Invariant Features for Accurate Predictions of Quantum Chemical UV-vis Spectra of Organic Molecules

Justin Baker
University of Utah
baker@math.utah.edu

Massimiliano Lupo Pasini
Oak Ridge National Laboratory
lupopasinim@ornl.gov

Cory Hauck
Oak Ridge National Laboratory
hauckc@ornl.gov

Abstract

Including invariance of global properties of a physical system as an intrinsic feature in graph neural networks (GNNs) enhances the model's robustness and generalizability and reduces the amount of training data required to obtain a desired accuracy for predictions of these properties. Existing open source GNN libraries construct invariant features only for specific GNN architectures. This precludes the generalization of invariant features to arbitrary message passing neural network (MPNN) layers which, in turn, precludes the use of these libraries for new, user-specified predictive tasks. To address this limitation, we implement invariant MPNNs into the flexible and scalable HydraGNN architecture. HydraGNN enables a seamless switch between various MPNNs in a unified layer sequence and allows for a fair comparison between the predictive performance of different MPNNs. We trained this enhanced HydraGNN architecture on the ultraviolet-visible (UV-vis) spectrum of GDB-9 molecules, a feature that describes the molecule's electronic excitation modes, computed with time-dependent density functional tight binding (TD-DFTB) and available open source through the GDB-9-Ex dataset. We assess the robustness (i.e., accuracy and generalizability) of the predictions obtained using different invariant MPNNs with respect to different values of the full width at half maximum (FWHM) for the Gaussian smearing of the theoretical peaks. Our numerical results show that incorporating invariance in the HydraGNN architecture significantly enhances both accuracy and generalizability in predicting UV-vis spectra of organic molecules.

1 Introduction

Scientific relevance. The ultraviolet-visible (UV-vis) spectrum of an organic molecule refers to the pattern of ultraviolet and visible light that is absorbed or transmitted by the molecule [1]. When electromagnetic radiation in the UV and visible regions (typically in the range of 200 to 800 nm) passes through a sample of the organic compound, certain wavelengths of light are absorbed by the molecule, while others are transmitted or pass through unaffected. This fact provides the foundation of a powerful analytical technique used in chemistry to identify and characterize organic compounds, particularly those with conjugated systems or chromophores (parts of the molecule responsible for absorption of light). Predicting the UV-vis spectrum of compounds plays a crucial role in biotechnology, pharmaceutics, and energy research. More specifically, the UV-vis spectrum can be used to (1) identify chemical species present in a material or produced during a chemical reactions, (2) grow bacteria for biological studies, (3) perform drug design by identifying molecules

Preprint. Preliminary work.

that are easy to engage in chemical reactions, or (4) quantify the amount of nucleic acids (DNA or RNA) to determine their average concentrations in a mixture, as well as their purity.

State-of-the-art deep learning approaches. State-of-the-art methods for measuring a molecule's UV-vis spectrum involve labor-intensive experiments or computationally expensive first-principle calculations [2, 3]. These methods limit efficient screening of vast parameter spaces for molecular design. Recently, deep learning (DL) models have emerged as a promising alternative for predicting the UV-vis spectrum of organic molecules.

DL models excel at capturing complex data trends, essential for mapping a molecule's atomic structure to its UV-vis spectrum. They also offer rapid predictions compared to experiments and first-principle calculations, enabling efficient exploration of parameter spaces for desired optical properties. Some prominent DL architectures, like DeepMD [4], Behler-Parinello Neural Network (BPNN) [5], and DTNN [6], require costly atomic input descriptors, limiting their scalability for large atomic systems. In contrast, graph neural networks (GNNs) [7] extract topological information naturally by treating atoms as nodes and bonds as edges, eliminating the need for expensive descriptors and allowing scalable training on large atomic systems. Open-source GNN libraries like PyTorch Geometric (PyG) [8, 9], Deep Graph Library (DGL) [10–12], and TorchMD [13] lack crucial features for computational chemistry, including multi-task learning (MTL), seamless model replacement, hyperparameter control, distributed data parallelism (DDP), and regular maintenance.

An Oak Ridge National Laboratory (ORNL) open-source GNN implementation, HydraGNN [14, 15], addresses these gaps. HydraGNN supports MTL [16], is open-source[1], offers an object-oriented interface for message passing policies, provides HPC capabilities for scaling with DDP on supercomputers [17], and maintains compatibility with required software packages through continuous integration (CI) tests. However, the current HydraGNN version does not support MPNNs that preserve critical topological properties, like invariance, crucial for quantum chemistry applications. Invariance ensures a model yields consistent output despite certain input data transformations. For molecular applications, GNN predictions of total system properties (e.g., the UV-vis spectrum) should be invariant to rotations and translations to ensure robustness and generalization across structurally similar, unseen molecules.

Our contribution. We implement invariant message passing layers into the HydraGNN framework, enabling accurate and scalable learning for molecular properties. In particular, we (1) include positional inputs and user specified invariant embeddings into HydraGNN, (2) incorporate the invariant embeddings and message passing policies of the SchNet [18], DimeNet++, and EGNN and (3) provide a study on the robustness of the invariant features using the UV-vis spectrum of the open-source GDB-9-Ex dataset.

The HydraGNN architecture enhances each invariant model by allowing for a generalized encoding and decoding mechanism. The expansion of the HydraGNN architecture to include invariant models improves the flexibility and seamless MPNN switching by allowing for user specified positional inputs and invariant feature embeddings. By being included in the HydraGNN architecture, the invariant models can inherently take advantage of the built-in distributed computing capabilities of HydraGNN to ensure effective scaling on leadership class supercomputing facilities.

Our numerical results show the benefit of the new invariant features for improved predictions of the UV-vis spectra by training the HydraGNN model on the open source GDB-9-Ex dataset [19]. We assess the generalizability of the HydraGNN predictions with respect to variations in the Gaussian smearing applied to the theoretical peaks of the quantum chemical UV-vis spectra. Ensuring robustness with respect to different intensities of Gaussian smearing is a desirable property of surrogate models for UV-vis spectroscopy, as different experimentalists may need to tune such parameters based on the resolution and physical conditions of operability of the experimental instruments. In this respect, our study is the first of its kind.

Organization of the paper. The paper is organized as follows: In Section 2, we present the architecture of HydraGNN as well as several invariant GNNs. In Section 3, we provide a detailed review of the related works. In Section 4, we present the spectrum prediction task of the GDB-9-Ex
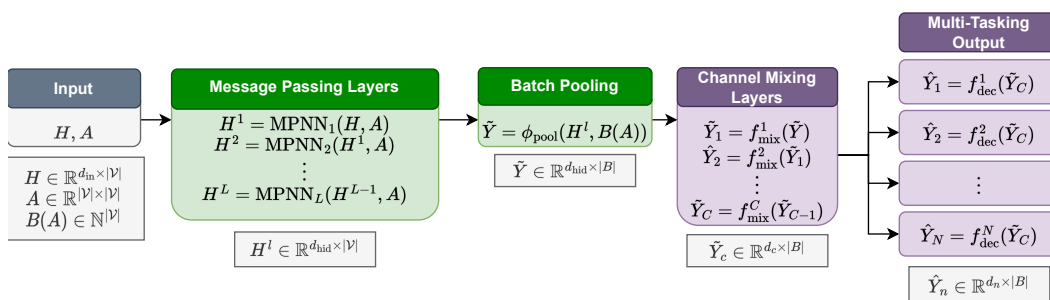
---

[1]https://github.com/ORNL/HydraGNN

Figure 1: Fixed GNN architecture for HydraGNN with input features $\boldsymbol{H}$ and adjacency matrix $\boldsymbol{A}$. Graph convolutional layers are shown in **green** and channel mixing layers are shown in **purple**.
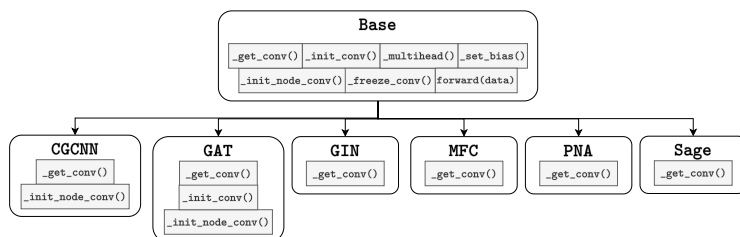

Figure 2: The object oriented programming structure of HydraGNN, in which each class inherits all of the functions from the `Base` class, and over-ride the selected functions that allow for the full functionality of the MPNN.

dataset. In Section 5, we detail the incorporation of invariant MPNNs into HydraGNN. In Section 6, we illustrate the advantage of using invariant features in the HydraGNN framework.

2 Background

Notation. We denote scalars by lower-case letters, vectors with lower-case boldface letters and matrices with upper-case boldface letters. For a vector $\boldsymbol{a}$, we use $\|\boldsymbol{a}\|$ and $\|\boldsymbol{a}\|_\infty$ to denote its ℓ_2 and ℓ_∞ norms, respectively. We denote vector concatenation as $\boldsymbol{a} \oplus \boldsymbol{b}$. For a matrix $\boldsymbol{A}$, we denote its induced ℓ_2 norm by $\|\boldsymbol{A}\|$. Given matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, we denote their entry-wise product as $\boldsymbol{A} \odot \boldsymbol{B}$.

2.1 HydraGNN architecture

The HydraGNN architecture is implemented in an object oriented programming structure which combines message passing and channel mixing layers for multi-task predictions of molecular properties.

Figure 1 illustrates the forward pass of the HydraGNN architecture. For graph structured input data $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we denote the number of nodes as $|\mathcal{V}|$ and the number of edges $|\mathcal{E}|$. The input data consists of node features $\boldsymbol{H} \in \mathbb{R}^{h_{\text{input}} \times |\mathcal{V}|}$ and graph connectivity matrix $\boldsymbol{A}$. The input features and their dimension h_{input} is determined by the task. For our application, each data batch contains $|B|$ molecules, and a molecule indicator function $B(\boldsymbol{A})$. The inputs are fed through L-message passing layers which update the hidden features $\boldsymbol{H}^l$ with hidden dimension $h_{\text{mpnn}} \times |\mathcal{V}|$ at each layer. The hidden features are then extracted from the atomic nodes to the molecular batch using a batch pooling function. A further C-layers of channel mixing are applied to update the internal features $\tilde{Y}^c$ with layer dependent feature dimensions $h_c \times |B|$. The final predictions are made in a multi-task format with N outputs of task dependent dimension $h_n \times |B|$. The non-task dependent layer count and feature dimensions $\{L, C, h_{\text{mpnn}}, h_c\}$ are chosen via hyperparameters.

Figure 2 details the object oriented programming structure utilized by each of the MPNN models in HydraGNN. The inheritance of the Base class for each MPNN enables the seamless change between message passing policies. It also enables a direct comparison between MPNNs in a unified framework of the Base class.

3

2.2 Invariant message passing

In an *invariant* MPNN, the internal features $\boldsymbol{H}^l$ are unchanged when subjected to a group action. Atomic structured data augments the internal features with Euclidean positional data $\boldsymbol{X}^l$. The group responsible for preserving Euclidean distances in a 3D space is the Euclidean group known as E(3). The E(3) group action encompasses all rotations, reflections and translations in the three dimensional space. Specifically, we say that an MPNN is E(3)-invariant if it satisfies the following

$$\text{MPNN}(\boldsymbol{H}^l, \boldsymbol{X}^l) = \text{MPNN}(\boldsymbol{H}^l, \boldsymbol{Q}\boldsymbol{X}^l + \boldsymbol{b}) \tag{1}$$

for any orthogonal matrix $\boldsymbol{Q} \in \mathbb{R}^{3\times3}$ and any vector $\boldsymbol{b} \in \mathbb{R}^3$.

To preserve invariance in the message passing policy, E(3)-invariant MPNNs use invariant feature embeddings to construct invariant features $\boldsymbol{f}_{ij}$ between two nodes, and use the invariant features in the construction of the message. An feature embedding is invariant if it satisfies $\boldsymbol{f}_{ij} = \phi_{\text{inv}}(\boldsymbol{X}) = \phi_{\text{inv}}(\boldsymbol{Q}\boldsymbol{X}+\boldsymbol{b})$. The message passing policy uses the invariant feature embedding to generate messages $\boldsymbol{m}_{ij}^l$ and update the hidden feature $\boldsymbol{h}_i^l$ at layer l as follows

$$\boldsymbol{f}_{ij} = \phi_{\text{inv}}(\boldsymbol{x}_i^l, \boldsymbol{x}_j^l), \quad \boldsymbol{m}_{ij}^l = \phi_e(\boldsymbol{h}_i^l, \boldsymbol{h}_j^l, \boldsymbol{f}_{ij}, \boldsymbol{e}_{ij})$$
$$\boldsymbol{m}_i^l = \bigoplus_{j\in\mathcal{N}(i)} \boldsymbol{m}_{ij}^l \quad \boldsymbol{h}_i^{l+1} = \phi_h(\boldsymbol{h}_i^l, \boldsymbol{m}_i^l) \tag{2}$$

where $\boldsymbol{h}_i^l$ is the i-th column of $\boldsymbol{H}^l$, $\boldsymbol{e}_{ij}$ are the edge attributes for edge $(i,j) \in \mathcal{E}$, $\bigoplus$ is a permutation invariant differentiable operator, $\mathcal{N}(i)$ is the neighborhood of i determined by the edges $\mathcal{E}$ and ϕ_e, ϕ_h are learnable functions parametrized by the neural network.
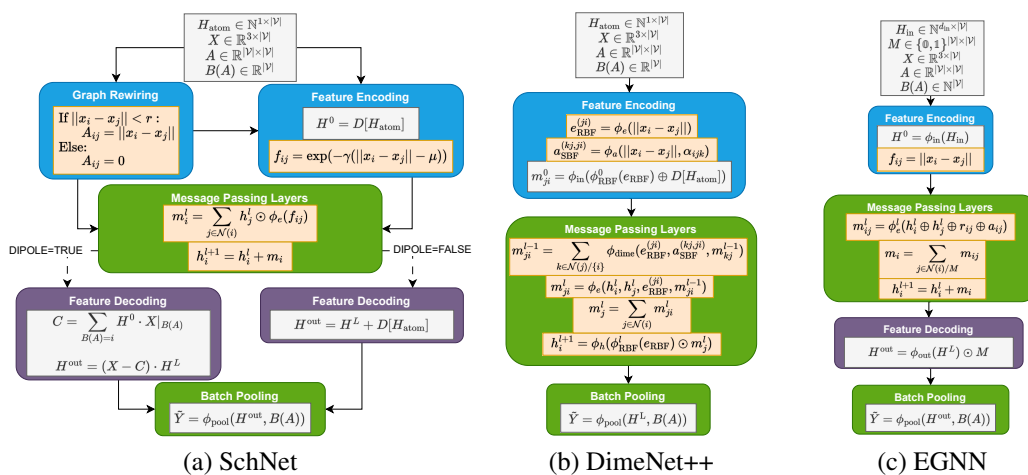

Figure 3: Procedural illustration of each invariant GNN model. The components highlighted in yellow are the essential capabilities of each invariant model that have been retained in their respective re-implementations in HydraGNN.

2.3 Invariant GNN models

Three state-of-the-art invariant GNN models with message passing policies are SchNet[18], DimeNet++[20] and EGNN[21]. Figure 3 illustrates each GNN architecture and a rigorous definition of the message passing policy is given in Appendix A.

SchNet. The SchNet[18] model uses distance based graph rewiring, and a parametrized Gaussian filter to construct invariant features $\boldsymbol{f}_{ij}$. The feature encoding of SchNet also uses a parametrized dictionary D to encode the atomic numbers of each atom, denoted $\boldsymbol{H}_{\text{atom}}$. The message passing policy of SchNet can be seen in Figure 3(a) where the edge features $\boldsymbol{f}_{ij}$ are used in each layer. The feature decoding is regulated by the DIPOLE hyperparameter and is task-specific. The SchNet model is limited in its adaptability to diverse feature inputs and outputs due to the atomic number encoding and task-specific decoding processes.

4

DimeNet++. The DimeNet++ [20] model directly updates the messages using directional message passing. The invariant features of DimeNet++ are constructed via radial basis and spherical basis functions. These embeddings use both the distance between two nodes, as well as the angles α_{ijk} among all triples of nodes to encapsulate directional information. Like SchNet, the feature encoding uses a dictionary parametrized encoding of the atomic numbers. Unlike SchNet, DimeNet++ [20] uses the invariant features to generate the initial feature encoding. The message passing policy of DimeNet++ is outlined in Figure 3(b). The DimeNet++ model directly updates the messages using the aggregation of directional messages which are generated by the parametrized function $\phi_{\text{dime}}(\cdot)$. The DimeNet++ model is limited in its adaptability to diverse feature inputs due to the atomic number encoding. It also is limmited in expressivity due to its lack of a feature decoding processes.

EGNN. The invariant EGNN[21] model is identical to the equivariant EGNN model, except that it does not perform positional updates. EGNN directly uses the distance between nodes as invariant features. Unlike SchNet and DimeNet, EGNN uses a flexible feature encoding. The message passing policy shown in Figure 3(c) is dependent on an edge mask M due to the model requiring all molecules to be zero-padded to the same length. This is the major limmiting factor for EGNN, as zero-padding is known to introduce numerical artifacts and boundary effects[22] and also increases memory usage and computational overhead.

3 Related Work

DL for UV-vis Spectroscopy. In [23], the authors compared the predictive performance of three different DL architectures, namely multilayer perceptrons (MLPs), convolutional neural networks (CNNs), and deep tensor neural networks (DTNNs) by represting the input molecular structures with a Coulomb matrix, which is an $N \times N$ square matrix where N denotes the number of atoms in the molecule. Since MLPs can process only data with fixed input size, the Coulomb matrices have been zero padded to ensure constant size of the molecule's representation across the entire dataset. Because of the zero padding, the MLP training may incur numerical artifacts and use an excessive memory storage. CNN and DTNN architectures can seamlessly handle Coulomb matrices of varying size, by treating the Coulomb matrix as a pixeled image. Although the accuracy and memory storage for CNNs and DTNNs improved with respect to MLPs, their performance does not scale well with the number of atoms in the molecule because they treat the Coulomb matrix as a dense matrix and do not take advantage of sparsity in the connectivity. GNN models are effective in reducing memory storage and training costs by representing molecular structures as graphs, which leverage the inherent sparsity in graph connectivity. In a study referenced in [24], various GNN models, employing MPNN layers, were evaluated for their ability to predict the UV-vis spectra of GDB-9 molecules, and their performance was compared to the DTNN model. The findings indicated that SchNet, a specific GNN variant, outperformed all other models considered in the study. However, there are two noteworthy aspects that warrant further consideration. Firstly, the study used electronvolts (eV) as the unit of measurement for quantifying the energy states of the UV-vis spectrum. In contrast, computational chemists typically prefer nanometers (nm) as the unit of measurement for quantifying UV-vis spectrum wavelengths. This choice facilitates a clearer distinction between various electronic excitation modes. Although energy and wavelength are inversely proportional, the conclusions drawn from a deep learning model's performance assessment using 'eV' may not directly apply to an analysis employing 'nm', as elaborated in section 4. Secondly, it's important to note that the quantum chemistry dataset used in the study is not available as an open-source resource. This limitation raises concerns about the reproducibility of the presented results.

Invariant Feature Learning for GNNs. Neural networks designed to achieve invariance have been introduced in previous works such as Tensorfield Networks (TFNs) [25], and similar ideas have been extended in the covariant Cormorant [26], NequIP [27], and SE(3)-Transformer [28] models, incorporating attention mechanisms into their structures. This family of models ensures invariance by mapping the original data structure, defined in the spatial domain, onto a functional space defined by spherical harmonic functions. Other similar approaches map the spatial domain to higher-dimensional spaces, like Lie group spaces, as seen in [29]. However, this mapping to high-dimensional functional spaces is computationally intensive and memory-intensive. To address this computational complexity and memory requirement, recent research introduced Steerable E(3) Equivariant Graph Neural Networks (SEGNN) [30]. SEGNN employs non-linear invariant features to capture local node interactions, enabling a reduction in the graph construction radius. This results in a sparser graph

5

representation, significantly reducing computational training time and memory usage. An alternative research direction suggests implementing invariant operations directly in the original Cartesian space. This approach efficiently maintains invariance, avoiding costly high-dimensional feature construction. This alternative approach's effectiveness is supported by various contributions in the literature, including E(n)-GNN [21], invariant GNN with attention mechanisms [31], GVP [32, 33], PaiNN [34], ET-Transformer [35], and graph attention transformers [36] architectures. For accurate modeling of complex, large atomic structures (e.g., proteins with 10k atoms), these models often require maintaining high graph connectivity, increasing training computational costs significantly. To tackle this challenge, Han et al. [37] introduced Equivariant Graph Hierarchy-based Network (EGHN), which identifies local input system substructures and establishes hierarchical connectivity between graph nodes. Fine-scale hierarchy captures local interactions among neighboring nodes, while coarse-scale hierarchy models long-range interactions among node clusters. The development of all these models has been motivated by different scientific needs, and this resulted in validating each model with respect to different datasets and/or target properties, which does not allow for a practical identification of what model could perform well on a new dataset. Moreover, the open-source implementations of these models are hardly comparable to each other because they each require a very specific format for the input features. This precludes the flexible handling of an arbitrary customizable set of features according to the user's needs.

4   Dataset: GDB-9-Ex

The SMILES strings of GDB-9 molecules [38] have been converted into 3D Cartesian coordinates, and geometry optimization has been performed, first by using the inexpensive Merck force field MMF94 implemented in RDKit, and then by refining the geometry optimization using static density functional tight-binding (DFTB). The optimized atomic structure of each molecule has been used as input to excited states calculations to compute the first 50 electronic excitation modes and oscillator strengths of the UV-vis spectrum. More quantum chemical details about the computational approach performed to compute the UV-vis spectrum are provided in [39].

Gaussian smearing.   Experimental measurements of the UV-vis spectrum can be affected by various factors, including precision of the instrument, temperature effects, and the finite size of the sampling intervals used during data collection. All these factors contribute to a broadening of the energy levels of the measured UV-vis spectrum. First-principles calculations are typically performed at absolute zero temperature, where the electrons occupy the lowest energy states available. This results into a calculated UV-vis spectrum characterized by a set of sharp theoretical peaks, each associated with a discrete energy excitation gap. The discrepancy between the physical conditions under which the UV-vis spectrum is measured with experiments and quantum chemical approaches does not allow for a direct comparison between experiments and simulations. Gaussian smearing, also known as Gaussian broadening or Gaussian convolution, is a common technique applied to the quantum chemical UV-vis spectrum (computed with first-principle calculations) to mimic the broadening of experimental measurements. By convoluting the sharp theoretical peaks with Gaussian functions, the peaks in the quantum chemical UV-vis spectrum are spread out and become broader. The degree of broadening applied to the theoretical peaks is often selected according to the instrumental resolution and experimental conditions to maximize the resemblance between the theoretical peaks and the experimental broadened peaks. The degree of the broadening is either controlled by the standard deviation σ of the Gaussian function, or by the Full Width at Half Maximum (FWHM), which is the width of the Gaussian function at half of its maximum height. The FWHM and σ are related by the formula $\text{FWHM} = 2\sigma\sqrt{2\ln(2)}$.

Electronvolts vs. nano-meters to represent the UV-vis spectrum.   The excitation states of a molecule described in the UV-vis spectrum can either be described in electronvolts (eV) or nano-meters (nm). The measure unit 'eV' quantifies the energy of an excited state of an orbital, whereas the measure unit 'nm' quantifies the wavelength of the excited state. Wavelength and energy of an excited state of a molecule are related by the formula $E = \hbar\nu = \hbar\frac{c}{\lambda}$, where E is the energy, $\hbar$ is the Planck's constant, ν is the frequency, c is the speed of light, and λ is the wavelength. Since GDB-9 molecules are relatively small in size (they contain at most nine non-hydrogen atoms), they are very stable (and therefore very difficult to excite) from a chemical standpoint. Therefore, most of the theoretical peaks of the quantum chemical UV-vis spectrum for each molecule are concentrated in a narrow 'eV' region in the high energy range. The Gaussian smearing condenses the majority of the
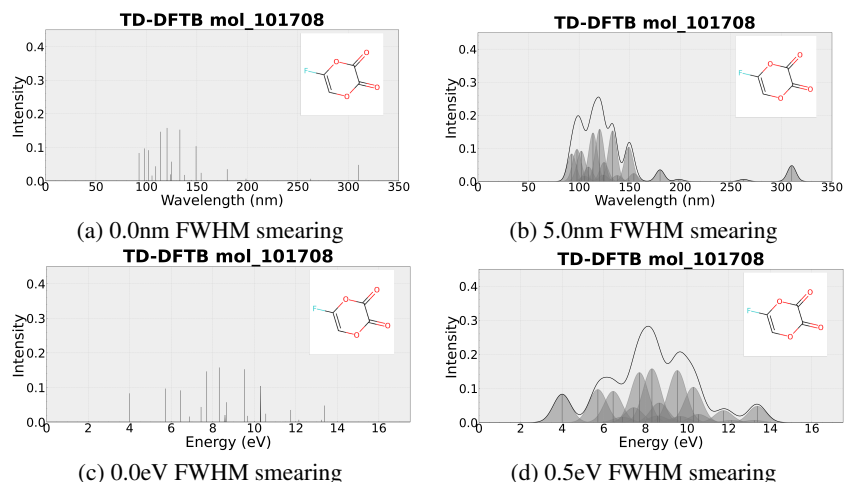
6

(a) 0.0nm FWHM smearing

(b) 5.0nm FWHM smearing

(c) 0.0eV FWHM smearing

(d) 0.5eV FWHM smearing

Figure 4: Electronic excitation spectrum of C_4HFO_4 plotted in wavelength(top) and energy(bottom) with exact peaks (left) and Full Width Half Maximum (FWHM) smearing (right).

theoretical peaks under the same smeared peak, and this results into a very simple curve with few modes that can be easily learnt by a DL model. Since wavelength and energy are inversely related to each other, using 'nm' to represent the UV-vis spectra of GDB-9 molecules results into spreading out the theoretical peaks into a broad range of values. The Gaussian smoothing now transforms the theoretical peaks into a more complex curve (with more peaks), which makes the DL learning more challenging, but also more informative. In the numerical results of this work, we will perform an ablation study that describes the predictive performance of each GNN models using both 'eV' and 'nm' as measure units for the UV-vis spectrum. The use of 'eV' allows us to compare our results with the ones of previous works in the context of DL for UV-vis spectroscopy [23, 24], whereas the use of 'nm' ensures that our analysis can provide meaningful results for researchers in the domains of computational chemistry and chemical engineering.

Gaussian smearing applied to UV-vis spectra of GDB-9-Ex molecules. In Figure 4, we show the effects of different Gaussian smearing on the graph of the UV-vis spectrum when the UV-vis spectrum is represented in 'eV' and 'nm'. The grey-shaded curves illustrate the effect of the Gaussian smearing applied to each individual theoretical peak, and the black solid curve represents the sum of the distributions from each smoothed theoretical peak and that defines the smeared UV-vis spectrum. Figures 4(a) and 4(b) show the smeared UV-vis spectrum using the wavelength of the light in 'nm' on the abscissas, without Gaussian smearing (FHWM set to 0.0nm) and with a Gaussian smearing FWHM parameter set to 5.0nm, respectively. By contrast Figures 4(c) and 4(d) illustrate the same spectrum using the excitation energy level in 'eV' on the abscissas, without Gaussian smearing (FHWM set to 0.0eV) and with a Gaussian smearing FWHM parameter set to 0.5eV, respectively.

5 Proposed Model Architecture

We generalized the software infrastructure of HydraGNN to support invariant MPNNs. Specifically, the invariant MPNNs that we incorporated in HydraGNN are: SchNet, DimeNet++, and EGNN. The software development tasks that we performed to include invariant MPNNs in HydraGNN are the following: (1) generalize the type of data structures accepted as input by the MPNN layers to allow a customizable type of feature embedding; (2) customize feature embedding mechanisms for each invariant MPNN to enable the construction of model-specific invariant features; (3) include new MPNN classes that implement invariant message passing policies and that respect the underlying object oriented class inheritance characteristic of the HydraGNN architecture.

In Figure 3 we presented the core components of the SchNet, DimeNet++, and EGNN models. Components retained during their re-implementation into the HydraGNN architecture are highlighted in yellow. This re-design addresses the main limitations of each individual model and maintains invariance in the message passing layer.

7

For the SchNet model, both the graph rewiring and positional feature encoding via Gaussian smearing are combined into the customizable feature embedding, preserving the entire message passing layer. For DimeNet++, radial and spherical basis encodings are integrated into the customizable feature embedding, retaining the complete message passing layer. Additionally, the EGNN model inherits the feature embedding, thereby preserving both the edge attribute constructor and the hidden feature update components of the message passing layer.
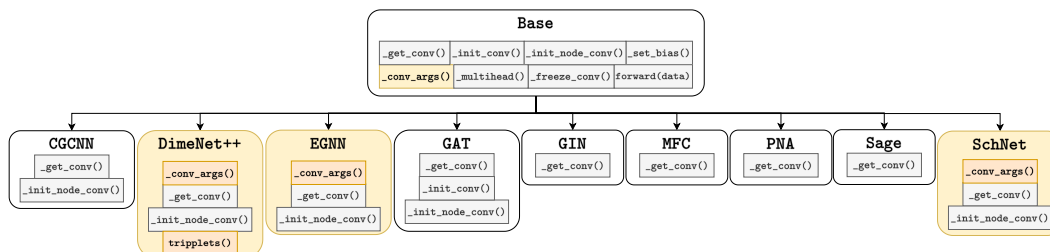

Figure 5: Changes in the object oriented programming structure of HydraGNN to include invariant MPNN layers SchNet, DimeNet++, and EGNN.

This restructuring and generalization of the HydraGNN capabilities brought the new MPNNs included (SchNet, DimeNet++, EGNN) into alignment under the HydraGNN umbrella, as shown in Figure 5. More specifically, through their inclusion in the HydraGNN architecture, these MPNN layers have been generalized with respect to their original implementations, as their implementation in HydraGNN allows users to use their own customized input features. This comprehensive setup makes HydraGNN a distinct GNN architecture that enables to seamlessly switch between different MPNN layers, thereby allowing to treat them as hyperparameters, and provides effective scalability across all the MPNN layers supported.

6 Numerical Results

In this section, we perform numerical tests to illustrate (1) the improvement of predictions on the the UV-vis spectrum of organic molecules using the new MPNN layers included in the HydraGNN architecture and (2) the efficacy of adding customized user-defined invariant features as input to the models. To this effect, we perform an ablation study on the FWHM in both 'eV' and 'nm' and on the MPNN layers SchNet, DimeNet++, EGNN, and PNA to determine the most robust model hat maintains accuracy and generalizability across different FWHM and different measure units to describe the spectrum. We then compare the invariant and non-invariant schemes for the SchNet model (the most performing model from the ablation study).

Training Procedure. In the ablation study across different MPNN layerswe compare HydraGNN's existing non-invariant PNA implementation with newly added SchNet, DimeNet++, and EGNN message passing policies. SchNet has two variants based on invariance, one using provided edge feature embeddings denoted as `SchNet-feat`, and the other utilizing graph rewiring for edge features reconstruction, detailed in Figure 3. Node and edge feature descriptions for each model are in Appendix B. The ablation study covers four variations of the GDB-9-Ex dataset, differing in spectrum description units and Gaussian smearing breadth. For energy state-based UV-vis spectra, we apply Gaussian smearing with 0.5eV and 1.0eV FWHM values. For wavelength-based spectra, we use 5.0nm and 10.0nm FWHM values. All versions use an 80%/10%/10% training/validation/test split, with fixed data splits for reproducibility. Training each HydraGNN model allows a maximum of 200 epochs with a batch size of 32 samples. Early stopping occurs if there's no improvement in validation loss after 20 epochs, and test results use the model with the lowest validation loss. We employ the AdamW optimizer with cosine annealing, reducing the learning rate to $1e$-5 after 200 epochs. Hyperparameter optimization (HPO) is conducted for each MPNN layer type (PNA, SchNet, DimeNet++, EGNN) over the number of MPNN layers L, hidden dimension h_{mpnn} for each MPNN layer, and initial learning rate. We use the Optuna library's Tree of Parzen Estimator (TPE) as the HPO methodology, sampling eight hyperparameter configurations sequentially. Due to computational constraints, HPO is performed on the 'eV' measurement unit dataset with a 0.5eV FWHM Gaussian smearing. These best-performing hyperparameter configurations are then applied to other dataset versions. Further details on these hyperparameters are in Table 2 of Appendix C.

Ablation Study on Type of MPNN Layer, Measure Unit, and FWHM Values. Table 1 reports the test MAE for each HydraGNN model on each version of the GDB-9-Ex dataset. We observe that the SchNet invariant model outperforms all other models across all the measure units and FWHM values.

Accuracy Improvement Using Invariant Features in SchNet. To validate the improvement in accuracy using invariant features, we show the predictions of the SchNet-feat and SchNet-inv architectures on some molecular samples of the test data in Figure 6. The selected molecules are characterized by spectra of increasing complexity from left to right. In particular, these data samples are extremely diverse in terms of molecular structure and in terms of number, location and height of the smeared peaks of the UV-vis spectrum. We observe that SchNet-inv better predicts the smeared spectrum across all three molecular samples. Additionally SchNet-inv better predicts the location and amplitudes of each individual smeared peak.

Model	0.5eV	1.0eV	5nm	10nm
DimeNet++	16.84	30.29	15.18	26.64
EGNN	3.76	4.26	5.65	7.16
PNA	3.60	4.11	5.57	7.24
SchNet-feat	3.39	3.63	5.29	6.49
SchNet-inv	**3.15**	**3.31**	**5.19**	**6.32**

Table 1: Test MAE values for each Gaussian smearing level using the optimal hyper parameters for each model found in Task 1.
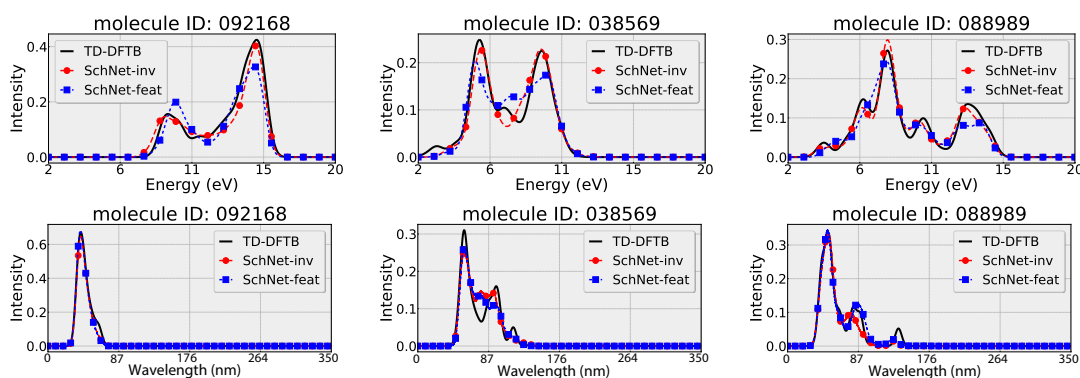

Figure 6: Predicted UV-vis spectrum of three molecules of increasing complexity using the SchNet-feat and SchNet-inv models trained on the 0.5eV(top) and 10nm(bottom) datasets. The SchNet-inv model is capable of more accurately predicting the intensity and location of spectral peaks.

7 Concluding Remarks

We enhanced HydraGNN, an open-source GNN architecture, by introducing invariant MPNN layers among its message passing policies. This improves HydraGNN's accuracy and generalizability in predicting molecular properties invariant to structural rotations and translations. We demonstrated this improvement by using invariant features to predict UV-vis spectra in the open-source GDB-9-Ex dataset, enhancing the model's ability to capture complex spectral features. Our model's robustness was confirmed across different measurement units and Gaussian smearing levels, showcasing its potential for AI-driven molecular design aligned with the scientific goals of the US-DoE.

Future work will expand on this by incorporating equivariant MPNNs into HydraGNN, further enhancing its accuracy and generalizability for predicting atomic properties in non-equilibrium molecular structures, such as charge transfer, dipole moment, and interatomic forces. Additionally, we aim to integrate the invariant and equivariant-preserving HydraGNN architecture into a generative framework to address computational chemistry's dimensionality challenge and efficiently solve inverse problems [40].

Acknowledgements

References

[1] Nicholar C. Thomas. The early history of spectroscopy. *Journal of Chemical Education*, 68(631), 1991. doi: 10.1021/ed068p631. URL https://pubs.acs.org/doi/10.1021/ed068p631. 1

[2] Migue A. L. Marques, Carsten A. Ullrich, Fernando Nogueira, Angel Rubio, Kieron Burke, and Eberhard K. U. Gross. Time-dependent density functional theory; lecture notes in physics. 2006. doi: 10.1007/b11767107. URL https://link.springer.com/book/10.1007/b11767107. 2

[3] Sijin Ren, Joseph Harms, and Marco Caricato. An eom-ccsd-pcm benchmark for electronic excitation energies of solvated molecules. *Journal of Chemical Theory and Computation*, 13 (1):117–124, 2017. doi: 10.1021/acs.jctc.6b01053. URL https://doi.org/10.1021/acs.jctc.6b01053. PMID: 27973775. 2

[4] Linfeng Zhang, Jiequn Han, Han Wang, Roberto Car, and Weinan E. Deep potential molecular Dynamics: A scalable model with the accuracy of quantum mechanics. *Phys. Rev. Lett.*, 120 (14):143001, April 2018. doi: 10.1103/PhysRevLett.120.143001. URL https://link.aps.org/doi/10.1103/PhysRevLett.120.143001. 2

[5] Jörg Behler and Michele Parrinello. Generalized neural-network representation of high-dimensional potential-energy surfaces. *Phys. Rev. Lett.*, 98(14):146401, April 2007. doi: 10.1103/PhysRevLett.98.146401. URL https://link.aps.org/doi/10.1103/PhysRevLett.98.146401. 2

[6] Kristof T. Schütt, Michael Gastegger, Alexandre Tkatchenko, and Klaus-Robert Müller. Quantum-chemical insights from interpretable atomistic neural networks. In Wojciech Samek, Grégoire Montavon, Andrea Vedaldi, Lars Kai Hansen, and Klaus-Robert Müller, editors, *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, volume 11700 of *Lecture Notes in Computer Science*, pages 311–330. Springer, 2019. doi: 10.1007/978-3-030-28954-6_17. URL https://doi.org/10.1007/978-3-030-28954-6_17. 2

[7] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2009. doi: 10.1109/TNN.2008.2005605. URL https://ieeexplore.ieee.org/document/4700287. 2

[8] Matthias Fey and Jan Eric Lenssen. Fast graph representation learning with PyTorch Geometric. In *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019. URL https://rlgm.github.io/papers/2.pdf. 2

[9] PyTorch Geometric. URL https://pytorch-geometric.readthedocs.io/en/latest/. 2

[10] Minjie Wang, Da Zheng, Zihao Ye, Quan Gan, Mufei Li, Xiang Song, Jinjing Zhou, Chao Ma, Lingfan Yu, Yujie Gai, Tianjun Xiao, Tong He, George Karypis, Jinyang Li, and Zheng Zhang. Deep graph library: A graph-centric, highly-performant package for graph neural networks. *arXiv: Learning*, 2019. URL https://api.semanticscholar.org/CorpusID:221304724. 2

[11] Da Zheng, Minjie Wang, Quan Gan, Xiang Song, Zheng Zhang, and George Karypis. Scalable graph neural networks with deep graph library. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, WSDM '21, page 1141–1142, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450382977. doi: 10.1145/3437963.3441663. URL https://doi.org/10.1145/3437963.3441663.

[12] Deep Graph Library (DGL). URL https://www.dgl.ai. 2

[13] Stefan Doerr, Maciej Majewski, Adrià Pérez, Andreas Krämer, Cecilia Clementi, Frank Noé, Toni Giorgino, and Gianni De Fabritiis. TorchMD: A deep learning framework for molecular simulations. *Journal of Chemical Theory and Computation*, 17(4):2355–2363, mar 2021. doi: 10.1021/acs.jctc.0c01343. URL https://pubs.acs.org/doi/10.1021/acs.jctc.0c01343. 2

[14] Massimiliano Lupo Pasini, Samuel Temple Reeve, Pei Zhang, and Jong Youl Choi. HydraGNN, Version 1.0. [Computer Software] https://doi.org/10.11578/dc.20211019.2, oct 2021. URL https://github.com/ORNL/HydraGNN. 2

[15] Massimiliano Lupo Pasini, Pei Zhang, Samuel Temple Reeve, and Jong Youl Choi. HydraGNN, Version 2.0. [Computer Software] https://doi.org/10.11578/dc.20230501.1, may 2023. URL https://github.com/ORNL/HydraGNN/releases/tag/v2.0.0. 2

[16] Massimiliano Lupo Pasini, Pei Zhang, Samuel Temple Reeve, and Jong Youl Choi. Multi-task graph neural networks for simultaneous prediction of global and atomic properties in ferromagnetic systems. *Machine Learning: Science and Technology*, 3(2):025007, may 2022. doi: 10.1088/2632-2153/ac6a51. URL https://dx.doi.org/10.1088/2632-2153/ac6a51. 2

[17] Jong Youl Choi, Pei Zhang, Kshitij Mehta, Andrew Blanchard, and Massimiliano Lupo Pasini. Scalable training of graph convolutional neural networks for fast and accurate predictions of homo-lumo gap in molecules. *Journal of Cheminformatics*, 14(1):1–10, 2022. doi: 10.1186/s13321-022-00652-1. URL https://jcheminf.biomedcentral.com/articles/10.1186/s13321-022-00652-1. 2

[18] Kristof Schütt, Pieter-Jan Kindermans, Huziel Enoc Sauceda Felix, Stefan Chmiela, Alexandre Tkatchenko, and Klaus-Robert Müller. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 991–1001, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/303ed4c69846ab36c2904d3ba8573050-Abstract.html. 2, 4, 14

[19] Massimiliano Lupo Pasini, Pilsun Yoo, Kshitij Mehta, and Stephan Irle. GDB-9-Ex: Quantum chemical prediction of UV/Vis absorption spectra for GDB-9 molecules. 11 2022. doi: 10.13139/OLCF/1890227. URL https://www.osti.gov/biblio/1890227. 2

[20] Johannes Klicpera, Shankari Giri, Johannes T. Margraf, and Stephan Gunnemann. Fast and uncertainty-aware directional message passing for non-equilibrium molecules. *ArXiv*, abs/2011.14115, 2020. doi: 10.48550/arXiv.2011.14115. URL https://doi.org/10.48550/arXiv.2011.14115. 4, 5, 15

[21] Víctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E(n) equivariant graph neural networks. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 9323–9332. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/satorras21a.html. 4, 5, 6, 15

[22] Carlo Innamorati, Tobias Ritschel, Tim Weyrich, and Niloy J. Mitra. Learning on the edge: investigating boundary filters in CNNs. *International Journal of Computer Vision*, 128: 773–782, 2020. doi: 10.1007/s11263-019-01223-y. URL https://doi.org/10.1007/s11263-019-01223-y. 5

[23] Kunal Ghosh, Annika Stuke, Milica Todorović, Peter Bjørn Jørgensen, Mikkel N. Schmidt, Aki Vehtari, and Patrick Rinke. Deep learning spectroscopy: Neural networks for molecular excitation spectra. 6(9):1801367. ISSN 2198-3844, 2198-3844. doi: 10.1002/advs.201801367. URL https://onlinelibrary.wiley.com/doi/10.1002/advs.201801367. 5, 7

[24] Kanishka Singh, Jannes Münchmeyer, Leon Weber, Ulf Leser, and Annika Bande. Graph Neural Networks for Learning Molecular Excitation Spectra. *Journal of Chemical Theory and Computation*, 18(7):4408–4417, July 2022. ISSN 1549-9618, 1549-9626. doi: 10.1021/acs.jctc.2c00255. URL https://pubs.acs.org/doi/10.1021/acs.jctc.2c00255. 5, 7

[25] Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation- and translation-equivariant neural networks for 3D point clouds. *ArXiv*, 1802.08219, 2018. URL https://arxiv.org/abs/1802.08219. 5

[26] Brandon Anderson, Truong Son Hy, and Risi Kondor. Cormorant: Covariant molecular neural networks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/03573b32b2746e6e8ca98b9123f2249b-Paper.pdf. 5

[27] Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P. Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E. Smidt, and Boris Kozinsky. E(3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature Communications*,

13:2453, 2022. doi: 10.1038/s41467-022-29939-5. URL https://doi.org/10.1038/s41467-022-29939-5. 5

[28] Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. Se(3)-transformers: 3d roto-translation equivariant attention networks. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1970–1981. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/15231a7ce4ba789d13b722cc5c955834-Paper.pdf. 5

[29] Marc Finzi, Samuel Stanton, Pavel Izmailov, and Andrew Gordon Wilson. Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020. URL https://proceedings.mlr.press/v119/finzi20a.html. 5

[30] Johannes Brandstetter, Rob Hesselink, Elise van der Pol, Erik J Bekkers, and Max Welling. Geometric and Physical Quantities improve E(3) Equivariant Message Passing. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=_xwr8gOBeV1. 5

[31] Tuan Le, Frank Noe, and Djork-Arné Clevert. Representation learning on biomolecular structures using equivariant graph attention. In *The First Learning on Graphs Conference*, 2022. URL https://openreview.net/forum?id=kv4xUo5Pu6. 6

[32] Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael John Lamarre Townshend, and Ron Dror. Learning from protein structure with geometric vector perceptrons. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=1YLJDvSx6J4. 6

[33] Bowen Jing, Stephan Eismann, Pratham N. Soni, and Ron O. Dror. Equivariant graph neural networks for 3d macromolecular structure. *CoRR*, abs/2106.03843, 2021. URL https://arxiv.org/abs/2106.03843. 6

[34] Kristof Schütt, Oliver T. Unke, and Michael Gastegger. Equivariant message passing for the prediction of tensorial properties and molecular spectra. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 9377–9388. PMLR, 2021. URL http://proceedings.mlr.press/v139/schutt21a.html. 6

[35] Philipp Thölke and Gianni De Fabritiis. Equivariant transformers for neural network based molecular potentials. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=zNHzqZ9wrRB. 6

[36] Yi-Lun Liao and Tess Smidt. Equiformer: Equivariant graph attention transformer for 3d atomistic graphs, 2022. URL https://openreview.net/forum?id=_efamP7PSjg. 6

[37] Jiaqi Han, Wenbing Huang, Tingyang Xu, and Yu Rong. Equivariant graph hierarchy-based neural networks. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=ywxtmG1nU_6. 6

[38] Raghunathan Ramakrishnan, Pavlo O Dral, Matthias Rupp, and O Anatole von Lilienfeld. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data*, 1, 2014. doi: 10.1038/sdata.2014.22. URL https://doi.org/10.1038/sdata.2014.22. 6

[39] Massimiliano Lupo Pasini, Kshitij Mehta, Pilsun Yoo, and Stephan Irle. Two excited-state datasets for quantum chemical UV-vis spectra of organic molecules. *Scientific Data*, 2023. doi: 10.1038/s41597-023-02408-4. 6

[40] Christian Etmann Brynjulf Owren Carola-Bibiane Schönlieb Elena Celledoni, Matthias J Ehrhardt and Ferdia Sherry. Equivariant neural networks for inverse problems. *Inverse Problems*, 37(8), 2021. doi: 10.1088/1361-6420/ac104f. URL https://iopscience.iop.org/article/10.1088/1361-6420/ac104f. 9

[41] Gabriele Corso, Luca Cavalleri, Dominique Beaini, Pietro Liò, and Petar Veličković. Principal neighbourhood aggregation for graph nets. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 13260–13271. Curran Associates, Inc.,

2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/99cad265a1768cc2dd013f0e740300ae-Paper.pdf. 14

Supplementary materials for
Invariant Features for Accurate Predictions of Quantum Chemical UV-vis Spectra of Organic Molecules

A   A Brief Review of Message Passing Neural Networks

This section provides the core MPNN details for the PNA, SchNet, DimeNet++ and EGNN models.

MPNN Framework.   The message passing neural network updates the set of node features $\boldsymbol{H}^l$ at layer l by aggregating messages among the nodes neighbors as follows.

$$\boldsymbol{m}_{ij}^l = \phi_{\boldsymbol{e}}(\boldsymbol{h}_i^{(l)}, \boldsymbol{h}_j^{(l)}, \boldsymbol{e}_{ij})$$
$$\boldsymbol{m}_i^l = \bigoplus_{j \in \mathcal{N}(i)} \boldsymbol{m}_{ij}^l \tag{3}$$
$$\boldsymbol{h}^{(l+1)} = f_{\boldsymbol{h}}(\boldsymbol{h}^{(l)}, \boldsymbol{m}_i^l)$$

The messages $\boldsymbol{m}_{ij}^l$ are generated among all edges in the graph using a parametrized function $\phi_{\boldsymbol{e}}$. Then for each node i the messages are aggregated among its neighbors $\mathcal{N}(i)$ using the aggregation function $\bigoplus$. Finally the node feature is updated using a parametrized function $\phi_{\boldsymbol{h}}$. The particular function parametrizations and aggregation mechanisms define the MPNNs behavior.

PNA[41].   The PNA model uses the following message passing network.

$$S(d, \alpha) = \left(\frac{\log(d+1)}{\delta} \right)^{\alpha}, \quad d > 0, \quad 1 \le \alpha \le 1$$
$$\boldsymbol{m}_{ij}^l = \phi_{\boldsymbol{e}}(\boldsymbol{h}_i^{(l)}, \boldsymbol{h}_j^{(l)}, \boldsymbol{e}_{ij})$$
$$\boldsymbol{m}_i^l = \begin{bmatrix} \boldsymbol{I} \\ S(D, \alpha = 1) \\ S(D, \alpha = -1) \end{bmatrix} \otimes \begin{bmatrix} \mu \\ \sigma \\ \max \\ \min \end{bmatrix} \left(\boldsymbol{m}_{ij}^l |_{j \in \mathcal{N}(i)} \right) \tag{4}$$
$$\boldsymbol{h}^{(l+1)} = \phi_{\boldsymbol{h}}(\boldsymbol{h}^{(l)}, \boldsymbol{m}_i^l)$$

In the PNA model, the aggregation scheme is designed to amplify or attenuate edge statistics, for instance the mean μ and/or standard deviation σ. The amplification factor $S(d, \alpha = 1)$ and attenuation factor $S(D, \alpha = -1)$ can be controlled via hyperparameters α. Additionally, the selection of edge feature statistics can be adjusted via hyperparameters.

SchNet[18].   The SchNet model uses the following message passing network.

$$\boldsymbol{f}_{ij} = \exp(-\gamma(||x_i - x_j|| - \mu))$$
$$\boldsymbol{m}_{ij}^l = h_j^l \odot \phi_v e(f_{ij}))$$
$$\boldsymbol{m}_i^l = \sum_{j \in \mathcal{N}(i)} \boldsymbol{m}_{ij}^l \tag{5}$$
$$\boldsymbol{h}_i^{l+1} = \boldsymbol{h}_i + \boldsymbol{m}_i^l$$

In SchNet, the invariant features are captured using a Gaussian filter. The filter coefficients are determined by γ and μ, with γ controlling the width and μ controlling the position of the Gaussian. The messages are then generated by modulating the features along the neighboring nodes. The HydraGNN architecture uses the default SchNet hyperparameters for controlling the Gaussian filter.

We also consider a variant of the above message passing network SchNet-feat which uses provided edge features instead of invariant features. The message passing network is provided below.

14

$$\boldsymbol{m}_{ij}^l = h_j^l \odot \phi_v e(\boldsymbol{e}_{ij}))$$

$$\boldsymbol{m}_i^l = \sum_{j \in \mathcal{N}(i)} \boldsymbol{m}_{ij}^l \tag{6}$$

$$\boldsymbol{h}_i^{l+1} = \boldsymbol{h}_i + \boldsymbol{m}_i^l$$

DimeNet++[20]. The DimeNet++ model utilizes the directional messsage passing architecture of DimeNet. In particular, it updates the messages directly at each step as opposed to the hidden features.

$$\boldsymbol{e}_{\mathrm{RBF}}^{(ji)} = \phi_e(||\boldsymbol{x}_i - \boldsymbol{x}_j||)$$

$$\boldsymbol{a}_{\mathrm{SBF}}^{(kj,ji)} = \phi_a(||\boldsymbol{x}_i - \boldsymbol{x}_j||, \alpha_{ijk})$$

$$\boldsymbol{m}_{ji}^0 = \phi_{\mathrm{in}}(\phi_{\mathrm{RBF}}^0(\boldsymbol{e}_{\mathrm{RBF}}) \oplus D[\boldsymbol{H}_{\mathrm{atom}}])$$

$$\boldsymbol{m}_{ji}^{l-1} = \sum_{k \in \mathcal{N}(j)/\{i\}} \phi_{\mathrm{dime}}(\boldsymbol{e}_{\mathrm{RBF}}^{(ji)}, \boldsymbol{a}_{\mathrm{SBF}}^{(kj,ji)}, \boldsymbol{m}_{kj}^{l-1}) \tag{7}$$

$$\boldsymbol{m}_{ji}^l = \phi_e(\boldsymbol{h}_i^l, \boldsymbol{h}_j^l, \boldsymbol{e}_{\mathrm{RBF}}^{(ji)}, \boldsymbol{m}_{ji}^{l-1})$$

$$\boldsymbol{m}_j^l = \sum_{j \in \mathcal{N}(i)} \boldsymbol{m}_{ji}^l$$

$$\boldsymbol{h}_i^{l+1} = \phi_h(\phi_{\mathrm{RBF}}^l(\boldsymbol{e}_{\mathrm{RBF}}) \odot \boldsymbol{m}_j^l)$$

The initial message of the DimeNet framework is built from the parametrized dictionary encoding of the atomic numbers. The invariant features are constructed via raidal and spherical basis functions dependent upon the distances between nodes and angles among all node triplets.

EGNN[21]. The EGNN model is designed to be equivariant. However, as the authors of [21] note, if the positional update is removed the model becomes invariant. The invariant message passing network of EGNN is given below.

$$\boldsymbol{f}_{ij} = ||\boldsymbol{x}_i^l - \boldsymbol{x}_j^l||^2 \boldsymbol{m}_{ij}^l \quad = \phi_m(\boldsymbol{h}_i^l, \boldsymbol{h}_j^l, \boldsymbol{f}_{ij}, \boldsymbol{e}_{ij})$$

$$\boldsymbol{m}_i^l = \sum_{j \in \mathcal{N}(i)} \boldsymbol{m}_{ij}^l \tag{8}$$

$$\boldsymbol{h}_i^{l+1} = \phi_h(\boldsymbol{h}_i^l, \boldsymbol{m}_i^l)$$

In this model the invariant features are the squared Euclidean distances between pairs of nodes.

B Additional Details on Molecular Datasets

The node features selected for the UV-vis spectrum datasets are

- the number of valence electrons
- the electron affinity
- the atomic weight
- the ionic energy.

Additional edge features are used the supporting models PNA, SchNet-feat and EGNN. For these models, the following edge features are used

- if the bond is a single bond
- if the bond is a double bond
- if the bond is a triple bond
- if the bond is an aromatic bond

- The bond length
- the planar angle
- the azimuth angle.

C   Additional Details on Hyperparameter Optimization

Model	hidden_dimension (L)	num_conv_layers (h_{mpnn})	learning_rate (lr)
DimeNet++	64	4	0.01
EGNN	512	5	0.0005
PNA	512	6	0.0005
SchNet-feat	256	2	0.0005
SchNet-inv	1024	3	0.0005

Table 2: Optimal hyperparameters for HydraGNN with varying MPNNs.

D   Additional Ablation Studies

In Figure 7 we illustrate several more comparisons between SchNet-feat and SchNet-inv models. We also compare between 0.5eV(left) and 10nm(right) datasets predictions. In general SchNet-inv is able to more accurately capture the intensity and location of the peaks. Additionally, capturing outlying peaks in the 10nm dataset is significantly more challenging as illustrated in molecules 047788 and 017766.
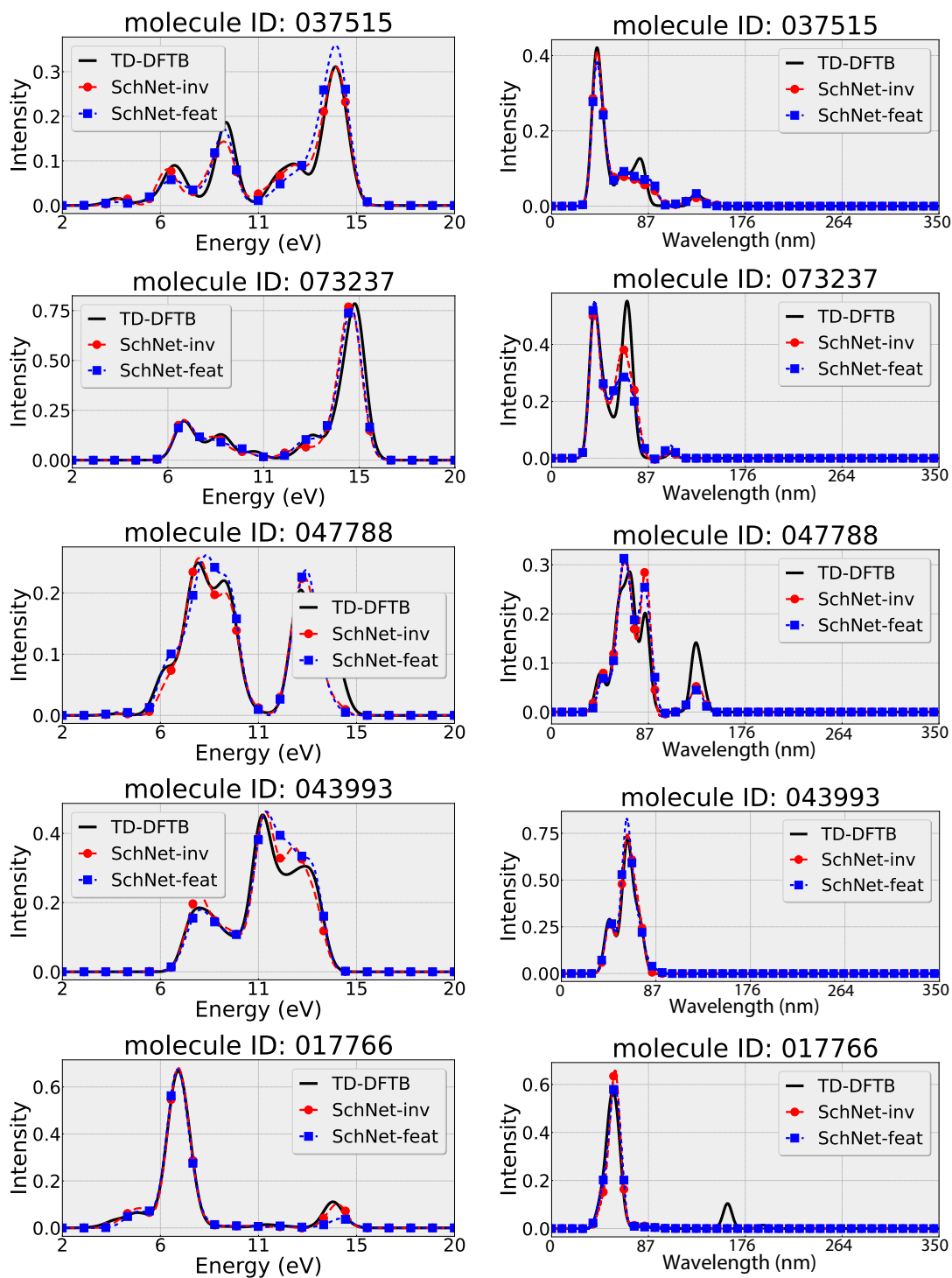
Figure 7: Additional comparisons between SchNet-inv and SchNet-feat for several molecules in the test split of 0.5eV and 10.0nm GDB-9-Ex datasets.