Is GPT-3 all you need for machine learning for chemistry?

Anonymous Author(s)
Affiliation
Address
email

Abstract

Pre-trained large language models (LLMs) are a powerful platform for building custom models for various applications. They have also found success in chemistry, but typically need to be pre-trained on large chemistry datasets such as reaction databases or protein sequences. In this work, we analyze whether one of the largest pre-trained LLMs, GPT-3, can be directly used for chemistry applications by fine-tuning on only a few data points from a chemistry dataset, i.e., without pre-training on a chemistry-specific dataset. We show that GPT-3 can achieve performance competing with baselines on three case studies (polymers, metal-organic frameworks, photoswitches) with representations as simple as the chemical name in both classification and regression settings. Moreover, we demonstrate that GPT-3 can also be fine-tuned for use in inverse design tasks, i.e., to generate a molecule that has properties as specified in a prompt.

1 Introduction

Large language models (LLMs) are billion parameter neural networks also known as foundation models.[1] They are referred to as this because they can serve as the foundation for various downstream tasks. This insight that models trained on large amounts of unlabeled data can be fine-tuned with only a few data for specific tasks has led to a revolution in how machine learning systems are being built. An archetypal example of such a foundation model is the Generative Pre-trained Transformer 3 (GPT-3),[2] a model trained by OpenAI.[3,4] GPT-3 has 175 billion parameters, which were trained for months on hundreds of billions of tokens of text mostly crawled from the Web. Since the release, GPT-3 was used as the foundation for the code-completion model Codex (powering GitHub's Copilot),[5] which has already been tested for chemistry applications,[3] or to even write newspaper articles.[6]

2 Background

Given that most molecules and materials science can be represented as text, there have also been efforts to use LLM to predict their properties.[7] One example of a text-based representation of chemistry is the International Union of Pure and Applied Chemistry (IUPAC) name of chemicals (e.g., 1,3,7-Trimethylpurine-2,6-dione). Although it allows chemists to uniquely identify compounds, it has not been widely used for predictive models. More success has been found with using compacter line notations such as SMILES[8] (e.g., `CN1C=NC2=C1C(=O)N(C(=O)N2C)C`), or more recently, SELFIES (e.g., `[C][N][C][=N][C][=C][Ring1][Branch1][C][=Branch1][C][=O][N][Branch1][=Branch2][C][=Branch1][C][=O][N][Ring1][Branch2][C][C]`),[9,10] for sequence-to-sequence tasks such as reaction prediction[11,12] or protein design[13] and also regression tasks such as molecular property or yield prediction.[14,15] However, all these applications require large datasets[16] (even though they can be unlabeled in some cases) and often also changes in the architecture or training

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

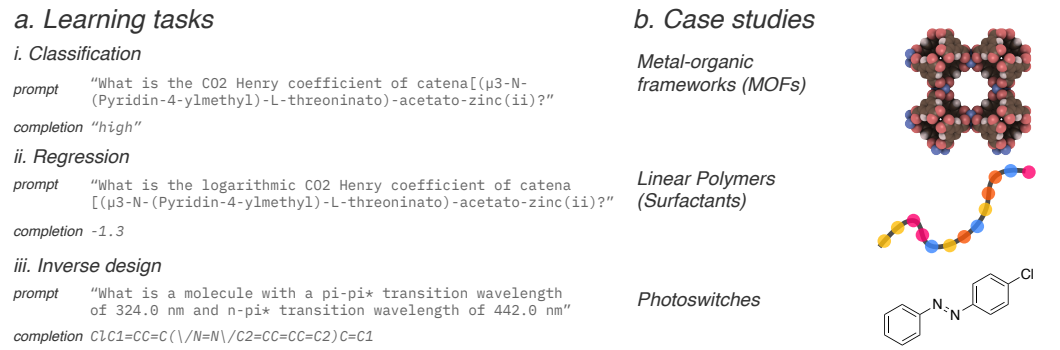

Figure 1: a) We used GPT-3 for three learning tasks: i) For classification, we discretize our target variable and then fine-tune GPT-3 to predict the class label. ii) For regression, we round the target value to two decimal points and then fine-tune GPT-3 to predict the target value. iii) For inverse design, we prompt a model with a sentence containing the desired properties for a molecule. b) We test our approach on three case studies; from extended crystals such as metal-organic frameworks, over linear polymers that could be used as surfactants, to photoswitch molecules.

procedure (e.g., the addition of special regression heads or losses).
In this work, we show that GPT-3 can be easily—using simple string-based representations such as the IUPAC name or a SMILES—fine-tuned to achieve performance competitive with strong baselines on multiple chemistry and materials science tasks.

3 Methods

To use GPT-3 for chemistry and materials science regression and classification problems, we use the language-interfaced fine-tuning (LIFT) framework proposed by Dinh et al.[17] in which the tasks are encoded in text prompts of the form of `what is <property> of <material encoding>?` (Figure 1). Note that the actual prompt also contains special tokens to indicate the end of a prompt or completion (`###`, `@@`). For all classification and regression predictions, we only considered the highest probability output. We use the OpenAI API with default settings for all experiments shown in this manuscript. In total, the computations used for this project consumed about $1.00\,\mathrm{k\$}$. All the code and data used in this study can be found under MIT license at `https://tinyurl.com/gpt3forchem`.

To analyze the potential of using GPT-3 for chemistry and materials science applications, we considered tasks across multiple domains, from molecules, over polymers to extended crystal structures:

Dispersants We analyze the adsorption energy of linear copolymers onto a model surface. This energy is relevant for dispersant applications where the polymer is supposed to prevent the flocculation of suspended particles, e.g., to increase the color brightness of pigments. We represent polymers with a simple bead notation, e.g., `ABAC`.[18]

Photoswitches Photoswitchable molecules can be converted between its *cis*/*trans* form by the irradiation with light.[19,20] Photoswitches have various potential applications, e.g., in energy storage, electronics, or photopharmacology. Key descriptors for the photoisomerization, and our learning objectives, are the wavelengths for the $\pi - \pi^*$ and $n - \pi^*$ transitions. We represent the molecules using SMILES strings, SELFIES, and IUPAC names (which we retrieved from the NCI/CADD chemical identifier resolver).

Metal-organic frameworks Metal-organic frameworks (MOFs) are one of the most active fields of chemistry as they promise to be designable materials across multiple scales.[21–24] They have been the focus of the design of gas separation processes such as carbon capture or methane storage, as well as for the use in photocatalytic applications. Here, we use gas separation indicators (for carbon dioxide and methane) computed using grand canonical Monte-Carlo simulations as well as band gaps, as provided in the mofdscribe package.[25–27] We represent MOFs with MOFid[28] and chemical names (as retrieved from the Cambridge Structural Database, CSD[29]).

4 Results and discussion

4.1 Classification

A common application of natural language models is classification. To test the applicability of GPT-3 in this setting for chemistry problems, we converted all case studies into classification tasks by binning our continuous targets into five equally sized bins. To compare with regression models as baselines, apply the same binning procedure to the outputs of the regression models. In Table 1 (and Table 3) we compare the performance of our approach with baselines across our case studies. Across all case studies, we find the fine-tuning of GPT-3 to be competitive with our baseline models (in several cases achieving better predictive performance). This is in particular also the case for the few-shot classification setting in which we provide the model only with ten examples of the particular dataset in the training step. Interestingly, models trained on the chemical name outperform those trained on more conventional line notations such as SELFIES and SMILES. One possible reason for this is that the chemical name provides more meaningful chemical context than the line notations.

Table 1: Classification metrics for the photoswitch case study. Values indicate the means and standard deviation of at least three (typically ten) independent runs. The baseline is based on the GPR model proposed by Griffiths et al. [19] The metrics for the MOF and dispersant case studies can be found in the Appendix in Table 3.

model	accuracy (macro)	F_1 micro	F_1 macro
photoswitch few shot (10 training points)			
GPT-3 on IUPAC names	0.82 ± 0.03	0.64 ± 0.06	0.34 ± 0.08
GPT-3 on SMILES	0.75 ± 0.03	0.34 ± 0.12	0.17 ± 0.07
GPT-3 on SELFIES	0.76 ± 0.04	0.33 ± 0.11	0.18 ± 0.07
baseline (GPR)	0.75 ± 0.04	0.40 ± 0.10	0.26 ± 0.15
photoswitch (300 training points)			
GPT-3 on SMILES	0.87 ± 0.05	0.66 ± 0.16	0.61 ± 0.12
GPT-3 on SELFIES	0.88 ± 0.01	0.70 ± 0.03	0.62 ± 0.11
baseline (GPR)	0.91 ± 0.02	0.78 ± 0.05	0.77 ± 0.07

4.2 Regression

Given the good performance in the classification setting, we investigated if our approach can also be used to directly solve the regression task, i.e., predict floating point numbers instead of classes. For this, we use the same prompt and completion templates but replace the class label with the rounded target values (to ensure that our training completions have a finite fixed number of two decimal points). Note that we do not change the loss function (i.e., we still use the default cross-entropy loss used for token prediction in language models) or the model architecture. Also in the regression setting (see Table 2 and Table 4) we observe performance competing with or exceeding the one of the baselines.

Table 2: Regression metrics for the photoswitch case study. Values indicate the means and standard deviation of (typically ten) independent runs. The baseline is based on the GPR model proposed by Griffiths et al. [19] We could not retrieve the IUPAC name for all molecules, wherefore there is no metric for the model trained on IUPAC names for 350 training points. Metrics for the MOF and dispersant case studies can be found in the Appendix in Table 4.

photoswitch few shot (50 training points)

model	MAE / nm
GPT-3 on IUPAC names	27.23 ± 7.19
GPT-3 on SMILES	50.13 ± 6.58
GPT-3 on SELFIES	47.82 ± 1.03
baseline	26.82 ± 4.04

photoswitch (350 training points)

model	MAE / nm
GPT-3 on SMILES	21.38 ± 4.58
GPT-3 on SELFIES	23.18 ± 3.07
baseline	14.10 ± 3.13

4.3 Inverse design

The perhaps ultimate goal of machine learning in chemistry is to create a model that can generate molecules with a desired set of properties. This is also known as inverse design.[30] A very convenient input form for inverse design models might be natural language, such that the chemist can prompt the model with an English prompt specifying the desired properties. Here, investigated this setting: Can a fine-tuned GPT-3 propose valid molecules that satisfy the constraints in a prompt?

For this, we focus on the polymer and photoswitch case studies as we can use existing tools to verify the predictions of the model (for the MOF design, there is currently no direct mapping between MOFid or chemical name and the crystal structure, wherefore we cannot easily validate predictions made by the model). In both cases, we still performed a train test split (0.9/0.1) to obtain independent prompts for the test of the model that follows the same distribution as the data in the training set.

Dispersants We task the model to find monomer sequences that have specific adsorption energy as well as a specific composition. Therefore, it is natural to evaluate the model by measuring i) how many of the generated monomer sequences are valid, ii) how well they satisfy the composition prompt, and iii) how far the adsorption energy for the generated polymer deviates from the desired one. We find that the model succeeds in generating valid polymer strings that also closely satisfy the constraints (Table 5). Only at high softmax temperatures it rarely generates invalid polymer sequences.

Photoswitches For photoswitches, we also observe that the model can generate valid SMILES strings (Table 6). Interestingly, we observe here a stronger influence on the softmax temperature. A higher temperature leads the model to generate more novel molecules, that it has not seen before in the training set, however, this also causes it to produce fewer valid ones. In addition, the errors on the prompt agreement for the transition wavelengths tend to increase.

5 Discussion

Limitations and outlook Although fine-tuning of pre-trained large language models such as GPT-3 shows encouraging results, there are still many questions that our work did not address. For instance, we found in preliminary experiments that the generalization performance depends on fine-tuning hyperparameters, such as the number of epochs that we did not optimize. Moreover, it is well known that prompt engineering can lead to drastically improved predictive performance,[3,31] which we also did not systematically explore in this work (Appendix A.7). Additionally, the use of the OpenAI API limits the types of analyses we can perform on the model. For instance, it is not possible to analyze the embeddings of the fine-tuned models. For this reason, we are in the process of replicating our experiments with the open-source GPT-J model (which will also allow us to customize the tokenization).[32]

Societal impacts While our research might be used to accelerate the discovery of new materials and drugs that can have a wide range of applications, it could also be used for malicious purposes such as the development of chemical agents or weapons.

6 Conclusions

We showed that fine-tuned GPT-3 models can show performance competing with, or even surpassing, baselines in both classification and regression settings after fine-tuning with little data. Moreover, this approach also shows encouraging results for the inverse design of materials. After fine-tuning with only a few examples, we could generate valid SMILES strings and monomer sequences that show good agreement with the desired properties specified in the prompt. Overall, this paradigm of using large pre-trained LLMs might democratize access to machine learning for the discovery of materials as competing performance can be achieved without any customization of the architecture or training procedure and without the development of specific featurization approaches—in our case, the IUPAC name could sometimes beat the performance of hand-crafted features.

References

[1] Bommasani, R. et al. On the Opportunities and Risks of Foundation Models. *CoRR* **2021**, *abs/2108.07258*.

[2] Brown, T. B. et al. Language Models are Few-Shot Learners. *CoRR* **2020**, *abs/2005.14165*.

[3] Hocky, G. M.; White, A. D. Natural language processing models that automate programming will transform chemistry research and teaching. *Digital Discovery* **2022**, *1*, 79–83.

[4] White, A. D.; Hocky, G. M.; Gandhi, H. A.; Ansari, M.; Cox, S.; Wellawatte, G. P.; Sasmal, S.; Yang, Z.; Liu, K.; Singh, Y.; Ccoa, W. J. P. Do large language models know chemistry? **2022**,

[5] Chen, M. et al. Evaluating Large Language Models Trained on Code. *CoRR* **2021**, *abs/2107.03374*.

[6] A Robot Wrote This Entire Article. Are You Scared yet, Human? *The Guardian* **2020**,

[7] Schwaller, P.; Vaucher, A. C.; Laplaza, R.; Bunne, C.; Krause, A.; Corminboeuf, C.; Laino, T. Machine intelligence for chemical reaction space. *WIREs Comput Mol Sci* **2022**, *12*.

[8] Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences* **1988**, *28*, 31–36.

[9] Krenn, M. et al. SELFIES and the future of molecular string representations. *arXiv 2204.00056* **2022**,

[10] Krenn, M.; Häse, F.; Nigam, A.; Friederich, P.; Aspuru-Guzik, A. Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. *Machine Learning: Science and Technology* **2020**, *1*, 045024.

[11] Schwaller, P.; Laino, T.; Gaudin, T.; Bolgar, P.; Hunter, C. A.; Bekas, C.; Lee, A. A. Molecular Transformer: A Model for Uncertainty-Calibrated Chemical Reaction Prediction. *ACS Central Science* **2019**, *5*, 1572–1583.

[12] Pesciullesi, G.; Schwaller, P.; Laino, T.; Reymond, J.-L. Transfer learning enables the molecular transformer to predict regio- and stereoselective reactions on carbohydrates. *Nature Communications* **2020**, *11*.

[13] Yang, Z.; Milas, K. A.; White, A. D. Now What Sequence? Pre-trained Ensembles for Bayesian Optimization of Protein Sequences. *bioRxiv* **2022**,

[14] Winter, B.; Winter, C.; Schilling, J.; Bardow, A. A smile is all you need: Predicting limiting activity coefficients from SMILES with natural language processing. *arXiv 2204.00056* **2022**,

[15] Schwaller, P.; Vaucher, A. C.; Laino, T.; Reymond, J.-L. Prediction of chemical reaction yields using deep learning. *Machine Learning: Science and Technology* **2021**, *2*, 015016.

[16] Frey, N.; Soklaski, R.; Axelrod, S.; Samsi, S.; Gomez-Bombarelli, R.; Coley, C.; Gadepally, V. Neural Scaling of Deep Chemical Models. *ChemRxiv* **2022**,

[17] Dinh, T.; Zeng, Y.; Zhang, R.; Lin, Z.; Gira, M.; Rajput, S.; Sohn, J.-y.; Papailiopoulos, D.; Lee, K. LIFT: Language-Interfaced Fine-Tuning for Non-Language Machine Learning Tasks. *arXiv 2206.06565* **2022**,

[18] 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*.

[19] Griffiths, R.-R.; Greenfield, J. L.; Thawani, A. R.; Jamasb, A. R.; Moss, H. B.; Bourached, A.; Jones, P.; McCorkindale, W.; Aldrick, A. A.; Fuchter, M. J.; Lee, A. A. Data-Driven Discovery of Molecular Photoswitches with Multioutput Gaussian Processes. *arXiv 2008.03226* **2022**,

[20] Thawani, A. R.; Griffiths, R.-R.; Jamasb, A.; Bourached, A.; Jones, P.; McCorkindale, W.; Aldrick, A.; Lee, A. The Photoswitch Dataset: A Molecular Machine Learning Benchmark for the Advancement of Synthetic Chemistry. *ChemRxiv* **2020**,

[21] Yaghi, O. M. Reticular chemistry in all dimensions. *ACS central science* **2019**, *5*, 1295–1300.

[22] Lyu, H.; Ji, Z.; Wuttke, S.; Yaghi, O. M. Digital reticular chemistry. *Chem* **2020**, *6*, 2219–2241.

[23] Furukawa, H.; Cordova, K. E.; O'Keeffe, M.; Yaghi, O. M. The chemistry and applications of metal-organic frameworks. *Science* **2013**, *341*, 1230444.

[24] Jablonka, K. M.; Ongari, D.; Moosavi, S. M.; Smit, B. Big-Data Science in Porous Materials: Materials Genomics and Machine Learning. *Chemical Reviews* **2020**, *120*, 8066–8129.

[25] Jablonka, K. M.; Rosen, A. S.; Krishnapriyan, A. S.; Smit, B. An ecosystem for digital reticular chemistry. *ChemRxiv* **2022**,

[26] Rosen, A. S.; Iyer, S. M.; Ray, D.; Yao, Z.; Aspuru-Guzik, A.; Gagliardi, L.; Notestein, J. M.; Snurr, R. Q. Machine learning the quantum-chemical properties of metal–organic frameworks for accelerated materials discovery. *Matter* **2021**, *4*, 1578–1597.

[27] Rosen, A. S.; Fung, V.; Huck, P.; O'Donnell, C. T.; Horton, M. K.; Truhlar, D. G.; Persson, K. A.; Notestein, J. M.; Snurr, R. Q. High-throughput predictions of metal–organic framework electronic properties: theoretical challenges, graph neural networks, and data exploration. *npj Comput Mater* **2022**, *8*.

[28] Bucior, B. J.; Rosen, A. S.; Haranczyk, M.; Yao, Z.; Ziebel, M. E.; Farha, O. K.; Hupp, J. T.; Siepmann, J. I.; Aspuru-Guzik, A.; Snurr, R. Q. Identification Schemes for Metal–Organic Frameworks To Enable Rapid Search and Cheminformatics Analysis. *Crystal Growth & Design* **2019**, *19*, 6682–6697.

[29] Groom, C. R.; Bruno, I. J.; Lightfoot, M. P.; Ward, S. C. The Cambridge structural database. *Acta Crystallographica Section B: Structural Science, Crystal Engineering and Materials* **2016**, *72*, 171–179.

[30] Sanchez-Lengeling, B.; Aspuru-Guzik, A. Inverse molecular design using machine learning: Generative models for matter engineering. *Science* **2018**, *361*, 360–365.

[31] Kojima, T.; Gu, S. S.; Reid, M.; Matsuo, Y.; Iwasawa, Y. Large Language Models are Zero-Shot Reasoners. *arXiv 2205.11916* **2022**,

[32] Wang, B.; Komatsuzaki, A. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`, 2021.

[33] Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. 2016; pp 785–794.

[34] Bergstra, J.; Bardenet, R.; Bengio, Y.; Kégl, B. Algorithms for hyper-parameter optimization. *Advances in neural information processing systems* **2011**, *24*.

[35] Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; Koyama, M. Optuna: A Next-generation Hyperparameter Optimization Framework. Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2019.

[36] Krishnapriyan, A. S.; Montoya, J.; Haranczyk, M.; Hummelshøj, J.; Morozov, D. Machine learning with persistent homology and chemical word embeddings improves prediction accuracy and interpretability in metal-organic frameworks. *Sci Rep* **2021**, *11*.

[37] Krishnapriyan, A. S.; Haranczyk, M.; Morozov, D. Topological Descriptors Help Predict Guest Adsorption in Nanoporous Materials. *J. Phys. Chem. C* **2020**, *124*, 9360–9368.

[38] Willems, T. F.; Rycroft, C. H.; Kazi, M.; Meza, J. C.; Haranczyk, M. Algorithms and tools for high-throughput geometry-based analysis of crystalline porous materials. *Microporous Mesoporous Mater.* **2012**, *149*, 134–141.

[39] Fernandez, M.; Trefiak, N. R.; Woo, T. K. Atomic Property Weighted Radial Distribution Functions Descriptors of Metal–Organic Frameworks for the Prediction of Gas Uptake Capacity. *J. Phys. Chem. C* **2013**, *117*, 14095–14105.

[40] Morgan, H. L. The Generation of a Unique Machine Description for Chemical Structures-A Technique Developed at Chemical Abstracts Service. *J. Chem. Doc.* **1965**, *5*, 107–113.

[41] RDKit contributors, RDKit: Open-source cheminformatics. `http://www.rdkit.org`.

[42] Ralaivola, L.; Swamidass, S. J.; Saigo, H.; Baldi, P. Graph kernels for chemical informatics. *Neural Networks* **2005**, *18*, 1093–1110.

[43] de G. Matthews, A. 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.

[44] Bjerrum, E. J. SMILES Enumeration as Data Augmentation for Neural Network Modeling of Molecules. *CoRR* **2017**, *abs/1703.07076*.

[45] Arús-Pous, J.; Johansson, S. V.; Prykhodko, O.; Bjerrum, E. J.; Tyrchan, C.; Reymond, J.-L.; Chen, H.; Engkvist, O. Randomized SMILES strings improve the quality of molecular generative models. *J Cheminform* **2019**, *11*.

[46] Schwaller, P.; Vaucher, A. C.; Laino, T.; Reymond, J.-L. Data augmentation strategies to improve reaction yield predictions and estimate uncertainty. *ChemRxiv* **2020**,

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Table 3: Classification metrics for the MOF and dispersant case studies. Values indicate the means and standard deviation of (typically ten) independent runs. For the baseline runs indicated with stars (*), hyperparameter optimization fails because not all classes occur in all splits. For the ones with † it did not finish within the allocated computational budget. In those cases, we report the performance without hyperparameter optimization.

model	accuracy (macro)	F_1 micro	F_1 macro
dispersants (2000 training points)			
GPT-3 fine tuned	0.93 ± 0.00	0.81 ± 0.02	0.77 ± 0.08
baseline (XGBoost)	0.91 ± 0.00	0.79 ± 0.00	0.79 ± 0.00
MOF bandgap (3000 training points)			
chemical name	0.92 ± 0.01	0.66 ± 0.06	0.79 ± 0.02
MOFid	0.91 ± 0.02	0.77 ± 0.04	0.61 ± 0.04
baseline (XGBoost†)	0.81 ± 0.01	0.53 ± 0.02	0.49 ± 0.05
MOF CO_2 Henry coefficient few shot (10 training points)			
chemical name	0.78 ± 0.01	0.42 ± 0.05	0.13 ± 0.01
MOFid	0.76 ± 0.04	0.39 ± 0.05	0.16 ± 0.03
baseline (XGBoost*)	0.74 ± 0.02	0.34 ± 0.04	0.20 ± 0.02
MOF CO_2 Henry coefficient (1000 training points)			
chemical name	0.84 ± 0.02	0.54 ± 0.01	0.33 ± 0.10
MOFid	0.80 ± 0.01	0.51 ± 0.03	0.30 ± 0.04
baseline (XGBoost†)	0.82 ± 0.00	0.55 ± 0.01	0.41 ± 0.02
MOF CH_4 deliverable capacity few shot (10 training points)			
chemical name	0.72 ± 0.01	0.29 ± 0.03	0.15 ± 0.03
MOFid	0.75 ± 0.04	0.32 ± 0.02	0.12 ± 0.03
baseline (XGBoost*)	0.72 ± 0.01	0.31 ± 0.03	0.23 ± 0.02
MOF CH_4 deliverable capacity (1000 training points)			
chemical name	0.77 ± 0.01	0.43 ± 0.02	0.36 ± 0.03
MOFid	0.76 ± 0.01	0.39 ± 0.02	0.31 ± 0.03
baseline (XGBoost†)	0.89 ± 0.00	0.72 ± 0.01	0.71 ± 0.01

A Appendix

A.1 MOF and dispersant case studies

We report metrics in Table 3 and Table 4.

A.2 Dispersant inverse design

We report metrics in Table 5.

A.3 Photoswitch inverse design

We report metrics in Table 6.

A.4 Fine-tuning hyperparameters

We performed all experiments shown in the main text with the default fine-tuning settings of the OpenAI API and the smallest model (`ada`) as, in preliminary experiments, we did not find a performance increase that would justify the high cost of the largest model (`davinci`). In preliminary experiments, we found that for optimal predictive performance the number of fine-tuning epochs should be optimized as a function of the number of training points. However, due to limitations in

Table 4: Regression metrics for the MOF and dispersant case study. Values indicate the means and standard deviation of (typically ten) independent runs.

model	MAE
dispersants few shot (10 training points)	
GPT-3 fine tuned	$(3.13 \pm 0.37)\,\mathrm{k_B T}$
baseline	$(12.32 \pm 0.00)\,\mathrm{k_B T}$
dispersants (2000 training points)	
GPT-3 fine tuned	$(0.52 \pm 0.03)\,\mathrm{k_B T}$
baseline	$(1.16 \pm 1.30)\,\mathrm{k_B T}$
MOF bandgap (few shot) (10 training points)	
MOFid	$(1.11 \pm 0.03)\,\mathrm{eV}$
chemical name	$(1.34 \pm 0.33)\,\mathrm{eV}$
baseline	$(1.64 \pm 0.00)\,\mathrm{eV}$
MOF bandgap (1000 training points)	
MOFid	$(0.58 \pm 0.06)\,\mathrm{eV}$
chemical name	$(0.54 \pm 0.03)\,\mathrm{eV}$
baseline	$(1.11 \pm 0.08)\,\mathrm{eV}$

Table 5: Prompt agreement and training set similarity for dispersant inverse design. We measure the maximum common subsequence with respect to all sequences in the training set and normalize it by the sequence length. Composition mismatch is computed with respect to each component and then aggregated by computing the mean. Mean absolute error is computed with respect to the performance of the XGBoost baseline model (see Table 4).

temperature	fraction valid	maximum common subsequence	composition mismatch	MAE / $k_\mathrm{B}T$
0.0	1.0 ± 0.0	0.52 ± 0.01	4.24 ± 0.04	3.10 ± 0.03
0.25	1.0 ± 0.0	0.53 ± 0.01	4.24 ± 0.04	3.09 ± 0.01
0.5	1.0 ± 0.0	0.54 ± 0.00	4.24 ± 0.04	3.06 ± 0.02
0.75	1.0 ± 0.0	0.55 ± 0.01	4.24 ± 0.04	3.07 ± 0.02
1.0	1.0 ± 0.0	0.55 ± 0.00	4.27 ± 0.03	3.07 ± 0.03
1.25	1.00 ± 0.01	0.55 ± 0.01	4.20 ± 0.08	3.10 ± 0.02
1.5	0.96 ± 0.01	0.58 ± 0.00	3.98 ± 0.12	3.08 ± 0.06
test set	n/a	0.55 ± 0.00	n/a	3.09 ± 0.02

Table 6: Prompt agreement and validity metrics for photoswitch inverse design. The mean absolute errors (MAEs) are computed with respect to the predictions of a Gaussian Process Regression as reported by Griffiths et al.[19] (compare baselines in Table 1). The mean similarity is the mean Tanimoto similarity to RDKit fingerprints in the training set.

temperature	fraction valid	fraction in training set	mean similarity	$\pi - \pi^*$ MAE / nm	$n - \pi^*$ MAE / nm
0.0	1.00 ± 0.00	0.81 ± 0.32	0.33 ± 0.01	32.61 ± 4.71	21.76 ± 4.39
0.25	1.00 ± 0.01	0.80 ± 0.19	0.32 ± 0.01	36.20 ± 4.67	20.44 ± 3.68
0.5	0.99 ± 0.02	0.65 ± 0.11	0.31 ± 0.01	41.76 ± 5.10	19.98 ± 3.41
0.75	0.97 ± 0.03	0.55 ± 0.06	0.30 ± 0.01	44.67 ± 6.52	20.05 ± 3.80
1.0	0.79 ± 0.07	0.44 ± 0.08	0.28 ± 0.01	46.38 ± 9.58	20.21 ± 4.47
1.25	0.49 ± 0.07	0.32 ± 0.11	0.28 ± 0.01	45.69 ± 13.76	18.66 ± 4.14
1.5	0.18 ± 0.05	0.20 ± 0.14	0.22 ± 0.04	45.43 ± 17.47	18.29 ± 10.16

Table 7: Hyperparameter ranges for the XGBoost classification baselines.

name	range	sampling
`n_estimators`	4–10000	uniform
`max_depth`	4.00–100.00	uniform
`learning_rate`	0.00–0.05	log uniform
`colsample_bytree`	0.20–1.00	log uniform
`subsample`	0.00–1.00	log uniform
`alpha`	1.00×10^{-6}–10.00	log uniform
`lambda`	1.00×10^{-8}–10.00	log uniform

Table 8: Hyperparameter ranges for the XGBoost regression baselines. We fixed the number of estimators to 10000.

name	range	sampling
`max_depth`	4.00–12.00	uniform
`learning_rate`	0.01–0.05	log uniform
`colsample_bytree`	0.20–0.60	log uniform
`subsample`	0.40–0.80	log uniform
`alpha`	0.01–10.00	log uniform
`lambda`	1.00×10^{-8}–10.00	log uniform
`gamma`	1.00×10^{-8}–10.00	log uniform
`min_child_weight`	10.00–1000.00	log uniform

computational resources, we did not optimize this. For all classification and regression experiments, we only considered the zero temperature, i.e., argmax output.

A.5 Baselines

Polymers As a baseline for the polymer case studies we used gradient-boosted decision tree classifiers and regressors (as implemented in XGBoost[33]), respectively. We use the feature set reported in Jablonka et al.[18] that includes statistics of the monomer sequence (composition, statistics of clusters of monomers, Shannon entropy of the sequence). We optimize the hyperparameters (using 5-fold cross-validation) of the model for 100 trials using the tree-structured Parzen estimators[34] strategy implemented in Optuna.[35] We considered the hyperparameter ranges listed in Table 7 and Table 8.

MOFs As the baseline model for the MOF case studies we used gradient-boosted decision tree classifiers and regressors (as implemented in XGBoost), respectively. We used the default feature set provided via mofdscribe[25] (including persistent homology features,[36,37] pore shape descriptors,[38] and atomic property-labeled radial distribution functions[39]). We optimize the hyperparameters (using 5-fold cross-validation) of the model for 100 trials using the tree of Parzen estimators strategy implemented in Optuna.[35] We considered the hyperparameter ranges listed in Table 7 and Table 8.

Photoswitches For the photoswitch case study, we use the model proposed by the original authors[19,20]: We describe molecules using the fragprints (Morgan fingerprints[40] augmented with 85-dimensional fragment descriptor computed with RDKit[41]) proposed in the original work and train Gaussian process regressors (GPR) with a Tanimoto kernel[42] (as implemented in GPflow[43]). We reproduce the performance metrics reported in the original work.

A.6 SMILES randomization as data augmentation

We also investigated the use of data augmentation via SMILES randomization[44–46] on the photoswitch case study using 10 random enumerations. As shown in Table 9, we do not observe significant changes in predictive performance.

Table 9: Classification metrics for the photoswitch case study using SMILES randomization as data augmentation.

no. training points	accuracy (macro)	F_1 micro	F_1 macro
no augmentation			
10	0.75 ± 0.03	0.34 ± 0.12	0.17 ± 0.07
50	0.80 ± 0.03	0.46 ± 0.14	0.29 ± 0.10
100	0.82 ± 0.04	0.51 ± 0.17	0.40 ± 0.14
200	0.86 ± 0.02	0.61 ± 0.19	0.55 ± 0.18
300	0.87 ± 0.05	0.66 ± 0.16	0.61 ± 0.12
350	0.88 ± 0.07	0.69 ± 0.23	0.63 ± 0.23
augmentation without including canonical SMILES			
10	0.77 ± 0.01	0.38 ± 0.15	0.28 ± 0.12
50	0.83 ± 0.02	0.34 ± 0.32	0.28 ± 0.26
100	0.84 ± 0.06	0.45 ± 0.34	0.44 ± 0.33
200	0.86 ± 0.09	0.61 ± 0.30	0.58 ± 0.28
300	0.88 ± 0.08	0.68 ± 0.26	0.64 ± 0.25
350	0.89 ± 0.08	0.72 ± 0.26	0.67 ± 0.27
augmentation including canonical SMILES			
10	0.77 ± 0.03	0.31 ± 0.22	0.22 ± 0.16
50	0.83 ± 0.02	0.50 ± 0.23	0.47 ± 0.22
100	0.86 ± 0.02	0.62 ± 0.18	0.59 ± 0.18
200	0.88 ± 0.04	0.62 ± 0.28	0.59 ± 0.27
300	0.89 ± 0.05	0.62 ± 0.34	0.57 ± 0.32
350	0.91 ± 0.03	0.76 ± 0.07	0.76 ± 0.11

A.7 Prompt engineering

We also tested prefixing the prompts with strings as "I'm an export polymer chemist" but did not observe increases in predictive performance. Further, systematic analyses will be the subject of future work.