Seminário de Lógica e IA (SNAIL) - IME-USP

Machine Learning for Graphs and Some Applications to Polymer Science

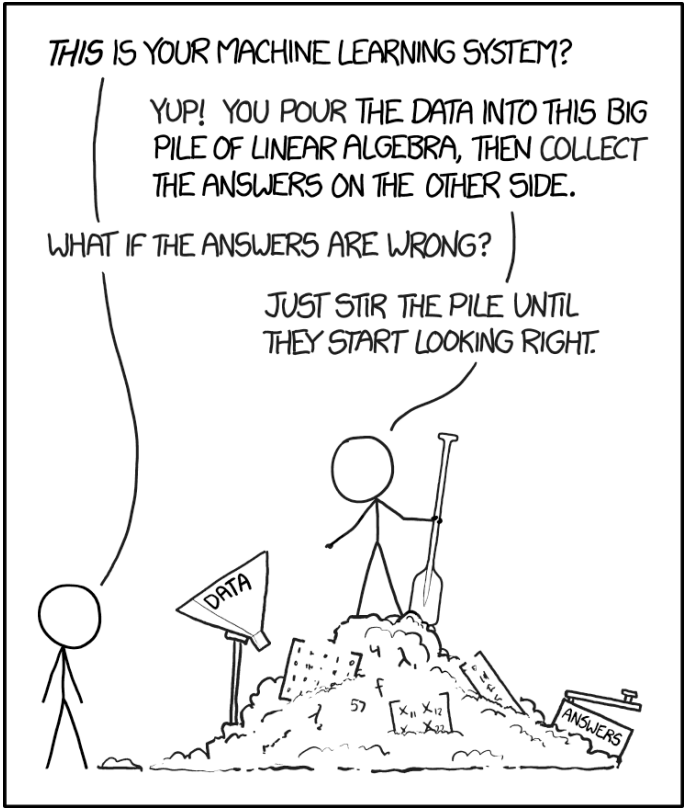
David Kohan Marzagão
Lecturer at King's College London
Visiting Researcher at University of Oxford

david.kohan@kcl.ac.uk

Joint work with Shannon R. Petersen, Georgina L. Gregory, Yichen Huang, David A. Clifton, Charlotte K. Williams, Clive R. Siviour

18 April 2023

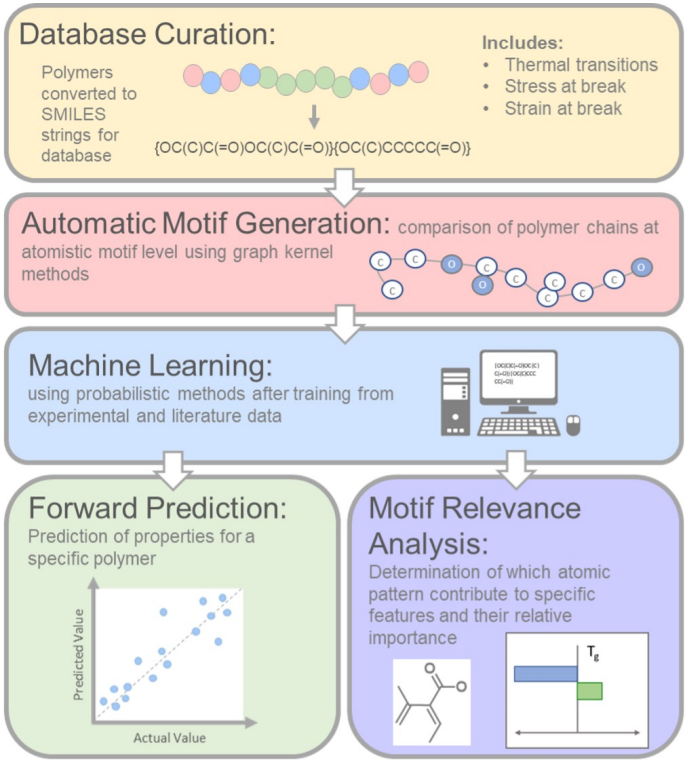
For the Mathematicians out there...



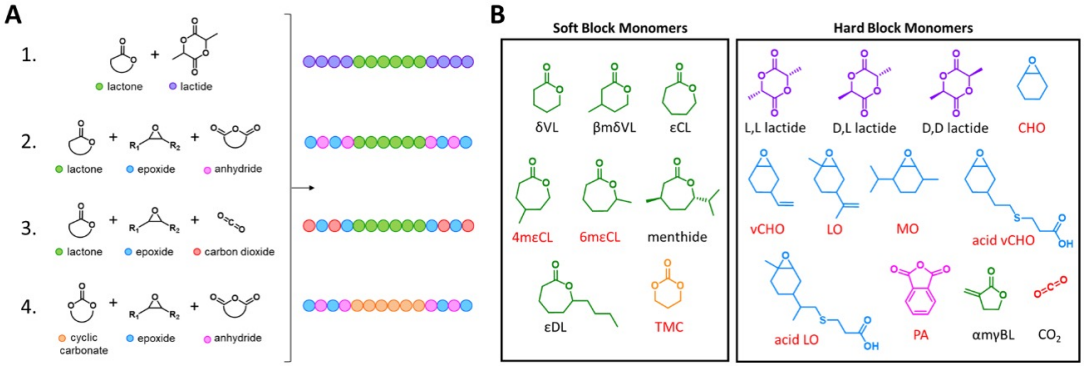
Motivation

- An informal (and maybe not too precise) way to describe this project is:
  - Can we predict properties of polymers...
  - ... without using expert knowledge from polymer science?

Outline


Database Curation:
Polymers converted to SMILES strings for database
{OC(C)C(=O)OC(C)C(=O)}{OC(C)CCCCC(=O)}
Includes:
• Thermal transitions
• Stress at break
• Strain at break
Automatic Motif Generation: comparison of polymer chains at atomistic motif level using graph kernel methods
Machine Learning:
using probabilistic methods after training from experimental and literature data
Forward Prediction:
Prediction of properties for a specific polymer
Predicted Value
Actual Value
Motif Relevance Analysis:
Determination of which atomic pattern contribute to specific features and their relative importance
T_g

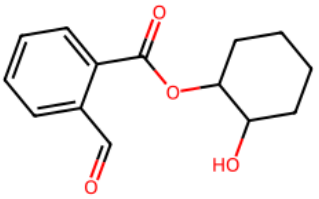
Outline



- Very small dataset. **(about 80 polymers!)**
- Each point is result of extensive lab work.

Outline

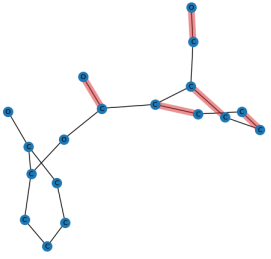
- For example, **phthalic anhydride (CHO PA)**:



- Has SMILES:

OC1C(CCCC1)OC(=O)C1=CC=CC=C1C(=O)

- And becomes a graph like:

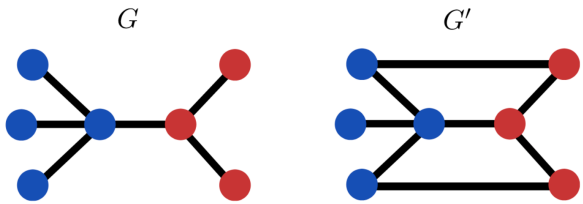


Database Curation

- Things to consider:
  - How do monomers combine when SMILES are concatenated?
  - Some polymer chains are created akin sampling without replacement!
  - Taking the entire chain into account gives us the number of atoms and overall size.
  - These can be reasonably large graphs with thousands of nodes.

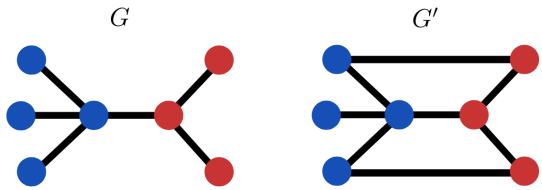
Some Background - Graph Kernels

A Very Very Short Introduction



- Putting simply Graph Kernel methods seek to **compare graphs**.
  - Not a simple task in theory!
  - Consider the analogous problem in the context of vectors, for example.
- Not to be confused with Kernels **on** Graphs. **(they compare nodes on graphs!)**
- Not to be confused with Graph Neural Networks!
  - Weisfeiler-Lehman kernel (coming up) is very much a basis of several GNN architectures.
- (All these beatiful graphs and diagrams were taken from *Borgwardt et al., 2020*)

The Simplest of Examples



$\phi_{\mathrm{N}}(G) = [\,4 \;\; 3\,]$ $\qquad$ $\phi_{\mathrm{N}}(G') = [\,4 \;\; 3\,]$

- The **Node Histogram** graph kernel disregards edges. It only counts the number of each label in each graph and then take the inner product as a measure of similarity.
- In this case,

$$k_N(G, G') = \langle \phi(G), \phi(G') \rangle_{\mathcal{H}} = 25 \tag{1}$$

- Despite its simplicity, the node histogram kernel often performs well in some datasets!

	based on	graph type	node type	edge type	complexity
node histogram kernel	nodes	none	labelled attributed $(1.1,\ 0.7)$	none	$\mathcal{O}(nd_v)$

Introduction

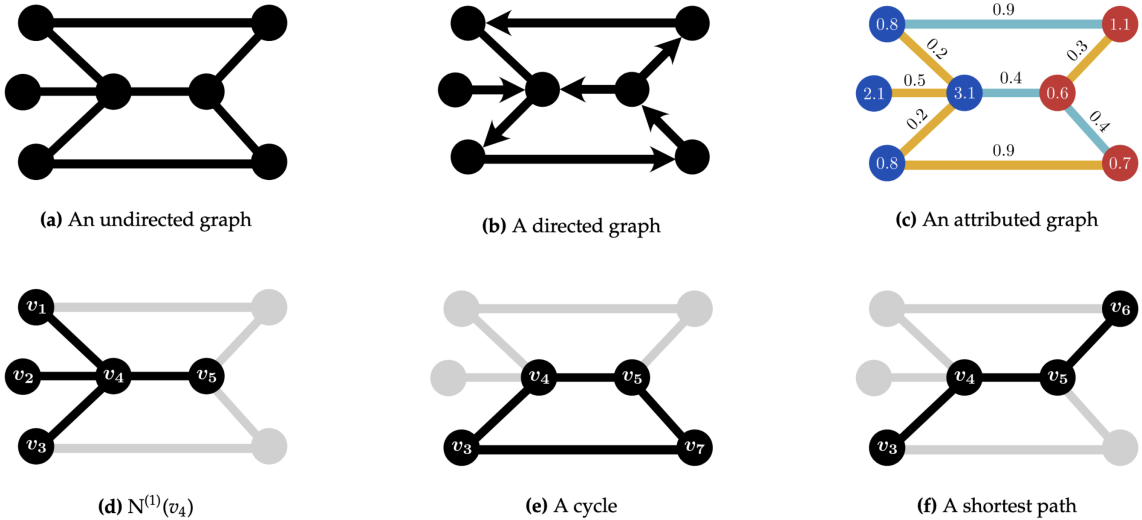
- A **graph kernel**, or kernel for graphs, is a kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, where $\mathcal{X} = \mathcal{G}$ is a family of graphs. We have $k(G, G') = \langle \phi(G), \phi(G') \rangle_{\mathcal{H}}$.
  - The function $\phi : \mathcal{X} \to \mathcal{H}$ is a feature map that represents inputs a of set $\mathcal{X}$ as elements of f a vector space $\mathcal{H}$.
  - There is some magic going on here. The fact that we are dealing with kernels allows us to have implicit functions ϕ. **(unlike what we've seen thus far)**
    - It all boils down to the matrix K (formed by $k(G_i, G_j)$) being positive semi-definite, i.e., all eigenvalues non-negative.
- Things to consider:
  - Node and edge labels. **(colours, classes, ...)**

$$l_V : V \to \Sigma_V \quad \text{and} \quad l_E : E \to \Sigma_E$$

  - Node and edge attributes. **(a value in $\mathbb{R}$, for example...)**

$$\mathcal{A}_V : V \to \mathbb{R}^d \quad \text{and} \quad \mathcal{A}_E : E \to \mathbb{R}^d$$

Labels and Attributes for Nodes and Edges



(a) An undirected graph

(b) A directed graph

(c) An attributed graph

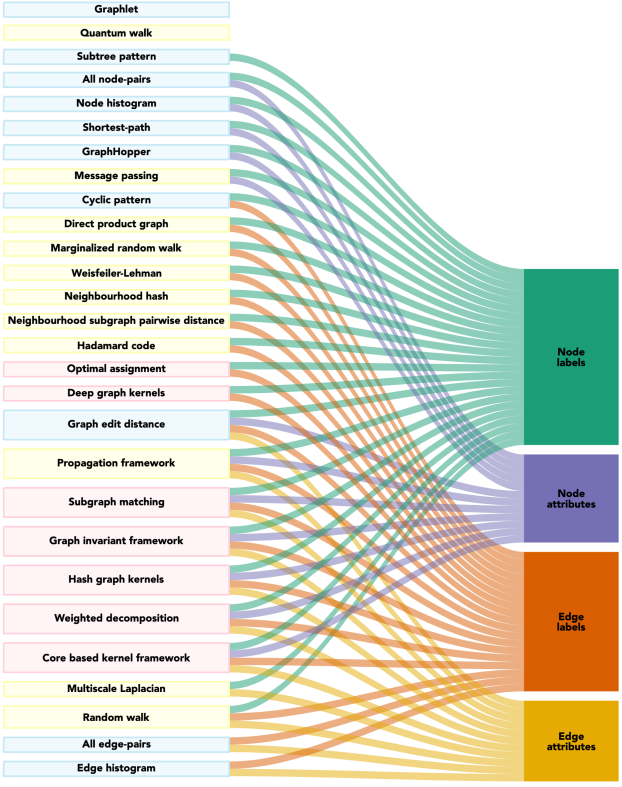
(d) $N^{(1)}(v_4)$

(e) A cycle

(f) A shortest path

- Specific graph kernels often make assumptions of whether a graph is directed or attributed, for example.
  - But it does not in general ask for graph properties, such as 'no cycles', or 'complete', or even 'connected'(!)

Labels and Attributes for Nodes and Edges

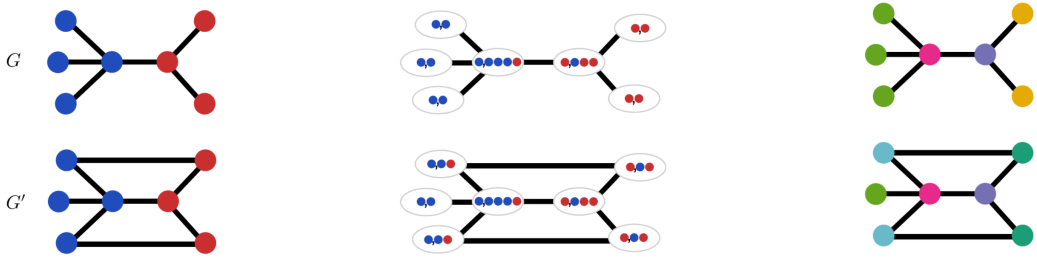


Complete Graph Kernels

Definition (Complete Graph Kernel) A kernel $k(G, G') = \langle \phi(G), \phi(G') \rangle_{\mathcal{H}}$ is called **complete** if ϕ is injective.

- Efficiently (poly-time) computing a complete graph kernel $\Rightarrow$ solving graph isomorphism problem in poly-time. (*Gärtner et al., 2003*, **Proposition 1.**)
  - What if implicit kernel?

Information Propagation Example - Weisfeiler-Lehman Graph Kernels



(a) G and G', *i.e.* $h = 0$.

(b) The sorted multisets.

(c) $h = 1$.

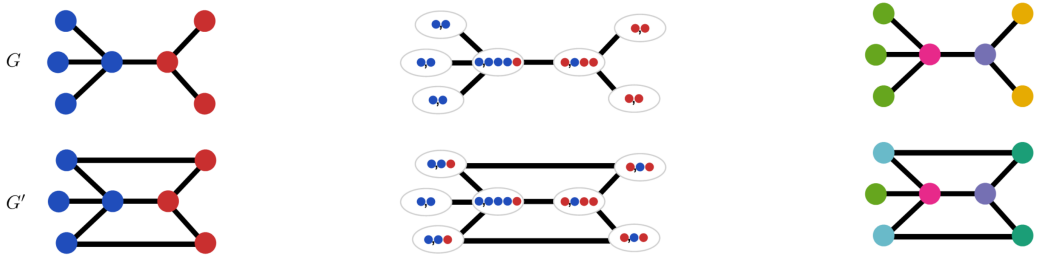
(d) The hash function.

$$\phi(G) = [\,4\ \ 3\ \ 3\ \ 2\ \ 0\ \ 0\ \ 1\ \ 1\,] \qquad \phi(G') = [\,4\ \ 3\ \ 1\ \ 0\ \ 2\ \ 2\ \ 1\ \ 1\,]$$

(e) The feature vector representations of G and G'.

- In this example, we have $k(G, G') = \langle \phi(G), \phi(G') \rangle = 30$.
- The complexity of the relabelling iterations is $O(hm)$, where $m = |E|$ and h is the propagation depth. Graphs can be computed in parallel.

Information Propagation Example - Weisfeiler-Lehman Graph Kernels



(a) G and G', *i.e.* $h = 0$.

(b) The sorted multisets.

(c) $h = 1$.

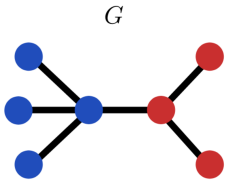
(d) The hash function.

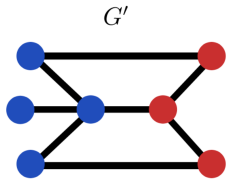
$$\phi(G) = [\,4\ \ 3\ \ 3\ \ 2\ \ 0\ \ 0\ \ 1\ \ 1\,] \qquad \phi(G') = [\,4\ \ 3\ \ 1\ \ 0\ \ 2\ \ 2\ \ 1\ \ 1\,]$$

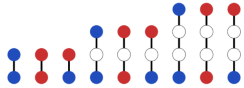
(e) The feature vector representations of G and G'.

	based on	graph type	node type	edge type	complexity
Weisfeiler-Lehman kernel Shervashidze and Borgwardt, 2009	label refinement	undirected directed	labelled	labelled	$\mathcal{O}(hm)$

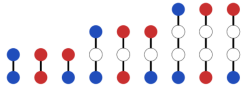
Bag of Structures Example - Shortest Path Kernel







$\phi_{\text{path}}(G) = [\,3\ \ 2\ \ 1\ \ 3\ \ 1\ \ 5\ \ 0\ \ 0\ \ 6\,]$



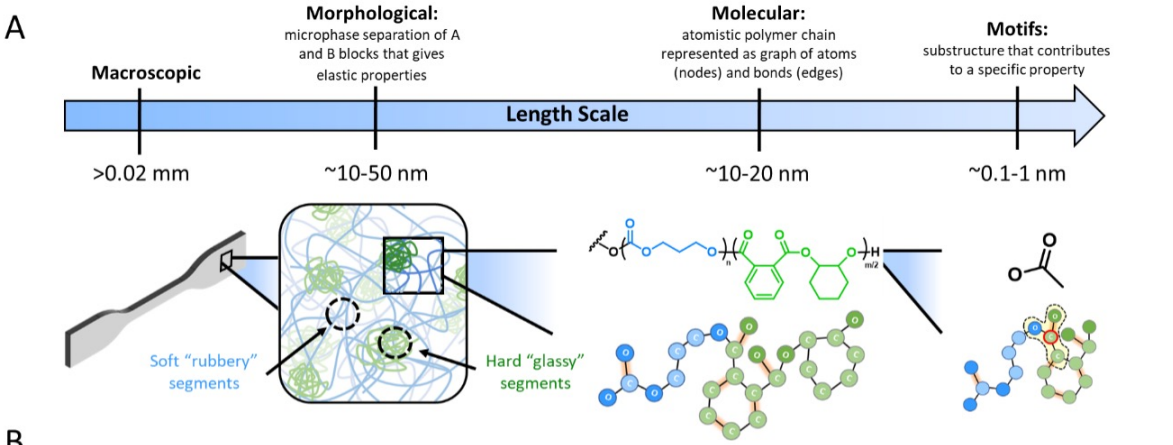
$\phi_{\text{path}}(G') = [\,3\ \ 2\ \ 3\ \ 3\ \ 1\ \ 5\ \ 0\ \ 0\ \ 4\,]$

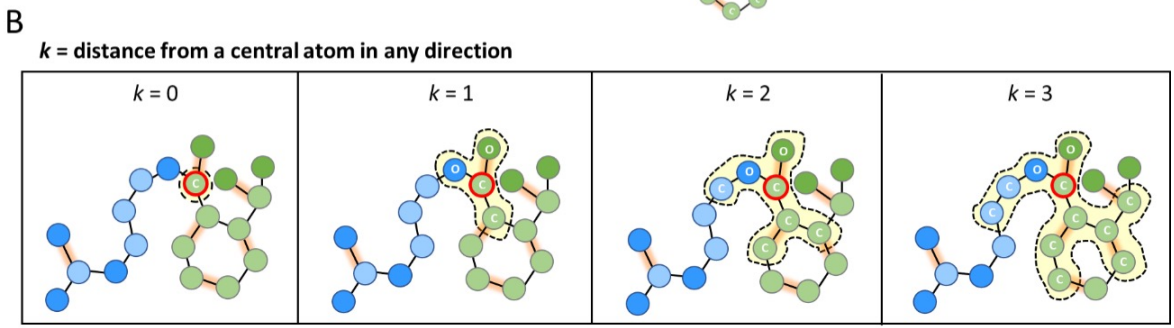
- This is one possible implementation of the shortest path kernel. For this, we have $k(G, G') = \langle \phi_{path}(G), \phi_{path}(G') \rangle = 75$.

	based on	graph type	node type	edge type	complexity
shortest path kernel Borgwardt and Kriegel, 2005	 paths	 undirected directed	 labelled	 labelled	$\mathcal{O}(n^4 d_v)$

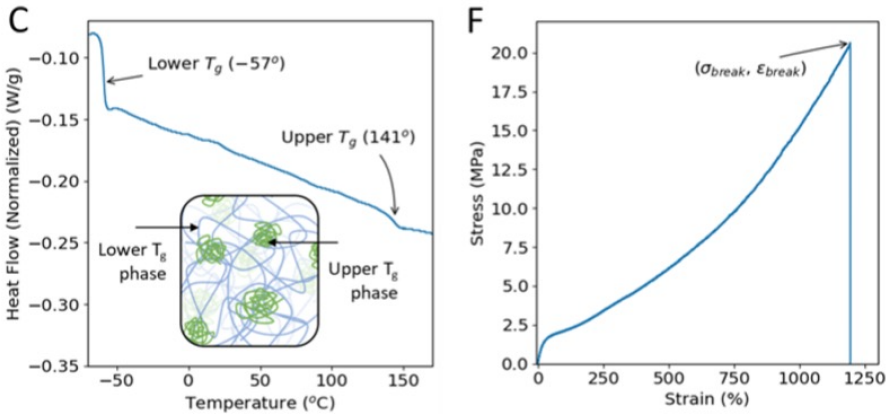
Back to Polymers

Lenght Scale of Polymers



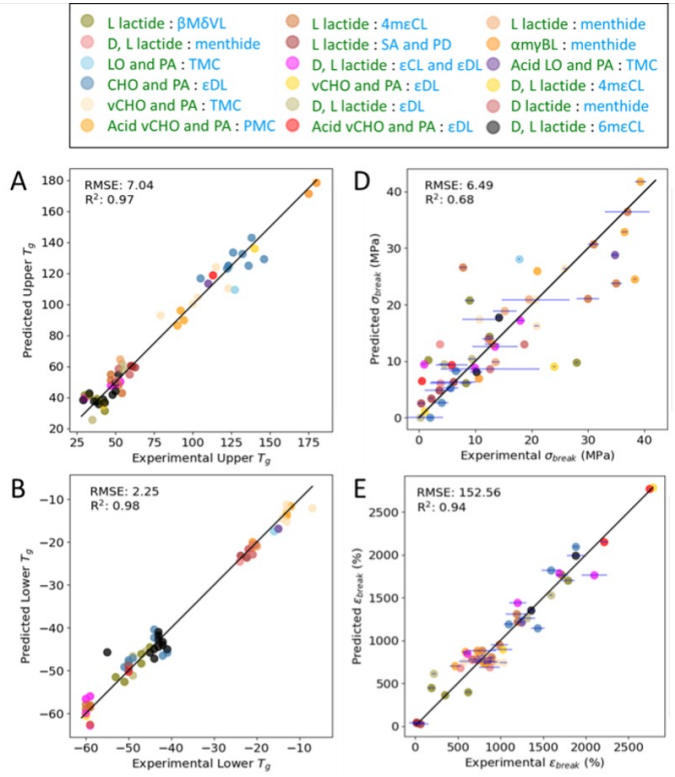


Predictions

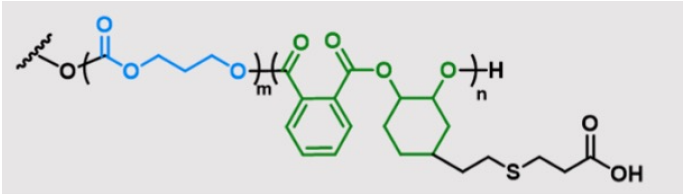


- What are we predicting? **(these are regression tasks)**
  - **Thermal Properties:** Glass Transition Temperature (Tg).
    - Both Lower and Higher.
  - **Mechanical Properties:** Stress at break and Strain at break.

Predictions

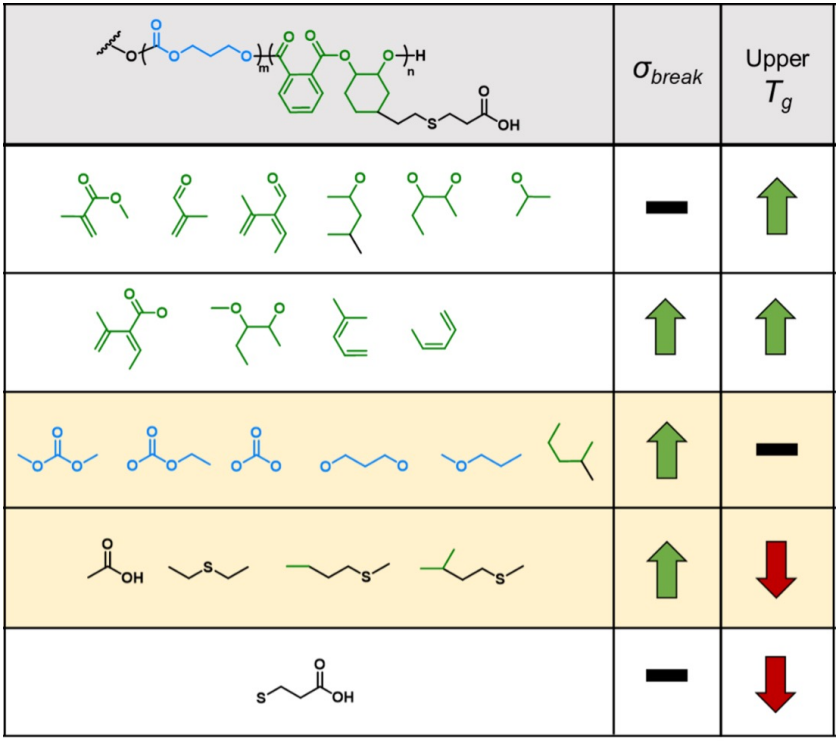


Explanations



- Consider the polymer above.
  - It is a carboxylic acid functionalized poly(ester-b-TMC-b-ester). (!)
- In lay terms, chemists **know** that:
  - The blue part (PTMC) is associated with a higher stress at break (σ_{break})
    - It undergoes strain induced crystallization.
  - The green part (PA co vCHO) is associated with a higher Upper T_g.
  - The black part (Carboxylic acid group) is associated with a higher σ_{break} and lower Upper T_g.
- Our dataset is consistent with the above, i.e., there are some examples associated with each of the claims.

Explanations given by LIME (*Ribeiro et al., 2016*)



Miscellaneous Comments and Challenges

- Transforming back into polymers.
- We could learn from the strings, for example.
- Modification in the WL algorithm needed to be done because of double-bonds.
- How to deal with aromatic rings.
- Others would not use all patterns possible patterns.

Conclusions

- Our algorithm predicts properties of polymers without the need of inputting motifs "by hand".
- We can create explanations by going from WL patterns back to polymer "chunks" (i.e., subgraphs).
- Such explanations are in line with what chemists already understand about these polymers.
- No restrictions to which polymers can be predicted.
    - Better if all atoms have been seen before by the algorithm.
- Future work: synthesising polymers is expensive and time-consuming. How can we use such an algorithm to explore thousands of possible combinations of monomers to create good (strong, elastic) polymers?

The End

Some references

[Borgwardt et al., 2020] Borgwardt, K., Ghisu, E., Llinares-López, F., O'Bray, L., and Rieck, B. (2020). Graph kernels: State-of-the-art and future challenges.

[Gärtner et al., 2003] Gärtner, T., Flach, P., and Wrobel, S. (2003). On graph kernels: Hardness results and efficient alternatives. In *Learning theory and kernel machines*, pages 129–143. Springer.

[Ribeiro et al., 2016] Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1135–1144, New York, NY, USA. Association for Computing Machinery.