PAPER

PolyCL: contrastive learning for polymer representation learning *via* explicit and implicit augmentations†

Jiajun Zhou, [iD] [a] Yijie Yang, [iD] [a] Austin M. Mroz [iD] [ab] and Kim E. Jelfs [iD] *[a]

Polymers play a crucial role in a wide array of applications due to their diverse and tunable properties. Establishing the relationship between polymer representations and their properties is crucial to the computational design and screening of potential polymers *via* machine learning. The quality of the representation significantly influences the effectiveness of these computational methods. Here, we present a self-supervised contrastive learning paradigm, PolyCL, for learning robust and high-quality polymer representation without the need for labels. Our model combines explicit and implicit augmentation strategies for improved learning performance. The results demonstrate that our model achieves either better, or highly competitive, performances on transfer learning tasks as a feature extractor without an overcomplicated training strategy or hyperparameter optimisation. Further enhancing the efficacy of our model, we conducted extensive analyses on various augmentation combinations used in contrastive learning. This led to identifying the most effective combination to maximise PolyCL's performance.

1 Introduction

Polymers, with their remarkable diversity and extensive adaptability, have emerged as a key material class across various applications,[1] including medicine and medical devices,[2] agriculture,[3] solar cells,[4] and electronics.[5] Polymers are made from combinations of small, organic molecule-based monomeric building blocks and thus there is an enormous chemical space to be explored. The complexity of polymers can also be reflected in the extended polymer material space, including the variety of processing and synthetic conditions for the production of polymer products that can vary their performance.[6–8] The effective exploration of the extensive chemical space of polymers is a major challenge in the discovery of functional polymers for target applications. Indeed, this space is far too large to feasibly explore with conventional trial-and-improvement experimental approaches alone. The integration of computational modelling and machine learning has significantly accelerated this process, enabling the rapid identification of promising candidates.[9,10] However, there exist many challenges to training robust ML models for polymer property prediction,

including limited high-quality data,[11] scarcity of data in a specific property space,[12] and highly diverse polymer representations.[10,13–15] Indeed, polymer representations pose a challenge for many reasons, including difficulties describing repeating structures built from monomeric units, and the lack of representations that incorporate macroscopic packing.[11,14]

The design of polymer representations, which refers to the machine-readable way that the molecular features of polymers are encoded, is critical for the performance of property prediction models. Conventional methods for creating machine-readable polymer representations involve creating handcrafted fingerprints, where molecular structural information features are depicted through manually designed descriptors.[16] Indeed, several types of handcrafted fingerprints[17–19] and refined fingerprint strategies[10,15] within this category have found success in the polymer literature. While these methods have found success, handcrafted fingerprints are often designed using the expert's chemical intuition and heuristic principles. In addition to the potential to introduce bias, these methods are fairly labour-intensive and time-consuming.

Deep neural networks have been increasingly used to automatically extract dense molecular representations from polymers.[20] This approach leverages the power of deep learning to alleviate the aforementioned challenges associated with manual feature extraction. Polymers can be abstracted into molecular graphs.[13,21,22] Alternatively, molecules can be converted to one-dimensional sequence representations, such as SMILES.[23] Here, polymer-SMILES representations are used that include brackets with a special character "[*]" to represent

[a]Department of Chemistry, Molecular Sciences Research Hub, Imperial College London, White City Campus, Wood Lane, London, W12 0BZ, UK. E-mail: k.jelfs@imperial.ac.uk

[b]I-X Centre for AI in Science, Imperial College London, White City Campus, Wood Lane, London, W12 0BZ, UK

† Electronic supplementary information (ESI) available. See DOI: https://doi.org/10.1039/d4dd00236a

connection points between monomers, to reproduce the repeating nature of these materials (example shown in Fig. 1). For example, a Long Short-Term Memory (LSTM) model was trained on polymers represented as SMILES strings for property prediction.[10,24] In addition, BigSMILES was designed to extend SMILES for the representation of the stochasticity of polymer molecules by introducing additional notations.[14] Deep neural networks have also been employed with tailored representations of polymers to predict specific properties, such as degree of polymerization, monomer stoichiometry, gas permeability and radius of gyration.[25–27]

Machine learning-based predictive models are typically trained in a supervised fashion and act as automatic feature extractors. While this supervised training pattern is beneficial for specific downstream tasks, it may lead to learnt representations exhibiting domain-dependent characteristics and suffering from limited generalisability to other tasks.[28–31] Further, supervised learning methods rely on labelled data of both high quantity and high quality. Within (polymer) chemistry, acquiring high-quality, labelled data is resource-intensive. The scarcity of labelled data in chemistry[32] may lead to overfitting and, therefore, impair the model's generalisability to other data in the target domain.[33] The limitations of supervised learning directly motivates self-supervised learning for chemical property prediction.

Self-supervised models learn from the inherent structure of the data, without the requirement of data labelling.[34] In polymer science, the value of creating a universal representation that is a target-agnostic feature of self-supervised learning has already been observed.[35] Initial demonstrations of self-supervised learning in polymer science have largely focused upon transformer architectures.[36] Masked language modeling,[37] where random tokens are obscured from the input to be predicted by the transformer, served as the training strategy to guide the pre-training of transformers. This training strategy was proven effective in Transpolymer and polyBERT for the production of machine-learnt polymer representations using transformer architecture.[16,35] However, these works did not directly assess the effectiveness of the representation learnt by their pre-trained strategies and only inferred the quality of the learned representation from the performance of the downstream tasks using the model. Yet, inferring representation performance from the model performance is especially challenging when the model includes complex ML techniques, such as data-augmentation (including non-canonical SMILES strings to increase the size of the dataset),[35] and multi-task learning (training downstream tasks on all datasets simultaneously).[16]

Contrastive learning is among the most competitive forms of self-supervised learning that learn meaningful representations from comparing and contrasting data.[38] The idea of contrastive learning is to pull together similar data samples and separate dissimilar samples in the representation space.[39] This idea has been demonstrated to achieve successful representation learning in molecular systems.[40–42] In addition, contrastive learning is capable of incorporating extra modality to form modality pairs such as structures and text description,[43] SMILES and IUPAC names,[44] SMILES and the molecular graph,[45] into the molecular representation *via* multi-modal alignment. However, contrastive learning is yet, to the best of our knowledge, to be applied to polymer science.

As illustrated in Fig. 2, an efficient approach to contrastive learning entails the formation of positive pairs by creating two distinct representations of the same, original polymer-SMILES molecule (here, termed anchor molecule) through data augmentation. This process is critical as it enables the learning model to recognise and reinforce the essential features of the molecule by comparing these different views. Concurrently, the anchor molecule and other molecules in the current (with their respective positive pair) are automatically considered negative pairs.[46] Thus, the construction of positive pairs is exceptionally important because their formation directly impacts the identification of negative pairs, which are imperative to helping the algorithm understand the relationship between different positive pairs and, ultimately, better map out the representation space.

In chemistry, common approaches to augmentation are explicit – allowing observable modifications to the representation structure (*e.g.* removing a token from a SMILES string). Typical explicit augmentation modes for molecular graphs include node dropping, edge masking/perturbation, attribute masking and subgraph extraction.[40,45] Implementation of explicit augmentation methods for SMILES representations remain limited and under-explored.[45] In addition, augmentation can also be implemented in an implicit fashion, where different perturbations to the embedding are implemented during the training process (*e.g.* natural dropout).[47,48] Despite the demonstrated effectiveness of implicit augmentation, this approach also remains an area of limited attention. Furthermore, there is a need to understand the effects arising from the heterogeneous combination of both types of augmentation strategies (*i.e.* implicit and explicit).

Here, we present PolyCL, a contrastive learning framework for polymer representation learning for improved predictive performance. To construct effective positive pairs, we also proposed a novel combinatorial augmentation strategy to include both explicit and implicit augmentations. Our results show that PolyCL outperforms other supervised and pre-trained models under the lightweight and flexible transfer learning setting where the fine-tuning of PolyCL is not required. Here, we emphasise that this construction eliminates the need to fine-tune an entire model (*e.g.* pre-trained model + prediction head), instead PolyCL may be independently implemented as a feature extractor for polymer property prediction tasks. We find that the learnt representation from our contrastive learning
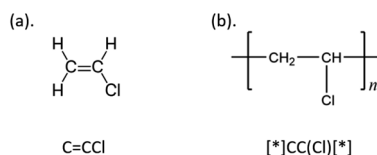

Fig. 1 Example of (a) SMILES of vinyl chloride and (b) polymer–SMILES of polyvinyl chloride, along with the corresponding chemical structures.
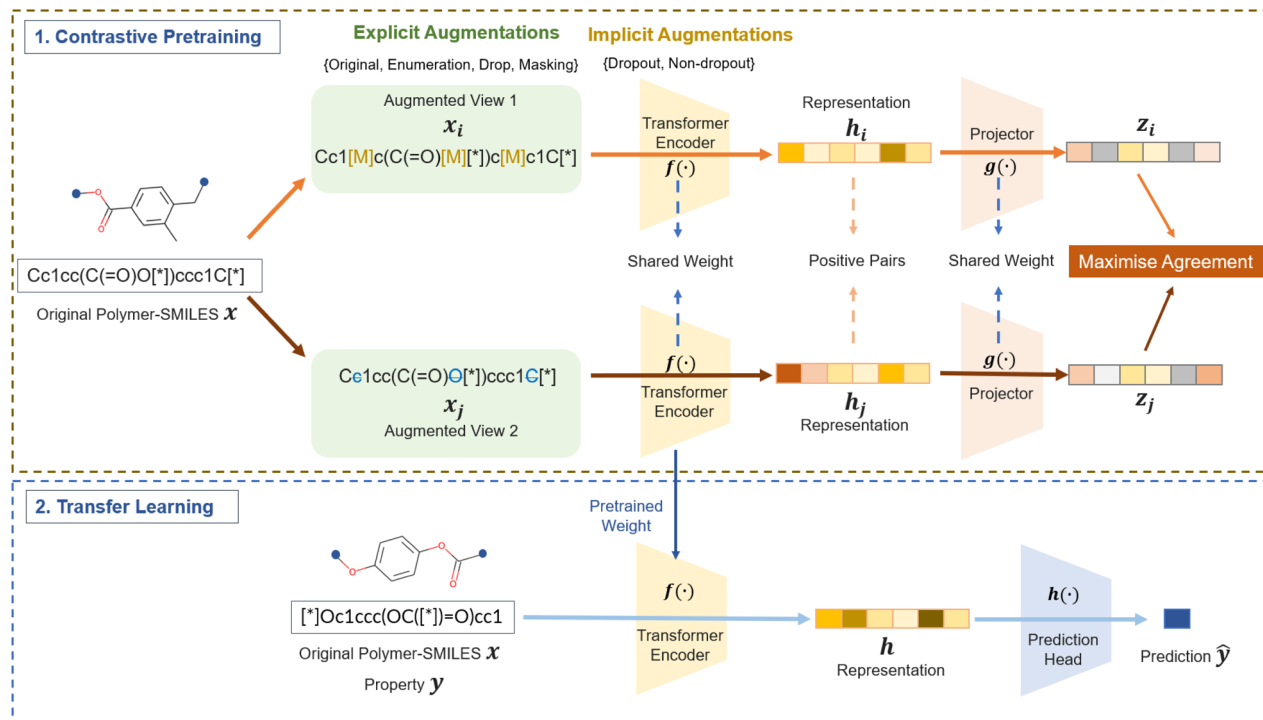
Fig. 2 A schematic illustration of the PolyCL pipeline. (1) Polymer contrastive representation learning with different augmentation strategies for constructing effective positive pairs. The agreement of positive pairs projected to their latent representations is maximised by the loss function of contrastive learning. Masking and drop in augmented views 1 and 2 are shown as sample explicit augmentations for the input original polymer–SMILES. (2) Transfer learning by leveraging the acquired polymer representation to apply in the prediction of downstream tasks.

strategy has improved quality, exhibits practical robustness to non-canonized SMILES input, and show how our polymer representation can be used for a variety of downstream tasks *via* simple transfer learning. The dataset and model are available at: **http://github.com/JiajunZhou96/PolyCL**.

2 Methods

2.1 Dataset

We randomly selected 1 million polymers from the unsupervised polymer dataset curated by Xu *et al.*[35] to use as the pretrain dataset for contrastive learning. Datasets for downstream regression tasks were sourced from data by Xu *et al.*[35] to benchmark against other models. Specifically, we focused on homopolymer datasets, where the inputs are comprised only of the SMILES strings of the monomers. For extension of the approach to copolymers or multi-component polyelectrolyte systems in the future, extra descriptors can be easily concatenated with the polymer representations produced from our model to collaboratively encode additional information. We used seven different property datasets covering a wide range including band gap (both chain (E_{gc}) and bulk (E_{gb})), electron affinity (E_{ea}), ionisation energy (E_i), Density Functional Theory (DFT)-calculated dielectric constant (EPS), crystallisation tendency (X_c), and refractive index (N_c). We did not use any data augmentation strategy to boost our downstream datasets. All datasets except for X_c were originally calculated using DFT. X_c

was obtained by combining experimental heat of fusion and a group contributions method.[12,49]

2.2 Polymer encoding

Polymers are often linearly concatenated by the repeating units of monomers, exhibiting inherently sequential structures.[50] Therefore, there are advantages to representing a polymer as a sequence-based molecular representation. SMILES strings[23] are commonly employed for depicting individual monomers within polymers. Different to the representation of small molecules, polymers necessitate the explicit indication of connecting points between monomers. As we start the training process with the pre-trained checkpoint of polyBERT,[16] we maintained the use of polymer-SMILES to make full use of the model. In comparison, polymer-SMILES extends the traditional SMILES representation by marking connecting points with the special token "[*]", following the standard syntactic rules of the SMILES format. Subsequently, the input polymer-SMILES were encoded by the pre-trained polyBERT model with the corresponding tokeniser,[16] which is a variation of the Deberta-v2 (ref. 51) language model with a transformer architecture.[36]

2.3 Contrastive representation learning objective

To effectively guide the training of the model to the intended objective, we applied the normalised temperature-scaled cross-entropy (NT-Xent) loss.[46] In a batch consisting of $2N$ semantically similar views derived from N samples, for each positive

pair (i, j), the remaining $2(N - 1)$ samples in the batch are implicitly considered as negative examples. Therefore, the NT-Xent loss for a positive pair (i, j) is described by

$$\mathscr{L}_{i,j} = -\log \frac{\exp\left(\text{sim}\left(z_i, z_j\right) / \tau\right)}{\sum_{k=1}^{2N} \mathbb{1}_{\{k \neq i\}} \exp\left(\text{sim}\left(z_i, z_k\right) / \tau\right)} \quad (1)$$

where z_i, z_j are the representations of two positive data samples, $\text{sim}(u, v)$ denotes the cosine similarity $\frac{u^T v}{\|u\| \|v\|}$, τ is the temperature parameter, which is empirically set to 0.05. An indicator function $\mathbb{1}_{\{k \neq i\}}$ is used to skip the case where both k and i refer to the same sample.

Here, we used the pre-trained PolyBERT[16] as our encoder, $f(\cdot)$, and maintained the default settings for all hyper-parameters in the transformer architecture. The projector $g(\cdot)$ is a two-layer MLP that maps the pooled 600-dimensional representation h to a 128-dimensional latent vector z for similarity evaluations. During the contrastive pre-training, we enabled mixed precision training. AdamW[52] was used as the optimiser with a learning rate of 1×10^{-5} to minimise the NT-Xent loss. A gradient clipping mechanism was employed with a max grad norm set to 1.0. We trained the model for 10 epochs in total, taking approximately 22 hours on 8 NVIDIA Tesla V100 GPUs.

2.4 Constructing augmentations

Contrastive learning can be enhanced by the use of effective data augmentation modes, a benefit observed across various data modalities.[40,45,46] The challenge for contrastive learning is the construction of effective positive pairs. This can be achieved by applying augmentation strategies to create different views of the same polymer molecules, which should subtly alter the attributes of the polymer representations. We aim to create differences in two vectors of polymer representations h_i and h_j, while preserving the key semantic information referring to the original polymer molecule x. In this case, the use of the original molecule can be considered as the baseline.

Augmentations can be empirically categorised into two modes; "explicit" and "implicit". Explicit augmentations are direct and observable modifications to the input data. As shown in Fig. 2, explicit augmentations include enumeration, token masking (Masking) and token drop (Drop). Enumeration randomly generates one non-canonical version of a polymer SMILES based on its canonical SMILES string. Drop deletes 10% of tokens in the SMILES string. Masking substitutes 10% of tokens in the SMILES string with a special token. Therefore, the same molecule can be transformed into two different SMILES strings to construct an effective positive pair.

Beyond explicit augmentations, the subtle modifications in how the input data is represented in the intermediate layers within the model are referred to as implicit augmentations, as shown in Fig. 2. Following the work of SimCSE,[46] we used the inherent dropout module inside our transformer encoder to create differences in molecular embedding for the same input. With implicit augmentations enabled, the dropout ratio for hidden layers and attention probabilities in the configuration of

the transformer encoder is 0.1; when disabled, both values are set to 0. In addition, we have also combined both explicit and implicit augmentations for the construction of positive pairs to study the cooperative effect of augmentation strategies.

We employed a two-step strategy to identify the best-performing combinations of augmentations. First, we applied our contrastive pretraining approach to all possible combinations of explicit augmentations. Subsequently, we evaluated the predictive performance of the resulting models across all downstream tasks within a transfer learning framework. The contrastive learning model pretrained without augmentations served as the baseline for comparison. We assessed the number of downstream tasks in which each model outperformed the baseline. Based on this evaluation, we selected the most promising combinations of explicit augmentations as candidates. These combinations were then integrated with implicit augmentations for further analysis and evaluations of performance.

2.5 Transfer learning

We used transfer learning to evaluate the quality of learnt representations. We fine tune the prediction head and leave the pre-trained model unchanged during transfer learning. In the implementation of this approach, all trainable parameters in the pre-trained model were frozen and gradients were turned off before the training of transfer learning models of downstream tasks.

The experimental setup employs an MLP regressor featuring a single hidden layer and ReLU activation, integrated with the PolyCL feature extractor. "[CLS]" pooling serves as the readout function, extracting a 600-dimensional polymer representation. Specifically, this approach transforms token-level embeddings for each polymer sequence into a comprehensive sentence-level embedding, wherein sequence information is encapsulated by the appended "[CLS]" token. The hidden size within the MLP is consistent with the input size for all pre-trained models (including the benchmarking study of PolyBERT and Transpolymer). A dropout ratio of 0.1 is applied polymer-SMILES strings are encoded by the tokeniser of PolyBERT.[16] An l_2 loss function is implemented for regression tasks. During the regression phase, AdamW[52] was used as the optimiser with a learning rate of 0.001 and no weight decay. For each downstream dataset, a 5-fold cross-validation strategy is employed, accompanied by a 500-epoch training protocol. An early-stopping monitor is activated after 50 epochs, with a patience setting of 50 epochs. The performance on the unseen validation datasets is evaluated using the root-mean-square error (RMSE) and the coefficient of determination (R^2). To show the general expressiveness of the learnt representation, all hyperparameters for the transfer learning performed on PolyCL are set by simple heuristics and not tuned specifically by a validation process.

2.6 Alignment and uniformity

The quality of the learned representation can be alternatively evaluated by the quantitative metrics of alignment and uniformity introduced by Wang and Isola.[53] Alignment refers to the

distance between known positive pairs $(x, x^+) \sim p_{pos}$, as shown in eqn (2). A lower alignment value between positive pairs indicates improved feature similarity:

$$\ell_{align} \triangleq \mathop{\mathbb{E}}_{(x,x^+)\sim p_{pos}} \|f(x) - f(x^+)\|^2 \tag{2}$$

where x is a polymer-SMILES, x^+ is a known positive view to x. $f(x)$ is a neural encoder to transfer polymer-SMILES to a representation. $\mathbb{E}$ is the expectation.

Uniformity is a measure of the distribution of learnt representations in the unit hypersphere; this is defined by the log of the mean Gaussian potential between each embedding pair $x, y \mathrm{i.i.d.} \sim p_{data}$, where each variable in the pair is an independent and identically distributed random variable, as shown in eqn (3). A lower uniformity indicates the learnt embedding distribution is capable of preserving maximal information:

$$\ell_{uniform} \triangleq \log \mathop{\mathbb{E}}_{x,y \overset{\mathrm{i.i.d.}}{\sim} p_{data}} e^{-2\|f(x)-f(y)\|^2} \tag{3}$$

To effectively evaluate the alignment and uniformity, a dataset distribution that has never been seen by any of the pre-trained models needed to be constructed to ensure a fair cross-model comparison. Here, we randomly sampled 60 000 polymers from the excluded development dataset of polyBERT[16] for evaluation. Each polymer was augmented once by the SMILES enumeration method to create a positive pair, thereby preserving semantics.

2.7 Benchmarking other models

The implementation details of all supervised learning models are shown in ESI Section S7,† including random forest, XGBoost, neural networks, GCN and GIN. For all pre-trained models, we froze all parameters and consolidated the pooling method as "[CLS]" pooling. We also used a simple MLP regressor as shown in Section 2.5 with only adaptation on the input layer size to fit different sizes of input representations from different pre-trained models.

3 Results

3.1 Polymer contrastive learning

The PolyCL architecture for obtaining a machine-learned polymer representation is shown in Fig. 2. The detailed neural network modules are shown in Fig. S1.† In the pre-training phase, the repeating units of polymers were encoded to polymer-SMILES, x.[16] Then, we converted each original x into two views x_i and x_j, i.e. positive pairs in two branches of the model. All views are processed by a transformer encoder $f(\cdot)$ to obtain the contextualised embedding. Here, we used the pre-trained polyBERT model[16] as the encoder to obtain a more effective prior than random initialisation, for subsequent fine-tuning by our PolyCL framework. Then, we applied [CLS] pooling, which generates compressed representations of the polymer-SMILES[46,54] on the contextualised embedding to obtain the polymer representation h_i and h_j. The projector was introduced by SimCLR,[46] which inspired the architecture of PolyCL. Here,

these pooled representations h_i and h_j are further projected as z_i and z_j using a projector $g(\cdot)$ into a latent space. Additionally, any pairs in which the source instances within each pair originate from different original polymer molecules are considered negative pairs. The objective function of contrastive learning is the normalised temperature-scaled cross-entropy (NT-Xent) loss, aiming to develop machine-learned representations by attracting positive pairs while distancing negative pairs in the latent space.[55]

In the transfer learning phase, we extracted the representation using the pre-trained model and then used a prediction head to predict any property of interest. The pre-trained transformer encoder is employed to encode polymers to their representations. Here, we demonstrate how the prediction process using a simple prediction head $h(\cdot)$, constructed with two-layered multi-layer perceptrons (MLP) with random initialisation, can be used to train the mapping from polymer representations to properties $\hat{y}$. However, the transfer learning process is flexible in selecting predictive models that best serve the requirements of downstream tasks.

3.2 Transfer learning results

The primary objective of our study is to create an effective and expressive machine-learnt representation for polymers. Transfer learning is employed to assess the utility of knowledge extracted from a pre-trained model. To evaluate the expressiveness of the representation, polymer representations produced by PolyCL are directly adopted without any task-specific refinement. In practice, we achieved our objective by fine-tuning only the prediction head, while keeping all parameters of the pre-trained model frozen.

There are two key advantages to this strategy. Firstly, this approach ensures that the representation is independent of the further fine-tuning of the underlying pre-trained model during the transfer learning, allowing for a fairer evaluation of the representation's quality. Secondly, this approach aligns with common real-world applications better; typically, only the polymer representation is incorporated into subsequent models, instead of the specialised fine-tuning of the pre-trained model with these later models. PolyCL should serve simply as a flexible representation generator and, therefore, requires no extra computational resources to fine-tune pre-trained parameters during usage. To explore how our model performs, we compared PolyCL with supervised models including random forest (RF), XGBoost (XGB), and neural networks (NN), each trained on either ECFP fingerprints[56] or the domain-specific Polymer Genome (PG) fingerprints.[12] We also implemented cross-modal comparison between the above fingerprints and graph representations encoded by graph convolutional networks (GCN)[57] and graph isomorphic networks (GIN).[58] Finally, we compared with other machine-learnt representations via self-supervised learning strategies: PolyBERT[16] and Transpolymer.[35]

The results of our transfer learning are shown in Table 1. We conducted our transfer learning on seven different datasets sourced from Xu et al.,[35] including band gap (both chain (E_{gc})

and bulk (E_{gb})), electron affinity (E_{ea}), ionisation energy (E_i), DFT-calculated dielectric constant (EPS), crystallisation tendency (X_c), and refractive index (N_c). Following previous works,[12,35] we assessed the five-fold average R^2 on the unseen validation datasets. Among seven supervised models and three self-supervised models, PolyCL achieves the overall best R^2 and four individual best performances across the seven property datasets. PolyCL has a significant advantage in predictive performances over the second-best model in the ionisation energy (E_i), dielectric constant (EPS) and refractive index (N_c) datasets, by 2.4%, 2.4%, 4.3%, respectively. This performance shows that the chemical and structural information captured in the SMILES representation by our model can be generalised to different types of properties, and help to construct more efficient models for quantitative structure–activity relationships. Therefore, the contrastive learning strategy enables the generation of a more expressive representation.

Compared with supervised learning methods, polymer representations produced by self-supervised learning achieved a higher overall performance (Avg. R^2) and robustness across all datasets. Only Polymer Genome (PG) fingerprints[12] can reach comparable performance in given tasks – specifically, the band gap for a chain (E_{gb}) and in the bulk (E_{gc}). However, this fingerprinting method also shows lower robustness, which is observed considering its prediction on different datasets, for example, crystallisation (X_c). ECFP generally exhibits superior overall performance and robustness compared to PG fingerprints; however, in specific tasks involving prediction, PG fingerprints tend to outperform due to their higher target-specificness. In addition, our implementation of graph neural networks suggests that graph representation remains an efficient way to represent polymers; however, these representations do not show better predictive performance than traditional fingerprinting methods.

Furthermore, we replaced the canonical polymer SMILES in all downstream datasets with one of their enumerated SMILES strings and used these modified datasets to evaluate models on the transfer learning setting as shown above. Since non-canonicalized SMILES are commonly encountered in practical applications, this modification reflects real-world usage. The results are summarized in Table 2. Compared with other self-supervised learning strategies, PolyCL shows the most robust results across all datasets. Compared with the use of canonicalized datasets, the use of non-canonicalized datasets leads to an average performance decay of 4.7% and 8.3% for Transpolymer and PolyBERT, respectively. In contrast, the performance of PolyCL is still maintained at a comparable level with a slight performance decrease of only 2.6%. The results demonstrate that the polymer representation obtained by PolyCL outperforms other self-supervised learning strategies in terms of robustness and invariance to different molecular representations.

As an additional assessment, we also assessed the results of fine-tuning, which means that all parameters in both the pre-trained model and the prediction head are unfrozen and fine-tuned (as shown in Fig. S3†). Although fine-tuning is not our focus here, we show that our model achieves competitive results compared with other self-supervised models, including polyBERT and Transpolymer, in this experimental setting.

3.3 Effect of augmentation combinations

The combination of augmentation modes can yield differences in the effectiveness of learnt representations. Here, we assess the effect of augmentation combinations by freezing the pre-trained model and only fine-tuning the prediction head during transfer learning, as described in Section 2.5. The effects of explicit augmentation are shown in Fig. 3(a), and the effects of implicit and mixed augmentations in Fig. 3(b). Here, the original input (without augmentations) is used in both branches (*i.e.* x_i, $x_i = x$), and serves as the contrastive learning baseline (white blocks in Fig. 3). As seen in Fig. 3(a), augmentation strategies directly impact the contrastive learning performance. Over the majority of the datasets, augmentations result in enhanced performance compared with the no-

---

Table 1 The average R^2 values on the unseen validation datasets with five-fold cross-validation. Seven polymer property datasets were used for predictive benchmarking: band gap (both chain (E_{gc}) and bulk (E_{gb})), electron affinity (E_{ea}), ionisation energy (E_i), DFT-calculated dielectric constant (EPS), crystallisation tendency (X_c), and refractive index (N_c). RF$_{ECFP}$, XGB$_{ECFP}$, NN$_{ECFP}$, GP$_{PG}$, NN$_{PG}$, GCN and GIN are supervised models. TransPolymer, PolyBERT and PolyCL are self-supervised models. '#Params' indicates the number of parameters. The numbers in bold indicate the best results for a given property

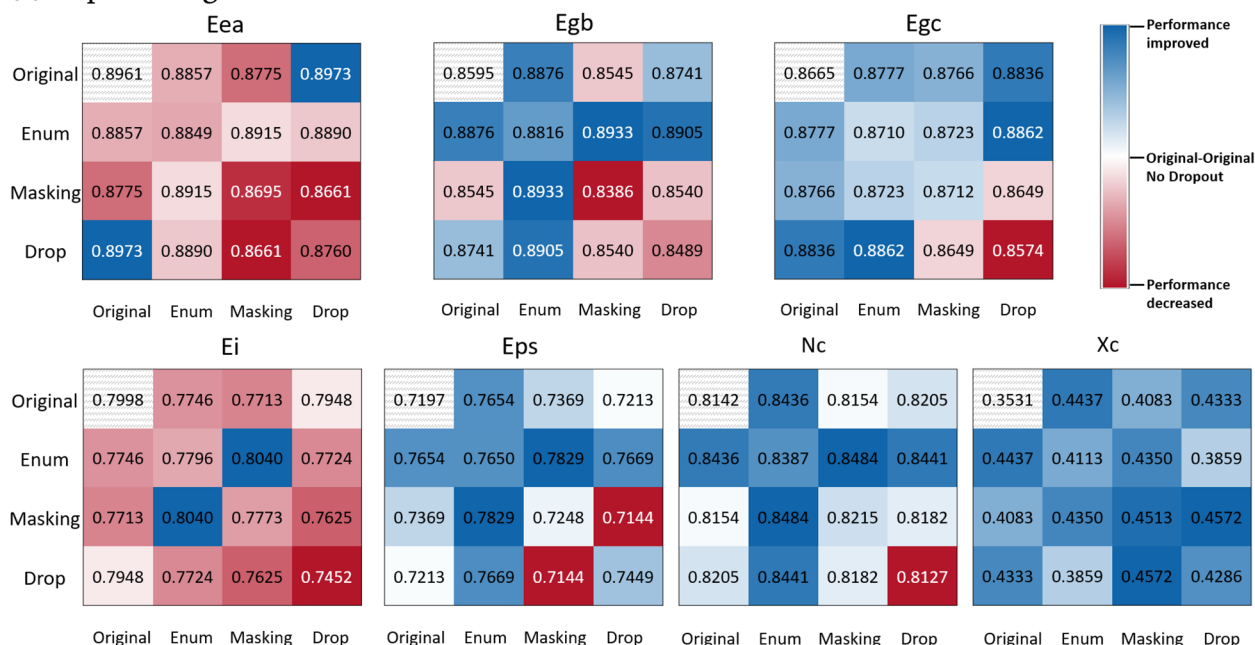| Model information		Datasets							
Model	#Params	E_{ea}	E_{gb}	E_{gc}	E_i	EPS	N_c	X_c	Avg. R^2
RF$_{ECFP}$	—	0.8401	0.8643	0.8704	0.7421	0.6840	0.7540	0.4345	0.7413
XGB$_{ECFP}$	—	0.8350	0.8568	0.8679	0.7221	0.6728	0.7574	0.3842	0.7280
NN$_{ECFP}$	264k	0.8543	0.8708	0.8838	0.7562	0.7473	0.8066	0.3975	0.7595
GP$_{PG}$[a]	—	0.90	**0.91**	**0.90**	0.77	0.68	0.79	<0	< 0.71
NN$_{PG}$[a]	—	0.87	0.90	0.89	0.74	0.71	0.78	< 0	< 0.70
GCN	70k	0.8544	0.8043	0.7988	0.6646	0.7404	0.5238	0.3316	0.6739
GIN	218k	0.8829	0.8350	0.8181	0.7841	0.6925	0.6317	0.3902	0.7192
TransPolymer[35]	82.1M	0.8943	0.8961	0.8756	0.7919	0.7568	0.8109	**0.4552**	0.7830
PolyBERT[16]	25.2M	0.9065	0.8830	0.8783	0.7670	0.7694	0.8017	0.4367	0.7775
PolyCL	25.2M	**0.9071**	0.8884	0.8832	**0.8112**	**0.7876**	**0.8460**	0.4043	**0.7897**

[a] The R^2 values of these two lines are directly taken from the single-task learning experiments of Kuenneth *et al.*[12]

Table 2 The average R^2 values on the unseen validation datasets of all downstream datasets using the enumerated form of SMILES strings with five-fold cross-validation. The numbers in bold indicate the best results for a given property

Model	$E_{ea_{Enum}}$	$E_{gb_{Enum}}$	$E_{gc_{Enum}}$	$E_{i_{Enum}}$	EPS_{Enum}	$N_{c_{Enum}}$	$X_{c_{Enum}}$	Avg. R^2
TransPolymer	0.8674	0.8593	0.8597	0.7720	0.6942	0.7788	0.3920	0.7462
polyBERT	0.8618	0.8191	0.8298	0.6884	0.7304	0.8109	0.2525	0.7133
PolyCL	**0.8870**	**0.8814**	**0.8748**	**0.7862**	**0.7405**	**0.8165**	**0.3962**	**0.7689**

(a) Explicit Augmentations

(b) Implicit and Mixed Augmentations

Augmentation Mode	Explicit Type	Implicit Type	Eea	Egb	Egc	Ei	EPS	Nc	Xc
No-Augmentation	Original-Original	No Dropout	**0.8961**	**0.8595**	**0.8665**	**0.7998**	**0.7197**	**0.8142**	**0.3531**
Implicit	Original-Original	Dropout	**0.9010**	**0.8765**	**0.8856**	**0.7999**	**0.7317**	**0.8070**	**0.2942**
Mixed	Original-Enum	Dropout	**0.9080**	**0.8902**	**0.8841**	**0.7587**	**0.7787**	**0.8338**	**0.4016**
Mixed	Enum-Masking	Dropout	**0.9071**	**0.8884**	**0.8832**	**0.8112**	**0.7876**	**0.8460**	**0.4043**
Mixed	Original-Drop	Dropout	**0.8838**	**0.8856**	**0.8892**	**0.7823**	**0.7081**	**0.8067**	**0.3958**
Mixed	Enum-Drop	Dropout	**0.8918**	**0.8838**	**0.8869**	**0.7679**	**0.7713**	**0.8364**	**0.3578**

Fig. 3 Predictive performance of transfer learning evaluated by R^2 values on downstream datasets using contrastive learning trained with different augmentation combinations. (a) Explicit augmentations only (where "Enum" refers to Enumeration) (b) Implicit and selected mixed augmentation strategy. The striped background cells are the results using the contrastive learning model pretrained with no augmentation (the baseline result). Blue blocks show improved performance relative to the baseline. Red blocks show decreased performance relative to the baseline. The intensity of the colour reflects the magnitude of the deviation.

augmentation baseline (labelled 'Original–Original'). This is especially apparent for X_c, which exhibits low baseline task performance. Here, any combination of augmentations results in a drastic increase in the quality of the representation; this is reflected by the improved performance over the baseline for all augmentation strategies. However, not all combinations of augmentations are suitable for a specific task and the best combination is task-dependent, aligning with the conclusion of previous studies.[38]

In addition, explicit augmentations yielded decreased performance relative to the baseline for the electron affinity (E_{ea}) and ionisation energy (E_i) datasets. This decline in performance likely stems from two factors: the strong baseline performance and the sensitivity of these properties to precise molecular orbital and electronic configurations. Augmentations that change or obscure parts of the structure may distort the representation that hints at the electronic properties. Hence explicit augmentations have detrimental effects, which involves making direct, observable changes to the molecule (*e.g.* removing an atom, or breaking a bond, *etc.*). Alternatively, this may also be an indication of the underlying difficulty of the downstream task. For example, X_c is a non-trivial polymer property to assess experimentally and computationally,[49] whereas methods to assess electron affinity and ionisation potential are better established. Indeed, the group contribution method used to obtain X_c may be particularly susceptible to the data augmentation strategies that we use, which increases the complexity of this task and introduces noise. The remaining downstream properties are generally correlated with broader structural motifs. The augmentations allow for augmented data to benefit model robustness by introducing slight structural perturbations.

Considering all of the downstream datasets, we observe that some explicit augmentation combinations demonstrate superior performance relative to others (Fig. 3(a)). The combinations of Original-Drop and Enumeration-Masking are the best explicit combinations, leading to improved performances compared to the no-augmentation baseline in six of the seven downstream datasets. The second best combination of augmentations are Original-Enumeration and Enumeration-Drop, which demonstrated improved performance for five of the seven downstream datasets. From the results for the augmentation combination study, we observe that including either the original or enumeration augmentation strategies improves performance. It can be intuitively explained that these two augmentations preserve the original and complete semantics of polymer molecules. This aligns with previous research, which has shown that enumeration as an augmentation strategy can enhance the performance of machine learning tasks.[35,45,59,60] During Masking and Drop, though the local data structure of polymer-SMILES is preserved, these augmentation types introduce semantic impairment. Therefore, combinations that result in superior performance preserve the full semantics in one branch; this serves as an anchor to give a hint to the parallel branch to complete its full semantics. The strategy behind these combinations might encourage the contrastive pre-training objective to learn more effective representations.

Implicit augmentations (3(b)) have a unique advantage in creating high-performing contrastive learning strategies, as it outperforms other strategies relative to the baseline for five of the seven datasets; this is comparable to the high-performing explicit augmentation combinations. After confirming the effectiveness of implicit augmentations, we combined the best-performing explicit combinations with implicit augmentations (listed as mixed augmentations in Fig. 3(b)) to identify whether this resulted in improved performance. The addition of implicit augmentations led to varying effects on the performance of explicit combinations. For Original-Enumeration and Enumeration-Masking, implicit augmentations further improved the expressiveness of the resulting representations. However, Original-Drop and Enumeration-Drop suffer from the loss of efficacy.

Surprisingly, Enumeration-Masking with implicit dropout was the overall best performing combination. This result might demonstrate that the diversified use of augmentation modes is beneficial to the construction of the contrastive learning objective. We can intuitively explain why this combination works. As analysed above, Masking of the original SMILES in one branch conceals part of the information and Enumeration in another branch assists recovery of the original SMILES from its enumerated form. In addition, the semantics in both branches are further disturbed to create slight differences by dropout noises to encourage the comparison. The entire process is comprehensive and effective. Therefore, we chose to apply this augmentation mode, which is the product of the combination of explicit and implicit augmentations, to train our final PolyCL model.

3.4 Alignment and uniformity analysis

As shown in Fig. 4, different augmentations yield different training directions in the alignment and uniformity space from the training start point (that of polyBERT). We traced the change of alignment and uniformity during the contrastive pre-training process. In the initial 20% of total epochs, alignment and uniformity loss was measured at every 2% checkpoint of total epochs. After that, alignment and uniformity loss was measured at every 20% of total epochs. In all PolyCL training processes, we observe that the changes in the alignment and uniformity loss of the first 20% epochs are faster than the remaining epochs, especially in the change of alignment. For the No Augmentation training process (use of only original molecules in both branches), pre-training leads to increased alignment loss but decreased uniformity loss. Since two polymer representations in each positive pair are identical, comparing them is ineffective; this results in the contrastive objective failing to direct the learning of underlying structures by constructing effective positive pairs. On the contrary, the application of only implicit dropout leads to improved distribution (lower uniformity loss) relative to the naïve case. However, the magnitude of the alignment loss is comparable and the increase in alignment loss is accompanied by the decrease in uniformity loss. However, the overall change in both metrics is insignificant compared with other augmentation
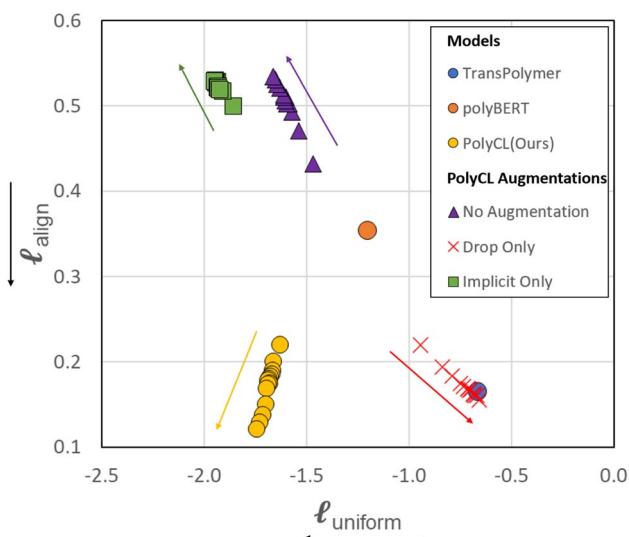
Fig. 4 Cross-model comparison on the alignment-uniformity space. For PolyBERT and Transpolymer, the alignment and uniformity of only the final published model is shown. For PolyCL and PolyCL with different augmentation combinations, the intermediate progress during contrastive pre-training is recorded and evaluated with alignment and uniformity. The coloured arrows denote the direction of change during training. The axis label arrows denote the favourable direction.

combinations, which indicates that Implicit Only may only have a slight effect on learning representations. The Drop-only case (Drop is applied to both branches) reveals decreased performance, as shown by the high uniformity loss and low alignment loss; this indicates that Drop can still recognise the similarity in feature embeddings, however it fails to capture the diversity in the data. This is further reinforced by the transfer learning results in Fig. 3(a), where Drop-only only performs well for two of the seven datasets.

Contrary to the other augmentation combinations in Fig. 4, PolyCL applies Enumeration-Masking with implicit dropout. From Fig. 4, it can be seen that the alignment and uniformity converge to the ideal quadrant ($\ell_{uniform} = -1.7431$, $\ell_{align} = 0.1209$) during the pre-training guided by the contrastive learning objective – indicating superior performance. This observation aligns with the transfer learning results in Section 3.2, and with conclusions from previous studies,[46,53] which showed that improved alignment and uniformity is generally linked to improved performance of the pre-trained representation. The low alignment loss of PolyCL specifically demonstrates that effective alignment contributes to the robust predictive results in Section 3.2 when PolyCL representations are applied to enumerated datasets.

We have also evaluated the alignment and uniformity of the pre-trained models polyBERT and Transpolymer. Poly-BERT ($\ell_{uniform} = -1.1983$, $\ell_{align} = 0.3538$) has a balanced alignment and uniformity, with both values lying in the middle region, compared to other results. For Transpolymer ($\ell_{uniform} = -0.6640$, $\ell_{align} = 0.1649$), the alignment loss is comparable to the best contrastive learning models, while the uniformity loss is similar to the Drop-only model.

While PolyCL outperforms other pre-trained models under these two evaluations, it should be noted that polyBERT is the prior of PolyCL. Therefore, the properly trained contrastive pre-training results in the improvement of the model in both alignment and uniformity. It also emphasises the importance of augmentation strategy, as not all augmentations will result in the improvement of both metrics through the training process. Though there is no evident link between the transfer learning performance on specific tasks and the alignment and uniformity, the overall transfer learning performance can be positively correlated to the alignment and uniformity matrix.

3.5 Representation space analysis

Polymers are transformed into dense and continuous representations by the pre-trained PolyCL. The representation space was evaluated by t-SNE analysis,[61] as shown in Fig. 5. t-SNE analysis arranges data so that points with similar features are plotted in close proximity to each other. Therefore, this method is well-suited to inspecting whether our pre-training method can effectively capture patterns in the learned representations. In Fig. 5(a), the unsupervised dataset was embedded in the representation space and coloured by the molecular weight of each polymer repeating unit. The results show smooth transitions between regions of low to high molecular weight. This suggests that the embedding captures the underlying structure
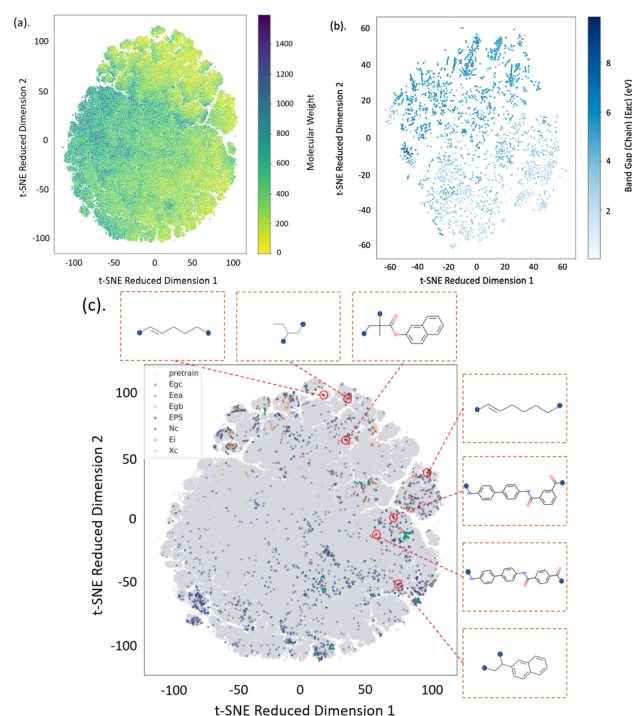

Fig. 5 t-SNE dimensional reduction analysis of the polymer representation space learnt by PolyCL. Visualisation of the continuous representation of polymer repeating units: (a) The unsupervised pre-trained dataset coloured by molecular weight; (b) The E_{gc} dataset coloured by the band gap (chain) property and (c) all available datasets coloured by the data origin, with selected polymers shown. The blue dot denotes the connection point of the repeating unit to the polymer chain.

and size of different polymers that correlates closely with their molecular weight differences.

In Fig. 5(b), polymers from a sampled downstream property dataset (E_{gc}) were embedded in the representation space coloured by the value of the band gap (chain) ground truth; the gradient of this representation shows that the sampled downstream property, the chain band gap (E_{gc}), is highly related to the embedded structural features of polymers. In other property datasets, this gradient was also observed (as shown in Fig. S2†). Due to the limited number of datapoints in each remaining dataset, the gradient is less evident than the TSNE visualisation of E_{gc} dataset (shown in Fig. 5(b)).

Our results also suggest that the representation space effectively captures changes in key physical properties implied by the structural features that the original t-SNE was trained on. In Fig. 5(c), all available data is encoded to a representation space and colored by the data source. Here, we observe that the initial, unsupervised dataset comprehensively covers the chemical space encompassed by all of the downstream datasets. We also visualised the molecular structures corresponding to randomly selected points in the embedding. The results show that the structural features learnt by contrastive learning align with human understanding, yet slight divergence. It can be seen from the visualisation that neighbouring representations do not necessarily have similar structures in their molecular graphs. This discrepancy may be due to the different emphasis of SMILES strings and molecular graphs on encoding molecular structures and the special focus of contrastive learning strategies to learn the representations.

3.6 Future work

We have introduced the use of contrastive learning to obtain high-quality representations. Here, we used a basic polymer representation to capture the core structural information of polymers and allow direct comparison to previous methods. However, incorporating more refined input features that encode the macroscopic polymer information could potentially enhance the predictive performance of polymer representations. Furthermore, the inclusion of polymer types may hint at the high-level structural variations that affect the behavior of polymers. In future work, the joint use of contrastive learning and a polymer representation that considers the complexities in the macromolecular level should be explored. This approach would maximise the potential of both representation learning techniques and polymer chemistry. In addition, noisy, missing data and the variability of polymer representations can hurt the application of algorithms. Future work can also leverage machine learning to improve the adaptability of algorithm to better handle noise, incomplete data and a broader range of polymer types.

4 Conclusion

We present a self-supervised pre-training paradigm, PolyCL, that uses contrastive learning to achieve effective polymer representation learning using unsupervised data. We have comprehensively explored varying explicit and implicit augmentation modes and found that the inclusion of both types of augmentations can result in high-performing contrastive learning. Our analysis suggests that the PolyCL-learnt representation excels in preserving chemical information and enhancing model generalisability – as shown by its superior performance in transfer learning objectives across all seven chemical properties including band gap (both chain (E_{gc}) and bulk (E_{gb})), electron affinity (E_{ea}), ionisation energy (E_i), DFT-calculated dielectric constant (EPS), crystallisation tendency (X_c), and refractive index (N_c). Additionally, PolyCL demonstrates enhanced accuracy in chemical property prediction and stability across diverse datasets, while also improving representation robustness to variable and non-standardized datasets. PolyCL produces high-quality machine-learnt representations, which we expect will be beneficial for a wide range of downstream property-prediction tasks for polymer informatics. The dataset and model are available at: **http://github.com/JiajunZhou96/PolyCL**.

Data availability

Data, model and scripts for producing results described in this paper are available publicly at **https://github.com/JiajunZhou96/PolyCL**.

Author contributions

J. Z. developed the PolyCL models and analysed the results. J. Z and Y. Y performed the calculations. A. M. assisted in project design and execution. K. E. J. supervised the project. J. Z. wrote the manuscript and all authors contributed to the final version.

Conflicts of interest

There are no conflicts of interest to declare.

Acknowledgements

Notes and references

1 W. Sha, Y. Li, S. Tang, J. Tian, Y. Zhao, Y. Guo, W. Zhang, X. Zhang, S. Lu, Y.-C. Cao, et al., *InfoMat*, 2021, **3**, 353–361.

2 M. F. Maitz, *Biosurf. Biotribol.*, 2015, **1**, 161–176.

3 F. Puoci, F. Iemma, U. G. Spizzirri, G. Cirillo, M. Curcio, N. Picci, Am. J. Agric, et al., *Biol. Sci.*, 2008, **3**, 299–314.

4 G. Li, R. Zhu and Y. Yang, *Nat. Photonics*, 2012, **6**, 153–161.

5 M. Jaiswal and R. Menon, *Polym. Int.*, 2006, **55**, 1371–1384.

6 K. Matyjaszewski, *Prog. Polym. Sci.*, 2005, **30**, 858–875.

7 K. Sada, *Polym. J.*, 2018, **50**, 285–299.

8 W. H. Binder and R. Sachsenhofer, *Macromol. Rapid Commun.*, 2007, **28**, 15–54.

9 L. Chen, G. Pilania, R. Batra, T. D. Huan, C. Kim, C. Kuenneth and R. Ramprasad, *Mater. Sci. Eng., R*, 2021, **144**, 100595.

10 A. Mannodi-Kanakkithodi, G. Pilania, T. D. Huan, T. Lookman and R. Ramprasad, *Sci. Rep.*, 2016, **6**, 1–10.

11 T. B. Martin and D. J. Audus, *ACS Polym. Au*, 2023, **3**, 239–258.

12 C. Kuenneth, A. C. Rajan, H. Tran, L. Chen, C. Kim and R. Ramprasad, *Patterns*, 2021, **2**, 100238.

13 M. Zeng, J. N. Kumar, Z. Zeng, R. Savitha, V. R. Chandrasekhar and K. Hippalgaonkar, *arXiv*, 2018, preprint, arXiv:1811.06231, DOI: **10.48550/arXiv.1811.06231**.

14 T.-S. Lin, C. W. Coley, H. Mochigase, H. K. Beech, W. Wang, Z. Wang, E. Woods, S. L. Craig, J. A. Johnson, J. A. Kalow, et al., *ACS Cent. Sci.*, 2019, **5**, 1523–1531.

15 H. Doan Tran, C. Kim, L. Chen, A. Chandrasekaran, R. Batra, S. Venkatram, D. Kamal, J. P. Lightstone, R. Gurnani, P. Shetty, et al., *J. Appl. Phys.*, 2020, **128**, 171104.

16 C. Kuenneth and R. Ramprasad, *Nat. Commun.*, 2023, **14**, 4099.

17 L. Tao, G. Chen and Y. Li, *Patterns*, 2021, **2**, 100225.

18 M.-X. Zhu, H.-G. Song, Q.-C. Yu, J.-M. Chen and H.-Y. Zhang, *Int. J. Heat Mass Transfer*, 2020, **162**, 120381.

19 S. Wu, Y. Kondo, M.-a. Kakimoto, B. Yang, H. Yamada, I. Kuwajima, G. Lambard, K. Hongo, Y. Xu, J. Shiomi, et al., *npj Comput. Mater.*, 2019, **5**, 66.

20 J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li and M. Sun, *AI Open*, 2020, **1**, 57–81.

21 O. Queen, G. A. McCarver, S. Thatigotla, B. P. Abolins, C. L. Brown, V. Maroulas and K. D. Vogiatzis, *npj Comput. Mater.*, 2023, **9**, 90.

22 J. Park, Y. Shim, F. Lee, A. Rammohan, S. Goyal, M. Shim, C. Jeong and D. S. Kim, *ACS Polym. Au*, 2022, **2**, 213–222.

23 D. Weininger, *J. Chem. Inf. Comput. Sci.*, 1988, **28**, 31–36.

24 G. Chen, L. Tao and Y. Li, *Polymers*, 2021, **13**, 1898.

25 B. K. Phan, K.-H. Shen, R. Gurnani, H. Tran, R. Lively and R. Ramprasad, *npj Comput. Mater.*, 2024, **10**, 186.

26 M. Aldeghi and C. W. Coley, *Chem. Sci.*, 2022, **13**, 10486–10498.

27 S. Jiang and M. Webb, *ChemRxiv*, 2024, preprint, DOI: **10.26434/chemrxiv–2024–ld2k6**.

28 R. Irwin, S. Dimitriadis, J. He and E. J. Bjerrum, *Mach. learn.sci. technol.*, 2022, **3**, 015022.

29 Y. Cui, Y. Song, C. Sun, A. Howard and S. Belongie, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4109–4118.

30 K. He, R. Girshick and P. Dollár, *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4918–4927.

31 A. Tendle and M. R. Hasan, *Mach. Learn. Appl.*, 2021, **6**, 100124.

32 F.-Y. Sun, J. Hoffmann, V. Verma and J. Tang, *arXiv*, 2019, preprint, arXiv:1908.01000, DOI: **10.48550/arXiv.1908.01000**.

33 Y. Wang, J. Wang, Z. Cao and A. Barati Farimani, *Nat. Mach. Intell.*, 2022, **4**, 279–287.

34 R. Balestriero, M. Ibrahim, V. Sobal, A. Morcos, S. Shekhar, T. Goldstein, F. Bordes, A. Bardes, G. Mialon, Y. Tian et al., *arXiv*, 2023, preprint, arXiv:2304.12210.

35 C. Xu, Y. Wang and A. Barati Farimani, *npj Comput. Mater.*, 2023, **9**, 64.

36 A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser and I. Polosukhin, *Adv. Neural Inf. Process. Syst.*, 2017, **30**, **https://proceedings.neurips.cc/paper_files/paper/2017**.

37 J. Devlin, M.-W. Chang, K. Lee and K. Toutanova, *arXiv*, 2018, preprint, arXiv:1810.04805, DOI: **10.48550/arXiv.1810.04805**.

38 Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid and P. Isola, *Adv. Neural Inf. Process. Syst.*, 2020, **33**, 6827–6839.

39 C. Yang, Z. An, L. Cai and Y. Xu, *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022, pp. 3045–3053.

40 Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang and Y. Shen, *Adv. Neural Inf. Process. Syst.*, 2020, **33**, 5812–5823.

41 Y. Yin, Q. Wang, S. Huang, H. Xiong and X. Zhang, *Proceedings of the AAAI conference on artificial intelligence*, 2022, pp. 8892–8900.

42 Z. Cao, R. Magar, Y. Wang and A. Barati Farimani, *J. Am. Chem. Soc.*, 2023, **145**, 2958–2967.

43 S. Liu, W. Nie, C. Wang, J. Lu, Z. Qiao, L. Liu, J. Tang, C. Xiao and A. Anandkumar, *Nat. Mach. Intell.*, 2023, **5**, 1447–1457.

44 Z. Guo, P. Sharma, A. Martinez, L. Du and R. Abraham, *arXiv*, 2021, preprint, arXiv:2109.08830, DOI: **10.48550/arXiv.2109.08830**.

45 G. A. Pinheiro, J. L. Da Silva and M. G. Quiles, *J. Chem. Inf. Model.*, 2022, **62**, 3948–3960.

46 T. Chen, S. Kornblith, M. Norouzi and G. Hinton, *International Conference on Machine Learning*, 2020, pp. 1597–1607.

47 T. Gao, X. Yao and D. Chen, *arXiv*, 2021, preprint, arXiv:2104.08821, DOI: **10.48550/arXiv.2104.08821**.

48 J. Xia, L. Wu, J. Chen, B. Hu and S. Z. Li, *Proceedings of the ACM Web Conference 2022*, 2022, pp. 1070–1079.

49 S. Venkatram, R. Batra, L. Chen, C. Kim, M. Shelton and R. Ramprasad, *J. Phys. Chem. B*, 2020, **124**, 6046–6054.

50 A. Rudin and P. Choi, *The elements of polymer science and engineering*, Academic press, 2012.

51 P. He, X. Liu, J. Gao and W. Chen, *arXiv*, 2020, preprint, arXiv:2006.03654, DOI: **10.48550/arXiv.2006.03654**.

52 I. Loshchilov and F. Hutter, *arXiv*, 2017, preprint, arXiv:1711.05101, DOI: **10.48550/arXiv.1711.05101**.

53 T. Wang and P. Isola, *International Conference on Machine Learning*, 2020, pp. 9929–9939.

54 N. Reimers and I. Gurevych, *arXiv*, 2019, preprint, arXiv:1908.10084, DOI: **10.48550/arXiv.1908.10084**.

55 R. Hadsell, S. Chopra and Y. LeCun, *2006 IEEE Computer Society Conference on Computer vision and Pattern Recognition*, 2006, pp. 1735–1742.

56 D. Rogers and M. Hahn, *J. Chem. Inf. Model.*, 2010, **50**, 742–754.

57 T. N. Kipf and M. Welling, *arXiv*, 2016, preprint, arXiv:1609.02907, DOI: **10.48550/arXiv.1609.02907**.

58 K. Xu, W. Hu, J. Leskovec and S. Jegelka, *arXiv*, 2018, preprint, arXiv:1810.00826, DOI: **10.48550/arXiv.1810.00826**.

59 E. J. Bjerrum, *arXiv*, 2017, preprint, arXiv:1703.07076, DOI: **10.48550/arXiv.1703.07076**.

60 H. Qiu, L. Liu, X. Qiu, X. Dai, X. Ji and Z.-Y. Sun, *Chem. Sci.*, 2024, **15**, 534–544.

61 L. Van der Maaten and G. Hinton, *J. Mach. Learn. Res.*, 2008, **9**, 2579–2605.