Supporting Information: Fast Evaluation of the Adsorption Energy of Organic Molecules on Metals via Graph Neural Networks

Sergio Pablo-García[1,2,†], Santiago Morandi[3,4,†], Rodrigo A. Vargas-Hernández[1,5], Kjell Jorner[1,2,6], Núria López[3,*], and Alán Aspuru-Guzik[1,2,*]

[1]Department of Chemistry, University of Toronto, Lash Miller Chemical Laboratories 80 St. George Street, ON M5S 3H6, Toronto, Canada
[2]Department of Computer Science, University of Toronto, Sandford Fleming Building, 40 St. George Street, ON M5S 2E4, Toronto, Canada
[3]Institute of Chemical Research of Catalonia, The Barcelona Institute of Science and Technology, Av. Països Catalans 16, 43007, Tarragona, Spain
[4]Department of Physical and Inorganic Chemistry, Universitat Rovira i Virgili, Campus Sescelades, N4 Block, C. Marcel·lí Domingo 1, 43007, Tarragona, Spain
[5]Vector Institute for Artificial Intelligence, 661 University Ave. Suite 710, ON M5G 1M1, Toronto, Canada
[6]Department of Chemistry and Chemical Engineering, Chalmers University of Technology, Kemigården 4, SE-412 96, Gothenburg, Sweden
[†]Authors Equally Contributed to this Work
[*]Corresponding Authors: email: nlopez@iciq.es; alan@aspuru.com

October 5, 2022

Notes

Figures

Tables

Notes

Note S1: FG-dataset

The functional groups dataset (FG-dataset) created to develop the GNN model presented in this work includes 204 closed-shell molecules. For each of them, the FG-dataset contains the gas-phase molecule and the molecule adsorbed on 12 metal surfaces (Ag, Au, Cd, Cu, Ir, Ni, Os, Pd, Pt, Rh, Ru, Zn). One adsorption configuration is included for each molecule/metal combination except for the aromatics with one ring, which comprise two different configurations for each metal. This leads to a total of 2849 DFT samples in the FG-dataset. The set of molecules represents the most common functional groups: for each chemical family, all the existing configurations containing up to C_4 are included, except aromatics where $C_{>4}$ are considered. The chemical families included in the FG-dataset are the following:

- Alkanes, alkenes and alkynes. **Table S1**
- Alcohols, aldehydes, ketones and ethers (1 Oxygen). **Table S1**
- Carbonates, carboxylic acids and esters (2–3 Oxygens). **Table S2**
- Amines and imines (1 Nitrogen). **Table S3**
- Amidines (2 Nitrogens). **Table S4**
- Thiols, thials, thioketones and thioethers (1 Sulfur). **Table S5**
- Amides and oximes (1 Oxygen + 1 Nitrogen). **Table S6–S7**

- Carbamate esters (2 Oxygens + 1 Nitrogen). **Table S8**
- Aromatic molecules with up to two rings containing O, N and S. **Table S9**

Note S2: Adsorption Conformational Search

Even simple C_{2-3} adsorbates could have approximately 100 conformations. Thus, for the initial DFT adsorption geometries we followed a simplified conformational analysis based on the heuristic rules devised in Refs. [1, 2]. These rules can be summarised as follows:

(i). The unsaturated bonds were placed close to the surface.

(ii). Heteroatoms (O, N and S) were placed close to the surface.

(iii). Carbon tails face the surface.

(iv). If the intermediate did not converge to a reasonable structure, the molecule was readjusted manually, trying up to 6 conformations that preserve the rules (i-iii).

Note S3: Automation of DFT Data Generation

To build and obtain the E_{DFT} of the adsorbate/metal combinations included in the FG-dataset, we executed the following procedure:

(i). First, the metallic surfaces were built starting from their respective bulks.

(ii). A list of SMILES [3] of the molecules in the FG-dataset has been fed to Open Babel [4] in order to generate the .xyz geometry files of the molecules. The applied force field is the MMFF94 [5].

(iii). The .xyz geometry files have been converted with pymatgen [6] to POSCAR files representing the molecules in cubic cells with dimension of 20 Å. The gas-phase molecules have been calculated with this initial geometry.

(iv). The relaxed molecules were copied and placed at a distance of 2.0–2.3 Å from the Rh slab, our reference metal. The conformational rules explained in **Note S2** have been followed.

(v). The resulting structures were sent to our computational resources and relaxed with VASP. Once full convergence has been reached, the geometry of the relaxed molecules on Rh has been extracted and automatically placed on the slabs of the all the other metals preserving the adsorbate distance from the Rh slab.

(vi). The obtained structures were checked for conformation errors, and if needed, manually built and relaxed again. Typical problems encountered were related to the fragmentation of the adsorbate during the relaxation due to the unstable initial geometry.

For the BM-Dataset, the molecules were manually built, adsorbed on the metal surfaces and relaxed using VASP, including the gas phase molecules.

Structures obtained from both datasets were uploaded to ioChem-BD.[7, 8]

Note S4: Graph Representation Algorithm

To convert the three-dimensional structures obtained from DFT to their respective graph representation, we applied a modified version of the algorithm presented in Ref. [9], implemented in pyRDTP. To define the neighbors of each atom, the algorithm reads the relaxed three-dimensional atomic positions from the geometry file (CONTCAR) generated by VASP, and applies the Voronoi tessellation method. This method creates a partition of the three-dimensional space, defining a region for each atom that consists of all points of the space closer to that atom than to any other. Two atoms are considered connected if they share a Voronoi facet and their distance is less than the sum of the atomic covalent radii plus a tolerance distance. In this work we used as covalent radii those provided by Cordero et al. [10] multiplied by a scaling factor of 1.5 for metals, and a tolerance parameter of 0.5 Å to help in the detection of metal-adsorbate connections.

Once the connectivity is defined, the graphs are generated representing the atoms as nodes and the detected connections as edges. The metal atoms not directly connected to the adsorbate are not considered during the graph generation process. The atomic elements are embedded to the nodes using the one-hot encoding approach as implemented in Scikit-Learn (**Figure S1)**.[11] This step is needed to convert categorical variables (as atomic elements) into ML-suitable data structures. This algorithm is applied to all the samples in both FG-dataset and BM-dataset.

For a fraction of the adsorption systems, the graph conversion results into inaccurate representations, as for specific geometries the M—A distance is so

high that the algorithm is unable to properly define their connectivity. This is due to the fact that the algorithm is based on a purely geometrical criterion that defines the edges relying on a set of empirical covalent radii, which are fixed for each element and do not account for all the possible phenomena occurring in catalytic systems. Thus, the strategy for minimizing the amount of bad representations has been to fine tune the tolerance parameter and the covalent radii scaling factor. In order to discard inaccurate graph representations and properly curate the graph dataset to be suitable for the GNN model training, we implemented the following sieves (**Figure S2)**:

(i). A first filter that discards the graphs representing adsorption configurations without the presence of metal atoms.

(ii). A second filter that verifies the correct connectivity of C and H atoms within the molecules: Connectivity of carbon atoms is properly defined if the number of edges connecting it to other atoms in the molecule is equal or less than 4, while hydrogen atoms are correctly connected if its number of edges is exactly one.

(iii). A filter to prevent the inclusion in the dataset of DFT samples that contain more than one adsorbate on the slab or with a final geometry in which the adsorbate has dissociated in multiple fragments.

(iv). A last filter for removing duplicate graphs deriving from the presence of stereo-isomers adsorbed in the same configuration on the metal surface.

The amount of graphs pruned out after the first two filters intrinsically depends on the graph conversion algorithm: a higher applied tolerance pa-

rameter would reduce the number of discarded graphs after the first filter, but at the same time it increases the number of removed graphs in the second filter, due to the creation of nonphysical connections within the adsorbates.

Note S5: GNN Model Architecture

The architecture of the GNN model developed in this work is shown in **Figure S4.** The input graphs are represented by a set of node feature vectors, each of them being a 17-dimensional array (12 metals + C, H, O, N and S) needed for representing the chemical element via the one-hot encoder, and by the graph connectivity in coordinate format (**Figure S1**). In order to transform this mathematical graph representation into the prediction of the DFT energy of the associated chemical system, the following transformations are applied in the listed order:

(i). First, each node feature vector is transformed via one fully connected layer which increases the dimensionality of the vector from 17 to 128, $\mathbb{B}^{17} \rightarrow \mathbb{R}^{128}$, where $\mathbb{B}$ denotes the Boolean space used to define the nodes via one hot encoding.

(ii). Then, two additional fully connected layers are applied, keeping the dimensionality of the nodes ($\mathbb{R}^{128} \rightarrow \mathbb{R}^{128}$). Up to now, no information about the graph connectivity is exploited.

(iii). Three GraphSAGE [12] convolutional layers are applied to all the nodes to capture the information from the neighbors by exploiting the graph connectivity. Between each convolution, a fully-connected layer is ap-

plied.

(iv). Finally, the information embedded in the nodes is compressed into a global graph representation with the Graph Multiset Transformer (GMT),[13] a global pooling layer which returns back a scalar value, namely the prediction of the DFT energy of the chemical system represented by the initial input graph. $\mathbb{R}^{N\times 128} \rightarrow \mathbb{R}$, where N is the number of nodes in the graph.

The activation function used in the node-level layers (all except the pooling layer) is the rectified linear unit (ReLU). [14] In total, the developed GNN model contains 398,081 trainable parameters, 224,513 of them belonging to the pooling layer due to its high internal complexity and the remaining parameters equally distributed among the other layers.

Note S6: GNN Training

The model training has been performed minimizing the mean absolute error (MAE) as loss function with the ADAM algorithm as optimizer.[15] The learning rate (LR) value is steered with the reduce-on-plateau scheduler. The initial learning rate has been set to 10^{-3} and is reduced exponentially by the scheduler every time in which there is no improvement after 5 epochs (patience) in the MAE of the validation set. The minimum LR possible has been set to 10^{-6}, while the decrease factor has been set to 0.7. In each training 250 epochs are performed. During each epoch, the training set is fed to the model in batches of 32 samples, performing a backward propagation and updating the model parameters after each batch.

The performance of the GNN model has been assessed by applying a stratified splitting of the FG-dataset by chemical families followed by a 5-fold cross validation. The first (**Figure S6**) allows a proper distribution of all the chemical families among the splits, while the latter provides a robust estimation of the GNN generalization performance. The cross validation approach follows the process depicted in **Figure S7**: After the partition in 5 stratified splits, each split is employed as test set: for each split used as test set, 4 splits are left and each of them is employed as validation set. This leads to a final validation approach consisting of 20 independent learning processes, each one performed with a unique combination of the splits among train, validation and test set. The generalization performance of the model is finally assessed by averaging the MAE of the absolute errors of the learning processes performed employing the same test set. **Figure S8** shows the values of the learning rate and MAE among the trained models. It is important to mention that after the creation of the train, validation and test sets in each learning process, a standardized target scaling is applied using the mean and standard deviation of the energy values of the samples from the train and validation sets, discarding the samples from the test set as its inclusion would lead to a data leakage. The target scaling is essential to ensure a stable learning process: if the target variable among the graph dataset has a large spread of values, it may result in large error gradients causing model parameters to change dramatically, making the learning process unstable.

Note S7: BM-dataset

The big molecules dataset (BM-dataset) used for testing the GNN model includes three industrially relevant materials:

- Biomass. Ref. [16], **Figure S10**
- Polyurethane precursors. Ref. [17], **Figure S11**
- Plastics. Ref. [18] **Figure S12**

These three groups consist of complex chemical structures that can be seen as combinations of the functional groups present in the FG-dataset. For each group, 5 representative molecules of larger size compared to those in the FG-dataset have been selected and relaxed through DFT to simulate the gas-phase and adsorption configuration on 2 metals chosen according to the existing applications and studies. The BM-dataset (45 samples, 30 adsorptions + 15 gas-phase) is used as an additional test for assessing the GNN performance on samples coming from a distribution distinct from that used to build the model (FG-dataset). For example, for the plastic group we represent polyethylene (PE), polypropylene (PP, both syndio- and isotactic), polystyrene (PS) and polyethylene tereftalate (PET) as molecules composed by a reasonable number of monomers. We generated the DFT adsorption systems of these molecules on Pt and Ru metal surfaces as these represent potential candidates for applications related to chemical recycling technology.

Note S8: Hyperparameter Optimization

Before testing the proposed model on the BM-dataset, we performed a hyperparameter optimization study to explore the vast space defined by all the variables that can affect the final performance of the GNN model. The hyperparameters can be defined as all the variables that are not model parameters, but that affect model performance at the same extent. These can be divided in two groups:

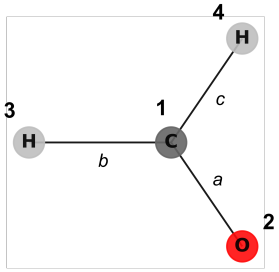
- Training-related: they define the training process. Examples are learning rate, optimization algorithm, batch size, etc.
- Model-related: they define the architecture of the model and include kind of activation function (ReLu, Tanh, etc.), number and depth of layers, bias inclusion, etc.

We adopted the hyperband asynchronous algorithm (ASHA) [19] implemented in RayTune. [20] ASHA combines random search and aggressive early stopping in order to optimize the hyperparameters and is based on the proved fact that in order to find the best hyperparameter settings, just a small amount of iterations (epochs) is sufficient to discriminate between bad models and promising candidates.

The hyperparameter space, shown in **Table S12** has been investigated picking randomly 2000 different settings and for each of those, a model training has been run by ASHA with a grace period of 15 epochs (e.g., the poorly-performing models are discarded after having trained them for a minimum of 15 iterations) and a maximum of 200 for the best ones. The final hyper-

parameter setting is the one that minimizes the MAE of the energy of the samples belonging to the BM-dataset.

Figures



Formaldehyde

Node embedding:
One-hot encoding

Node	C	H	O	N	S
1	1	0	0	0	0
2	0	0	1	0	0
3	0	1	0	0	0
4	0	1	0	0	0

Graph connectivity:
COO format

Edge	node 1	node 2
a	**1**	**2**
b	**1**	**3**
c	**1**	**4**

Figure S1: Example of graph data structure representation.

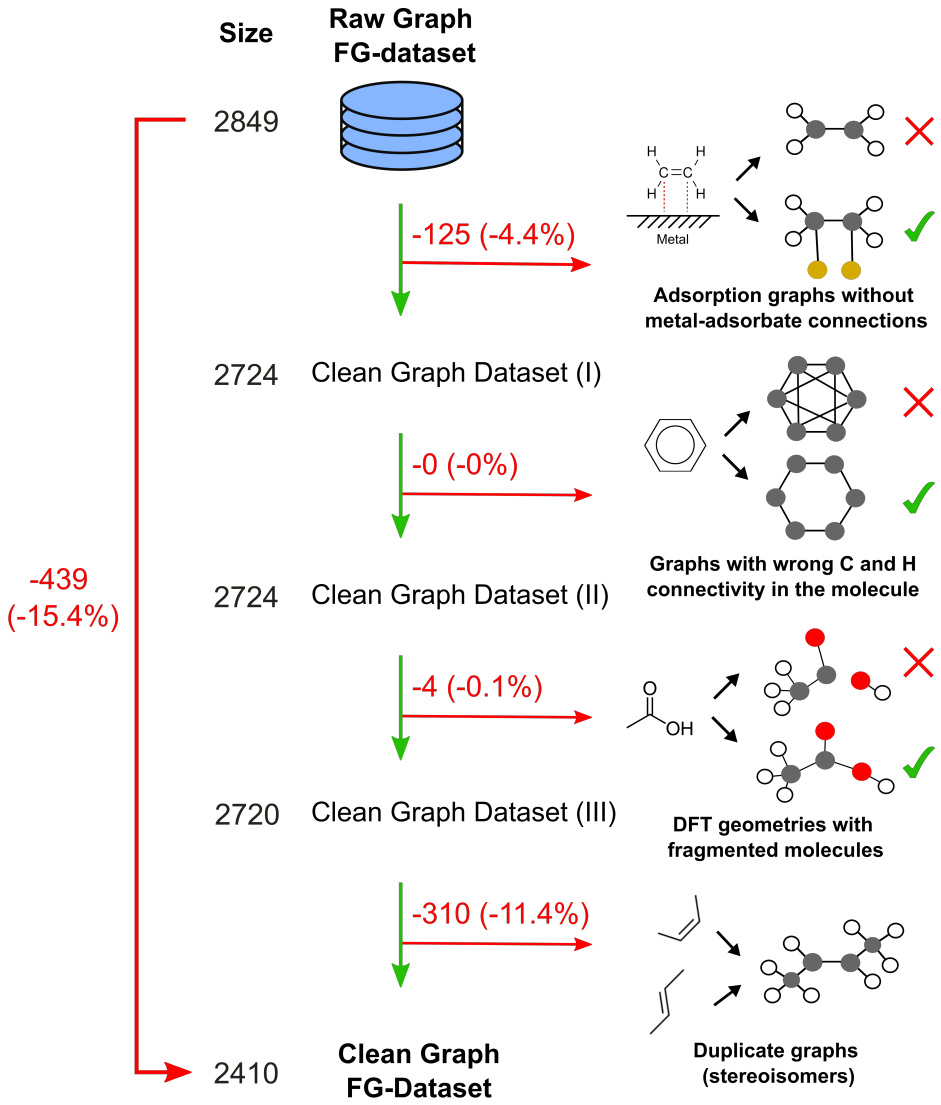


Figure S2: Data cleaning workflow applied to the raw graph FG-dataset.

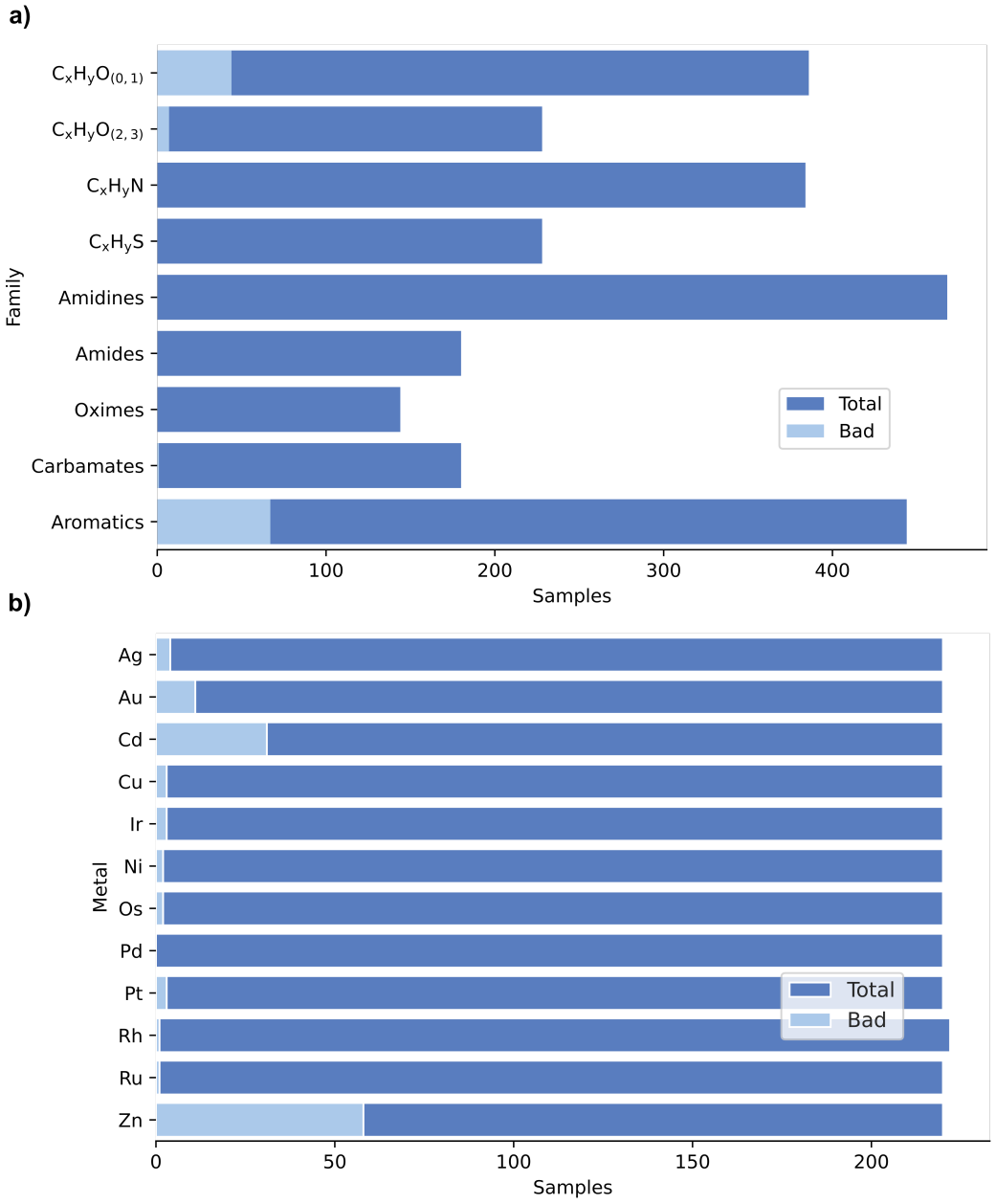


Figure S3: Graph representations of the FG-dataset without metal-adsorbate connections. a) Distribution by chemical family and b) by metal.

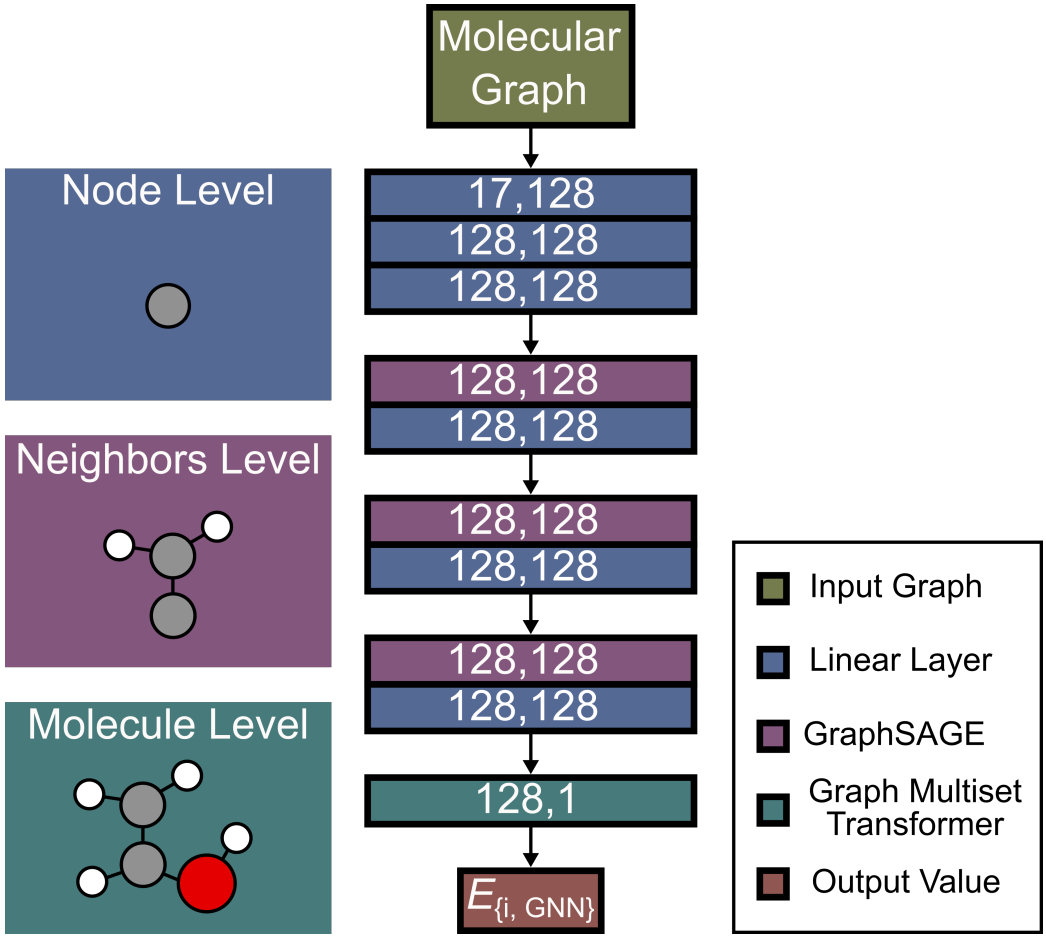


Figure S4: GNN model architecture before hyperparameter optimization.

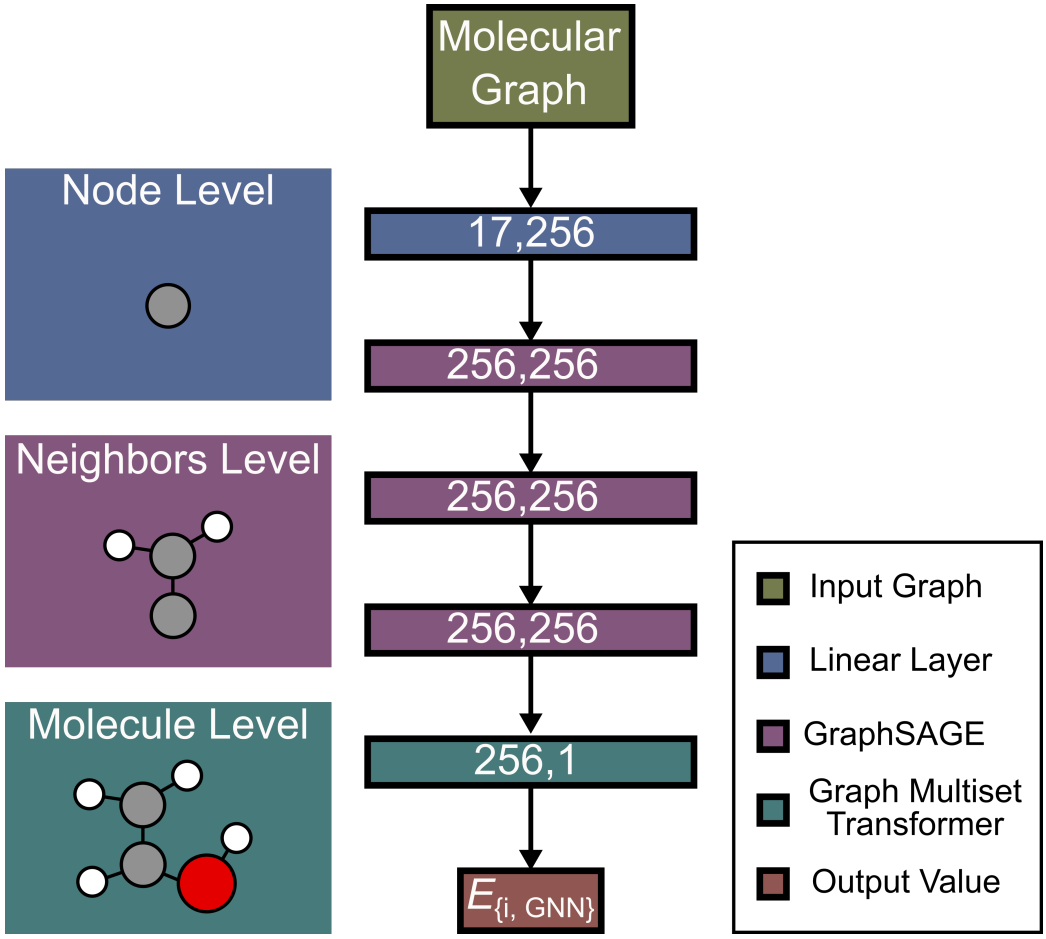


Figure S5: GNN model architecture after hyperparameter optimization.

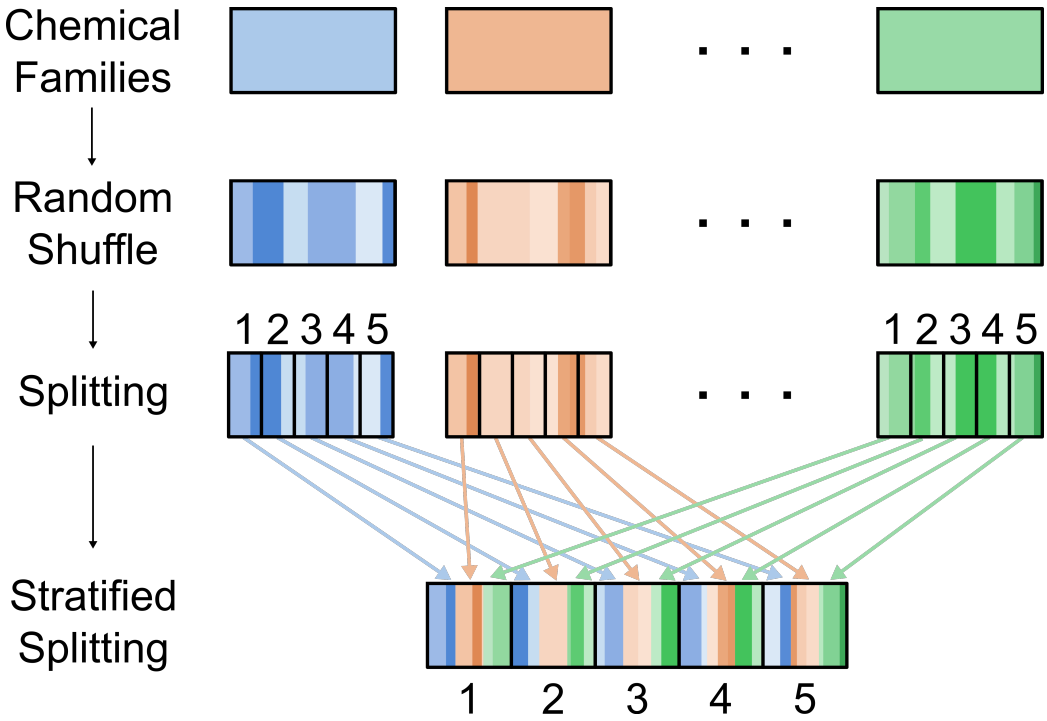


Figure S6: Stratified data splitting procedure.

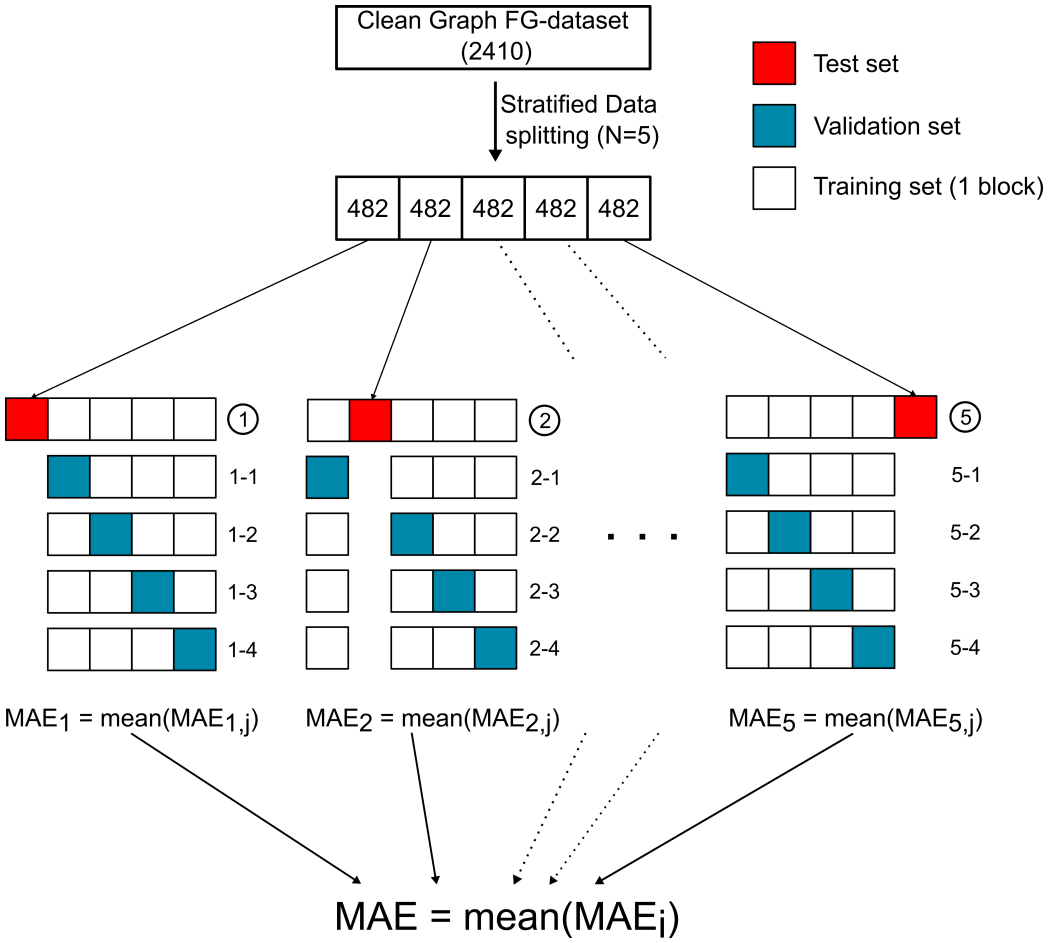


Figure S7: Cross validation approach for estimating the GNN generalization performance.

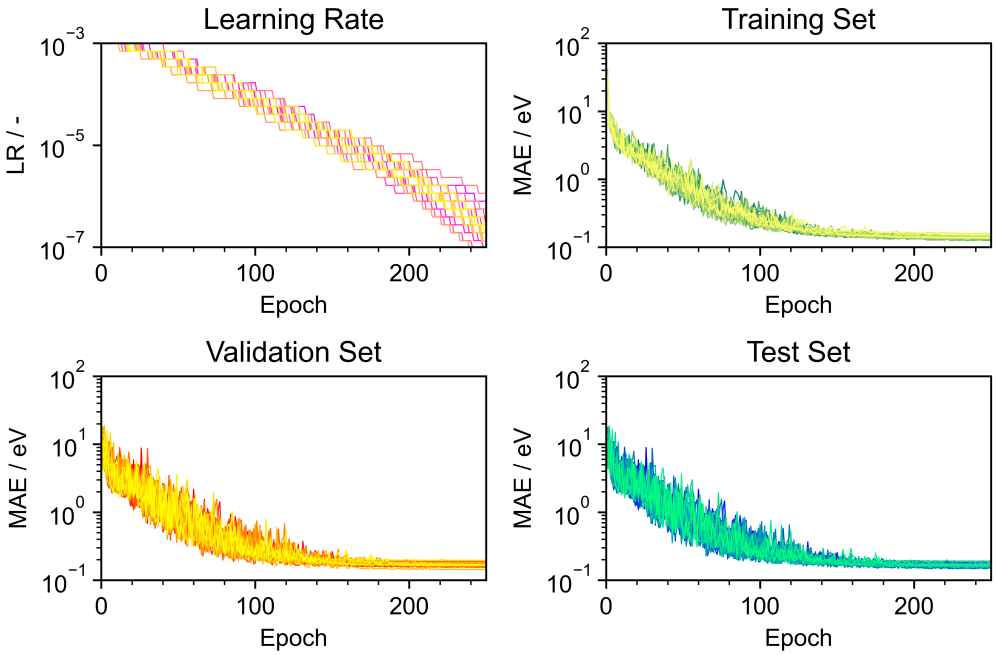


Figure S8: Learning rate and MAE of the train/validation/test sets during the training processes of the cross validation.

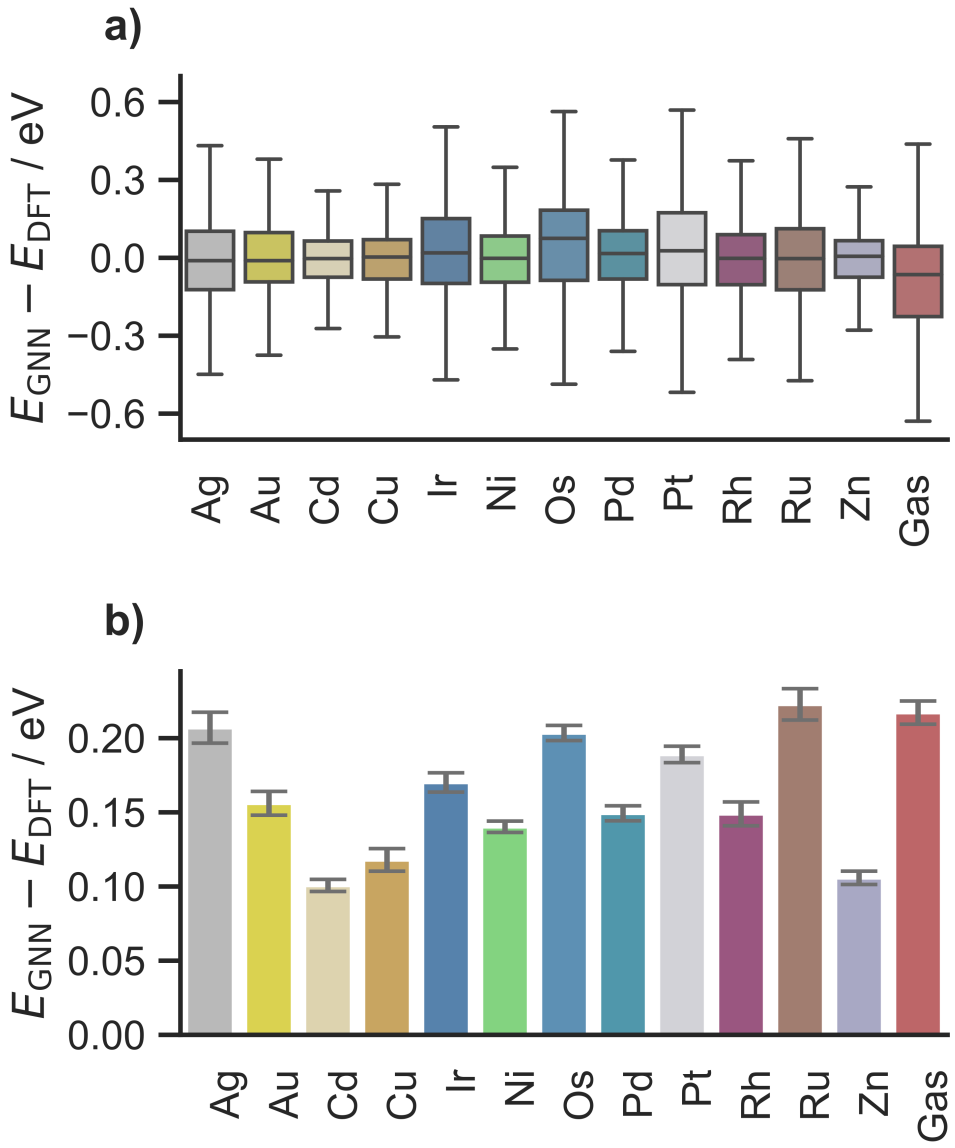
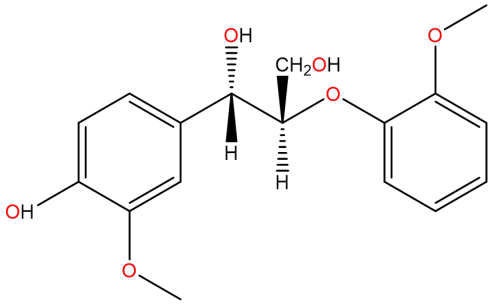
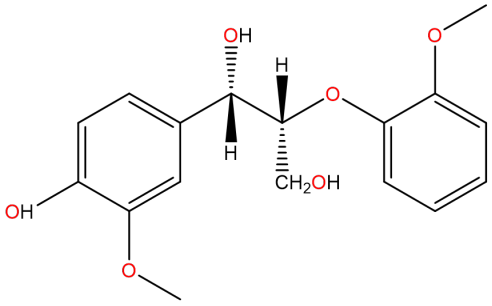


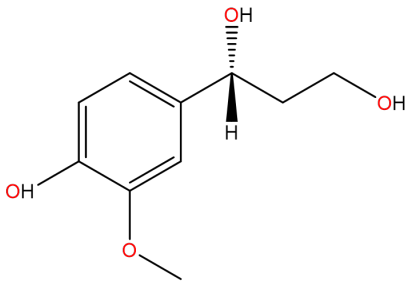
Figure S9: a) Box-plot of the test error distribution sorted by metal and b) mean error and standard error of the mean of the predictions obtained by the cross validation models.



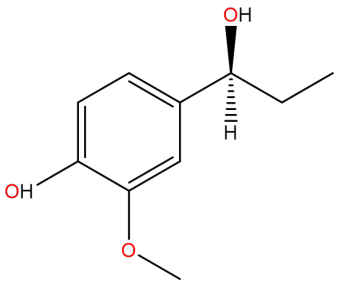
(1*S*,2*S*)-1-(4-hydroxy-3-methoxyphenyl)-2-(2-methoxyphenoxy)propane-1,3-diol



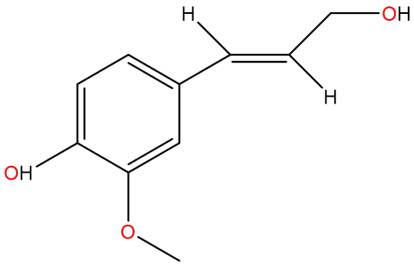
(1*S*,2*R*)-1-(4-hydroxy-3-methoxyphenyl)-2-(2-methoxyphenoxy)propane-1,3-diol



(*R*)-1-(4-hydroxy-3-methoxyphenyl)propane-1,3-diol



(*S*)-4-(1-hydroxypropyl)-2-methoxyphenol



(*E*)-4-(3-hydroxyprop-1-en-1-yl)-2-methoxyphenol

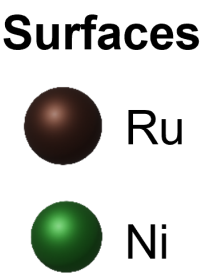
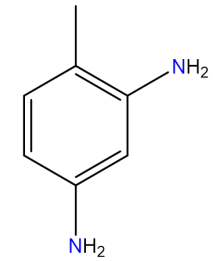
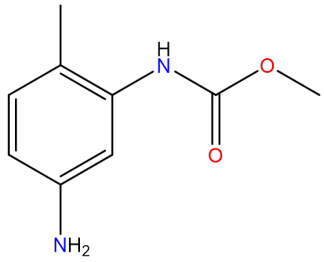


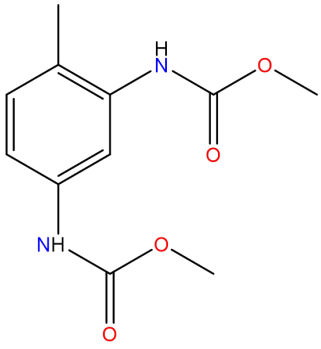
Figure S10: BM-dataset: Biomass molecules and metals.



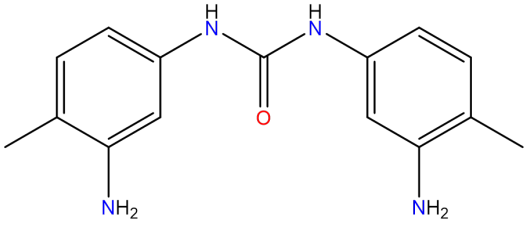
4-methylbenzene-1,3-diamine



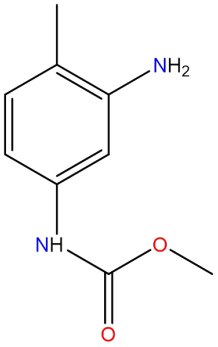
methyl (5-amino-2-methylphenyl)carbamate



dimethyl (4-methyl-1,3-phenylene)dicarbamate



1,3-bis(3-amino-4-methylphenyl)urea



methyl (3-amino-4-methylphenyl)carbamate

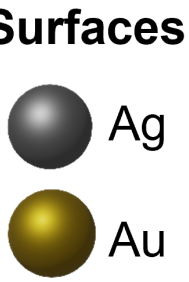


Figure S11: BM-dataset: Polyurethane molecules and metals.

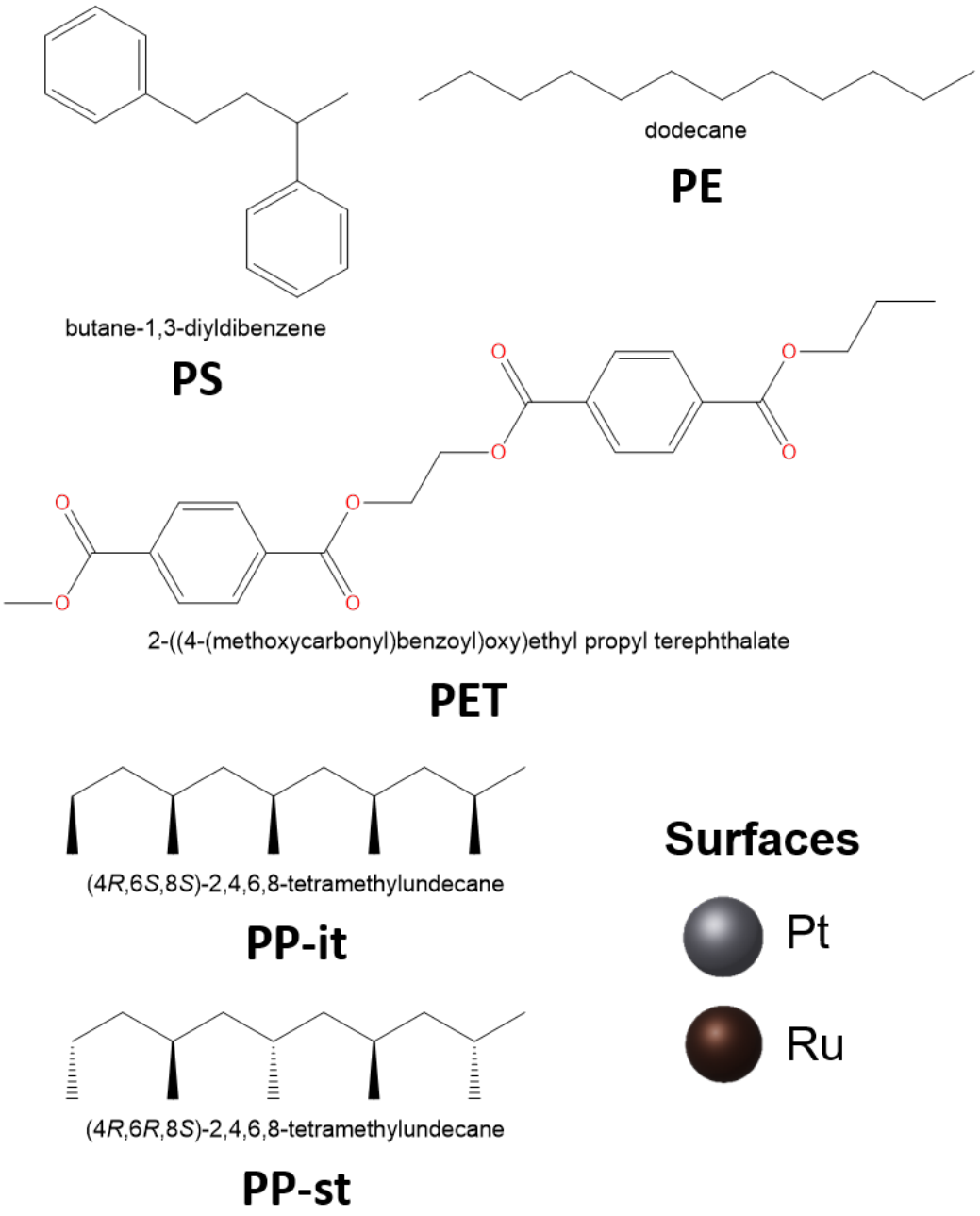


Figure S12: BM-dataset: Plastic molecules and metals.

Tables

Table S1: FG-dataset: Hydrocarbons, alcohols, aldehydes, ketones and ethers.

IUPAC name	**SMILES**
Formaldehyde	C=O
Acetylene	C#C
Ethylene	C=C
Acetaldehyde	CC=O
Ethane	CC
Dimethylether	COC
Ethanol	CCO
Propyne	C#CC
Propylene	C=CC
Acetone	CC(=O)C
Propane	CCC
Ethylmethylether	CCOC
N-propanol	CCCO
I-propanol	CC(C)O
2-butyne	CC#CC
1-butyne	C#CCC
1-butene	C=CCC
2-butene-cis	C/C=C\C
2-butene-trans	C/C=C/C
I-butene	CC(=C)C
Butanone	CCC(=O)C
N-butane	CCCC
I-butane	CC(C)C
N-butanol	CCCCO
2-butanol-R	C[C@H](CC)O
2-butanol-S	C[C@@H](CC)O
T-butanol	CC(C)(C)O
Propionaldehyde	CCC=O
Butyraldehyde	CCCC=O
Isobutyraldehyde	CC(C=O)C
Ethoxyethane	CCOCC
1-methoxypropane	CCCOC

Table S2: FG-dataset: Carbonates, carboxylic acids and esters.

IUPAC name	**SMILES**
Formic acid	C(=O)O
Carbonic acid	C(=O)(O)O
Acetic acid	C(=O)(C)O
Methyl formate	C(=O)OC
Methyl hydrogen carbonate	C(=O)(O)OC
Propionic acid	CCC(=O)O
Ethyl formate	C(=O)OCC
Methyl acetate	C(=O)(C)OC
Ethyl hydrogen carbonate	C(=O)(OCC)O
Dimethyl carbonate	C(=O)(OC)OC
Butyric acid	CCCC(=O)O
Isobutyric acid	CC(C(=O)O)C
Propyl formate	C(=O)OCCC
Isopropyl formate	C(=O)OC(C)C
Ethyl acetate	CC(=O)OCC
Methyl propionate	CCC(=O)OC
Propyl hydrogen carbonate	C(=O)(OCCC)O
Isopropyl hydrogen carbonate	C(=O)(O)OC(C)C
Ethyl methyl carbonate	C(=O)(OC)OCC

Table S3: FG-dataset: Amines and imines.

IUPAC name	**SMILES**
Methanimine	C=N
Methanamine	CN
Ethanimine	CC=N
Methylmethanimine	C=NC
Ethanamine	CCN
Dimethylamine	CNC
Propan-1-imine	CCC=N
N-ethylmethanimine	C=NCC
(E)-N-methylethanimine	C/C=N/C
(Z)-N-methylethanimine	C/C=N\C
Propan-2-imine	CC(=N)C
Propan-1-amine	CCCN
Propan-2-amine	CC(C)N
N-methylethanamine	CCNC
Trimethylamine	CN(C)C
Butan-1-imine	CCCC=N
2-methylpropan-1-imine	CC(C)C=N
N-propylmethanimine	C=NCCC
N-isopropylmethanimine	C=NC(C)C
(E)-N-ethylethanimine	C/C=N/CC
(Z)-N-ethylethanimine	C/C=N\CC
(E)-N-methylpropan-1-imine	CC/C=N/C
(Z)-N-methylpropan-1-imine	CC/C=N\C
Butan-2-imine	CCC(=N)C
N-methylpropan-2-imine	CC(=NC)C
Butan-1-amine	CCCCN
(R)-butan-2-amine	C[C@H](CC)N
(S)-butan-2-amine	C[C@@H](CC)N
2-methylpropan-2-amine	CC(C)(C)N
Diethylamine	CCNCC
N-methylpropan-1-amine	CCCNC
N,N-dimethylethanamine	CCN(C)C

Table S4: FG-dataset: Amidines.

IUPAC name	**SMILES**
Formimidamide	C(=N)N
N-methylformimidamide	C(=N)NC
acetimidamide	C(=N)(C)N
(E)-N'-methylformimidamide	C(=[NH2])[N]C
(Z)-N'-methylformimidamide	C(=[NH2])[N]C
N-ethylformimidamide	C(=N)NCC
Propionimidamide	C(=N)(CC)N
(E)-N'-ethylformimidamide	C(=[NH2])[N]CC
(Z)-N'-ethylformimidamide	C(=[NH2])[N]CC
(E)-N'-methylacetimidamide	C(=[NH2])(C)[N]C
(Z)-N'-methylacetimidamide	C(=[NH2])(C)[N]C
N,N-dimethylformimidamide	C(=N)N(C)C
N-methylacetimidamide	C(=N)(C)NC
(E)-N,N'-dimethylformimidamide	C(=N\C)/NC
(Z)-N,N'-dimethylformimidamide	C(=N\C)\NC
N-isopropylformimidamide	C(=N)NC(C)C
N-propylformimidamide	C(=N)NCCC
Isobutyrimidamide	C(=N)(C(C)C)N
Butyrimidamide	C(=N)(CCC)N
(E)-N'-isopropylformimidamide	C(=[NH2])[N]C(C)C
(Z)-N'-isopropylformimidamide	C(=[NH2])[N]C(C)C
(E)-N'-propylformimidamide	C(=[NH2])[N]CCC
(Z)-N'-propylformimidamide	C(=[NH2])[N]CCC
(E)-N'-ethylacetimidamide	C(=[NH2])(C)[N]CC
(Z)-N'-ethylacetimidamide	C(=[NH2])(C)[N]CC
(E)-N'-methylpropionimidamide	C(=[NH2])(CC)[N]C
(Z)-N'-methylpropionimidamide	C(=[NH2])(CC)[N]C
N-ethyl-N-methylformimidamide	C(=N)N(C)CC
N-methylpropionimidamide	C(=N)(CC)NC
N-ethylacetimidamide	C(=N)(C)NCC
N,N-dimethylacetimidamide	C(=N)(C)N(C)C
(E)-N'-ethyl-N-methylformimidamide	C(=N\CC)/NC
(Z)-N'-ethyl-N-methylformimidamide	C(=N\CC)\NC
(E)-N-ethyl-N'-methylformimidamide	C(=N\C)/NCC
(Z)-N-ethyl-N'-methylformimidamide	C(=N\C)\NCC
(E)-N,N,N'-trimethylformimidamide	C(=N\C)/N(C)C
(Z)-N,N,N'-trimethylformimidamide	C(=N\C)\N(C)C
(E)-N,N'-dimethylacetimidamide	C(=N\C)(\C)/NC
(Z)-N,N'-dimethylacetimidamide	C(=N\C)(/C)\NC

Table S5: FG-dataset: Thiols, thials, thioketones and thioethers.

IUPAC name	**SMILES**
Methanethial	C=S
Methanethiol	CS
Ethanethial	CC=S
Ethanethiol	CCS
Dimethylsulfane	CSC
Propanethial	CCC=S
Propane-2-thione	CC(=S)C
Propane-1-thiol	CCCS
Propane-2-thiol	CC(C)S
Ethyl(methyl)sulfane	CCSC
Butanethial	CCCC=S
2-methylpropanethial	CC(C)C=S
Butane-2-thione	CCC(=S)C
Butane-1-thiol	CCCCS
(R)-butane-2-thiol	C[C@H](CC)S
(S)-butane-2-thiol	C[C@@H](CC)S
2-methylpropane-2-thiol	CC(C)(C)S
Diethylsulfane	CCSCC
Methyl(propyl)sulfane	CCCSC

Table S6: FG-dataset: Amides.

IUPAC name	**SMILES**
Formamide	C(=O)N
N-methylformamide	C(=O)NC
Acetamide	C(=O)(C)N
N-ethylformamide	C(=O)NCC
Propionamide	C(=O)(CC)N
N,N-dimethylformamide	C(=O)N(C)C
N-methylacetamide	C(=O)(C)NC
N-isopropylformamide	C(=O)NC(C)C
N-propylformamide	C(=O)NCCC
Isobutyramide	C(=O)(C(C)C)N
Butyramide	C(=O)(CCC)N
N-ethyl-N-methylformamide	C(=O)N(C)CC
N-ethylacetamide	C(=O)(C)NCC
N-methylpropionamide	C(=O)(CC)NC
N,N-dimethylacetamide	C(=O)(C)N(C)C

Table S7: FG-dataset: Oximes.

IUPAC name	**SMILES**
Formaldehyde oxime	C=NO
(E)-acetaldehyde oxime	C(=N\O)/C
(Z)-acetaldehyde oxime	C(=N\O)\C
(E)-propionaldehyde oxime	C(=N\O)/CC
(Z)-propionaldehyde oxime	C(=N\O)\CC
Propan-2-one oxime	C(=NO)(C)C
(E)-butyraldehyde oxime	C(=N\O)/CCC
(Z)-butyraldehyde oxime	C(=N\O)\CCC
(E)-isobutyraldehyde oxime	C(=N\O)/C(C)C
(Z)-isobutyraldehyde oxime	C(=N\O)\C(C)C
(E)-butan-2-one oxime	C(=N\O)(\C)/CC
(Z)-butan-2-one oxime	C(=N\O)(\CC)/C

Table S8: FG-dataset: Carbamate esters.

IUPAC name	**SMILES**
carbamic acid	C(=O)(N)O
methylcarbamic acid	C(=O)(NC)O
methyl carbamate	C(=O)(N)OC
ethylcarbamic acid	C(=O)(NCC)O
ethyl carbamate	C(=O)(N)OCC
dimethylcarbamic acid	C(=O)(N(C)C)O
methyl methylcarbamate	C(=O)(NC)OC
isopropylcarbamic acid	C(=O)(NC(C)C)O
propylcarbamic acid	C(=O)(NCCC)O
isopropyl carbamate	C(=O)(N)OC(C)C
propyl carbamate	C(=O)(N)OCCC
ethyl(methyl)carbamic acid	C(=O)(N(CC)C)O
methyl ethylcarbamate	C(=O)(NCC)OC
ethyl methylcarbamate	C(=O)(NC)OCC
methyl dimethylcarbamate	C(=O)(N(C)C)OC

Table S9: FG-dataset: Aromatic molecules.

IUPAC name	**SMILES**
Furan	o1cccc1
Thiophene	s1cccc1
Pyrrole	[nH]1cccc1
Pyridine	c1ccncc1
Cyclopentadiene	C1C=CC=C1
Benzene	c1ccccc1
Phenol	Oc1ccccc1
Thiophenol	Sc1ccccc1
Aniline	Nc1ccccc1
Toluene	Cc1ccccc1
Para-xylene	Cc1ccc(C)cc1
Meta-xylene	Cc1cccc(C)c1
Ortho-xylene	Cc1ccccc1C
Benzofuran	o1ccc2ccccc12
Isobenzofuran	o1cc2ccccc2c1
Benzo[b]thiophene	s1ccc2ccccc12
Benzo[c]thiophene	s1cc2ccccc2c1
1H-indole	[nH]1ccc2ccccc12
2H-indole	C1C=C2C=CC=CC2=N1
Quinoline	c1ccc2ncccc2c1
Isoquinoline	c1ccc2cnccc2c1
1H-indene	C1C=Cc2ccccc12
2H-indene	C1C=C2C=CC=CC2=C1
Naphthalene	c1ccc2ccccc2c1

Table S10: Standard deviation and standard error of the mean of the prediction error of the models obtained by cross validation sorted by chemical family. Values are reported in eV.

Chemical Family	**Standard Deviation**	**Standard Error**
$C_xH_yO_{(0,1)}$	0.26	0.0073
$C_xH_yO_{(2,3)}$	0.13	0.0024
C_xH_yN	0.23	0.0027
C_xH_yS	0.26	0.0044
Amidines	0.23	0.0039
Amides	0.15	0.0046
Oximes	0.19	0.0043
Carbamates	0.16	0.0035
Aromatics	0.48	0.0102

Table S11: Standard deviation and standard error of the mean of the prediction error of the models obtained by cross validation sorted by metal (the last row considers the gas-phase subset). Values are reported in eV.

Metal	**Standard Deviation**	**Standard Error**
Ag	0.23	0.0104
Au	0.16	0.0080
Cd	0.10	0.0041
Cu	0.12	0.0076
Ir	0.17	0.0065
Ni	0.14	0.0038
Os	0.20	0.0051
Pd	0.15	0.0051
Pt	0.19	0.0055
Rh	0.15	0.0080
Ru	0.22	0.0106
Zn	0.11	0.0045
Gas	0.22	0.0078

Table S12: Summary of the performed hyperparameter optimization. All the pooling-wise options refer to the GMT implementation in Pytorch Geometric (PyG).

Hyperparameter	**Type**	**Search space**	**Optimum**
Batch size	int	16, 32, 64, 128	16
Loss function	func	MAE, MSE	MAE
Initial lr	float	1e-1, 1e-2, 1e-3, 1e-4	1e-3
Lr-patience	int	5, 7, 9	5
Lr-factor	float	0.5, 0.7, 0.9	0.7
Minimum lr	float	1e-7, 1e-8, 1e-9	1e-8
Amsgrad	bool	True, False	True
Layers depth	int	64, 128, 256	256
Bias inclusion	bool	True, False	False
Linear layers	int	0, 1, 2, 3, 4	1
Convolutional layers	int	1, 2, 3, 4, 5	3
Convolution type	func	GraphSAGE, GATv2	GraphSAGE
Normalized conv.	bool	True, False	False
Root-weighted conv.	bool	True, False	True
Pool ratio	float	0.25, 0.50, 0.75	0.25
Pool heads	int	1, 2, 4	1
Pool sequence	list[func]	(see PyG Docs)	[GMPool_I]
Pool normalization	bool	True, False	False

Table S13: Adsorption energy of BM-dataset samples obtained with the optimized GNN and compared to DFT. Values are reported in eV.

Molecule	**Family**	**Metal**	E_{ads}^{DFT}	E_{ads}^{GNN}	**Absolute error**
mol1	Polyurethanes	Ag	-0.75	0.12	0.87
mol2	Polyurethanes	Ag	-0.81	-0.77	0.04
mol3	Polyurethanes	Ag	-0.77	-0.40	0.38
mol4	Polyurethanes	Ag	-0.84	-1.06	0.22
mol5	Polyurethanes	Ag	-1.11	-1.20	0.09
mol1	Polyurethanes	Au	-0.50	-0.12	0.38
mol2	Polyurethanes	Au	-0.40	-0.84	0.44
mol3	Polyurethanes	Au	-0.36	-0.19	0.17
mol4	Polyurethanes	Au	-0.39	0.00	0.39
mol5	Polyurethanes	Au	-0.84	-0.60	0.24
PE	Plastics	Pt	-1.17	-0.72	0.45
PP-it	Plastics	Pt	-0.93	0.00	0.93
PP-st	Plastics	Pt	-0.86	0.00	0.86
PS	Plastics	Pt	-2.95	-3.41	0.45
PET	Plastics	Pt	-2.86	-1.89	0.97
PE	Plastics	Ru	-0.18	-0.04	0.15
PP-it	Plastics	Ru	0.02	0.00	0.02
PP-st	Plastics	Ru	0.08	0.00	0.08
PS	Plastics	Ru	-3.10	-3.47	0.36
PET	Plastics	Ru	-4.78	-4.16	0.63
mol1	Biomass	Ni	-2.86	-2.51	0.36
mol2	Biomass	Ni	-2.39	-2.18	0.21
mol3	Biomass	Ni	-2.79	-1.64	1.15
mol4	Biomass	Ni	-1.92	-1.34	0.58
mol5	Biomass	Ni	-1.60	-1.51	0.09
mol1	Biomass	Ru	-3.65	-3.58	0.07
mol2	Biomass	Ru	-3.15	-2.95	0.20
mol3	Biomass	Ru	-3.41	-2.41	1.00
mol4	Biomass	Ru	-2.50	-2.16	0.33
mol5	Biomass	Ru	-1.99	-1.44	0.55

References

(1) García-Muelas, R.; López, N. *J. Chem. Phys. C* **2014**, *118*, 17531–17537, DOI: 10.1021/jp502819s.

(2) García-Muelas, R.; Li, Q.; López, N. *ACS Catalysis* **2015**, *5*, 1027–1036, DOI: 10.1021/cs501698w.

(3) Weininger, D. *Journal of Chemical Information and Modeling* **1988**, *28*, 31–36, DOI: 10.1021/ci00057a005.

(4) O'Boyle, N. M.; Morley, C.; Hutchison, G. R. *Chemistry Central Journal* **2008**, *2*, 5, DOI: 10.1186/1752-153x-2-5.

(5) Halgren, T. A. *Journal of Computational Chemistry* **1996**, *17*, 490–519, DOI: https://doi.org/10.1002/(SICI)1096-987X(199604)17:5/6<490::AID-JCC1>3.0.CO;2-P.

(6) Ong, S. P.; Richards, W. D.; Jain, A.; Hautier, G.; Kocher, M.; Cholia, S.; Gunter, D.; Chevrier, V. L.; Persson, K. A.; Ceder, G. *Computational Materials Science* **2013**, *68*, 314–319, DOI: 10.1016/j.commatsci.2012.10.028.

(7) Morandi, S. DFT data repository, DOI: 10.19061/iochem-bd-1-257.

(8) Álvarez-Moreno, M.; de Graaf, C.; López, N.; Maseras, F.; Poblet, J. M.; Bo, C. *Journal of Chemical Information and Modeling* **2014**, *55*, 95–103, DOI: 10.1021/ci500593j.

(9) Isayev, O.; Oses, C.; Toher, C.; Gossett, E.; Curtarolo, S.; Tropsha, A. *Nat. Commun.* **2017**, *8*, 15679, DOI: 10.1038/ncomms15679.

(10) Cordero, B.; Gómez, V.; Platero-Prats, A. E.; Revés, M.; Echeverría, J.; Cremades, E.; Barragán, F.; Alvarez, S. *Dalton Transactions* **2008**, *nil*, 2832, DOI: 10.1039/b801115j.

(11) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V., et al. *Journal of machine learning research* **2011**, *12*, 2825–2830.

(12) Hamilton, W. L.; Ying, R.; Leskovec, J. Inductive Representation Learning on Large Graphs, 2017, DOI: 10.48550/ARXIV.1706.02216.

(13) Baek, J.; Kang, M.; Hwang, S. J. Accurate Learning of Graph Representations with Graph Multiset Pooling, 2021, DOI: 10.48550/ARXIV.2102.11533.

(14) Agarap, A. F. *CoRR* **2018**, DOI: https://doi.org/10.48550/arXiv.1803.08375.

(15) Kingma, D. P.; Ba, J. *CoRR* **2014**, DOI: https://doi.org/10.48550/arXiv.1412.6980.

(16) Li, Q.; López, N. *ACS Catalysis* **2018**, *8*, 4230–4240, DOI: 10.1021/acscatal.8b00067.

(17) Puértolas, B.; Rellán-Piñeiro, M.; Núñez-Rico, J. L.; Amrute, A. P.; Vidal-Ferran, A.; López, N.; Pérez-Ramírez, J.; Wershofen, S. *ACS Catalysis* **2019**, *9*, 7708–7720, DOI: 10.1021/acscatal.9b02086.

(18) Ding, S.; Hülsey, M. J.; Pérez-Ramírez, J.; Yan, N. *Joule* **2019**, *3*, 2897–2929, DOI: 10.1016/j.joule.2019.09.015.

(19) Li, L.; Jamieson, K.; Rostamizadeh, A.; Gonina, E.; Hardt, M.; Recht, B.; Talwalkar, A. **2018**, DOI: 10.48550/ARXIV.1810.05934.

(20) Liaw, R.; Liang, E.; Nishihara, R.; Moritz, P.; Gonzalez, J. E.; Stoica, I. *arXiv preprint arXiv:1807.05118* **2018**.