Leveraging theory for enhanced machine learning

Debra J. Audus,*,† Austin McDannald,‡ and Brian DeCost‡

†*Materials Science and Engineering Division, National Institute of Standards and Technology, Gaithersburg, MD*

‡*Materials Measurement Science Division, National Institute of Standards and Technology, Gaithersburg, MD*

E-mail: debra.audus@nist.gov

Abstract

The application of machine learning to the materials domain has traditionally struggled with two major challenges: a lack of large, curated datasets and the need to understand the physics behind the machine learning prediction. The former problem is particularly acute in the polymers domain. Here we aim to simultaneously tackle these challenges through the incorporation of scientific knowledge, thus, providing improved predictions for smaller datasets—both under interpolation and extrapolation—and a degree of explainability. We focus on imperfect theories, as they are often readily available and easier to interpret. Using a system of a polymer in different solvent qualities, we explore numerous methods for incorporating theory into machine learning using different machine learning models including Gaussian process regression. Ultimately, we find that encoding the functional form of the theory performs best followed by an encoding of the numeric values of the theory.

Machine learning (ML) continues to make astonishing strides in the polymer domain such as determination of new polymers for gas-separation polymer membranes,[1] the ability to generate new polymer structures,[2] transformation of polymer names to structures,[3]

prediction of polymer size based on sequence,[4] near instantaneous property prediction,[5–7] and many others.[8–19] However, these advances often rely on large, high-quality, datasets normally manually curated from sources such as PoLyInfo[20] or generated computationally. Furthermore, they rarely provide physical insight and often fail to extrapolate—representing significant barriers to further ML advances within the polymers domain.[21–26]

The lack of large datasets is currently being tackled by several efforts to compile data[8,13,20,27,28] including through natural language processing[29–33] and the generation of large computational datasets.[5–7] It is also being tackled by transfer learning, a growing technique in ML where knowledge is transferred between tasks (e.g., prediction of properties), domains (e.g., scientific literature or English literature) or both as detailed in an excellent review.[34] Most commonly knowledge is transferred from a task or domain where data is plentiful to a task or domain where data is limited. For example, Li *et al.* transferred knowledge between domains by using a deep convolutional neural net that was pre-trained on a non-scientific corpus to perform microstructure reconstruction and structure-property predictions for polymer nanocomposites and other materials.[11] Yamada *et al.* demonstrated exceptional success at transferring knowledge between tasks by predicting polymer properties with datasets sizes of order 10 after testing a large number of models trained on other properties as their source task.[12]

A powerful method for tackling data scarcity and potentially offering interpretability that is gaining momentum in the materials domain is the idea of utilizing domain knowledge (e.g., physical laws).[35–38] Depending on the details, this approach can improve predictions via methods such as physically inspired feature selection[39] or the utilization of constraints such as translational invariance.[17] In some cases, it can provide interpretability. For example, Menon *et al.* introduced a hierarchical ML approach which first models a collection of basic physical properties using simple physical models. These then serve as inputs to a linear model for a complex target property.[40] The physics are directly included, and the final expressions are simple and easy to interpret. They also applied this approach to other problems.[41–43]

Here we build on these ideas by incorporating an imperfect theory into ML and apply it to a problem in the polymers domain. We often have simple physical models, such as scaling theories,[44] that provide physical insight but rarely quantitative predictions. Such theories also might have limited domains of applicability. The question that we aim to answer is can we use an imperfect theory to improve ML prediction while providing insight for small datasets. Choice of a simple theory is important, as it is trivial to evaluate and is interpretable.

To answer this question, we consider a canonical polymer physics problem, namely, the polymer size as measured by the radius of gyration, R_g, of a single polymer chain in different solvent qualities for different chain lengths, N. We know that once the chain length becomes long enough, $R_g \sim N^\nu$ where ν is a scaling exponent and is equal to 1/3 for a bad solvent, 1/2 for a theta solvent and 0.588 for a good solvent.[45] As detailed in Methods, we build a dataset by simulating coarse-grained polymers of different lengths and in different solvent qualities, represented by the parameter α. $\alpha = 1$ for a bad solvent, $\alpha \approx 0.35$ for a theta solvent and $\alpha = 0$ for a good solvent. We then compute R_g and its uncertainty. Supporting Information (SI) contains a visualization of the data.

To test our methods, we focus on Gaussian process regression (GPR)[46] as it intrinsically includes uncertainty estimates, which are often important in science. GPR starts with the specification of the prior, or equivalently, the prediction before explicit use of the training data. The prediction is then updated using training data and Bayes theorem. The prior is assumed to be a multidimensional Gaussian with a mean of $m(\mathbf{x})$ and a covariance of $k(\mathbf{x}, \mathbf{x}')$, also known as the kernel. In our case, $\mathbf{x}$ is the vector of inputs and is equal to $[\alpha, \ln N]$. Note that we have already made use of $\ln R_g^2 \sim 2\nu \ln N$ by selecting $\ln N$ as input and $\ln R_g^2$ as output. For simplicity, a Gaussian Process can be written as $\mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$. Unless stated otherwise, we take $m(\mathbf{x})$ to be a constant determined through model optimization and $k(\mathbf{x}, \mathbf{x}')$ to be the squared exponential kernel; this choice enforces smoothness and ensures that predictions for points that are close in $\mathbf{x}$ are similar. Since we predict $\ln R_g^2$ in the

thermodynamic limit, we expect a single value for a given set of input parameters. If one were to consider an output that was a distribution rather than an single value, modifications such as the addition of a white kernel would be necessary.

GPR models typically use homoscedastic noise, or equivalently, a constant variance in the model errors; this variance is then determined through model optimization. However, Gaussian processes with heteroscedastic noise[47] can model data with known variances by propagating uncertainties of both the model and the underlying process to the final probabilistic predictions. We take this more realistic approach.

To incorporate theory, we use several methods as detailed in Fig. 1a. There are two benchmarks: the direct method, which has no knowledge of the theory and the theory method, which has no knowledge of the data. For the theory method, we utilize that of a theta solvent ($R_g = (N/6)^{1/2}$) as we require numeric values. This particular choice has limited applicability, providing an excellent case study for exploring the utility of combining simple physical models with ML. The direct method improves with dataset size as seen in Fig. 1b as expected whereas the imperfect theory performs worse for all but the very smallest dataset sizes as it incorrectly assumes $\nu = 1/2$. The goal of the remaining methods is to perform at least as well as the direct method for all dataset sizes or, equivalently, to avoid a negative transfer[34] of knowledge.

The next three methods utilize the numerical values of the theory for the theta point. Two of which are adopted from Hutchinson *et al.* where they compared techniques for transfer learning to overcome data scarcity.[48] Specifically, we consider the difference method where the difference between source output ($\ln R^2_{g,theory} \equiv \ln N + \ln 1/6$) and target output ($\ln R^2_{g,dataset}$) is learned, and latent variable where source output is used to augment the target input ($[\ln N, \alpha]$). A key distinction between their work and ours is that we use theory for our source output, which eliminates the need for a surrogate ML model for the source task and provides physical insight. Inspired by the difference method, we consider a quotient method that uses the ratio in place of the difference.

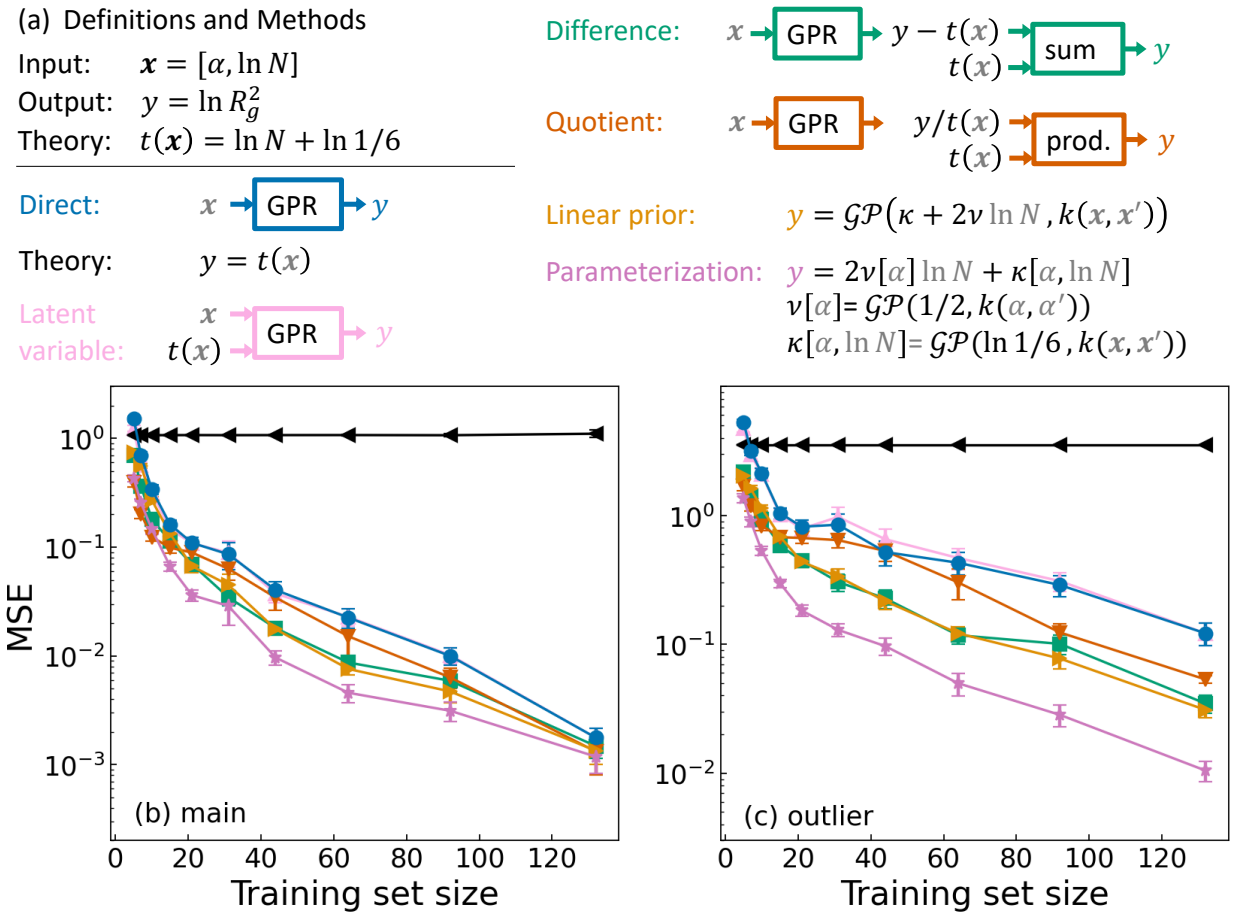

Figure 1: (a) A list of methods considered and definitions used therein. (b-c) Mean squared error (MSE) as a function of training set size. (b) is tested on the remaining dataset, while (c) is tested on the outlier dataset consisting of $N = 1024$. Error bars represent standard error computed from at least 10 different test sets. MSE is always larger than the largest uncertainty in the datasets. Full details on how model performance was determined can be found in Methods.

Of these three methods, the difference consistently performs either the best or no worse than both direct and theory whereas the quotient method performs the best for very small dataset sizes but worse than the difference method for larger dataset sizes. The difference model is mathematically equivalent to using the numerical values of the theory as the prior mean ($m(\mathbf{x}) = \ln N + \ln 1/6$) instead of a constant. The GPR takes the prior as an initial guess and updates its prediction based on nearby data; see Fig. 2. When the GPR extrapolates (no training data nearby, relative to the squared exponential kernel length scale), the predictions tend toward the priors (thin lines). Since the prior for the difference method encodes the $\ln N$ dependence, it is a better initial guess. Thus, it is not surprising that this simple encoding of theory would improve predictions. The quotient method reweights the data and does not have the same interpretation as the difference method. Since it exhibits negative transfer, the difference method is preferred.

We also consider two methods utilizing the form of the theory rather than numerical values. The first method is the linear prior, where the prior mean, the prediction before any data is used, is set equal to a linear combination of the inputs, i.e., $2\nu \ln N + \lambda\alpha + \kappa$ where ν, λ, and κ are learned constants. This is an extension of the difference method where $\nu = 0.5$ and $\lambda = 0$. This performs equal to or better than the benchmarks for all training set sizes but worse than the difference method for small training set sizes, which is likely due to overfitting. The second method is the parameterization method, which more fully utilizes the form of the theory in the prior. Instead of representing $\ln R_g^2$ as $\mathcal{GP}(\ln N + m, k(\mathbf{x}, \mathbf{x}'))$ as in the difference method, one can write $\ln R_g^2$ as $2\nu[\alpha](\ln N) + \kappa[\alpha, \ln N]$ where $\nu[\alpha] = \mathcal{GP}(1/2, k(\alpha, \alpha'))$ and $\kappa[\alpha, \ln N] = \mathcal{GP}(\ln 1/6 + m_\kappa, k(\mathbf{x}, \mathbf{x}'))$. m and m_κ are constants determined during model optimization. This method performs the best of all of the methods. This improved prior better represents the underlying data; see Fig. 2.

In the SI, we consider two other ML models: GPR with homoscedastic noise and random forest (RF). We find that the results for the GPR with heteroscedastic noise and homoscedastic noise are qualitatively similar. Overall, the GPR with homoscedastic noise performs bet-

ter but this is due to overfitting, does not provide accurate uncertainty estimates and does not respect that the uncertainty inherently depends on both N and α. RF performs worse than GPR showing that model choice is important. Unlike for GPR, the quotient method performs best for RF for our particular problem due to the structure of the ML model.

Most ML techniques are intrinsically interpolative rather than extrapolative. One of the exciting benefits of incorporating theory is improved performance under extrapolation. To test this, we consider an outlier dataset composed of $N = 1024$ (the largest N for the main dataset was 512) and the full range of α; see Fig. 1c. The performance degrades compared to the original training set, but the best performing methods outperform the direct method.

Improved interpolation and extrapolation can be explicitly demonstrated by considering the dataset size required to achieve a desired MSE. For interpolation, we find that dataset sizes of 99, 74, and 51 are required to achieve a MSE of 10^{-2} for the direct, difference and parameter methods, respectively. For extrapolation, dataset sizes of 109, 47, and 20 are required for achieve a MSE of 0.2. Thus, about a quarter (half) less data is required for the difference method and about half (a fifth) less data is required for the parameterization method compared to the direct method for interpolation (extrapolation).

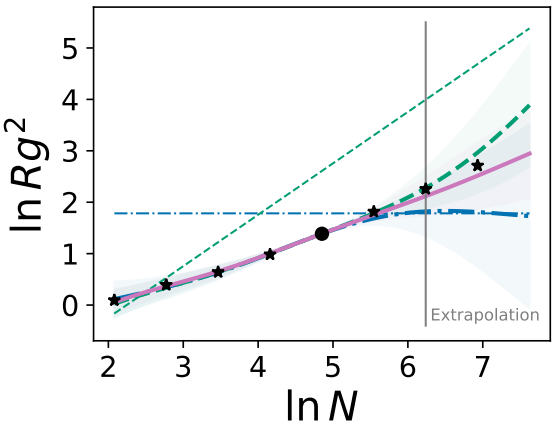

Figure 2: Predictions for different methods for incorporating theory: direct (blue), difference (green), parameterization (purple). Thick lines represent predicted means, and shaded regions indicate 95 % confidence intervals. Thin lines are learned priors. Circles are training data and stars are test data. Predictions are plotted for $\alpha = 1$ (bad solvent). Training data has 30 randomly selected points of which only point is for $\alpha = 1$.

We have shown that incorporating theory can improve ML performance for small dataset sizes ($\mathcal{O}(10-100)$). However, a benefit to using theory is that there is the possibility to extract knowledge. For the difference method, we use a dataset with 30 training points; see Fig. 3. Deviations increase away from $\alpha \approx 0.4$ demonstrating that the theta solvent expression is not accurate across the full range of α. We also see that the theory performs worse for larger values of N. Although we expected these results *a priori*, if we were using a theory where the limits were unknown, this analysis would help elucidate that.

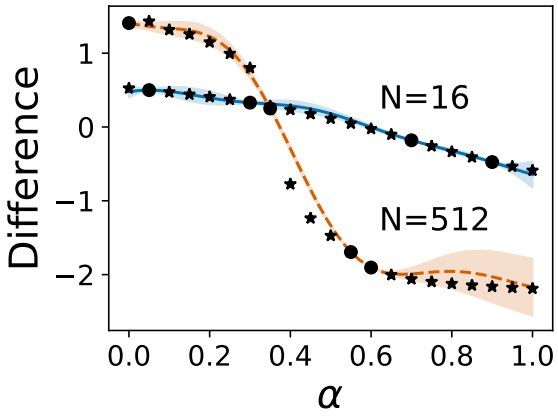

Figure 3: The learned difference heteroscedastic GPR for a dataset size of 30. Line represents predicted mean and shaded regions indicate 95 % confidence intervals. Circles denote points training data and stars denote test data.

We consider the parameterization method, which includes the functional form of the theory. Thus, the latent functions directly relate to known physics. The prediction for ν directly includes the known values for both good solvent (0.588) and bad solvent (1/3) with a crossover region in between as seen in Fig. 4a. We also know that for large N the scaling prefactor should be independent of N. As seen in Fig. 4b, the prefactor deviates from theory and plateaus before reverting to a learned constant. Compared to simply fitting scaling theory for large N at different α values, the GPR can account for deviations at small N and use information at other α values highlighting the power of parameterization.

In this letter, we have explored a variety of methods for combining theory and ML. In particular, the difference method, which uses theoretical values and the parameterization

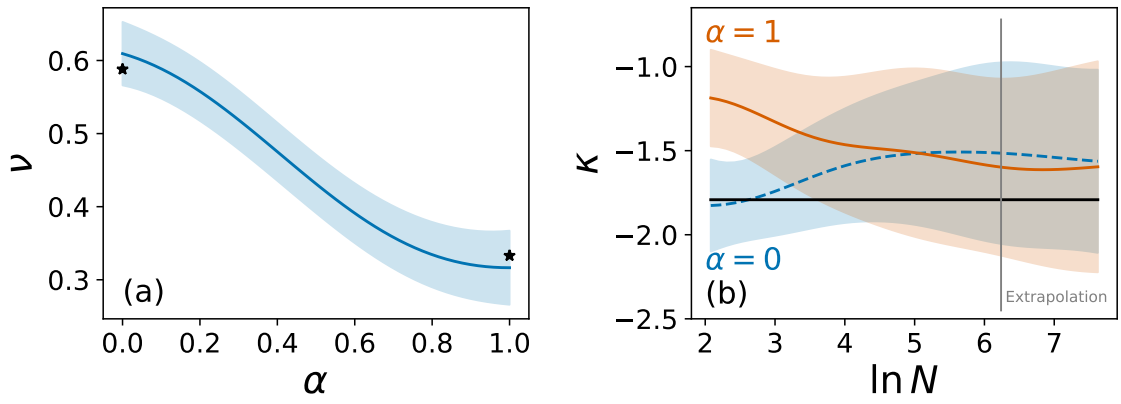

Figure 4: The learned parameterization heteroscedastic GPR for a dataset size of 30. (a) The radius of gyration scaling exponent, ν, for various solvent qualities. Stars represent known theoretical limits. (b) κ parameter for $\alpha = 0$ (good solvent; blue) and $\alpha = 1$ (bad solvent; red) and various chain lengths. The black line represents the theoretical expectation for a theta solvent. In both cases, colored lines represent predicted mean and shaded regions indicate 95 % confidence intervals.

method, which encodes knowledge of the functional form, when used in conjunction with GPR perform particularly well while providing physical insight. This can be thought of as careful, informed model selection. In our example, we worked with simple, relatively easy to generate simulated data that was uniformly sampled across parameter space. The ideas discussed in this letter are even more interesting when considering experimental data, slightly more complicated theory and biased datasets. To make these concepts easy to apply, we have provided Jupyter Notebooks[49] (see SI for details).

Methods

To generate our dataset, we simulate a single coarse-grained polymer in an implicit solvent. Bonds are modeled with a finite extensible nonlinear elastic (FENE) potential[50] with $K = 30$ and $R_0 = 1.5$. The non-bonded potential is

$$U(r) = (1 - \alpha)U^{WCA}(r) + \alpha U^{LJ}(r) \tag{1}$$

where U^{LJ} is the Lennard-Jones potential with a cutoff of 2.5 and U^{WCA} is the Weeks-Chandler-Andersen potential.[51] Reduced units are used throughout. α ranges from 0 to 1 in

increments of 0.05 and N ranges from 8 to 1024 in powers of 2 with the data at $N = 1024$ used as an outlier test set. This results in a dataset size of 147, with 21 additional data points in the outlier test set.

The initial configuration is generated using a random walk. Subsequently, molecular dynamics simulations using Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS)[52] are run in the canonical ensemble using a Langevin thermostat with a temperature of 1, a damping parameter of 1, and a timestep of 0.005. Simulations are run for a total of $2 \cdot 10^9$ timesteps with the first $2 \cdot 10^6$ timesteps discarded for equilibration. R_g^2 is then computed every 1000 timesteps. From this, the average and standard error of R_g^2 is computed and, ultimately, $\langle \ln R_g^2 \rangle$ and uncertainty. For $N = 1024$ and $\alpha < 0.4$, we found the uncertainty was too high, so we reran the simulations starting from a different initial configuration and combined the production data.

Gaussian process regression with heteroscedastic noise is implemented using the GPFlow python package[53,54] along with a new class derived from the gpflow.models.GPR to handle the additional input of a vector of variances for each output. The squared exponential kernel is used with a length scale for each input. Model hyperparameters are determined by maximizing the log likelihood of the training data with the Adam optimizer as implemented in tensorflow.[55] MSE is evaluated through a modified k-fold cross-validation procedure (see SI for details) where each data point is used at least once for testing.

Supporting Information Available

The Supporting Information (SI) is available free of charge at .

- SI: Dataset visualization, additional machine learning method details, details on implementing and interpreting the parameterization method, and a comparison of machine learning models
- https://doi.org/10.18434/mds2-2637: input data, MSE data and a link to both the

code used to generate the MSE data and Jupyter Notebooks to reproduce all figures

- https://github.com/usnistgov/TaML: code including Jupyter notebooks

References

(1) Barnett, J. W.; Bilchak, C. R.; Wang, Y.; Benicewicz, B. C.; Murdock, L. A.; Bereau, T.; Kumar, S. K. Designing exceptional gas-separation polymer membranes using machine learning. *Science advances* **2020**, *6*, eaaz4301.

(2) Ma, R.; Luo, T. PI1M: a benchmark database for polymer informatics. *Journal of Chemical Information and Modeling* **2020**, *60*, 4684–4690.

(3) Lin, C.; Wang, P.-H.; Hsiao, Y.; Chan, Y.-T.; Engler, A. C.; Pitera, J. W.; Sanders, D. P.; Cheng, J.; Tseng, Y. J. Essential step toward mining big polymer data: polyname2structure, mapping polymer names to structures. *ACS Applied Polymer Materials* **2020**, *2*, 3107–3113.

(4) Webb, M. A.; Jackson, N. E.; Gil, P. S.; de Pablo, J. J. Targeted sequence design within the coarse-grained polymer genome. *Science advances* **2020**, *6*, eabc6216.

(5) Huan, T. D.; Mannodi-Kanakkithodi, A.; Kim, C.; Sharma, V.; Pilania, G.; Ramprasad, R. A polymer dataset for accelerated property prediction and design. *Scientific data* **2016**, *3*, 1–10.

(6) Kim, C.; Chandrasekaran, A.; Huan, T. D.; Das, D.; Ramprasad, R. Polymer genome: a data-powered polymer informatics platform for property predictions. *The Journal of Physical Chemistry C* **2018**, *122*, 17575–17585.

(7) Doan Tran, H.; Kim, C.; Chen, L.; Chandrasekaran, A.; Batra, R.; Venkatram, S.; Kamal, D.; Lightstone, J. P.; Gurnani, R.; Shetty, P., et al. Machine-learning predictions

of polymer properties with Polymer Genome. *Journal of Applied Physics* **2020**, *128*, 171104.

(8) Zhao, H.; Li, X.; Zhang, Y.; Schadler, L. S.; Chen, W.; Brinson, L. C. Perspective: NanoMine: A material genome approach for polymer nanocomposites analysis and design. *APL Materials* **2016**, *4*, 053204.

(9) Meenakshisundaram, V.; Hung, J.-H.; Patra, T. K.; Simmons, D. S. Designing sequence-specific copolymer compatibilizers using a molecular-dynamics-simulation-based genetic algorithm. *Macromolecules* **2017**, *50*, 1155–1166.

(10) Cubuk, E. D.; Ivancic, R.; Schoenholz, S. S.; Strickland, D.; Basu, A.; Davidson, Z.; Fontaine, J.; Hor, J. L.; Huang, Y.-R.; Jiang, Y., et al. Structure-property relationships from universal signatures of plasticity in disordered solids. *Science* **2017**, *358*, 1033–1037.

(11) Li, X.; Zhang, Y.; Zhao, H.; Burkhart, C.; Brinson, L. C.; Chen, W. A transfer learning approach for microstructure reconstruction and structure-property predictions. *Scientific reports* **2018**, *8*, 1–13.

(12) Yamada, H.; Liu, C.; Wu, S.; Koyama, Y.; Ju, S.; Shiomi, J.; Morikawa, J.; Yoshida, R. Predicting materials properties with little data using shotgun transfer learning. *ACS central science* **2019**, *5*, 1717–1730.

(13) Brinson, L. C.; Deagen, M.; Chen, W.; McCusker, J.; McGuinness, D. L.; Schadler, L. S.; Palmeri, M.; Ghumman, U.; Lin, A.; Hu, B. Polymer nanocomposite data: curation, frameworks, access, and potential for discovery and design. *ACS Macro Letters* **2020**, *9*, 1086–1094.

(14) Ethier, J. G.; Casukhela, R. K.; Latimer, J. J.; Jacobsen, M. D.; Shantz, A. B.; Vaia, R. A. Deep Learning of Binary Solution Phase Behavior of Polystyrene. *ACS Macro Letters* **2021**, *10*, 749–754.

(15) Jablonka, K. M.; Jothiappan, G. M.; Wang, S.; Smit, B.; Yoo, B. Bias free multiobjective active learning for materials design and discovery. *Nature communications* **2021**, *12*, 1–10.

(16) Arora, A.; Lin, T.-S.; Rebello, N. J.; Av-Ron, S. H.; Mochigase, H.; Olsen, B. D. Random Forest Predictor for Diblock Copolymer Phase Behavior. *ACS Macro Letters* **2021**, *10*, 1339–1345.

(17) Xuan, Y.; Delaney, K. T.; Ceniceros, H. D.; Fredrickson, G. H. Deep learning and self-consistent field theory: A path towards accelerating polymer phase discovery. *Journal of Computational Physics* **2021**, 110519.

(18) Wessels, M. G.; Jayaraman, A. Computational reverse-engineering analysis of scattering experiments (CREASE) on amphiphilic block polymer solutions: cylindrical and fibrillar assembly. *Macromolecules* **2021**, *54*, 783–796.

(19) Statt, A.; Kleeblatt, D. C.; Reinhart, W. F. Unsupervised learning of sequence-specific aggregation behavior for a model copolymer. *Soft matter* **2021**, *17*, 7697–7707.

(20) Polymer Database. `https://polymer.nims.go.jp/en`, accessed 2020-10-29.

(21) Audus, D. J.; de Pablo, J. J. Polymer informatics: Opportunities and challenges. *ACS macro letters* **2017**, *6*, 1078–1082.

(22) Peerless, J. S.; Milliken, N. J.; Oweida, T. J.; Manning, M. D.; Yingling, Y. G. Soft matter informatics: current progress and challenges. *Advanced Theory and Simulations* **2019**, *2*, 1800129.

(23) Jackson, N. E.; Webb, M. A.; de Pablo, J. J. Recent advances in machine learning towards multiscale soft materials design. *Current Opinion in Chemical Engineering* **2019**, *23*, 106–114.

(24) Wu, S.; Yamada, H.; Hayashi, Y.; Zamengo, M.; Yoshida, R. Potentials and challenges of polymer informatics: exploiting machine learning for polymer design. *arXiv preprint arXiv:2010.07683* **2020**,

(25) Sha, W.; Li, Y.; Tang, S.; Tian, J.; Zhao, Y.; Guo, Y.; Zhang, W.; Zhang, X.; Lu, S.; Cao, Y.-C., et al. Machine learning in polymer informatics. *InfoMat* **2021**, *3*, 353–361.

(26) Chen, L.; Pilania, G.; Batra, R.; Huan, T. D.; Kim, C.; Kuenneth, C.; Ramprasad, R. Polymer informatics: Current status and critical next steps. *Materials Science and Engineering: R: Reports* **2021**, *144*, 100595.

(27) Tchoua, R. B.; Qin, J.; Audus, D. J.; Chard, K.; Foster, I. T.; de Pablo, J. Blending education and polymer science: Semiautomated creation of a thermodynamic property database. *Journal of chemical education* **2016**, *93*, 1561–1568.

(28) A Community Resource for Innovation in Polymer Technology. `http://cript.mit.edu/`, Accessed: 2022-05-10.

(29) Tchoua, R. B.; Chard, K.; Audus, D. J.; Ward, L. T.; Lequieu, J.; De Pablo, J. J.; Foster, I. T. Towards a hybrid human-computer scientific information extraction pipeline. 2017 IEEE 13th international conference on e-Science (e-Science). 2017; pp 109–118.

(30) Tchoua, R.; Ajith, A.; Hong, Z.; Ward, L.; Chard, K.; Audus, D.; Patel, S.; de Pablo, J.; Foster, I. Active learning yields better training data for scientific named entity recognition. 2019 15th International Conference on eScience (eScience). 2019; pp 126–135.

(31) Tchoua, R. B.; Ajith, A.; Hong, Z.; Ward, L. T.; Chard, K.; Belikov, A.; Audus, D. J.; Patel, S.; de Pablo, J. J.; Foster, I. T. Creating training data for scientific named entity recognition with minimal human effort. International Conference on Computational Science. 2019; pp 398–411.

(32) Hong, Z.; Tchoua, R.; Chard, K.; Foster, I. SciNER: extracting named entities from scientific literature. International Conference on Computational Science. 2020; pp 308–321.

(33) Shetty, P.; Ramprasad, R. Automated knowledge extraction from polymer literature using natural language processing. *Iscience* **2021**, *24*, 101922.

(34) Pan, S. J.; Yang, Q. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* **2009**, *22*, 1345–1359.

(35) Wagner, N.; Rondinelli, J. M. Theory-guided machine learning in materials science. *Frontiers in Materials* **2016**, *3*, 28.

(36) Karpatne, A.; Atluri, G.; Faghmous, J. H.; Steinbach, M.; Banerjee, A.; Ganguly, A.; Shekhar, S.; Samatova, N.; Kumar, V. Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE Transactions on knowledge and data engineering* **2017**, *29*, 2318–2331.

(37) Childs, C. M.; Washburn, N. R. Embedding domain knowledge for machine learning of complex material systems. *MRS Communications* **2019**, *9*, 806–820.

(38) DeCost, B. L.; Hattrick-Simpers, J. R.; Trautt, Z.; Kusne, A. G.; Campo, E.; Green, M. L. Scientific AI in materials science: a path to a sustainable and scalable paradigm. *Machine learning: science and technology* **2020**, *1*, 033001.

(39) Pugar, J. A.; Childs, C. M.; Huang, C.; Haider, K. W.; Washburn, N. R. Elucidating the Physicochemical Basis of the Glass Transition Temperature in Linear Polyurethane Elastomers with Machine Learning. *The Journal of Physical Chemistry B* **2020**, *124*, 9722–9733.

(40) Menon, A.; Gupta, C.; Perkins, K. M.; DeCost, B. L.; Budwal, N.; Rios, R. T.; Zhang, K.; Póczos, B.; Washburn, N. R. Elucidating multi-physics interactions in

suspensions for the design of polymeric dispersants: a hierarchical machine learning approach. *Molecular Systems Design & Engineering* **2017**, *2*, 263–273.

(41) Menon, A.; Thompson-Colón, J. A.; Washburn, N. R. Hierarchical machine learning model for mechanical property predictions of polyurethane elastomers from small datasets. *Frontiers in Materials* **2019**, *6*, 87.

(42) Menon, A.; Póczos, B.; Feinberg, A. W.; Washburn, N. R. Optimization of silicone 3D printing with hierarchical machine learning. *3D Printing and Additive Manufacturing* **2019**, *6*, 181–189.

(43) Bone, J. M.; Childs, C. M.; Menon, A.; Poczos, B.; Feinberg, A. W.; LeDuc, P. R.; Washburn, N. R. Hierarchical Machine Learning for High-Fidelity 3D Printed Biopolymers. *ACS Biomaterials Science & Engineering* **2020**, *6*, 7021–7031.

(44) de Gennes, P. *Scaling concepts in polymer physics*; Cornell Univ. Pr.: Ithaca [u.a.], 1979; p 324 S.

(45) Rubinstein, M.; Colby, R. *Polymer Physics*; Oxford University Press, 2003.

(46) Rasmussen, C. E.; Williams, C. K. I. *Gaussian processes for machine learning*; The MIT Press, 2006.

(47) Dallaire, P.; Besse, C.; Chaib-Draa, B. Learning Gaussian process models from uncertain data. International Conference on Neural Information Processing. 2009; pp 433–440.

(48) Hutchinson, M. L.; Antono, E.; Gibbons, B. M.; Paradiso, S.; Ling, J.; Meredig, B. Overcoming data scarcity with transfer learning. *arXiv preprint arXiv:1711.05099* **2017**,

(49) Kluyver, T.; Ragan-Kelley, B.; Pérez, F.; Granger, B.; Bussonnier, M.; Frederic, J.; Kelley, K.; Hamrick, J.; Grout, J.; Corlay, S.; Ivanov, P.; Avila, D.; Abdalla, S.; Willing, C.

Jupyter Notebooks – a publishing format for reproducible computational workflows. Positioning and Power in Academic Publishing: Players, Agents and Agendas. 2016; pp 87 – 90.

(50) Kremer, K.; Grest, G. S. Dynamics of entangled linear polymer melts: A molecular-dynamics simulation. *The Journal of Chemical Physics* **1990**, *92*, 5057–5086.

(51) Weeks, J. D.; Chandler, D.; Andersen, H. C. Role of repulsive forces in determining the equilibrium structure of simple liquids. *The Journal of chemical physics* **1971**, *54*, 5237–5247.

(52) Thompson, A. P.; Aktulga, H. M.; Berger, R.; Bolintineanu, D. S.; Brown, W. M.; Crozier, P. S.; in 't Veld, P. J.; Kohlmeyer, A.; Moore, S. G.; Nguyen, T. D.; Shan, R.; Stevens, M. J.; Tranchida, J.; Trott, C.; Plimpton, S. J. LAMMPS - a flexible simulation tool for particle-based materials modeling at the atomic, meso, and continuum scales. *Comp. Phys. Comm.* **2022**, *271*, 108171.

(53) Matthews, A. G. d. G.; van der Wilk, M.; Nickson, T.; Fujii, K.; Boukouvalas, A.; León-Villagrá, P.; Ghahramani, Z.; Hensman, J. GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research* **2017**, *18*, 1–6.

(54) van der Wilk, M.; Dutordoir, V.; John, S.; Artemev, A.; Adam, V.; Hensman, J. A Framework for Interdomain and Multioutput Gaussian Processes. *arXiv:2003.01115* **2020**,

(55) Abadi, M. et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015; `https://www.tensorflow.org/`, Software available from tensorflow.org.

Graphical TOC Entry