Supporting Information

A graph representation of molecular ensembles for polymer property prediction

Matteo Aldeghi [1], Connor W. Coley [1,2,*]

[1] Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

[2] Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

[*] Email: ccoley@mit.edu

This PDF includes:

- Extended Methods (pp. S2–S5)
- Figures S1–S4 (pp. S6–S9)
- Table S1–S2 (pp. S10–S11)

Other supporting materials for this manuscript include the following:

- Data S1: a dataset of computed electron affinity and ionization potential values for 42,966 copolymers (CSV). The files includes SMILES strings for monomers A and B (“monoA” and “monoB” headers), their relative mole fractions (“fracA” and “fracB”), the copolymer chain architecture (“chain_arch”), the raw electron affinity and ionization potential values computed with xTB (“EA (ev)” and “IP (eV)”), and their calibrated values with respect to the standard hydrogen potential (“EA vs SHE (eV)” and “IP vs SHE (eV)”).
- An implementation of the wD-MPNN at: https://github.com/coleygroup/polymer-chemprop.
- Data and scripts needed to reproduce the results in this paper at: https://github.com/coleygroup/polymer-chemprop-data.

Extended Methods

Copolymer dataset

For each copolymer considered, we generated up to 32 sequences and 8 conformers per sequence. Sequences were generated at random using custom *RDKit*[1] (v2021.03) scripts, while maintaining the specified chain architecture and monomer ratio. When more than 32 sequences were possible, only 32 at random were kept. Conformers were generated using the ETKDG approach[2] as implemented in *RDKit*, while automatically falling back to ETDG[2] in rare failure cases. All conformers were then minimized with the MMFF94 force field[3–7], also via *RDKit*.

All the octamer structures generated as described above were further minimized with GFN2-xTB[8] using the *xtb* program[9], with energy tolerance of 5 × 10^{-5} Eh and gradient tolerance of 4 × 10^{-3} Eh/α. Vertical IP and EA properties were then computed with IPEA-xTB[10], a specially reparameterized version of GFN1-xTB[11]. All xTB calculations used the analytical linearized Poisson-Boltzmann model for implicit solvation[12,13] with the default parameters for benzene in the *xtb* code. The IP and EA values were Boltzmann averaged across the 8 conformers at 298.15 K, and then averaged across all sequences associated with a specific copolymer. Finally, IP and EA were adjusted to reflect their relative values with respect to the standard hydrogen electrode potential (4.44 V), and then corrected according to the DFT calibration in a low dielectric environment performed by Wilbraham *et al*.[14].

Architecture of the wD-MPNN

Consider a molecular graph $G = \{V, E\}$ with vertices V, edges E, node features x_v, and edge features e_{vu}, where $v \in V$ and $vu \in E$. Node features are initialized as a one-hot encoding of the atomic number, degree, formal charge, chirality, number of hydrogens, hybridization, and aromaticity of the atom, and scaled atomic mass of each atom. Bond features are initialized as a one-hot encoding of the bond type (singe, double, triple, or aromatic), whether the bond is conjugated, in a ring, and contains stereochemical information. The hidden features of directed edges h^0_{vu} are built by concatenating the features x_v and e_{vu} (i.e., those of each atom and its outgoing edge) and passing the resulting vector through a single neural network layer

$$h^0_{vu} = \tau(W_i \mathrm{cat}(x_v, e_{vu})),$$

where τ is the an activation function (here chosen to be ReLU), and $W_i \in \mathbb{R}^{h \times h_i}$ is a learned matrix with h being a user-defined hidden size (here chosen to be 300) and h_i the size of $\mathrm{cat}(x_v, e_{vu}))$. These hidden features are then updated via repeated message passing operations. Here we will use a total of $T = 3$ message passing steps. At very step $t \in \mathbb{Z}_{<T}$, updated features are obtained as

$$h^{t+1}_{vu} = \tau\left(h^0_{vu} + W_h \sum_{k \in \{N(v) \setminus u\}} w_{kv} h^t_{kv}\right),$$

where $W_h \in \mathbb{R}^{h \times h}$ is a learned matrix, $N(v) \setminus u$ are the neighbors of v aside from u, and w_{kv} are user-provided directed edge weights according to the probability of v being a neighbor of u in the polymer graph (Figure 1). The addition of h^0_{vu} at every step acts as a skip connection. For a standard

molecule, $w_{vu} = 1 \; \forall \; v, u \in V$, but for a polymer these weights allow us to distinguish between different monomer sequences, as they weigh the messages $m_{kv}^{t+1} = \sum_{k \in \{N(v) \backslash u\}} w_{kv} h_{kv}^{t}$ according to the relevance of each incoming edge.

After T message passing iterations, atoms representations h_v are obtained by summing over all incoming edge representation $h_k^T v$, concatenating them with the original atomic features x_v, and passing the resulting vector through a single neural network layer,

$$h_v = \tau\left(W_o \; \mathrm{cat}\left(x_v, \sum_{k \in N(v)} w_{kv} h_{kv}^{T}\right)\right),$$

where $W_o \in \mathbb{R}^{h \times h_o}$ is a learned matrix, h_o is the size of the concatenated vector, and w_{kv} are the weights of the incoming edges. In such a way, the output atom features depend on the probability of each incoming edge being present in the molecule ensemble. Finally, a molecule feature vector is obtained as an average (or sum) of all atom feature vectors,

$$h = \frac{1}{|V|} \sum_{v \in G} w_v h_v,$$

where w_v are weights reflecting the stoichiometric ratio between different monomers. The learned representation h is then used as the input of a feed-forward neural network to predict the molecular or polymer properties of interest. In our tests, we use a shallow network with two layers, where the first is the input layer with hidden dimension h, and the second is the output layer with dimension matching the number of labels being predicted.

Baseline representations and models

The predictive performance of wD-MPNN developed was compared to that of established ML approaches. These included the base D-MPNN model (standard Chemprop[15]), and random forest (RF) and fully-connected neural network (NN) models trained on fingerprint representations. All models were trained as multi-task models aimed at predicting both EA and IP.

Baselines for the EA and IP dataset

The D-MPNN baseline used a model with the same architecture described in Figure 2, but without information on monomer stoichiometry or the partial bonds connecting repeating units. The input for this model was a disconnected graph of the separate monomeric units. In Table 1, this model is referred to as “Monomer”, because it only captures the chemical structure of the monomeric units. The extension of this model in which weighted edges are used is referred to as “Monomer + Chain architecture”. The extension of this model in which weighted atomic embeddings are used is referred to as “Monomer + Stoichiometry”. The proposed wD-MPNN includes both these features and is thus referred to in Table 1 as “Monomer + Chain architecture + Stoichiometry”.

Extended-Connectivity Fingerprints (ECFP)[16] were used as input representations for RF and NN models, as implemented in *scikit-learn*[17] (v1.0). These were obtained with *RDKit*[1] using 2048 bits and

radius 2. We tested both binary and count fingerprints, constructed from the monomeric units alone, as well as from an ensemble of oligomeric sequences sampled uniformly at random. In the first case, the monomeric units were converted to a single *RDKit* molecule object and the fingerprints computed. This is the fingerprint equivalent of the representation provided to the baseline D-MPNN model. In the second case, we sampled up to 32 octameric sequences, while satisfying the stoichiometry and chain architecture of the polymer (Figure 3), computed fingerprints for all resulting oligomers, and used their average as input for the RF and NN models. This fingerprint representation of polymers, which is proposed in this study, attempts to capture the ensemble nature of the polymeric material. Despite the sampling procedure, binary fingerprints are not able to capture information on stoichiometry, while count fingerprints can. As shown in the Results section, this latter approach was found to be the most competitive among all baselines considered. Patel *et al*.[18] and Kuenneth *et al*.[19] proposed the use of a weighted sum of the binary fingerprints of each monomer to capture the stoichiometry of different monomer units in a copolymer. This approach is somewhat similar to ours based on count fingerprints and multiple sampled sequences, however, ours captures information on chain architecture in addition to stoichiometry.

All models were evaluated on the same cross-validation splits, which included train, validation, and test sets. Both random and monomer splits were evaluated, as discussed in the Results. No hyperparameter optimization was performed for the D-MPNN and wD-MPNN models and the validation set was used only for early stopping. The NN models used 1 or 2 hidden layers with 128 nodes, ReLU activation functions, layer normalization, a batch size of 256, and dropout rate between 0% and 50% (choices informed by previous work[20]). Parameters were optimized with Adam and a learning rate between 10^{-5} and 10^{-2}. Training was performed with early stopping, based on validation set performance, for up to 200 epochs and a patience of 50. The number of hidden layers, dropout rate, and learning rate were tuned against the validation set with 20 iterations of Bayesian optimization using *Hyperopt*[21]. The RF models were trained in two ways that differed in how the validation set was used. In the first approach, RF models with default *scikit-learn* parameters (100 trees) were trained on the concatenation of the training and validation sets. In the second approach, the RF hyperparameters were tuned against the validation set with 20 iterations of Bayesian optimization using *Hyperopt*[21], in which the number of trees (100 to 500), the maximum tree depth (5 to 20), the minimum number of samples required to split a node (2 to 6), and the minimum number of samples required to be at a leaf node (1 to 5), were optimized. Because the first approach was found to provide higher test-set performance overall (Table S2), we discuss only those results in the main text. While this selection does slightly inflate the estimated performance of the RF baselines, we found these to still be significantly inferior to those obtained with the wD-MPNN model.

Baselines for the diblock copolymer phases dataset

The D-MPNN baselines used in these experiments are the same as those used for the EA and IP prediction tasks, as described above. However, in the wD-MPNN we also introduced information on polymer size by scaling the molecular embeddings h by $1 + log(N)$, where N is the degree of polymerization. We thus considered an additional baseline model, referred to as “Monomers + Size” in Figure 5, in which the base D-MPNN also scales its molecular vector representations by chain length. The degree of polymerization for the diblock copolymer, as well as average chain lengths for each block, were obtained from the number-average molar mass of the copolymer, the volume fractions of each block, and the bulk densities of each block. Molar mass and volume fractions are reported for all

dataset entries in the Block Copolymer Phase Behavior Database[22], while bulk densities were taken from Table S1 of Arora *et al.*[23] for all entries in which they were missing.

RF models (Figure S4) as implemented in *scikit-learn*[17] (v1.0) were used with count-vector Extended-Connectivity Fingerprints[16] with radius 2 and 2048 bits using *RDKit*[1]. Fingerprint representations in which only the chemical structure of the monomers was considered are referred to as "Monomers" in Figure S4, consistently with naming used for the D-MPNN tests. In this case, the count fingerprints of the homopolymers or copolymers constituting each block were obtained separately and then summed. The effect of chain architecture was captured by building 32 cyclic dodecameric sequences by sampling from the available monomers uniformly at random, for each of the two polymer blocks separately. The average of the count fingerprints across the 32 sequences was then taken as the representation of each block. The averaged fingerprints from each of the two blocks was then summed to obtain the final diblock copolymer representation. This input representation is referred to as "Monomers + Chain architecture". The effect of block stoichiometry was captured by scaling the count fingerprints of each block by the block's mole ratio. This input representation is referred to as "Monomers + Stoichiometry". The effect of copolymer size was considered by scaling the fingerprint representation of the diblock copolymer (i.e., after summing the fingerprints of each block) by $1 + log(N)$, as it was done for the D-MPNN representations. This input representation is referred to as "Monomers + Size" in Figure S4. The richest fingerprint-based representation considered was called "Monomers + Chain architecture + Size + Stoichiometry", and used all of the above transformations of the base fingerprints representation. That is, for each block, count fingerprints averaged across 32 sampled sequences were obtained. These were then scaled by the mole fractions of the blocks, then summed, and then scaled proportionally to chain length.

In addition to the above, also pure descriptor-based representations that disregard the chemistry of the copolymer blocks were tested in conjunction with RF models. These are tagged as "no chem" in Figure S4. In this case, the input for the RF was the mole fraction of block A ("Stoichiometry") only, the average chain length of the copolymer ("Size") only, or both of these ("Size + Stoichiometry").

All models for this task were evaluated on the same stratified 10-fold cross-validation splits, in which the fraction of positive labels for each class were approximately preserved across folds. As for the EA/IP regression task, each fold included train, validation, and test set. For the D-MPNN models, the validation set was used only for early stopping, while RF models were trained on both training and validation sets together. RF classification models used 100 trees (`n_estimators=100`), the Gini criterion to evaluate node split quality (`criterion="gini"`), balanced class weights (`class_weight="balanced"`), and nodes were expanded until all leaves were pure (`max_depth=None`).

Figure S1 | Performance of the wD-MPNN and baseline models for the prediction of electron affinity (EA) and ionization potential (IP) on a monomer 9-fold cross-validation split. Each parity plot shows the computed DFT values against the ones predicted by the ML models. The parity line is shown as a black dashed line. The scatter/density plots display the predictions of each model for all folds of the 9-fold cross validation. Each fold contains a monomer A (Figure 3) that is absent from all other folds. The color intensity is proportional to the probability density, with brighter colors indicating areas with higher density of points. The average coefficient of determination (R^2) and root-mean-square error (RMSE, in eV) across all folds are shown. (a) Performance of the baseline D-MPNN model, which used a graph representation of the monomeric units as input. (b) Performance of the wD-MPNN model, which is augmented with information about chain architecture and monomer stoichiometry. (c) Performance of a random forest (RF) model that used a binary fingerprint (FP) representation of the monomeric units as input. (d) Performance of a RF model that used a binary fingerprint representation of the polymer as input, which was obtained as the average fingerprint of a set of oligomeric sequences sampled uniformly at random, while satisfying the correct stoichiometry and chain architecture of the specified polymer. (e) Performance of a RF model that used a count fingerprint representation of the monomeric units as input. (f) Performance of a RF model that used a count fingerprint representation of the polymer as input.

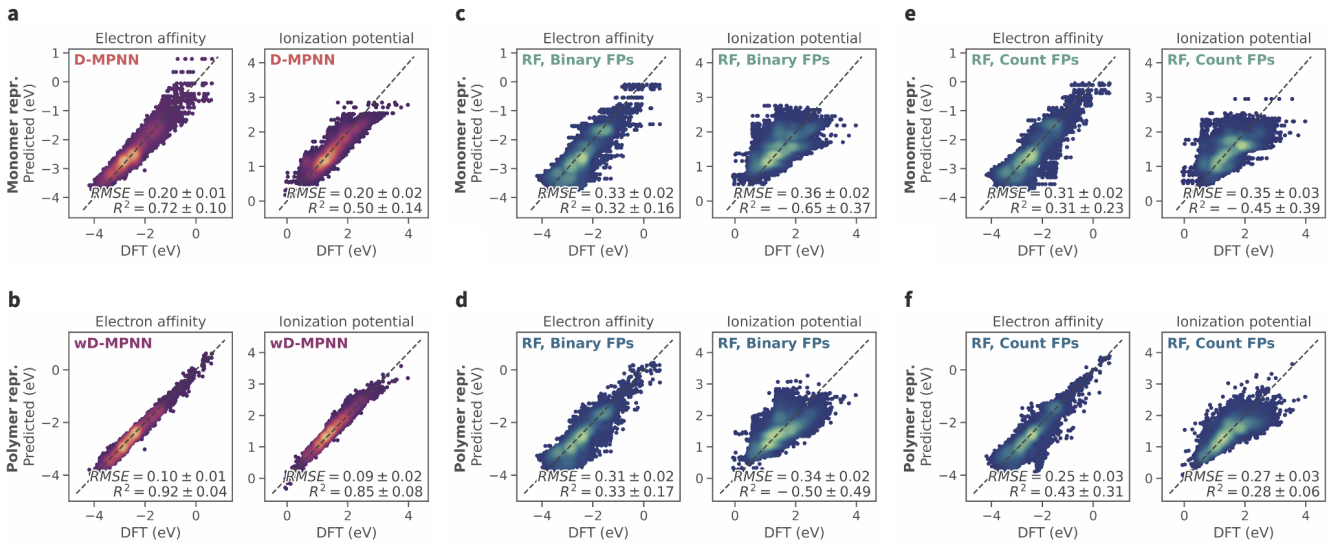

Figure S2 | Performance of the NN baseline for the prediction of electron affinity (EA) and ionization potential (IP). Each parity plot shows the computed DFT values against the ones predicted by the NN model trained on fingerprints. The parity line is shown as a black dashed line. The scatter/density plots display the predictions of each model for all folds. The color intensity is proportional to the probability density, with brighter colors indicating areas with higher density of points. The average coefficient of determination (R^2) and root-mean-square error (RMSE, in eV) across all folds are shown. (a) Results obtained on a random 10-fold cross-validation split and using binary fingerprints. (b) Results obtained on a random 10-fold cross-validation split and using count fingerprints. (c) Results obtained on a monomer 9-fold cross-validation split and using binary fingerprints. (d) Results obtained on a monomer 9-fold cross-validation split and using count fingerprints. Results obtained with wD-MPNN, D-MPNN, and RF models on the same data are shown in Figure 4 and Figure 1.

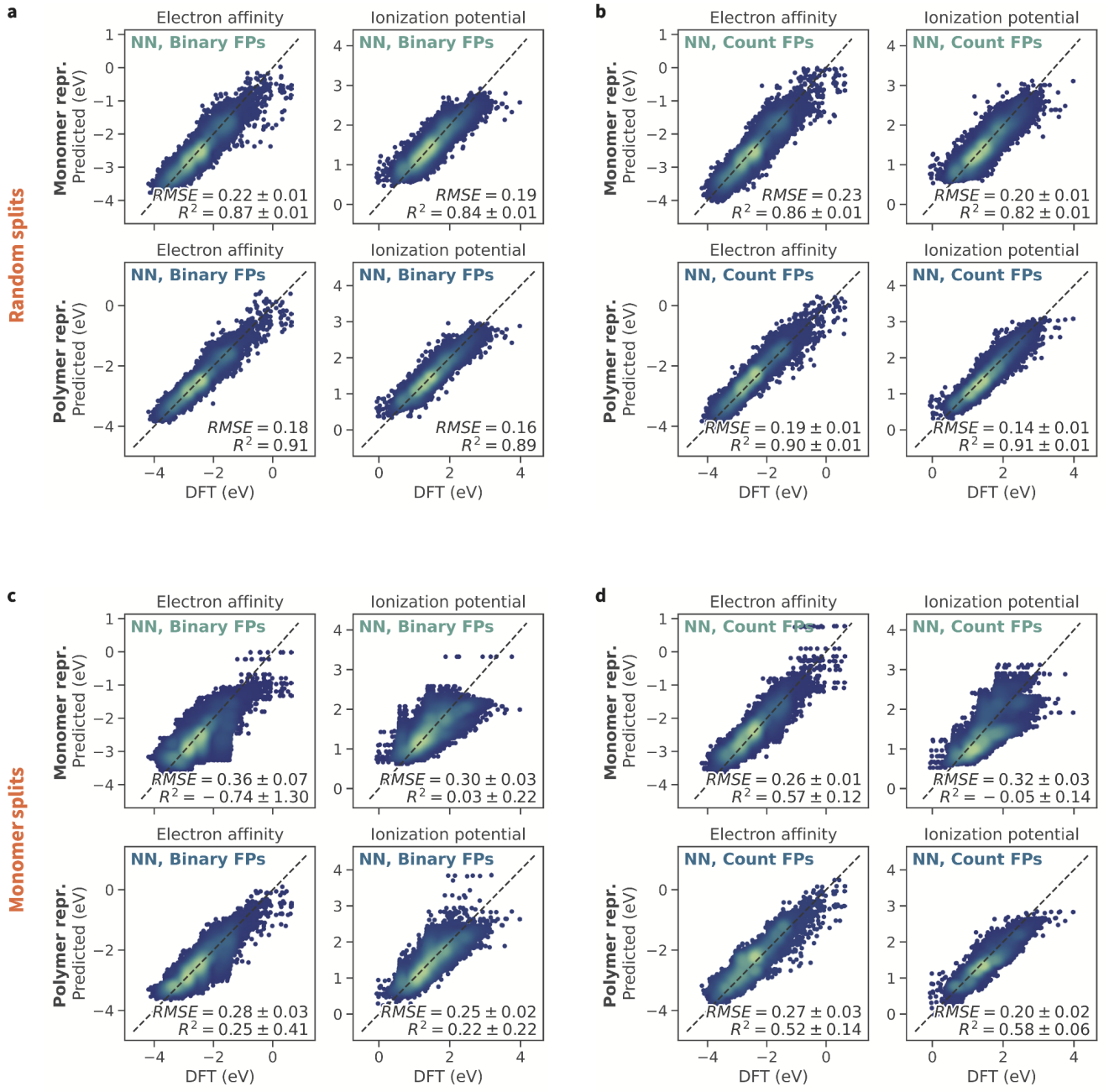

Figure S3 | Performance of the wD-MPNN model and baselines at different levels of data availability for training. Among the RF baselines, only results for the approach that used a count fingerprint representation of the polymer as input (i.e., the most competitive RF baseline found) are shown. The plots show the test set performance of these three models for multiple random splits, in which the size of the training set was increased from 0.1% (43) to 80% (34,373) of the whole dataset. The size of the validation set was kept constant at 10% (4,296). All remaining data was used as part of the test set, which thus had size between 89.9% (38,626) to 10% (4,296). 10 random splits were used. The markers in the plots show the mean test set performance across these repeated splits. Standard errors of the mean are not shown because they are small and would not be visible. The largest R^2 error is 0.03, and the largest RMSE error is 0.01 eV. Overall, these results show a cross-over point, at a training set size of 2% (859), in terms of performance between the wD-MPNN and RF approaches. While RF has better performance in the lower-data regime, the wD-MPNN overtakes RF when more data is available. Similarly, the performance of D-MPNN and wMPNN starts diverging after 1% (430) of the data is available for training. Note that the logarithmic scale on the x-axis gives the impression of an inflection point for the RF model performance at larger training set sizes; in reality, the rate of improvement decreases, resulting in diminishing returns as more data is provided, as for all other models.

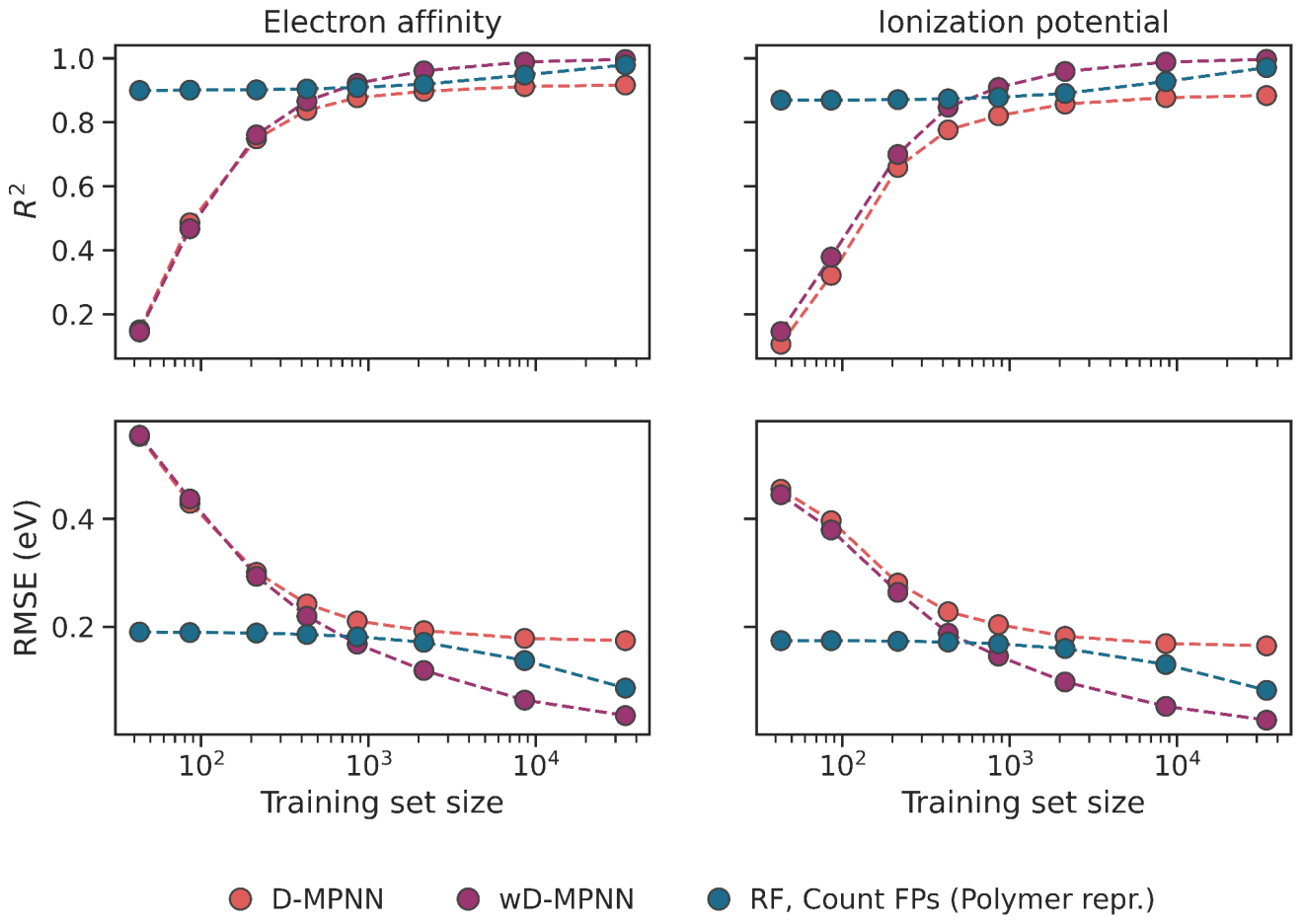

Figure S4 | Performance of multiple models and input representations on the classification of diblock copolymer phases. Performance is measured as the average area under the precision-recall curve (PRC) across the five labels (phases) to be predicted. Each marker reflects the average PRC value, for one fold of a 10-fold cross validation, across the five classification labels. The boxes show the first, second, and third quartiles of the data, and the whiskers extend to up to 1.5 times the interquartile range. Models are grouped into (i) D-MPNN, in which the one labeled as "Monomers + Chain arch. + Size + Stoichiometry" corresponds to the full wD-MPNN, (ii) RF (no chem.), in which only one or two descriptors are used as input, without considering the chemical structure of the constituting monomers, and (iii) RF, in which the chemical structure of the monomers, as well as different aspects of the polymer, are considered. A detailed explanation of each model and input is provided in the SI Extended Methods.

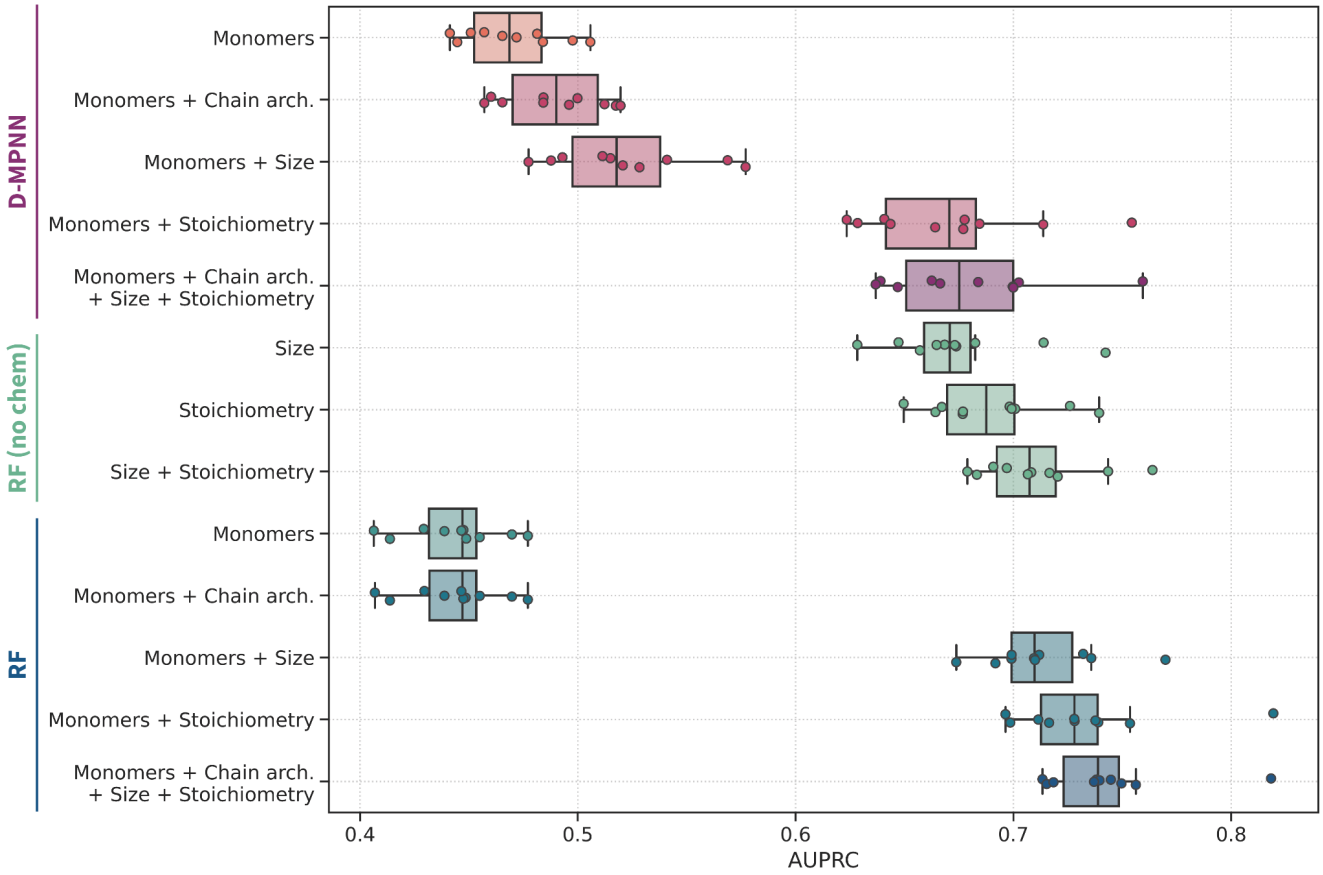

Table S1 | Performance of wD-MPNN and ablated architectures for the prediction of copolymer EA and IP values on the original and artificially inflated datasets. The average R^2 and RMSE values obtained from 10-fold cross validation based on random splits, for the prediction of IP and EA, are shown. The standard error of the mean is reported in parenthesis and it applies to the least significant digits (e.g., 0.917(1) is equivalent to 0.917 ± 0.001). Under the header "Representation", "monomers" indicates the model was provided with the graph structure of separate monomer units (i.e., the baseline D-MPNN); "chain architecture" indicates the model was provided with information on how the monomer units may connect to one another to form an ensemble of possible sequences, via the definition of edge weights, used as shown in Figure 2b and 2c; "stoichiometry" indicates the model was provided with information on monomer stoichiometry, which was used to weigh learnt node representations as shown in Figure 2d. The model using all this information altogether corresponds to the wD-MPNN. Under the header "Dataset", results for three different datasets are reported. The original dataset used EA and IP values as they were computed. In the dataset with inflated chain architecture importance, the standard deviation of EA and IP values was increased by a factor of 5 for any combination of monomer pair and stoichiometry. In the dataset with inflated chain architecture importance, the standard deviation of EA and IP values was increased by a factor of 5 for any combination of monomer pairs and chain architecture.

		Dataset					
		Original dataset		Inflated chain architecture importance		Inflated stoichiometry importance	
Representation	**Prop.**	R^2	*RMSE (eV)*	R^2	*RMSE (eV)*	R^2	*RMSE (eV)*
Monomers	EA	0.917(1)	0.173(1)	0.735(3)	0.343(2)	0.355(4)	0.760(3)
	IP	0.883(1)	0.165(1)	0.647(3)	0.333(2)	0.265(4)	0.737(2)
Monomers + Chain arch.	EA	0.929(1)	0.159(1)	0.890(1)	0.221(1)	0.356(4)	0.759(3)
	IP	0.898(1)	0.154(1)	0.861(1)	0.208(1)	0.266(3)	0.736(2)
Monomers + Stoichiometry	EA	0.987(0)	0.069(0)	0.788(2)	0.307(2)	0.971(0)	0.161(1)
	IP	0.982(0)	0.065(1)	0.714(2)	0.299(2)	0.972(0)	0.144(1)
Monomers + Chain arch. + Stoichiometry	EA	0.997(0)	0.035(0)	0.978(1)	0.098(1)	0.985(0)	0.115(1)
	IP	0.997(0)	0.027(0)	0.981(0)	0.077(1)	0.988(0)	0.094(1)

Table S2 | Performance of RF models with and without hyperparameter optimization. Average cross-validated RMSE values are shown. The standard error of the mean is reported in parenthesis and it applies to the least significant digits (e.g., 0.327(25) is equivalent to 0.327 ± 0.025). The use of hyperparameter optimization is indicated by the "w/ opt." label. In this case 20 optimization iterations with *Hyperopt* were performed. The use of a fixed set of hyperparameters is indicated by the "w/o opt." label. For this task, and a search limited to 20 iterations, there did not seem to be a performance gain with hyperparameter tuning. In fact, default hyperparameters returned slightly better performance on average.

		Cross-validation split			
		Random split		Monomer split	
Approach		*EA (eV)*	*IP (eV)*	*EA (eV)*	*IP (eV)*
RF, Binary FPs Monomer Repr.	w/o opt.	0.188(1)	0.179(1)	0.327(25)	0.362(18)
	w/ opt.	0.190(1)	0.181(1)	0.354(32)	0.399(36)
RF, Count FPs Monomer Repr.	w/o opt.	0.188(1)	0.179(1)	0.306(24)	0.353(34)
	w/ opt.	0.187(1)	0.178(1)	0.323(26)	0.378(47)
RF, Binary FPs Polymer Repr.	w/o opt.	0.151(1)	0.145(1)	0.311(19)	0.340(23)
	w/ opt.	0.163(1)	0.154(1)	0.355(15)	0.388(41)
RF, Count FPs Polymer Repr.	w/o opt.	0.086(1)	0.083(1)	0.251(25)	0.273(32)
	w/ opt.	0.107(1)	0.104(1)	0.273(30)	0.297(50)

References

1 G. A. Landrum, *RDKit: Open-source cheminformatics*, .
2 S. Riniker and G. A. Landrum, *J. Chem. Inf. Model.*, 2015, **55**, 2562–2574.
3 T. A. Halgren, *Journal of Computational Chemistry*, 1996, **17**, 490–519.
4 T. A. Halgren, *J. Comput. Chem.*, 1996, **17**, 520–552.
5 T. A. Halgren, *J. Comput. Chem.*, 1996, **17**, 553–586.
6 R. B. N. Thomas A. Halgren, *J. Comput. Chem.*, 1996, **17**, 587–615.
7 T. A. Halgren, *J. Comput. Chem.*, 1996, **17**, 616–641.
8 C. Bannwarth, S. Ehlert and S. Grimme, *J. Chem. Theory Comput.*, 2019, **15**, 1652–1671.
9 P. Atkinson, C. Bannwarth, F. Bohle, G. Brandenburg, E. Caldeweyher, M. Checinski, S. Dohm, S. Ehlert, S. Ehrlich, I. Gerasimov, S. Grimme, C. Hölzer, A. Katbashev, J. Koopman, C. Lavinge, S. Lehtola, F. März, M. Müller, F. Musil, H. Neugebauer, J. Pisarek, C. Plett, P. Pracht, F. Pultar, J. Seibert, P. Shushkov, S. Spicher, M. Stahn, M. Steiner, T. Strunk, J. Stückrath, T. Rose, J. Unsleber, *xtb*, .
10 V. Ásgeirsson, C. A. Bauer and S. Grimme, *Chem. Sci.*, 2017, **8**, 4879–4895.
11 S. Grimme, C. Bannwarth and P. Shushkov, *J. Chem. Theory Comput.*, 2017, **13**, 1989–2009.
12 F. Fogolari, A. Brigo and H. Molinari, *J. Mol. Recognit.*, 2002, **15**, 377–392.
13 R. Tuinier, *J. Colloid Interface Sci.*, 2003, **258**, 45–49.
14 L. Wilbraham, E. Berardo, L. Turcani, K. E. Jelfs and M. A. Zwijnenburg, *J. Chem. Inf. Model.*, 2018, **58**, 2450–2459.
15 K. Yang, K. Swanson, W. Jin, C. Coley, P. Eiden, H. Gao, A. Guzman-Perez, T. Hopper, B. Kelley, M. Mathea, A. Palmer, V. Settels, T. Jaakkola, K. Jensen and R. Barzilay, *J. Chem. Inf. Model.*, 2019, **59**, 3370–3388.
16 D. Rogers and M. Hahn, *J. Chem. Inf. Model.*, 2010, **50**, 742–754.
17 F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot and É. Duchesnay, *J. Mach. Learn. Res.*, 2011, **12**, 2825–2830.
18 R. A. Patel, C. H. Borca and M. A. Webb, *Mol. Syst. Des. Eng.*, 2022, -.
19 C. Kuenneth, W. Schertzer and R. Ramprasad, *Macromolecules*, 2021, **54**, 5957–5961.
20 L. Wilbraham, R. S. Sprick, K. E. Jelfs and M. A. Zwijnenburg, *Chem. Sci.*, 2019, **10**, 4973–4984.
21 J. Bergstra, D. Yamins and D. Cox, in *Proceedings of the 30th International Conference on Machine Learning*, eds. S. Dasgupta and D. McAllester, PMLR, Atlanta, Georgia, USA, 17--19 Jun 2013, vol. 28, pp. 115–123.
22 N. J. Rebello, A. Arora, H. Mochigase, T.-S. Lin, D. J. Audus and B. D. Olsen, .
23 A. Arora, T.-S. Lin, N. J. Rebello, S. H. M. Av-Ron, H. Mochigase and B. D. Olsen, *ACS Macro Lett.*, 2021, **10**, 1339–1345.