RIGR: Resonance Invariant Graph Representation for Molecular Property Prediction

Akshat Shirish Zalte,[†] Hao-Wei Pang,[†] Anna Doner,[†] and William H. Green[*,†,‡]

[†]*Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, MA, 02139, USA*

[‡]*MIT Energy Initiative, Massachusetts Institute of Technology, Cambridge, MA 02139, USA*

E-mail: whgreen@mit.edu

Abstract

Graph neural networks, which rely on Lewis structure representations, have emerged as a powerful tool for predicting molecular and reaction properties. However, a key limitation arises with molecules exhibiting resonance, where multiple valid Lewis structures represent the same species. This causes inconsistent predictions for the same molecule based on the chosen resonance form in common property prediction frameworks like Chemprop, which implements a directed message-passing neural network (D-MPNN) architecture on the input molecular graph. To address this issue of resonance variance, we introduce the Resonance Invariant Graph Representation (RIGR) of molecules that ensures, by construction, that all resonance structures are mapped to a single representation, eliminating the need to choose from or generate multiple resonance structures. Implemented with the D-MPNN architecture, RIGR is evaluated on a large dataset with resonance-exhibiting radicals and closed-shell molecules, comparing it against the Chemprop featurizer. Using 60% fewer features, RIGR demonstrates comparable or superior prediction performance. Alternative approaches, such as data augmentation

with resonance forms, are assessed, and their limitations are explored. Available open-source as an optional featurization scheme in Chemprop, RIGR is benchmarked across a wide range of property prediction tasks, showcasing its potential as a general graph featurizer beyond resonance handling.

1 Introduction

Machine learning (ML), especially deep learning, has become a powerful tool for accurately predicting molecular properties and chemical reactivity. Fast and accurate property prediction is of great interest to pharmaceutical and materials research, where it can significantly accelerate the discovery of novel drugs and materials. Deep learning models fundamentally seek to learn a mapping from the molecular input to the target property by optimizing its parameters. The choice of molecular representation is critical for capturing the useful relationship between the molecule and the target property to achieve generalizable models.

Many approaches have been developed to represent molecules for property prediction, utilizing a variety of molecular representations, such as graphs, strings, precomputed feature vectors like Morgan fingerprint,[1,2] and atomic coordinate sets.[3] While traditional methods often rely on fixed molecular representations crafted by experts, many recent works have used graph-based ML techniques that generate learned representations.[4–6] These models can learn two types of mappings: one from the input molecular graph to a latent embedding and one from the latent embedding to the final target property. State-of-the-art deep learning architectures for molecular property prediction, particularly directed message-passing neural networks (D-MPNNs), which are used in the open-source Chemprop software package,[7,8] have shown success in predicting various chemical properties, including solvation thermodynamics,[9–12] critical properties,[13] reaction barriers,[14–19] and infrared spectra,[20] among others. Learned representations have been shown to enhance performance, scalability, adaptability, robustness, and generalizability in various applications.[7,21]

The input to 2D graph neural networks (GNNs) for learning the molecular embedding is often a molecular graph. A Simplified Molecular Input Line Entry System (SMILES)[22,23] string is a convenient way to encode a Lewis structure in a linear text format, making it the standard input for most of the cheminformatics packages and chemical data sets. While Lewis structures are intuitive and easily encoded using SMILES, they only represent local-

ized electronic configurations. This limitation becomes apparent in the cases of electron delocalization, where a single Lewis structure cannot fully represent the molecule. The delocalization of electrons across multiple structures, commonly observed in organic chemistry, is defined as resonance or mesomerism. While multiple resonance structures can represent the same molecule, it is a single physical entity with a unique value for each property. In most machine learning packages for property prediction, such as Chemprop, if a different resonance structure of the same molecule is provided as input, it is interpreted as a different molecule, leading to varying predictions. This inconsistency introduces ambiguity for users who must choose which resonance form to pick for representing a species, with each form leading to different predictions. The number of choices is compounded for tasks involving chemical reactions, such as reaction enthalpy or rate-constant prediction, where both reactants and products may exhibit resonance, such as for resonance-stabilized radical reactions common in thermal or oxidative kinetics. One way to implicitly learn resonance invariance is by generating and including all possible resonance structures in the training data set with the same target values.[19] This data augmentation approach significantly increases training costs, but more critically, reliable generation of resonance structures is a complex task. There is no universal method for accurately producing these structures for all molecule types. It often requires manual effort to add custom resonance pathways or modify existing ones, which can be time-consuming and prone to errors. Some 3D GNNs, such as SchNet[24] and PhysNet,[25] which rely solely on atomic coordinates and identities, are invariant to different resonance forms. However, they incur high computational costs and depend on the availability of high-quality 3D data, which can be challenging to obtain for many molecules. As machine learning tools become integral to chemistry, there is a need to develop innovative methods to handle chemical resonance more effectively. To overcome these problems, we present Resonance Invariant Graph Representation (RIGR), a method for featurizing graphs in a resonance-invariant manner. RIGR ensures a consistent molecular representation by generating only the resonance-invariant features or descriptors of the input molecule. While

this concept can be applied to most deep learning packages for molecular property prediction, in this work, we focus on Chemprop due to its popularity in the community.

RIGR is adapted from Chemprop's native featurizer by eliminating resonance-variant features such as bond order and formal charge, which results in treating all resonance structures of a molecule as identical. The motivation behind RIGR stems from a fundamental understanding of quantum chemistry, which abstracts the concept of chemical bonds and relies on a set of atomic coordinates and a guess geometry to solve for electronic densities that are used to determine contributions to molecular properties. On the other hand, ML approaches utilize graphs with additional atom and bond features that provide extra information to assist the model. However, the extra features are not strictly necessary for representing the underlying molecule and can make the representation inaccurate for molecules with resonance. In this paper, we provide a comprehensive comparison of the chemistry-abiding RIGR against Chemprop's native featurizer for predicting the standard heat of formation on a dataset primarily consisting of resonance-active molecules, including both radicals and closed-shell species. We evaluate and compare three types of models: those trained with the RIGR featurizer (`RIGR`), those trained with the native featurizer (`Native`), and those trained using the native featurizer on training set augmented with multiple resonance structures (`Native + Aug`). We discuss challenges related to data augmentation and introduce metrics that quantify the variance in property prediction across different resonance structures and its effect on model performance. We ultimately establish RIGR as a suitable featurizer that is applicable across a wide variety of property prediction tasks, both with and without resonance.

2 Methods

We begin by describing the construction of the RIGR featurizer and the machine learning architecture used in this study. Following that, we discuss the details of data preparation

and augmentation, model training, and the Shapley value analysis.

2.1 RIGR Featurizer

RIGR is implemented as a featurizer in the recently released Chemprop v2.[26] We first give a brief overview of the Chemprop architecture. The input is a SMILES string, which is converted into a molecular graph, $\mathcal{G}(V, E)$, using RDKit,[27] where atoms are vertices V and bonds are edges E. Atoms and bonds are separately featurized using atom-level and bond-level features derived from their identity and topology. The architecture employs a directed message-passing neural network[7] (D-MPNN), where messages are passed between directed edges instead of between nodes as in traditional MPNNs. The learned atomic embeddings are aggregated into a molecular embedding, which can be optionally concatenated with additional molecule-level features. Finally, a feed-forward neural network (FFNN) predicts target molecular properties y using the molecular embedding as the input.

The implementation of RIGR involves modifying the graph featurization step within the Chemprop architecture. Only the atom and bond features that are independent of resonance are retained. In its simplest implementation, the RIGR featurizer is constructed by deleting the resonance-variant features without introducing any new ones.

Given $\mathcal{G}(V, E)$, for each vertex v, initial feature vectors $\{x_v \mid v \in V\}$ are derived from one-hot encodings of the atomic number, the number of bonds linked to each atom, the number of bonded hydrogens, and the atomic mass (scaled by dividing by 100). Notably, formal charge, hybridization, and aromaticity are excluded as they vary between resonance structures. For each edge e, initial feature vectors $\{e_{vw} \mid \{v, w\} \in E\}$ are based solely on whether the bond is part of a ring, omitting features like bond type and whether the bond is conjugated. The initial directed edge features e^d_{vw} are obtained by concatenating the atom features of the first atom in the bond x_v with the corresponding undirected bond features e_{vw}. For two different

resonance structures of the same molecule, featurized as $\mathcal{G}_{\text{rigr}}$ and $\mathcal{G}^*_{\text{rigr}}$, Equation (1) holds.

$$\mathcal{G}_{\text{rigr}} \cong \mathcal{G}^*_{\text{rigr}} \implies y(\mathcal{G}_{\text{rigr}}) = y(\mathcal{G}^*_{\text{rigr}}) \tag{1}$$

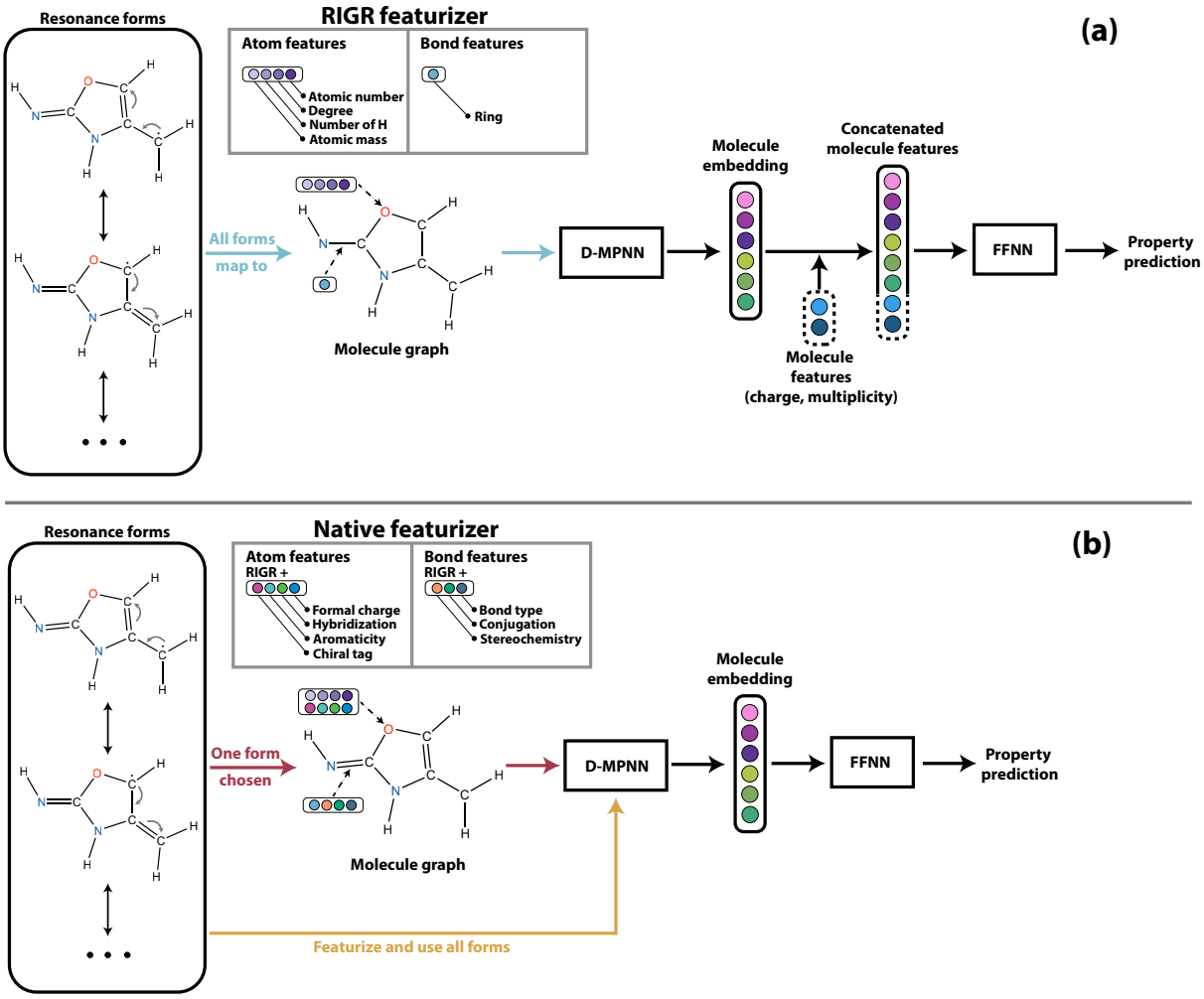

Figure 1: Schematic of the D-MPNN architectures used in this study. (a) The RIGR featurizer maps all resonance forms to a single graph representation. Additional molecular descriptors can be incorporated before the FFNN. (b) Baseline methods use the native featurizer, which includes additional features. In this case, either a single resonance form is selected (`Native`), or all forms are featurized and used for training (`Native + Aug`).

Since the formal charge is removed from the atom featurizer, a molecular-level feature for the net charge of the molecule can be added to distinguish free radicals from cationic or

anionic species. This addition is essential when molecules with a net charge, like ions, are present in the dataset. The net charge descriptor q_m is incorporated by concatenating it with the aggregated molecular embedding h_{agg} (as shown in Equation (2)). Figure 1 shows the Chemprop architecture using the RIGR and native featurizers.

$$h_m = \text{cat}(h_{agg}, q_m) \tag{2}$$

2.2 Data Preparation

In this section, we describe the datasets used in this study. We introduce the primary dataset and discuss the methods for augmenting the dataset with multiple resonance forms. Various other datasets are also incorporated to test the broader applicability of RIGR.

2.2.1 Primary Dataset

For this study, it is essential to use a dataset with a significant portion of resonance-exhibiting molecules to avoid conducting a mere ablation test of atom and bond features. We selected a subset of the QuantumPioneer[28] (QP) dataset generated in-house, which includes a large number of resonance-exhibiting radicals as well as closed-shell molecules. QP is among the largest and most diverse open-source quantum chemistry databases, containing over 350,000 small organic molecules with up to 21 heavy atoms, including elements like H, C, N, O, F, Si, P, S, Cl, Br, and I. The QP dataset provides molecular geometries and vibrational frequencies optimized at the ω-B97X-D/def2-SVP level of theory and single-point energies calculated at the DLPNO-CCSD(T)-F12a/def2-TZVP level of theory. The dataset also includes thermodynamic properties, such as enthalpies of formation, that are derived via advanced atom-energy and bond-additivity correction schemes.[29,30] For our analysis, we sampled 50,000 molecules, limiting our selection to those containing only H, C, N, and O atoms. Canonicalized SMILES were used to ensure that each molecule has a unique SMILES

representation. For species with equivalent resonance structures, such as benzene, canonicalized SMILES provide a single representation, effectively imposing a resonance invariant representation in these systems. For the target property, we selected standard heat of formation (ΔH_f°). The distribution of different species in the final dataset is shown in Table 1. There are no species with a net charge in the dataset. Additional data statistics are provided in Section S1 of the Supporting Information (SI).

Table 1: Number of molecules in each subset of the dataset.

Subset	Number of molecules
Resonance radical	20,000
Resonance closed shell	20,000
No resonance[a]	10,000
Total	50,000

[a] This also includes species with only equivalent resonance structures that can be represented using a single canonicalized SMILES.

2.2.2 Data Augmentation

Augmenting the data with all possible resonance structures for each species provides an implicit way to learn resonance invariance, which has been shown to improve performance on tasks such as barrier height prediction for radical reactions.[19] This approach involves generating a set of representative non-equivalent resonance structures for each molecule and assigning them the same target value. While several automated methods for chemical resonance generation exist, each has its limitations. For instance, RDKit[27] has a native resonance generation function, `Chem.ResonanceMolSupplier`, but it is not designed for radicals, which comprise a large part of our dataset.

A more suitable method was developed by Grinberg Dana et al.[31], which efficiently generates representative resonance structures for a wide range of chemical species, including radicals and biradicals, consisting of elements H, C, O, N, and S. This method accounts

for both localized resonance pathways, which apply to two- and three-atom systems, and global approaches for aromatic species. However, the number of localized structures generated can be excessive due to the combinatorial nature of resonance pathways. Grinberg Dana's method also includes filtering the generated structures by using the octet rule and formal charge heuristics to obtain a more representative selection of resonance forms. The algorithm has been implemented in both the Reaction Mechanism Generator[32,33] (RMG) and the Reaction Data and Molecular Conformer[34] (RDMC) software packages. We chose RMG as it has better aromaticity determination, an important factor given the prevalence of aromatic molecules in our dataset. While the generation of resonance molecules is generally accurate, we encountered issues when converting them into SMILES strings, particularly for heteroatomic aromatic radicals in the dataset. The algorithm implemented in RMG sometimes generated resonance SMILES that corresponded to multi-radical species. To resolve this, we leveraged the fact that the spin multiplicity is conserved across different resonance structures of the same molecule. We used the `fix_mol` function from RDMC to fix the resonance forms by saturating bi-radicals and carbene systems to ensure that the spin multiplicity corresponding to all resonance SMILES is the same as the input molecule. Finally, all the SMILES strings are canonicalized to remove the redundant equivalent resonance forms. Figure 2 shows the distribution of resonance structures for resonance-exhibiting radicals and closed-shell molecules. As expected, most of the species with a large number of resonance forms are radicals, not closed-shell molecules.

2.2.3 Other Datasets

To evaluate the broader applicability of the RIGR featurizer beyond its resonance invariance capabilities, we conducted comparative tests across a wide range of tasks using models trained with both RIGR and the native Chemprop featurizer. Chemprop v2.0.3 has previously been benchmarked on various regression and classification tasks.[26] We replicated all the benchmarks using RIGR to identify any performance differences between the two. Table 2

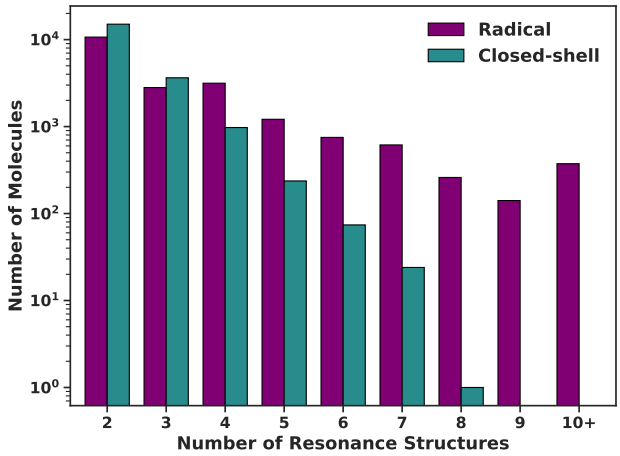

Figure 2: Distribution of the number of non-equivalent resonance forms generated using our method for radicals and closed-shell species in the dataset. The maximum number of resonance structures for any molecule in the dataset is 24.

summarizes the datasets employed in this benchmarking study.

Table 2: Summary of the benchmarking datasets.

Dataset/Category	Property/Data type	Type	N tasks	N data	Metric(s)[a]	Ref.[b]
MoleculeNet & OGB	HIV (HIV replication inhibition)	Class.	1	41,127	ROC-AUC	35
	PCBA (biological activities)	Class.	128	437,929	PRC-AUC, AP	35,36
	QM9 (DFT calculated properties)	Regr.	12	133,885	MAE, RMSE	35,37
SAMPL	logP	Regr.	1	23,469[c]	RMSE	38,39
Reaction barrier heights	E2	Regr.	1	1264	MAE	15,16,40,41
	S_N2	Regr.	1	2361	MAE	15,16,40,41
	Cycloaddition	Regr.	1	5269	MAE	42
	RDB7	Regr.	1	23,852[d]	MAE	43
	RGD1-CNHO	Regr.	1	353,984[d]	MAE	44
UV/Vis Absorption	UV/Vis peak absorption wavelength	Regr.	1	26,395	MAE, RMSE, R^2	45–48
PCQM4MV2	HOMO-LUMO gap	Regr.	1	3,452,151	MAE, RMSE	49

[a] The metric(s) originally reported in the Chemprop paper.[8]
[b] References for the data and data splits.
[c] The size of the training set. The SAMPL6, SAMPL7, and SAMPL9 data are used as a test set.
[d] Reverse reactions are included, ensuring that each forward-reverse pair is assigned to the same set, train, validation, or test.

2.3 Model Training

This study spans three types of models: those trained with the RIGR featurizer (`RIGR`), those trained with the native featurizer (`Native`), and those trained with the native featurizer alongside data augmentation (`Native + Aug`). Figure 1 shows the D-MPNN architecture used for training the models. In this section, we first outline the data splitting methods,

then describe the training and evaluation of models on these splits, and introduce new metrics to assess model performance in the context of resonance.

2.3.1 Data Splits

To test interpolative performance, we used random splitting to allocate 80% of the data to training, 10% to validation, and 10% to testing. While random splits are common in the literature, they primarily assess performance on relatively simple interpolation tasks.[50] However, a major application of property prediction models is to predict properties for molecules and reactions that are very different from those already studied. Therefore, evaluating model performance on more challenging extrapolative tasks is crucial to assess generalizability.[51] While scaffold splitting[52] is a common approach, it is unsuitable for our dataset as most molecules lack assigned scaffolds. Instead, we used a K-Means clustering-based method to create chemically dissimilar training, validation, and test sets.[19] We first generated 2048-bit Morgan fingerprints[1,2] with a radius of 4 for each molecule using RDKit,[27] then applied principal component analysis (PCA) to reduce the dimensionality of these sparse vectors. K-Means clustering is then used to group the molecules into 18 clusters, with each cluster assigned to one of the training, validation, and test sets to achieve an approximate 80:10:10 split. A detailed description for the method used for generating extrapolative cluster splits is provided in Section S2 of the SI. The clusters are randomly shuffled between the training and validation sets five times while keeping the test set constant to generate five splits. For models trained on datasets augmented with resonance structures, the training, validation, and test sets are augmented independently to prevent data leakage. We assigned weights in the augmented training set that are inversely proportional to the number of resonance structures, ensuring that molecules with more resonance structures are not unfairly prioritized.

2.3.2 Ensemble Training and Hyperparameter Optimization

For both random and K-Means splitting, we trained an ensemble model for each split with five different weight initializations and averaged the predictions for all molecules across the ensemble. We report the average performance metric across the five splits, using the standard deviation to estimate the uncertainty in the reported metric. To compare the performance of models trained with the RIGR and native featurizers, we performed a two-tailed paired t-test with a significance level α of 0.05 to assess any significant differences. For comparisons involving more than two models, pairwise t-tests are conducted with a Bonferroni correction to control for familywise error rate (FWER).[53] More information about model comparisons using statistical tests is given in Section S4.1 of the SI.

Hyperparameter optimization was performed without ensembling on a single data split. During tuning, we optimized the number of message-passing steps, the hidden size of the message-passing layers, the number of layers, the hidden size of the feed-forward neural network, and the dropout ratio. Both tuning and production models utilized summation to aggregate atomic features into molecular feature vectors, with the optimized hyperparameters summarized in Table S1. Furthermore, molecular graphs with explicit hydrogens were used as input for all models to clearly distinguish between keto and enol tautomers when employing the RIGR featurizer.

To evaluate the impact of data set size on model performance, we downsampled the training and validation sets to create training sets containing 200, 500, 1,000, 2,000, 5,000, 10,000, and 20,000 unique molecules. For the downsampled datasets, we used a single data split for both random and K-Means splitting and trained five-ensemble models without any hyperparameter re-optimization.

2.3.3 Resonance Metrics

For the test set augmented with multiple resonance SMILES, we introduce two additional metrics: Resonance Range (RR), which captures the variation in predicted values across different resonance structures of the same molecule, and Maximum Resonance Deviation (MRD), which quantifies the maximum deviation between a single resonance structure and the ground truth. We penalized the outliers using root-mean-square (RMS) versions of these metrics, with the mathematical equations provided in Equation (3) and Equation (4), where X is the target property, N is the number of molecules, and j counts over the resonance forms.

$$\text{RMSRR(X)} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\max_{j}(X_{i,j}^{pred}) - \min_{j}(X_{i,j}^{pred})\right)^2} \tag{3}$$

$$\text{RMSMRD(X)} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\max_{j}\left(X_{i,j}^{true} - X_{i,j}^{pred}\right)\right)^2} \tag{4}$$

From Equation (3), it is evident that for models trained using RIGR, the value of RMSRR would be zero by design, as all resonance forms will have the same prediction (Equation (1)). The RMSMRD metric for an augmented test set will always be greater than or equal to the RMSE calculated on the test set without augmentation, with equality holding for RIGR. This metric is intended to capture the worst-case performance of a model calculated by selecting the resonance form with the maximum deviation from the true value. This metric assesses the model robustness for resonance-related uncertainties, quantifying how much the choice of resonance form can affect the overall performance.

2.4 Shapley Analysis

Shapley values are one of the most widely used approaches for model explainability, offering insights into how individual features contribute to the final output of an ML model.[54–59]

This method assesses the relative impact of each input feature by comparing its contribution to the overall prediction, relative to an average baseline prediction. In this work, we utilized Chemprop v2's implementation of Shapley value analysis, which is based on the popular SHAP (SHapley Additive exPlanations) Python package and builds on the work by Li et al.[60], who performed Shapley analysis for additional quantum mechanical descriptors. This analysis enabled us to quantify the marginal contribution of individual atom and bond features to the final prediction. For further theoretical background on SHAP, we refer readers to the foundational work by Lundberg and Lee.[61]

We conducted the Shapley analysis using the primary dataset without any data augmentation. This analysis was performed for ML models trained on 80% of the dataset, using both the full native featurizer and the RIGR featurizer. To compute the Shapley values for each molecule in the test set, we used the `PermutationExplainer` from SHAP to calculate the expected Shapley value of each feature, sampling $1,000$ different combinations of included and excluded features during inference. This process was repeated for every molecule, and we then averaged the absolute Shapley values of each feature across all molecules to determine the expected average effect of each feature on the model's output. Since we have an ensemble of five models, we report averaged results to account for variations between different model initializations. Finally, we assessed the relative importance of the features by comparing the magnitudes of their corresponding Shapley values for the models trained using native and RIGR featurizers.

3 Results and Discussion

In this section, we first discuss the performance of RIGR against key baselines on the primary dataset, followed by a summary of the benchmarking results across all the other datasets. A Shapley value analysis is also presented to understand feature importance in Chemprop.

3.1 Detailed Comparison between RIGR and Baselines

We first compare the models trained with RIGR and native featurizers on the unaugmented test set to gauge the impact of feature reduction on model performance for both interpolative and extrapolative tasks. Root mean square error (RMSE) is used as the evaluation metric as it penalizes the outliers to give a realistic estimate of model performance. Figure 3 presents the results for the K-Means split, demonstrating that there is no performance loss; in fact, `RIGR` consistently outperforms `Native` across all subsets of the test set, including species that do not exhibit resonance. The paired t-test confirms that the difference in performance between models is statistically significant (RMSE and p-values tabulated in Table S3). The trends are identical for the random split, with similar test RMSE indicating that the model trained using RIGR generalizes well.

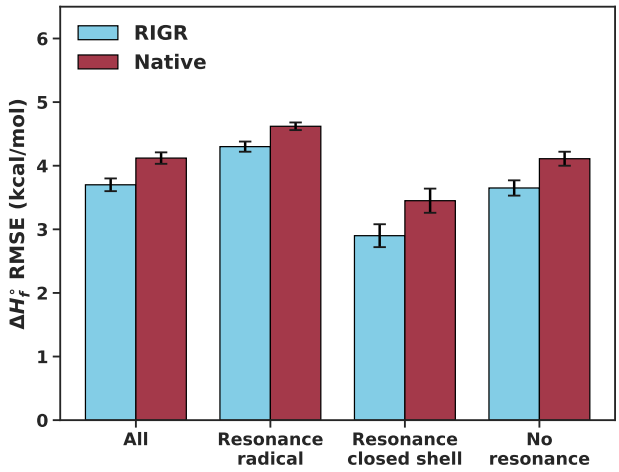

Figure 3: Results showing the test set RMSE values for the entire test set and its subsets, comparing models trained using RIGR and native featurizers on K-Means split. The RMSE is reported as the mean across five folds, with error bars representing the standard deviation. `RIGR` significantly outperforms `Native` across all subsets.

For a comprehensive comparison between the `RIGR`, `Native`, and `Native + Aug` models and to quantify the effect of resonance on model performance, we tested these models on a test set augmented with multiple resonance forms. In addition to RMSE, we used RMSRR and RMSMRD (described in Section 2.3) metrics to assess both resonance variance and overall model performance. Figure 4 shows the performance on the K-Means split, and the trends

are similar for the random split (results tabulated in Table S4).

For each metric on every subset, a pairwise comparison of the three models was conducted to assess statistical significance using the paired t-test with Bonferroni correction. Firstly, `Native` consistently performs significantly worse than both `RIGR` and `Native + Aug` across all subsets, considering all metrics. Based on the RMSRR values for `Native`, the average range of predictions for the same radical with multiple resonance structures is around 3 kcal mol^{-1}, with the maximum RMSRR being greater than 25 kcal mol^{-1}. The substantial difference in predicted enthalpy, depending on which resonance structure is chosen as the input, reveals the inconsistency in predictions caused by resonance and underscores the necessity of RIGR. Comparing `Native` with `Native + Aug` shows that training on augmented data improves performance across all three metrics. `Native + Aug` achieves a slightly lower RMSE than `RIGR` on the entire set; however, this difference is not statistically significant. A similar trend is observed for RMSMRD, where `RIGR` and `Native + Aug` show comparable values, both significantly lower than `Native`. Notably, RMSMRD provides a more pessimistic error estimate than RMSE and weighs each molecule in the test set equally. While data augmentation reduces the RMSRR for both resonance radicals and closed-shell molecules, the models are not fully resonance invariant. By design, the RMSRR for `RIGR` is 0, whereas `Native + Aug` fails to achieve near-zero values, even with the large dataset. In summary, using RIGR as a featurizer without augmentation or employing the full native featurizer with augmentation leads to similar improvements in test RMSE. While `RIGR` is fully resonance invariant, `Native + Aug`, despite resulting in lower RMSRR and RMSMRD compared to `Native`, remains resonance-variant due to its implicit handling of resonance, in contrast to the more rigorous approach of `RIGR`.

While augmenting the dataset with multiple resonance structures does improve consistency and performance, it is important to note that its implementation can be challenging, depending on the types of chemical species in the dataset. This process may require meticulously

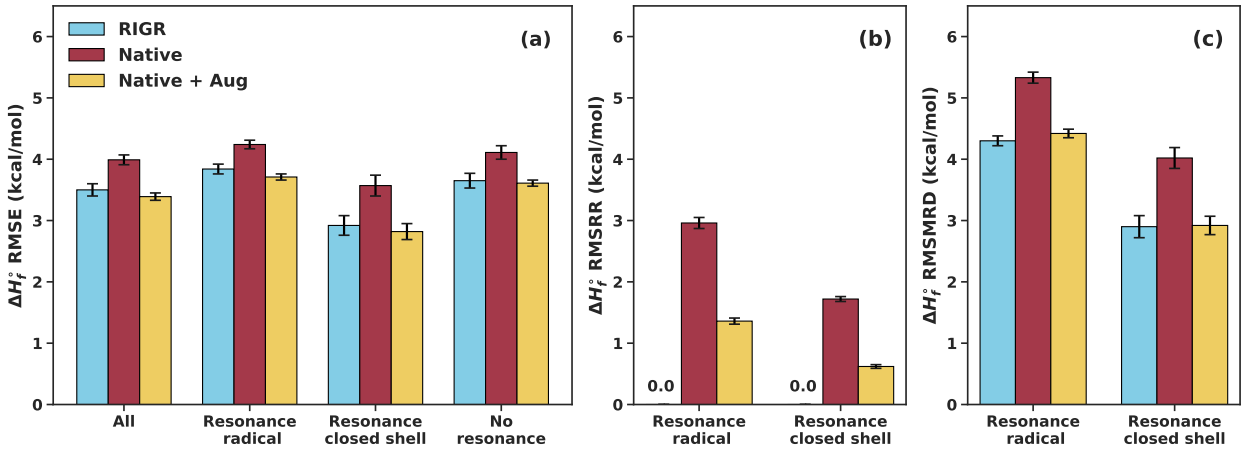

Figure 4: Comparison of `RIGR`, `Native` and `Native + Aug` models trained on K-Means split showing augmented test set (a) RMSE, (b) RMSRR, and (c) RMSMRD values. Each metric is reported as the mean across five folds, with error bars representing the standard deviation. In terms of RMSE and RMSMRD, `RIGR` and `Native + Aug` outperform `Native`, whereas RMSRR is superior (equal to zero) for `RIGR`.

modifying existing algorithms or extending them by incorporating additional templates and filters to obtain a representative set of valid resonance structures, particularly for species with elements beyond C, H, O, and N. For example, the resonance generation algorithm used in this work is inadequate for species with a net charge, such as ions. Even if generating resonance forms were assumed universally feasible, the training costs could become significantly higher depending on the prevalence of resonance-active species in the dataset. For the training set with 40,000 unique molecules and roughly 2.5 resonance structures per molecule, using RIGR instead of performing data augmentation leads to a 56% reduction in training time and a 42% improvement in maximum RAM usage while training. Table 3 shows the exact data for these training cost metrics. Therefore, in addition to being consistent with our knowledge of chemistry, RIGR offers a much easier alternative to data augmentation, eliminating the need for additional resonance generation and data pruning, thereby saving time during both data processing and model training.

The results from the downsampled training sets, as depicted in Figure 5, exhibit trends consistent with those observed for the largest training set. The RMSRR and RMSMRD

Table 3: Training cost benchmark results using the default Chemprop v2 hyperparameters, with training conducted for 100 epochs. The training set contains 40,000 unique molecules. A local cluster equipped with an RTX 4090 24GB GPU was utilized for the training.

Training time (s)		Max. RAM usage (GB)	
RIGR	Native + Aug	RIGR	Native + Aug
641	1460	**2.16**	3.75

were calculated for all resonance-active species in the test set, with detailed results for the random split provided in the SI. The goal of this downsampling was to test whether a threshold exists for the minimum number of molecules required for a model trained with a reduced featurizer to perform comparably to one trained with the full featurizer. These additional features could potentially help the model maintain similar performance as the size of the training set decreases. However, based on the results, `RIGR` consistently outperforms `Native`, even when the training set includes as few as 200 molecules.

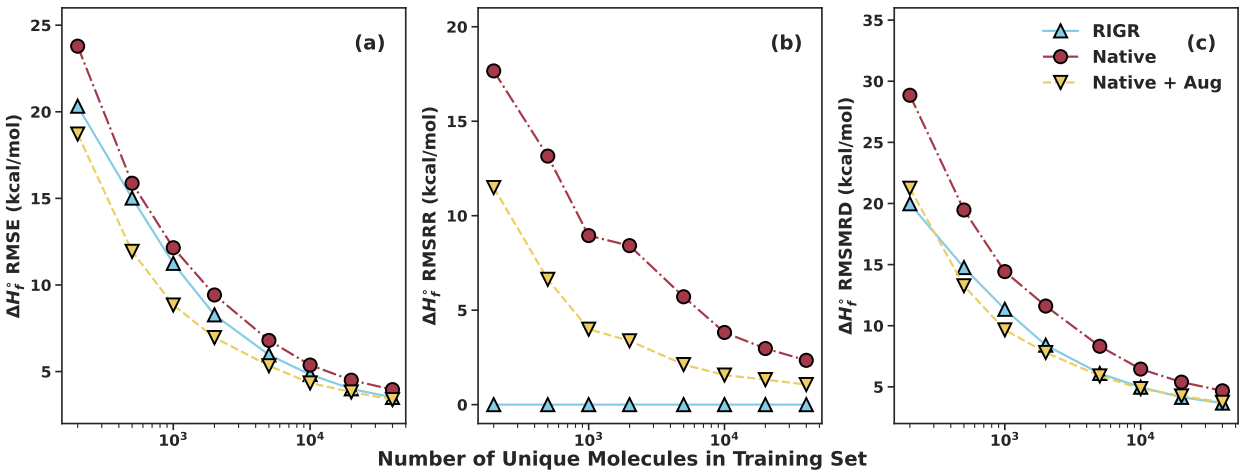

Figure 5: Plots showing the augmented test set (a) RMSE, (b) RMSRR, and (c) RMSMRD values for `RIGR`, `Native` and `Native + Aug` models trained on K-Means split downsampled subsets. The x-axis denotes the number of unique molecules in the training set, which is different from the actual training set size for `Native + Aug`. The performance loss in RMSE and RMSMRD due to decreasing training set size for `RIGR` is similar to other models.

For RMSE, `Native + Aug` outperforms `RIGR` on smaller datasets, but when the training set includes more than 5,000 molecules, both models show comparable performance. The

RMSRR results indicate that data augmentation increasingly reduces resonance variance as the dataset size increases. The trends for RMSMRD show similar performance for `RIGR` and `Native + Aug`, both significantly outperforming `Native`. In conclusion, there appears to be no lower limit on dataset size for using `RIGR`, as no critical information seems to be lost through the feature reduction employed by RIGR, for this dataset. This is further validated by using Shapley value analysis for the two featurizers, discussed in Section 2.4.

3.2 Shapley Value Analysis

The Shapley value analysis of the model trained with the native featurizer reveals that some features contribute far more significantly than others. Figure 6 shows the results from the analysis for `Native` and `RIGR` models on K-Means split. The results for the random split are presented in Section S4.3 of the SI.

Atom-level features such as aromaticity and bond-level features like conjugation insignificantly contribute to the final prediction of the model, making their absence in the RIGR featurizer inconsequential. Interestingly, based on the Shapley analysis of the native featurizer, some of the most important features, like hybridization, bond type, and formal charge, are absent in RIGR. However, models trained using RIGR perform equally well and have a very similar overall distribution of predictions. This result suggests a redundancy of information inherent in Chemprop's native featurizer and the Lewis structures it is based on. The essential information from the features omitted in RIGR is indirectly encoded by a combination of atom identity (atomic number and mass), degree, number of attached hydrogen atoms, and the overall charge and multiplicity of the molecule. This becomes more evident from the Shapley value analysis of models trained using RIGR featurizer. The contributions from the features removed to create RIGR are simply redistributed among the remaining features. RIGR can potentially serve as a general featurizer beyond its resonance invariance capabilities. Models with fewer, more relevant features can maintain or even improve predic-

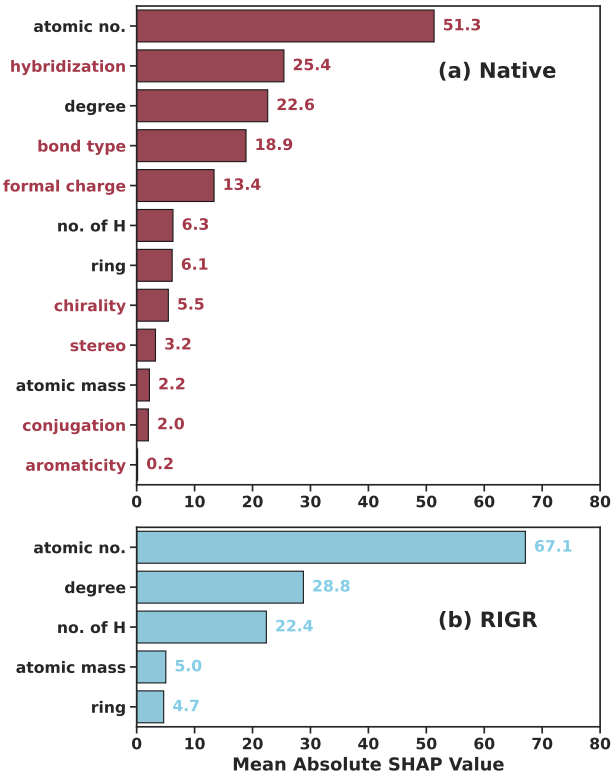

Figure 6: Histogram showing mean absolute Shapley values for models trained on K-Means split using (a) Native and (b) RIGR featurizers. The values are reported for a single split, averaged over 5 model initializations. Features removed from the native featurizer are shown in red, with their contribution compensated by the remaining features in RIGR to maintain overall performance.

tive accuracy while simplifying the interpretation of results.[62] RIGR offers this alternative for molecular property prediction.

3.3 RIGR beyond Resonance

We next evaluate RIGR's potential as a general featurizer by testing it on a range of molecular property datasets that were used for benchmarking Chemprop. A summary of the datasets used for this analysis is provided in Table 2. For general usage, RIGR incorporates net charge as a molecule-level descriptor. The same training procedure and hyperparameter tuning, as implemented for benchmarking the performance of Chemprop v2, were applied to train models with RIGR to ensure a fair comparison. The results show a consistent per-

formance between the models trained with RIGR and native featurizer across all property prediction benchmarks. Table 4 and Table 5 present some of the results from this benchmarking, highlighting the performance similarity between `RIGR` and `Native`. Other results are tabulated in Table S6, Table S7, and Table S8 and the optimized hyperparameters for each task are listed in Table S9 under Section S4.4 of the SI. All the results show that the models trained with RIGR exhibit performance comparable to those using the native featurizer.

Table 4: Comparison of model performance on the test set, trained with RIGR and Native featurizer, for regression tasks, including UV/Vis peak absorption wavelength, logP for the SAMPL6 challenge, barrier height of organic reactions (S_N2 and RGD1-CNHO datasets), and thermochemical properties in the QM9 dataset including Internal energy at 0 K (U0) and Enthalpy at 298.15 K (H298).

	RIGR			Native		
	MAE	RMSE	R^2	MAE	RMSE	R^2
UV/Vis	15.9	30.6	0.915	16.9	31.1	0.913
SAMPL6	0.32	0.39	0.656	0.34	0.46	0.525
S_N2	2.53	3.48	0.921	2.62	3.48	0.921
RGD1-CNHO[a]	6.19	10.40	0.877	5.75	9.60	0.896
QM9 (U0)[b]	1.08	2.43	1.000	1.02	2.39	1.000
QM9 (H298)[c]	1.75	3.11	1.000	1.90	3.26	1.000

[a] Bond stereochemistry and atom chirality are included in the featurizer.
[b] Individual training.
[c] Multi-task training.

Table 5: Comparison of model performance on the test set, trained with RIGR and Native featurizer, for classification task on HIV and PCBA datasets. Both models are trained on random splits.

	RIGR			Native		
	ROC-AUC	PRC-AUC	AP	ROC-AUC	PRC-AUC	AP
HIV	0.8045	0.3090	0.3114	0.7713	0.3048	0.3067
PCBA	0.9027	0.2068	0.2127	0.9085	0.2146	0.2200

4 Conclusion

In this work, we introduced RIGR, a 2D molecular graph featurizer that, by construction, ensures all resonance forms of a chemical species share the same representation. RIGR achieves this by using only the resonance-invariant subset of atom and bond features from the Chemprop package. Despite utilizing 60% fewer features, RIGR matches or even outperforms native Chemprop on several property prediction tasks, including on smaller datasets. We present a detailed comparison between models trained using RIGR and those trained with the native Chemprop featurizer, both with and without augmenting the data with resonance forms. While data augmentation was found to be useful, in many cases, generating a valid set of resonance structures is complicated and error-prone. RIGR offers a convenient and fast solution for ensuring resonance invariance and is demonstrated to have value as a general graph featurizer for wider applications, with robust performance on various property prediction tasks.

5 Data and Software Availability

RIGR is open source on GitHub, with a detailed user guide available at `https://github.com/akshatzalte/chemprop/tree/rigr_home`. The repository includes all scripts, datasets, data splits, best hyperparameters, and model files needed to reproduce the results in this paper. RIGR is available as an option for the multi-hot atom featurization scheme in Chemprop version 2.1.2 or above. Example scripts and Jupyter notebooks are also provided. The primary dataset, with a detailed description, is available on Zenodo, along with the corresponding data splits, at `https://zenodo.org/records/14942335`. The datasets used for benchmarking are the same as Chemprop v2 and can be accessed via Zenodo at `https://zenodo.org/records/10078142`.

6 Supporting Information

The supporting information includes additional dataset analysis, a detailed procedure for creating extrapolative data splits, statistical significance tests with corresponding p-values, and the hyperparameter settings for all Chemprop models used in this study.

7 Acknowledgements

This project was funded by BASF under MIT award number 88803720. A.S.Z. gratefully acknowledges fellowship support from MathWorks. Additional financial support for this project was provided by the Machine Learning for Pharmaceutical Discovery and Synthesis Consortium (MLPDS) at MIT. We thank Prof. Connor Coley, Angiras Menon, Shih-Cheng Li, Xiaorui Dong, Sayandeep Biswas, and Haoyang Wu for their valuable discussions on this work. We also appreciate Jackson Burns and Nathan Morgan for their contributions to implementing RIGR as a featurizer in Chemprop. Additionally, we thank Kariana Moreno Sader for her immense efforts in improving the clarity and quality of the figures in this research paper. Finally, the authors thank Prof. Markus Kraft for his thorough review and feedback on the manuscript.

References

(1) Morgan, H. L. The Generation of a Unique Machine Description for Chemical Structures – A Technique Developed at Chemical Abstracts Service. *Journal of Chemical Documentation* **1965**, *5*, 107–113.

(2) Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *Journal of Chemical Information and Modeling* **2010**, *50*, 742–754.

(3) Pattanaik, L.; Coley, C. W. Molecular Representation: Going Long on Fingerprints. *Chem* **2020**, *6*, 1204–1207.

(4) Wu, Z.; Ramsundar, B.; Feinberg, E. N.; Gomes, J.; Geniesse, C.; Pappu, A. S.; Leswing, K.; Pande, V. MoleculeNet: A Benchmark for Molecular Machine Learning. *Chemical Science* **2018**, *9*, 513–530.

(5) Struble, T. J.; Coley, C. W.; Jensen, K. F. Multitask Prediction of Site Selectivity in Aromatic C–H Functionalization Reactions. *Reaction Chemistry & Engineering* **2020**, *5*, 896–902.

(6) Guo, J.; Sun, M.; Zhao, X.; Shi, C.; Su, H.; Guo, Y.; Pu, X. General Graph Neural Network-Based Model to Accurately Predict Cocrystal Density and Insight from Data Quality and Feature Representation. *Journal of Chemical Information and Modeling* **2023**, *63*, 1143–1156.

(7) Yang, K.; Swanson, K.; Jin, W.; Coley, C.; Eiden, P.; Gao, H.; Guzman-Perez, A.; Hopper, T.; Kelley, B.; Mathea, M. Analyzing Learned Molecular Representations for Property Prediction. *Journal of Chemical Information and Modeling* **2019**, *59*, 3370–3388.

(8) Heid, E.; Greenman, K. P.; Chung, Y.; Li, S.-C.; Graff, D. E.; Vermeire, F. H.; Wu, H.;

Green, W. H.; McGill, C. J. Chemprop: A Machine Learning Package for Chemical Property Prediction. *Journal of Chemical Information and Modeling* **2023**, *64*, 9–17.

(9) Boobier, S.; Hose, D. R. J.; Blacker, A. J.; Nguyen, B. N. Machine Learning with Physicochemical Relationships: Solubility Prediction in Organic Solvents and Water. *Nature Communications* **2020**, *11*, 5753.

(10) Fowles, D. J.; Palmer, D. S.; Guo, R.; Price, S. L.; Mitchell, J. B. O. Toward Physics-Based Solubility Computation for Pharmaceuticals to Rival Informatics. *Journal of Chemical Theory and Computation* **2021**, *17*, 3700–3709.

(11) Chung, Y.; Vermeire, F. H.; Wu, H.; Walker, P. J.; Abraham, M. H.; Green, W. H. Group Contribution and Machine Learning Approaches to Predict Abraham Solute Parameters, Solvation Free Energy, and Solvation Enthalpy. *Journal of Chemical Information and Modeling* **2022**, *62*, 433–446.

(12) Vermeire, F. H.; Chung, Y.; Green, W. H. Predicting Solubility Limits of Organic Solutes for a Wide Range of Solvents and Temperatures. *Journal of the American Chemical Society* **2022**, *144*, 10785–10797.

(13) Biswas, S.; Chung, Y.; Ramirez, J.; Wu, H.; Green, W. H. Predicting Critical Properties and Acentric Factors of Fluids Using Multitask Machine Learning. *Journal of Chemical Information and Modeling* **2023**, *63*, 4574–4588.

(14) Grambow, C. A.; Pattanaik, L.; Green, W. H. Deep Learning of Activation Energies. *The Journal of Physical Chemistry Letters* **2020**, *11*, 2992–2997.

(15) Heid, E.; Green, W. H. Machine Learning of Reaction Properties via Learned Representations of the Condensed Graph of Reaction. *Journal of Chemical Information and Modeling* **2022**, *62*, 2101–2110.

(16) Heinen, S.; von Rudorff, G. F.; von Lilienfeld, O. A. Toward the Design of Chemical Reactions: Machine Learning Barriers of Competing Mechanisms in Reactant Space. *The Journal of Chemical Physics* **2021**, *155*, 064105.

(17) Jorner, K.; Brinck, T.; Norrby, P.-O.; Buttar, D. Machine Learning Meets Mechanistic Modelling for Accurate Prediction of Experimental Activation Energies. *Chemical Science* **2021**, *12*, 1163–1175.

(18) Spiekermann, K. A.; Pattanaik, L.; Green, W. H. Fast Predictions of Reaction Barrier Heights: Toward Coupled-Cluster Accuracy. *The Journal of Physical Chemistry A* **2022**, *126*, 3976–3986.

(19) Spiekermann, K. A.; Dong, X.; Menon, A.; Green, W. H.; Pfeifle, M.; Sandfort, F.; Welz, O.; Bergeler, M. Accurately Predicting Barrier Heights for Radical Reactions in Solution Using Deep Graph Networks. *The Journal of Physical Chemistry A* **2024**, *128*, 8384–8403.

(20) McGill, C.; Forsuelo, M.; Guan, Y.; Green, W. H. Predicting Infrared Spectra with Message Passing Neural Networks. *Journal of Chemical Information and Modeling* **2021**, *61*, 2594–2609.

(21) Aldeghi, M.; Coley, C. W. A Graph Representation of Molecular Ensembles for Polymer Property Prediction. *Chemical Science* **2022**, *13*, 10486–10498.

(22) Weininger, D. SMILES, a Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules. *Journal of Chemical Information and Computer Sciences* **1988**, *28*, 31–36.

(23) Weininger, D.; Weininger, A.; Weininger, J. L. SMILES. 2. Algorithm for Generation of Unique SMILES Notation. *Journal of Chemical Information and Computer Sciences* **1989**, *29*, 97–101.

(24) Schütt, K. T.; Sauceda, H. E.; Kindermans, P.-J.; Tkatchenko, A.; Müller, K.-R. SchNet – A Deep Learning Architecture for Molecules and Materials. *The Journal of Chemical Physics* **2018**, *148*, 241722.

(25) Unke, O. T.; Meuwly, M. PhysNet: A Neural Network for Predicting Energies, Forces, Dipole Moments, and Partial Charges. *Journal of Chemical Theory and Computation* **2019**, *15*, 3678–3693.

(26) Greenman, K. P.; Graff, D.; Morgan, N.; Zheng, J.; Pang, H.-W.; Li, S.-C.; Zalte, A.; Wu, O.; Burns, J.; Doner, A.; Menon, A.; Coley, C.; Green, W. H. Chemprop v2.0.3. `https://github.com/chemprop/chemprop`, 2024.

(27) RDKit: Open-Source Cheminformatics. `https://www.rdkit.org`, Accessed: 2024-10-09.

(28) Wu, H.; Pang, H.-W.; Dong, X.; Burns, J. W.; Spiekermann, K.; Menon, A.; Zheng, J.; Biswas, S.; Li, S.-C.; Green, W. Quantumpioneer: Self-Evolving Machine for High-Throughput Automated Potential Energy Surface Exploration and Closed-Loop Chemical Reactivity Discovery. Proceedings of the AIChE 2023 Annual Meeting. 2023.

(29) Wu, H.; Payne, A. M.; Pang, H.-W.; Menon, A.; Grambow, C. A.; Ranasinghe, D. S.; Dong, X.; Grinberg Dana, A.; Green, W. H. Toward Accurate Quantum Mechanical Thermochemistry: (1) Extensible Implementation and Comparison of Bond Additivity Corrections and Isodesmic Reactions. *The Journal of Physical Chemistry A* **2024**, *128*, 4335–4352.

(30) Wu, H.; Doner, A. C.; Pang, H.-W.; Green, W. H. Toward Accurate Quantum Mechanical Thermochemistry: (2) Optimal Methods for Enthalpy Calculations from Comprehensive Benchmarks of 284 Model Chemistries. 2025; Preprint, `https://chemrxiv.org/engage/chemrxiv/article-details/67953191fa469535b9d8f826`.

(31) Grinberg Dana, A.; Liu, M.; Green, W. H. Automated Chemical Resonance Generation and Structure Filtration for Kinetic Modeling. *International Journal of Chemical Kinetics* **2019**, *51*, 760–776.

(32) Liu, M.; Grinberg Dana, A.; Johnson, M. S.; Goldman, M. J.; Jocher, A.; Payne, A. M.; Grambow, C. A.; Han, K.; Yee, N. W.; Mazeau, E. J.; Blondal, K.; West, R. H.; Goldsmith, C. F.; Green, W. H. Reaction Mechanism Generator v3.0: Advances in Automatic Mechanism Generation. *Journal of Chemical Information and Modeling* **2021**, *61*, 2686–2696.

(33) Johnson, M. S.; Dong, X.; Grinberg Dana, A.; Chung, Y.; Farina, D. J.; Gillis, R. J.; Liu, M.; Yee, N. W.; Blondal, K.; Mazeau, E.; Grambow, C. A.; Payne, A. M.; Spiekermann, K. A.; Pang, H.-W.; Goldsmith, C. F.; West, R. H.; Green, W. H. RMG Database for Chemical Property Prediction. *Journal of Chemical Information and Modeling* **2022**, *62*, 4906–4915.

(34) Dong, X.; Pattanaik, L.; Li, S.-C.; Spiekermann, K.; Pang, H.-W.; Green, W. H. RDMC: Reaction Data and Molecular Conformer Software Package, version 0.1.0. `https://github.com/xiaoruiDong/RDMC`, 2023.

(35) Wu, Z.; Ramsundar, B.; Feinberg, E. N.; Gomes, J.; Geniesse, C.; Pappu, A. S.; Leswing, K.; Pande, V. MoleculeNet: A Benchmark for Molecular Machine Learning. *Chemical Science* **2018**, *9*, 513–530.

(36) Hu, W.; Fey, M.; Zitnik, M.; Dong, Y.; Ren, H.; Liu, B.; Catasta, M.; Leskovec, J. Open Graph Benchmark: Datasets for Machine Learning on Graphs. 2021; Preprint, `https://arxiv.org/abs/2005.00687`.

(37) Ramakrishnan, R.; Dral, P. O.; Rupp, M.; Von Lilienfeld, O. A. Quantum Chemistry Structures and Properties of 134 kilo Molecules. *Scientific Data* **2014**, *1*, 140022.

(38) The SAMPL Challenges. `https://github.com/samplchallenges`, Accessed: 2024-08-02.

(39) Koscher, B.; Canty, R. B.; McDonald, M. A.; Greenman, K. P.; McGill, C. J.; Bilodeau, C. L.; Jin, W.; Wu, H.; Vermeire, F. H.; Jin, B. Autonomous, Multi-Property-Driven Molecular Discovery: From Predictions to Measurements and Back. *Science* **2023**, *382*, eadi1407.

(40) von Rudorff, G. F.; Heinen, S. N.; Bragato, M.; von Lilienfeld, O. A. Thousands of Reactants and Transition States for Competing E2 and S2 Reactions. *Machine Learning: Science and Technology* **2020**, *1*, 045026.

(41) Stuyver, T.; Coley, C. W. Quantum Chemistry-Augmented Neural Networks for Reactivity Prediction: Performance, Generalizability, and Explainability. *The Journal of Chemical Physics* **2022**, *156*, 084104.

(42) Stuyver, T.; Jorner, K.; Coley, C. W. Reaction Profiles for Quantum Chemistry-Computed [3+2] Cycloaddition Reactions. *Scientific Data* **2023**, *10*, 66.

(43) Spiekermann, K.; Pattanaik, L.; Green, W. H. High Accuracy Barrier Heights, Enthalpies, and Rate Coefficients for Chemical Reactions. *Scientific Data* **2022**, *9*, 417.

(44) Zhao, Q.; Vaddadi, S. M.; Woulfe, M.; Ogunfowora, L. A.; Garimella, S. S.; Isayev, O.; Savoie, B. M. Comprehensive Exploration of Graphically Defined Reaction Spaces. *Scientific Data* **2023**, *10*, 145.

(45) Joung, J. F.; Han, M.; Jeong, M.; Park, S. Experimental Database of Optical Properties of Organic Compounds. *Scientific Data* **2020**, *7*, 1–6.

(46) Ju, C.-W.; Bai, H.; Li, B.; Liu, R. Machine Learning Enables Highly Accurate Predictions of Photophysical Properties of Organic Fluorescent Materials: Emission Wave-

lengths and Quantum Yields. *Journal of Chemical Information and Modeling* **2021**, *61*, 1053–1065.

(47) Venkatraman, V.; Raju, R.; Oikonomopoulos, S. P.; Alsberg, B. K. The Dye-Sensitized Solar Cell Database. *Journal of Cheminformatics* **2018**, *10*, 1–9.

(48) Venkatraman, V.; Kallidanthiyil Chellappan, L. An Open Access Data Set Highlighting Aggregation of Dyes on Metal Oxides. *Data* **2020**, *5*, 45.

(49) Nakata, M.; Shimazaki, T. PubChemQC Project: A Large-Scale First-Principles Electronic Structure Database for Data-Driven Chemistry. *Journal of Chemical Information and Modeling* **2017**, *57*, 1300–1308.

(50) Spiekermann, K. A.; Stuyver, T.; Pattanaik, L.; Green, W. H. Comment on 'physics-based representations for machine learning properties of chemical reactions'. *Machine Learning: Science and Technology* **2023**, *4*, 048001.

(51) Bradshaw, J.; Zhang, A.; Mahjour, B.; Graff, D. E.; Segler, M. H.; Coley, C. W. Challenging reaction prediction models to generalize to novel chemistry. 2025; Preprint, `https://arxiv.org/abs/2501.06669`.

(52) Bemis, G. W.; Murcko, M. A. The Properties of Known Drugs. 1. Molecular Frameworks. *Journal of Medicinal Chemistry* **1996**, *39*, 2887–2893.

(53) Goeman, J. J.; Solari, A. Comparing Three Groups. *The American Statistician* **2022**, *76*, 168–176.

(54) Pelegrina, G. D.; Siraj, S. Shapley Value-Based Approaches to Explain the Quality of Predictions by Classifiers. *IEEE Transactions on Artificial Intelligence* **2024**, *5*, 4217–4231.

(55) Beechey, D.; Smith, T. M. S.; Şimşek, Explaining Reinforcement Learning with Shapley

Values. Proceedings of the 40th International Conference on Machine Learning. 2023; pp 2003–2014.

(56) Zheng, Q.; Wang, Z.; Zhou, J.; Lu, J. Shap-CAM: Visual Explanations for Convolutional Neural Networks Based on Shapley Value. Computer Vision – ECCV 2022. 2022; pp 459–474.

(57) Dewi, C.; Tsai, B.-J.; Chen, R.-C. Shapley Additive Explanations for Text Classification and Sentiment Analysis of Internet Movie Database. Recent Challenges in Intelligent Information and Database Systems. 2022; pp 69–80.

(58) Mastropietro, A.; Pasculli, G.; Feldmann, C.; Rodríguez-Pérez, R.; Bajorath, J. EdgeSHAPer: Bond-centric Shapley Value-based Explanation Method for Graph Neural Networks. *iScience* **2022**, *25*, 105043.

(59) Rodríguez-Pérez, R.; Bajorath, J. Interpretation of Machine Learning Models Using Shapley Values: Application to Compound Potency and Multi-Target Activity Predictions. *Journal of Computer-Aided Molecular Design* **2020**, *34*, 1013–1026.

(60) Li, S.-C.; Wu, H.; Menon, A.; Spiekermann, K. A.; Li, Y.-P.; Green, W. H. When Do Quantum Mechanical Descriptors Help Graph Neural Networks to Predict Chemical Properties? *Journal of the American Chemical Society* **2024**, *146*, 23103–23120.

(61) Lundberg, S.; Lee, S.-I. A Unified Approach to Interpreting Model Predictions. 2017; Preprint, `http://arxiv.org/abs/1705.07874`.

(62) Guyon, I.; Elisseeff, A. An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research* **2003**, *3*, 1157–1182.

TOC Graphic