MSDE

PAPER

View Article Online
View Journal | View Issue

Cite this: *Mol. Syst. Des. Eng.*, 2022, 7, 661

Featurization strategies for polymer sequence or composition design by machine learning†

Roshan A. Patel, Carlos H. Borca and Michael A. Webb *

Received 3rd November 2021,
Accepted 10th March 2022

DOI: 10.1039/d1me00160d

rsc.li/molecular-engineering

The emergence of data-intensive scientific discovery and machine learning has dramatically changed the way in which scientists and engineers approach materials design. Nevertheless, for designing macromolecules or polymers, one limitation is the lack of appropriate methods or standards for converting systems into chemically informed, machine-readable representations. This featurization process is critical to building predictive models that can guide polymer discovery. Although standard molecular featurization techniques have been deployed on homopolymers, such approaches capture neither the multiscale nature nor topological complexity of copolymers, and they have limited application to systems that cannot be characterized by a single repeat unit. Herein, we present, evaluate, and analyze a series of featurization strategies suitable for copolymer systems. These strategies are systematically examined in diverse prediction tasks sourced from four distinct datasets that enable understanding of how featurization can impact copolymer property prediction. Based on this comparative analysis, we suggest directly encoding polymer size in polymer representations when possible, adopting topological descriptors or convolutional neural networks when the precise polymer sequence is known, and using chemically informed unit representations when developing extrapolative models. These results provide guidance and future directions regarding polymer featurization for copolymer design by machine learning.

Design, System, Application

Machine learning and artificial intelligence are revolutionizing paradigms for materials design, providing powerful and efficient tools to model materials properties and accelerate discovery. Supplying informative, numerical representations of target systems—a process known as featurization—is critical to usefully deploying machine learning in this context. Although there are myriad featurization methods for small molecules, identifying suitable methods of representation and understanding their limitations is less developed for polymers, particularly systems with more than one constitutional unit. Herein we present, explore, and evaluate the efficacy of multiple featurization strategies for copolymers in several distinct prediction tasks. By systematic controlled comparisons over multiple datasets, we identify elements of copolymer featurization strategies that result in predictive machine learning models, which is key to successful surrogate modeling by machine learning for design. Overall, this work provides examples of multiple copolymer featurization strategies, baseline expectations for performance, and general guidance that can be leveraged in future copolymer design campaigns.

1 Introduction

Polymers are ubiquitous and versatile materials that can facilitate a wide range of complex tasks in biology, industry, and beyond.[1–4] However, the expansive chemical, sequence, and topological space that facilitates such diverse applications can obfuscate the design of next-generation, fit-for-purpose polymeric materials.[5–9] For example, using a limited set of just three different monomer types, there are on the order of 10^{47} distinct copolymers that can be generated with degree of polymerization between 10 and 100. Thus, while theory and modeling are invaluable for understanding the origins of observed phenomena and informing the design of specific, well-defined polymer systems,[10–17] intricate studies may severely limit exposure to unknown but promising regions of design space.[18] In addition, resource limitations (time, monetary, or computational) likely preclude exhaustive characterization of combinatorial search spaces.[19]

Over the last two decades, artificial intelligence has emerged as a useful tool for accelerating materials design by (i) facilitating accurate surrogate modeling of quantitative structure–property relationships (QSPRs) and (ii) providing more efficient ways to explore chemical space.[20–27]

Department of Chemical and Biological Engineering, Princeton University, Princeton, NJ, USA. E-mail: mawebb@princeton.edu

† Electronic supplementary information (ESI) available: Simulation details and calculated properties for dataset A, a description of model architectures and hyperparameters, metadata for all machine learning models, IDP sequences and property labels forming dataset A, and a complete list of model performance metrics and machine learning models. See DOI: 10.1039/d1me00160d

Supervised machine learning (ML) models can be trained to cheaply estimate properties of materials from known examples; when used for screening or coupled to robust optimization algorithms,[28] such models can aid in efficiently identifying promising candidate materials. While flourishing in the domain of "hard" materials and small molecules, applications of ML to polymer design have been relatively limited by comparison for a number of practical and technical reasons.[16,29–34] For example, there are numerous large, open-access databases for small molecules and ordered materials, but data availability and accessibility remains a major challenge for polymer ML.[29,35,36] Presently, this challenge is overcome by either (i) laborious, brute-force data sourcing and curation or (ii) in-house data generation. The former approach has been largely useful for polymer informatics in the space of homopolymers,[37–40] while the latter has been typically necessary to design systems with sequence[41–44] or compositional control[45–48] over multiple monomers or constitutional units[49] (CUs). In the near term, advancements in automated polymer synthesis[50] and in hierarchical polymer simulation,[33] coupled with efficient data acquisition schemes,[34] are likely to substantially enhance capabilities to generate requisite data for training ML models on-the-fly. With evident activity to facilitate acquisition of suitable polymer data, a fundamental consideration that follows is how to represent polymer data to ML algorithms.

In the context of ML-guided design, the method of featurization or representation, *i.e.*, how a molecule or system is converted into a numerical input, is a fundamental consideration that not only dictates what information is available for constructing QSPRs but also what ML algorithms are suitable for the QSPR task.[34] In general, featurization can profoundly impact what patterns are extracted and exploited by ML algorithms,[51–53] which can subsequently affect how much data and time is required to train accurate models. Because featurization also defines the mapping of system chemistry to a vector space, it has clear implications on the span of possible solutions for a given optimization task. Consequently, the development and investigation of machine-readable representations for property prediction is of significant interest. Although there are numerous viable strategies to facilitate ML on small molecules and ordered materials,[19,54–65] there is comparatively little guidance regarding how to effectively featurize polymers for ML.

The featurization of polymers has been mostly dictated by the source of training data and the scope of intended design space. One strategy that has enjoyed considerable success is to simply adapt existing molecular featurization strategies to describe constitutional repeat units[49] (CRUs) of the polymer; this approach has been useful for designing homopolymers.[39,66,67] However, using only the CRU to define QSPR neglects potential hierarchical and/or topological complexity that may inform property prediction tasks. To partially address this limitation, Ramprasad and coworkers have described hierarchical polymer fingerprints that combine atomic-level connectivity descriptors, larger lengthscale property descriptors, and morphological descriptors;[68–70] this approach has also been recently extended to describe stochastic binary copolymers.[47] Constructing a low-dimensional latent space embedding of higher-dimensional feature vectors using variational autoencoders[71] (VAEs) is another attractive and complementary approach to aforementioned techniques. This has been recently exemplified by Shmilovich *et al.* to describe the chemical space spanned by coarse-grained tripeptides for the purpose of identifying peptides with specific self-assembly behavior.[41] Batra *et al.* have also demonstrated the use of VAEs to translate a modified, polymer-based SMILES grammar into a suitable vector space for constructing Gaussian process regression models to predict glass-transition temperatures and bandgaps of homopolymers.[67]

Featurization for sequence-defined polymer systems can be pursued in several ways. For example, feature extraction architectures may be used to learn relevant sequence and topological correlations during supervised ML. In this vein, Webb *et al.* built ML models that leveraged recurrent and convolutional neural network (CNN) architectures to predict and later design the radii of gyration for CG polymers by simply manipulating sequence.[43] Mohapatra *et al.* similarly combined Morgan fingerprints (a molecular featurization strategy) with CNNs to optimize fast-flow peptide synthesis.[44] The use of graph neural network architectures[71] to represent macromolecular chemistry is also at early stages of exploration.[72] As an alternative to using feature extraction architectures, Jablonka *et al.* generated a hand-crafted vector of descriptors, which contained descriptions of composition, sequence entropy, and sub-sequence clusters, to guide the *in silico* design of coarse-grained (CG) polymer dispersants.[73] While these developments are generally promising, it remains unclear under what circumstances and to what extent any given polymer featurization strategy outperforms another.

We introduce a series of relatively simple featurization strategies for copolymers and evaluate their performance in supervised learning regression tasks derived from four distinct datasets. Following the introduction of the datasets and featurization approaches, we critically examine the role of polymer size, the expression and manner of sequence representation, and the impact of using chemically informed CU descriptions in different prediction scenarios. Through this comparative study, we identify key attributes amongst successful strategies that can serve as guidance for future ML-guided copolymer design problems.

2 Methodology

2.1 Datasets

To evaluate the efficacy of potential polymer featurization strategies, we consider their performance in several, distinct supervised regression tasks. These tasks are defined in the

context of four datasets, which will be referenced as datasets A, B, C, and D. Dataset A is introduced in the present paper and comprises properties obtained by CG simulation for a set of intrinsically disordered proteins (IDPs). The remaining three datasets are obtained from literature sources (dataset B from ref. 43, dataset C from ref. 48, and dataset D from ref. 74); these datasets feature different property labels, design spaces, and CU metadata.

2.1.1 Dataset A: coarse-grained IDPs. Dataset A contains simulation-derived properties for 2585 IDPs. The CUs for IDPs correspond to the various amino acids, but their disordered sequences precludes definition of a single CRU for each sequence. The IDPs are thus fairly described as linear, stochastic polymers with known sequence. The IDPs within dataset A have a degree of polymerization, denoted as N, between 20 and 600 CUs (amino acids). The specific sequences were sourced from version 9.0 of the DisProt database;[75,76] upon initial acquisition, care was taken to eliminate any duplicate sequences and ensure that all 2585 IDPs were unique.

Properties of the IDPs under infinite-dilution conditions (*i.e.*, single-chain properties) were computed at 300 K *via* molecular dynamics (MD) with the LAMMPS simulation package.[77] All IDPs were modeled using the improved hydropathy scale (HPS) CG model from Regy *et al.*[78] Specific properties extracted for use as labels in regression tasks include the radius of gyration R_g, the heat capacity C_v, and the end-to-end decorrelation time τ_N. Dataset A is provided in the ESI† as well as additional details regarding the simulations and calculations.

2.1.2 Dataset B: monodisperse coarse-grained polymers. Dataset B is sourced from ref. 43, which used ML and Bayesian optimization to direct the design of sequence-defined polymers with target mean-square radius of gyration $\langle R_g^2 \rangle$. The dataset contains 1540 regular copolymers (*i.e.*, they have a well-defined CRU) and 200 stochastic copolymers; for each copolymer, the label is $\langle R_g^2 \rangle$ obtained from CG simulation. The copolymers contain up to four distinct CUs from ten possible CUs, and each CU features one of two types of backbone beads and up to two pendant beads, also of two possible types; all copolymers have N = 400 CUs. Unless otherwise noted, all performance metrics and models are derived only from the regular copolymers of dataset B.

There are several notable differences between datasets A and B. First, dataset B features CG polymers that are monodisperse. Second, dataset B contains fewer total possible CUs than dataset A, and the number of unique CUs in any given polymer is restricted to a subset of that total in dataset B but not in dataset A. Finally, the CG polymers in dataset B are not necessarily linear, although the side-chains are small. Like dataset A, the data originates from MD simulation, such that the sequences and simulation metadata are precisely known.

2.1.3 Dataset C: experimental methacrylate copolymers. Dataset C is sourced from ref. 48, which uses a computer-guided materials discovery approach to design statistical copolymers of methacrylates to serve as high contrast ^{19}F MRI agents.[48] There are six possible CUs that can be combined in varying proportions and degrees of polymerization, but the polymer sequences are unknown. Of the 397 unique copolymers reported in the study, we use 271 copolymers that were labeled with the signal-to-noise ratio (SNR) from NMR experiments; the SNR is always treated as the target output for our regression task. In addition, the dataset describes the fractions of incorporation for each possible methacrylate, the mean number-averaged molecular weight of the polymers, and the polydispersity. Like dataset A and in contrast to dataset B, the polymethacrylates are linear copolymers. In contrast to all other datasets discussed, the data is experimentally obtained. This dataset is also smallest in size.

2.1.4 Dataset D: linear bipolymers with patterned surfaces. Dataset D is sourced from ref. 74, which trains support vector regression (SVR) models to predict the adhesion free energy of CG copolymers on patterned surfaces as a function of polymer sequence; a separate SVR model is developed for each of four surfaces. The copolymers studied are comprised of up to two distinct CUs and have N = 20. Considering all four surfaces, dataset D contains 80 000 data points with known polymer sequence labeled with an adhesion free energy ΔF_{ad} for a given surface. Compared to datasets A and B, which are also generated by CG MD simulation, the copolymers in dataset D are shorter and have fewer unique CUs. However, ΔF_{ad} is comparatively more complex than the single-chain properties reported in datasets A and B.

Rather than training separate ML models for each of the surfaces present in dataset D, we pursue a different approach that additionally uses the surface as an input feature. To encode the identity of the surface for which the ΔF_{ad} label is computed, all polymer feature vectors are appended with a four-dimensional one-hot vector prior to being passed to densely connected neural network layers. For explicit-sequence featurization strategies (section 2.3.1), representations of the polymer are first processed with feature-extraction architectures prior to concatenation with the one-hot encoding vector that indicates the surface.

2.2 Overview of featurization strategies

Fig. 1 illustrates the origins and relationships amongst the various polymer featurization strategies explored in this paper. Common to all strategies is the essential characterization of a polymer as a set of bonded or topologically connected CUs (Fig. 1, left top and middle); the CUs can be numerically described *via* a vector that distinguishes its chemical characteristics from other CUs *via* what is colloquially referred to as a "fingerprint" (Fig. 1, left bottom). Across datasets A–D, the CUs are respectively amino acids, sets of CG polymer beads, methacrylate monomers of differing chemistry, and CG beads; the specific fingerprints employed for these CUs are described in section 2.2.1. From this starting point, we explore two broad paradigms of

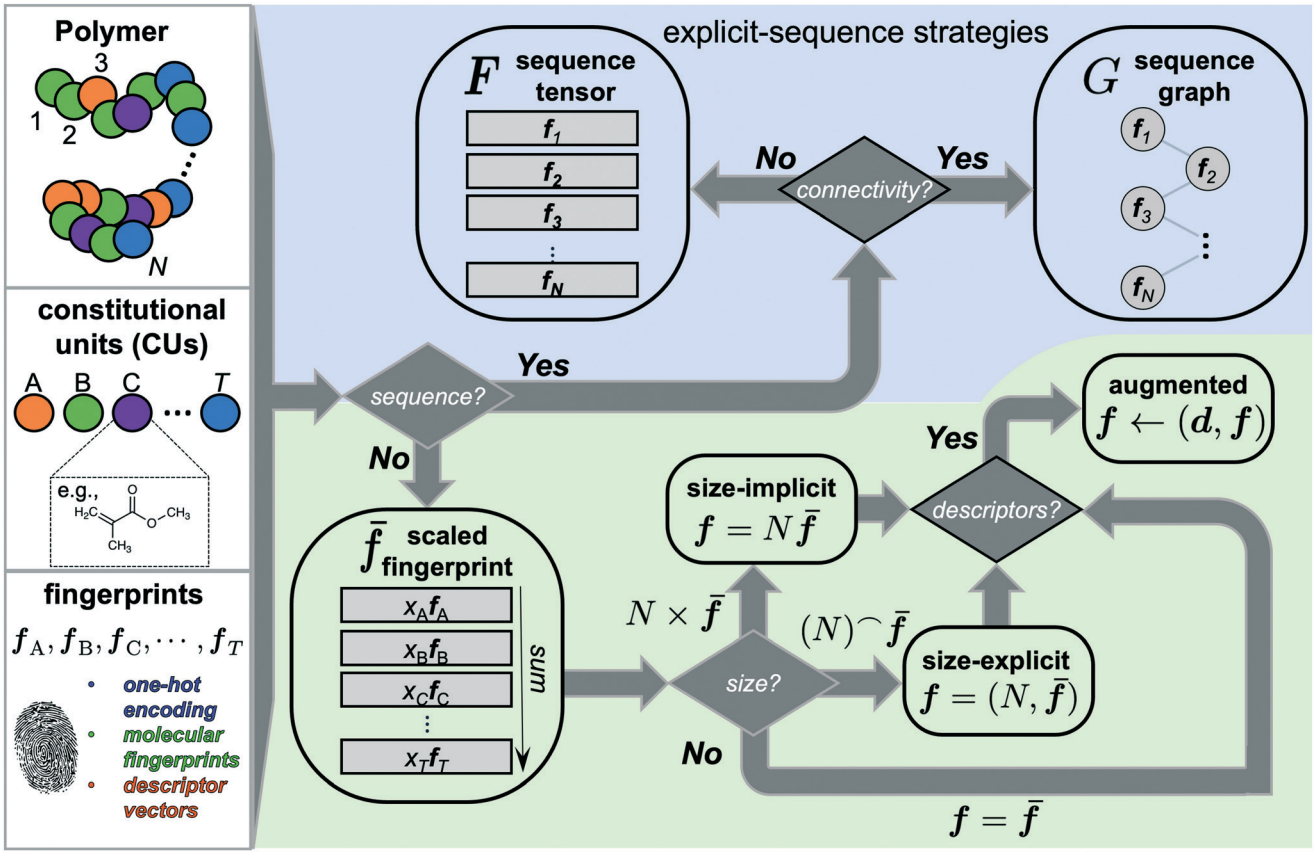

Fig. 1 Diagrammatic depiction of the relationships amongst various polymer featurization strategies used in this study. All strategies have a common conceptual starting point of a polymer being a set of N topologically connected constitutional units (CUs); the constitutional units are assigned types A, B, C, …, T, depending on their chemistry. The chemistry of the various CUs can be numerically represented *via* fingerprints, denoted as f_K for a CU of type K. We consider featurization strategies that either explicitly represent the polymer sequence (blue) or those that do not (green). In the figure, quantities subscripted with alphabetic characters are associated with CU types, quantities subscripted with arabic numerals are associated with indexed CUs within the polymer, and quantities with no subscript are associated with the polymer.

featurization strategies: those that explicitly represent sequence information (Fig. 1, top in blue) and those that do not (Fig. 1, bottom in green). While the latter may be considered for most prediction/design tasks, the former may not be viable, depending on data source or synthetic limitations.

2.2.1 Fingerprints

2.2.1.1 One-hot encoding. We view one-hot encoding (OHE), which is commonly used to represent categorical variables, as the simplest of chemical fingerprints. In this approach, CU fingerprints are N_T-dimensional vectors where N_T is the number of distinct CUs in the dataset. For notational convenience, we will assume here and in subsequent sections that CUs of type A, B, C, …, T are numerically indexed by 1, 2, 3, …, N_T. The elements of the OHE fingerprint for a CU of type K are thus given by

$$f_K[i] = \delta_{ki}, \text{ for } i = 1, \ldots, k, \ldots, N_T \quad (1)$$

where k is the numerical index for the CU of type K, $f_k[i]$ provides the value of the fingerprint in the ith dimension, and δ_{ki} is the Kronecker delta. The result is that the kth element of f_k is equal to one, and all remaining are equal to zero. Therefore, the dimensionality of OHE fingerprints are 20, 10, 6, and 2 for datasets A, B, C, and D, respectively. Notably, the OHE fingerprint simply identifies CUs and does not express chemical similarity. Within this representation, one may view the different CUs as being orthogonal in chemical space.

2.2.1.2 Molecular fingerprints. For datasets A and C, we also make use of conventional molecular fingerprinting techniques as applied to each of the various CUs. In particular, we use RDKit[79] to obtain Morgan fingerprints for each CU.[54] The Morgan fingerprint, like other extended-connectivity fingerprints,[55] generally denote the presence or absence of chemical substructures. The uniqueness and information content of the Morgan fingerprint depends on both the vector dimensionality as well as the radius of the substructure search. We find that the mean pairwise geometric similarities amongst CUs approximately plateaus at 2048 dimensions and 4 Å for dataset A and 2048 dimensions and 5 Å for dataset C. Therefore, we choose these as the hyperparameters for CU fingerprint generation. Following generation of fingerprints for all CUs in a given

dataset, we remove dimensions that possess only zeros or only ones. This yields a final dimensionality of 152 and 66 for the Morgan fingerprints used for dataset A and C, respectively. This approach is not used for datasets B and D as there are no underlying chemical structures to represent the CUs.

2.2.1.3 Descriptor vectors. Describing molecules or systems using a vector of physiochemical descriptors is another common strategy in molecular featurization when constructing QSPR. We adopt a similar strategy here as applied to CUs.

For *in silico*-derived datasets (datasets A, B, and D), we use simulation metadata by formulating vectors of force-field parameters that are specific to each CU. Because the force-field parameters express information such as the CU size or its interaction with other moieties, they are somewhat similar to common descriptors like accessible surface area, partitioning coefficients, or properties derived from quantum chemical calculations. The descriptor vector for the kth CU formed from simulation metadata is given by

$$f_K = \begin{cases} \left(m_K, q_K, \sigma_{k,1}, \ldots \sigma_{k,n}, \lambda_{k,1} \ldots, \lambda_{k,n}\right) & \text{for Dataset A} \\ \text{flatten}\begin{Bmatrix} \left(\varepsilon_{k_0,1} \ldots, \varepsilon_{k_0,4}\right) \\ \left(\varepsilon_{k_1,1} \ldots, \varepsilon_{k_1,4}\right) \\ \left(\varepsilon_{k_2,1} \ldots, \varepsilon_{k_2,4}\right) \\ \left(\sigma_{k_0,1} \ldots, \sigma_{k_0,4}\right) \\ \left(\sigma_{k_1,1} \ldots, \sigma_{k_1,4}\right) \\ \left(\sigma_{k_2,1} \ldots, \sigma_{k_2,4}\right) \end{Bmatrix} & \text{for Dataset B.} \\ \left(\varepsilon_{k,1}, \varepsilon_{k,2}, r_{k,1}, r_{k,2}\right) & \text{for Dataset D} \end{cases} \quad (2)$$

For dataset A, m_K is the mass of the kth CU, q_K is its charge, $\sigma_{k,i}$ and $\lambda_{k,i}$ respectively represent the pairwise Lennard-Jones interaction diameter and strength of hydrophobic interactions between the kth and ith CUs; in the HPS model,[78] arithmetic means are used to define cross interactions. For dataset B, $\varepsilon_{k_j,i}$ and $\sigma_{k_j,i}$ are the energy minimum and diameter for the interaction between the CG bead in position j of the kth CU and bead type i; there are four dimension in each row to account for the four distinct CG bead types that make up the ten possible CUs. Here, j is 0 for the backbone position, 1 for the first pendant position, and 2 for the second pendant position. For CUs that do not feature CG beads in one or both of the pendant positions, the entries are zero. In ref. 43, Lorentz–Berthelot combination rules define cross interactions. For dataset D, $\varepsilon_{k,i}$ is the minimum pairwise interaction energy between the kth and ith CUs and $r_{k,i}$ is the cutoff distance for their interaction. Cross interactions are defined as specified in ref. 74. In all cases, properties that do not vary amongst CUs (*e.g.*, the bead size for dataset B and D) are excluded from f_K as they would represent constants to the ML algorithm, but they could be included if required for extensibility. Lower-dimensional forms of the descriptor vectors in eqn (2) that exclude cross interactions are also considered.

While datasets B and D stem from properties of phenomenological CG polymers of no specific chemistry, the polymers in datasets A and C have CUs with underlying chemical structures. Consequently, we also consider descriptor vectors of nearly 1600 descriptors derived using the Mordred python package.[58] For a given set of CUs, we remove any descriptors with zero variance. We also remove descriptors that exhibit significant correlation with other descriptors in stepwise fashion. Specifically, we compute the number of instances for which a descriptor exhibits a Pearson correlation coefficient >0.85 with the set of all current descriptors, and then we remove the descriptor with the greatest number of instances and repeat until all descriptors possess pairwise Pearson correlation coefficients less than 0.85. Although this process is not guaranteed to retain the maximum number of uncorrelated features, it is a reasonable approximation to the NP-hard problem of vertex cover. This process yields a 257-dimensional descriptor vector for dataset A and a 47-dimensional descriptor vector for dataset C for use as CU fingerprints.

2.3 Featurization paradigms

We consider featurization strategies that both explicitly represent polymer sequences as well as those that rely more on composition-based or "scaled" representations. The different approaches are shown in Fig. 1. In all cases, property predictions are ultimately made based on the output of a densely-connected deep neural network (DNN), and predictions are made only for global polymer properties. The polymer representations do not utilize or depend on the coordinates of the CUs or their relative distances with respect to other CUs in the polymer. For datasets A, B and D, we have sequence information and expect the "reversed" sequences to have identical properties as the forward sequence; accordingly, predictions from the DNN should ideally be invariant to sequence inversion. However, invariance to sequence inversion is specific to the coarse-grained representations of our polymers and does not universally hold. For example, the asymmetry of amino acids or nucleotides as CUs imparts directionality to the sequence, such that the forward and reverse sequences do not have the same bonding connectivity or geometric structure; therefore, those sequences are not expected to exhibit the same global polymer properties. Property invariance to sequence inversion may be handled during construction of the feature vectors themselves, enforced by use of specific ML algorithms and architectures, or approximately addressed *via* data augmentation.

2.3.1 Explicit sequence representation

2.3.1.1 Sequence graph. The sequence graph featurization approach explicitly represents the polymer sequence and connectivity amongst CUs. Specifically, the polymer is represented as a graph $G = (V, E)$. V is a set nodes that contain fingerprint-embeddings of each CU within the polymer, and E is a set of edges that indicate how CUs are

topologically connected. To process this representation, a graph convolutional network (GCN) is used to update the CU-fingerprint embeddings, which are then aggregated and passed to a DNN for final property prediction. We hypothesized that this approach would encode useful sequence information for the property prediction task and tested this strategy for dataset A. We considered two graph convolutional architectures: the graph convolutional layer[80] and the graph attention layer.[81] Both layers aggregate and utilize neighbor embeddings when updating a node embedding; however, the graph attention layer possesses additional parameters that allow neighbors to have differing levels of importance when performing the update. After a maximum of two graph convolutions, the node embeddings are aggregated and passed as input to a DNN.

A potential benefit of the sequence graph representation strategy is that the outputs from both the graph convolutional and graph attention layers are permutationally equivariant, meaning the output of per-node features is not sensitive to the order of graph nodes. Therefore, when paired with subsequent sum or average pooling layers, the final polymer property prediction is invariant to sequence inversion. Accordingly, datasets A, B, and D do not require data augmentation when training graph neural network models.

2.3.1.2 Sequence tensor. We additionally consider representations for which the CU fingerprints are stacked to form a tensor. In this approach, one dimension tracks the ordering of CUs within the polymer sequence, and the remaining dimensions relate to the CU fingerprint. For dataset A, where polymers have varying degrees of polymerization, all sequences are padded with zeros to match the length of the longest polymer.

To process the sequence tensor, we employ two approaches. In the first, a one-dimensional convolutional neural network (CNN) architecture leads into a DNN; this strategy is analogous to the "property-coloring" scheme discussed in ref. 43. The essential premise is that convolution operations performed over windows of the sequence can extract high-level, hierarchical feature correlations that may be useful for polymer property prediction. The CNN works by sliding a kernel over the numerical representation of the polymer and extracting sequence-level features. This operation, paired with pooling and subsequent convolutions, allows the model to directly construct hierarchical features. Inspired by demonstrated utility in modeling polymer sequences,[43] we also test long-short term memory (LSTM) architectures. The LSTM is a type of recurrent neural network (RNN) that processes a sequence in a unit-by-unit fashion, with model-specific parameters and operations that retain information from previously processed units. Similarly to CNNs, LSTMs can facilitate algorithmic identification and extraction of sequence features that could relate to polymer properties.

Notably, the sequence tensor representation does not natively retain invariance to sequence inversion as processed by our current RNN and CNN architectures. While the intermediate outputs of the CNN and pooling operations are equivariant to sequence inversions, the subsequent flattening operation and feeding into a DNN preserves the order of CU features. Thus, the final output is not invariant to sequence inversion. To address this, we take a two-fold approach when training models that takes a sequence tensor as input. First, we augment training data with inverted sequences labeled with the same property value as the forward sequence. Second, we average the output of the forward and reverse sequence to make predictions during testing. The former strategy acts as a form of regularization, whereas the latter ensures invariance to sequence inversion and can be seen as a type of test-time augmentation.[71] Dataset B is further augmented with sequences constructed from cyclic permutations of the four CUs comprising the repeat pattern of the polymer, as previously described.[43]

2.3.2 Scaled fingerprints. The scaled fingerprint approach can be employed in settings when precise polymer sequence is known as well as when such information is absent or ambiguous. Here, the representation effectively constitutes a weighted average of CU fingerprints $\bar{f} = \sum_k x_k f_k$ where the weight associated with the kth CU is determined based on, *e.g.*, its fraction of incorporation in the polymer x_k; this representation is effectively the same as that described by Kuenneth *et al.*[47] This representation can be derived from the sequence tensor by simply summing along the sequence axis and dividing by N. In theory, it can also be obtained from the graph of CU embeddings if there are no node update operations and instead the embeddings are pooled together using fractions of incorporation as attention-like parameters. The lack of any graphical operations highlights a potential limitation of such a polymer fingerprint: information regarding polymer connectivity or CU patterning is absent. Nevertheless, one advantage is that it can be constructed in most experimental and *in silico* design problems. For the polymers in dataset C, this is the only viable option because no sequence information is present.

The scaled fingerprint $\bar{f}$ can be modified in several ways, depending on the availability of other descriptors. One common descriptor may be the size of the polymer, which is observed to vary amongst polymers in datasets A and C, for example. We consider two approaches to encoding the information on polymer size. In the first, we simply multiply $\bar{f}$ by the measure of polymer size (*e.g.*, N) to obtain a final polymer fingerprint f; we refer to this as a size-implicit scaled fingerprint. We note that when the representation of size is the degree of polymerization and the weights for computing $\bar{f}$ are fractions of incorporation, the resulting feature vector is effectively a "Bag-of-features" or possibly "Bag-of-words" representation. For example, if a dimension in the CU fingerprint contains its average charge, then the size-implicit scaled fingerprint will report the net charge of the polymer. If the CU fingerprint is given by OHE and the CU is a monomer, then one obtains an enumeration of how many

monomers of each type are present in the polymer, or a "Bag-of-monomers." In the second approach, we add another dimension to $\bar{f}$ to include the polymer size; we refer to this as a size-explicit scaled fingerprint.

In addition to these (optional) modifications, we also consider augmenting scaled fingerprints with additional descriptors of the polymer. This approach can be used to partially address the lack of connectivity information in the scaled fingerprints by adding dimensions for sequence-level or topological descriptions. We refer to this approach as an augmented fingerprint and test it for the simulation-derived datasets (datasets A, B, and D). For dataset A, we consider sequence charge decoration (SCD), which captures the spacing of charge along a polymer chain, and sequence hydropathy decoration (SHD), which captures information about the spacing of hydrophobic components along a polymer chain.[82] For datasets B and D, we compute blockiness parameters for each polymer as

$$b_j = 1 - \frac{1}{N}\sum_{k=0}^{N-1} \mathbb{1}(f_j(k), f_j(k+1)), \tag{3}$$

with $\mathbb{1}(f_j(k), f_j(k+1))$ as an indicator function that is equal to one if and only if all dimensions of the CU fingerprints f_k and f_{k+1} related to position j of the CU are identical; it is zero otherwise. In the context here, $\mathbb{1}(f_j(k), f_j(k+1)) = 1$ implies that the CG bead at position j in the kth CU is the same as the CG bead at position j in the $(k+1)$th CU. For polymers in dataset B, $j = 0$, 1, or 2, such that the scaled fingerprint is augmented by three dimensions. For polymers in dataset D, $j = 0$ (they are linear polymers), such that the scaled fingerprint is augmented by a single dimension.

Because scaled fingerprints are constructed by summation of CU descriptors and augmentation with sequence-level descriptors that are permutationally invariant, the polymer representation itself is invariant to sequence inversion. Thus, the output of any model using this featurization strategy will also retain this property.

2.4 Model training and evaluation

The performance of each featurization strategy is obtained by averaging performance metrics obtained using a nested, five-fold cross-validation procedure. In particular, each dataset is initially split into five outer folds. For each outer fold, a set of optimal hyperparameters for the ML model is obtained by an inner five-fold cross-validation. The hyperparameter optimization is facilitated by using the tree-structured Parzen estimator (TPE) approach as implemented in Hyperopt[83] to minimize the average mean-squared-error (MSE) across inner folds. The search is conducted in a staged fashion wherein 100 random sets of hyperparameter combinations and evaluations are followed by 100 Bayesian optimization steps with the TPE algorithm. For ML models using LSTMs, hyperparameters were identified only using random search due to the computational expense associated with their training. Using the best set of hyperparameters, a model is trained and evaluated on the outer test fold. This process is repeated until every fold has served as a test fold. Care is taken to ensure augmented data variants do not simultaneously appear in both the train and test splits. The coefficient of determination, r^2, and mean absolute error, MAE, are used to assess model performance over all test sets. Through exploratory analysis, we find that the predictive performance of a model built with a particular featurization strategy can be sensitive to changes in hyperparameters. Thus, to best target comparisons between different featurization strategies, all models are hyperparameter-optimized before being tested for prediction. Additional details and discussion can be found in section 2.5 of the ESI.†

All reported metrics represent the average values across test sets, and errors indicate the standard error of the mean. To represent variation of MAE over consistent scales, we also introduce a normalized MAE, which corresponds to MAE divided by the average property value in the given dataset. The hyperparameter domains, performance metrics, and other training settings are provided in the ESI.† All neural networks were trained using Tensorflow,[84] and Spektral[85] was used to implement graph convolutional network layers.

3 Results and discussion

3.1 Representation of polymer size

Many polymer properties directly depend on the degree of polymerization or molecular weight of a polymer,[86,87] which make it an important candidate descriptor in polymer featurization. While the notion of polymer size is seemingly already expressed in the explicit-sequence featurization strategies, we sought to first quantify the impact of size representation by comparing the performance of ML models trained with scaled fingerprints (SFP), size-implicit SFPs, and size-explicit SFPs for datasets A and B (Fig. 2).

Fig. 2A shows that fingerprints that use either size-explicit or size-implicit representations of the polymer significantly improve ML models trained to predict properties in dataset A. In particular, we observe in excess of a 50% decrease in MAE compared to using a simple scaled fingerprint for all prediction tasks. In the case of the properties tested (R_g, C_v, and τ_N), these results are overall expected because polymer size has clear implications for each. However, for a given fingerprint type, we generally do not observe a statistically significant advantage to using size-explicit *versus* size-implicit representations. Thus, for polymers in dataset A, the inclusion of N is crucial to a successful polymer featurization, but there is flexibility in the method of representation.

By comparison, Fig. 2B shows that there is no clear advantage in providing a measure of polymer size in the polymer fingerprint for ML prediction tasks over dataset C. In this case, the representation of polymer size is the mean number-averaged molecular weight, and we do not observe statistically significant reductions in MAE compared to

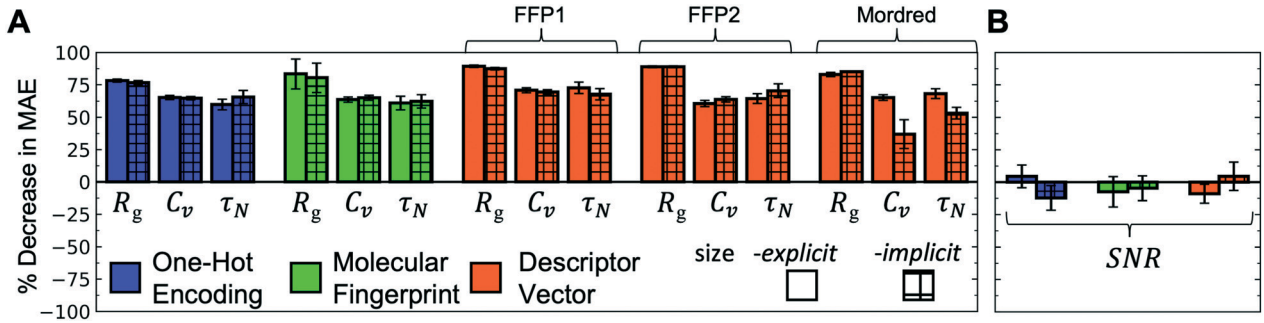

Fig. 2 Comparison of size-explicit *versus* size-implicit scaled fingerprint strategies for various fingerprints as applied to property prediction tasks for (A) coarse-grained intrinsically disordered proteins in dataset A and (B) stochastic methacrylate copolymers as ^{19}F MRI agents in dataset C. Both panels illustrate the percent decrease in mean absolute error (MAE) for a simple scaled fingerprint compared to that with either size-explicit representation (plain bars) or size-implicit representation (hatched bars). In (A), results are shown for ML models trained to predict the radius of gyration R_g, the heat capacity C_v, and the end-to-end decorrelation time τ_N. In (B), the property label is an experimentally determined signal-to-noise ratio (SNR) as reported in ref. 48. Both panels examine the effect on MAE using one-hot encoding (purple), molecular fingerprints (green), and descriptor vector (orange) approaches to CU fingerprinting. In (A), three descriptor vectors are used. The first two are vectors of force-field parameters (FFP); FFP1 excludes cross interactions while FFP2 additionally uses cross interaction parameters; the third is obtained from the chemical structure using Mordred.[58]

models trained using only scaled fingerprints, irrespective of the CU fingerprinting technique. We speculate that the SNR property label is not especially sensitive to polymer size over the size-range explored in dataset C: the standard deviation of molecular weight is 1100 g mol^{-1} compared to the mean of 7770 g mol^{-1} across the dataset. In contrast, the range of polymer sizes in dataset A spans from N = 20 up to 600. In addition to the lack of variability in molecular weight, other factors may include the overall dataset size and statistical noise associated with SNR, such that any potential effect of molecular weight is obfuscated by measurement noise. Nevertheless, inclusion of polymer size does not remarkably decrease the performance of ML models compared to the simple scaled fingerprint. Therefore, for most design tasks, it seems generally advisable to include either an implicit or explicit description of polymer size in the polymer feature vector.

3.2 Effect of explicit sequence representation

Many polymer materials systems may have the opportunity to exploit the sequential or topological arrangement of CUs to tailor properties or enhance figures-of-merit. Previous studies have variously explored both recurrent neural networks and CNNs in polymer property prediction tasks, presumably to extract and correlate sequence patterns with property labels; however, such strategies are rarely compared. To provide some guidance regarding polymer featurization when sequence is known, we constructed and compared the performance of three ML models that use explicit-sequence representation for the IDPs in dataset A to predict their radius of gyration R_g. In particular, models are developed using sequence tensors with one-dimensional CNNs, sequence graphs with GCNs, and sequence tensors with long-short-term memory (LSTM) networks. To control for any potential role of different CU fingerprinting strategies, all comparisons are made between models that use OHE for the CU fingerprints.

Fig. 3 summarizes the performance for the different sequence-processing strategies, with panels A–C providing correlation plots between ML predictions and the "ground truth" results obtained from MD simulation and panel D comparing the normalized MAE. We find that all architectures perform respectably in predicting R_g, with r^2 in excess of ~0.9. Among the various strategies compared in Fig. 3, the CNN exhibits statistically lower MAE compared to both the GCNs (22% lower) and the LSTM (18% lower). Interestingly, comparison of Fig. 3B and C suggests that the use of sequence graphs with GCNs is superior to using sequence tensors and LSTMs, although Fig. 3D illustrates slightly lower MAE for the LSTM architecture. The reason is clear from inspection of Fig. 3C, which reveals that processing sequence tensors with LSTMs provides reliable predictions for short chains while systematically underestimating R_g for larger chains. This suggests that the LSTM architecture may not encode an effective representation of polymer size, which was shown to have significant impact for dataset A prediction tasks in section 3.1. While we expected similar performance between CNN and GCN, we believe that the GCN performance was somewhat limited by the lengthscale of node embeddings and the number of allowable graph convolutions in our architectures. Conversely, the CNN could aggregate features over much larger length scales by utilizing larger kernel windows, which were found to span ~20 CUs after hyperparameter optimization.

Based on the overall success of the explicit-sequence representations, we also examined performance for dataset B, for which similar architectures were examined in ref. 43. In that study, Webb *et al.* developed an ML model that used a two-dimensional CNN (labeled as property-coloring) to process regular copolymer sequences; the performance of

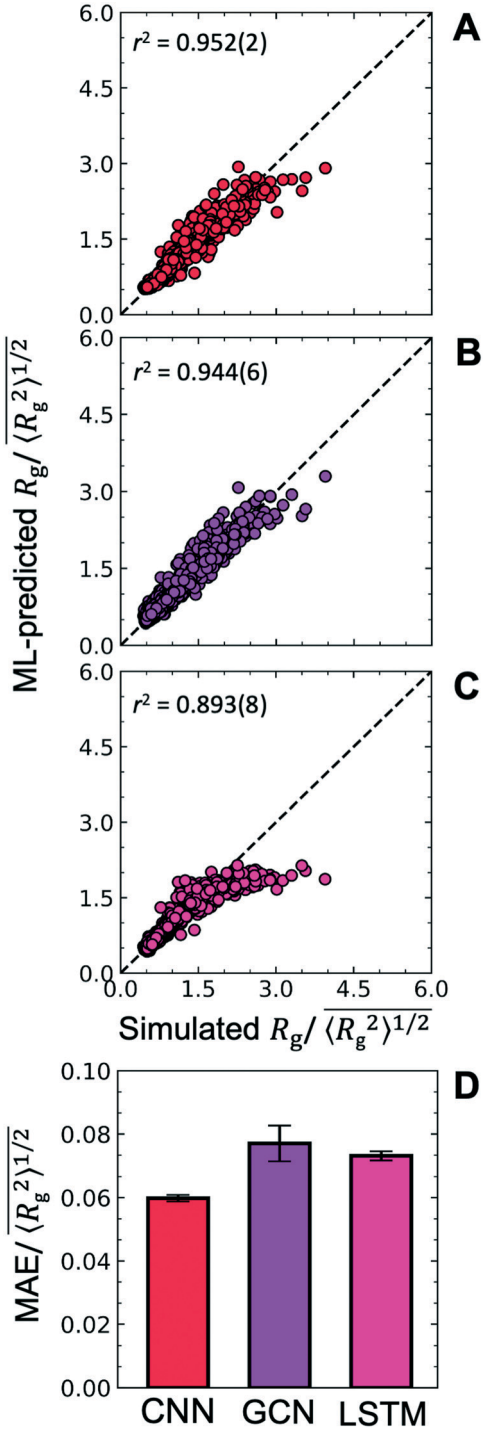

Fig. 3 Comparison of explicit-sequence featurization strategies for R_g prediction tasks in dataset A. Note that the axes labels are shared for panels (A)–(C); all data points in the correlation plots correspond to when the given polymer is in the held-out test fold during cross validation. Panel (D) reports the normalized MAE for sequence models in the prediction task. Standard errors and means for all quantities are obtained from the results of five-fold cross-validation. In the labels, $\langle\cdot\rangle$ denotes an ensemble average (obtained from statistical sampling from simulation) and $\overline{\cdot}$ denotes an average over the dataset.

that model for a simple 80/20 train/test split was reported as r^2 = 0.958 and MAE = 106 σ^2, where σ is the characteristic size of a CU with a single CG bead. In the present paper, we find similarly good performance with a one-dimensional CNN over OHE CU fingerprints (r^2 = 0.946 and MAE = 111 σ^2 obtained using five-fold cross-validation). Additionally, Webb *et al.* reported r^2 = 0.895 and MAE = 130 σ^2 for an LSTM model that predicts $\langle R_g^2\rangle$ for stochastic copolymer sequences using training data only from regular copolymer sequences. Interestingly, for the same task, we find somewhat better performance (r^2 = 0.926 MAE of 110 σ^2) using an ensemble model obtained from the five-fold cross-validation procedure, *i.e.*, the predicted labels are an average of predictions generated by five separate models. Although hyperparameter optimization was not reported in ref. 43, the present results indicate that the CNN model can capture sequence correlations and generalize these patterns to non-regular sequences somewhat better than the LSTM architecture.

3.3 Sequence and topology representations

The results of section 3.2 demonstrate explicit-sequence representations can be effective; however, it is not clear to what extent the ML regression model efficiently leverages this sequence-level information in its predictions. To assess the importance of sequence information on property prediction, we compared three featurization strategies that utilize different levels of sequence information; we considered prediction tasks on the simulation-derived datasets A, B, and D because the sequences are precisely known. The first strategy (CNN) uses a sequence tensor processed by a one-dimensional CNN. The second strategy (SFP) uses a scaled fingerprint, such that there is no explicit sequence information. The third strategy (aug. SFP) uses the same feature vector as the second strategy but the polymer fingerprint is additionally augmented with some descriptors (see section 2.3.2) that provide some characterization of sequence and/or topology. All strategies use a OHE fingerprint to distinguish the CUs. The results are provided in Fig. 4.

Fig. 4A–D compare the performance of the three featurization strategies for predicting R_g for the polymers in dataset A. Surprisingly, we find that the ML model that uses size-implicit SFPs (effectively a "Bag-of-Amino Acids") statistically outperforms the sequence tensor/CNN model both in terms of r^2 (0.952 for the CNN in Fig. 4A *versus* 0.972 in Fig. 4B) and MAE (see Fig. 4D). Meanwhile, using aug. SFPs yields the most accurate models. In fact, simply adding these descriptors reduces the MAE by 32% compared to the simple SFP approach. Thus, while comparing Fig. 4A and B suggests that R_g in dataset A is primarily driven by CU composition and polymer size, comparing Fig. 4B and C indicates that there are sequence-level effects that can influence R_g within the dataset. In theory, both the model derived from the simple scaled fingerprint as well as that augmented with sequence descriptors are within the function space of the sequence tensor/CNN model, which performs the worst of the three. We speculate that this is primarily due

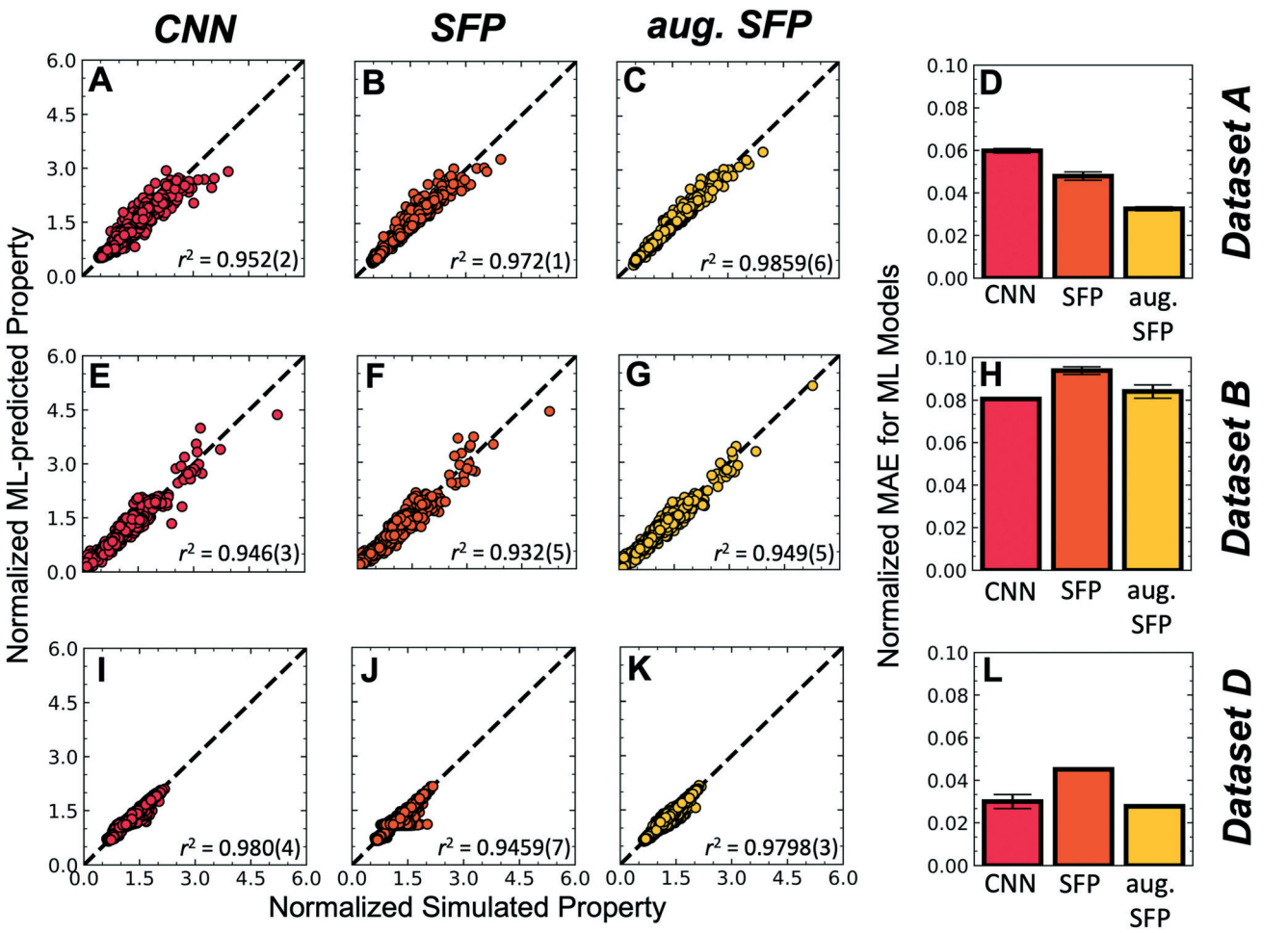

Fig. 4 Comparison of featurization strategies with varying levels of sequence information. Note that the axes labels are shared for panels (A)–(C), (E)–(G), and (I)–(K). The coefficients of determination are reported with standard errors for the last digit in parentheses. All data points in correlation plots correspond to when the given polymer is in the held-out test fold during cross validation. Panels (D, H and L) report the average normalized MAE for CNN models, scaled fingerprint models, and scaled fingerprint models augmented with topological descriptors for prediction tasks associated with datasets A, B, and D, respectively. Standard errors and means are obtained from the results of five-fold cross-validation.

to data limitations. In particular, the properties examined are principally governed by composition and polymer size, and sequence variation is perhaps a perturbative or noise-level effect. Consequently, it is difficult to extract meaningful sequence patterns on R_g (or other properties in dataset A) from the sequences in the DisProt database. Thus, it is more data-efficient to directly encode descriptors of sequence in the feature vector.

Fig. 4E–H compare the performance of the three featurization strategies for predicting $\langle R_g^2 \rangle$ for the polymers in dataset B. The CNN strategy is comparable to the aug. SFP strategy in terms of its performance metrics. Both are statistically superior to the SFP strategy, reducing MAE by 14% and 11% upon including sequence-level information *via* the CNN and sequence descriptors, respectively; the r^2 improves from 0.932 to 0.946 and 0.949. We attribute the relative success of the sequence tensor/CNN strategy, which is not encountered for dataset A, to several factors. First, the properties of polymers in dataset B likely exhibit amplified sequence effects compared to those in dataset A. In particular, the polymers in dataset B experience variations to intramolecular bonding potentials due to sequence,[43] while this is not the case for the HPS model for CG IDPs.[78] Secondly, there are relatively fewer unique non-bonded interactions amongst CG beads for polymers in dataset B compared to those for polymers in dataset A. Thirdly, by construction, there are well-defined, systematic sequence patterns in dataset B, while the origin of sequences in dataset A is comparatively uncontrolled. We believe the combination of these factors facilitate feature extraction from polymers in dataset B.

Fig. 4I–L compare the performance of the three featurization strategies for predicting ΔF_{ad} for the polymers in dataset D. Analogously with the discussion surrounding dataset B, we find that explicitly representing the sequence or providing sequence-level descriptors statistically improves the predictive capabilities of ML models compared to models that do not possess sequence information. In particular, there is a 34% reduction in MAE when using the CNN strategy *versus* SFP and a 39% reduction when using aug. SFP

versus simple SFP. Notably, both the CNN strategy and the aug. SFP strategy exhibit r^2 that rival the highest reported r^2 in ref. 74, although here we develop a single model for all surfaces based on DNN whereas Shi *et al.* develops separate SVR models for each surface, such that direct comparisons are difficult. The sequence tensor/CNN strategy likely again performs well due to the relatively small number of CUs and a comparatively abundant number of training examples, which enables facile extraction of relevant sequence patterns. Another contributing factor may be the monodispersity of sequence length in dataset C compared to that of dataset A.

Considering all the data in Fig. 4, ML models built with aug. SFPs are consistently good across prediction tasks. This suggests that this simple fingerprinting approach may be preferred or at least a viable alternative to more complicated strategies that use CNNs or GCNs, even when precise sequence or topological information is known. From a practical standpoint, such models would also be cheaper to optimize. One potential advantage to the aug. SFP approach is the opportunity to leverage domain-specific knowledge or make use of well-known descriptors as we have here. On the other hand, this may also bias the ML models and limit the information content of feature vectors to only human-crafted descriptors. In principle, using sequence tensors or graphs with convolutional networks provides an overall more flexible, unbiased approach to featurizing polymers. Because we do not observe remarkably poor performance with this approach for any prediction task here, using explicit-sequence featurization strategies are still likely viable, but they may not immediately provide the most accurate property predictions.

For design tasks, both explicit-sequence featurization or aug. SFPs would be reasonable for use in surrogate modeling during property optimization. A potential advantage of the explicit-sequence featurization is that optimization to identify a specific polymer is well defined. By contrast, additional effort would be required to chemically invert the optimal descriptor vectors into a sequence were one to optimize directly in the feature space of an aug. SFP. Optimization could be used in sequence space with surrogate evaluations performed with the aug. SFP featurization strategy, but this may undesirable due to possible degeneracy in the sequence-to-aug. SFP mapping.

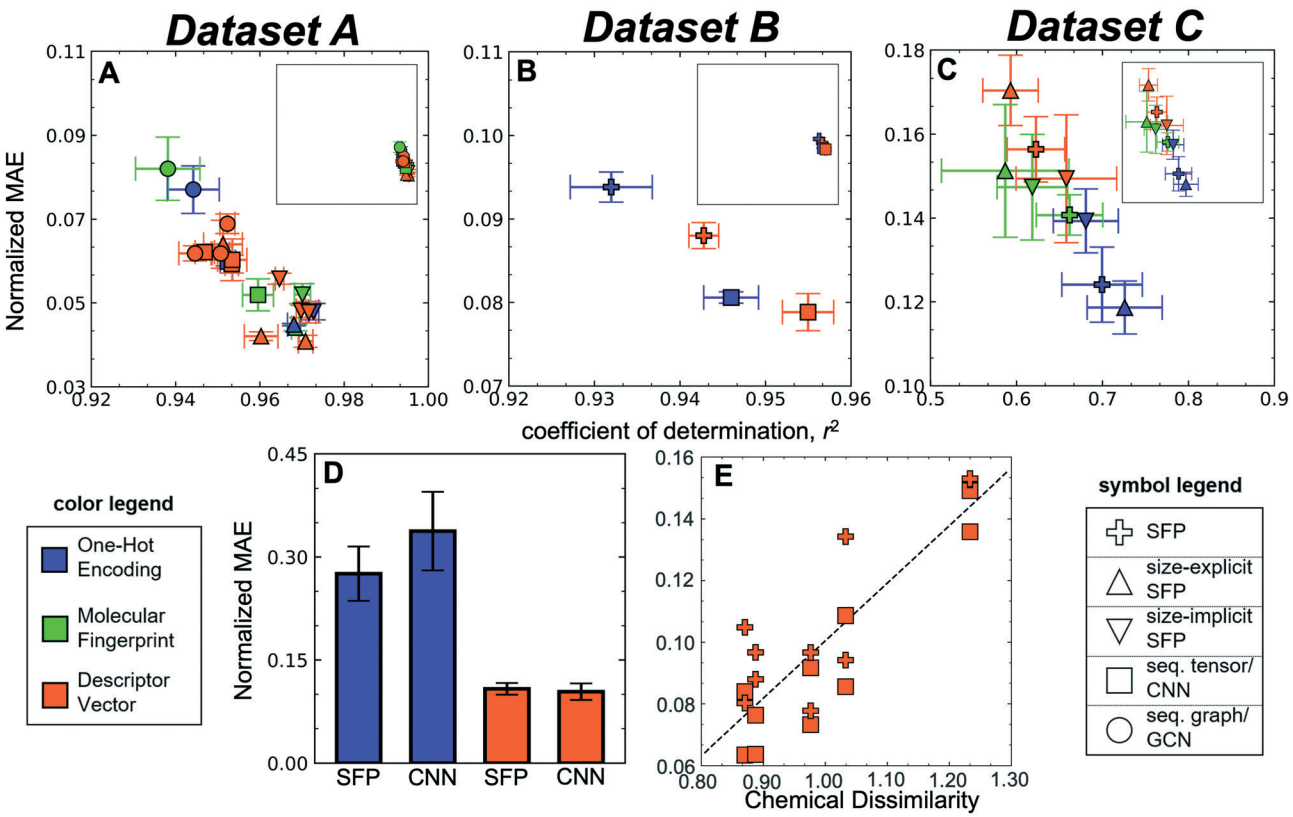

Fig. 5 Comparison of CU fingerprints combined with different featurization strategies. Model performance given by the coefficient of determination r^2 and the normalized mean absolute error (MAE) for prediction tasks on (A) datasets A, (B) dataset B, and (C) dataset C. Standard errors are obtained from the results of five-fold cross-validation. The insets on all graphs range over the same intervals (0.5 to 1.0 for the abscissa and 0.0 to 0.2 for the ordinate) for visual reference across panels. (D) The extrapolative ability of models using different CU fingerprints to make predictions on polymers with previously unseen CUs. Each bar corresponds to a different featurization strategy; the analysis is based on dataset B. (E) The correlation of extrapolation error with chemical dissimilarity of the unseen CUs in the test set to CUs present in the training set (Pearson's correlation coefficient = 0.86) the chemical dissimilarity is quantified as the sum of Euclidean distances of a CU descriptor vector to all others in the chemical space. The color and symbol legends apply to all panels, as relevant.

3.4 Impact of constitutional unit fingerprints

In previous sections, we simplified comparisons by using only OHE fingerprints of the CUs, achieving overall excellent predictive accuracy. Still, OHE is a limited representation that is deficient in any notion of chemical similarity amongst CUs, such that all CUs are equidistant in the chemical feature space. In addition, the dimensionality of OHE fingerprints scales with the number of possible CUs, which may be problematic for less restricted design spaces. Both factors limit the transferability of ML models constructed with OHE fingerprints of CUs. We hypothesized that using chemical fingerprints or descriptor vectors would enhance the predictive capabilities of ML models by allowing for a better expression of chemical similarity. To investigate the utility of these chemically-informed encodings, different representations of CUs were used in conjunction with the SFP and explicit-sequence featurization strategies for regression tasks across datasets A, B, and C (Fig. 5). Because datasets A and C have CUs that can be described by real chemical structures, despite dataset A featuring CG polymers, Fig. 5A and C compare OHE, molecular fingerprints, and descriptor vectors for use as fingerprints of the CUs. For dataset B, the comparison is limited to only OHE *versus* descriptor vectors as there are no underlying chemical structures for the CUs. We do not investigate this comparison for dataset D since the two representations are identical for this simple system: representations using OHE are related to representations in the basis of force-field parameters by a linear transformation.

Fig. 5A reveals that most SFP-based strategies with size representation perform similarly, irrespective of the type of CU fingerprint and the prediction task for dataset A. Meanwhile, there is no evident systematic advantage for any given CU fingerprint when used along with explicit-sequence featurization strategies. In fact, the models utilizing the OHE CU fingerprints are either the best or within statistical error of the best-performing models (controlling for a given model type and prediction task). The most noticeable result is that graph-based models have generally larger errors, but overall, all models exhibit overall high accuracy.

Examination of Fig. 5B, which considers OHE and descriptor vector CU fingerprints in both SFP and sequence graph/GCN featurization strategies for polymers in dataset B, provides somewhat similar conclusions. In this case, however, using descriptor vectors does consistently enhance predictive capabilities compared to using OHE for the CU fingerprints. While the advantage is more striking when using SFPs than when using explicit-sequence featurization, the differences remain overall modest when considering the proximity of all points for generally accurate models.

In stark contrast, Fig. 5C clearly demonstrates relative success of OHE fingerprints for CUs compared to either molecular fingerprints or descriptor vectors. Between molecular fingerprints and descriptor vectors as the CU fingerprints, molecular fingerprints seem to provide overall more accurate models, but the advantage is not always statistically significant. We note that the error bars are larger here than in either Fig. 5A or B due to the dataset being smaller and the labels being more prone to statistical noise. Consequently, we expect that the low-dimensionality of SFP-based models with OHE is an advantage in data-scarce regimes and in prediction tasks with larger measurement uncertainties.

To probe the utility of using chemically informed CU feature vectors when constructing extrapolative ML models, models were retrained and tested for the prediction task in dataset B using alternative train-test splits. Specifically, train-test splits were constructed such that a single CU type is missing from all polymers in the training data but present in all polymers in the test set. Only dataset B was used for this investigation as it was the only dataset with data composition such that reasonably sized train-test splits could be constructed for all CUs. Fig. 5D shows that models trained on polymer representations constructed using descriptor vector CU fingerprints extrapolate to polymers with "unseen" CUs significantly better than those constructed from simple OHEs. We hypothesized that the model MAE would be closely related to the chemical dissimilarity of the unseen CU to those in the training data. To investigate this, we defined chemical dissimilarity as the Euclidean distance of a chosen CU to all other CUs in the chemical space and examined its correlation with MAE; the results are shown in Fig. 5E. Together, these results support the idea that representing a CU in a chemically informed vector space can allow for the ML model to extrapolate to new chemical systems by encoding relationships between nominally distinct units. Thus, when design tasks allow exploration outside of the chemical space of the training data (*e.g.*, in a generative approach), we recommend the use of chemically specific CU fingerprints.

Conclusions

In this paper, we introduced, examined, and compared the performance of various polymer featurization strategies for diverse ML regression tasks derived from four distinct datasets. We considered polymer featurization from the perspective that polymers are comprised of constitutional units, which may be described in numerous ways, and that the precise sequence or topology of CUs may or may not be known, depending on the design space or synthetic capabilities. Therefore, we outlined a series of approaches that invoked varying degrees of sequence-level information. We additionally considered the special role of polymer size in property prediction when it is a known variable in the dataset.

Our results indicate that the "best" polymer featurization strategy is context-dependent, and its performance may also be degenerate with other featurization strategies. For example, in regression tasks associated with datasets A and B, descriptor vectors performed as well as, if not better, than

models that use OHE. However, for the lone experimental dataset, OHE CU representations definitively outperformed molecular fingerprinting or descriptor vector strategies, although we expect this advantage to diminish for larger datasets. Matching our intuition, we find that featurizing polymers with chemically informed representations of chemical units, as opposed to simple OHEs, facilitates extrapolation, which may be useful in some design paradigms.

In situations where sequence information is known, we found consistent advantages to leveraging sequence information compared to relying solely on composition. However, explicit-sequence representations coupled with feature extraction architectures did not outperform simpler models built using fingerprints augmented with sequence descriptors. Because sequence descriptors are derivable from explicit sequence representations, this result likely stems from data limitations. Here, scaled fingerprints augmented with sequence descriptors seemingly provide a data-efficient approach to encode essential sequence characteristics for ML models, which is advantageous for polymer design tasks. Finally, we find that some representation of polymer size is either necessary to achieve accurate ML models or, at worst, inconsequential, depending on the property prediction task.

The current work also points to several interesting questions for polymer featurization that can be considered for future polymer design problems. For example, while we found that processing sequence information through CNNs was generally more effective and computationally expeditious compared to GCNs or LSTMs, the performance limitations or applicability of all these approaches are still not fully understood. We also did not assess the performance or viability of low-dimensional polymer embeddings achieved using unsupervised ML techniques[88] or variational autoencoders.[41,67] Another consistent theme uncovered by exploration of multiple datasets is the potential sensitivity of polymer featurization to dataset construction. For example, we believe that the comparatively poor performance of explicit-sequence models for dataset A is because sequence effects must be ascertained from random occurrence of sequence motifs across the dataset, and any relevant effects are small by comparison to those arising from composition or polymer size. This highlights a need to carefully consider dataset construction, if one aims to use explicit-sequence representations.

Conflicts of interest

There are no conflicts to declare.

Acknowledgements

This research was supported by the National Science Foundation under DMREF Award Number NSF-DMR-2118861. The authors thanks Jeetain Mittal for assistance related to the implementation of the HPS model for intrinsically disordered proteins. The authors also thank Jiale Shi and Jonathan Whitmer for discussing and sharing data on free energy of adhesives of model polymers. The authors also acknowledge Princeton Research Computing and an Azure Cloud computing mini-grant obtained from the Center for Statistics and Machine Learning at Princeton University *via* a gift from Microsoft Corporation.

References

1 A. J. Liu, G. S. Grest, M. C. Marchetti, G. M. Grason, M. O. Robbins, G. H. Fredrickson, M. Rubinstein and M. O. de la Cruz, Opportunities in theoretical and computational polymeric materials and soft matter, *Soft Matter*, 2015, **11**, 2326–2332.

2 J.-F. Lutz, J.-M. Lehn, E. W. Meijer and K. Matyjaszewski, From precision polymers to complex materials and systems, *Nat. Rev. Mater.*, 2016, **1**, 16024.

3 J. J. de Pablo, N. E. Jackson, M. A. Webb, L.-Q. Chen, J. E. Moore, D. Morgan, R. Jacobs, T. Pollock, D. G. Schlom, E. S. Toberer, J. Analytis, I. Dabo, D. M. DeLongchamp, G. A. Fiete, G. M. Grason, G. Hautier, Y. Mo, K. Rajan, E. J. Reed, E. Rodriguez, V. Stevanovic, J. Suntivich, K. Thornton and J.-C. Zhao, New frontiers for the materials genome initiative, *npj Comput. Mater.*, 2019, **5**, 41.

4 S. L. Perry and C. E. Sing, 100th Anniversary of Macromolecular Science Viewpoint: Opportunities in the Physics of Sequence-Defined Polymers, *ACS Macro Lett.*, 2020, **9**, 216–225.

5 K. Matyjaszewski, Macromolecular engineering: From rational design through precise macromolecular synthesis and processing to targeted macroscopic material properties, *Prog. Polym. Sci.*, 2005, **30**, 858–875.

6 J.-F. Lutz, M. Ouchi, D. R. Liu and M. Sawamoto, Sequence-Controlled Polymers, *Science*, 2013, **341**, 1238149.

7 G. Polymeropoulos, G. Zapsas, K. Ntetsikas, P. Bilalis, Y. Gnanou and N. Hadjichristidis, 50th Anniversary Perspective: Polymers with Complex Architectures, *Macromolecules*, 2017, **50**, 1253–1290.

8 Z. M. Sherman, M. P. Howard, B. A. Lindquist, R. B. Jadrich and T. M. Truskett, Inverse methods for design of soft materials, *J. Chem. Phys.*, 2020, **152**, 140902.

9 A. J. Gormley and M. A. Webb, Machine learning in combinatorial polymer chemistry, *Nat. Rev. Mater.*, 2021, **6**, 642–644.

10 C. Peter and K. Kremer, Multiscale simulation of soft matter systems – from the atomistic to the coarse-grained level and back, *Soft Matter*, 2009, **5**, 4357.

11 T. Yamamoto, Computer modeling of polymer crystallization – Toward computer-assisted materials design, *Polymer*, 2009, **50**, 1975–1985.

12 S. M. Loverde, Computer simulation of polymer and biopolymer self-assembly for drug delivery, *Mol. Simul.*, 2014, **40**, 794–801.

13 M. A. Webb, Y. Jung, D. M. Pesko, B. M. Savoie, U. Yamamoto, G. W. Coates, N. P. Balsara, Z.-G. Wang and T. F.

Miller, Systematic Computational and Experimental Investigation of Lithium-Ion Transport Mechanisms in Polyester-Based Polymer Electrolytes, *ACS Cent. Sci.*, 2015, **1**, 198–205.

14 M. A. Morris, T. E. Gartner and T. H. Epps, Tuning Block Polymer Structure, Properties, and Processability for the Design of Efficient Nanostructured Materials Systems, *Macromol. Chem. Phys.*, 2017, **218**, 1600513.

15 A. Jayaraman, 100th Anniversary of Macromolecular Science Viewpoint: Modeling and Simulation of Macromolecules with Hydrogen Bonds: Challenges, Successes, and Opportunities, *ACS Macro Lett.*, 2020, **9**, 656–665.

16 T. Bereau, Computational compound screening of biomolecules and soft materials by molecular simulations, *Modell. Simul. Mater. Sci. Eng.*, 2021, **29**, 023001.

17 S. Dhamankar and M. A. Webb, Chemically specific coarse-graining of polymers: Methods and prospects, *J. Polym. Sci.*, 2021, 1–31.

18 G. Chen, Z. Shen, A. Iyer, U. F. Ghumman, S. Tang, J. Bi, W. Chen and Y. Li, Machine-Learning-Assisted De Novo Design of Organic Molecules and Polymers: Opportunities and Challenges, *Polymer*, 2020, **12**, 163.

19 C. W. Coley, Defining and Exploring Chemical Spaces, *Trends Chem.*, 2021, **3**, 133–145.

20 B. Meredig, A. Agrawal, S. Kirklin, J. E. Saal, J. W. Doak, A. Thompson, K. Zhang, A. Choudhary and C. Wolverton, Combinatorial screening for new materials in unconstrained composition space with machine learning, *Phys. Rev. B*, 2014, 094104.

21 R. Gómez-Bombarelli, J. Aguilera-Iparraguirre, T. D. Hirzel, D. Duvenaud, D. Maclaurin, M. A. Blood-Forsythe, H. S. Chae, M. Einzinger, D.-G. Ha, T. Wu, G. Markopoulos, S. Jeon, H. Kang, H. Miyazaki, M. Numata, S. Kim, W. Huang, S. I. Hong, M. Baldo, R. P. Adams and A. Aspuru-Guzik, Design of efficient molecular organic light-emitting diodes by a high-throughput virtual screening and experimental approach, *Nat. Mater.*, 2016, **15**, 1120–1127.

22 R. Gómez-Bombarelli and A. Aspuru-Guzik, *Handbook of Materials Modeling*, Springer International Publishing, 2018, pp. 1–24.

23 K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev and A. Walsh, Machine learning for molecular and materials science, *Nature*, 2018, **559**, 547–555.

24 A. Agrawal and A. Choudhary, Deep materials informatics: Applications of deep learning in materials science, *MRS Commun.*, 2019, **9**, 779–792.

25 S. Chibani and F.-X. Coudert, Machine learning approaches for the prediction of materials properties, *APL Mater.*, 2020, **8**, 080701.

26 O. A. von Lilienfeld and K. Burke, Retrospective on a decade of machine learning for chemical discovery, *Nat. Commun.*, 2020, **11**, 4895.

27 R. Vasudevan, G. Pilania and P. V. Balachandran, Machine learning for materials design and discovery, *J. Appl. Phys.*, 2021, **129**, 070401.

28 B. Shahriari, K. Swersky, Z. Wang, R. P. Adams and N. de Freitas, Taking the Human Out of the Loop: A Review of Bayesian Optimization, *Proc. IEEE*, 2016, **104**, 148–175.

29 D. J. Audus and J. J. de Pablo, Polymer Informatics: Opportunities and Challenges, *ACS Macro Lett.*, 2017, **6**, 1078–1082.

30 R. Ramprasad, R. Batra, G. Pilania, A. Mannodi-Kanakkithodi and C. Kim, Machine learning in materials informatics: recent applications and prospects, *npj Comput. Mater.*, 2017, **3**, 54.

31 A. L. Ferguson, Machine learning and data science in soft materials engineering, *J. Phys.: Condens. Matter*, 2017, **30**, 043002.

32 J. S. Peerless, N. J. B. Milliken, T. J. Oweida, M. D. Manning and Y. G. Yingling, Soft Matter Informatics: Current Progress and Challenges, *Adv. Theory Simul.*, 2018, **2**, 1800129.

33 N. E. Jackson, M. A. Webb and J. J. de Pablo, Recent advances in machine learning towards multiscale soft materials design, *Curr. Opin. Chem. Eng.*, 2019, **23**, 106–114.

34 L. Chen, G. Pilania, R. Batra, T. D. Huan, C. Kim, C. Kuenneth and R. Ramprasad, Polymer informatics: Current status and critical next steps, *Mater. Sci. Eng., R*, 2021, **144**, 100595.

35 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, K. F. Jensen and B. D. Olsen, BigSMILES: A Structurally-Based Line Notation for Describing Macromolecules, *ACS Cent. Sci.*, 2019, **5**, 1523–1531.

36 R. Ma and T. Luo, PI1M: A Benchmark Database for Polymer Informatics, *J. Chem. Inf. Model.*, 2020, **60**, 4684–4690.

37 C. Kim, A. Chandrasekaran, A. Jha and R. Ramprasad, Active-learning and materials design: the example of high glass transition temperature polymers, *MRS Commun.*, 2019, **9**, 860–866.

38 L. Chen, C. Kim, R. Batra, J. P. Lightstone, C. Wu, Z. Li, A. A. Deshmukh, Y. Wang, H. D. Tran, P. Vashishta, G. A. Sotzing, Y. Cao and R. Ramprasad, Frequency-dependent dielectric constant prediction of polymers using machine learning, *npj Comput. Mater.*, 2020, **6**, 61.

39 J. W. Barnett, C. R. Bilchak, Y. Wang, B. C. Benicewicz, L. A. Murdock, T. Bereau and S. K. Kumar, Designing exceptional gas-separation polymer membranes using machine learning, *Sci. Adv.*, 2020, **6**, eaaz4301.

40 L. Tao, V. Varshney and Y. Li, Benchmarking Machine Learning Models for Polymer Informatics: An Example of Glass Transition Temperature, *J. Chem. Inf. Model.*, 2021, **61**, 5395–5413.

41 K. Shmilovich, R. A. Mansbach, H. Sidky, O. E. Dunne, S. S. Panda, J. D. Tovar and A. L. Ferguson, Discovery of Self-Assembling pi-Conjugated Peptides by Active Learning-Directed Coarse-Grained Molecular Simulation, *J. Phys. Chem. B*, 2020, **124**, 3873–3891.

42 R. Kumar, N. Le, Z. Tan, M. E. Brown, S. Jiang and T. M. Reineke, Efficient Polymer-Mediated Delivery of Gene-Editing Ribonucleoprotein Payloads through Combinatorial

Design, Parallelized Experimentation, and Machine Learning, *ACS Nano*, 2020, **14**, 17626–17639.
43 M. A. Webb, N. E. Jackson, P. S. Gil and J. J. de Pablo, Targeted sequence design within the coarse-grained polymer genome, *Sci. Adv.*, 2020, **6**, eabc6216.
44 S. Mohapatra, N. Hartrampf, M. Poskus, A. Loas, R. Gómez-Bombarelli and B. L. Pentelute, Deep Learning for Prediction and Optimization of Fast-Flow Peptide Synthesis, *ACS Cent. Sci.*, 2020, **6**, 2277–2286.
45 B. K. Wheatle, E. F. Fuentes, N. A. Lynd and V. Ganesan, Design of Polymer Blend Electrolytes through a Machine Learning Approach, *Macromolecules*, 2020, **53**, 9449–9459.
46 J. N. Kumar, Q. Li, K. Y. T. Tang, T. Buonassisi, A. L. Gonzalez-Oyarce and J. Ye, Machine learning enables polymer cloud-point engineering via inverse design, *npj Comput. Mater.*, 2019, 73.
47 C. Kuenneth, W. Schertzer and R. Ramprasad, Copolymer Informatics with Multitask Deep Neural Networks, *Macromolecules*, 2021, **54**, 5957–5961.
48 M. Reis, F. Gusev, N. G. Taylor, S. H. Chung, M. D. Verber, Y. Z. Lee, O. Isayev and F. A. Leibfarth, Machine-Learning-Guided Discovery of 19F MRI Agents Enabled by Automated Copolymer Synthesis, *J. Am. Chem. Soc.*, 2021, **143**, 17677–17689.
49 J. Kahovec, R. B. Fox and K. Hatada, *Nomenclature of regular single-strand organic polymers (IUPAC Recommendations 2002)*, 2002, 74, 1921–1956.
50 R. Upadhya, S. Kosuri, M. Tamasi, T. A. Meyer, S. Atta, M. A. Webb and A. J. Gormley, Automation and data-driven design of polymer therapeutics, *Adv. Drug Delivery Rev.*, 2021, **171**, 1–28.
51 L. M. Ghiringhelli, J. Vybiral, S. V. Levchenko, C. Draxl and M. Scheffler, Big Data of Materials Science: Critical Role of the Descriptor, *Phys. Rev. Lett.*, 2015, **114**, 105503.
52 F. A. Faber, L. Hutchison, B. Huang, J. Gilmer, S. S. Schoenholz, G. E. Dahl, O. Vinyals, S. Kearnes, P. F. Riley and O. A. von Lilienfeld, Prediction Errors of Molecular Machine Learning Models Lower than Hybrid DFT Error, *J. Chem. Theory Comput.*, 2017, **13**, 5255–5264.
53 T. J. Wills, D. A. Polshakov, M. C. Robinson and A. A. Lee, Impact of Chemist-In-The-Loop Molecular Representations on Machine Learning Outcomes, *J. Chem. Inf. Model.*, 2020, **60**, 4449–4456.
54 H. L. Morgan, The Generation of a Unique Machine Description for Chemical Structures-A Technique Developed at Chemical Abstracts Service, *J. Chem. Doc.*, 1965, **5**, 107–113.
55 D. Rogers and M. Hahn, Extended-Connectivity Fingerprints, *J. Chem. Inf. Model.*, 2010, **50**, 742–754.
56 S. Kearnes, K. McCloskey, M. Berndl, V. Pande and P. Riley, Molecular graph convolutions: moving beyond fingerprints, *J. Comput.-Aided Mol. Des.*, 2016, **30**, 595–608.
57 Z. Wu, B. Ramsundar, E. N. Feinberg, J. Gomes, C. Geniesse, A. S. Pappu, K. Leswing and V. Pande, MoleculeNet: a benchmark for molecular machine learning, *Chem. Sci.*, 2018, **9**, 513–530.
58 H. Moriwaki, Y.-S. Tian, N. Kawashita and T. Takagi, Mordred: a molecular descriptor calculator, *J. Cheminf.*, 2018, 4.
59 K. T. Schütt, H. E. Sauceda, P.-J. Kindermans, A. Tkatchenko and K.-R. Müller, SchNet – A deep learning architecture for molecules and materials, *J. Chem. Phys.*, 2018, **148**, 241722.
60 R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams and A. Aspuru-Guzik, Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules, *ACS Cent. Sci.*, 2018, **4**, 268–276.
61 F. Sandfort, F. Strieth-Kalthoff, M. Kühnemund, C. Beecks and F. Glorius, A Structure-Based Platform for Predicting Chemical Reactivity, *Chem*, 2020, **6**, 1379–1390.
62 L. Pattanaik and C. W. Coley, Molecular Representation: Going Long on Fingerprints, *Chem*, 2020, **6**, 1204–1207.
63 M. Krenn, F. Häse, A. Nigam, P. Friederich and A. Aspuru-Guzik, Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation, *Mach. Learn.: Sci. Technol.*, 2020, **1**, 045024.
64 A. Capecchi, D. Probst and J.-L. Reymond, One molecular fingerprint to rule them all: drugs, biomolecules, and the metabolome, *J. Cheminf.*, 2020, **12**, 42.
65 F. Musil, A. Grisafi, A. P. Bartók, C. Ortner, G. Csányi and M. Ceriotti, Physics-Inspired Structural Representations for Molecules and Materials, *Chem. Rev.*, 2021, **121**, 9759–9815.
66 S. Wu, Y. Kondo, M. A. Kakimoto, B. Yang, H. Yamada, I. Kuwajima, G. Lambard, K. Hongo, Y. Xu, J. Shiomi, C. Schick, J. Morikawa and R. Yoshida, Machine-learning-assisted discovery of polymers with high thermal conductivity using a molecular design algorithm, *npj Comput. Mater.*, 2019, **5**, 66.
67 R. Batra, H. Dai, T. D. Huan, L. Chen, C. Kim, W. R. Gutekunst, L. Song and R. Ramprasad, Polymers for Extreme Conditions Designed Using Syntax-Directed Variational Autoencoders, *Chem. Mater.*, 2020, **32**, 10489–10500.
68 A. Mannodi-Kanakkithodi, G. Pilania, T. D. Huan, T. Lookman and R. Ramprasad, Machine Learning Strategy for Accelerated Design of Polymer Dielectrics, *Sci. Rep.*, 2016, **6**, 20952.
69 C. Kim, A. Chandrasekaran, T. D. Huan, D. Das and R. Ramprasad, Polymer Genome: A Data-Powered Polymer Informatics Platform for Property Predictions, *J. Phys. Chem. C*, 2018, **122**, 17575–17585.
70 H. D. Tran, C. Kim, L. Chen, A. Chandrasekaran, R. Batra, S. Venkatram, D. Kamal, J. P. Lightstone, R. Gurnani, P. Shetty, M. Ramprasad, J. Laws, M. Shelton and R. Ramprasad, Machine-learning predictions of polymer properties with Polymer Genome, *J. Appl. Phys.*, 2020, **128**, 171104.
71 A. D. White, Deep Learning for Molecules and Materials, 2021.
72 S. Mohapatra, J. An and R. Gómez-Bombarelli, GLAMOUR: Graph Learning over Macromolecule Representations, 2021.

73 K. M. Jablonka, G. M. Jothiappan, S. Wang, B. Smit and B. Yoo, Bias free multiobjective active learning for materials design and discovery, *Nat. Commun.*, 2021, **12**, 2312.
74 J. Shi, M. J. Quevillon, P. H. A. Valença and J. K. Whitmer, Predicting Adhesive Free Energies of Polymer–Surface Interactions with Machine Learning, 2021.
75 D. Piovesan, F. Tabaro, I. Mičetić, M. Necci, F. Quaglia, C. J. Oldfield, M. C. Aspromonte, N. E. Davey, R. Davidović, Z. Dosztányi, A. Elofsson, A. Gasparini, A. Hatos, A. V. Kajava, L. Kalmar, E. Leonardi, T. Lazar, S. Macedo-Ribeiro, M. Macossay-Castillo, A. Meszaros, G. Minervini, N. Murvai, J. Pujols, D. B. Roche, E. Salladini, E. Schad, A. Schramm, B. Szabo, A. Tantos, F. Tonello, K. D. Tsirigos, N. Veljković, S. Ventura, W. Vranken, P. Warholm, V. N. Uversky, A. K. Dunker, S. Longhi, P. Tompa and S. C. Tosatto, DisProt 7.0: a major update of the database of disordered proteins, *Nucleic Acids Res.*, 2016, **45**, D219–D227.
76 A. Hatos, B. Hajdu-Soltész, A. M. Monzon, N. Palopoli, L. Álvarez, B. Aykac-Fas, C. Bassot, G. I. Benítez, M. Bevilacqua, A. Chasapi, L. Chemes, N. E. Davey, R. Davidović, A. K. Dunker, A. Elofsson, J. Gobeill, N. S. G. Foutel, G. Sudha, M. Guharoy, T. Horvath, V. Iglesias, A. V. Kajava, O. P. Kovacs, J. Lamb, M. Lambrughi, T. Lazar, J. Y. Leclercq, E. Leonardi, S. Macedo-Ribeiro, M. Macossay-Castillo, E. Maiani, J. A. Manso, C. Marino-Buslje, E. Martínez-Pérez, B. Mészáros, I. Mičetić, G. Minervini, N. Murvai, M. Necci, C. A. Ouzounis, M. Pajkos, L. Paladin, R. Pancsa, E. Papaleo, G. Parisi, E. Pasche, P. J. B. Pereira, V. J. Promponas, J. Pujols, F. Quaglia, P. Ruch, M. Salvatore, E. Schad, B. Szabo, T. Szaniszló, S. Tamana, A. Tantos, N. Veljkovic, S. Ventura, W. Vranken, Z. Dosztányi, P. Tompa, S. C. E. Tosatto and D. Piovesan, DisProt: intrinsic protein disorder annotation in 2020, *Nucleic Acids Res.*, 2019, **48**, D269–D276.
77 A. P. Thompson, H. M. Aktulga, R. Berger, D. S. Bolintineanu, W. M. Brown, P. S. Crozier, P. J. in t Veld, A. Kohlmeyer, S. G. Moore, T. D. Nguyen, R. Shan, M. J. Stevens, J. Tranchida, C. Trott and S. J. Plimpton, LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales, *Comput. Phys. Commun.*, 2022, **271**, 108171.
78 R. M. Regy, J. Thompson, Y. C. Kim and J. Mittal, Improved coarse-grained model for studying sequence dependent phase separation of disordered proteins, *Protein Sci.*, 2021, 1371–1379.
79 RDKit: Open-source cheminformatics, http://www.rdkit.org.
80 T. N. Kipf and M. Welling, Semi-Supervised Classification with Graph Convolutional Networks, 2017.
81 P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò and Y. Bengio, Graph Attention Networks, 2018.
82 W. Zheng, G. Dignon, M. Brown, Y. C. Kim and J. Mittal, Hydropathy Patterning Complements Charge Patterning to Describe Conformational Preferences of Disordered Proteins, *J. Phys. Chem. Lett.*, 2020, **11**, 3408–3415.
83 J. Bergstra, D. Yamins and D. D. Cox, *Proceedings of the 30th International Conference on International Conference on Machine Learning*, Atlanta, GA, USA, 2013, vol. 28, p. I-115-I-123.
84 TensorFlow Developers, TensorFlow, 2021.
85 D. Grattarola and C. Alippi, *Graph Neural Networks in TensorFlow and Keras with Spektral*, 2020, arXiv:2006.12138v1.
86 M. Doi and S. Edwards, *The theory of polymer dynamics*, Clarendon Press, Oxford, 1986.
87 M. Rubinstein and R. Colby, *Polymer physics*, Oxford University Press, Oxford New York, 2003.
88 E. Asgari and M. R. K. Mofrad, Continuous Distributed Representation of Biological Sequences for Deep Proteomics and Genomics, *PLoS One*, 2015, **10**, e0141287.