

POLYMER REPRESENTATION LEARNING MODEL

Technical Field

[0001] The present disclosure relates to a polymer representation learning model that can use multiple data types as input and produce an embedding that captures aspects of a class of polymers contained within the input data. Such techniques can be useful to provide a recommendation related to production of a particular polymer within the class of polymers (such as chemicals and/or molecules with which to make a particular polymer within the class of polymers or manufacturing conditions for producing the particular polymer), or application of the particular polymer without actually having to make or test multiple polymers of the class.

Background

[0002] Machine learning and statistical analysis can be used to assist in designing chemicals and materials. Some empirical modeling methods use training data including independent variables that describe the chemical system of interest. Examples of such variables include descriptors (“X variables” or “X data”) and the desired attributes (“Y variables” or “Y data”) of the chemical product to be produced. Various algorithms can detect and encapsulate the patterns between X and Y variables. Model tools can be developed to enable users to test hypotheses about predicted outcomes of a new set of input variables or optimize inputs to meet a desired Y variable specification for a chemical product. One task in creating such a model for materials informatics is determining what data to use as X data. In order for a machine learning algorithm to detect patterns between X and Y, the X data variables should contain relevant information about the system and how it behaves. The less dense the correlated information content of the data, the more likely the algorithm is to miss patterns in the data (e.g., underfit) and/or attribute a pattern to random variation in the inputs (e.g., overfit). Underfitting or overfitting results in a less usable model. It would be preferable to represent the unique properties of the materials in the system with minimal sparsity and noise. Furthermore, models tend to be application specific. For example, they may only capture the relationships of X variables relevant to a Y variable for a particular application.

Summary of the Disclosure

[0003] The present disclosure relates to a representation model for polymeric systems using available data, referred to herein as a polymer representation learning model. The polymer representation learning model allows mapping to manufacturing conditions and identification of suitable materials for defined applications and improved or optimized application conditions, requiring only small datasets for the targeted task. The polymer representation learning model can use multiple data types as input and produce an embedding that captures aspects of a class of polymers contained within the input data. The polymer representation learning model can learn to fill in missing data (e.g., due to small or incomplete data sets) in the input data by training against other polymers within the class of polymers, thereby helping to mitigate the significant time and cost barrier to implementing some previous approaches to machine learning in the polymer characterization space that require relatively high levels of data quantity and/or quality without a tolerance for missing data. In contrast, some previous approaches could require acquisition of additional data and/or higher quality data prior to being able to implement effective machine learning. Embodiments described herein overcome such hurdles while also using much smaller data quantities than traditional machine learning approaches.

[0004] A polymer used in an application is not one molecule, but a distribution of many molecules of varying sizes, compositions, configurations, and architectures, each of which may contribute to the performance of the overall blend in different ways. Processes to produce polymers are tightly controlled to result in a set of defined distributions (e.g., with respect to molecular weight, chemistry, short chain branching, etc.). Small changes in molecular weight distributions, comonomer incorporation distributions, and branching structure can have significant impact on final product performance. The present disclosure provides a polymer representation learning model that can promote understanding how production of a polymer leads to specific properties of the polymer and/or application of the polymer. The polymer representation learning model can be used for inverse design to identify process conditions that can be used to make a polymer with specific properties or application of the polymer with specific results. The polymer representation learning model can use process information to describe material properties in its target application.

[0005] For many polymer systems, including commercial materials, the full structure-property relationships are not established for all application relevant behaviors, which is partly due to the significant impact of processing on performance. The polymer representation learning

model can enable domain-specific feature engineering to produce useful representations of classes of polymers and their applications by overcoming the hurdle posed by complex polymer structure-property relationships.

[0006] The above summary of the present disclosure is not intended to describe each disclosed embodiment or every implementation of the present disclosure. The description that follows more particularly exemplifies illustrative embodiments. In several places throughout the application, guidance is provided through lists of examples, which examples can be used in various combinations. In each instance, the recited list serves only as a representative group and should not be interpreted as an exclusive list.

Brief Description of the Drawings

[0007] Figure 1 is a block diagram illustrating inputs and outputs for a polymer representation learning model.

[0008] Figure 2 is a method flow diagram for determining a manufacturing condition for production of a particular polymer within a class of polymers.

[0009] Figure 3 is a method flow diagram for determining an application condition for application of a particular polymer within a class of polymers.

[0010] Figure 4 is a method flow diagram for providing a recommendation related to production of a particular polymer or application of the particular polymer.

[0011] Figure 5 is a block diagram illustrating training data including a representative dataset of resins used to develop a polymer representation learning model.

[0012] Figure 6 is a block diagram illustrating development of a polymer representation learning model.

[0013] Figure 7 is a graph illustrating an example reconstruction for molecular weight distribution.

[0014] Figure 8 is a block diagram illustrating prediction of data instances descriptive of an application of a respective polymer within a class of polymers.

[0015] Figure 9 is an example machine within which a set of instructions, for causing the machine to perform various methodologies discussed herein, can be executed.

Detailed Description

[0016] Some well-developed use cases of data-driven modeling in chemical design are around small molecule discovery (e.g., using the properties of an individual molecule to predict its performance in some system). “Properties of a molecule” can mean many things, and researchers have investigated representing molecules via their physical properties (e.g., boiling point, measured solubility, etc.), their electronic properties (e.g., density functional theory (DFT) descriptors), and their structures (e.g., simplified molecular-input-line-entry-system (SMILES) or graph representations of the chemical structure). Most of this data tends to be relatively dense continuous variables with dimensionality in the tens to hundreds. These problems are difficult but tractable for current machine learning algorithms owing to the fixed molecular weight of small molecules. In contrast, these methods of characterizing materials into a digital representation do not extend well to large molecules, such as polymers, which are used in commercial material design, partly due to the distributional nature of the molecular weight of polymeric systems.

[0017] The way that a material is represented in a machine learning model has a significant effect on the model’s ability to detect patterns between the material and its performance in an application. There are several challenges in describing materials, such as polymers, for modeling. For example, the distribution of molecules of varying sizes, compositions, configurations, and architectures that make up a polymer contribute to the performance of an overall blend as used in an application in different ways. Distributions are typically carefully designed for specific applications. Thus, attempts to represent a polymer by its average size (e.g., weight average molecular weight, number average molecular weight, etc.) are usually ineffective for more complex relationships. Further, such descriptors are often difficult to define for distributions of varying shape (e.g., branching distribution).

[0018] Another example of a challenge in describing materials is that for copolymers and three-dimensional polymers, such as branched or elastomeric networks, there can be many ways to assemble the correct number of monomers to match the molecule sizes within the distribution. The order of assembly can affect the material performance in an application. Representations of polymers that include an average empirical structure or a distribution of empirical structures of

varying sizes are still missing information about the polymer microstructure, such as if it is an alternating, block, or random structure.

[0019] Another example of a challenge in describing materials is that crystalline or segmental alignment size distribution and microstructure of polymers are often kinetically controlled, resulting in materials that are unique to their application specific manufacturing conditions. Such materials may be referred to as “product-by-process.” Thus, process information may also be valuable to understand what the structure of a material is for its use and performance in application.

[0020] Another example of a challenge in describing materials is that polymer blends or formulations are used to tailor properties for specific applications. Traditional polymer blends may be produced in the melt state, such as by co-extrusion of multiple materials to mix the melts together. However, some structures rely on the combination of materials after melting, such as in layered film structures (e.g., composed of up to ten layers). The blend composition is as important as the manufacturing conditions that were used to produce the formulations and structures. Small molecules added to systems for various purposes, such as antiblock or slip agents, can impact application processing, such as crystallization kinetics.

[0021] Another example of a challenge in describing materials is that aspects of the material that directly contribute to aspects of its performance may not be known. The assumption that every component of a material contributes directly to its performance is not always true. Due to the distributions in size, composition, configuration, and architecture, a minority portion of the distribution may drive the majority of the performance of a specific material property, such as is the case for high molecular weight tail in low density polyethylene for melt strength. This makes performing domain-specific feature engineering to produce useful representations challenging due to the complexity involved.

[0022] Another example of a challenge in describing materials is that there may be incomplete characterization (both molecular and performance related) or different characterization methods used on different polymers within a class of polymers. This can lead to large blocks of missing data. Due to data inconsistency and the production of a limited number of polymer samples, the sample size may be in the tens to hundreds versus a more complete dataset that would include, for example, tens or hundreds of thousands of samples.

[0023] Representation learning is a process in machine learning to automatically discover representations needed for feature detection or classification from raw data. Meaningful patterns can be extracted from raw data to create representations that are easier to understand and process than the raw data. These representations can be designed for interpretability, reveal hidden features, or be used for transfer learning. A latent space is an embedding of a set of items in which items resembling each other are positioned closer to one another. The dimensionality of the latent space is less than the feature space from which the data points are drawn. However, in some previous approaches, the latent space is specific to a particular application and may only capture the relationships of X variables that are relevant to a particular Y variable, for example. In contrast, the polymer representation learning model described herein can produce a latent space including representations relevant to all system-relevant relationships (e.g., to any Y variable). The latent space produced by the polymer representation learning model can be used for different Y variables without the model having to relearn the latent space. In this regard, the polymer representation learning model provides utility somewhat analogous to a foundational model, but without requiring vast data sets or intensive training typically associated with foundational models.

[0024] Artificial neural networks (ANNs) are networks that can process information by modeling a network of neurons, such as neurons in a human brain, to process information (e.g., stimuli) that has been sensed in a particular environment. Similar to a human brain, neural networks typically include a multiple neuron topology, which can be referred to as artificial neurons. An ANN operation refers to an operation that processes inputs using artificial neurons to perform a given task. The ANN operation may involve performing various machine learning algorithms to process the inputs. Example tasks that can be processed by performing ANN operations can include machine vision, speech recognition, machine translation, social network filtering, and medical diagnosis, among others. An ANN can perform machine learning tasks by forming probability weight associations between an input and an output. The probability weight associations can be provided by a plurality of nodes that comprise the ANN. The nodes together with weights, biases, embeddings, and/or activation functions can be used to generate an output of the ANN based on the input to the ANN. Nodes of the ANN can be grouped to form layers of the ANN.

[0025] Some proposals exist that attempt to address challenges describing the molecular composition of a polymer material. Simple approaches use defined features such as distributional moments or deconvolution, or features derived from approaches like functional data analysis. Another example is the use of machine learning via convolutional neural networks (CNNs) on analytical characterization data converted to images to capture the complete distribution of polymer sizes. A CNN is a regularized type of feed-forward network that learns feature engineering by itself via filters or kernel optimization. A CNN uses convolution layers, which help with efficiently capturing spatial relations. In contrast, the attention layers in transformer models are effective in capturing various sequential patterns that make them powerful for language and images. Attention is a machine learning method that determines the relative importance of each component in a sequence relative to the other components in that sequence. Attention encodes vectors called embeddings across a fixed-width sequence. A graph neural network (GNN) proposal includes polymer structure distribution as an ensemble of graphs. A GNN is an ANN for processing data that can be represented as a digital formulation graph. GNNs can use pairwise message passing such that graph nodes iteratively update their representations by exchanging information with neighboring nodes that are connected by an edge. Another GNN proposal is to capture periodicity in polymer structures, partially addressing the microstructure challenge. Various mechanisms have been proposed to adapt text string representations of structures from small molecules to polymers that include microstructure definition. Some proposals forego structure specification and describe the material based on its ingredients and process via GNN embeddings.

[0026] Each of these methods seeks to find a wholly comprehensive representation of a polymer as the source of the X data for predictive modeling (meaning that the representation is driven by the prediction target in supervised learning). However, many approaches require full knowledge of the exact chemical target structure and/or structure distribution, which is often not known in commercial use of polymers. Thus, achieving data quality necessary to use these methods is a significant time and cost barrier to implementation.

[0027] A prior approach using multiple types of input polymer data, where monomer structures are incorporated as graphs then combined with numerical descriptors of the material or its process uses supervised learning to produce an embedding representation of a polymer that

represents the product-by-process. However, it is unclear how generalizable such supervised learning-based representations are to new tasks.

[0028] At least one embodiment described herein addresses the above and other deficiencies by combining available data in a multimodal machine learning algorithm to learn a vector representation of a material class, rather than attempting to find one description or characterization that includes all aspects of polymer properties. Many previous approaches can produce representations of the material, but because they use supervised learning the embeddings are learned by reducing prediction error relative to the specified labeled properties (Y values). Only the representations of the aspects of the material that are relevant to that particular property are learned. For the digital representation of the material to be the most generalizable and thus more useful in future modeling problems, an unsupervised approach would be desirable. Some embodiments may make use of an ANN, such as a CNN, GNN, autoencoder, transformer, etc.. An autoencoder is a type of ANN used to learn efficient encodings (embeddings) of unlabeled data via unsupervised learning. The autoencoder learns an encoding function that transforms the input data to a lower dimensional embedding and a decoding function that recreates the input data from the encoded representation, and is useful for dimensionality reduction.

[0029] As used herein, the singular forms “a”, “an”, and “the” include singular and plural referents unless the content clearly dictates otherwise. Furthermore, the word “may” is used throughout this application in a permissive sense (i.e., having the potential to, being able to), not in a mandatory sense (i.e., must). The term “include,” and derivations thereof, mean “including, but not limited to.” The term “coupled” means directly or indirectly connected and, unless stated otherwise, can include a wireless connection.

[0030] The figures herein follow a numbering convention in which the first digit or digits correspond to the drawing figure number and the remaining digits identify an element or component in the drawing. Similar elements or components between different figures may be identified by the use of similar digits. For example, 102 may reference element “02” in Figure 1, and a similar element may be referenced as 602 in Figure 6. Analogous elements between different figures may be referenced with a hyphen and extra numeral or letter. See, for example, elements 104-1, 104-2, and 104-3 in Figure 1. Such analogous elements may be generally referenced without the hyphen and extra numeral or letter. For example, elements 104-1, 104-2, and 104-3 may be collectively referenced as 104. As will be appreciated, elements shown in the

various embodiments herein can be added, exchanged, and/or eliminated so as to provide a number of additional embodiments of the present disclosure. In addition, as will be appreciated, the proportion and the relative scale of the elements provided in the figures are intended to illustrate certain embodiments of the present invention and should not be taken in a limiting sense.

[0031] Figure 1 is a block diagram illustrating inputs and outputs for a polymer representation learning model 102. The inputs and outputs can include first data instances 104, each descriptive of production of a respective polymer within the class of polymers. First data instances 104 are related to how the polymer is made. Examples of first data instances 104 include data describing molecules 104-1 within the polymer, data describing additional chemicals 104-2 used to produce the polymer, and manufacturing conditions 104-3, etc. The data describing molecules 104-1 within the polymer can include data describing monomers, repeat units, polymeric constitutional units derived from monomers, initiators, chain transfer agents, etc. The data describing molecules 104-1 can cover molecular, oligomeric, and/or polymeric species, chemical descriptors, etc. that exist within the molecular structures that make up the polymeric material included in the final product. The data describing additional chemicals 104-2 can include catalysts, solvents, processing aids, etc. The data describing additional chemicals 104-2 can be defined as molecular, oligomeric, and/or polymeric species, chemical descriptors, etc. that aid in the production of the polymeric material but are not necessarily intended to be included in the final product. In some examples, molecular, oligomeric, and/or polymeric species can have structures defined with SMILES, BigSMILES, etc. Examples of the data describing manufacturing conditions 104-3 include flowsheet and unit operation design details, time, and the key performance indicators for each relevant unit operation such as temperatures, pressures, flows, rotational speeds, measured and controlled variables, etc. In some embodiments, the first data instances 104 can be featurized to reduce their dimensionality (e.g. per unit operation or plant section). In some embodiments, the first data instances 104 can be raw data. The first data instances 104 can help inform the imputation of missing data (described in more detail below) used to build the polymer representation learning model 102 and the polymer representation learning model 102 can map the manufacturing conditions 104-3 used to make a particular polymer.

[0032] The inputs and outputs for the polymer representation learning model 102 can include second data instances 106, each descriptive of a respective polymer within a class of polymers. As used herein, a data instance is one data set (e.g., one chromatogram) for the specific instance of a polymer. Second data instances 106 describe characteristics of the polymer itself. Second data instances 106 can include physical measurements, which can be generated experimentally. For example, second data instances 106 can include characterization data 106-1 of the polymer and properties 106-2 of the polymer under standard conditions. Non-limiting examples of characterization data 106-1 include molecular weight distributions, infrared (IR) spectra, comonomer distribution data, rheology data, crystallization image data, etc. Second data instances 106 can also include machine learning descriptors 106-3 of the polymer, which can be generated computationally, experimentally, or through application of domain knowledge. Examples of second data instances 106 include time series data, various distributions, point values, properties under standard conditions (e.g., American Society for Testing and Materials (ASTM) defined standards, tensile properties), etc.

[0033] The inputs and outputs for the polymer representation learning model 102 can include third data instances 108, each descriptive of an application of a respective polymer within the class of polymers. The third data instances 108 describe the application space of the polymer. Examples of third data instances 108 include formulation components 108-2, additive descriptors 108-3, test conditions 108-5, application conditions 108-1, application performance properties 108-4, etc. The third data instances 108 describe the conditions and methods used to produce specific forms of a final article using one or more polymers and the associated performance properties of the final article. Examples of such production methods include film extrusion or casting, injection molding, blow molding for films and solid articles, etc. In some embodiments, the polymer representation learning model 102 may receive or use only a subset of the third data instances 108 for specific applications of final articles made with a specific class of polymers. For example, low-density polyethylene does not typically enter the same applications as linear-low-density polyethylene, but the chemical makeup and structural elements are sufficiently similar to warrant use of a common polymer representation learning model 102. The third data instances 108 can help inform the imputation of the molecular characterization 106-1 and the polymer representation learning model 102 can use the third data instances 108 to

optimize application conditions 108-1 for specified materials or identify suitable molecular architectures for specific applications.

[0034] The first data instances 104, second data instances 106, and third data instances 108 can include multiple different data types, making the polymer representation learning model 102 multimodal. Examples of data types include numerical data, text data, image data, X-Y series data, graph data, etc. Data types can have varying data sources as there are many ways to collect data that describes a polymer. Data types can include numerical or text data (e.g., bulk physical properties, BigSMILES, PolyGrammar, etc.), images (e.g., microscopy, chromatography / separations, spectroscopy, etc.), paired X-Y series data (e.g., rheology data such as linear and extensional viscosities as a function of shear rates and temperatures), a structure graph, a formulation graph, etc. Some individual data instances may belong to more than one data type or be capable of being categorized as more than one data type. BigSMILES is an extension of SMILES that aims to provide an efficient representation system for macromolecules. PolyGrammar is a parametric, context-sensitive grammar designed specifically for polymers. PolyGrammar uses a symbolic hypergraph representation and simple production rules to represent and generate valid polyurethane structures. Each data type can capture some information about the material and leave some behind. For a generalizable representation of the material to be learned, independent of which aspects affect which performance variables, that data which is available can be used by the polymer representation learning model 102 to characterize the material. In some embodiments, the polymer representation learning model 102 can select which data is to be used for a particular purpose.

[0035] Some previous approaches operate with data dimensionality in the hundreds. Using all available raw data describing a polymer could amount to dimensionality of polymer input data in the thousands to hundreds of thousands of variables. In order to reduce the dimensionality of the representation to make the data tractable for everyday use and to avoid model overfitting and underfitting, it is advantageous to apply representation learning by the polymer representation learning model 102. Representation learning is also known as feature learning. Representation learning is the auto discovery of representations needed for feature detection and/or classification from raw data. The polymer representation learning model 102 can be operated to learn an embedding for the class of polymers based on the first data instances 104, the second data instances 106, and the third data instances 108. This will produce a lower

dimensionality embedding that sufficiently captures the aspects contained within the original data.

[0036] The polymer representation learning model 102 can conduct the lower dimensionality embedding either hierarchically or after fusion of the data sources. Data fusion refers to integrating multiple data sources to produce more consistent, accurate, and useful information than that provided by an individual data source. One example of data fusion is concatenation, but methods used may include any association, state-estimation, or decision-based fusion approach. The polymer representation learning model 102 can conduct the lower dimensionality embedding hierarchically by learning a respective embedding for each of the data types and fusing the respective embeddings to the lower dimensionality embedding for the class of polymers. Alternatively, the polymer representation learning model 102 can conduct the lower dimensionality embedding by fusing the first data instances 104 and at least one of the second data instances 106 and the third data instances 108 before learning the embedding. The polymer representation learning model 102 can select the most promising approach for a given system. For example, the polymer representation learning model 102 can determine, based on the class of polymers, whether to fuse the first data instances 104, the second data instances 106, and/or third data instances 108 (or any subset thereof) and learn the embedding for the class of polymers or learn a respective embedding for each of the data types and fuse the respective embeddings to the embedding for the class of polymers.

[0037] As another example, the polymer representation learning model 102 can learn the embedding by a hybrid approach. The polymer representation learning model 102 can fuse a first portion of at least two of the first data instances 104, the second data instances 106, and third data instances 108 and learn a first intermediate embedding. The term “portion” of the various available data instances refers to less than all of the various available data instances. For example, if ten first data instances 104 and ten second data instances 106 are available, a first portion could include five first data instances 104 and three second data instances 106. In this context, “portion” does not mean, for example, all of the first data instances 104 to the exclusion of any second data instances 106. The polymer representation learning model 102 can learn a second intermediate embedding for a second portion of the at least two of the first data instances 104, the second data instances 106, and the third data instances 108. Intermediate embeddings

may be learned for any quantity of portions. The polymer representation learning model 102 can fuse the intermediate embeddings to the embedding for the class of polymers.

[0038] The polymer representation learning model 102 learns the embedding with the use of the data inputs of varying types, thereby not leaving behind perspective on the material. The polymer representation learning model 102 is a representation model for any class of polymers. A class of polymers includes those polymers that are similar enough to be measured by the same set of analytical techniques. Missing data and/or small data sets may pose a challenge for training the polymer representation learning model 102. Missing data can refer to a missing data instance for a particular polymer. For example, polymer A has rheology data but no chromatogram data, so the chromatogram data is considered to be missing data. Missing data can refer to missing data within a data instance for a particular polymer. For example, polymer A has an incomplete X-Y series data set, thereby constituting missing data. In the commercial use of polymers, it is common for there to be missing data related to the polymers.

[0039] To address the issue of missing data and/or small data sets, the polymer representation learning model 102 can use self-supervised learning, semi-supervised learning, and/or unsupervised learning. Supervised learning involves training with labeled data. Unsupervised learning involves training without labeled data. An unsupervised learning task is one that models the underlying structure of the data without explicit labels (Y data) for each sample. Such methods can identify previously unknown patterns or features in the data. Some examples of methods for unsupervised learning include principal component analysis (PCA) and autoencoder ANNs, which are dimension reduction techniques that emphasizes data variance, clustering algorithms such as k-means, similarity analyses such as k-nearest neighbor, association rules mining, and anomaly detection. Semi-supervised learning involves training without labeled data, but input-label pairs can be constructed from each data point. Large scale models, amongst them most foundation models, are trained using self-supervised methods where the model structure identifies the learning pattern as part of the training process. The polymer representation learning model 102 can use polymer informatics to generate random polymers for unsupervised learning and/or as a prior for optimization for polymer design. The polymer representation learning model 102 can learn to fill in missing sections of data for a particular polymer by training against other polymers within the class of polymers to which the particular polymer belongs. Different polymers within the class of polymers may share data blocks. The

polymer representation learning model 102 can impute data for missing data for a particular polymer based on other data instances within the class of polymers. The polymer representation learning model 102 can impute missing data for a particular polymer within the same data instance provided there is some non-missing data for the data instance. The polymer representation learning model 102 can be trained with the generated or imputed data, the first data instances 104, the second data instances 106, and the third data instances 108 to tune the embedding for the class of polymers.

[0040] Data descriptive of manufacturing conditions 104-3 and application performance properties 108-4 can be used to impute characterization of the data collected for a class of polymers to enable training of the polymer representation learning model 102. A subset of the data describing molecules within the polymer 104-1, additional chemicals 104-2, manufacturing conditions 104-3, application performance properties 108-4, and characterization 106-1 can be used to impute information not in the subset.

[0041] The polymer representation learning model 102 can provide a recommendation related to production of a particular polymer within the class of polymers or application of the particular polymer based on the embedding for the class of polymers and the first data instances 104, the second data instances 106, and the third data instances 108. The recommendation can include a determination of at least one of a manufacturing condition 104-3, a monomer, and a catalyst for production the particular polymer, for which a corresponding first data instance 104 is not available. The recommendation can include a determination of application conditions 108-1 for a specific application of the particular polymer, for which a corresponding third data instance 108 is not available. The recommendation can include a determination of application performance properties for the specific application of the particular polymer. The recommendation can include a determination of at least one of a formulation component 108-2 and an additive descriptor 108-3 for a specific application of the particular polymer, for which a corresponding third data instance 108 is not available.

[0042] The polymer representation learning model 102 can be used for different classes of polymers by receiving and operating on different data instances, each descriptive of a respective different polymer within a different class of polymers. In some embodiments, the polymer representation learning model 102 can impute data for missing data for the different particular polymer based on other data instances within the class of polymers.

[0043] A test case was conducted for creating a generalized representation of complex polymer formulation, which can be used to create possible standardized descriptors of a polymeric system. The representation uses a PolyDAT schema (a JavaScript Object Notation (JSON) file) designed to capture manufacturing conditions 104-3, characterization data 106-1, and structural information of polymers and small molecules, with polymer species represented by a BigSMILES string and small molecules represented by a SMILES string. The PolyDAT schema is a universal embedding of polymeric product formulation, albeit in a very high-dimensional space with a benefit of human readability. This schema can then be used to create a variety of descriptors in the form of numerical values that are ready to be used by a machine learning model for feature extraction or property prediction.

[0044] Figure 2 is a method flow diagram for determining a manufacturing condition for production of a particular polymer within a class of polymers. The methods described herein may be performed, in some examples, using a computing system such as those described with respect to Figure 9. The methods can be performed by processing logic that can include hardware (e.g., processing device, circuitry, dedicated logic, programmable logic, microcode, hardware of a device, integrated circuit, etc.), software (e.g., instructions run or executed on a processing device), or a combination thereof. Although shown in a particular sequence or order, unless otherwise specified, the order of the processes can be modified. Thus, the illustrated embodiments should be understood only as examples, and the illustrated processes can be performed in a different order, and some processes can be performed in parallel. Additionally, one or more processes can be omitted in various embodiments. Thus, not all processes are required in every embodiment. Other process flows are possible.

[0045] As illustrated at 210, the method can include receiving a plurality of first data instances, each descriptive of production of a respective polymer within a class of polymers, by the polymer representation learning model. As illustrated at 212, the method can include receiving a plurality of second data instances, each descriptive of a respective polymer within the class of polymers, by a polymer representation learning model. The plurality of second data instances can include experimentally generated and computationally generated data. As illustrated at 214, the method can include receiving a plurality of third data instances, each descriptive of application of a respective polymer within the class of polymers, by the polymer

representation learning model. The first, second, and third data instances include a plurality of different data types.

[0046] Although not specifically illustrated, the method can include receiving at least two of a group of three types of data instances. The three types of data instances are the first data instances, the second data instances, and the third data instances. With reference back to Figure 1, in some cases data of all three instance types (first data instances 104, second data instances 106, and third data instances 108) may not be available. For example, third data instances 108 may not be available and therefore the representation learning model 102 can operate based on receiving first data instances 104 and second data instances 106. As another example, second data instances 106 may not be available and therefore the representation learning model 102 can operate based on receiving first data instances 104 and third data instances 108.

[0047] As illustrated at 216, the method can include conducting representation learning by the polymer representation learning model to learn an embedding for the class of polymers based on at least two of the first, second, and third pluralities of data instances. Conducting representation learning to learn the embedding can include determining, by the polymer representation learning model based on the class of polymers, a sequence by which to fuse the at least two of the first, second, and third pluralities of data instances and learning the embedding according to the determination.

[0048] As illustrated at 218, the method can include determining at least one manufacturing condition and any other molecules or additional chemicals for production of the particular polymer, for which a corresponding first data instance is not available based on the embedding for the class of polymers and the at least two of the first, second, and third pluralities of data instances. Although not specifically illustrated, the method can include imputing, by the polymer representation learning model, imputed data for missing data for a particular polymer based on other data instances within the class of polymers. The method can include training the polymer representation learning model with the imputed data and the at least two of the first, second, and third pluralities of data instances to tune the embedding for the class of polymers.

[0049] Figure 3 is a method flow diagram for determining an application condition for application of a particular polymer within a class of polymers. As illustrated at 320, the method can include receiving a plurality of first data instances, each descriptive of production of a respective polymer within a class of polymers, by the polymer representation learning model.

[0050] As illustrated at 322, the method can include receiving a plurality of second data instances, each descriptive of a respective polymer within the class of polymers, by a polymer representation learning model. As illustrated at 324, the method can include receiving a plurality of third data instances, each descriptive of application of a respective polymer within the class of polymers, by the polymer representation learning model. The first, second, and third data instances include a plurality of different data types. Although not specifically illustrated, the method can include receiving at least two of a group of three types of data instances as described above with respect to Figure 2.

[0051] As illustrated at 326, the method can include conducting representation learning by the polymer representation learning model to learn an embedding for the class of polymers based on the at least two of the first, second, and third pluralities of data instances. Conducting representation learning to learn the embedding can include determining, by the polymer representation learning model based on the class of polymers, a sequence by which to fuse the at least two of the first, second, and third pluralities of data instances and learning the embedding according to the determination.

[0052] As illustrated at 328, the method can include determining at least one application condition and any formulation components and additives for a specific application of the particular polymer, for which a corresponding third data instance is not available. Although not specifically illustrated, the method can include imputing, by the polymer representation learning model, imputed data for missing data for a particular polymer based on other data instances type within the class of polymers. The method can include training the polymer representation learning model with the imputed data and the at least two of the first, second, and third pluralities of data instances to tune the embedding for the class of polymers.

[0053] Figure 4 is a method flow diagram for providing a recommendation related to production of a particular polymer or application of the particular polymer. As illustrated at 430, the method can include receiving, by a polymer representation learning model, a plurality of first data instances, each descriptive of production of a respective polymer within a class of polymers.

[0054] As illustrated at 432, the method can include receiving, by the polymer representation learning model, at least one of a group of two types of data instances. As illustrated at 434, the group of two types of data instances includes a plurality of second data instances, each descriptive of a respective polymer within the class of polymers. As illustrated at

436, the group of two types of data instances includes a plurality of third data instances, each descriptive of application of a respective polymer within the class of polymers. The first, second, and third data instances include a plurality of different data types. Examples of the different data types include numerical data, images, paired X-Y series data, graphs, and text data. In some embodiments, the plurality of different data types include at least two data types from this list. In some embodiments, the plurality of different data types include at least three data types from this list.

[0055] As illustrated at 438, the method can include operating the polymer representation learning model to learn an embedding for the class of polymers based on the first and the at least one of the second and third pluralities of data instances. Learning the embeddings can include masking a specific data instance and minimizing reconstruction loss of the specific data instance as described in more detail with respect to Figure 6.

[0056] In some embodiments, the first plurality of data instances and the at least one of the second and third pluralities of data instances can be fused to learn the embedding for the class of polymers. In such embodiments, the embeddings are learned after the input data instances are fused. In some embodiments, a respective embedding for the first plurality of data instances and the at least one of the second and third pluralities of data instances can be learned. Then, the respective embeddings can be fused to the embedding for the class of polymers. In such embodiments, the embeddings are learned for the different data instance types and then the embeddings are fused. The order of fusion and embedding can be case dependent. The modalities used may affect which order provides a better architecture. Numeric-based modalities such as scalar and paired series can be fused raw, but a data instance that is an image cannot be meaningfully fused with the numeric-based modalities. In such an example, embedding can be performed prior to fusion.

[0057] In some embodiments, a determination can be made based on the class of polymers under consideration, whether to fuse the data instances before learning the embeddings or to learn the embeddings for each data instance and then fuse the embeddings. More specifically, a determination can be made whether to fuse the first plurality of data instances and the at least one of the second and third pluralities of data instances and learn the embedding for the class of polymers or to learn a respective embedding for each of the first plurality of data instances and the at least one of the second and third pluralities of data instances and fuse the

respective embeddings to the embedding for the class of polymers. The determination can be made based on the data schema for a class of polymers to accommodate the modalities available to a particular class. In some embodiments, the determination can be loss driven in the application (e.g., what fits the data better).

[0058] In some embodiments, learning the embedding can include fusing a first portion of the first plurality of data instances and the at least one of the second and third pluralities of data instances and learning a first intermediate embedding, learning a second intermediate embedding for a second portion of the first plurality of data instances and the at least one of the second and third pluralities of data instances, and fusing the first and the second intermediate embeddings to the embedding for the class of polymers. Such embodiments may be useful, for example, for cases in which different portions of the data instances become available at different times. For example, the model can operate with the first intermediate embedding until the second portion becomes available and then learn the embedding for the second portion, fuse the embeddings, and operate with the fused embeddings going forward. Such embodiments may be useful when the first portion of the data instances are more closely related to each other than they are to the second portion of the data instances. In such an example, the result of fusing the different intermediate embeddings may be more meaningful (may allow the model to produce more meaningful results) than fusing the portions of the data instances before learning the embedding. In other words, better embeddings may be learned for the more similar portions of the data instances, ultimately yielding a better overall embedding for the class of polymers.

[0059] As illustrated at 439, the method can include providing a recommendation related to production or application of a particular polymer within the class of polymers based on the embedding for the class of polymers, the first plurality of data instances, and the at least one of the second and third pluralities of data instances. Providing the recommendation related to production or application of the particular polymer can include imputing data from the polymer representation learning model for missing data for a particular polymer based on other data instances within the class of polymers. For example, first data instances and second data instances are available for a specific polymer, except for the molecular weight distribution. The polymer representation learning model can impute the molecular weight distribution of the particular polymer (based on the embeddings and the other data instances as described herein) and provide the imputed molecular weight distribution. As a related example, the

recommendation related to production of the particular polymer can be a recommendation to change an input to the production of the particular polymer in order to achieve a more desired molecular weight for the particular polymer as result of its production. Providing the recommendation related to production of the particular polymer can include determining at least one of a manufacturing condition, a monomer, and a catalyst for production the particular polymer, for which a corresponding first data instance is not available.

[0060] An example of the recommendation related to application of the particular polymer can be a prediction of its performance in a blown film (e.g., a characterization of the resulting mechanical and optical performance in the specific application) based on the embeddings and other data instances related to application performance of other polymers within the class. The polymer representation learning model can be trained with the imputed data (the data that was imputed by the model), the first plurality of data instances, and the at least one of the second and third pluralities of data instances to tune the embedding for the class of polymers. Providing the recommendation related to application of the particular polymer can include determining at least one of an application condition, any formulation components, and any additives for a specific application of the particular polymer, for which a corresponding third data instance is not available. Providing the recommendation related to application of the particular polymer can include determining application performance properties for the specific application of the particular polymer.

[0061] Figures 5-8 pertain to an example implementation of at least one embodiment. Figure 5 is a block diagram illustrating training data consisting of a representative dataset of resins used to develop a polymer representation learning model 502. By way of a non-limiting example, a polymer representation learning model 502 was developed for the purpose of imputation (of missing first data instances 504-2 and/or second data instances 506-2) using available first data instances 504-1 and available second data instances 506-1 for a representative dataset of polyethylene resins. With respect to Figure 5, Figure 6, and Figure 8, the designation “-1” in the reference numerals indicates actual (input) data, while the designation “-2” indicates imputed or predicted (output) data from the model. Robust imputation can be useful for preserving historical data upon changes in measurement technology and/or for data not being captured during material generation. Imputation can also be helpful for efficient material characterization. The polymer representation learning model can include a fusion module that

fuses various data sources (e.g., first data instances 504-1 and second data instances 506-1) and learns a multimodal representation 548.

[0062] In this example, the first data instances 504 capture resin manufacturing information in the form of the formulation for a multiple reactor set-up. Referring back to Figure 1, resin manufacturing information is an example of additional chemicals 104-2. This formulation may include the respective contribution of each reactor component to the overall resin and for each component, the catalyst molecular structure allowing for multi-catalyst systems, the target melt index (I2), and the target density. Reactors can be qualified by producing products that are characterized to be sure that they are up to specification. The second data instances 506 include a plurality of X-Y paired series including molecular weight and comonomer content distributions in addition to a plurality of scalar numeric bulk resin properties including melt index and density. Gel permeation chromatography (GPC) is an example of size-exclusion chromatography that separates high molecular weight or colloidal analytes on the basis of size or diameter. GPC can provide a quantitative molecular weight distribution (MWD) of a polymer sample. The resulting GPC chromatogram represents a weight distribution of the polymer as a function of retention volume. Molecular weight distribution refers to the variation in molecular weights of polymer chains within a sample. MWD can be used to predict many physical properties of polymeric materials. The properties of a polymer are closely related to short-chain branching distribution (SCBD), which refers to the arrangement and frequency of short branches along the main polymer chain in materials like polyethylene. The SCBD influences the physical properties and performance of the polymer, affecting aspects such as crystallization and mechanical strength.

[0063] Figure 6 is a block diagram illustrating development of a polymer representation learning model 602. Each data instance may also be referred to as a mode. The primary modes (including first data instances 604 and second data instances 606) are defined such that for a single polymer it is assumed that the entire mode is either missing or non-missing for training. In some instances, the data instances are defined based on practical considerations (e.g., the molecular weight distribution comes from a single test measurement) thus the data will either be missing or non-missing in its entirety. In other cases, the modes are defined to simplify the model training due to the small training set size. The general properties (“Gen Prop”) containing melt index and density is one such example.

[0064] For each of the modes, an individual representation (mode-representation) can be generated (e.g., if the mode requires it) by mode encoders 640. The mode encoders 640 can generate embeddings, reduce the dimensionality of the data instances, and/or can convert the data instances to vectors (as illustrated in Figure 5). Example approaches include non-negative matrix factorization, autoencoders, or a mathematical operation that provides a reconstructable output with a smaller dimension from a provided input. Similarly, for first data instances, such as catalyst molecular structures, mode-representations can use either molecular one-hot encoding (OHE) or another descriptor generating method (e.g., predicted properties from SMILES text or embeddings from molecular graphs or images). These mode-representations are then fused via concatenation or other methods into a fused latent space representation 644, which may use additional mathematical operations (like multi-head attention 646) to maximize the ability of the multimodal representation 648 to reconstruct the mode-representations. The fused latent space representation 644 is the fused input vector (either raw or embedded), which combines mode representations. The multi-head attention 646 is an example of a machine learning model that identifies the important sequences for reconstruction of the input latent spaces. The output of the multi-head attention 646 is the multimodal representation 648. The multimodal representation 648 may be learned using self-supervised training via masking 650 of one or more modalities and minimizing the reconstruction losses 656 of original or embedded data instances as illustrated at 652. At 652, the multimodal representation 648 can be concatenated with an OHE vector (e.g., the mode indicator for which the latent space is to be generated). To perform imputation using this multimodal representation 648, a translation model 654 decodes (e.g., via mode decoders 642) the embeddings to produce predicted mode-representations for the provided input data (data instances 604-1, 606-1) from available data instances and outputs the imputed data instances 604-2, 606-2. The translation model, in this example, includes fully connected layers for transformation. The original data format can be recreated from the mode-representation. For catalyst selection, the mode-representation can be used to identify a preferred candidate from a number of available catalysts through a nearest-neighbor search, or the mode-representation can be used in a generative manner to identify new catalyst candidates.

[0065] For this specific example, data were imputed for each training mode to demonstrate model performance. Results are summarized in Table 1, where “Prod. Param.” means production parameters, “Catalyst Ident.” means catalyst identification, RMSE is root

mean squared error, and “Orig. Magnitude” means original magnitude. For scalar data such as melt index, density, and the split weight percent, the relative errors from test samples are 0.20 on scaled input data. The error increases to an RMSE of about 10 for data reconstructed to the original space, which is likely due to an imbalance in the training data. The relative error for reconstructed molecular weight distribution and short chain branching distribution is 0.07 and 0.60, respectively, yielding very good imputation predictions. The catalyst identification is almost 70% for top 3 accuracy and 88% for top 5 accuracy. Lastly, for X-Y paired series data, graphical assessment of curve reconstruction is used in lieu of objective metric like pointwise-mean squared error. Results exceed expectations given the limited training data set used of less than one hundred samples.

Table 1

	RMSE – Orig. Magnitude		RMSE – Scaled Inputs		Top 3 Accuracy	Top 5 Accuracy
	Training	Test	Training	Test		
MWD	0.058	0.067	0.102	0.132		
SCBD	0.624	0.597	0.146	0.173		
I2, Density	10.892	10.595	0.152	0.204		
Prod. Param.	1.993	2.434	0.158	0.209		
Catalyst Ident.					68%	88%

[0066] For one specific polymer, the polymer representation learning model 602 was operated to recommend alternative first data instances 604-2, namely the catalysts used to produce the polymer, in addition to imputing missing second data instances 606-2 such as the polymer density and molecular weight distribution. In both use cases the known catalysts, molecular weight distribution, and density are treated as missing data (unseen by the polymer representation learning model). In this specific example denote the true catalyst pair as “Catalyst 1” and “Catalyst 2”. The polymer representation learning model 602 returns the top 3 closest catalysts from a catalyst portfolio list. Without loss of generality, the model predictions for “Catalyst 1” include [“Catalyst 3”, “Catalyst 1”, “Catalyst 4”] in rank order and for “Catalyst 2” the predictions include [“Catalyst 2”, “Catalyst 5”, “Catalyst 6”]. Therefore, a reasonable alternative catalyst pair recommendation is “Catalyst 3” and “Catalyst 5” due to their proximity to the predicted catalyst embeddings.

[0067] Figure 7 is a graph illustrating an example reconstruction for molecular weight distribution using the fused representation (multimodal representation 648 as illustrated in Figure 6) for imputation provided data with all modalities except molecular weight distribution. The expression $d(W_f)/d(\log(M))$ along the vertical axis of the graph refers to the differential weight distribution function (W_f) in terms of the logarithm of molar mass (M), which is along the horizontal axis of the graph. This function provides insights into how the weight of molecules is distributed across different molar masses in a sample, which is useful for understanding the characteristics of polymers. Regarding the imputation case, the reconstruction for molecular weight distribution is illustrated in Figure 7 and the imputed density value 0.926 grams per cubic centimeter (g/cc) closely matches the true value of 0.922 g/cc.

[0068] Figure 8 is a block diagram illustrating prediction of data instances descriptive of an application of a respective polymer within a class of polymers. Figure 8 is similar to Figure 5, except that the input the polymer representation learning model 802 includes first data instances 804-1, second data instances 806-1, and third data instances 808-1 while the output is imputed third data instances 808-2 (data descriptive of an application of a respective polymer) for a representative dataset of polyethylene resins. In this example, the first data instances 804 capture resin manufacturing information in the form of the formulation for a multiple reactor set-up. This formulation may include the respective contribution of each reactor component to the overall resin and for each component, the catalyst molecular structure allowing for multi-catalyst systems, the target melt index (I2), and the target density. The second data instances 806 include a plurality of X-Y paired series including molecular weight and comonomer content distributions in addition to a plurality of scalar numeric bulk resin properties including melt index and density.

[0069] The input third data instances 808-1 include, for example, application conditions for “Line A” 808-1A and application conditions for “Line B” 808-1B, which describe application conditions when the particular polymer (e.g., polyethylene resin) is being used for a particular application, such as a blown film. In addition to the benefits described above, prediction using the learned multimodal representation 848 can provide final performance properties of the polymer with high data efficiency. The output third data instances 808-2 can include polymer application performance data such as dart drop impact, tear strength, tensile strength, haze, and gloss for the applied polymer in each of Line A and Line B.

[0070] Figure 9 is an example machine 990 within which a set of instructions 999, for causing the machine 990 to perform various methodologies discussed herein, can be executed. The machine 990 can be connected (e.g., networked) to other machines in a LAN, an intranet, an extranet, and/or the Internet. The machine 990 can operate in the capacity of a server or a client machine in client-server network environment, as a peer machine in a peer-to-peer (or distributed) network environment, or as a server or a client machine in a cloud computing infrastructure or environment.

[0071] The machine 990 can be a personal computer (PC), a tablet PC, a set-top box (STB), a Personal Digital Assistant (PDA), a cellular telephone, a web appliance, a server, a network router, a switch or bridge, or any machine capable of executing a set of instructions (sequential or otherwise) that specify actions to be taken by that machine. Further, while a single machine 990 is illustrated, the term “machine” shall also be taken to include any collection of machines that individually or jointly execute a set (or multiple sets) of instructions to perform any one or more of the methodologies discussed herein.

[0072] The example machine 990 includes a processing device 991, a main memory 992 (e.g., read-only memory (ROM), flash memory, dynamic random access memory (DRAM) such as synchronous DRAM (SDRAM) or Rambus DRAM (RDRAM), etc.), a static memory 993 (e.g., flash memory, static random access memory (SRAM), etc.), and a data storage system 994, which communicate with each other via a bus 995.

[0073] The processing device 991 represents one or more general-purpose processing devices such as a microprocessor, a central processing unit (CPU), or the like. More particularly, the processing device can be a complex instruction set computing (CISC) microprocessor, reduced instruction set computing (RISC) microprocessor, very long instruction word (VLIW) microprocessor, or a processor implementing other instruction sets, or processors implementing a combination of instruction sets. The processing device 991 can also be one or more special-purpose processing devices such as an application specific integrated circuit (ASIC), a field programmable gate array (FPGA), a digital signal processor (DSP), network processor, or the like. The processing device 991 is configured to execute instructions 999 for performing the operations and steps discussed herein. The machine 990 can further include a network interface device 996 to communicate over the network 997.

[0074] The data storage system 994 can include a machine-readable storage medium 998 (also known as a computer-readable medium) on which is stored one or more sets of instructions 999 or software embodying any one or more of the methodologies or functions described herein. The instructions 999 can also reside, completely or at least partially, within the main memory 992 and/or within the processing device 991 during execution thereof by the machine 990, the main memory 992 and the processing device 991 also constituting machine-readable storage media.

[0075] In some embodiments, the instructions 999 include instructions to implement functionality corresponding to the polymer representation learning model described herein. While the machine-readable storage medium 998 is shown in an example embodiment to be a single medium, the term “machine-readable storage medium” should be taken to include a single medium or multiple media that store the one or more sets of instructions. The term “machine-readable storage medium” shall also be taken to include any medium that is capable of storing or encoding a set of instructions for execution by the machine and that cause the machine to perform any one or more of the methodologies of the present disclosure. The term “machine-readable storage medium” shall accordingly be taken to include, but not be limited to, solid-state memories, optical media, and magnetic media, whether provided in a local or distributed manner (e.g., cloud storage).

[0076] Although specific embodiments have been described above, these embodiments are not intended to limit the scope of the present disclosure, even where only a single embodiment is described with respect to a particular feature. Examples of features provided in the disclosure are intended to be illustrative rather than restrictive unless stated otherwise. The above description is intended to cover such alternatives, modifications, and equivalents as would be apparent to a person skilled in the art having the benefit of this disclosure.

[0077] The scope of the present disclosure includes any feature or combination of features disclosed herein (either explicitly or implicitly), or any generalization thereof, whether or not it mitigates any or all of the problems addressed herein. Various advantages of the present disclosure have been described herein, but embodiments may provide some, all, or none of such advantages, or may provide other advantages.

[0078] In the foregoing Detailed Description, some features are grouped together in a single embodiment for the purpose of streamlining the disclosure. This method of disclosure is

not to be interpreted as reflecting an intention that the disclosed embodiments have to use more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive subject matter lies in less than all features of a single disclosed embodiment. Thus, the following claims are hereby incorporated into the Detailed Description, with each claim standing on its own as a separate embodiment.

Claims

What is claimed is:

1. A computer-implemented method, comprising:
 - receiving, by a polymer representation learning model, a plurality of first data instances, each descriptive of production of a respective polymer within a class of polymers;
 - receiving, by the polymer representation learning model, at least one of a group of two types of data instances comprising:
 - a plurality of second data instances, each descriptive of a respective polymer within the class of polymers; and
 - a plurality of third data instances, each descriptive of application of a respective polymer within the class of polymers;
 - wherein the first, second, and third data instances include a plurality of different data types;
 - operating the polymer representation learning model to learn an embedding for the class of polymers based on the first and the at least one of the second and third pluralities of data instances; and
 - providing a recommendation related to production or application of a particular polymer within the class of polymers based on the embedding for the class of polymers, the first plurality of data instances, and the at least one of the second and third pluralities of data instances.
2. The method of claim 1, wherein providing the recommendation related to production or application of the particular polymer comprises imputing data from the polymer representation learning model for missing data for a particular polymer based on other data instances within the class of polymers.
3. The method of claim 2, further comprising training the polymer representation learning model with the imputed data, the first plurality of data instances, and the at least one of the second and third pluralities of data instances to tune the embedding for the class of polymers.

4. The method of claim 1, further comprising fusing the first plurality of data instances and the at least one of the second and third pluralities of data instances to learn the embedding for the class of polymers.
5. The method of claim 1, further comprising operating the polymer representation learning model to learn a respective embedding for the first plurality of data instances and the at least one of the second and third pluralities of data instances and fusing the respective embeddings to the embedding for the class of polymers.
6. The method of claim 1, further comprising determining, based on the class of polymers, whether to:
 - fuse the first plurality of data instances and the at least one of the second and third pluralities of data instances and learn the embedding for the class of polymers; or
 - learn a respective embedding for each of the first plurality of data instances and the at least one of the second and third pluralities of data instances and fuse the respective embeddings to the embedding for the class of polymers.
7. The method of claim 1, wherein operating the polymer representation learning model to learn the embedding further comprises:
 - fusing a first portion of the first plurality of data instances and the at least one of the second and third pluralities of data instances and learning a first intermediate embedding;
 - learning a second intermediate embedding for a second portion of the first plurality of data instances and the at least one of the second and third pluralities of data instances; and
 - fusing the first and the second intermediate embeddings to the embedding for the class of polymers.
8. The method of claim 1, wherein providing the recommendation related to production of the particular polymer comprises determining at least one of a manufacturing condition, a monomer, and an additive for production the particular polymer, for which a corresponding first data instance is not available.

9. The method of claim 1, wherein providing the recommendation related to application of the particular polymer comprises determining at least one of an application condition, any formulation components, and any additives for a specific application of the particular polymer, for which a corresponding third data instance is not available.
10. The method of claim 9, wherein providing the recommendation related to application of the particular polymer further comprises determining application performance properties for the specific application of the particular polymer.
11. The method of claim 1, wherein operating the polymer representation learning model to learn the embedding further comprises masking a specific data instance and minimizing reconstruction loss of the specific data instance.
12. The method of claim 1, wherein the plurality of different data types include at least two data types of a group of data types including:
- numerical data;
 - images;
 - paired X-Y series data;
 - graphs; and
 - text data.
13. The method of claim 1, further comprising determining at least one of a manufacturing condition, any other molecules, and any additional chemicals used for production of the particular polymer, for which a corresponding first data instance is not available, based on the embedding for the class of polymers, the first plurality of data instances, and the at least one of the second and third pluralities of data instances.
14. A data processing apparatus, comprising a processor configured to perform the method of any one of claims 1-13.

15. A non-transitory machine readable medium storing instructions executable by a processor to perform the method of any one of claims 1-13.