical systems, Chem. Rev. **121**, 9816 (2021).

[3] A. C. Mater and M. L. Coote, Deep learning in chemistry, J. Chem. Inf. Model. **59**, 2545 (2019).

[4] A. Hagg and K. N. Kirschner, Open-source machine learning in computational chemistry, J. Chem. Inf. Model. **63**, 4505 (2023).

[5] J. Schüring, H. D. Schulz, W. R. Fischer, J. Böttcher, and W. H. Duijnisveld, in *Redox: Fundamentals, Processes and Applications* (Springer-Verlag, 1999).

[6] L. Stryer, in *Biochemistry (fourth ed.)* (Freeman and Company, 1995).

[7] F. Oeters, M. Ottow, H. Meiler, H. B. Lüngen, M. Koltermann, A. Buhr, J.-i. Yagi, L. Formanek, F. Rose, J. Flickenschild, R. Hauk, R. Steffen, R. Skroch, G. Mayer-Schwinning, H.-L. Bünnagel, and H.-G. Hoff, Iron, in *Ullmann's Encyclopedia of Industrial Chemistry* (John Wiley & Sons, Ltd, 2006).

[8] W. Baeckmann, W. Schwenck, and W. Prinz, in *Handbook of Cathodic Corrosion Protection, 3rd Edition* (Elsevier, 1997).

[9] Y. Holade, K. Servat, S. Tingry, T. W. Napporn, H. Remita, D. Cornu, and K. B. Kokoh, Advances in electrocatalysis for energy conversion and synthesis of organic molecules, ChemPhysChem **18**, 2573 (2017).

[10] R. Fedorov and G. Gryn'ova, Unlocking the potential: Predicting redox behavior of organic molecules, from linear fits to neural networks, Journal of Chemical Theory and Computation **19**, 4796 (2023).

[11] M. Kleinova, M. Hewitt, V. Brezova, J. C. Madden, M. T. D. Cronin, and Valko, Antioxidant properties of carotenoids: Qsar prediction of their redox potentials, Gen. Phys. Biophys. **26**, 97 (2006).

[12] D. D. Méndez-Hernández, J. G. Gillmore, L. A. Montano, D. Gust, T. A. Moore, A. L. Moore, and V. Mujica, Building and testing correlations for the estimation of one-electron reduction potentials of a diverse set of organic molecules, J. Phys. Org. Chem. **28**, 320 (2015).

[13] S. Ghule, S. R. Dash, S. Bagchi, K. Joshi, and K. Vanka, Predicting the redox potentials of phenazine derivatives using dft-assisted machine learning, ACS Omega **7**, 11742 (2022).

[14] B. G. Galuzzi, A. Mirarchi, E. L. Viganò, L. De Gioia, C. Damiani, and F. Arrigoni, Machine learning for efficient prediction of protein redox potential: The flavoproteins case, J. Chem. Inf. Model. **62**, 4748 (2022).

[15] V. Bhat, P. Sornberger, B. S. S. Pokuri, R. Duke, B. Ganapathysubramanian, and C. Risko, Electronic, redox, and optical property prediction of organic -conjugated molecules through a hierarchy of machine learning approaches, Chem. Sci. **14**, 203 (2023).

[16] L. Ralaivola, S. J. Swamidass, H. Saigo, and P. Baldi, Graph kernels for chemical informatics, Neural networks **18**, 1093 (2005).

[17] B. Gaüzère, L. Brun, and D. Villemin, Two new graphs kernels in chemoinformatics, Pattern Recognition Letters **33**, 2038 (2012).

[18] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, J. Bouwman, A. J. Brookes, T. Clark, M. Crosas, I. Dillo, O. Dumon, S. Edmunds, C. T. Evelo, R. Finkers, A. Gonzalez-Beltran, A. J. Gray, P. Groth, C. Goble, J. S. Grethe, J. Heringa, P. A. Hoen, R. Hooft, T. Kuhn, R. Kok, J. Kok, S. J. Lusher, M. E. Martone, A. Mons, A. L. Packer, B. Persson, P. Rocca-Serra, M. Roos, R. van Schaik, S.-A. Sansone, E. Schultes, T. Sengstag, T. Slater, G. Strawn, M. A. Swertz, M. Thompson, J. van der Lei, E. van Mulligen, J. Velterop, A. Waagmeester, P. Wittenburg, K. Wolstencroft, J. Zhao, and B. Mons, The fair guiding principles for scientific data management and stewardship, Sci Data **3**, 160018 (2016).

[19] N. Artrith, K. T. Butler, F.-X. Coudert, S. Han, O. Isayev, A. Jain, and A. Walsh, Best practices in machine learning for chemistry, Nat. Chem. **13**, 505 (2021).

[20] P. Wardman, Reduction Potentials of One-Electron Couples Involving Free Radicals in Aqueous Solution, Journal of Physical and Chemical Reference Data **18**, 1637 (1989).

[21] L. Michaelis and E. S. Hill, THE VIOLOGEN INDICATORS , J. Gen. Phys. **16**, 859 (1933).

[22] M. Hromadová and P. P. Lainé, Recent advances in electrochemistry of pyridinium-based electrophores: A structronic approach, Curr. Opin. Electrochem. **34**, 100996 (2022).

[23] M. M. Barsan, M. E. Ghica, and C. M. Brett, Electrochemical sensors and biosensors based on redox polymer/carbon nanotube modified electrodes: A review, Anal. Chim. Acta **881**, 1 (2015).

[24] F. De Proft, N. Sablon, D. J. Tozer, and P. Geerlings, Calculation of negative electron affinity and aqueous anion hardness using kohn–sham homo and lumo energies, Faraday Discuss. **135**, 151 (2007).

[25] Structures and energies the oredox159 database are freely accessible following the link `https://github.com/ANRMoMoPlasm/ORedOx159`.

[26] C. Adamo and V. Barone, Toward reliable density functional methods without adjustable parameters: The PBE0 model, J. Chem. Phys. **110**, 6158 (1999).

[27] F. Weigend and R. Ahlrichs, Balanced basis sets of split valence, triple zeta valence and quadruple zeta valence quality for H to Rn: Design and assessment of accuracy, Phys. Chem. Chem. Phys. **7**, 3297 (2005).

[28] M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, G. A. Petersson, H. Nakatsuji, X. Li, M. Caricato, A. V. Marenich, J. Bloino, B. G. Janesko, R. Gomperts, B. Mennucci, H. P. Hratchian, J. V. Ortiz, A. F. Izmaylov, J. L. Sonnenberg, D. Williams-Young, F. Ding, F. Lipparini, F. Egidi, J. Goings, B. Peng, A. Petrone, T. Henderson, D. Ranasinghe, V. G. Zakrzewski, J. Gao, N. Rega, G. Zheng, W. Liang, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, K. Throssell, J. A. Montgomery, Jr., J. E. Peralta, F. Ogliaro, M. J. Bearpark, J. J. Heyd, E. N. Brothers, K. N. Kudin, V. N. Staroverov, T. A. Keith, R. Kobayashi, J. Normand, K. Raghavachari, A. P. Rendell, J. C. Burant, S. S. Iyengar, J. Tomasi, M. Cossi, J. M. Millam, M. Klene, C. Adamo, R. Cammi, J. W. Ochterski, R. L. Martin, K. Morokuma, O. Farkas, J. B. Foresman, and D. J. Fox, Gaussian˜16 Revision C.01 (2016), gaussian Inc. Wallingford CT.

[29] E. J. Lynch, A. L. Speelman, B. A. Curry, C. S. Murillo, and J. G. Gillmore, Expanding and testing a computational method for predicting the ground state reduction potentials of organic molecules on the basis of empirical correlation to experiment, J. Org. Chem. **77**, 6423 (2012).

[30] H. Neugebauer, F. Bohle, M. Bursch, A. Hansen, and S. Grimme, Benchmark study of electrochemical redox potentials calculated with semiempirical and dft methods, J. Phys. Chem. A **124**, 7166 (2020).

[31] J. M. Guevara-Vela, E. Francisco, T. Rocha-Rinza, and A. Martin Pendas, Interacting quantum atoms – a review, Molecules **25**, 4028 (2020).

[32] R. Bader, in *Atoms in molecules: a quantum theory* (Oxford University Press, 1990).

[33] C. Morris, N. M. Kriege, F. Bause, K. Kersting, P. Mutzel, and M. Neumann, Tudataset: A collection of benchmark datasets for learning with graphs, arXiv preprint arXiv:2007.08663 (2020).

[34] B. Ramsundar, P. Eastman, P. Walters, V. Pande, K. Leswing, and Z. Wu, *Deep Learning for the Life Sciences* (O'Reilly Media, 2019) `https://www.amazon.com/Deep-Learning-Life-Sciences-Microscopy/dp/1492039837`.

[35] S. Kearnes, K. McCloskey, M. Berndl, V. Pande, and P. Riley, Molecular graph convolutions: moving beyond fingerprints, Journal of computer-aided molecular design **30**, 595 (2016).

[36] Rdkit: Open-source cheminformatics, `https://www.rdkit.org`.

[37] D. Weininger, Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules, J. Chem. Inf. Comput. Sci. **28**, 31 (1988).

[38] M. Rupp, A. Tkatchenko, K.-R. Müller, and O. A. von Lilienfeld, Fast and accurate modeling of molecular atomization energies with machine learning, Phys. Rev. Lett. **108**, 058301 (2012).

[39] M. J. Vainio and M. S. Johnson, Generating conformer ensembles using a multiobjective genetic algorithm, Journal of chemical information and modeling **47**, 2462 (2007).

[40] E. J. Baerends, O. V. Gritsenko, and R. van Meer, The kohn–sham gap, the fundamental gap and the optical gap: the physical meaning of occupied and virtual kohn–sham orbital energies, Phys. Chem. Chem. Phys. **15**, 16408 (2013).

[41] J. Hartmanis, Computers and intractability: a guide to the theory of np-completeness (michael r. garey and david s. johnson), Siam Review **24**, 90 (1982).

[42] K. Riesen and H. Bunke, Approximate graph edit distance computation by means of bipartite graph matching, Image and Vision computing **27**, 950 (2009).

[43] L. Jia, B. Gaüzère, F. Yger, and P. Honeine, A metric learning approach to graph edit costs for regression, in *Structural, Syntactic, and Statistical Pattern Recognition: Joint IAPR International Workshops, S+ SSPR 2020, Padua, Italy, January 21–22, 2021, Proceedings* (Springer, 2021) pp. 238–247.

[44] B. Gaüzère, S. Bougleux, and L. Brun, Approximating graph edit distance using gnccp, in *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)* (Springer, 2016) pp. 496–506.

[45] S. Bougleux, B. Gaüzère, and L. Brun, Graph edit distance as a quadratic program, in *2016 23rd International Conference on Pattern Recognition (ICPR)* (2016) pp. 1701–1706.

[46] D. B. Blumenthal, N. Boria, J. Gamper, S. Bougleux, and L. Brun, Comparing heuristics for graph edit distance computation, The VLDB Journal **29**, 419 (2020).

[47] H. Bunke and G. Allermann, Inexact graph matching for structural pattern recognition, Pattern Recognition Letters **1**, 245 (1983).

[48] N. S. Altman, An introduction to kernel and nearest-neighbor nonparametric regression, The American Statistician **46**, 175 (1992).

[49] K. P. Murphy, *Machine Learning: A Probabilistic Perspective* (MIT Press, 2012) Chap. 14.4.3, pp. 492–493.

[50] D. Haussler, *Convolution kernels on discrete structures*, Tech. Rep. (Technical report, Department of Computer Science, University of California at Santa Cruz, 1999).

[51] K. M. Borgwardt and H.-P. Kriegel, Shortest-path kernels on graphs, in *Data Mining, Fifth IEEE International Conference on* (IEEE, 2005) pp. 8–pp.

[52] F. Suard, A. Rakotomamonjy, and A. Bensrhair, Kernel on bag of paths for measuring similarity of shapes., in

ESANN (2007) pp. 355–360.

[53] B. Gaüzère, P.-A. Grenier, L. Brun, and D. Villemin, Treelet kernel incorporating cyclic, stereo and inter pattern information in chemoinformatics, Pattern Recognition **48**, 356 (2015).

[54] N. Shervashidze, P. Schweitzer, E. J. v. Leeuwen, K. Mehlhorn, and K. M. Borgwardt, Weisfeiler-lehman graph kernels, Journal of Machine Learning Research **12**, 2539 (2011).

[55] N. M. Kriege, F. D. Johansson, and C. Morris, A survey on graph kernels, Applied Network Science **5**, 1 (2020).

[56] K. Borgwardt, E. Ghisu, F. Llinares-López, L. O'Bray, and B. Rieck, Graph kernels: State-of-the-art and future challenges, arXiv preprint arXiv:2011.03854 (2020).

[57] S. Ghosh, N. Das, T. Gon**c**calves, P. Quaresma, and M. Kundu, The journey of graph kernels through two decades, Computer Science Review **27**, 88 (2018).

[58] L. Jia, B. Gaüzère, and P. Honeine, Graph kernels based on linear patterns: theoretical and experimental comparisons, Expert Systems with Applications **189**, 116095 (2022).

[59] B. Schölkopf and A. J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond* (MIT press, 2002).

[60] M. Fey and J. E. Lenssen, Fast graph representation learning with PyTorch Geometric, in *ICLR Workshop on Representation Learning on Graphs and Manifolds* (2019).

[61] L. Wu, P. Cui, J. Pei, L. Zhao, and X. Guo, Graph neural networks: foundation, frontiers and applications, in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (2022) pp. 4840–4841.

[62] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, A comprehensive survey on graph neural networks, IEEE Transactions on Neural Networks and Learning Systems (2020).

[63] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, Graph neural networks: A review of methods and applications, AI open **1**, 57 (2020).

[64] W. L. Hamilton, Graph representation learning, Synthesis Lectures on Artifical Intelligence and Machine Learning **14**, 1 (2020).

[65] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, Neural message passing for quantum chemistry, in *International conference on machine learning* (PMLR, 2017) pp. 1263–1272.

[66] T. N. Kipf and M. Welling, Semi-supervised classification with graph convolutional networks, arXiv preprint arXiv:1609.02907 (2016).

[67] M. Zhang, Z. Cui, M. Neumann, and Y. Chen, An end-to-end deep learning architecture for graph classification, in *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32 (2018).

[68] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, How powerful are graph neural networks?, arXiv preprint arXiv:1810.00826 (2018).

[69] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, *et al.*, Graph attention networks, stat **1050**, 10 (2017).

[70] Y. Shi, Z. Huang, S. Feng, H. Zhong, W. Wang, and Y. Sun, Masked label prediction: Unified message passing model for semi-supervised classification, arXiv preprint arXiv:2009.03509 (2020).

[71] G. Hoffmann, M. Balcilar, V. Tognetti, P. Héroux, B. Gaüzère, S. Adam, and L. Joubert, Predicting experimental electrophilicities from quantum and topological descriptors: a machine learning approach, Journal of Computational Chemistry **41**, 2124 (2020).

[72] The code for these ml experiments is available at the following repository: `https://github.com/jajupmochi/ged-cost-learn-framework/tree/master/`.