Supplementary Information

Alessio Fallani,[†,‡] Ramil Nugmanov,[*,‡] Jose Arjona-Medina,[‡] Jörg Kurt Wegner,[¶] Alexandre Tkatchenko,[†] and Kostiantyn Chernichenko[*,‡]

†*Department of Physics and Materials Science, University of Luxembourg, L-1511 Luxembourg City, Luxembourg*

‡*Drug Discovery Data Sciences, Janssen Pharmaceutica NV, Turnhoutseweg 30, 2340 Beerse, Belgium*

¶*Johnson & Johnson Innovative Medicine, 301 Binney Street, Cambridge, MA 02142, USA*

E-mail: rnugmano@its.jnj.com; kcherni1@its.jnj.com

Motivation for rollout as model proxy

If we want to do a spectral analysis of the considered network (i.e. which combination of input components are extracted), the choice of Attention Rollout is a rather natural one. This matrix, in fact, is a good proxy to understand how the information coming from the input is mixed throughout the network. One can retrieve this intuition by considering the action of a Self Attention Network (SAN) architecture with skip connections at layer l on the input $X_{l-1} \in \mathbb{R}^{n\times d}$ as formalized in.[1] This update can be written as:

$$X_l = X_{l-1} + A_l X_{l-1} H^T, \tag{1}$$

where $A_l \in \mathbb{R}^{n\times n}$ is the input-dependent attention matrix at layer l (hence dependent on X_{l-1}) and $H \in \mathbb{R}^{d\times d}$ is the combination of the value matrix W_V and the projection head W_{proj} (namely $H = W_{proj}^T W_V^T$) and does not depend on the input X_l for any l. If one is only

interested in how the input information gets mixed between tokens, one can notice that the action of H on X_l is a token-wise transformation and study the case for $H = \mathbb{I}$ (notice that for the same reason we did not consider the activation function) to obtain:

$$\begin{aligned} X_L &= (\mathbb{I} + A_l)X_{l-1} \\ &= \left[\prod_{l=L}^{l=1}(\mathbb{I} + A_l)\right] X \\ &= 2^L \tilde{A}_L X \end{aligned} \tag{2}$$

where $\tilde{A}_L = \prod_{l=L}^{l=1}(\mathbb{I} + A_l)$ is the attention rollout matrix as defined in.[2] Notice that this observation is valid only for a single-headed attention case. To handle multi-headed networks here we decide to follow the definition of $\tilde{A}_L$ from the previous section and just consider the corresponding single-headed case for which $A_l \to \langle A_l \rangle_{heads}$. Considering $\tilde{A}$ as a proxy for the model is of course a strong simplification of the network but as we will see it manages to be useful to gain spectral insights.

Spectral properties

Now that we motivated the use of the Attention Rollout matrix as a proxy for the transformer network, we look at its eigenvalues and eigenvectors in order to understand along which directions the input X is decomposed. For simplicity we will make use of the bra-ket notation. Given the Rollout matrix $\tilde{A}$ (we drop the subscript relative to network depth for simplicity of notation) we can write:

$$\tilde{A} = \sum_{i=0}^{N-1} a_i \left|a_i\right\rangle \left\langle a_i\right| \tag{3}$$

where a_i are the eigenvalues of $\tilde{A}$ and where$\left|a_i\right\rangle \left\langle a_i\right|$ are the projectors on the associated eigenvectors $\left|a_i\right\rangle$. It is important to state here that $\tilde{A}$ are in general non-symmetric positive definite matrices, hence their eigenvectors can be non-orthogonal and their eigenvalues are complex numbers with positive real parts. As additional observation, considering in the

definition of $\tilde{A}$ that attention matrices are row-stochastic matrices, we can say that $\tilde{A}$ is also row-stochastic. This fact implies that $\tilde{A}\,|\mathbf{1}\rangle = |\mathbf{1}\rangle$ (with $|\mathbf{1}\rangle$ normalized vector with all equal components) and that $|a_i| \leq 1 \;\forall\; i$, hence if we sort the eigenvalues in descending order of magnitude we can say that $a_0 = 1$ and that $|a_0\rangle = |\mathbf{1}\rangle$. If we consider the spectrum of the molecular graph Laplacian L as:

$$L = \sum_{i=0}^{N-1} l_i \,|l_i\rangle \,\langle l_i| \tag{4}$$

with $l_0 \leq l_1 \leq ... \leq l_{N-1}$, we see that because of basic properties of the graph Laplacian L[3] we have $\langle a_0|l_0\rangle = 1$.

Filtered convolution-like behavior and connection with graph signal oversmoothing

More in detail, one can analyze the matrix of the normalized eigenvectors overlap $C_{ij} = \langle a_i|l_j\rangle$, and consider the spectrum $\{a_i\}$ as the weight associated to this decomposition. If we now denote with $\mathcal{U}$ the set of i for which $\max_j C_{ij} > 0.9$ (threshold for overlap close to 1), then provided $\eta = \frac{\sum_{i\in\mathcal{U}\setminus 0}|a_i|}{\sum_{i=1}^{i=N-1}|a_i|} \sim 1$ we can write the action of the network proxy $\tilde{A}$ on the input $|x\rangle$ as:

$$\tilde{A}\,|x\rangle \sim \sum_{i\in\mathcal{U}} a_i \,|l_i\rangle \,\langle l_i|x\rangle \,, \tag{5}$$

which is interpretable as a filtered graph convolution where the relevant rollout eigenvalues $\{a_i|i \in \mathcal{U}\}$ play the role of filter. Notice that if this approximation is valid ($\eta \sim 1$), we can see the number of Laplacian eigenmodes considered in this sum as the *bandwidth* of a graph convolutional filter. Notice that independently of wether this approximation is valid or not (that is to say if there is some part of the spectrum that is not related to Laplacian eigenmodes) the set of $\{a_i|i \in \mathcal{U}\}$ can be associated to the amount of graph information captured by the model, effectively meaning that ζ can also be seen as measure of oversmoothing[1,4] (or lack thereof) in the sense of graph-spectral information for these

models. In order to study this effect on Graphormer we need some tailoring. In this network, in fact, we have a class token used for molecule level modeling which is not present in the molecular graph. To handle this problems, we consider first the full attention to obtain the rollout matrix $\tilde{A}$ and preserve row-stochasticity properties, but then when considering the eigenvectors $|a_i\rangle$ we slice only the components associated with atoms and then normalize before taking inner products with Laplacian eigenvectors. This can be seen in both models as looking at the token subspace relative to the molecular graph nodes.

Correlation between the representations of pretrained models and pretraining labels

Using the same methodology as applied for information conservation after fine-tuning, the representations of pretrained models were used as inputs for building a L1, L2-regularized linear regression models (ElasticNet) for the respective pretraining labels (Fig. 1).

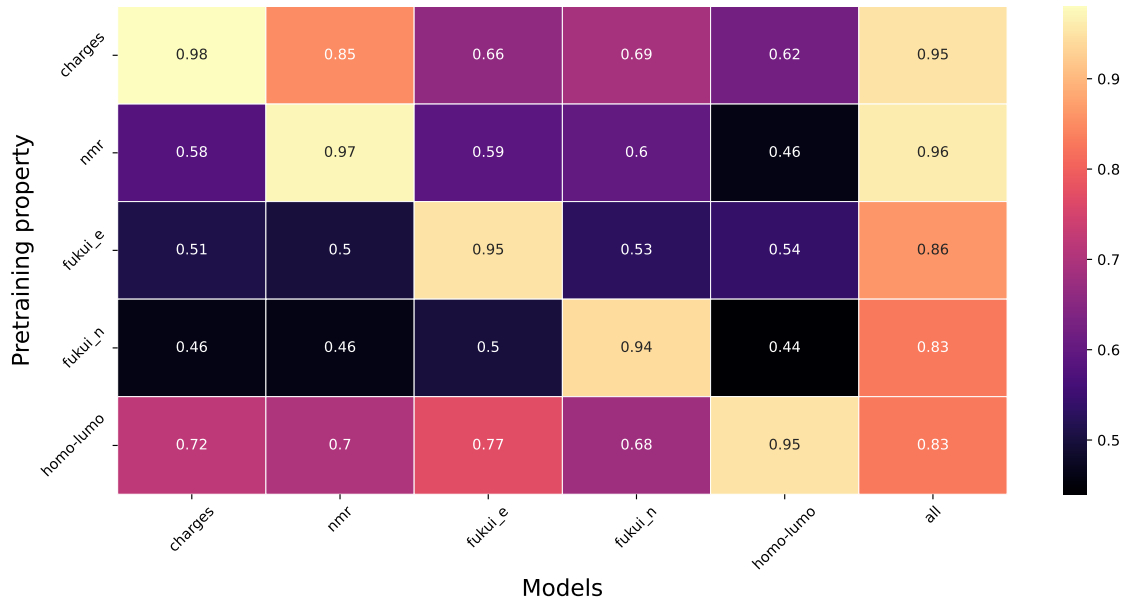


Figure 1: R^2 for the regression tasks using the representations of non-tuned pretrained models as inputs

Latent expressivity for single atomic property pretrainings

In Fig. 2 we we report the analysis of ρ_L as defined in the main text, for all the models pretrained on each single atomic quantum property. We can see how models pretrained on NMR and charges have a similar trend across layers as model pretrained on all atomic properties, while models pretrained on Fukui functions, while providing a similar trend, achieve a lower maximum value in the initial part of the network.

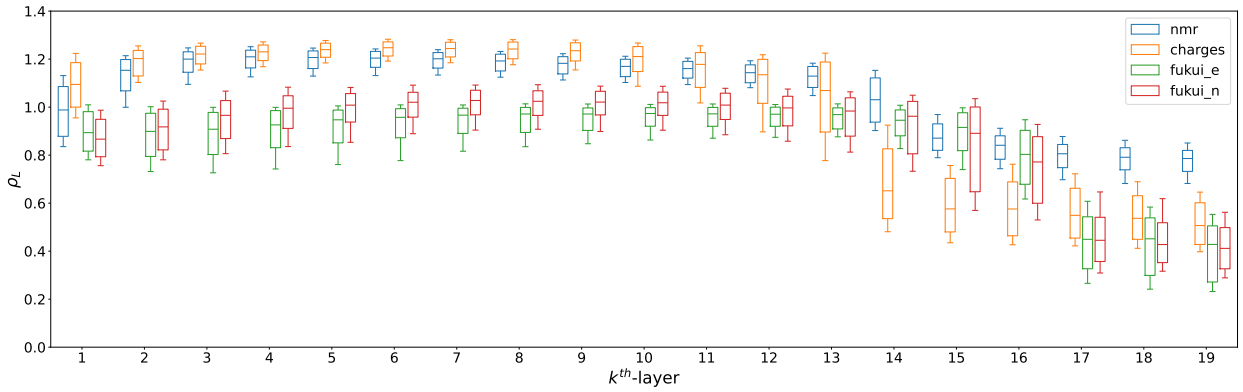


Figure 2: Boxplots of ρ_L computed at each layer except the last one for a sample of 2200 structures extracted uniformly from all the fine-tuning test sets (100 structures for each of the 22 tasks). We report this quantity for the models pretrained on each individual atomic property. The whiskers go from the 15th percentile to the 85th for better visualization of trends and outliers are excluded for the same reason.

Implementation Details

The Graphormer encoder network (see Fig. 3, Fig. 4) employed in this work is composed of 20 self-attention layers, with embedding dimension of 256 and 32 heads and dimension of the feed-forward component in each layer of 512. Transformer pre-norm is employed together with a dropout of 0.1. The optimizer used is AdamW with the default value of weight decay set to 0.01, and we employed mixed-precision training for both pretraining and fine-tuning. For what concerns the pretraining phase no test set was used in favor of a 0.8/0.2

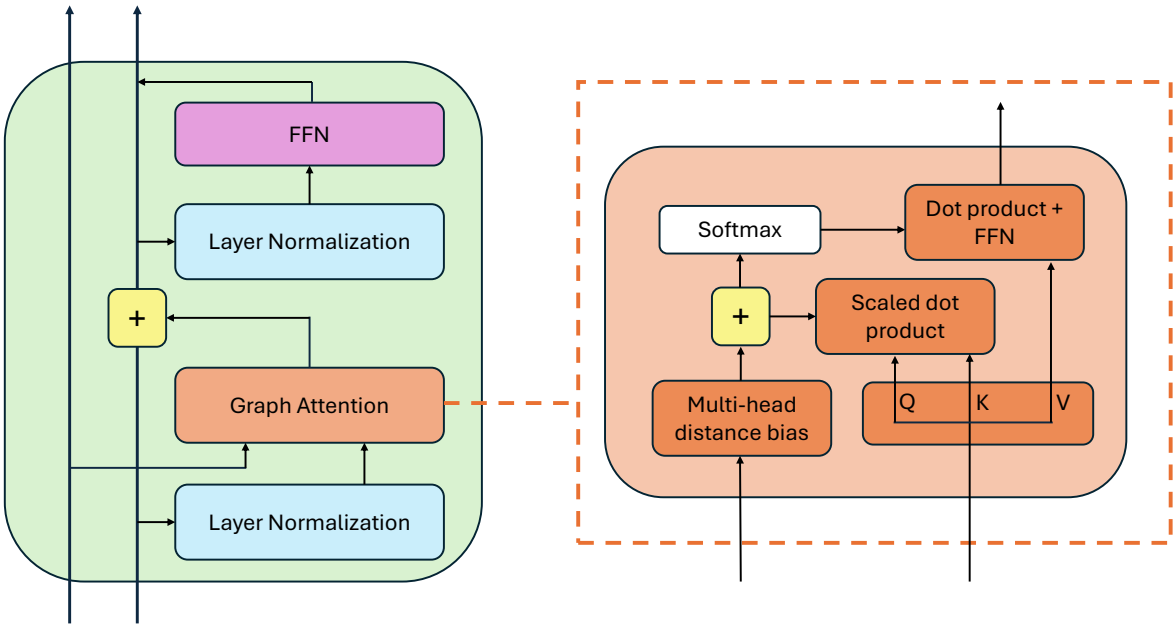
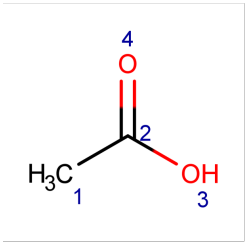


Figure 3: Schematic representation of Graphormer derivative architecture. On the left there is the block of operations in each layer, while on the right there is a zoom on the graph attention operation.

training/validation split. We employed L1 loss and a constant learning rate of 10^{-4} with a patience of 100 epochs for early stopping across all cases. For atomic QM pretraining we employed a batch size of 100 while given the higher number of samples and the duration of training we employed a batch size of 1000 for the pretraining on HOMO-LUMO gap. No scaling of the labels is employed at this stage with the exception of NMR shielding constants where we divided by a constant factor of 100 as otherwise convergence would have been too slow. For the fine-tuning cases the hyperparameters used in each downstream task are the same: the batch size used is 32, while for what concerns the learning rate a triangular cyclic scheduling was employed with a minimum value of 2×10^{-5} and a maximum value of 2×10^{-4}. The training is stopped with an early stopping criterion with patience of 200. The loss used for regression tasks is L1 loss, while for classification tasks a censored regression approach is taken using again L1 loss with right censor set at 0 for negative examples and left censor set at 1 for positive examples. For what concerns regression labels, given the diversity of



	CLS	C1	C2	O3	O4
Atom Type	X	6	6	8	8
Valence	X	4	3	2	1

	CLS	C1	C2	O3	O4
CLS	X	X	X	X	X
C1	X	0	1	2	2
C2	X	1	0	1	1
O3	X	2	1	0	2
O4	X	2	1	2	0

Figure 4: Example of the molecule representation in Graphormer. Atoms are encoded as pairs of atom types and valence. Bond connectivity is encoded as a topological distance matrix. A special CLS token introduced for aggregation. The distance from and top CLS is encoded with a special non-overlapped reserved value.

the tasks we opted for a standard scaling. We conclude this section providing the dataset dimension for each individual downstream task dataset in table 1, and a reference table for model tags in 2.

Regression modelling of JNJ data for human liver microsome intrinsic clearance

The dataset contains 138439 unique compounds. Measurements are reported for two assays: for 54611 compounds only for assay 1, for 66631 only for assay 2, and for 23358 compounds for both assays. For the multitask regression we used L2 loss masking out the missing values for the cases reporting only one clearance value.

UMAP (chemical space) plotting detais

20k data points were randomly sampled from each of the datasets: combined TDC, proprietary HLM intrinsic clearance, HOMO-LUMO gap, and QM136. The samples were then combined and, using the respective chython (v1.75) methods, the molecules were canonicalized and transformed into 1024-bit Morgan fingerprints (max radius 4). UMAP dimension-

Table 1: Dataset sizes for different tasks

Task	Dataset Size
caco2_wang	910
hia_hou	578
pgp_broccatelli	1218
bioavailability_ma	640
lipophilicity_astrazeneca	4200
solubility_aqsoldb	9982
bbb_martins	2030
ppbr_az	2790
vdss_lombardo	1130
cyp2d6_veith	13130
cyp3a4_veith	12328
cyp2c9_veith	12092
cyp2d6_substrate_carbonmangels	667
cyp3a4_substrate_carbonmangels	670
cyp2c9_substrate_carbonmangels	669
half_life_obach	667
clearance_microsome_az	1102
clearance_hepatocyte_az	1213
herg	655
ames	7278
dili	475
ld50_zhu	7385

Table 2: Reference for model tags

model tag	meaning
scratch	models trained without pretraining
all	models pretrained on the multitask modeling of atomic charges, NMR shielding constants, Fukui electrophilic functions and Fukui nucleophilic functions
charges	models pretrained on modeling atomic charges
nmr	models pretrained on modeling NMR shielding constants
fukui_n	models pretrained on modeling Fukui nucleophilic functions
fukui_e	models pretrained on modeling Fukui electrophilic functions
masking	models pretrained with the masking procedure outlined in the main text
homo-lumo	models pretrained on modeling the HOMO-LUMO gap of the molecules

ality reduction of the fingerprints into a 2D space was conducted using umap-learn package (v0.5.5) with the Jaccard distance metric, 0.1 minimum distance, 15 neighbours as main parameters. To improve data visibility on the UMAP plot, out of 80k, only 30k randomly sampled data points were depicted.

References

(1) Dovonon, G. J.-S.; Bronstein, M. M.; Kusner, M. J. Setting the Record Straight on Transformer Oversmoothing. 2024; `https://arxiv.org/abs/2401.04301`.

(2) Abnar, S.; Zuidema, W. H. Quantifying Attention Flow in Transformers. 2020; `https://arxiv.org/abs/2005.00928`.

(3) Fiedler, M. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal* **1973**, *23*, 298–305.

(4) Rusch, T. K.; Bronstein, M. M.; Mishra, S. A Survey on Oversmoothing in Graph Neural Networks. 2023; `https://arxiv.org/abs/2303.10993`.