Molecule-Augmented Attention Transformer

Łukasz Maziarka
Ardigen
Jagiellonian University
lukasz.maziarka@ardigen.com

Tomasz Danel
Ardigen
Jagiellonian University

Sławomir Mucha
Jagiellonian University

Krzysztof Rataj
Ardigen

Jacek Tabor
Jagiellonian University

Stanisław Jastrzębski
Molecule.one
New York University
staszek.jastrzebski@gmail.com

Abstract

Properties of a molecule depend on a variety of relationships between its atoms. On a high level, these relationships might include a spatial proximity, an existence of a chemical bond, or simply a co-occurrence of two atoms. However, the commonly used graph-based models use only the chemical bonds to define neighbourhood. Motivated by this we propose Molecule-Augmented Attention Transformer model. Our key innovation is augmenting the attention mechanism in Transformer using the inter-atomic distances, and the molecular graph structure. Experiments on molecular property prediction tasks show effectiveness of the approach.

1 Introduction

Deep learning has become a valuable tool for modeling molecules. During the years, the community has progressed from using handcrafted representations, to representing the molecule as a string of symbols, and finally to the currently popular approaches based on graph convolutional neural networks [Goh et al., 2017, Ching et al., 2018].

Graph convolutional networks gather information in each layer from adjacent nodes in the graph. Using the graph structure acts as a strong prior and improves generalization in a range of molecule modeling tasks [Wu et al., 2018]. Some of the most recent works generalize the gather operation in GCNs: Veličković et al. [2017], Shang et al. [2018] proposes to augment GCNs with an attention mechanism, Li et al. [2018] introduces a model that dynamically learns neighbourhood function in the graph.

In parallel to these advances, using the three-dimensional structure of the molecule is becoming increasingly popular [Schütt et al., 2017, Schütt et al., 2017, Gilmer et al., 2017, Feinberg et al., 2018, Cho and Choi, 2018, Lu et al., 2019]. These approaches factor in the inter-atomic distances, often to augment the graph-based neighbourhood.

Our main contribution is unifying these ideas in a model based on the widely successful Transformer architecture [Vaswani et al., 2017]. We propose the Molecule-Augmented Attention Transformer (MAT) that at each step augments the self-attention layer using the graph and the three-dimensional structure of the molecule. By design, we allow the model to use the molecule structure in a flexible manner adapted to the task. We show that our model achieves strong performance on a wide range of molecule property prediction tasks.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

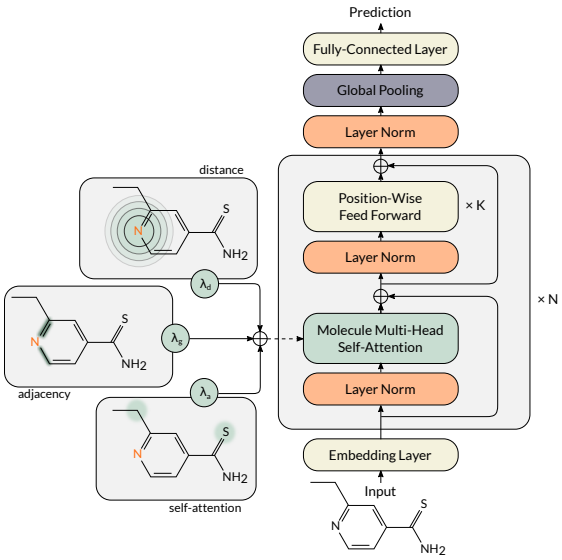


Figure 1: Molecule-Augmented Attention Transformer architecture. In the first layer we embed each atom using one-hot encoding and atomic features. The main innovation is the Molecule Multi-Head Self-Attention block that augments the self-attention module with distance, and graph structure of the molecule.

2 Molecule-Augmented Attention Transformer

In this section, we briefly review Transformer model and introduce our Molecule-Augmented Attention Transformer (MAT).

2.1 Transformer

On a high level, Transformer (for classification) consists of N blocks followed by a pooling and a classification layer. Each block is composed of a multi-head self-attention layer, followed by a feed-forward block that includes a residual connection and layer normalization.

The multi-head self-attention is composed of H heads. Each head takes as input hidden state $\mathbf{H}$ and computes first $\mathbf{Q}_i = \mathbf{H}\mathbf{W}_i^Q$, $\mathbf{K}_i = \mathbf{H}\mathbf{W}_i^H$, and $\mathbf{V}_i = \mathbf{H}\mathbf{W}_i^V$. These are used in the attention operation as follows:

$$\mathcal{A}(\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i) = \text{softmax}\left(\frac{\mathbf{Q}_i \mathbf{K}_i^T}{\sqrt{d_k}}\right)\mathbf{V}_i. \tag{1}$$

When H is larger than 1, the individual outputs are concatenated and projected down first. Finally, the output is processed by a feed-forward network and added residually to the current hidden state.

2.2 Molecule-Augmented Attention Transformer

Our main contribution is Molecule-Augmented Attention Transformer (MAT), a Transformer-based model adapted to processing molecules. The architecture is shown in Figure 1.

We base our model on the widely successful Transformer architecture. In Transformer the self-attention can be interpreted as a soft adjacency matrix between the elements of the input sequence [Battaglia et al., 2018]. As an example, Li et al. [2018] suggests that a flexible definition of the neighbourhood can considerably improve performance. Another benefit of using Transformer is its state-of-the-art performance in a range of applications, not only limited to the language domain [Bello et al., 2019].

Transformer architecture has been applied recently to chemistry by representing the molecule using SMILE(s) notation [Schwaller et al., 2019]. However, using a molecule structure improves performance across many tasks [Duvenaud et al., 2015, Schütt et al., 2017]. Motivated by this we propose Molecule-Augmented Attention Transformer guided by two design principles. First, we use the structure of the molecule as a prior for the attention strength. Second, to ensure flexibility, in the experiments we tune the strength of the prior individually for each dataset by treating it as a hyperparameter.

Molecule Multi-Head Self-Attention. To practically realise these guidelines, we augment the attention in Equation 1 using the molecule structure represented by the graph adjacency matrix $\mathbf{E} \in \{0, 1\}^{N \times N}$, and the inter-atomic distance matrix $\mathbf{D} \in \mathbb{R}^{N \times N}$. To allow for flexibility, we tune hyperparameters that influence the prior importance of these.

More concretely, we augment the self-attention matrix as follows. Let λ_a, λ_d, and λ_g denote scalars weighting the self-attention, distance, and adjacency matrices. Then we modify Equation 1 as follows:

$$\mathcal{A}\left(\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i\right) = \left(\lambda_a \text{softmax}\left(\frac{\mathbf{Q}_i \mathbf{K}_i^T}{\sqrt{d_k}}\right) + \lambda_d g(-\mathbf{D}) + \lambda_g \mathbf{E}\right) \mathbf{V}_i, \tag{2}$$

see also Figure 1. We will denote λ_a, λ_d, and λ_g jointly as λ. In the experiments, we treat λ_a, λ_d, and λ_g as hyperparameters and keep them frozen during training. We use as g either softmax (normalized over the rows), or an element-wise $g(d) = \exp(d)$. We use standard atom features to embed the molecule. Finally, the distance matrix $\mathbf{D}$ is computed using rdkit. Please refer to the Supplement on more details on the input representation.

3 Experiments

In this section, we demonstrate a strong performance of MAT across a range of molecule prediction tasks. We also include experiments on a simple toy task in the Supplement.

3.1 Predicting molecule properties

We first run experiments on a wide range of datasets that represent typical tasks encountered in molecule modeling: FreeSolv, ESOL, Blood-brain barrier permeability (BBBP), $\text{MetStab}_{\text{high}}$, $\text{MetStab}_{\text{low}}$. We expect that performance on these tasks to depend to a various degree on the geometry of the molecule. Please refer to the Supplement for more details on the datasets used.

Experimental setting. We compare Molecule-Augmented Attention Transformer to the following baselines: GCN, Random Forest (RF) and Support Vector Machine with RBF kernel (SVM); and to the following recently proposed models: Edge attention-based multi-relational graph convolutional networks (EAGCN) [Shang et al., 2018], Message Passing Neural Networks (MPNN) [Gilmer et al., 2017] and Weave [Kearnes et al., 2016]. For comparison we also include $\text{MAT}_{\text{graph}}$ model with fixed λ_d and λ_a to 0. We do not compare to other results in the literature on these datasets, because the vast majority of the prior work use custom splits for evaluation.

For all the models we tune the hyperparameters by a random search with a fixed budget of 100 trials. In the early experiments, we found that our model inherits Transformer's training instability [Schwaller et al., 2019]. To counteract this we tune many hyperparameters of MAT, which allows us to find a stable configuration. More precisely, we tune 16 different hyperparameters, including batch size, learning rate, number of epochs, and weight decay. Please refer to the Supplement for full details. To investigate the importance of the search budget we additionally include MAT_{500} where we increased the budget to 500 trials, and MAT_{566} where we additionally tuned λ_d and λ_g by testing all λ_d, λ_g values (66) from the grid, keeping fixed all the other hyperparameters to the optimal values before.

We use random split for FreeSolv and ESOL (consistent with Wu et al. [2018]), and for the MetStab datasets. For all the other datasets we use scaffold split. Test performance is based on the best validation epoch. Each training was repeated three times. All the other experimental details are reported in the Supplement. The code is available at `https://github.com/gmum/MAT`.

Results on the benchmark datasets. Table 1 summarizes the test set performance of the models on the three tasks from the MoleculeNet benchmark. We can observe MAT achieves the strongest result on BBBP and FreeSolv datasets, and loses to MPNN on ESOL. Interestingly, MAT_{566} matches the performance of MPNN on ESOL, showing that we undertuned some of the hyperparameters.

We can also observe that MAT achieves better performance than $\text{MAT}_{\text{graph}}$ across all tasks. This is an important sanity check showing that Euclidean distances, as well as the added flexibility, are both important factors in the overall strong performance of MAT.

Table 1: Test set performance on the benchmark datasets. Molecule-Augmented Attention Transformer achieves strongest performance on BBBP and FreeSolv. Additionally, increasing the hyperparameter search budget in MAT_{500} and MAT_{566} further improves performance, matching MPNN on ESOL.

	BBBP (AUC)	ESOL (RMSE)	FreeSolv (RMSE)
SVM	0.603 ± 0.000	0.493 ± 0.000	0.391 ± 0.000
RF	0.551 ± 0.005	0.533 ± 0.003	0.550 ± 0.004
GC	0.690 ± 0.015	0.334 ± 0.017	0.336 ± 0.043
Weave	0.703 ± 0.012	0.389 ± 0.045	0.403 ± 0.035
MPNN	0.700 ± 0.019	$\mathbf{0.303 \pm 0.012}$	0.299 ± 0.038
EAGCN	0.664 ± 0.007	0.459 ± 0.019	0.410 ± 0.014
$\text{MAT}_{\text{graph}}$	0.655 ± 0.011	0.333 ± 0.004	0.353 ± 0.032
MAT	$\mathbf{0.711 \pm 0.007}$	0.330 ± 0.002	$\mathbf{0.269 \pm 0.007}$
MAT_{500}	0.737 ± 0.006	0.316 ± 0.005	0.269 ± 0.007
MAT_{566}	0.736 ± 0.009	0.298 ± 0.005	0.259 ± 0.014

Results on the biophysical datasets. We run similar experiments on the biophysics datasets. Table 2 reports the results. Similarly to [Wu et al., 2018] we excluded MPNN from the comparison; in our case, the runs took over an order of magnitude longer than all the other models.

The key result is that GCN and MAT perform quite similarly; GCN achieves even better mean test performance on hERG and $\text{MetStab}_{\text{low}}$ datasets. This is in line with [Wu et al., 2018] that finds that simple models are currently strong baselines on biophysics datasets, and that simple models using distances might perform worse on biophysical tasks. Comparing this to the overall consistent performance of MAT shows that the flexibility in adapting λ is an important feature of our model.

Table 2: Test set AUC on the biophysical datasets. On the whole, in this setting GCN and MAT perform similarly.

	Estrogen Alpha	Estrogen Beta	hERG	$\text{MetStab}_{\text{low}}$	$\text{MetStab}_{\text{high}}$
SVM	0.933 ± 0.000	0.765 ± 0.000	0.810 ± 0.000	0.828 ± 0.000	0.822 ± 0.0
RF	0.928 ± 0.003	0.770 ± 0.004	0.769 ± 0.003	0.796 ± 0.004	0.706 ± 0.008
GCN	0.974 ± 0.005	0.726 ± 0.011	$\mathbf{0.917 \pm 0.015}$	$\mathbf{0.856 \pm 0.013}$	0.874 ± 0.014
Weave	0.961 ± 0.005	0.766 ± 0.018	0.765 ± 0.034	0.612 ± 0.009	0.778 ± 0.039
EAGCN	0.937 ± 0.031	0.724 ± 0.025	0.826 ± 0.011	0.779 ± 0.034	0.697 ± 0.019
$\text{MAT}_{\text{graph}}$	0.961 ± 0.002	0.754 ± 0.005	0.881 ± 0.000	0.826 ± 0.004	0.876 ± 0.01
MAT	$\mathbf{0.977 \pm 0.003}$	$\mathbf{0.790 \pm 0.003}$	0.906 ± 0.007	0.839 ± 0.009	$\mathbf{0.892 \pm 0.005}$
MAT_{500}	0.977 ± 0.003	0.768 ± 0.007	0.906 ± 0.007	0.859 ± 0.004	0.892 ± 0.005
MAT_{566}	0.981 ± 0.002	0.778 ± 0.006	0.92 ± 0.002	0.877 ± 0.013	0.894 ± 0.008

4 Conclusions

In this work, we have proposed Molecule-Augmented Attention Transformer model. The key innovation is the Molecule Multi-Head Self-Attention layer that augments the self-attention mechanism with the three-dimensional and graph structure of the molecule.

The flexibility in the model design allows for its easy extension. One natural direction for future work is integrating even more prior information, e.g. the forces between the atoms, or types of chemical bonds.

References

Garrett B. Goh, Nathan O. Hodas, and Abhinav Vishnu. Deep learning for computational chemistry. *Journal of Computational Chemistry*, 38(16):1291–1307, 2017.

Travers Ching, Daniel S Himmelstein, Brett K Beaulieu-Jones, Alexandr A Kalinin, Brian T Do, Gregory P Way, Enrico Ferrero, Paul-Michael Agapow, Michael Zietz, Michael M Hoffman, et al. Opportunities and obstacles for deep learning in biology and medicine. *Journal of The Royal Society Interface*, 15(141):20170387, 2018.

Zhenqin Wu, Bharath Ramsundar, Evan N. Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S. Pappu, Karl Leswing, and Vijay S. Pande. Moleculenet: a benchmark for molecular machine learning† †electronic supplementary information (esi) available. see doi: 10.1039/c7sc02664a. In *Chemical science*, 2018.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph Attention Networks. *arXiv e-prints*, art. arXiv:1710.10903, Oct 2017.

Chao Shang, Qinqing Liu, Ko-Shin Chen, Jiangwen Sun, Jin Lu, Jinfeng Yi, and Jinbo Bi. Edge Attention-based Multi-Relational Graph Convolutional Networks. *arXiv e-prints*, art. arXiv:1802.04944, Feb 2018.

Ruoyu Li, Sheng Wang, Feiyun Zhu, and Junzhou Huang. Adaptive graph convolutional neural networks. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Kristof T. Schütt, Farhad Arbabzadah, Stefan Chmiela, Klaus R. Müller, and Alexandre Tkatchenko. Quantum-chemical insights from deep tensor neural networks. *Nature Communications*, 8:13890, Jan 2017. doi: 10.1038/ncomms13890.

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 I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 991–1001. Curran Associates, Inc., 2017.

Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1263–1272, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL `http://proceedings.mlr.press/v70/gilmer17a.html`.

Evan N. Feinberg, Debnil Sur, Zhenqin Wu, Brooke E. Husic, Huanghao Mai, Yang Li, Saisai Sun, Jianyi Yang, Bharath Ramsundar, and Vijay S. Pande. Potentialnet for molecular property prediction. *ACS Central Science*, 4(11):1520–1530, 2018. doi: 10.1021/acscentsci.8b00507. URL `https://doi.org/10.1021/acscentsci.8b00507`.

Hyeoncheol Cho and Insung S. Choi. Three-dimensionally embedded graph convolutional network (3DGCN) for molecule interpretation. *CoRR*, abs/1811.09794, 2018.

Chengqiang Lu, Qi Liu, Chao Wang, Zhenya Huang, Pan. Lin, and Lixin He. Molecular property prediction: A multilevel quantum interactions modeling perspective. In *AAAI 2019*, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

Peter W. Battaglia, Jessica B. Hamrick, Victor Bapst, Alvaro Sanchez-Gonzalez, Vinícius Flores Zambaldi, Mateusz Malinowski, Andrea Tacchetti, David Raposo, Adam Santoro, Ryan Faulkner, Çaglar Gülçehre, Francis Song, Andrew J. Ballard, Justin Gilmer, George E. Dahl, Ashish Vaswani,

Kelsey Allen, Charles Nash, Victoria Langston, Chris Dyer, Nicolas Heess, Daan Wierstra, Pushmeet Kohli, Matthew Botvinick, Oriol Vinyals, Yujia Li, and Razvan Pascanu. Relational inductive biases, deep learning, and graph networks. *CoRR*, abs/1806.01261, 2018.

Irwan Bello, Barret Zoph, Ashish Vaswani, Jonathon Shlens, and Quoc V. Le. Attention augmented convolutional networks. *CoRR*, abs/1904.09925, 2019.

Philippe Schwaller, Teodoro Laino, Theophile Gaudin, Peter Bolgar, Christopher A Hunter, Costas Bekas, and Alpha A Lee. Molecular transformer: A model for uncertainty-calibrated chemical reaction prediction. *ACS central science*, 2019.

David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alan Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2224–2232. Curran Associates, Inc., 2015.

Steven Kearnes, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley. Molecular graph convolutions: Moving beyond fingerprints. *Journal of Computer-Aided Molecular Design*, 30, 03 2016. doi: 10.1007/s10822-016-9938-8.

Connor W. Coley, Regina Barzilay, William H. Green, Tommi S. Jaakkola, and Klavs F. Jensen. Convolutional embedding of attributed molecular graphs for physical property prediction. *Journal of Chemical Information and Modeling*, 57(8):1757–1772, 2017. doi: 10.1021/acs.jcim.6b00601. URL `https://doi.org/10.1021/acs.jcim.6b00601`. PMID: 28696688.

Greg Landrum. Rdkit: Open-source cheminformatics software. 2016. URL `https://github.com/rdkit/rdkit/releases/tag/Release_2016_09_4`.

Sabina Podlewska and Rafał Kafel. Metstabon—online platform for metabolic stability predictions. *International journal of molecular sciences*, 19(4):1040, 2018.

Anna Gaulton, Louisa J. Bellis, A. Patricia Bento, Jon Chambers, Mark Davies, Anne Hersey, Yvonne Light, Shaun McGlinchey, David Michalovich, Bissan Al-Lazikani, and John P. Overington. ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic Acids Research*, 40(D1): D1100–D1107, 09 2011. ISSN 0305-1048. doi: 10.1093/nar/gkr777. URL `https://doi.org/10.1093/nar/gkr777`.

Bharath Ramsundar, Peter Eastman, Patrick Walters, Vijay Pande, Karl Leswing, and Zhenqin Wu. *Deep Learning for the Life Sciences*. O'Reilly Media, 2019. `https://www.amazon.com/Deep-Learning-Life-Sciences-Microscopy/dp/1492039837`.

Sunghwan Kim, Jie Chen, Tiejun Cheng, Asta Gindulyte, Jia He, Siqian He, Qingliang Li, Benjamin A Shoemaker, Paul A Thiessen, Bo Yu, et al. Pubchem 2019 update: improved access to chemical data. *Nucleic acids research*, 47(D1):D1102–D1109, 2018.

A Acknowledgements

This work was partially supported by the National Science Centre (Poland), grant no: 2018/31/B/ST6/00993.

B Molecule-Augmented Attention Transformer: input representation

We describe here how we represent the input in Molecule-Augmented Attention Transformer.

Input features. We embed each atom as a 25 dimensional vector following [Coley et al., 2017], see Table 3. We leave for future work including features of chemical bonds.

Table 3: Featurization used to embed atoms in Molecule-Augmented Attention Transformer.

Indices	Description
$0-10$	Atomic identity as a one-hot vector of B, N, C, O, F, P, S, Cl, Br, I, other
$11-16$	Number of heavy neighbors as one-hot vector of 0, 1, 2, 3, 4, 5
$17-21$	Number of hydrogen atoms as one-hot vector of 0, 1, 2, 3, 4
22	Formal charge
23	Is in a ring
24	Is aromatic

Distance calculation. The distance matrices are calculated from 3d conformers calculated using UFFOPTIMIZEMOLECULE function from the RDKit package [Landrum, 2016], and the default parameters (MAXITERS=200, VDWTHRESH=10.0, CONFID=-1, IGNOREINTERFRAGINTERACTIONS=True).

C Experimental details

In this section we include details for experiments in Section 3.1.

- **FreeSolv, ESOL.** Regression tasks. Popular tasks for predicting water solubility in terms of the hydration free energy (FreeSolv) and logS (ESOL).
- **Blood-brain barrier permeability (BBBP).** Binary classification task. The blood-brain barrier (BBB) separates the central nervous system from the bloodstream.
- **MetStab$_{\text{high}}$, MetStab$_{\text{low}}$.** Binary classification tasks. The metabolic stability of a compound is a measure of the half-life time of the compound within an organism. The compounds for this task were taken from [Podlewska and Kafel, 2018].
- **hERG, Estrogen Alpha, Estrogen Beta.** Binary classification tasks. Often in drug discovery, it is important that a molecule is not potent towards a given target. hERG is a gene encoding a potassium channel present in heart muscle tissue. For these tasks, the compounds with known activities towards the receptors were extracted from ChEMBL [Gaulton et al., 2011] database and divided into active and inactive sets based on their reported inhibition constant (Ki), being < 100 nM and > 1000 nM, respectively.

Table 4 shows hyperparameter ranges used in experiments for Molecule-Augmented Attention Transformer. A short description of these hyperparameters is listed below:

- **model dim** – size of embedded atom features,
- **model N** – number of encoder module repeats (**N** in Figure 1),
- **model h** – number of molecule self-attention heads,
- **model N dense** – number of dense layers in the position-wise feed forward block (**K** in Figure 1),
- **lambda attention** – self-attention weight λ_a,

Table 4: Molecule-Augmented Attention Transformer hyperparameter ranges

	parameters
batch size	16, 32, 64
learning rate	0.01, 0.005, 0.001, 0.0005, 0.0001
epochs	30, 100
model dim	32, 64, 128, 256, 512, 1024
model N	1, 2, 4, 6, 8
model h	1, 2, 4, 8
model N dense	0, 1, 2
lambda attention	0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1
lambda distance	0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1
model dense output nonlinearity	'tanh', 'relu'
distance matrix kernel	'softmax', 'exp'
model dropout	0.0, 0.1, 0.2
weight decay	0.0, 0.001, 0.01
optimizer	'transformer', 'adam_anneal', 'adam_anneal_v2'
aggregation type	'mean', 'sum'
optimizer factor	1.0, 0.5, 0.1, 0.05, 0.01

- **lambda distance** – distance weight λ_d,
- **model dense output nonlinearity** – nonlinear function applied after all position-wise feed forward layers,
- **distance matrix kernel** – function g used to transform the distance matrix $\mathbf{D}$,
- **model dropout** – dropout applied after the embedding layer, position-wise feed forward layers, and residual layers (before sum operation),
- **weight decay** – optimizer weight decay,
- **optimizer** – optimizer used to train the model,
  - TRANSFORMER – optimizer introduced by Vaswani et al. [2017], we use the 30% of all training steps as the optimizer warmup,
  - ADAM_ANNEAL – Adam optimizer with linear annealing schedule,
  - ADAM_ANNEAL_V2 – Adam optimizer in which the learning rate is multiplied by 0.1 after 70% of all training epochs,
- **aggregation type** – aggregation function used in the global pooling layer,
- **learning rate** – (only for the optimizers different than TRANSFORMER)
- **optimizer factor** – (only for the TRANSFORMER optimizer) the factor by which we multiply the optimizer learning rate.

In our experiments, DeepChem [Ramsundar et al., 2019] implementation of baseline algorithms (SVM, RF, GC, MPNN, Weave) was used. We used the same hyperparameters for tuning as were used in DeepChem, having regard to their proposed default values.

Table 5: SVM hyperparameter ranges

	parameters
C	0.25, 0.4375, 0.625, 0.8125, 1., 1.1875, 1.375, 1.5625, 1.75, 1.9375, 2.125, 2.3125, 2.5, 2.6875, 2.875, 3.0625, 3.25, 3.4375, 3.625, 3.8125, 4.
gamma	0.0125, 0.021875, 0.03125, 0.040625, 0.05, 0.059375, 0.06875, 0.078125, 0.0875, 0.096875, 0.10625, 0.115625, 0.125, 0.134375, 0.14375, 0.153125, 0.1625, 0.171875, 0.18125, 0.190625, 0.2

Table 6: RF hyperparameter ranges

	parameters
n estimators	125, 218, 312, 406, 500, 593, 687, 781, 875, 968, 1062, 1156, 1250, 1343, 1437, 1531, 1625, 1718, 1812, 1906, 2000

Table 7: GC hyperparameter ranges

	parameters
batch size	64, 128, 256
learning rate	0.002, 0.001, 0.0005
n filters	64, 128, 192, 256
n fully connected nodes	128, 256, 512

Table 8: Weave hyperparameter ranges

	parameters
batch size	16, 32, 64, 128
nb epoch	20, 40, 60, 80, 100
learning rate	0.002, 0.001, 0.00075, 0.0005
n graph feat	32, 64, 96, 128, 256
n pair feat	14

Table 9: MPNN hyperparameter ranges

	parameters
batch size	8, 16, 32, 64
nb epoch	25, 50, 75, 100
learning rate	0.002, 0.001, 0.00075, 0.0005
T	1, 2, 3, 4, 5
M	2, 3, 4, 5, 6

Table 10: EAGCN hyperparameter ranges

	parameters
batch size	16, 32, 64, 128, 256, 512
EAGCN structure	'concate', 'weighted'
num epochs	100, 500, 1000
learning rate	0.01, 0.005, 0.001, 0.0005, 0.0001
dropout	0.0, 0.1, 0.3
weight decay	0.0, 0.001, 0.01, 0.0001
n sgc1 1	30, 60
n sgc1 2	5, 10, 15, 20, 30
n sgc1 3	5, 10, 15, 20, 30
n sgc1 4	5, 10, 15, 20, 30
n sgc1 5	5, 10, 15, 20, 30
n sgc2 1	30, 60
n sgc2 2	5, 10, 15, 20, 30
n sgc2 3	5, 10, 15, 20, 30
n sgc2 4	5, 10, 15, 20, 30
n sgc2 5	5, 10, 15, 20, 30
n den1	12, 32, 64
n den2	12, 32, 64

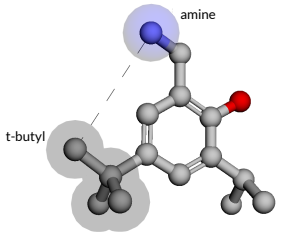
D Toy task

Task description. The essential feature of Molecule-Augmented Attention Transformer is that it augments the self-attention module using molecule structure. Here we investigate MAT on a task heavily reliant on distances between atoms; we are primarily interested in how the performance of MAT depends on λ_a, λ_d, λ_g that are used to weight the adjacency and the distance matrices in Equation 2.

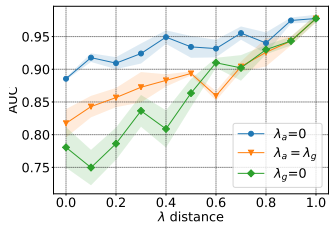
Naturally, many properties of molecules depend on their geometry. For instance, *steric effect* happens when a spatial proximity of a given group, blocks reaction from happening, due to an overlap in electronic groups. However, this type of reasoning can be difficult to learn based only on the graph information, as it does not always reflect the geometry well. Furthermore, focusing on distance information might require selecting low values for either λ_g or λ_a (see Figure 1).

To illustrate this, we designed a toy task to predict whether or not two substructures are closer to each other in space than a predefined threshold; see also Figure 2a. We expect that MAT will work significantly better than a vanilla graph convolutional network if λ_d is tuned well.

Experimental setting. We construct the dataset by sampling 2677 molecules from PubChem [Kim et al., 2018], and use 20 Å threshold between -NH_2 fragment and *tert*-butyl group to determine the binary label. The threshold was selected so that positive and negative examples are well balanced. All the other details are reported in the Supplement.



(a) The toy task is to predict whether two substructures (-NH_2 fragment and *tert*-butyl group) co-occur within given distance.



(b) Molecule-Augmented Attention Transformer performance on the toy task as a function of λ_d, for different settings of λ_g and λ_a.

Figure 2: MAT can efficiently use the inter-atomic distances to solve the toy task (see left). Additionally, the performance is heavily dependent on λ_d, which motivates tuning λ in the main experiments (see right).

Results. First, we plot Molecule-Augmented Attention Transformer performance as a function of λ_d in Figure 2b for three settings of λ: $\lambda_a = 0$ (blue), $\lambda_a = \lambda_g$ (orange), and $\lambda_g = 0$ (green). In all cases we find that using distance information improves the performance significantly. Additionally, we found that GCN achieves 0.93 AUC on this task, compared to 0.98 by MAT with $\lambda_d = 1.0$. These results both motivate tuning λ, and show that MAT can efficiently use distance information if it is important for the task at hand.

Further details. The molecules in the toy task dataset were downloaded from PubChem. The SMARTS query used to find the compounds was `(C([C;H3])([C;H3])([C;H3]).[NX3H2])`. All molecules were then filtered so that only those with exactly one *tert*-butyl group and one -NH_2 fragment were left. For each of them, five conformers were created with RDKit implementation of the Universal Force Field (UFF).

The task is a binary classification of the distance between two molecule fragments. If the distance between -NH_2 fragment and *tert*-butyl group is greater than a given threshold, the label is 1 (0 otherwise). As the distance we mean, maximal Euclidean distance between the closest heavy atoms in these two fragments across calculated conformers. We used 20 Å as the threshold as it leads to a balanced dataset. There are 2677 compounds total from which 1140 are in a positive class. The dataset was randomly splitted into training, validation, and test datasets.

In experiments the hyperparameters that yielded promising results on our datasets were used (listed in Table 11). The values of λ parameters were tuned, and their scores are shown in Figure 2b. All three λ parameters (λ_d, λ_g, λ_a) sum to 1 in all experiments.

To compare our results with a standard graph convolutional neural network, we run a grid search over hyperparameters shown in Table 12. The hyperparameters for which the best validation AUC score was reached are emboldened, and their test AUC score is 0.925 ± 0.006.

Table 11: MAT hyperparameters used.

	parameters
batch size	16
learning rate	0.0005
epochs	100
model dim	64
model N	4
model h	8
model N dense	2
model dense output nonlinearity	'tanh'
distance matrix kernel	'softmax'
model dropout	0.0
weight decay	0.001
optimizer	'adam_anneal'
aggregation type	'mean'

Table 12: Hyperparameters used for tuning GC.

	parameters
batch size	**16**, 32, 64
learning rate	**0.0005**
epochs	20, 40, **60**, 80, 100
n filters	64, **128**
n fully connected nodes	128, **256**

E Attention analysis

As demonstrated in the previous sections, MAT is able to utilize efficiently the additional knowledge stored in the distances. In this section, we run additional analysis.

First, we interpret the chemical function of each self-attention head on a random molecule from the BBBP dataset. Figure 3 visualizes each head in the third layer of MAT. Indeed, the self-attention outputs from each head are noticeably different, and seem to be interpretable. For instance, we see that head 4 puts large weight between the nitrogen atoms in the imidazole ring and the oxygen atoms of the oxo groups.

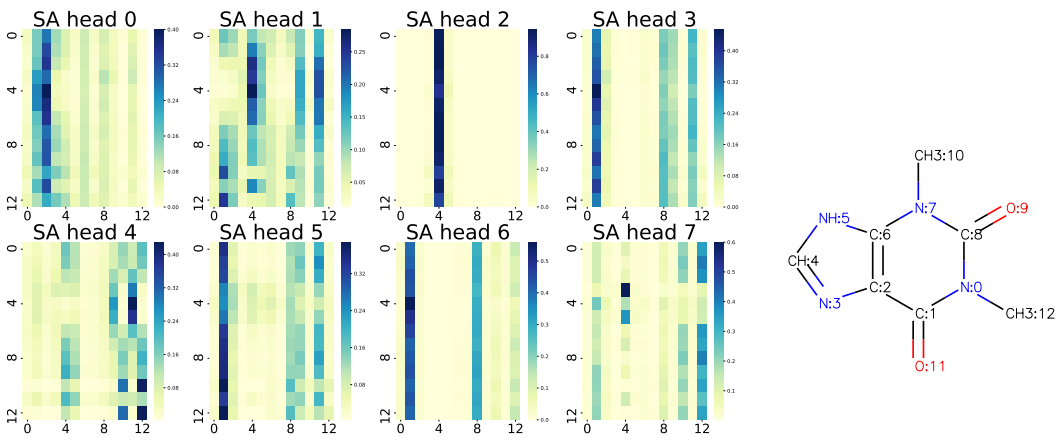


Figure 3: Attention strength in each of the 8 heads in the third layer of MAT (left), for a random molecule from BBBP (right).

To ensure that this is consistent across the dataset we calculate two statistics on the BBBP test set. We manually selected three pairs of heads and the atomic patterns: (i) head 4 from layer 0 that seems to focus on the non-carbon atoms connected by only one bond; (ii) head 5 from layer 3 that seems to

Table 13: Selected self-attention heads in MAT implement chemically interpretable functions. Each column represent one of three selected attention heads, and each row represent statistics for the selected atomic pattern.

	Head_1	Head_2	Head_3
Selected	5.23	1.24	1.27
Baseline	0.41	0.68	0.60

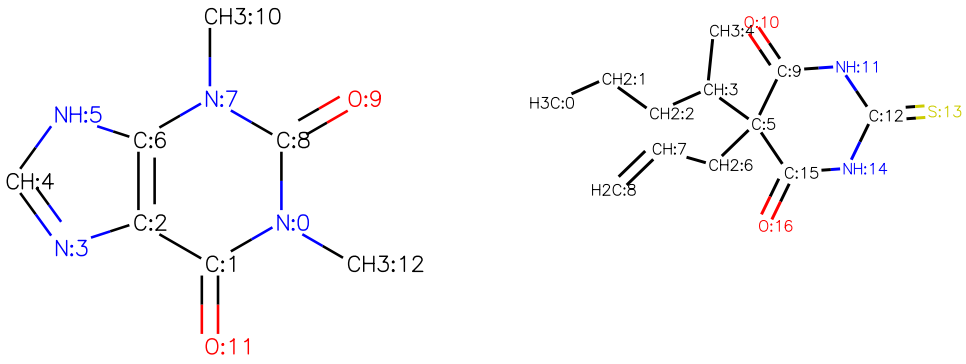
	Head_1	Head_2	Head_3
Percentage	81,3%	72,0%	68,7%

focus on the carbon atoms within ring structures; (iii) head 3 from layer 2 that seems to focus on the positions of electronegative atoms, but not nitrogen or oxygen.

For each head h and its associated atomic pattern a_h (e.g. "atom in a carbon ring") let $h(a_h)$ denote the overall attention strength assigned by the head h to the given atomic pattern a. We report $h(a_h)$ in Table 13 (left) $h(a_h)$ for the three selected heads and atomic patterns pairs, and compare to $h(a')$ for a random atomic pattern a'. In Table 13 (right) we report how often $h(a_h)$ is highest for all atoms a in the molecule. Both statistics corroborate that these attention heads implement a chemically interpretable function.

Below we also include a similar analysis of individual self-attention heads for a random molecule from the ESOL dataset, in Figure 5 to Figure 10. We also include all layers for the random molecule from the BBBP dataset in Figure 11 to Figure 14.

We plot all the self-attention heads for layers 1 to 6 in Figure 5 to Figure 10. The attention analysis shows the weights the neural networks puts on various correlations between atom types and positions. For example, layer 0 of head 0 shows, that the importance is put on the relative positions of the carbon atoms towards the nitrogen atoms in the thiobarbituric group. What is more, within the same layer, we can see that the network notices the relative positions of nitrogen and oxygen atoms, which contribute significantly to the solubility of the molecule. Within other layers, we can also notice that the network takes into consideration the positions of the aliphatic "tails", both between each other and towards the thiobarbituric moiety (Head 0, layer 3). Since the transformer also uses the distance matrix based on 3D coordinates, the acknowledgement of the positions of said aliphatic chains may contribute to the superior performance of the MAT algorithm in this particular case.



(a) BBBP (b) ESOL

Figure 4: The molecules we analyze.

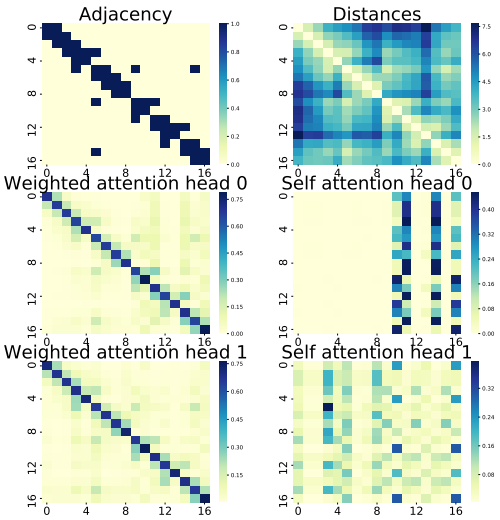


Figure 5: ESOL layer 1

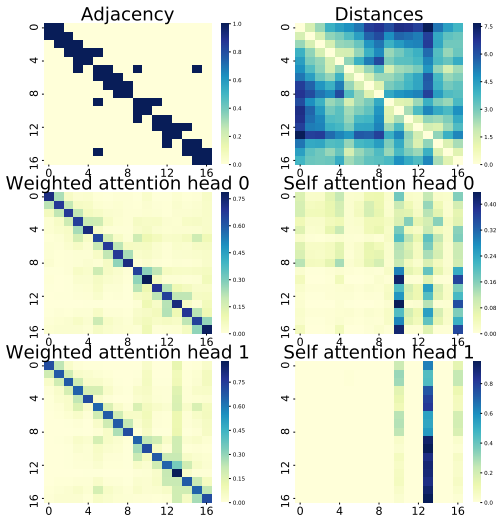


Figure 6: ESOL layer 2

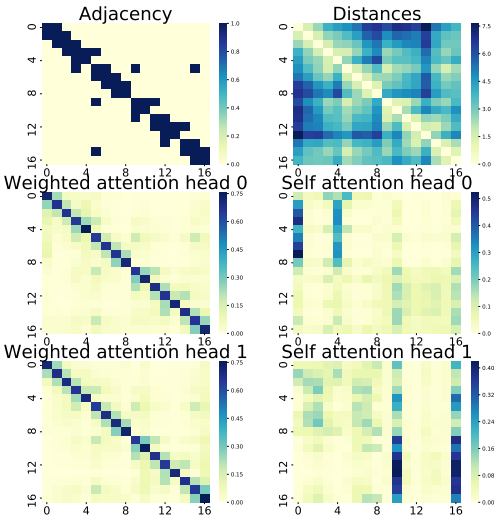


Figure 7: ESOL layer 3

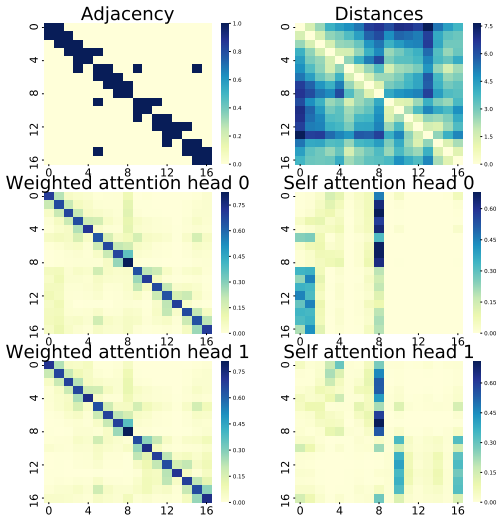


Figure 8: ESOL layer 4

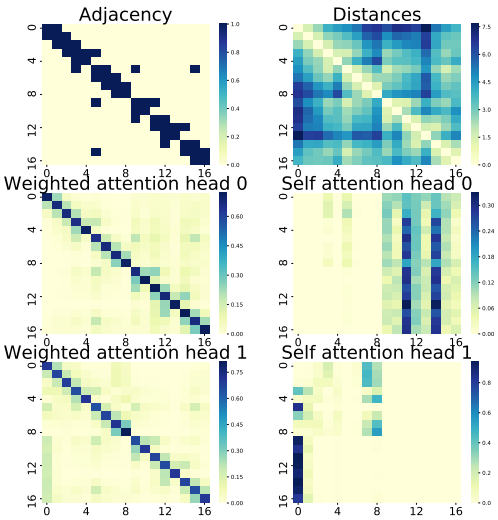


Figure 9: ESOL layer 5

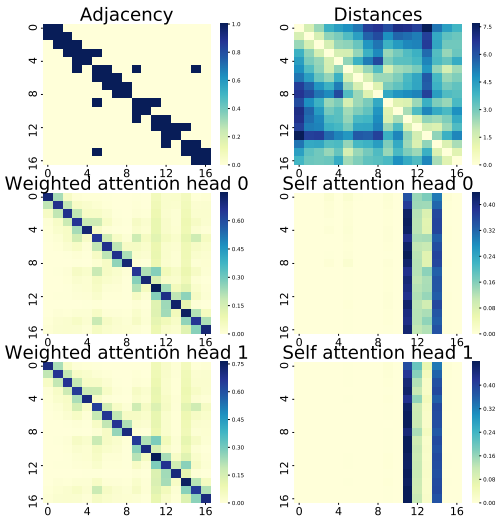


Figure 10: ESOL layer 6

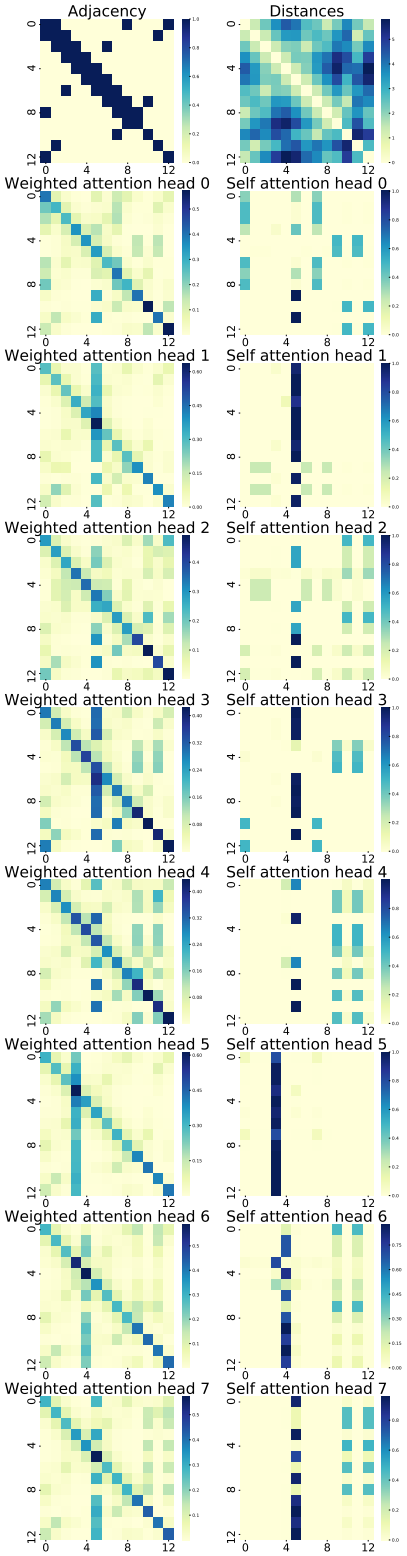


Figure 11: BBBP layer 1

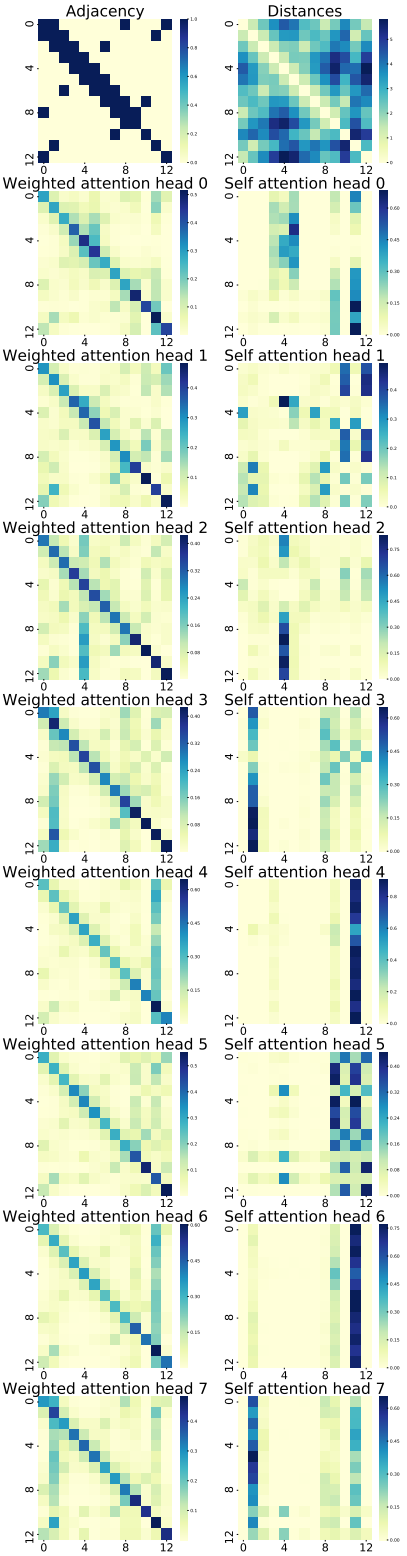


Figure 12: BBBP layer 2

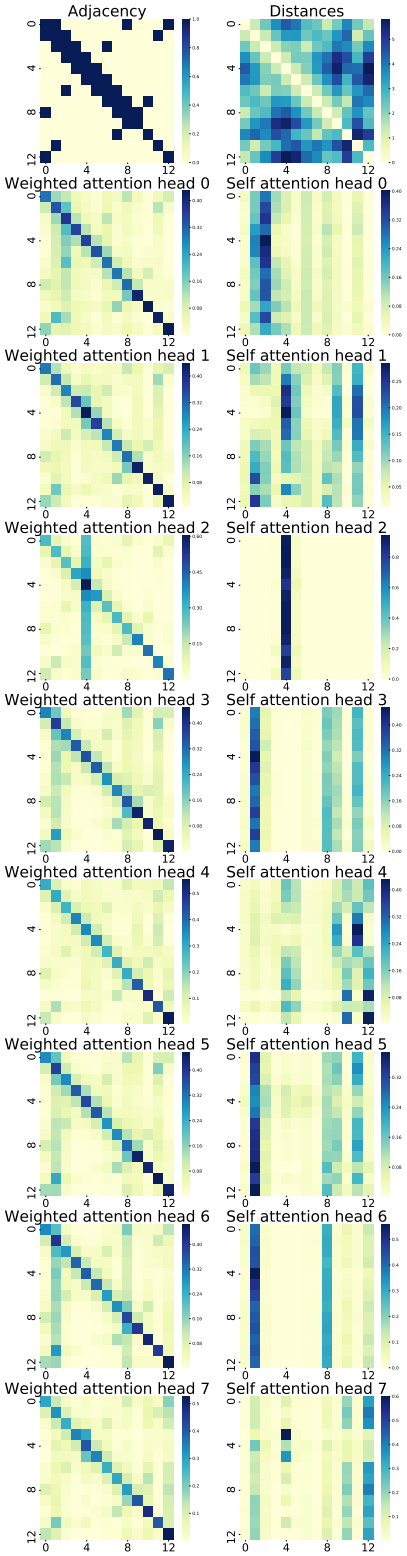


Figure 13: BBBP layer 3

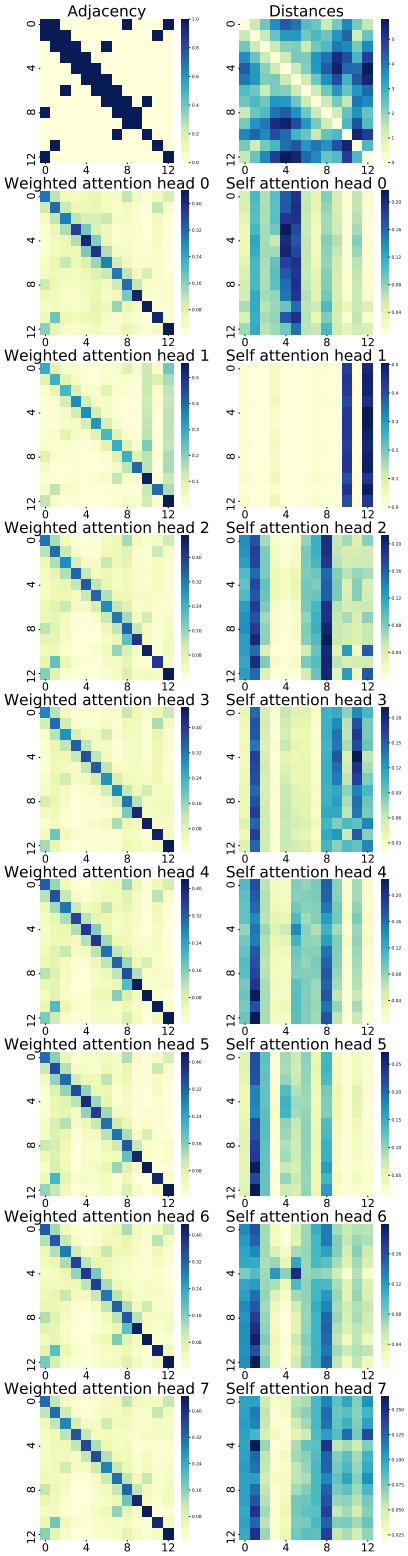


Figure 14: BBBP layer 4

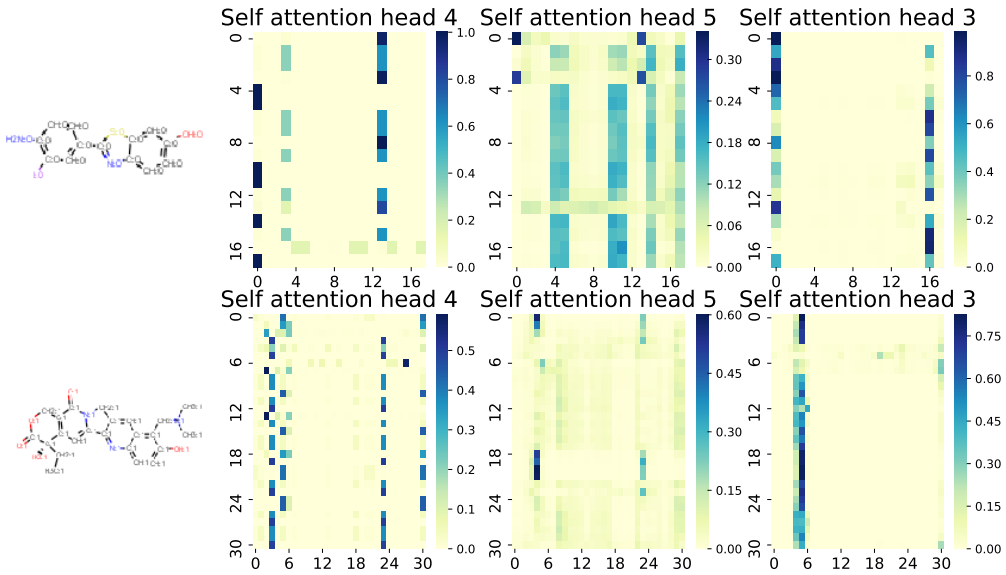


Figure 15: Two sample molecules from the BBBP test dataset, together with self-attention weights of three interesting attention heads (i) head 4 from layer 0; (ii) head 5 from layer 3; (iii) head 3 from layer 2.