

Supporting Information for Deep Learning Metal Complex Properties with Natural Quantum Graphs

Hannes Kneiding,^a Ruslan Lukin,^a Lucas Lang,^a Simen Reine,^a Thomas Bondo
Pedersen,^a Riccardo de Bin,^b David Balcells^{a,*}

^a*Hylleraas Centre for Quantum Molecular Sciences, Department of Chemistry, University
of Oslo, P.O. Box 1033, Blindern, 0315 Oslo, Norway;* ^b*Department of Mathematics,
University of Oslo, P. O. Box 1053, Blindern, Oslo, Norway*

E-mail: david.balcells@kjemi.uio.no

Table of Contents

tmQMg dataset statistics	S3
tmQMg dataset outliers	S7
tmQMg computational details	S8
GNN models details	S8
Baseline representation	S13
Linear fitting for energy targets	S14
Training dataset results	S15
Python libraries	S17
References	S18

tmQMg dataset statistics

Figure S1 shows the distribution of three different properties over the tmQMg dataset; *i.e.* the overall charge of the metal complexes, their size in number of atoms, and their chemical composition. These distributions show that the transition metal group with the largest weight is 10 (Ni-Pd-Pt), followed by 6, 8, 9, and 12, which have more similar weights; the most scarce metals in the dataset are those from group 3. The majority of complexes, 82%, are neutral, whereas only 15% and 3% are cationic or anionic, respectively. These numbers reflect the experimental biases introduced by the CSD database, which was used as the source of TMCs. Regarding the molecular charge, it is straightforward to balance the distribution of this property by means of generative models leveraging the tmQMg dataset. Further, it should be noted that the charge is one of the graph-level features passed as input to the predictive models. Other properties are more evenly distributed, including the size in number of atoms (55% are in the 36-to-65 range) and the periodic series to which the transition metal belongs (37% are 3d, and smaller but similar portions are 4d or 5d). The analysis of the chemical composition reflects the organic nature of the ligands, with 88% of the elements being either C or H. Metals aside, the most abundant heteroatom is N, followed, in this order, by O, Cl, and P. The selection of 60k TMCs in the tmQMg dataset excludes ~ 0.44 M compounds of the CSD that also contain metals – a significant proportion of them are polynuclear species and nanoporous materials far from the type of compounds targeted in this dataset.

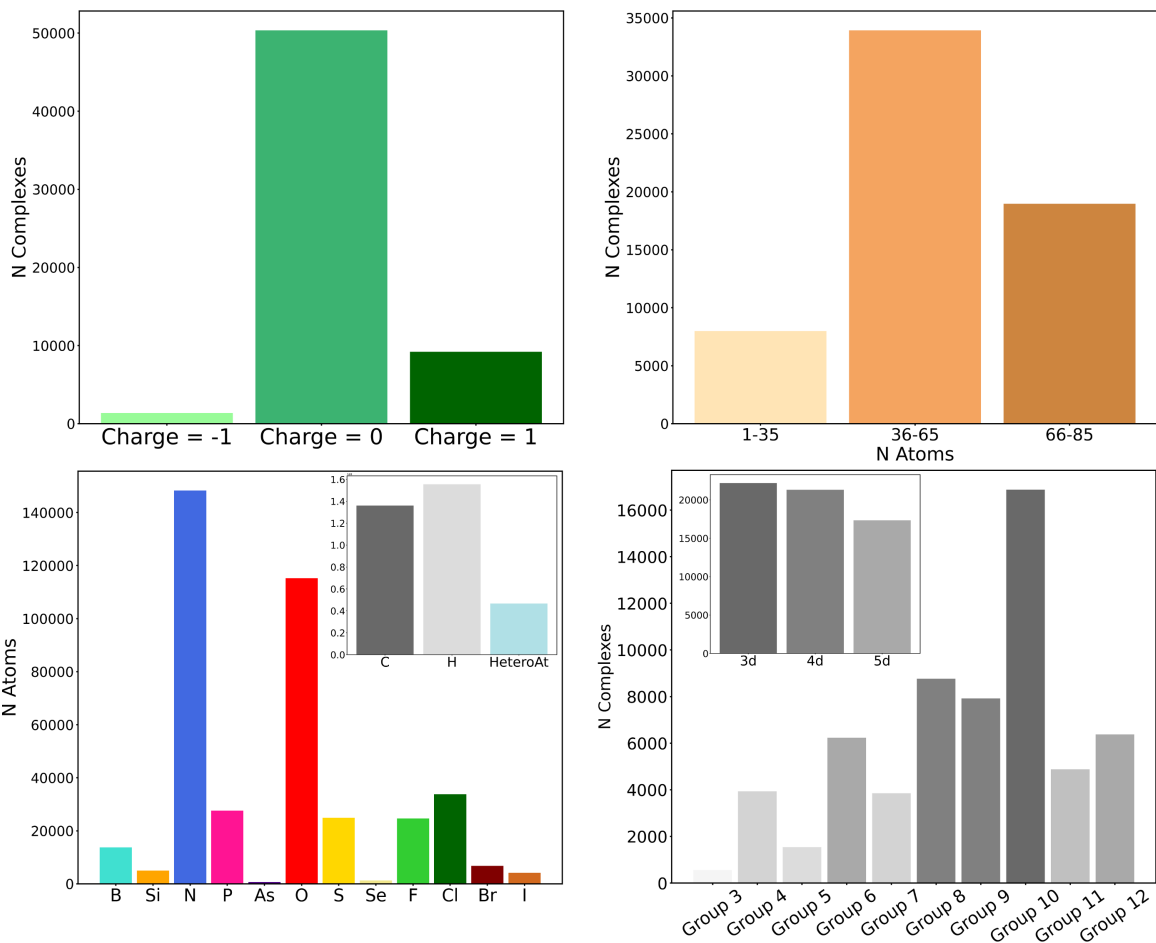


Figure S1: Top: Histograms showing the distribution of molecular charges (left) and size in atoms (right); Bottom: Histograms showing the composition distribution by non-metal elements (left) and metal elements (right). The y axis of the non-metal composition inset is in the 10^6 scale.

The pair plot shown in Figure S2 suggests that most quantum properties have a distribution close to normal, though some appear significantly skewed (*e.g.* lowest vibrational frequency) or with multiple peaks (*e.g.* metal charge). Further, many quantum properties are poorly correlated, with pairwise scatter plot distributions taking the appearance of irregular blobs. Nonetheless, there are distributions reflecting interesting trends; *e.g.*, in Figure S3 there is a poor correlation between the HOMO-LUMO gap (ϵ) and the polarizability (α), though it is apparent that the region in which both ϵ and α are large is sparsely populated. A similar scenario is seen in the scatter plot of entropy *versus* lowest vibrational frequency. Figures S2 and S3 also show examples of strong correlations, including that of the total

number of electrons with the polarizability, heat capacity, and entropy.

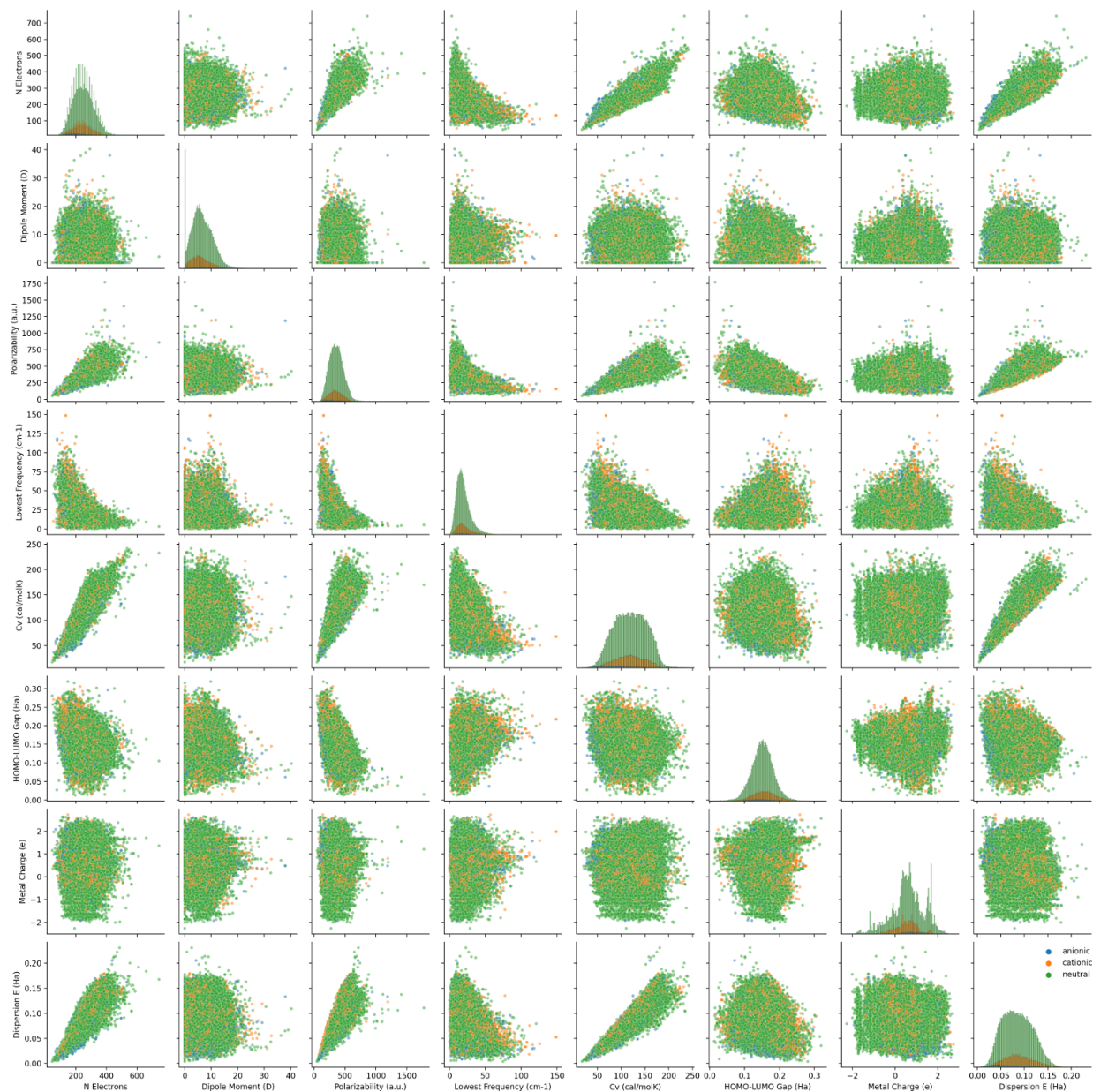


Figure S2: Pair plot of selected quantum properties, with scatter (off-diagonal) and histogram (diagonal) plots. Left-to-right (and top-to-bottom): Number of electrons, dipole moment, isotropic polarizability, lowest vibrational frequency, heat capacity at constant volume, HOMO-LUMO gap, natural charge of the metal center, and dispersion energy. The color code is orange for $q = +1$, green for $q = 0$, and blue for $q = -1$, where q is the overall charge of each TMC.

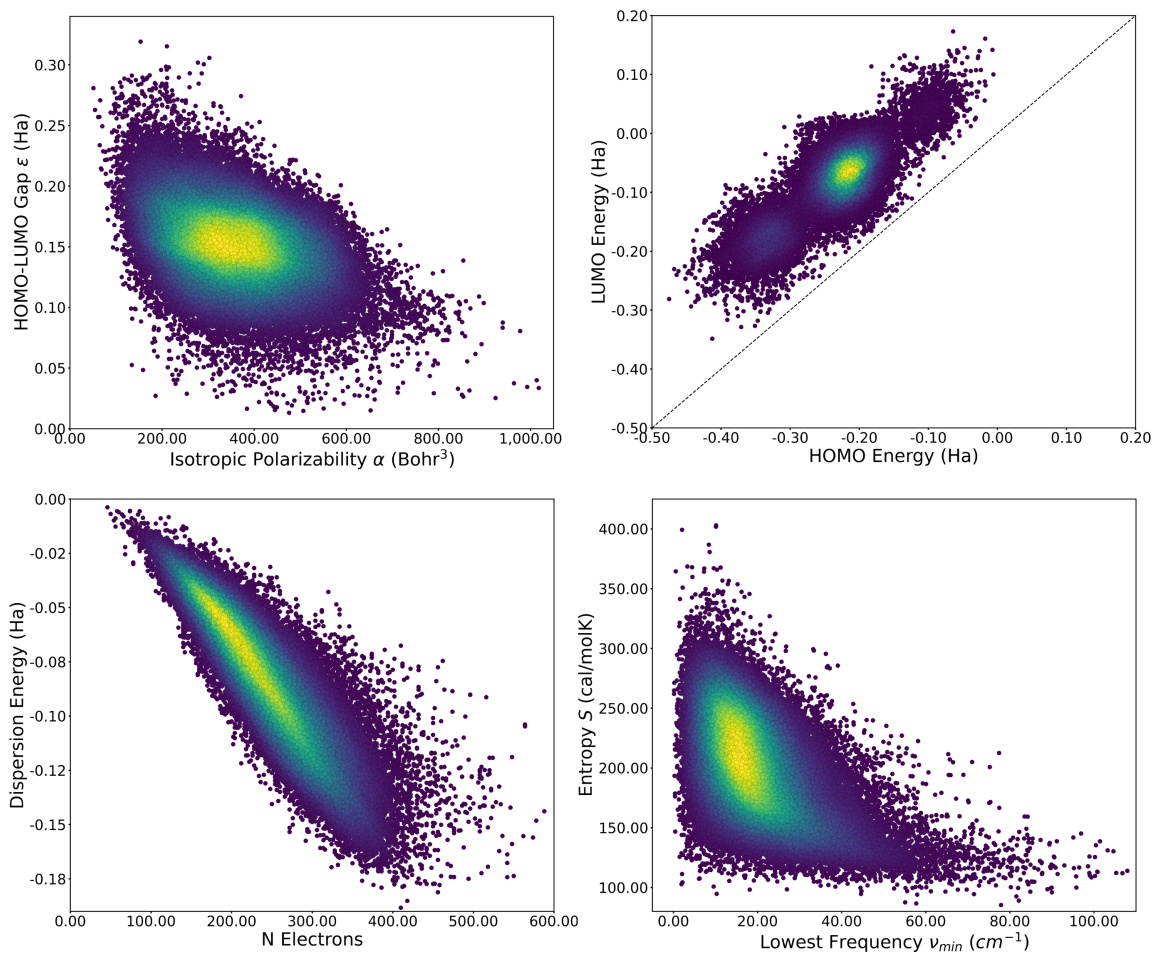


Figure S3: Pairwise scatter plots of selected quantum properties. The color transition from dark purple to light yellow reflects the increase in the density of data points.

tmQMg dataset outliers

In order to exclude the potentially faulty TMCs of the dataset, we employed outlier detection methods. We explored both parametric (elliptic envelope) and non-parametric (isolation forests) methods, which were applied on a per-metal basis; *i.e.* we stratified the dataset by metal center and applied outlier detection to each individual stratum. Both tested methods seemed to provide similar results when considering only a subset of all quantum properties individually. However, the formal requirement of a Gaussian distribution for the parametric methods was not satisfied by all strata. Therefore, we ruled out the elliptic envelope in favor of the isolation forests. Using the implementation of *scikit-learn*,¹ we applied a contamination value of 0.04 to exclude 2,390 outliers (*i.e.* $\sim 4\%$ of tmQMg).

We compared the performance of the MPNN GNN predicting the polarizability upon training the model with the tmQMg dataset including and excluding outliers. We saw an increase in model accuracy when excluding the outliers ($\Delta\text{MAE} \approx 1 \text{ Bohr}^3$ after 150 epochs), as illustrated by the learning curves shown in Figure S4.

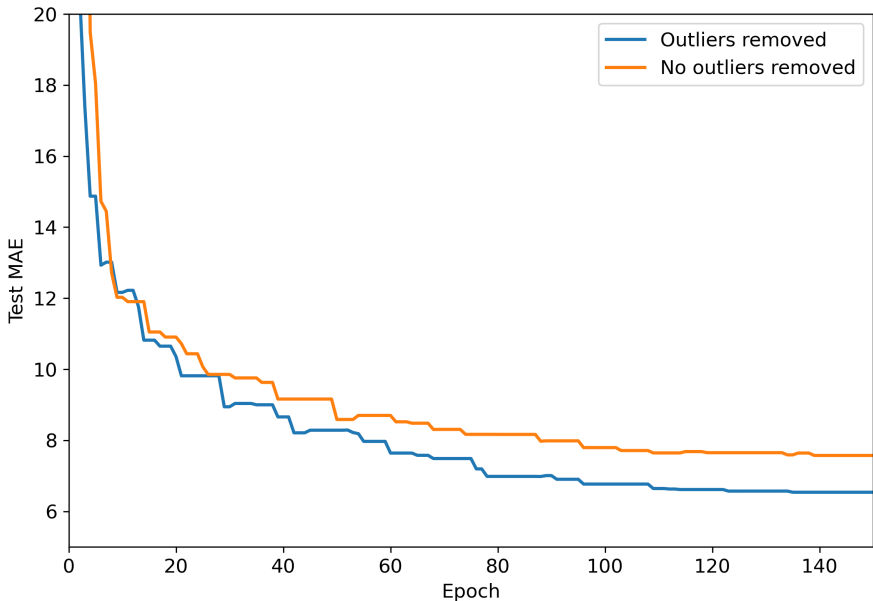


Figure S4: MAE learning curves on the test set using the dataset with and without outliers in the prediction of the polarizability.

tmQMg computational details

DFT calculations for the tmQMg dataset were carried out with the Gaussian16 program, using revision C.01. Geometry optimizations were done without any geometry or symmetry constraint using the pure PBE functional with the D3BJ dispersion correction and the double- ζ def2SVP basis set. Frequencies were computed at the same level of theory, together with the thermochemistry at $T = 298.15$ K and $p = 1$ atm. Only the TMCs yielding converged geometries without imaginary frequencies were included in the dataset. Single-point energies and properties were computed for the optimized geometries with the hybrid PBE0 functional, using the D3BJ dispersion correction and the triple- ζ def2TZVP basis set. The natural bond orbital analysis was also carried out at this level using the NBO7 program. All calculations were performed for the singlet state.

GNN models details

MPNN architecture

The tmQMg data was split into training, validation and testing sets with a 80:10:10 ratio. All categorical attributes, except the atomic number, were converted into one-hot encoded features (*e.g.*, for the type of NBO defining the edges: $(1, 0) = \text{BD}$, $(0, 1) = \text{3C}$). The values (x) of the node, edge, and whole-graph attributes, as well as those of the targets, were all standard-scaled before training with the z factor

$$z = \frac{x - \mu}{\sigma}$$

where μ and σ are the mean and the standard deviation, respectively. Standard-scaling facilitated the optimization of the model parameters (*i.e.* weights and biases minimizing the loss function). For most extensive properties, we applied a linear fitting procedure prior to training the model (*vide infra*).

The MPNN \oplus G GNN is a message passing neural network based on an architecture introduced by Gilmer et al.² In the first layer, it maps the node attribute vectors to a chosen embedding of size d (usually much larger than the input dimension) by passing the vectors through a two-layer multilayer perceptron (MLP). In parallel, the edge attribute vectors are passed through a similar neural network expanding and reshaping it to a $d \times d$ matrix. In the next layer, the embedded attribute vectors enter the message passing scheme defined by

$$\mathbf{m}(\mathbf{h}_v, \mathbf{h}_w, \mathbf{e}_{vw}) = \mathcal{A}(\mathbf{e}_{vw})\mathbf{h}_w \quad (1)$$

where \mathbf{m} is the message from node v to node w , \mathbf{h}_v and \mathbf{h}_w are the embedding vectors of nodes v and w , \mathbf{e}_{vw} is the edge connecting node v to node w , and \mathcal{A} denotes the neural network that maps the edge vector \mathbf{e}_{vw} to a $d \times d$ matrix. The messages are then aggregated for each node v by considering the neighboring nodes w , using the mean function. In the next layer, the aggregated message is passed to a gated recurrent unit, which, for k iterations (*i.e.* the model depth), updates the node embeddings. The resulting graph is fed to the set2set layer,³ which implements an attention mechanism over m iterations and pools the final graph embeddings into a single fixed-dimension vector. In the last layer, this vector is appended to the whole-graph attribute vector, in a modification of the architecture originally reported by Gilmer et al.,² and is passed to a two-layer MLP yielding the target prediction. In the article, the models with and without whole-graph attribution are referred to as MPNN \oplus G and MPNN, respectively; both use parameter-sharing throughout the message passing layers.

Table S1 shows the hyperparameters used to setup the MPNN \oplus G and MPNN models. Note that the number of node, edge, and graph attributes is dictated by the nature of the corresponding representation. For training, the Adam optimiser⁴ with an initial learning rate of 0.001 was used. In addition, an ‘‘on-plateau’’ learning rate decay was used with a patience of 5, reduction factor of 0.7, and a minimum learning rate of 0.00001. All models were trained using early-stopping and a batch size of 32.

Table S1: Hyperparameters used for the different graph representations.

Parameter	baseline	u-NatQG	d-NatQG
Number of node attributes	4	21	21
Number of edge attributes	2	18	26
Number of graph attributes	4	4	4
Embedding dimension d	128	128	128
Message passing depth k	4	4	6
set2set steps m	4	4	4

MPNN hyperparameter optimisation

Exploratory hyperparameter optimisation was performed for the *u-NatQG* and *d-NatQG* representations separately. The main focus was on the embedding dimension d , the message passing depth k , and the number of set2set steps m (Table S1). The effect of different batch sizes was also investigated but the resulting differences were insignificant and, therefore, the batch size of 32 was chosen for the sake of computational efficiency.

In general, for the embedding size we found that a larger value for d generally corresponds to better performance which is due to the resulting increased expressivity of the model. We tested this starting from 32 going up to 128, with the latter giving the best results. This however does not scale infinitely because there are diminishing returns as the model starts suffering from the curse of dimensionality. In fact, going from embedding size 32 to 64 gives a larger performance increase than going from 64 to 128.

For optimising the number of message passing steps k , we kept the number of set2set iterations constant at $m = 4$. The corresponding results suggest that after a certain point there is very little variance in performance with respect to the number of message passing steps, as shown in Figure S5 for the *d-NatQG* model predicting the polarizability. Among the tested settings, $k = 6$ performed best for *d-NatQG*. Convergence was faster for *u-NatQG*, for which $k = 4$ yielded the highest accuracy.

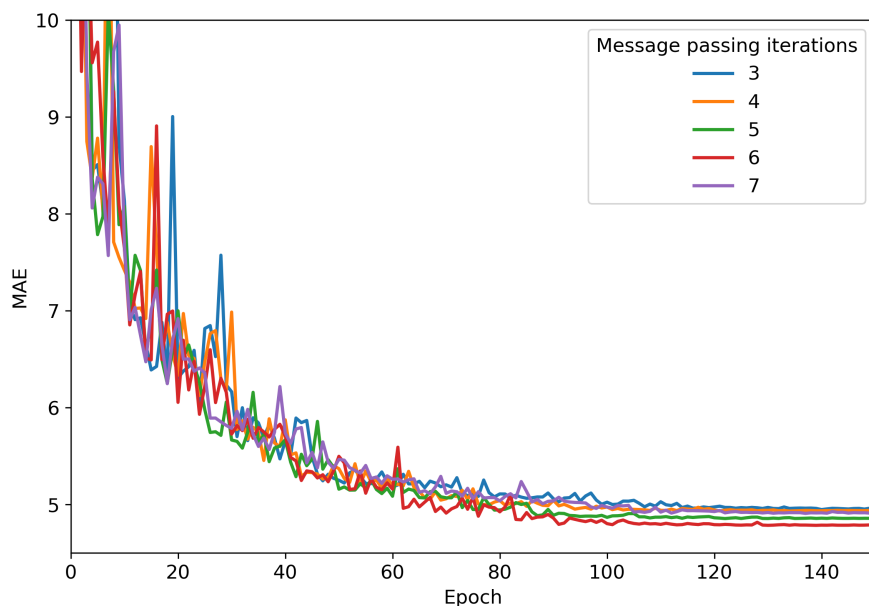


Figure S5: MAE learning curves for the validation set using the d - $NatQG$ representation and different number of message passing steps. Target = polarizability; embedding dimension $d = 128$; set2set iterations $m = 4$.

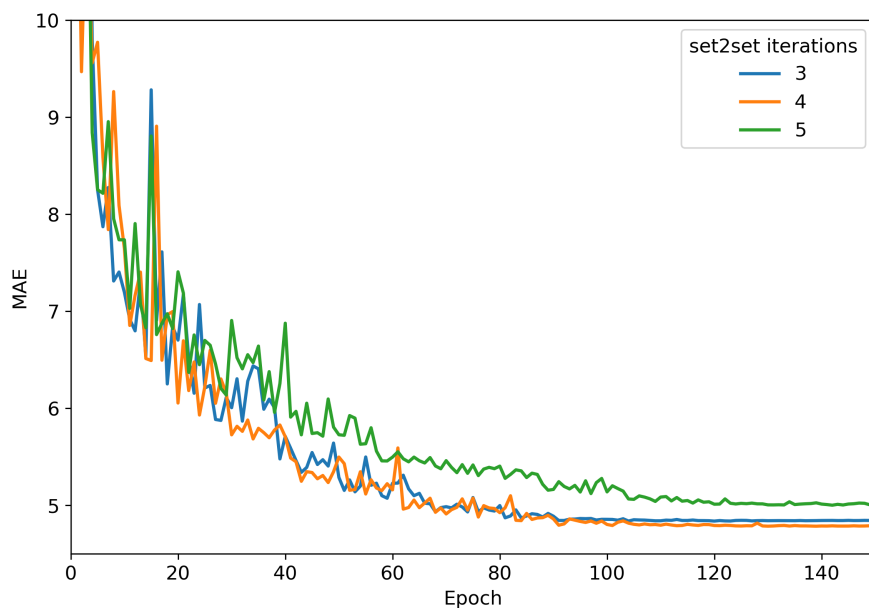


Figure S6: MAE learning curves for the validation set using the d - $NatQG$ representation and different number of set2set iterations. Target = polarizability; embedding dimension $d = 128$; number of message passing steps $k = 6$.

Next, we investigated the number of set2set iterations keeping the number of message passing steps constant at $k = 6$. In this case, there was a significant difference in performance: both $m = 3$ and $m = 5$ lead to an increase of the MAE by *ca.* 0.2 Bohr^3 relative to $m = 4$, as shown in Figure S6 for *d-NatQG*. This m value was also optimal for *u-NatQG*.

The MPNN hyperparameters used for the baseline were those derived for the *u-NatQG* representation, which has the same graph topology (*vide infra*).

MXMNet and CRG models

In all CRG models, we used a cutoff of 10 \AA to define the radial graphs. This value was also used to set the global layer of the MXMNet model. The models were trained by combining the exponential moving average method with a learning rate (lr) scheduler starting from $lr = 0.0001$, and a batch size of 32. The model hyperparameters were the following:

- MXMNet.⁵ Number of hidden channels = 128, layers = 3.
- SchNet.⁶ Number of hidden channels = 256, filters = 256, interactions = 6, Gaussian functions = 100; maximum number of neighbors = 32.
- EdgeUpdate.⁷ Number of hidden channels = 256, interactions = 6, Gaussian functions = 100.
- DimeNet++.⁸ Number of hidden channels = 128, interactions = 4, spherical harmonics = 7, radial basis functions = 6; basis embedding size = 8.
- ALIGNN.⁹ Number of atom input features = 92, edge input features = 80, triplet input features = 40, embedding features = 64, hidden features = 256.

Baseline representation

The baseline graph representation was based on the following three building blocks:

- Topology – The same used for the *u-NatQG* graphs.
- Node attributes – The (Z, T, S, χ) vector.
- Edge attributes – The (BO, d) vector.

Most of the data needed to inform the attribute vectors was immediately available; *i.e.* the atomic number (Z), the node degree (T), the electronegativity (χ), and the bond distance (d). However, the definition of the covalent radii (S) and the bond order (BO) were more complicated. For the covalent radii, we used the values derived by Alvarez¹⁰ based on a systematic approach for all elements up to $Z = 96$ that exploits the crystallographic data of the CSD. For elements with multiple values, the average radius was used. For the bond order, it was not possible to use cheminformatics software like RDKit, which, for many TMCs, cannot produce a graph containing bond orders from an xyz geometry or a SMILES string that, in some cases, cannot be generated either. The bond orders were thus derived from the Wiberg bond indices (WBI) available from the NBO data. Based on the results shown in Table S2, we defined the following intervals to map the WBI values into integer bond orders:

- $WBI \leq 1.4 \rightarrow BO = 1$
- $1.4 < WBI \leq 2.0 \rightarrow BO = 2$
- $WBI > 2.0 \rightarrow BO = 3$

Table S2: Wiberg bond orders of selected bonds in four different organic molecules calculated at the DFT (PBE/def2SVP) level. For butane, 2-butene and, 2-butyne, the reported bond order corresponds to the central bond. For benzene it is the average of the six C–C bonds.

Molecule	Wiberg bond order
Butane	1.01
2-Butene	1.91
2-Butyne	2.74
Benzene	1.43

Linear fitting for energy targets

In order to increase performance when predicting energy targets, we used a linear fitting procedure to extract the atomic contributions to the total energy. We assumed that the total energy can be separated into atomic contributions and their interactions; *i.e.*:

$$E = E_{atom} + E_{inter} \quad (2)$$

where we defined the atomic contributions as

$$E_{atom} = \sum_{A \in M} N_A E_A \quad (3)$$

where A denotes a specific element and N_A and E_A denote the number of atoms for this element in the TMC molecule M and its atomic contribution to the overall energy, respectively.

We can re-write this expression in vector form as follows:

$$\mathbf{E}_{atom} = \mathbf{N}_A \mathbf{E}_A \quad (4)$$

where $\mathbf{E}_{atom} \in \mathbb{R}^n$ is a vector containing the atomic contribution part of the energy for n TMCs, $\mathbf{E}_A \in \mathbb{R}^m$ is a vector containing the atomic energies per element, and $\mathbf{N}_A \in \mathbb{R}^{n \times m}$ is the element count matrix that contains the occurrences of each of the m elements for each

of the n TMCs.

The E_A values can be determined through linear regression on a training set containing n' TMCs:

$$\hat{\mathbf{E}}_A = \mathbf{N}^{train+} \mathbf{E}^{train} \quad (5)$$

where $\hat{\mathbf{E}}_A \in \mathbb{R}^m$ is a vector containing the predicted atomic energies of the m elements present in the dataset, $\mathbf{E}^{train} \in \mathbb{R}^{n'}$ is a vector containing the target energies of the n' TMCs used for training, and $\mathbf{N}^{train+} \in \mathbb{R}^{m \times n'}$ is the pseudo inverse of the element count matrix.

Using the predicted element-wise atomic energies, $\hat{\mathbf{E}}_A$, we can calculate the atomic contributions for the whole dataset, E_{atom} , using equation (4). We then obtain the interaction contribution with

$$E_{inter} = E - E_{atom} \quad (6)$$

In this framework, all GNN models were optimized for predicting only the interaction energy contributions. These values were thereafter added to the linearly-fitted atomic contributions to obtain the true target energy. This procedure was used to predict the electronic, dispersion, zero-point, enthalpy, and Gibbs energies. In principle, this procedure should be useful for all extensive properties though, for some of them, *e.g.* polarizability, we did not see a significant improvement.

Training dataset results

Table S3 provides the error metrics of the GNN models for the training dataset, including the prediction of the HOMO-LUMO gap, polarizability, and dipole moment. Table S4 contains the same information for all other targets, including also the baseline results.

Table S3: Mean absolute error (MAE) and r^2 score of the GNN models for the prediction of the HOMO-LUMO gap (in mHa) polarizability (in Bohr³) and dipole moment (in D) in the training dataset. The GNN architectures were based on different graphs including the (*u*-) and (*d*-*NatQG*) graphs and graphs derived from a cutoff radius (CRG). The base MXMNet model refers to the original implementation of Xie.⁵

Architecture and Graph		HOMO-LUMO gap		Polarizability		Dipole moment	
		MAE	r^2	MAE	r^2	MAE	r^2
MPNN	<i>u</i> - <i>NatQG</i>	$4.02 \cdot 10^{-2}$	0.999	$1.72 \cdot 10^{-1}$	0.999	$1.04 \cdot 10^{-2}$	0.999
	<i>d</i> - <i>NatQG</i>	$8.53 \cdot 10^{-2}$	0.999	$7.20 \cdot 10^{-2}$	0.999	$9.22 \cdot 10^{-3}$	0.999
MPNN \oplus G	<i>u</i> - <i>NatQG</i>	$6.02 \cdot 10^{-2}$	0.999	$9.25 \cdot 10^{-2}$	0.999	$2.25 \cdot 10^{-2}$	0.999
	<i>d</i> - <i>NatQG</i>	$8.60 \cdot 10^{-2}$	0.999	$7.82 \cdot 10^{-2}$	0.999	$1.67 \cdot 10^{-2}$	0.999
MXMNet	base	$1.31 \cdot 10^{-1}$	0.995	$2.71 \cdot 10^{-3}$	1.000	$1.99 \cdot 10^{-3}$	0.999
	<i>u</i> - <i>NatQG</i>	$1.59 \cdot 10^{-1}$	0.991	$2.24 \cdot 10^{-3}$	1.000	$2.22 \cdot 10^{-3}$	0.990
	<i>d</i> - <i>NatQG</i>	$3.42 \cdot 10^{-1}$	0.978	$3.58 \cdot 10^{-2}$	0.999	$3.51 \cdot 10^{-3}$	0.999
SchNet	CRG	1.60	0.989	$3.09 \cdot 10^{-2}$	1.000	$6.80 \cdot 10^{-4}$	1.000
EdgeUpdate	CRG	$2.31 \cdot 10^{-1}$	0.998	$1.02 \cdot 10^{-2}$	0.999	$1.02 \cdot 10^{-1}$	0.998
DimeNet++	CRG	$1.47 \cdot 10^{-1}$	0.992	$1.83 \cdot 10^{-3}$	1.000	$1.21 \cdot 10^{-3}$	0.999
ALIGNN	CRG	$2.14 \cdot 10^{-1}$	0.999	$3.90 \cdot 10^{-1}$	0.999	$1.03 \cdot 10^{-1}$	0.998

Table S4: MAE and r^2 score for the training dataset using the MPNN \oplus G model based on the *NatQG* graphs and a baseline representation including only generic properties (*i.e.* Z, T, S, χ , BO, and d). The units are mHa for all energies, cal/mol·K for the heat capacity and entropy, D for the dipole moment, Bohr³ for the polarizability, and cm⁻¹ for the largest vibrational frequency.

Property	Baseline		u-NatQG		d-NatQG	
	MAE	r^2	MAE	r^2	MAE	r^2
HOMO-LUMO gap	$1.64 \cdot 10^{-1}$	1.000	$6.02 \cdot 10^{-2}$	0.999	$8.60 \cdot 10^{-2}$	1.000
Polarizability	$3.89 \cdot 10^{-1}$	1.000	$9.25 \cdot 10^{-2}$	0.999	$7.82 \cdot 10^{-2}$	1.000
Dipole moment	$7.01 \cdot 10^{-1}$	0.928	$2.25 \cdot 10^{-2}$	0.999	$1.67 \cdot 10^{-2}$	1.000
HOMO energy	3.66	0.988	$3.51 \cdot 10^{-2}$	1.000	$3.93 \cdot 10^{-2}$	1.000
LUMO energy	$6.57 \cdot 10^{-1}$	1.000	$4.91 \cdot 10^{-2}$	1.000	$4.97 \cdot 10^{-2}$	1.000
Electronic energy ^a	5.45	1.000	$7.52 \cdot 10^{-2}$	1.000	$2.01 \cdot 10^{-1}$	1.000
Dispersion energy ^a	$1.39 \cdot 10^{-1}$	1.000	$1.79 \cdot 10^{-2}$	1.000	$1.64 \cdot 10^{-2}$	1.000
Zero-point energy ^a	$2.07 \cdot 10^{-2}$	1.000	$4.61 \cdot 10^{-3}$	1.000	$2.92 \cdot 10^{-3}$	1.000
Enthalpy energy ^a	4.30	1.000	$1.06 \cdot 10^{-1}$	1.000	$9.39 \cdot 10^{-2}$	1.000
Heat capacity ^b	$9.52 \cdot 10^{-2}$	1.000	$5.40 \cdot 10^{-2}$	1.000	$2.51 \cdot 10^{-2}$	1.000
Entropy energy	$2.21 \cdot 10^{-1}$	1.000	$4.65 \cdot 10^{-2}$	1.000	$5.04 \cdot 10^{-2}$	1.000
Gibbs energy ^a	2.63	1.000	$1.14 \cdot 10^{-1}$	1.000	$1.78 \cdot 10^{-1}$	1.000
Thermodynamic corrections ^c	$5.44 \cdot 10^{-1}$	1.000	$2.60 \cdot 10^{-1}$	1.000	$1.37 \cdot 10^{-1}$	1.000
Largest vibrational freq.	3.80	0.992	2.27	1.000	8.27	1.000

^aUsing linearly fitted atomic energy offsets; ^bAt constant volume (*i.e.* C_v); ^cDifference between the Gibbs and potential energies.

Python libraries

We used Python version 3.9.6. The following list provides the versions of the relevant libraries used for the numerical experiments:

- **numpy** 1.20.3¹¹
- **cuda toolkit** 10.2.89¹²
- **pytorch** 1.9.0=py3.9_cuda10.2_cudnn7.6.5_0¹³
- **pyg** 2.0.1=py39_torch_1.9.0_cu102¹⁴

References

- (1) Buitinck, L.; Louppe, G.; Blondel, M.; Pedregosa, F.; Mueller, A.; Grisel, O.; Niculae, V.; Prettenhofer, P.; Gramfort, A.; Grobler, J.; Layton, R.; VanderPlas, J.; Joly, A.; Holt, B.; Varoquaux, G. API design for machine learning software: experiences from the scikit-learn project. *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*. 2013; pp 108–122.
- (2) Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; Dahl, G. E. Neural Message Passing for Quantum Chemistry. *Proceedings of Machine Learning Research* **2017**, *70*, 1263–1272.
- (3) Vinyals, O.; Bengio, S.; Kudlur, M. Order matters: Sequence to sequence for sets. *arXiv:1511.06391* **2015**, *preprint*.
- (4) Kingma, D. P.; Ba, J. Adam: A method for stochastic optimization. *arXiv:1412.6980* **2014**, *preprint*.
- (5) Zhang, S.; Liu, Y.; Xie, L. Molecular mechanics-driven graph neural network with multiplex graph for molecular structures. *arXiv:2011.07457* **2020**, *preprint*.
- (6) Schütt, K. T.; Sauceda, H. E.; Kindermans, P.-J.; Tkatchenko, A.; Müller, K.-R. SchNet—a deep learning architecture for molecules and materials. *J. Chem. Phys.* **2018**, *148*, 241722.
- (7) Jørgensen, P. B.; Jacobsen, K. W.; Schmidt, M. N. Neural message passing with edge updates for predicting properties of molecules and materials. *arXiv:1806.03146* **2018**, *preprint*.
- (8) Klicpera, J.; Giri, S.; Margraf, J. T.; Günnemann, S. Fast and uncertainty-aware directional message passing for non-equilibrium molecules. *arXiv:2011.14115* **2020**, *preprint*.

- (9) Choudhary, K.; DeCost, B. Atomistic Line Graph Neural Network for improved materials property predictions. *npj Comput. Mater.* **2021**, *7*, 1–8.
- (10) Cordero, B.; Gómez, V.; Platero-Prats, A. E.; Revés, M.; Echeverría, J.; Cremades, E.; Barragán, F.; Alvarez, S. Covalent radii revisited. *Dalton Trans.* **2008**, 2832–2838.
- (11) Harris, C. R. et al. Array programming with NumPy. *Nature* **2020**, *585*, 357–362.
- (12) NVIDIA,; Vingelmann, P.; Fitzek, F. H. CUDA, release: 10.2.89. 2020; <https://developer.nvidia.com/cuda-toolkit>.
- (13) Paszke, A. et al. *Advances in Neural Information Processing Systems 32*; Curran Associates, Inc., 2019; pp 8024–8035.
- (14) Fey, M.; Lenssen, J. E. Fast Graph Representation Learning with PyTorch Geometric. ICLR Workshop on Representation Learning on Graphs and Manifolds. 2019.