nature machine intelligence

Article

https://doi.org/10.1038/s42256-023-00788-1

Leveraging large language models for predictive chemistry

In the format provided by the authors and unedited

Contents

Supplementary Note 1 Applications and Datasets

In this work, we tested the generative pre-trained transformer model 3 (GPT-3) model for a set of different applications in chemistry and material science. We selected those applications for which successful machine-learning approaches have been developed. This allows us to assess the performance of our GPT-3 approach with the state-of-the-art. We have divided these applications into three categories: properties of molecules, properties of materials, and reactions.

1.1 Molecules

Photoswitches Photoswitches are molecules that reversibly change their structure—and thus polarity—upon irradiation with light. This behavior can be useful for various applications. Incorporated into drug molecules, for instance, photoswitches might be used to control the activity of the drug molecule. Besides that, they might also be incorporated into molecular machines or used for molecular energy and information storage. See Crespi et al.[1] for a review of azobenzene photoswitches.

One would like to tailor the adsorption of a photoswitch molecule for a given application. For example, for therapeutic applications, one would like to red-shift the adsorption band to use lower energy light and minimize the potential for radiation damage. Often, one also wants to achieve some separation between the adsorption of the E and Z isomers, to ensure that they can be triggered independently.

One can tune the transition wavelength by modifying the structure. We have a reasonable intuition if we make small modifications to the structure. However, as soon as we make multiple modifications, this intuition is of limited use Griffiths et al.[2]. Hence, one has to rely on computational chemistry, via time-dependent density functional theory (DFT) (TDDFT), to obtain some guidance. However, these quantum calculations require expert knowledge and are computationally prohibitive for large-scale screenings. This motivated Griffiths et al. to develop a machine-learning approach to predict the transition wavelengths Griffiths et al.[2]

QMUGs A relevant property for many (opto)electronic application is the energetic difference between the highest occupied molecular orbital (HOMO) and lowest unoccupied molecular orbital (LUMO). Isert et al.[3] computed this (and related properties) for multiple conformers of about 500,000 molecules using different levels of theory.

Solubility Aqueous solubility is an essential characteristic of molecules, particularly for pharmacological applications. If a molecule does not dissolve in water, it will have poor bioavailability. Hence, aqueous solubility is an important factor in evaluating potential drug molecules.[4]

An experiment performed by Pat Walters[5] inspires our solubility case study. In this experiment, Walters compared different models trained on the ESOL dataset of[6] and tested

them on a dataset of pharmacologically relevant molecules, the DLS-100 dataset.[7] The models under consideration were a refitted version of the original ESOL, random forest (RF), a graph neural network (GNN) Weave modules,[8] as well as a single-layer graph convolutional neural networks.[9]

Hydration free energies A property closely related to solubility is the hydration free energy. It describes the change in free energy when a molecule is transferred from the gas phase into water. On a practical level, they are very useful for benchmarking force fields. However, they can also give insights into solvation mechanisms.[10]

Lipophilicity Similar to solubility, lipophilicity is an essential factor in pharmacokinetics.[11] It is typically described using the water-octanol partition coefficient (logD). Therefore, many works have focussed on predicting this coefficient.

Organic photovoltaics Organic photovoltaics (OPV) to provide photovoltaic systems that can be produced from Earth-abundant elements using low-energy techniques. A widely investigated device architecture consists of a p-type molecule or polymer and an n-type fullerene, forming a bulk heterojunction framework. A key performance indicator for OPV is the power conversion efficiency (PCE), i.e., the ratio of output power to input power.

1.2 Materials

Metal-organic frameworks Metal-organic frameworks (MOF) are one of the most exciting classes of materials as they promise to provide a framework for systematically designing materials across different scales.[12,13,14] They are composed of metal clusters that are connected via strong bonds to organic linkers. Due to their tunable porosity, but also the potential to tune the chemistry to a given application, they have been explored for a wide array of applications ranging from gas storage and separation, over sensing, to catalysis.

Henry coefficients For gas separations, the Henry coefficient is a key performance indicator. For diluted streams, the Henry coefficient multiplied by the pressure gives the number of adsorbed molecules. In various studies, it has been shown to be a valuable indicator for gas separation applications, e.g., carbon capture.[15]

Heat capacities If solid sorbents such as MOF are used in a carbon capture process, they must be heated to regenerate. The amount of needed heat crucially depends on the heat capacity of the materials. Moosavi et al.[16] has shown that neglecting this factor can change screening results if process performance metrics are considered.

Water stability If a MOF is used in an industrial process, it will invariably be exposed to some form of water: For example, from a humid flue gas stream or the steam used to

regenerate the material. A useful MOF must remain stable under such conditions. Unfortunately, as Burtch et al.[17] reviewed, water stability is a complex phenomenon in which a material might be both kinetically or thermodynamically stable. Since we do not have a good understanding of the degradation mechanism and the complex chemistry of MOF with large unit cells, this question can currently not be (routinely) addressed using tools from computational chemistry.

Polymers As another material case study, we considered linear polymers in a coarse-grained representation.[18] Those polymers are representative of dispersants that have various applications in industry, for instance, to stabilize the color brightness of pigments. An important performance parameter for this application is the free energy of adsorption of the polymer onto a model surface (e.g., of a pigment particle). Jablonka et al.[18] computed such free energies using mesoscale simulations. In order to build a surrogate model for the expensive simulations, Jablonka et al.[18] developed a machine learning (ML) approach based on features extracted from the monomer sequence. Besides the composition, this feature set also included cluster size statistics and measures of the entropy of the sequence and was manually tuned for optimum performance on this task.

High-entropy alloys High entropy alloys (HEA)[19,20,21] are materials that are composed of multiple elements (often > 5) in roughly equal proportions. Since their discovery, they have attracted much interest due to their promising properties, with some materials overcoming the strength-ductility tradeoff. In particular, Yeh et al.[19] proposed that the configurational entropy might stabilize the solid solutions at the expense of intermetallics (which often tend to be brittle, note, however, that the configurational entropy rule does not fully stand the test).[21] However, they also span a large chemical space that is difficult to explore using conventional techniques. A critical question is what phase(s) will form for a given alloy composition. Before the discovery for HEA Hume-Rothery put forward a set of general rules for the potential for forming solid solutions of binary alloys. Since then, various approaches have been put forward to predict the phase formation of HEA, including a ML approach by Pei et al.[22], who constructed feature vectors based on atomic properties. Here, we reuse their dataset, but simply provide GPT-3 with the composition, e.g., $Ag_{0.05}Zr_{0.95}$ as input.

Matbench tasks To ensure a fair comparison to top-performing models in material science, we also considered the tasks from the MatBench suite that are based on compositions.[23]

Bulk metallic glass formation ability The ability of a material to form glasses is known as the glass formation ability and is linked to the chemical composition. Dunn et al.[23] extracted a dataset of composition and the bulk metallic glass formation ability from the Landolt–Börnstein collection.[24]

Metallicity In addition, we also considered classifying compositions as metallic or non-metallic. For this Dunn et al.[23] extracted a dataset reported by Zhuo et al.[25]

Experimental band gaps Zhuo et al.[25] also extracted a dataset of compositions and band gaps of inorganic solids from Zhuo et al.[25]

Yield strength of steel For structural materials, it is essential to know at which stress it ceases to show elastic behavior. That is, above the yield point, a deformation will be permanent. Dunn et al.[23] extracted this information for steels from Citrine informatics.

1.3 Reactions

Very central to the central science are chemical reactions. A lot of time is often spent optimizing conditions to increase the yield. Recently, there have been many attempts to expedite this process using ML approaches such as Bayesian optimization (BO). In many works, it was found that one-hot encoding performs equivalent to chemically informed representations.[26,27,28] However, it would be much more convenient if the reactants were simply provided as text without preprocessing. Here, we focus on two datasets. First, experimentally determined yields of Pd-catalysed Buchwald–Hartwig C–N crosscouplings reported by Ahneman et al.[29] and, second, yields for Pd-catalysed Suzuki-Miyaura C-C cross-couplings reported by Perera et al.[30]

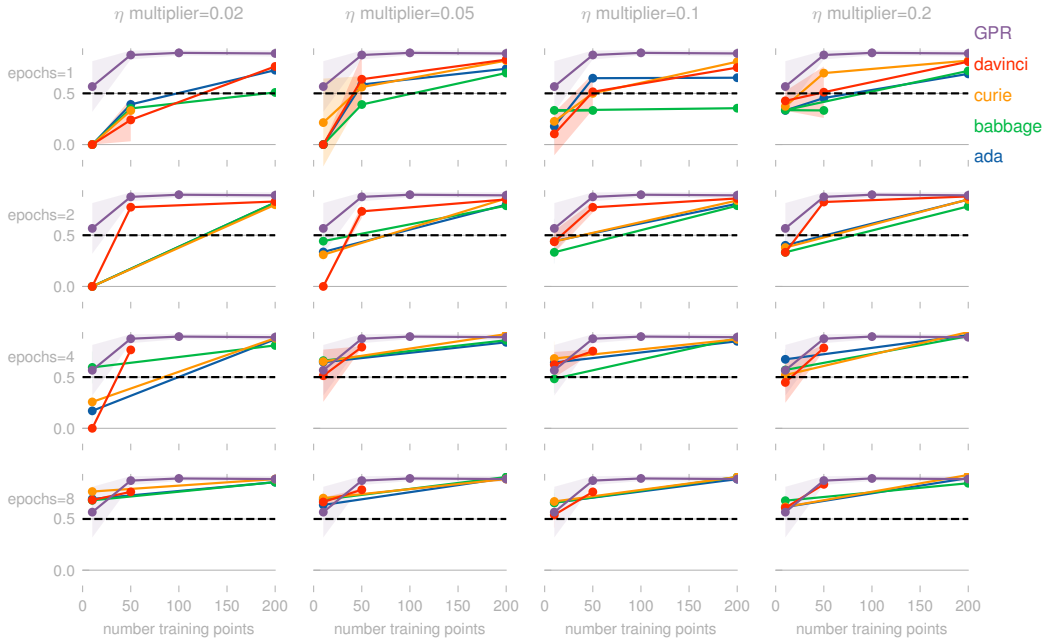
Supplementary Note 2 Fine-tuning parameters

As an initial experiment, we investigated the impact of a range of fine-tuning parameters. We focussed on balanced binary classification on the photoswitch using simplified molecular-input line-entry system (SMILES).

In Supplementary Figure 1 we observe that the fine-tuning parameters can have a pronounced effect on the classification performance. Particularly for only a few passes through the data (low number of epochs) and a low learning rate (η) the resulting model might even perform worse than random guessing.

Interestingly, we do not find the largest model to perform consistently better.

We decided to use the same settings for all subsequent experiments instead of optimizing them for every experiment. While this might limit the performance we observe, it also avoids overfitting.



Supplementary Figure 1 | Influence of fine-tuning parameters on the model performance. Error bands show the standard error of the mean.

Note that the model must also use the few examples we provide to learn the prompt structure. From Dinh et al. [31] we know that pre-training with prompts filled with random data can help the model learn.

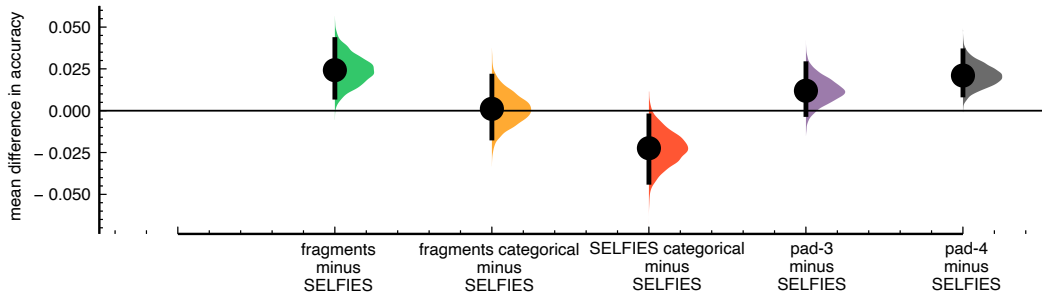
Supplementary Note 3 Fine tuning cost

For the data in Extended Data Table 1, the fine-tuning of the `ada` model for eight epochs currently costs 176 token $\cdot\ 0.0000004\$$ token$^{-1} \cdot 8 = 0.0005632\$$. The total cost for all experiments in the study was less than seven thousand dollars, where a large fraction of the money is related to using GPT-4 or the davinci model.

Supplementary Note 4 Influence of the molecular representation

There is not one unique way of representing molecules in text form. While the International Union for Pure and Applied Chemistry (IUPAC) name might have widespread use for the communication of synthesis protocols or papers, cheminformatics focussed on line representations such as SMILES,[32] INChI, and recently, SELFIES.[33] For an overview, see Krenn et al.[34].

To understand possible representation effects better, we investigated other possible representations of the molecules in the photoswitch dataset. First, we fragmented the molecules using extended functional group (EFG).[35] We then directly used those fragment SMILES as representation but also investigated the removal of chemical information using a numerical encoding. In addition, we also removed explicit chemical information from SELFIES by replacing the characters using a numerical encoding. To investigate potential sequence length effects, we also investigated padding the SELFIES characters. In Supplementary Figure 2, we show that both chemistry and the sequence length influence the predictive performance. Removing explicit chemistry information tends to decrease the performance, and reshaping it into relevant "buckets" via fragments tends to increase the performance. This indicates that while the fine-tuning of GPT-3 gives performance that is competitive with strong baselines, unlocking the ultimate power still requires, as is the case with all models, fine-tuning of the representation. In the case of large language model (LLM), this implies tuning prompt and string representations in contrast to conventional feature engineering.



Supplementary Figure 2 | Mean effect sizes for accuracy on the photoswitch case study for different representations. Top row shows the accuracy obtained in independent runs with different representations. The bottom row shows bootstrapped (5000 resamples) mean effect sizes versus the SELFIES baseline. We created the figures using the DABEST library.[36]

Supplementary Note 5 In-context learning

As Brown et al.[37] showed, LLM are few-shot learners that can generalize to new tasks without any gradient-based learning (e.g., fine-tuning).

For this reason, we also investigated this setting. To do so, we focused on classification on the photoswitch dataset and prompted different versions of GPT-3 with prompts of the form below:

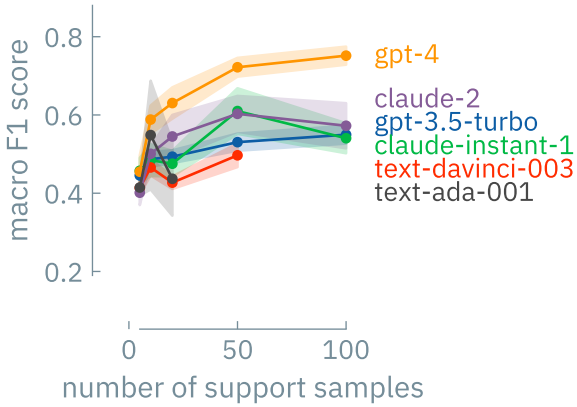
```
What is {property_name} of {queries} given the examples below?
Answer with a comma-separated list of predictions for
those {number} molecules.

Examples:
{examples}

Constraint:
Make sure to return exactly {number} comma separated predictions.
The predictions should be one of {allowed_values}.
Return only the predictions.

Answer:
```

Where `queries` is a list of molecules and `examples` is a list of examples of molecules with associated properties.

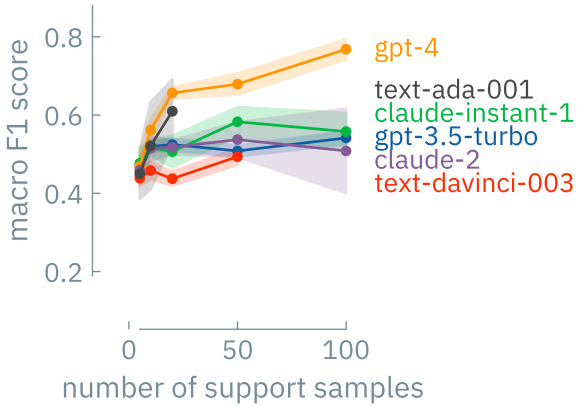
As evident from Supplementary Figure 3 and Supplementary Figure 4 we find that also with this approach, we find good results, competitive, or even outperforming the Gaussian process regression (GPR) baseline (also see Supplementary Note 6). Interestingly, we also observe a stronger dependence on the representation of the molecules when we use this prompt-based approach without fine-tuning. However, a fundamental limitation of this approach is that the context length of the model limits the number of examples one can provide. For this reason, the learning curves also add at different points (with larger models having larger context sizes and different representations needing a different number of tokens).

In addition to the OpenAI models, we also investigated Anthropic's Claude-v1 model.

These encouraging results motivate further work using other prompting strategies, including soft prompting,[38] such as instruction prompt tuning.[39]



Supplementary Figure 3 | Macro F_1 score on the photoswitch dataset for binary classification with few-shot prompts. Sampling at a temperature of 0.2. Colors indicate different models. For some models, the learning curves stop earlier due to their limited context window. Error bands show the standard error of the mean.



Supplementary Figure 4 | Macro F_1 score on the photoswitch dataset for binary classification with few-shot prompts. Sampling at a temperature of 0.8. Colors indicate different models. For some models, the learning curves stop earlier due to their limited context window. Error bands show the standard error of the mean.

Supplementary Note 6 Classification experiments

6.1 Note about baselines

We use MolCLR and TabPFN as baselines for the classification case studies in most cases.

MolCLR Wang et al.[40] proposed the Molecular Contrastive Learning of Representations via Graph Neural Networks approach in which GNN are pre-trained on large unlabeled datasets using a contrastive loss and graph augmentations such as atom masking, bond deletion, and subgraph removal. They could show that this pre-training improves the performance of GNN on various property prediction tasks. In this work, we fine-tune MolCLR on all tasks with the default settings reported by the original authors.

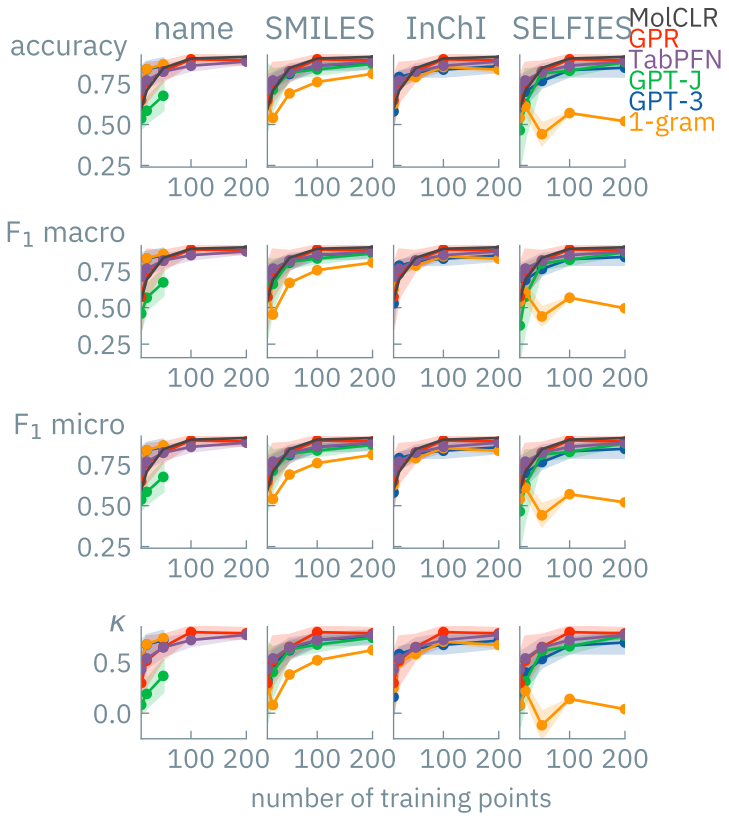
TabPFN tabular Prior-Data Fitted Network (PFN) (TabPFN)[41] is a transformer (25.82 million parameters) that has been pre-trained on millions of synthetic datasets (D_i sampled from some prior $p(\mathbf{D})$ with the task of predicting held-out points with a forward pass (i.e., it predicts posterior predictive distribution (PPD) $q_\theta(y_{\text{test}}|x_{\text{test}}, D)$). This can be thought of as gradient-based meta-learning.

GPR with fragprints For datasets that have SMILES available, we also use GPR models with Tanimoto kernels with "fragprint" descriptors, which are concatenations of fragment features and fingerprints.[2,42]

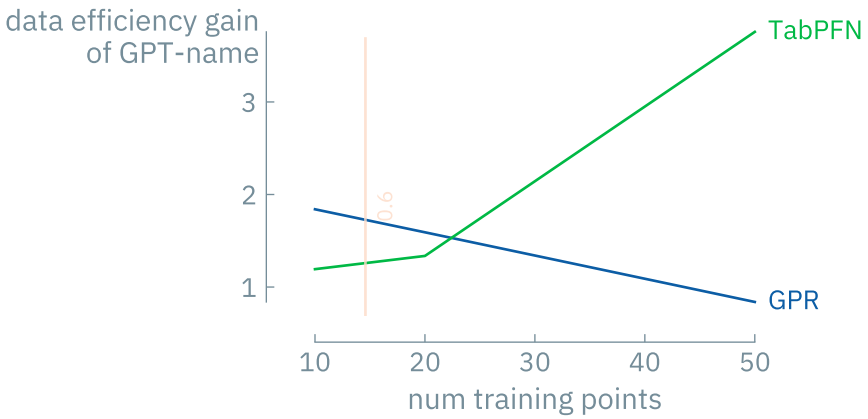
For specific case studies, there are additional baselines we took from the literature and which we describe in the corresponding section.

6.2 Photoswitches

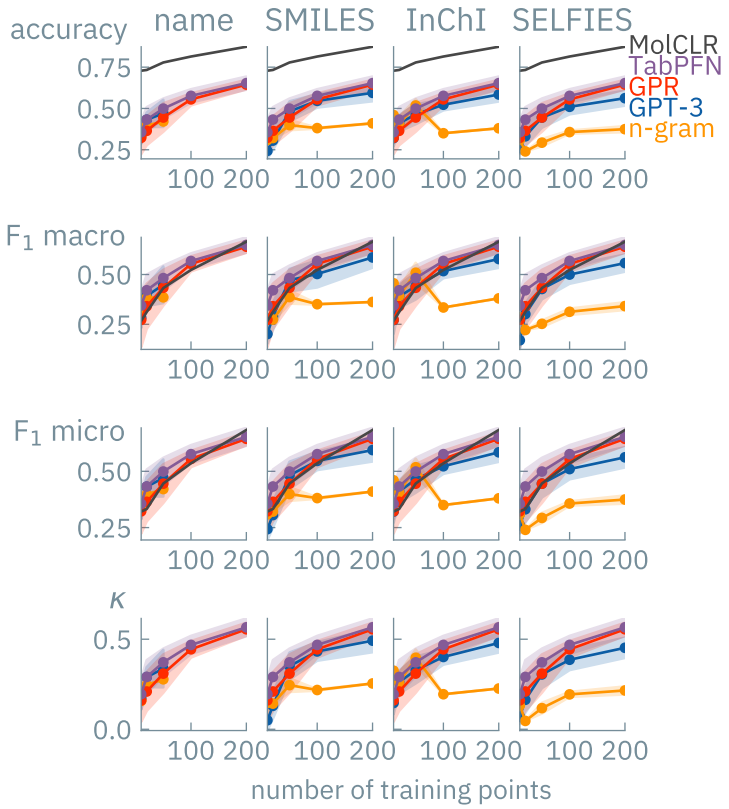
The learning curves for classification are shown in Supplementary Figure 5 and Supplementary Figure 7. Particularly for the binary classification, the GPT-3 model fine-tuned on IUPAC-names outperforms the baselines.



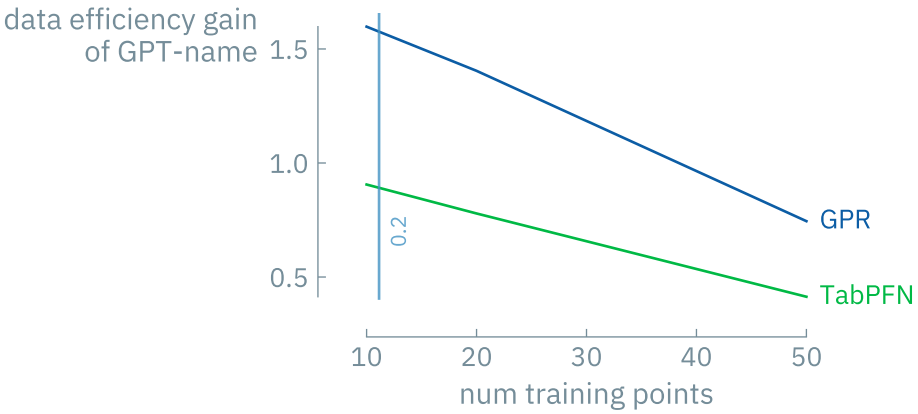
Supplementary Figure 5 | Learning curves for binary classification on the photoswitch dataset. We did a balanced split in two classes to predict the π-π^* transition wavelength of the *E* isomers. Error bands show the standard error of the mean.



Supplementary Figure 6 | Learning curve intersection points for binary classification on the photoswitch dataset. Vertical lines indicate the κ scores of the GPT-3 model.



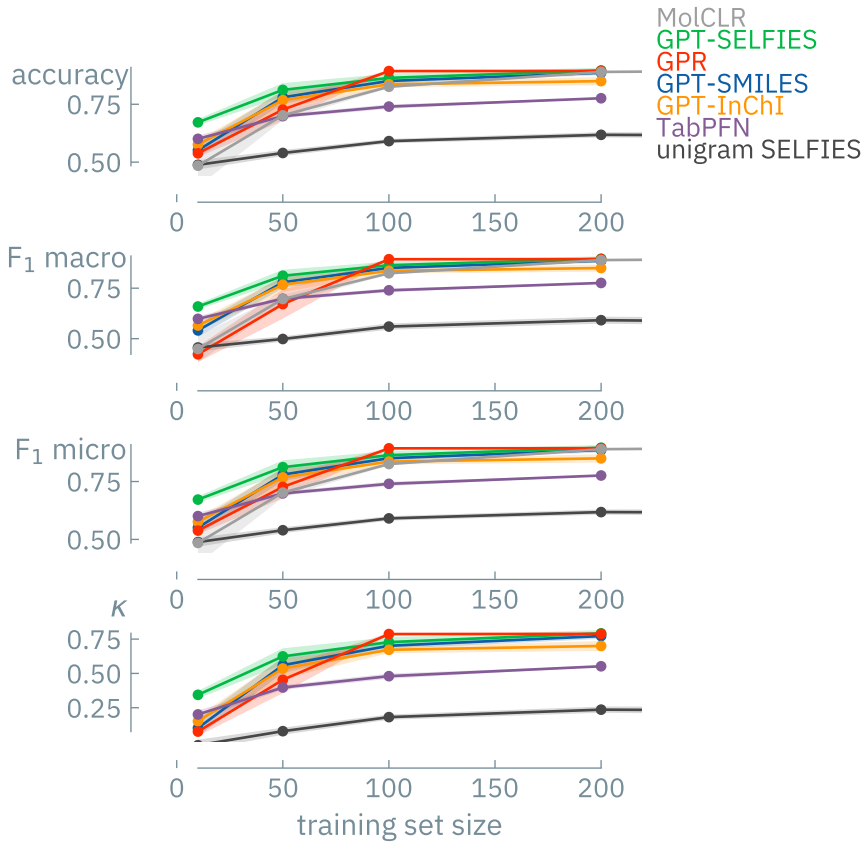
Supplementary Figure 7 | Learning curves for 5-class classification on the photoswitch dataset. We did a balanced split in five classes to predict the π-π^* transition wavelength of the *E* isomers. Error bands show the standard error of the mean.



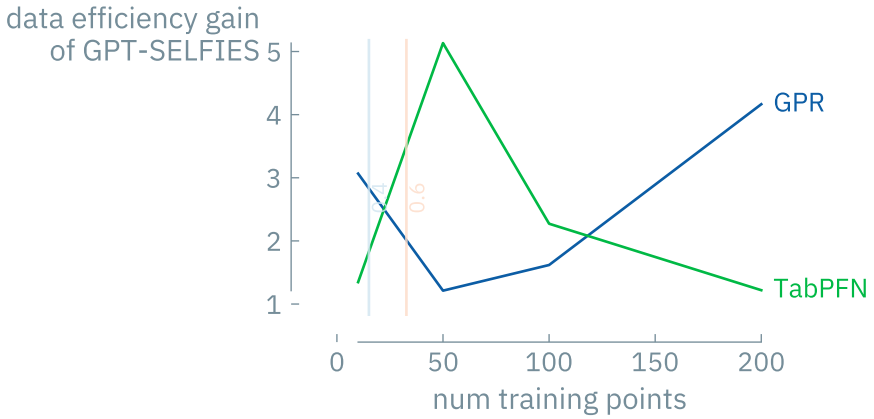
Supplementary Figure 8 | Learning curve intersection points for binary classification on the photoswitch dataset. Vertical lines indicate the κ scores of the GPT-3 model.

6.3 Free energy of solvation

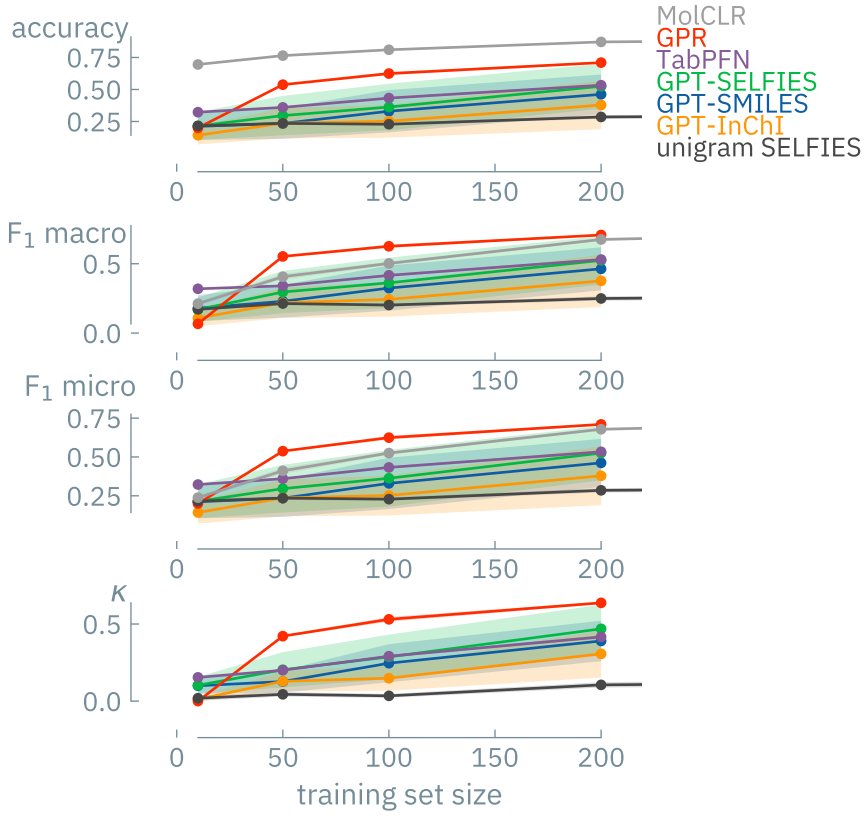
The learning curves for the classification tasks are shown in Supplementary Figure 9 and Supplementary Figure 11. Here, we find GPT-3 fine-tuned using the SELFIES representation to perform well.



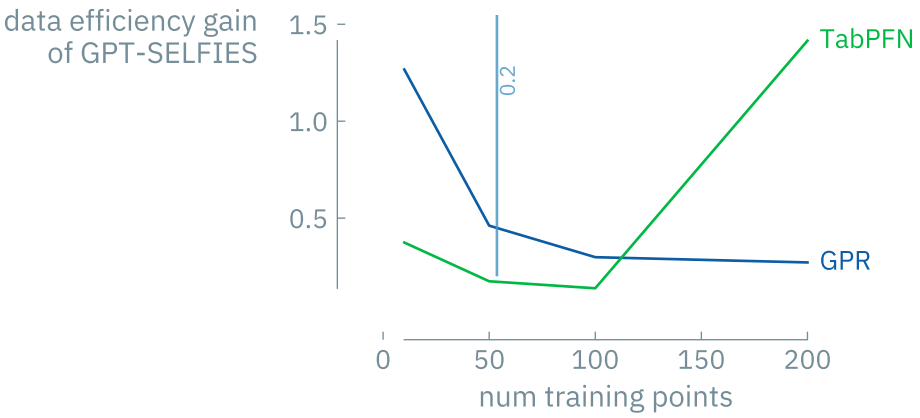
Supplementary Figure 9 | Learning curves for binary classification on the FreeSolv dataset. We performed a balanced split into two classes. Error bands show the standard error of the mean.



Supplementary Figure 10 | Learning curve intersection points for binary classification on the FreeSolv dataset. Vertical lines indicate the κ scores of the GPT-3 model.



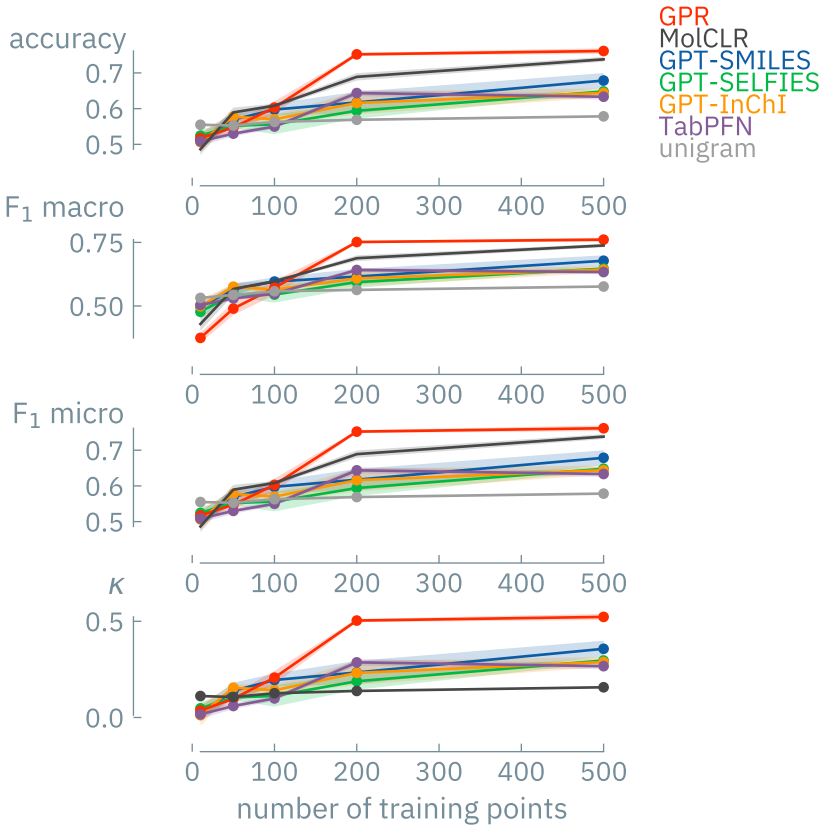
Supplementary Figure 11 | Learning curves for 5-class classification on the FreeSolv dataset. We performed a balanced split into five classes. Error bands show the standard error of the mean.



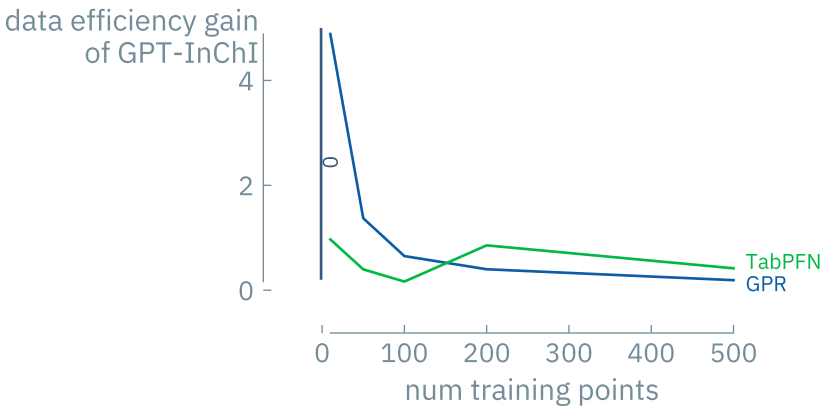
Supplementary Figure 12 | Learning curve intersection points for 5-class classification on the FreeSolv dataset. Vertical lines indicate the κ scores of the GPT-3 model.

6.4 Lipophilicity

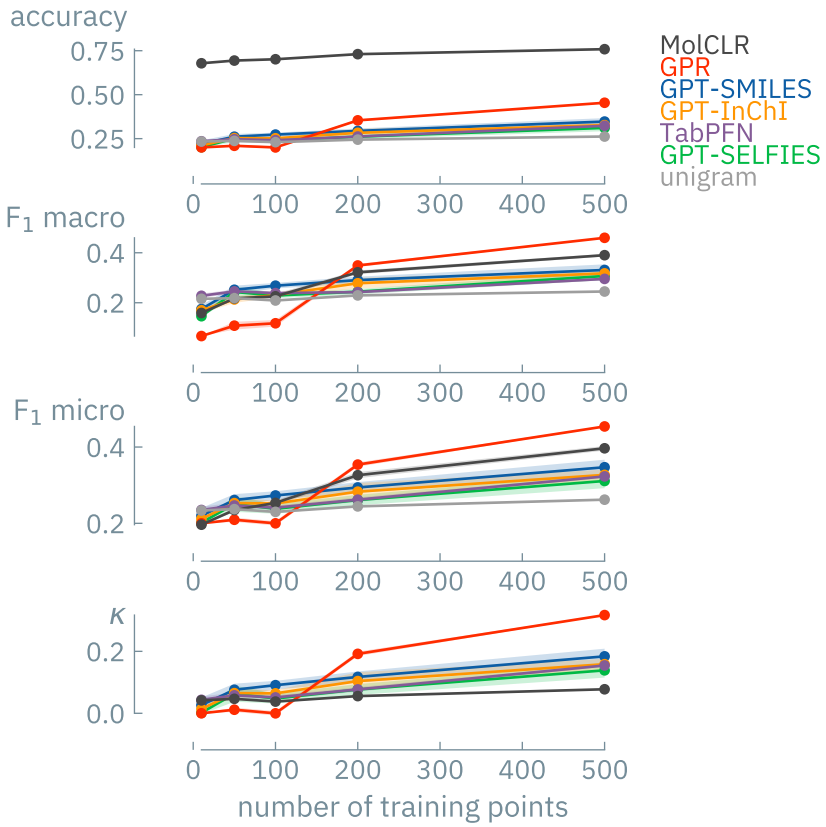
The learning curves for the classification tasks are shown in Supplementary Figure 13 and Supplementary Figure 15.



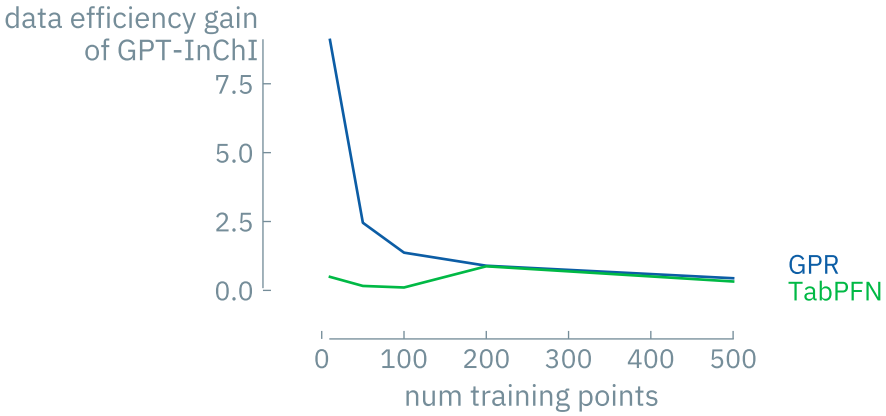
Supplementary Figure 13 | Learning curves for binary classification on the lipophilicity dataset. Error bands show the standard error of the mean.



Supplementary Figure 14 | Learning curve intersection points for binary classification on the lipophilicity dataset. Vertical lines indicate the κ scores of the GPT-3 model.



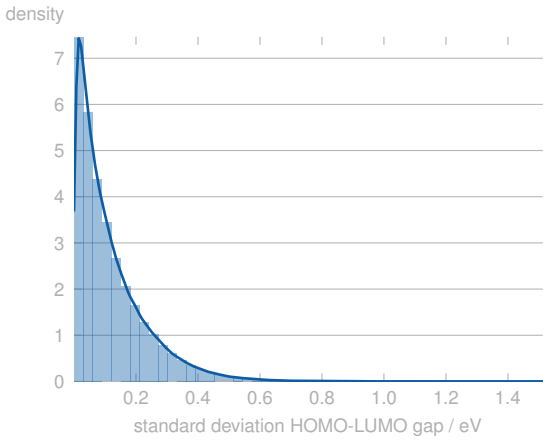
Supplementary Figure 15 | Learning curves for 5-class classification on the lipophilicity dataset. Error bands show the standard error of the mean.



Supplementary Figure 16 | Learning curve intersection points for 5-class classification on the FreeSolv dataset. Vertical lines indicate the κ scores of the GPT-3 model.

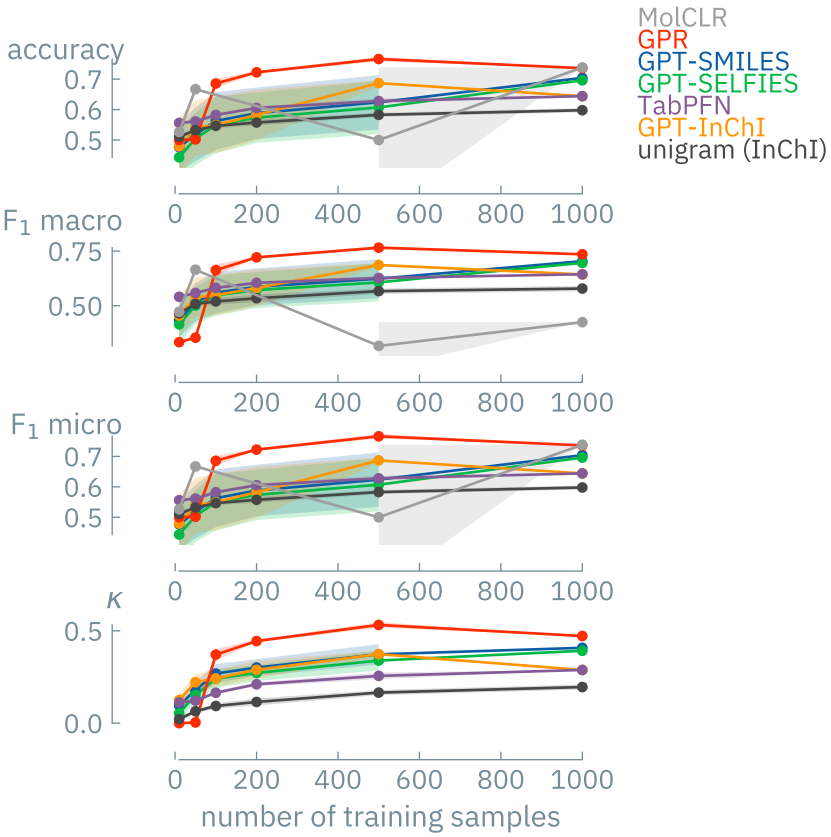
6.5 HOMO-LUMO gaps

First, it is essential to note that different conformers have different gaps between HOMO and LUMO. The distribution of the standard deviation of the HOMO-LUMO gaps for different conformers of a molecule in the QMUGs dataset is shown in Supplementary Figure 17.

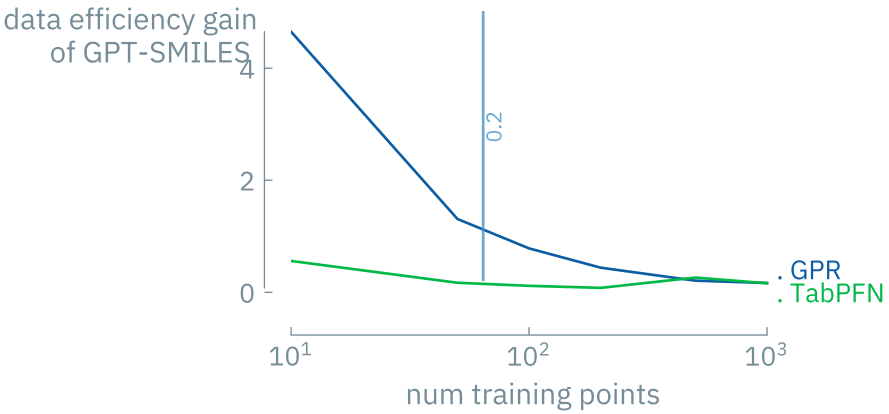


Supplementary Figure 17 | Distribution of standard deviations of the HOMO-LUMO gap of the three conformers of the molecules in the QMUGs dataset.

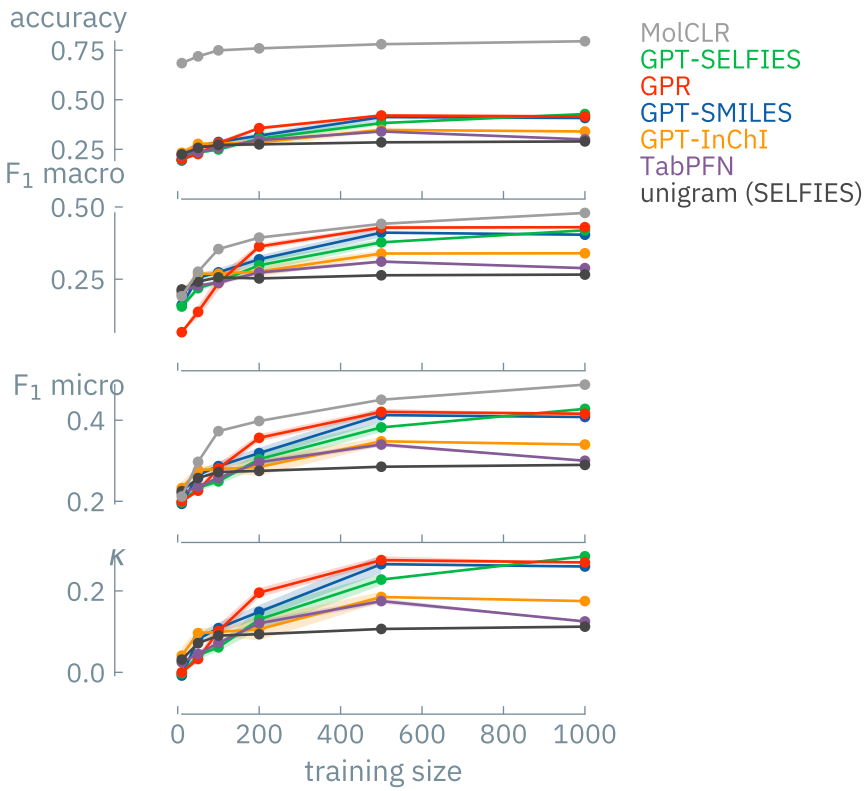
Since we base our model on line-representations, we cannot resolve those effects and train our models on the average HOMO-LUMO gap of the three conformers reported by Isert et al.[3] The learning curves for the classification tasks are shown in Supplementary Figure 18 and Supplementary Figure 20.



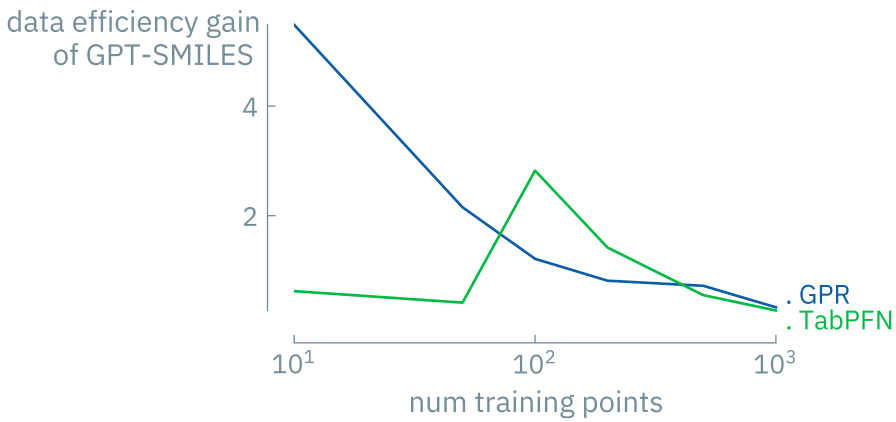
Supplementary Figure 18 | Learning curves for binary classification of HOMO-LUMO gaps on the QMUG dataset. Error bands show the standard error of the mean.



Supplementary Figure 19 | Learning curve intersection points for binary classification on the QMUG dataset.



Supplementary Figure 20 | Learning curves for 5-class classification of HOMO-LUMO gaps on the QMUG dataset. Error bands show the standard error of the mean.

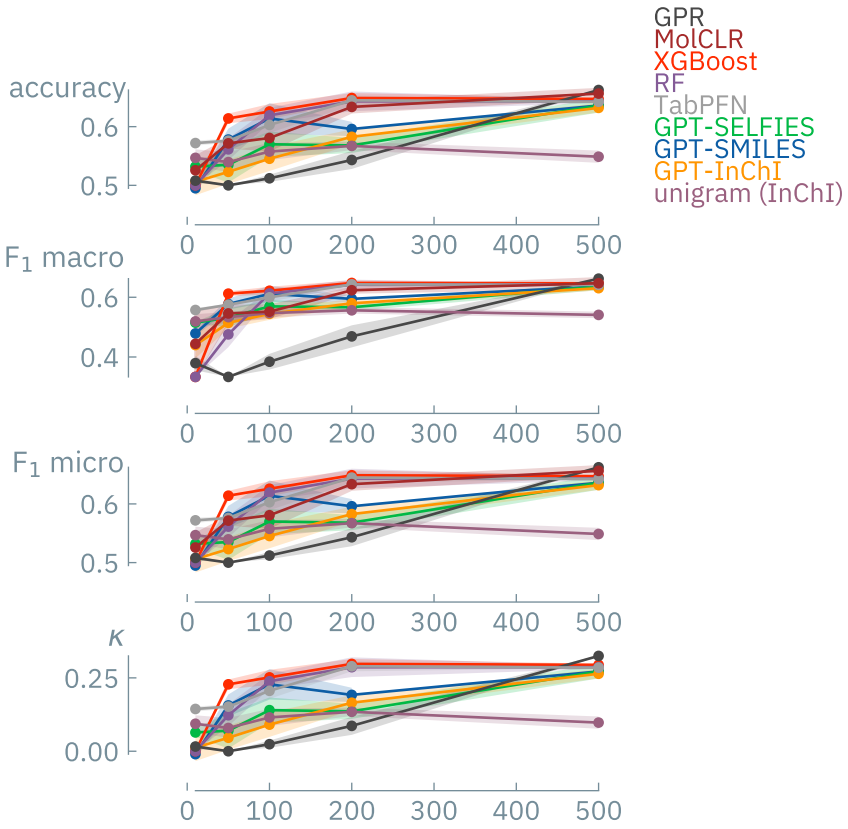


Supplementary Figure 21 | Learning curve intersection points for 5-class classification on the QMUG dataset.

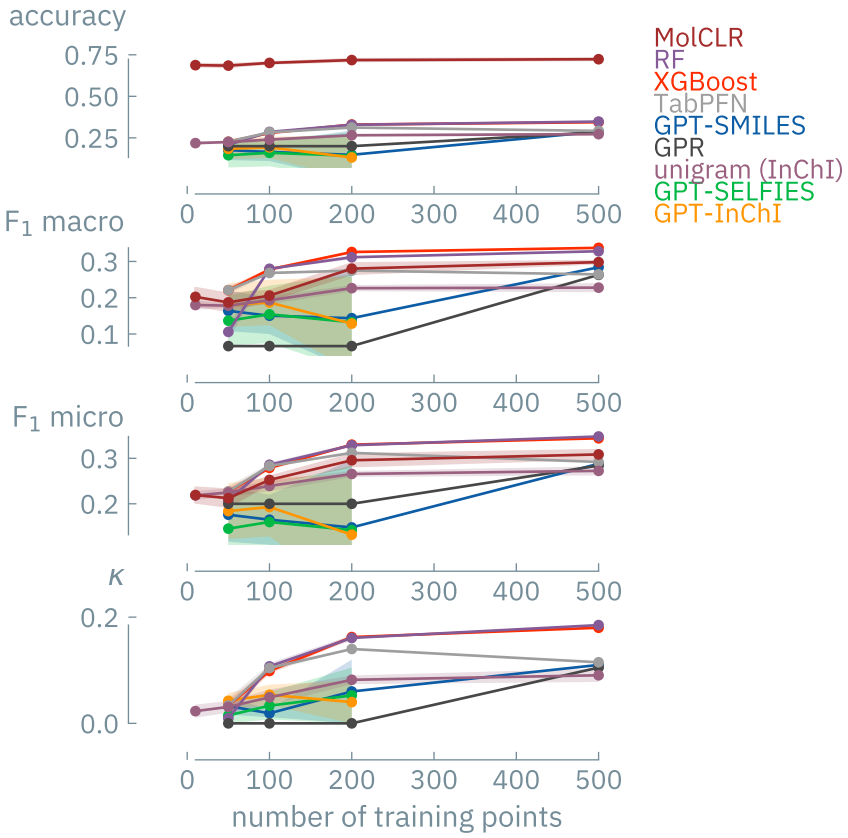
6.6 Photoconversion efficiencies in organic photovoltaics

Nagasawa et al. [43] reported a dataset of performance of around 1000 conjugated molecules compiled from the experimental literature. They also reported machine learning models for which they found good predictive performance for RF models using a extended connectivity fingerprint (ECFP). In their dataset, there are some duplicated entries which we grouped and aggregated using the mean value. Since the authors did not report the hyperparameters they used, we also implemented 5-fold cross-validated hyperparameter optimization on a RF model as a baseline and completed with the baselines we use throughout all case studies.

The learning curves for the classification tasks are shown in Supplementary Figure 22 and Supplementary Figure 23.

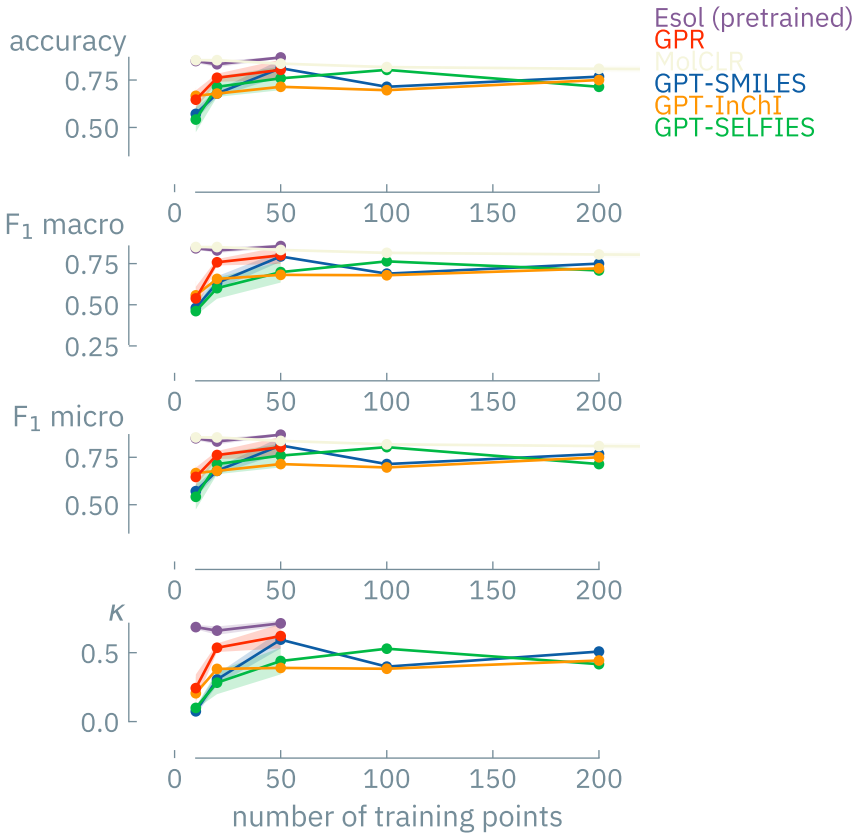


Supplementary Figure 22 | Learning curves for binary classification on the OPV dataset. We performed a balanced split into two classes based on the average PCE. Error bands show the standard error of the mean.

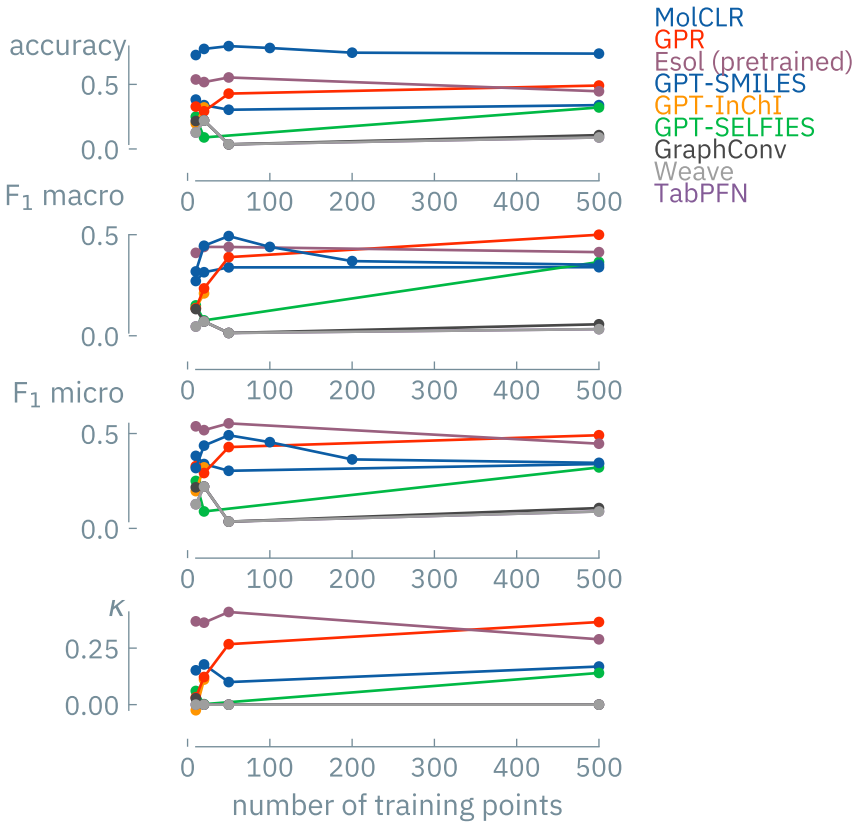


Supplementary Figure 23 | Learning curves for 5-class classification on the OPV dataset. We performed a balanced split into five classes based on the average PCE. Error bands show the standard error of the mean.

6.7 Solubility



Supplementary Figure 24 | Learning curves for binary classification on the solubility dataset. Error bands show the standard error of the mean.



Supplementary Figure 25 | Learning curves for five-class classification on the solubility dataset. Ordering of lines based on the scores for the largest training set.

Supplementary Tab. 1: ROC-AUC of Tox21 for fine-tuning on 6000 points. Error bars indicate the standard error of the mean.

target	ROC-AUC
NR-AR	0.68 ± 0.02
NR-AhR	0.73 ± 0.01
NR-ER-LBD	0.70 ± 0.01
NR-AR-LBD	0.62 ± 0.002
SR-ATAD5	0.58 ± 0.02
mean	0.67 ± 0.01

Supplementary Tab. 2: ROC-AUC of Tox21 for fine-tuning on 6000 points. Error bars indicate the standard error of the mean.

target	ROC-AUC
NR-AR	0.74 ± 0.02
NR-AhR	0.73 ± 0.01
NR-ER-LBD	0.59 ± 0.01
NR-AR-LBD	0.61 ± 0.03
NR-Aromatase	0.51 ± 0.01
NR-ER	0.59 ± 0.01
NR-PPAR-gamma	0.56 ± 0.06
SR-ATAD5	0.50 ± 0.002
SR-ARE	0.59 ± 0.01
mean	0.60 ± 0.02

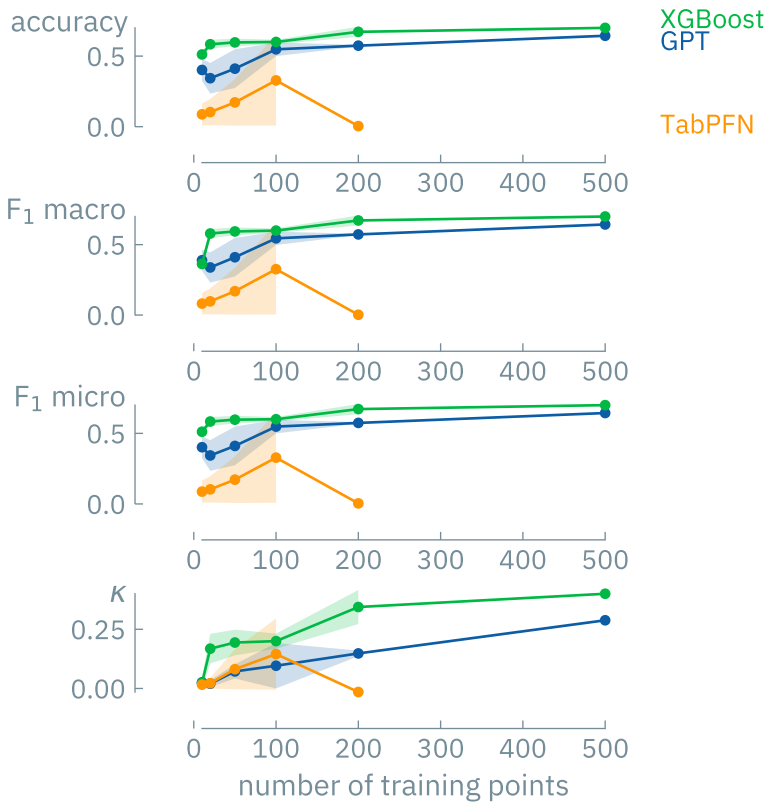
6.8 Tox21

The Tox21[44,45] set contains binary labels for 12 toxicological experiments. The datasets are highly imbalanced. We used scaffold splits for this case study.

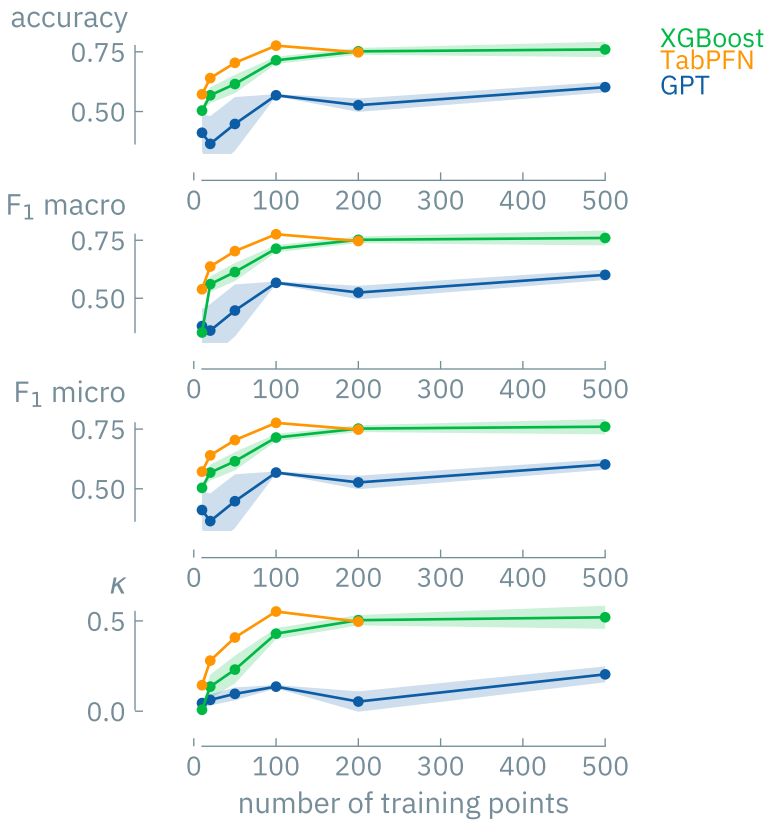
6.9 MOF Henry coefficients

6.9.1 Carbon dioxide and methane

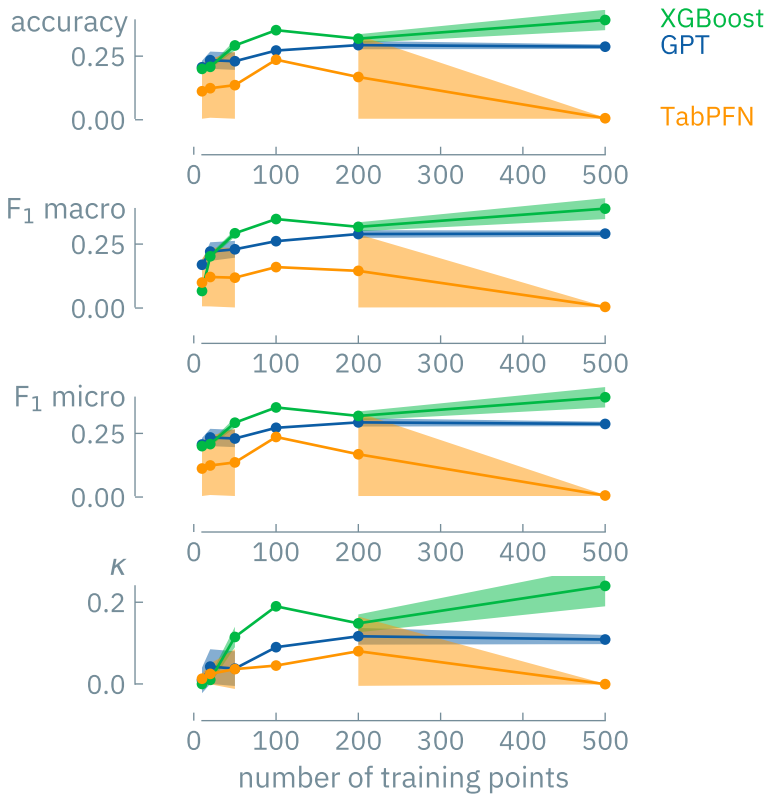
To fairly compare our results to a strong baseline, we reuse the features and models reported by Moosavi et al.[46] Note that the models reported by Moosavi et al.[46] were extensively tuned on the target task with specifically engineered features. Since the current implementation of TabPFN can only handle 100 features, we selected the 100 most relevant features according to the feature importance of a RF model.



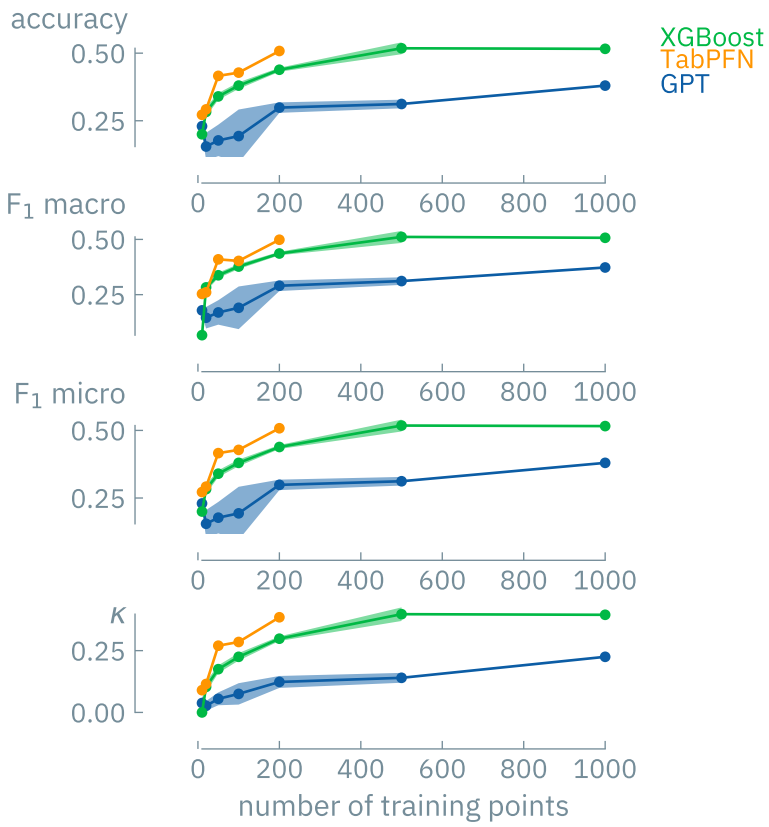
Supplementary Figure 26 | Learning curves for binary classification of CO_2 Henry coefficients. Error bands show the standard error of the mean.



Supplementary Figure 27 | Learning curves for binary classification of CH_4 Henry coefficients. Error bands show the standard error of the mean.



Supplementary Figure 28 | Learning curves for 5-class classification of CO_2 Henry coefficients. Error bands show the standard error of the mean.

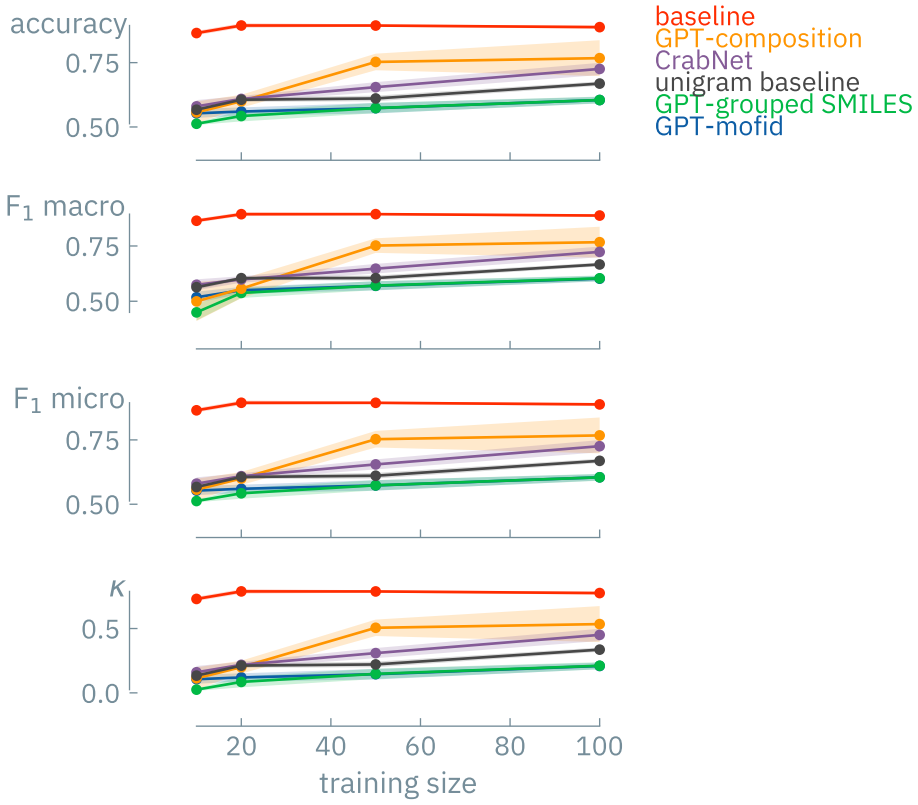


Supplementary Figure 29 | Learning curves for 5-class classification of CH_4 Henry coefficients. Error bands show the standard error of the mean.

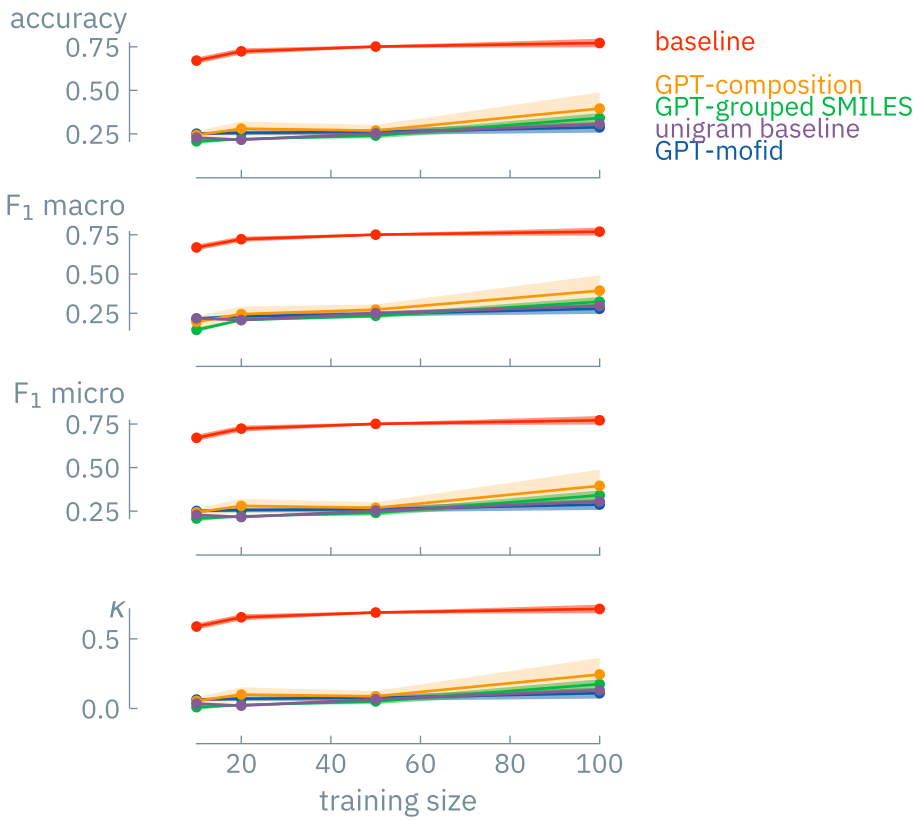
6.10 MOF heat capacities

We use data reported by Moosavi et al.[16]. We also use the model reported by Moosavi et al.[16], for which they extensively optimized hyperparameters. Notably, this model can achieve good data efficiency because it has been trained on local environments. Additionally, this model is a bootstrapped ensemble, which we replicated for our baseline. To ensure a fair comparison, we only trained on MOF and not covalent-organic frameworks (COF) and zeolites. Additionally, we only considered those MOF for which we could determine a mofid.[47] Note that, similar to Moosavi et al.[16] we did not utilize advanced splitting strategies other than stratification on the gravimetric heat capacity. Additionally, we performed the split on the structures and not on the sites and dropped duplicates based on the mofid.

Additionally, we investigated a composition-only approach (reduced formula of the primitive cell). As a baseline for this, we considered CrabNet[48].



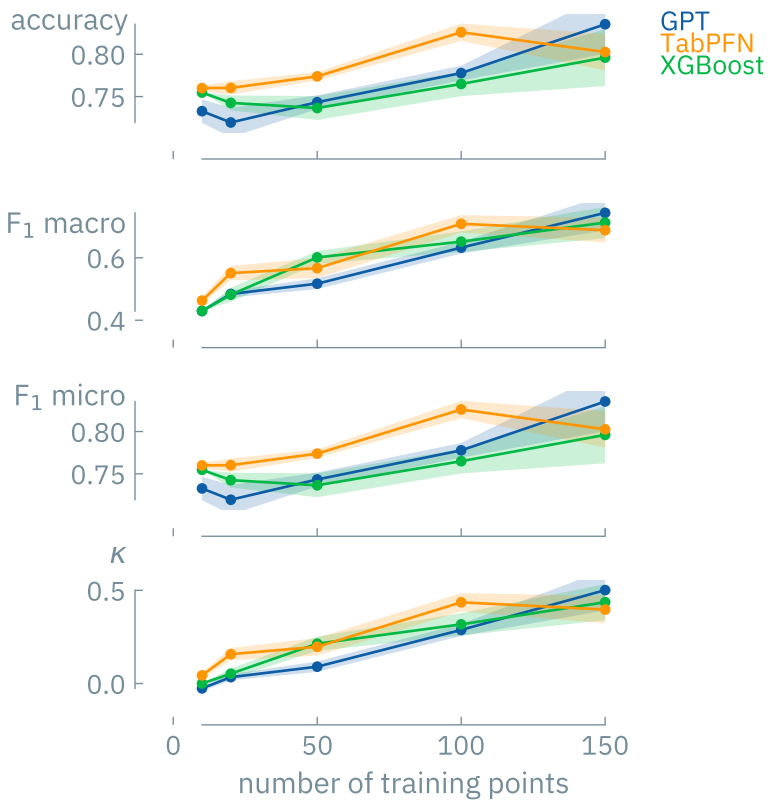
Supplementary Figure 30 | Learning curve for binary classification of MOF heat capacities. Error bands show the standard error of the mean.



Supplementary Figure 31 | Learning curve for 5-class classification of MOF heat capacities. Error bands show the standard error of the mean.

6.11 MOF water stability

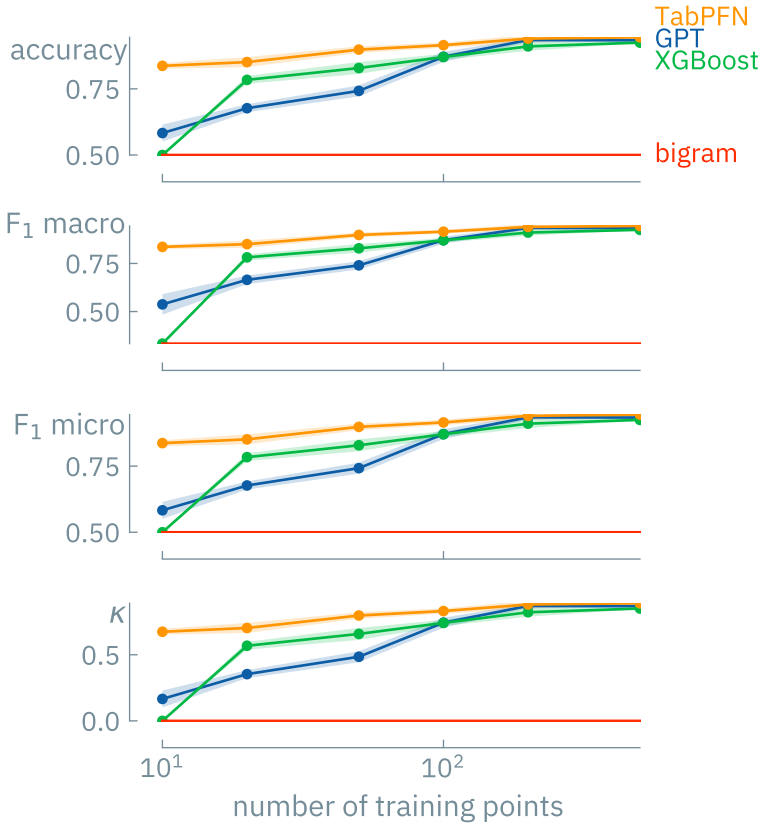
As baselines we use a gradient-boosted classifier and TabPFN on the features reported by Batra et al.[49] We follow the binary classification setting reported by Batra et al.[49] in which the kinetically and thermodynamically stable MOFs are merged into one class.



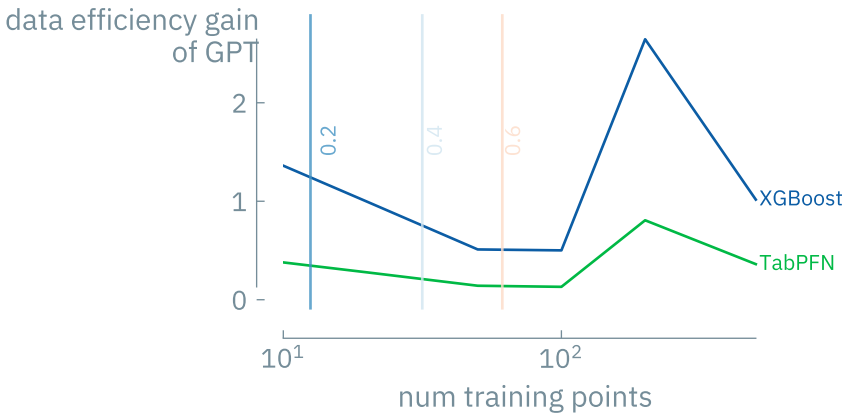
Supplementary Figure 32 | Learning curve for the classification of the water stability of MOFs. Error bands show the standard error of the mean.

6.12 Linear polymers

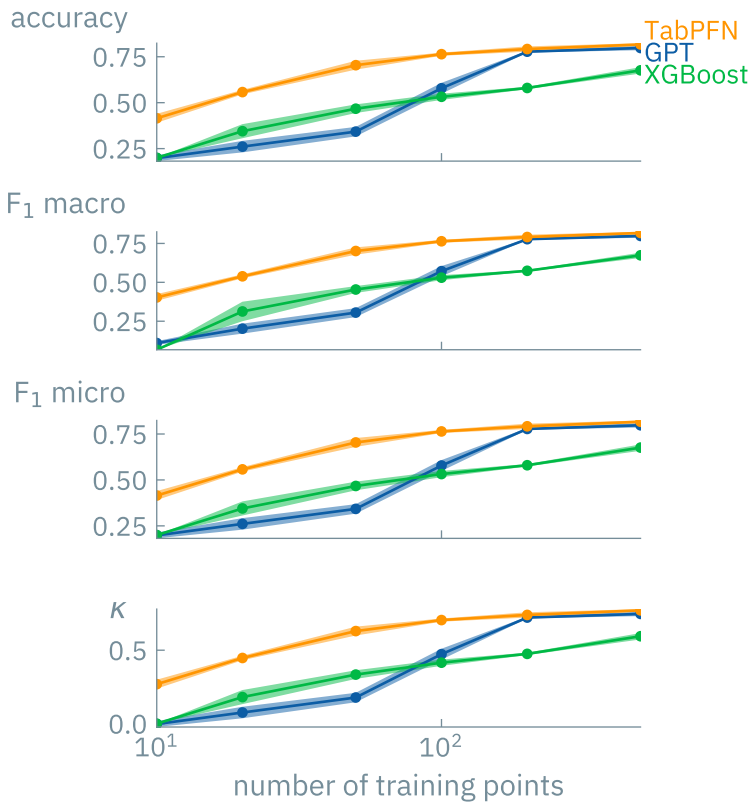
We investigated balanced binary and five-class classification. Supplementary Figure 33 and Supplementary Figure 35 show learning curves.



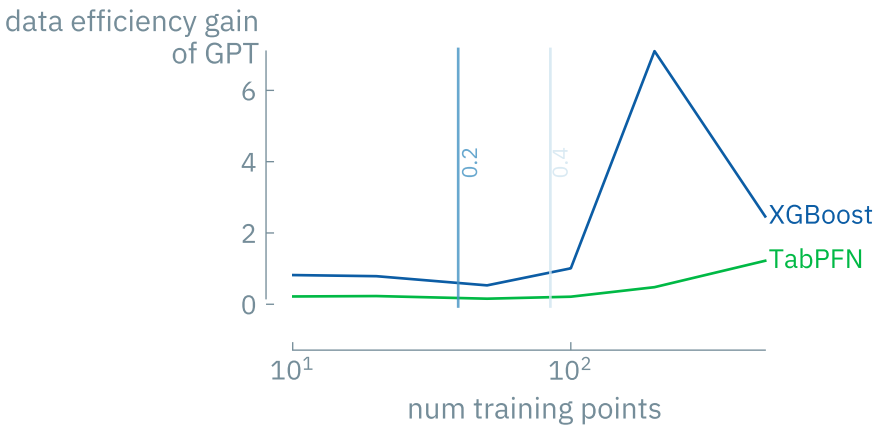
Supplementary Figure 33 | Classification performance for binary classification of free energy of adsorption of linear polymers. Error bands show the standard error of the mean.



Supplementary Figure 34 | Learning curve intersection points for binary classification on the polymer dataset. Vertical lines indicate the κ scores of the GPT-3 model.



Supplementary Figure 35 | Classification performance for classification of free energy of adsorption of linear polymers into five classes.

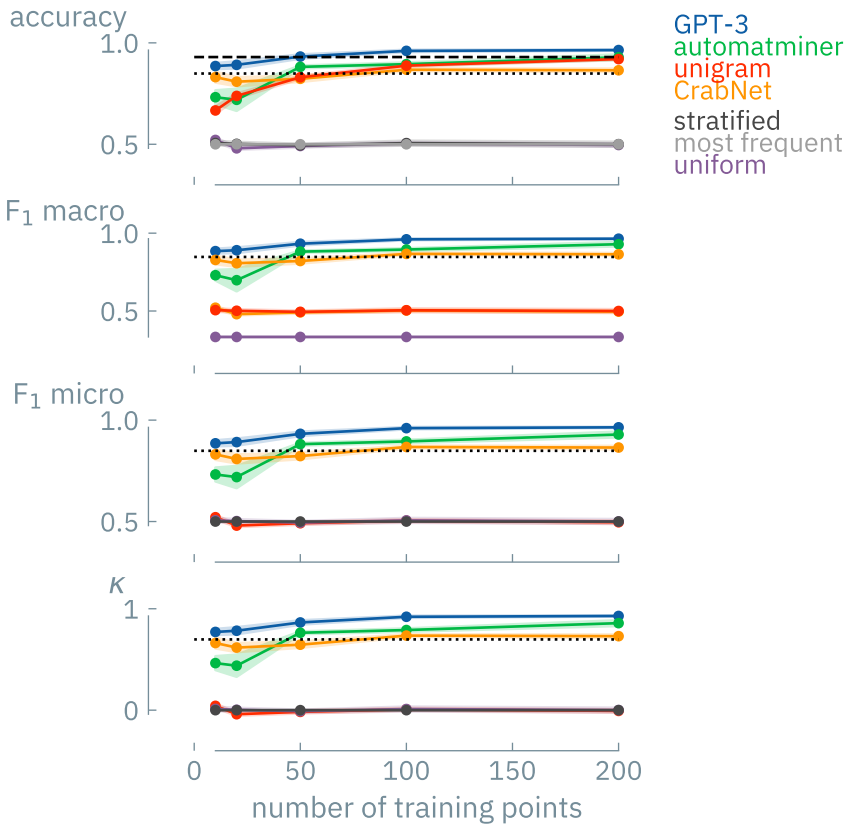


Supplementary Figure 36 | Learning curve intersection points for 5-class classification on the photoswitch dataset. Vertical lines indicate the κ scores of the GPT-3 model.

6.13 High-entropy alloys

6.13.1 Single- vs. multi-phase

Unfortunately, Pei et al.[22] did not report learning curves or a code implementation. However, we find that with very few points and without any feature engineering, we can match their predicted performance. The dataset is balanced. As an alternative baseline, we use automated ML, as implemented in the automatminer package[23] with the "express" preset.



Supplementary Figure 37 | Classification performance for classifying HEAs as "single-phase" and "multiphase", respectively. The dashed horizontal line indicates the performance reported in Pei et al.[22] using a dataset of 1252 points and 10-fold cross-validation, i.e., corresponding to a training set size of around 1126 points. Automatminer baseline computed using "express" preset. Error bands show the standard error of the mean.

As an additional deep-learning-based baseline, we considered CrabNet[48] if the default model architecture (due to the computational cost of comprehensive hyperparameter optimization and the fact that it performed favorably on matbench on different tasks with the same hyperparameters).

Testing on data not found on the internet Since one might suppose that some of the data appeared in the pertaining corpus we performed an additional experiment in which we

Supplementary Tab. 3: Zero-shot performance of the ada model for classification of alloy compositions into single and multiple phase.

metric	value
κ	-0.009
accuracy	0.66
F_1 macro	0.22

only tested on composition for which we did not find an exact using Google Search (about 24.00 %, via the SerpAPI).

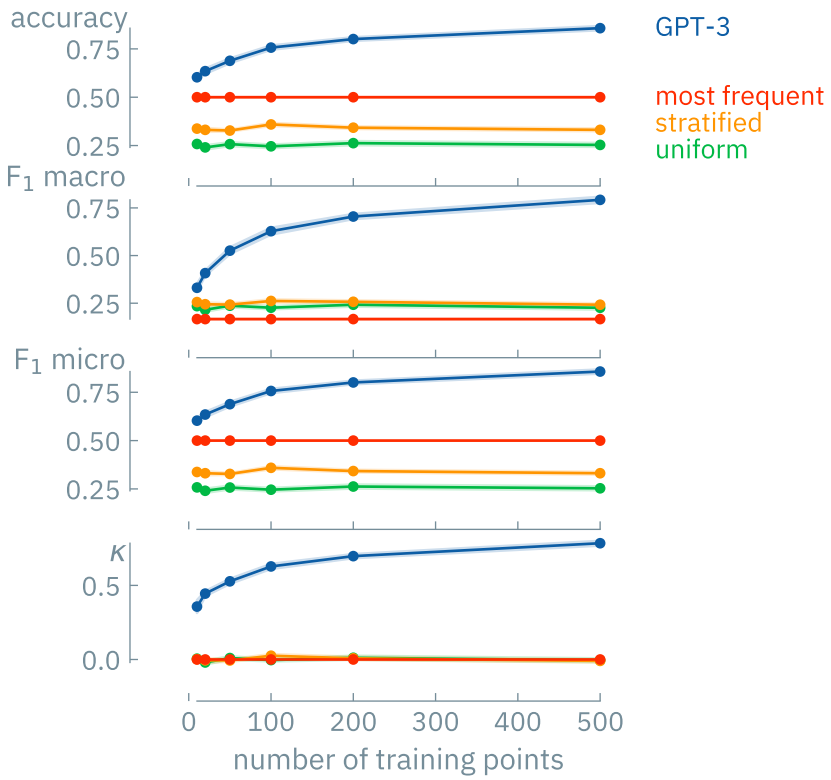
We also queried the model in zero shot setting and found the metrics listed in Supplementary Table 3.

6.13.2 Multiphase, vs. hcp, bcc, fcc

As an extension, we not only considered the binary classification into "single phase" and "multiphase" but directly predicted the structure type (hcp, fcc, bcc) for a given alloy composition. The classes are not balanced, with "multiphase" forming the majority class. As shown in Supplementary Figure 38, GPT-3 can learn to predict the phase based on only the composition and with very few data points.

We performed these experiments by fine-tuning for eight epochs with a learning rate decay rate of 0.02.

As a baseline, we used automated machine learning via automatminer (and the "express" setting).[23]

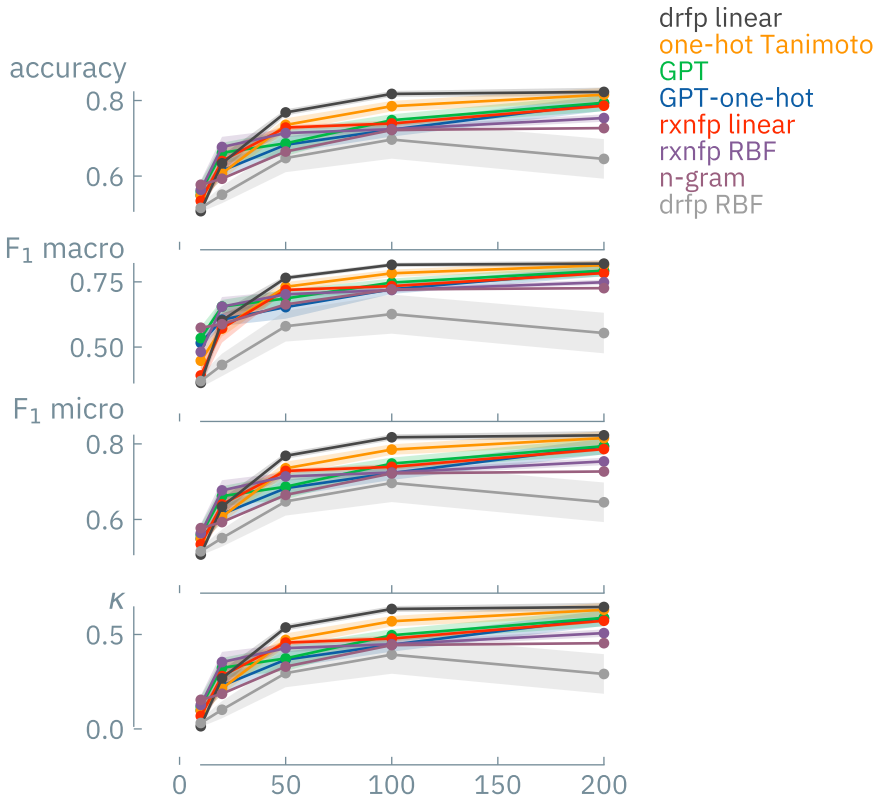


Supplementary Figure 38 | Classification performance for classifying HEAs into multiphase, hcp, fcc, bcc, respectively. Since Pei et al.[22] did not report this experiment, wherefore we use simply dummy models as baselines. Error bands show the standard deviations for 10 independent train/test splits.

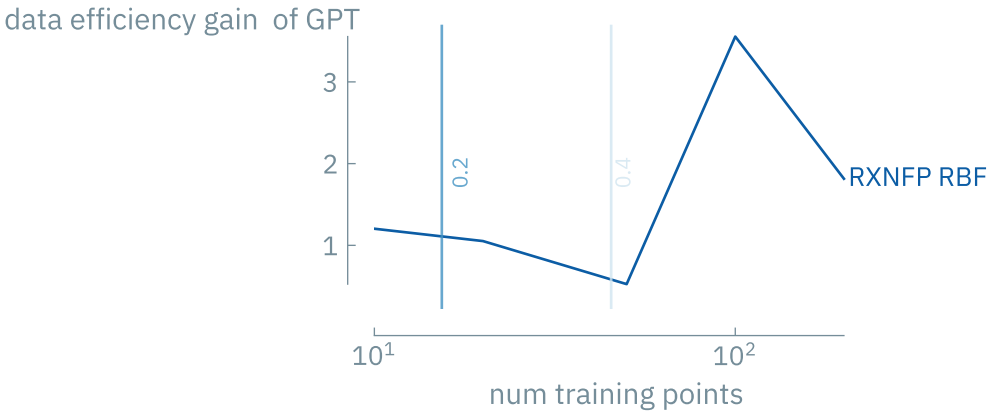
6.14 Reactions

For the reaction case studies, we use baselines based on GPR and natural language processing (NLP)-derived fingerprints as well as one-hot encoding and bag-of-SMILES representation. [50,51]

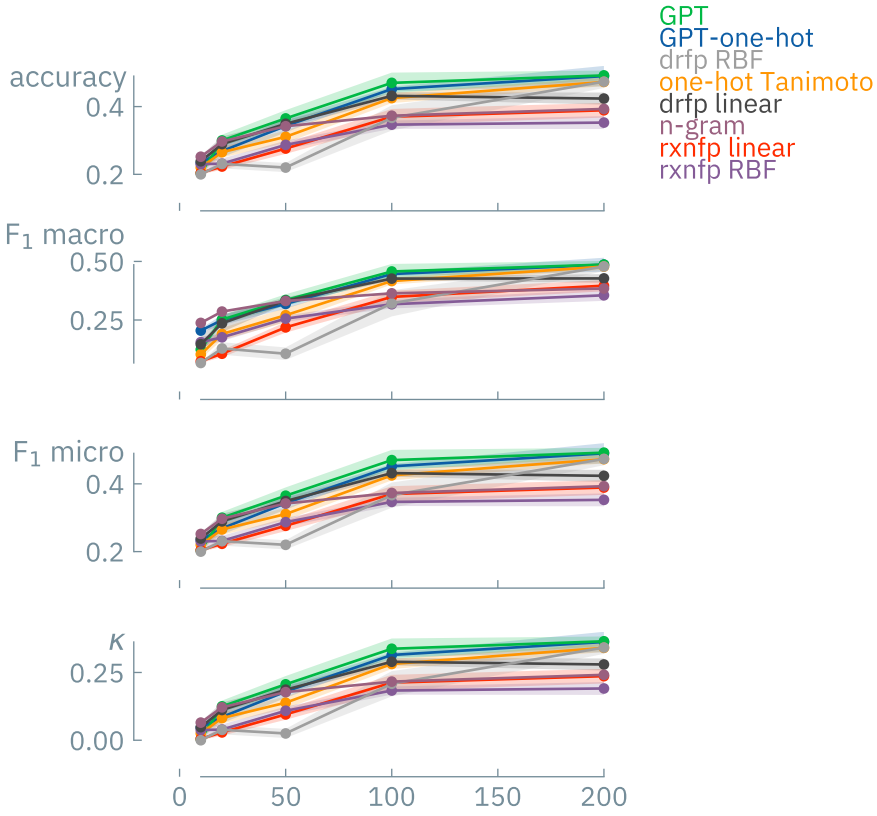
6.14.1 C-N cross-coupling



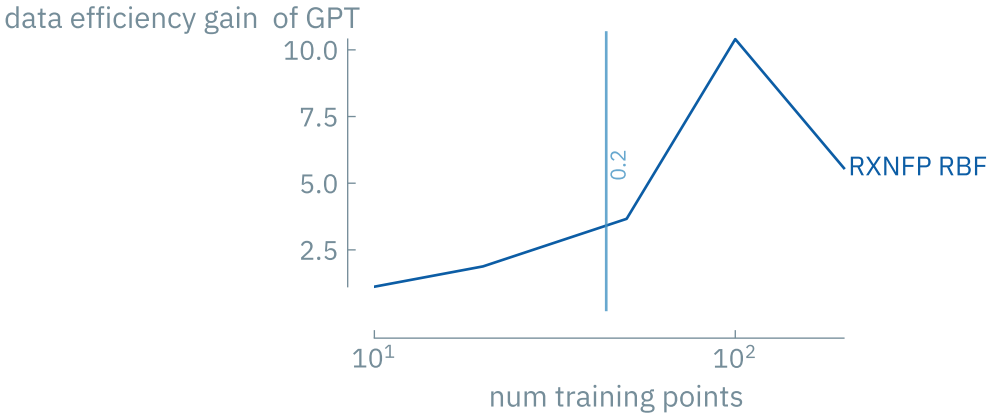
Supplementary Figure 39 | Learning curves for classification on binary classification of the reaction yield on the data set of Ahneman et al.[29].



Supplementary Figure 40 | Learning curve intersection points for binary classification on the Ahneman et al.[29] dataset. Vertical lines indicate the κ scores of the GPT-3 model.

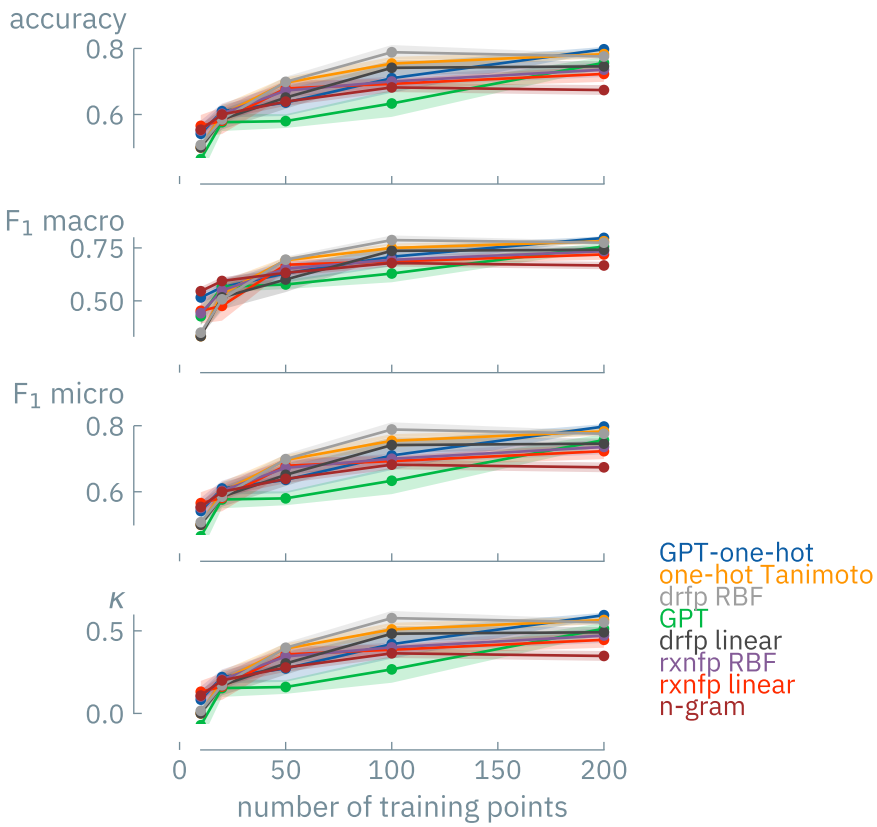


Supplementary Figure 41 | Learning curves for classification on the five-class classification of the reaction yield on the data set of Ahneman et al.[29]. Error bands show the standard error of the mean.

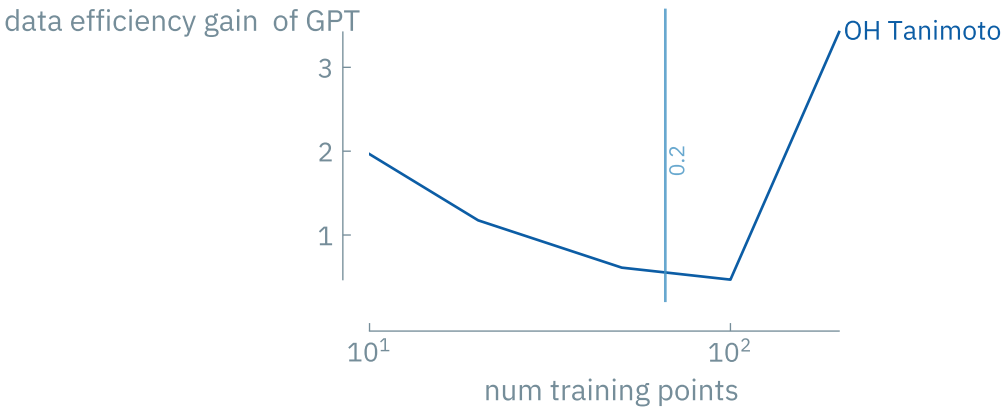


Supplementary Figure 42 | Learning curve intersection points for 5-class classification on the Ahneman et al.[29] dataset. Vertical lines indicate the κ scores of the GPT-3 model.

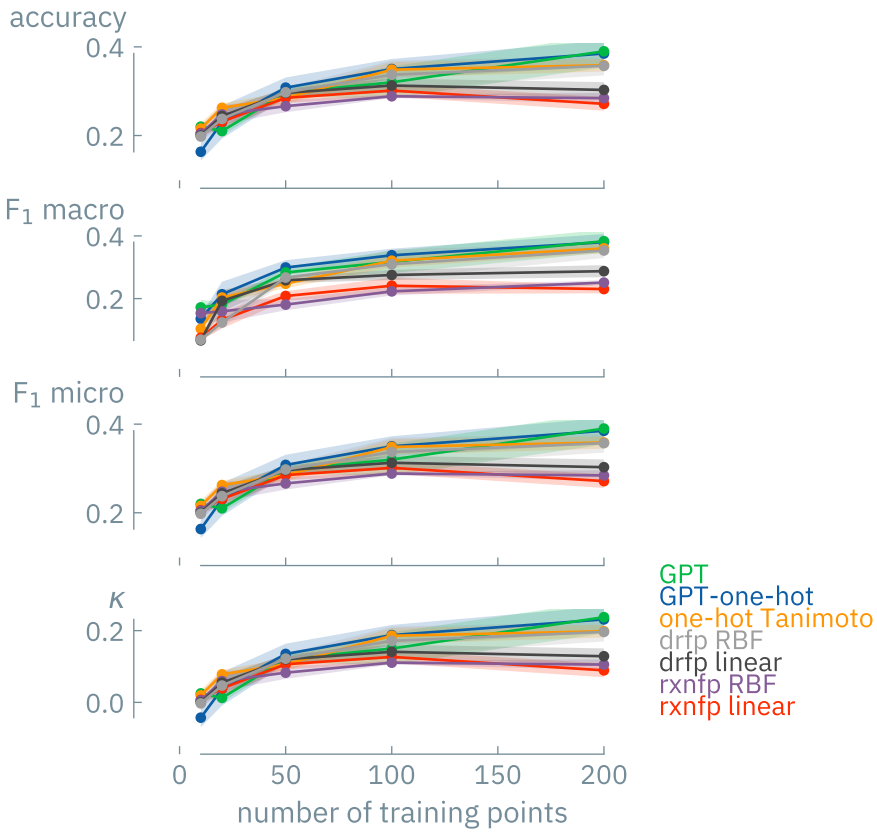
6.14.2 C-C cross-coupling



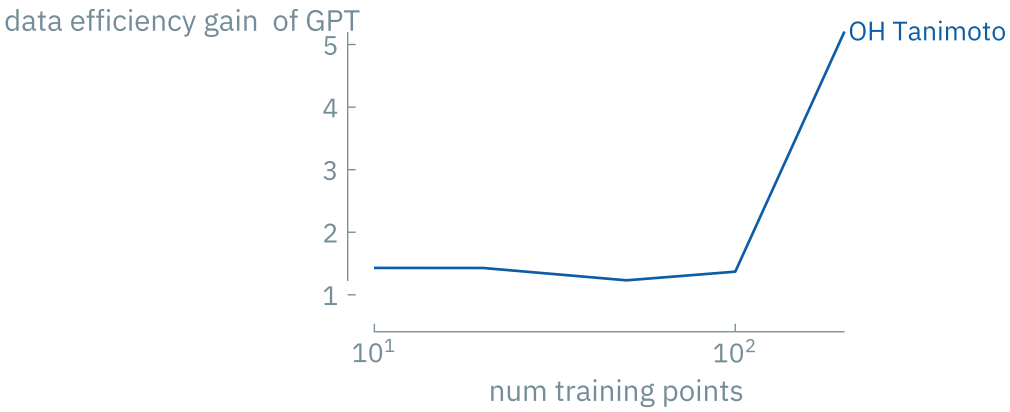
Supplementary Figure 43 | Learning curves for classification on binary classification of the reaction yield on the data set of Perera et al. [30]. Error bands show the standard error of the mean.



Supplementary Figure 44 | Learning curve intersection points for binary classification on the Perera et al.[30] dataset. Vertical lines indicate the κ scores of the GPT-3 model.



Supplementary Figure 45 | Learning curves for classification on the five-class classification of the reaction yield on the data set of Perera et al.[30]. Error bands show the standard error of the mean.



Supplementary Figure 46 | Learning curve intersection points for 5-class classification on the Perera et al.[30] dataset.

6.15 Matbench

6.15.1 Metallic glass formation ability

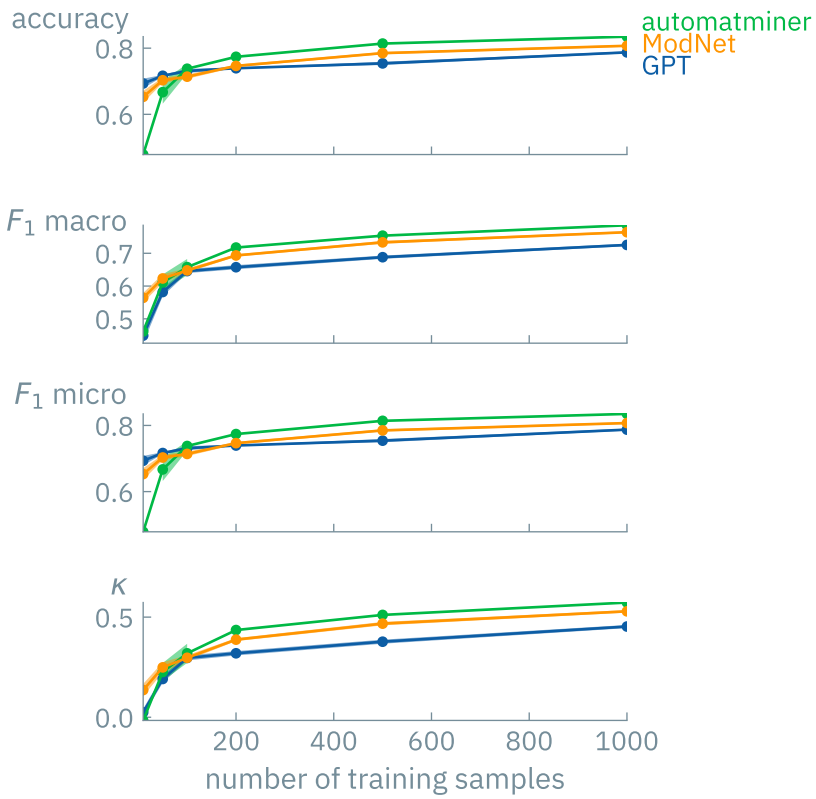
Supplementary Tab. 4: **Performance of fine-tuning of GPT-3 compared to the current matbench leaderboard for the `glass` task.**

model	accuracy	balanced accuracy	F_1	ROC-AUC
GPT-3	0.82 ± 0.01	0.77 ± 0.01	0.88 ± 0.01	0.77 ± 0.01
MODNet (v0.1.12)[52]	0.97 ± 0.01	0.96 ± 0.01	0.98 ± 0.00	0.96 ± 0.01
AMMExpress v2020[23]	0.87 ± 0.05	0.86 ± 0.02	0.90 ± 0.04	0.86 ± 0.02
RF-SCM/Magpie[23,53,54,55]	0.90 ± 0.01	0.86 ± 0.02	0.93 ± 0.01	0.86 ± 0.02
MODNet (v0.1.10)[52]	0.87 ± 0.01	0.81 ± 0.02	0.91 ± 0.01	0.81 ± 0.02
Dummy	0.59 ± 0.02	0.50 ± 0.02	0.71 ± 0.01	0.50 ± 0.02

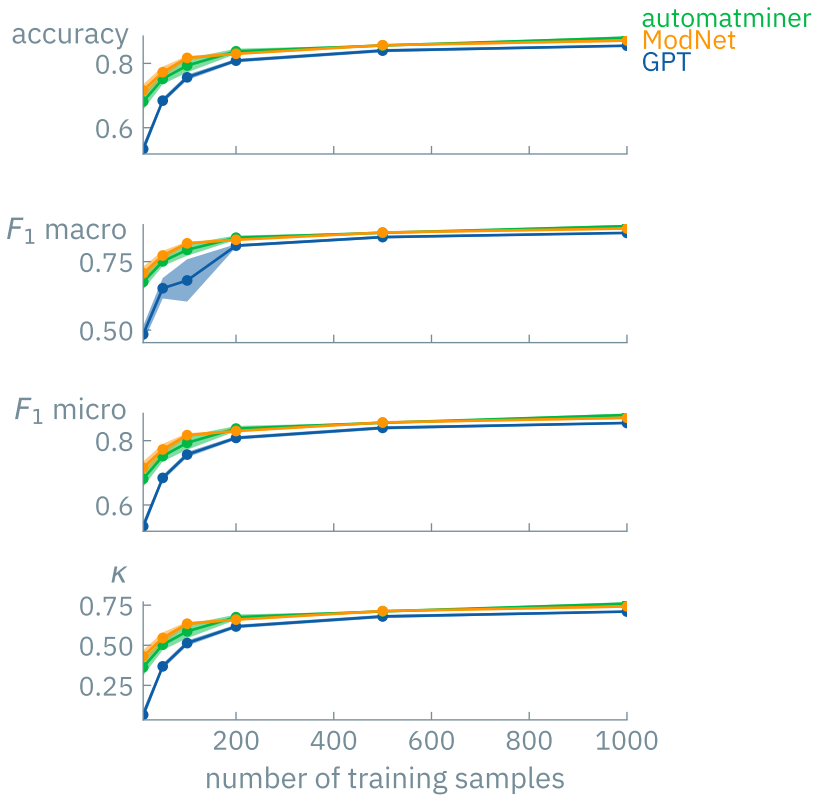
6.15.2 Metallic behavior

Supplementary Tab. 5: **Performance of fine-tuning of GPT-3 compared to the current matbench leaderboard for the `expt_is_metal` task.**

model	accuracy	balanced accuracy	F_1	ROC-AUC
GPT-3	0.89 ± 0.00	0.89 ± 0.00	0.89 ± 0.00	0.89 ± 0.00
AMMExpress v2020[23]	0.92 ± 0.00	0.92 ± 0.00	0.92 ± 0.00	0.92 ± 0.00
RF-SCM/Magpie[23,53,54,55]	0.92 ± 0.01	0.92 ± 0.01	0.92 ± 0.01	0.92 ± 0.01
MODNet (v0.1.10)[52]	0.92 ± 0.01	0.92 ± 0.01	0.92 ± 0.01	0.92 ± 0.01
Dummy	0.49 ± 0.01	0.49 ± 0.01	0.49 ± 0.02	0.49 ± 0.01



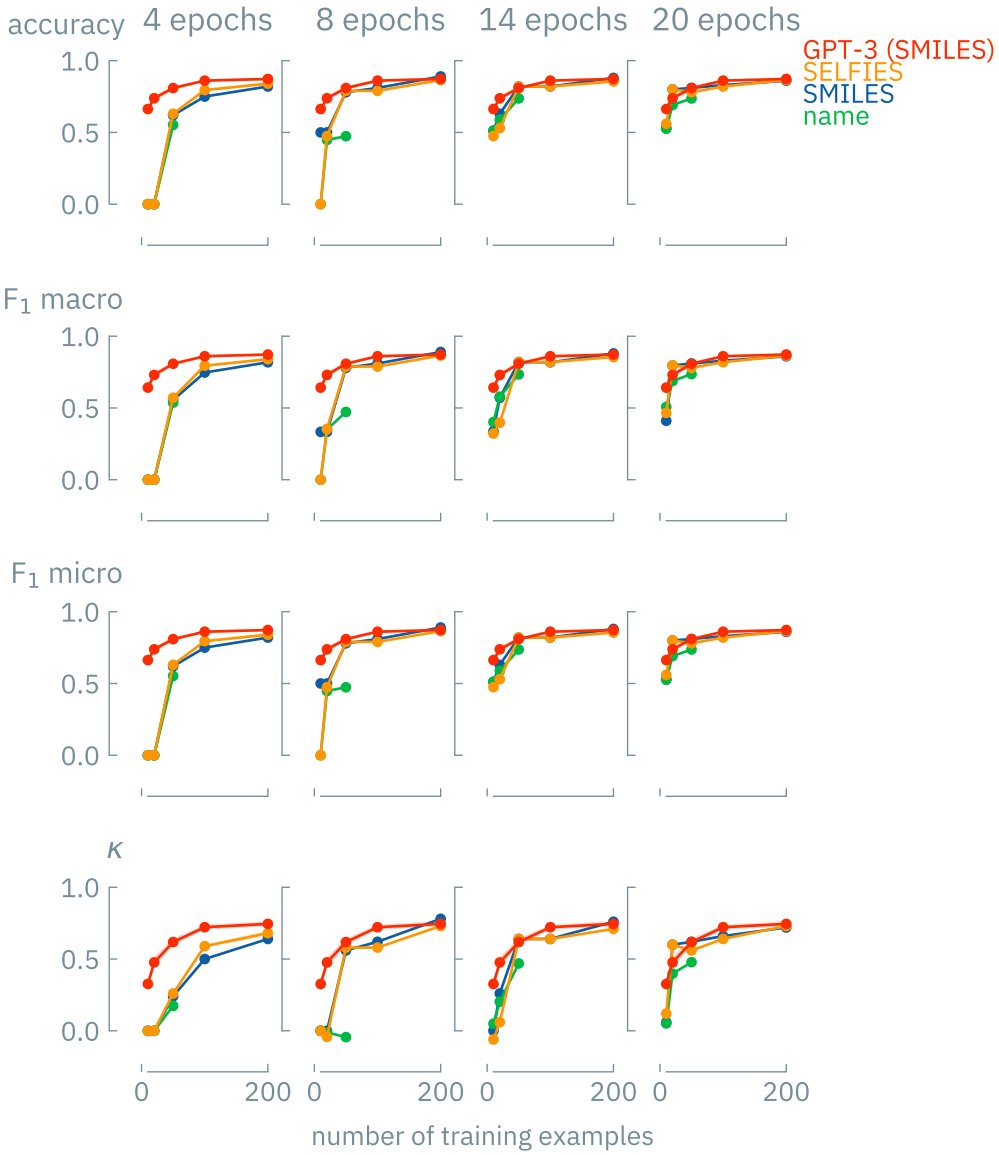
Supplementary Figure 47 | Learning curve analysis for the matbench `glass` task. As baselines, we considered the Automatminer with "express" settings and MODNet (with hyperparameters optimized by Breuck et al.[52] for this task).



Supplementary Figure 48 | Learning curve analysis for the matbench `is_metal` task. As baselines, we considered the Automatminer with "express" settings and MODNet (with hyperparameters optimized by Breuck et al.[52] for this task). Error bands show the standard error of the mean.

Supplementary Note 7 Experiments using GPT-J

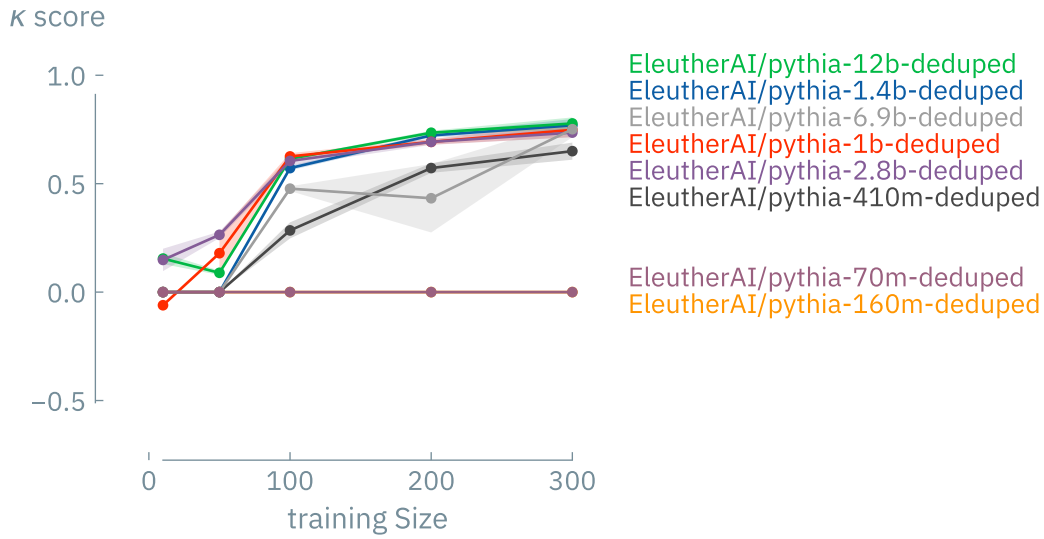
We also attempted to reproduce our results with the open-source GPT-J model. For an initial exploration of the influence of the tuning parameters, we focussed on the photoswitch dataset. Supplementary Figure 49 shows the learning curves as a function of the number of fine-tuning epochs for a fixed learning rate of 1.00×10^{-4}. Compared with the results we find for fine-tuning with OpenAI's GPT-3 we need more fine-tuning epochs than we used for GPT-3. However, we find competitive performance beyond the second point on the learning curve.



Supplementary Figure 49 | Performance of parameter-efficient fine-tuning of GPT-J for binary classification on the photoswitch case study.

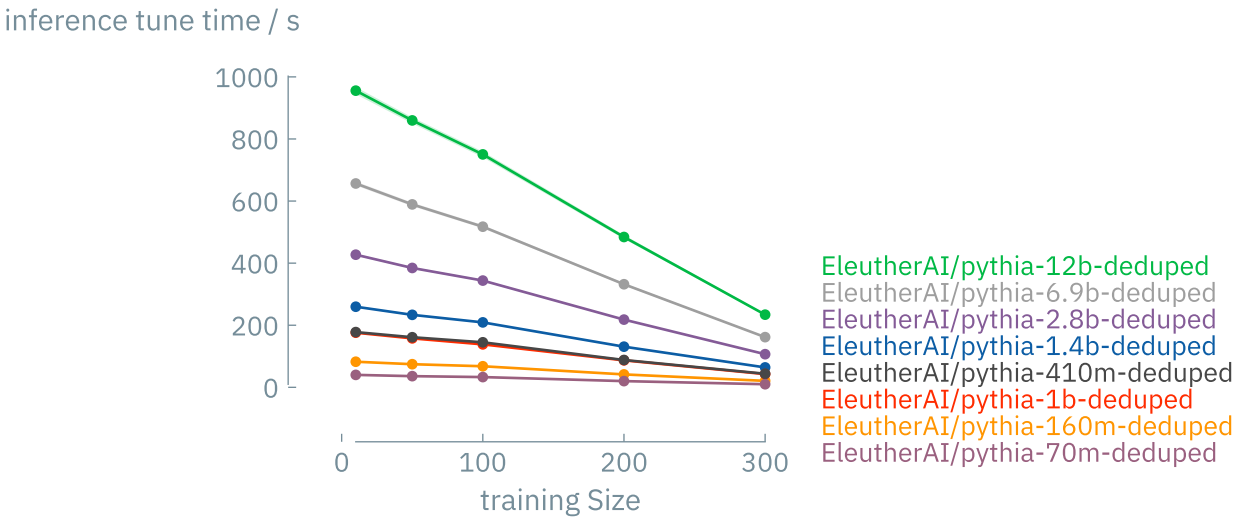
Supplementary Note 8 Scaling experiments

To understand how our approach performs with (open-source) LLMs of different parameter counts, we performed scaling experiments with the Pythia suite of models. [56]

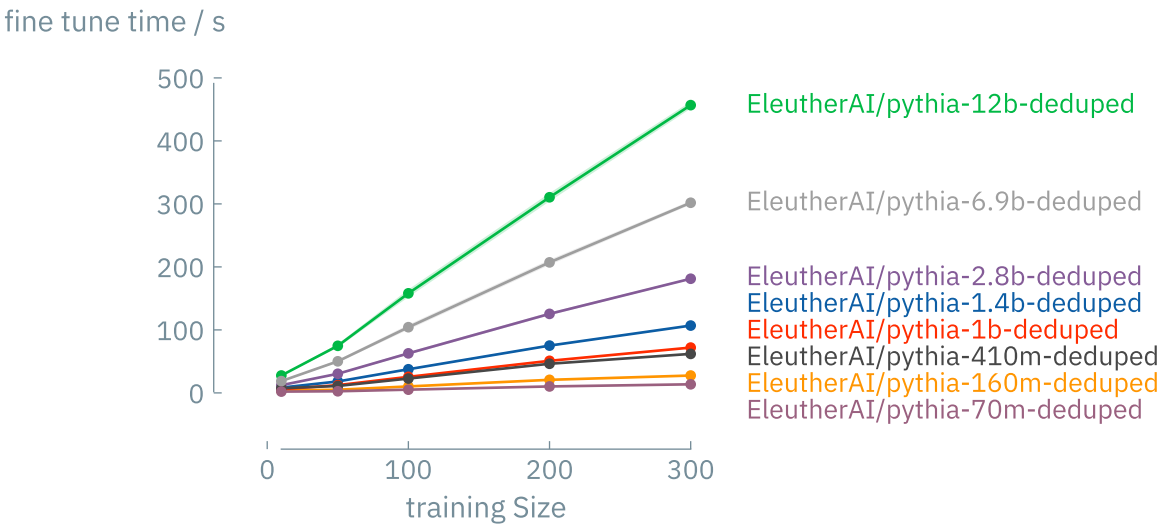


Supplementary Figure 50 | Performance of parameter-efficient language-interfaced of large-language models as a function of model size. We performed LIFT with PEFT techniques on a balanced classification task on the photoswitch dataset. From the plot, it is clear that increasing the model size leads to performance increases. However, already a 1B parameter model performs well in this experiment. Error bands show the standard error of the mean.

With the current default settings for fine-tuning and inference (on 380 points) batch sizes (32), we find on an A100 card the following (Supplementary Figure 51 and Supplementary Figure 52) training and inference times as a function of the model size.



Supplementary Figure 52 | Inference time (on the test set of 380 points) as a function of model size (based on the experiments for the scaling curves in Supplementary Figure 50)



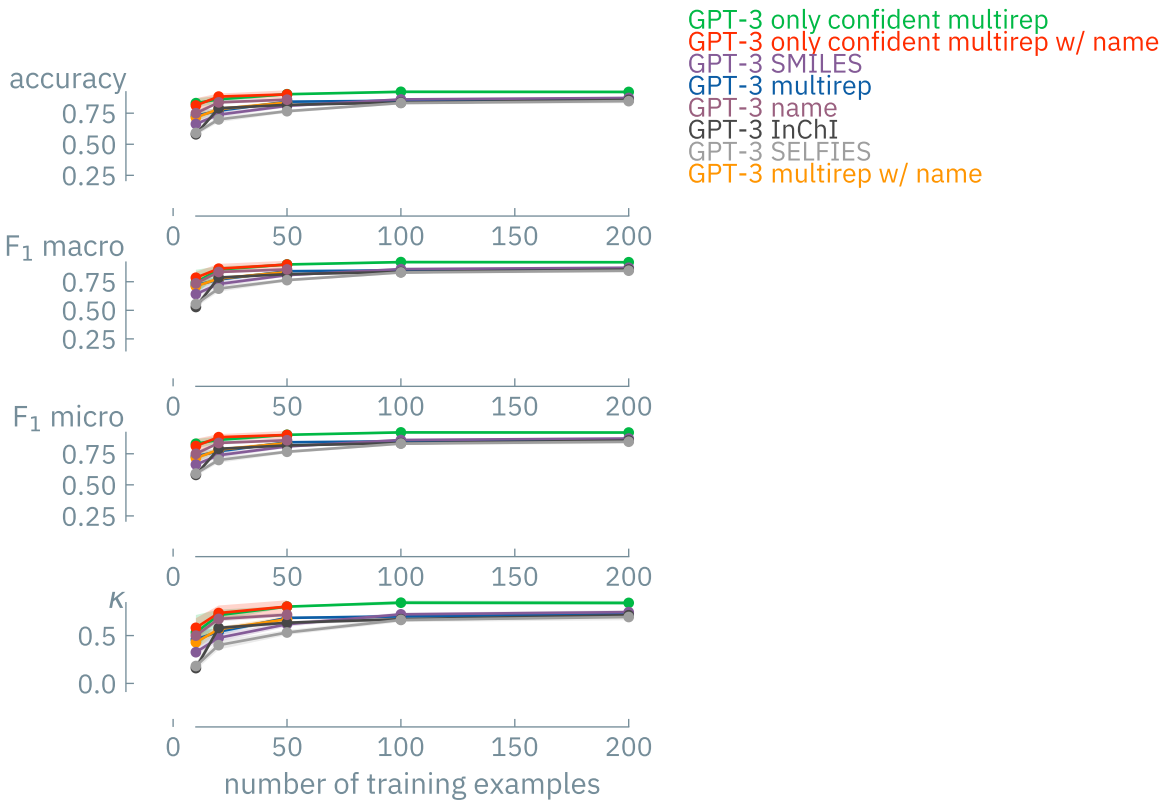
Supplementary Figure 51 | Fine-tune time for models of different size as a function of the training set size (based on the experiments for the scaling curves in Supplementary Figure 50).

Supplementary Note 9 Ensemble over representation

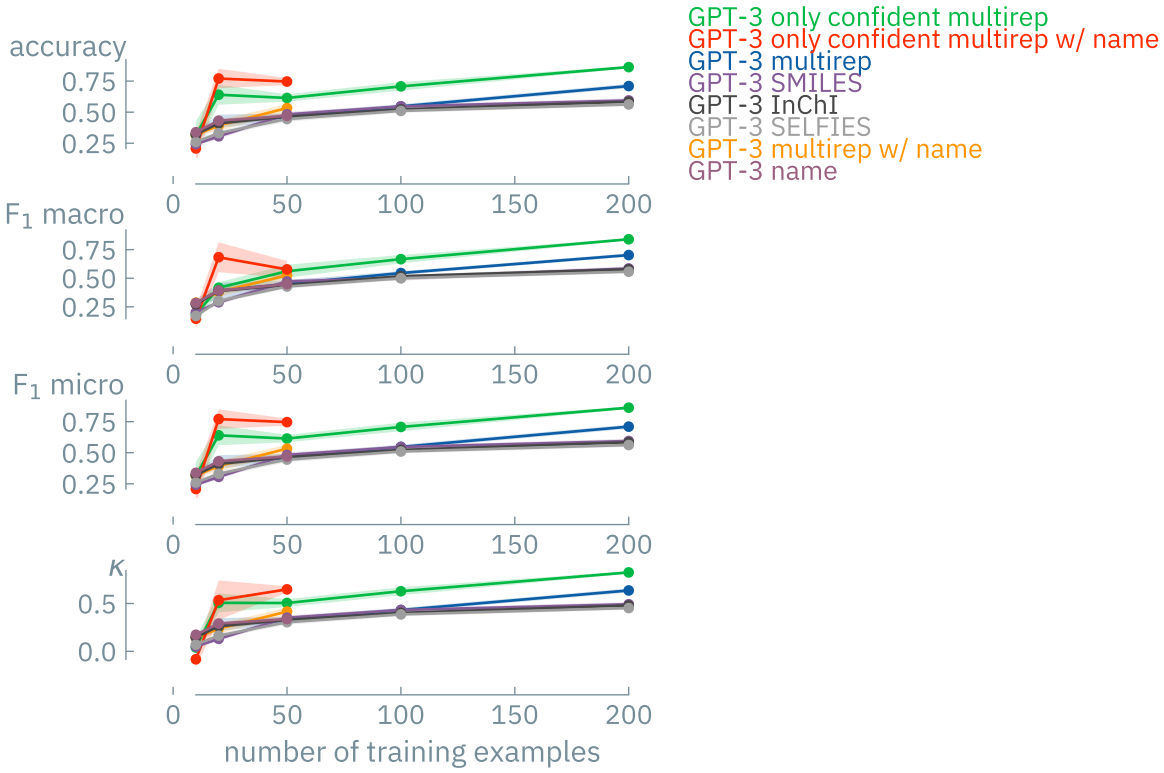
We additionally started to investigate how we can leverage the fact that the model works well across different representations. One simple approach, which we call MultiRep, is to use it as a data augmentation and rough uncertainty quantification scheme by simply showing the same data point in multiple representations to the same model.

That is, for one molecule, we will fine-tune the model on prompts constructed with k representations such as SMILES, self-referencing embedded strings (SELFIES), international chemical identifier (InChI), and IUPAC name, which will effectively augment the training dataset by a factor of k. Upon inference, we can query k time—one time per representation—and, in this way, obtain some rough measure of uncertainty.

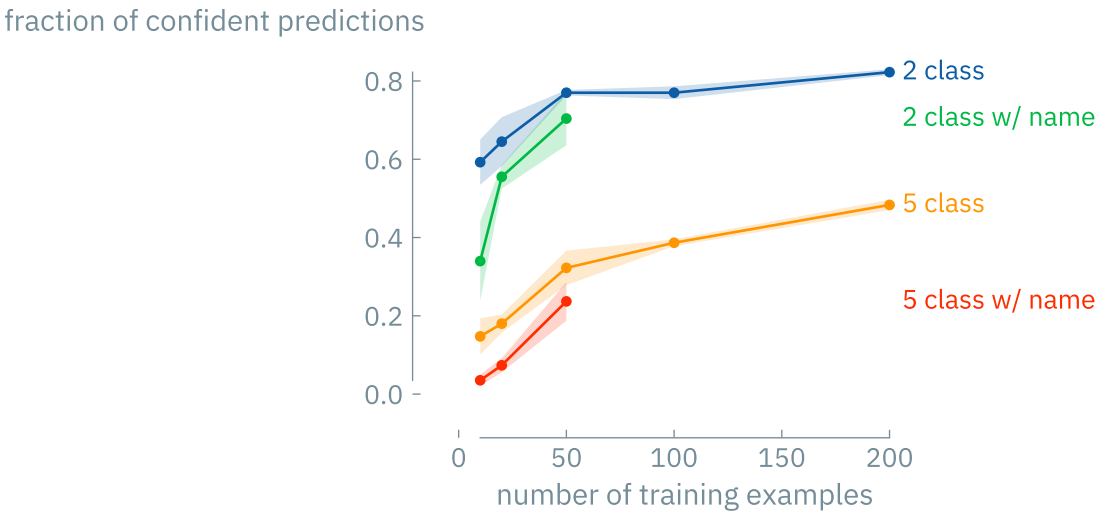
The performance of this approach is shown in Supplementary Figure 53 and Supplementary Figure 54, which highlights the large boost in predictive power this approach might bring.



Supplementary Figure 53 | Performance of the MultiRep approach for binary classification on the photoswitch case study. Error bands show the standard error of the mean.



Supplementary Figure 54 | Performance of the MultiRep approach for binary classification on the photoswitch case study. Error bands show the standard error of the mean.



Supplementary Figure 55 | Fraction of confident predictions for binary and 5-class classification on the photoswitch case study. Error bands show the standard error of the mean.

Supplementary Note 10 Regression

10.1 Note on regression using GPT-3

Without changes to the architecture and the training procedure, regression, i.e., the prediction of floating point numbers with, in principle, an infinite number of decimal places, is not possible. However, we can approximate regression in two ways. First, one could use many classes, such that the class bins are in the order of the experimental error (often larger than one decimal place). Second, one can try to directly predict rounded floating point numbers.

In both cases, one would expect the performance to be worse than in the classification setting. Here, we focused on the simplest case of directly predicting rounded floating point numbers. Typically, GPT-3 performs worse than baselines in this setting. However, it sometimes approaches the performance of the baselines.

Additionally, we show for the photoswitch case study the performance for approximating regression as classification with many bins.

Errors due to rounding Since we round the values (as we cannot predict numbers with an infinite number of decimal points), our model will not be able to perform better than this rounding error. This effect is shown in Figure 1.

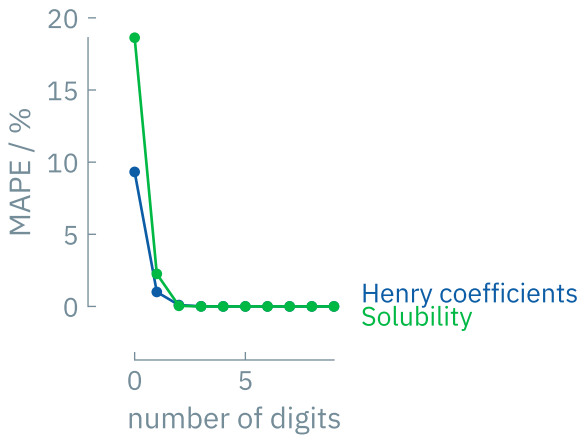
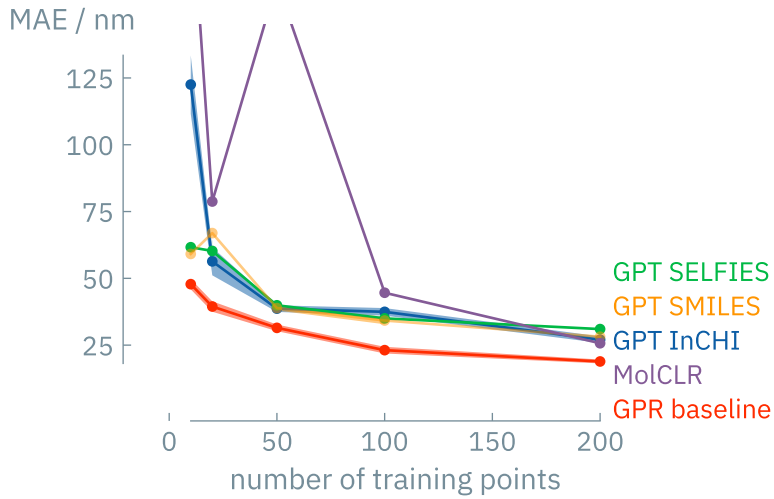


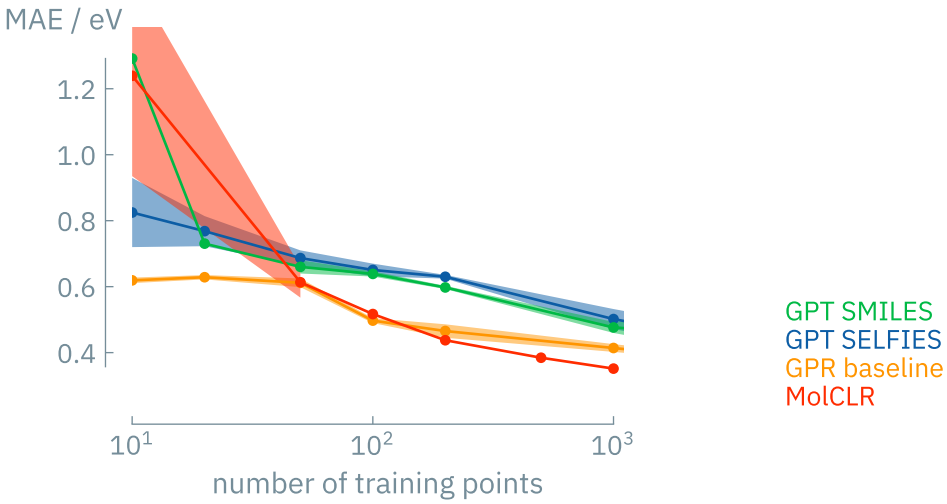
Figure 1: Mean absolute percentage error due to rounding to different numbers of digits on two datasets.

10.2 Photoswitches



Supplementary Figure 56 | Learning curve for regression on the photoswitch dataset. Error bands show the standard error of the mean.

10.3 HOMO-LUMO gaps



Supplementary Figure 57 | Learning curve for regression on the QMUG dataset. Error bands show the standard error of the mean.

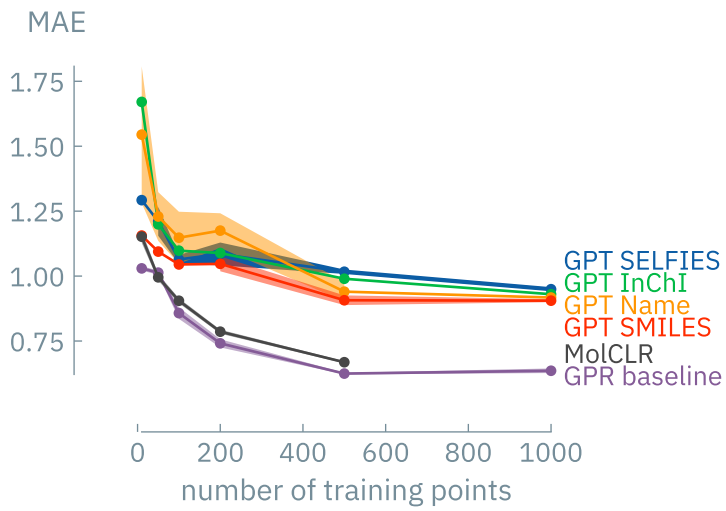
10.4 Solubility

Supplementary Figure 58 | Learning curve for regression on the solubility dataset. Error bands show the standard error of the mean.

10.5 Free energy of solvation

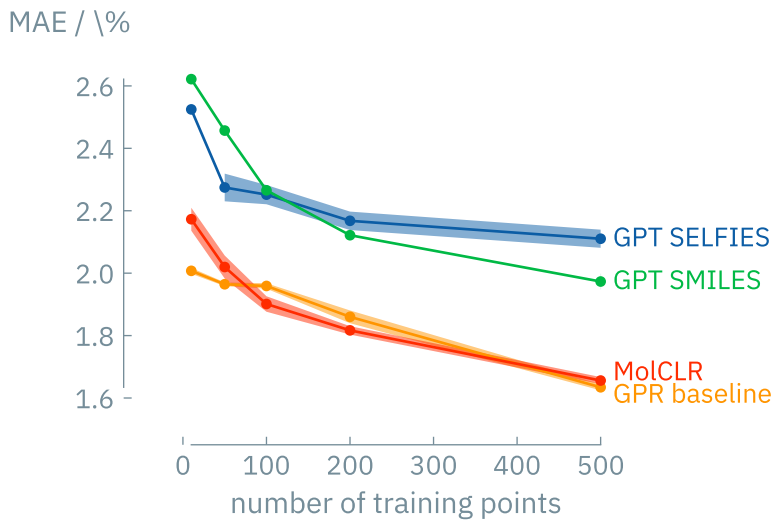
Supplementary Figure 59 | Learning curve for regression on the free energy of solvation dataset. Error bands show the standard error of the mean.

10.6 Lipophilicity



Supplementary Figure 60 | Learning curve for regression on the lipophilicity dataset. Error bands show the standard error of the mean.

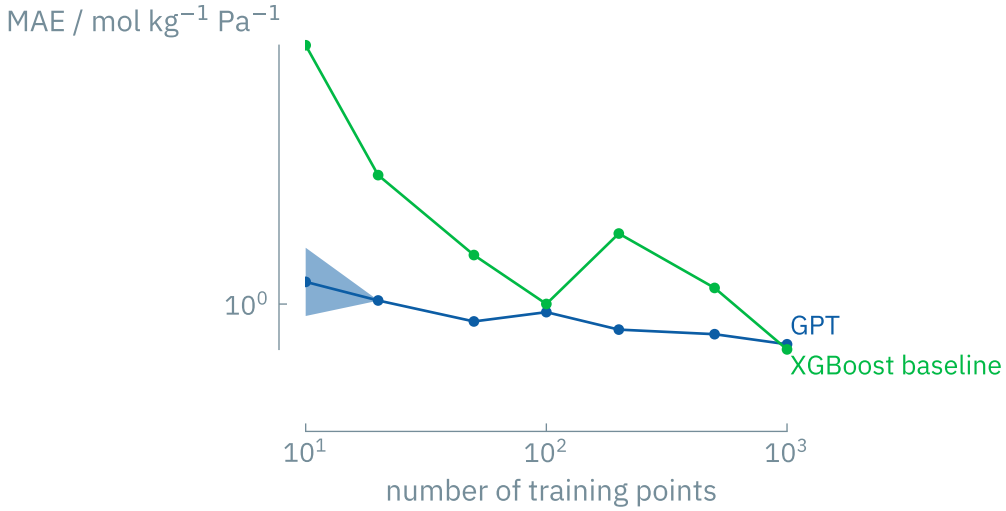
10.7 Photoconversion efficiency



Supplementary Figure 61 | Learning curve for regression on the OPV dataset. Error bands show the standard error of the mean.

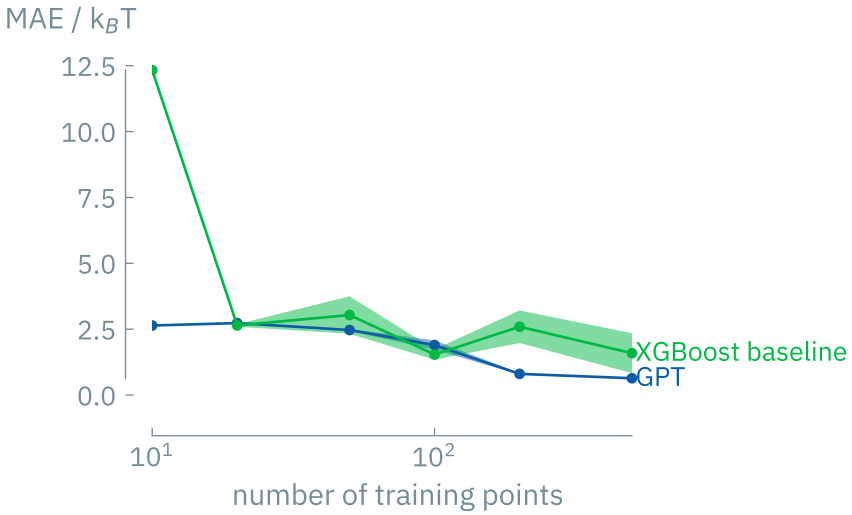
10.8 Henry coefficients

10.8.1 Carbon dioxide



Supplementary Figure 62 | Learning curves for regression for the prediction of the CO_2 Henry coefficient of MOFs. Error bands show the standard error of the mean.

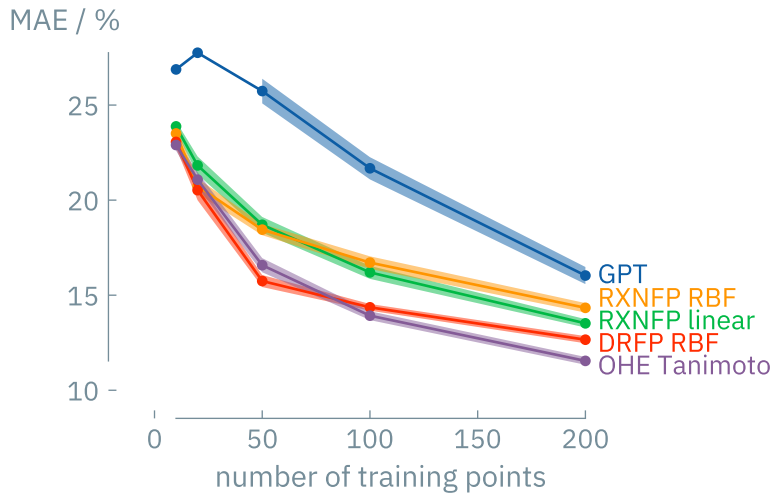
10.9 Surfactants



Supplementary Figure 63 | Learning curve for regression for predicting the adsorption free energy on the dispersant dataset. Error bands show the standard error of the mean.

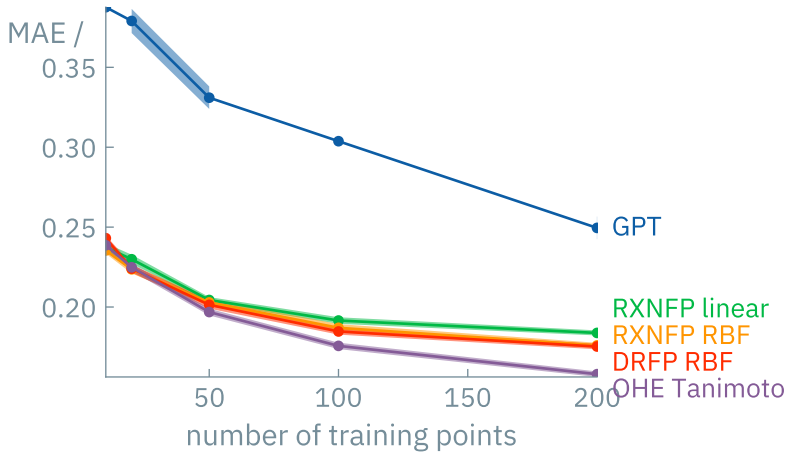
10.10 Reactions

10.10.1 C-N cross-coupling



Supplementary Figure 64 | Learning curve for regression yield prediction on the C-N cross-coupling dataset. Error bands show the standard error of the mean.

10.10.2 C-C cross-coupling



Supplementary Figure 65 | Learning curve for regression yield prediction on the C-C cross-coupling dataset. Error bands show the standard error of the mean.

10.11 Matbench

Also, in the regression setting, fine-tuning of GPT-3 on the composition performs better than the dummy model and is competitive with models on the matbench leaderboard. Supplementary Table 6 summarizes the metrics for the prediction of band gaps and Supplementary Table 7 for the prediction of the yield strength of steels.

10.11.1 Experimental band gaps

Supplementary Tab. 6: **Performance of a fine-tuned GPT compared to the matbench leaderboard for the `expt_gap` task.**

model	MAE	RMSE	MAPE	max error
GPT-3	0.46 ± 0.02	1.06 ± 0.09	0.52 ± 0.07	9.36 ± 1.96
Ax SAASBO	0.33 ± 0.01	0.81 ± 0.06	0.36 ± 0.05	7.19 ± 1.97
CrabNet v1.2.7 [48,57]				
MODNet (v0.1.12) [52]	0.33 ± 0.02	0.77 ± 0.07	0.35 ± 0.04	7.11 ± 1.47
Dummy	1.14 ± 0.03	1.44 ± 0.07	0.95 ± 0.17	8.93 ± 1.23

10.11.2 Yield strength

Supplementary Tab. 7: **Performance of a fine-tuned GPT compared to the matbench leaderboard for the `steels` task.**

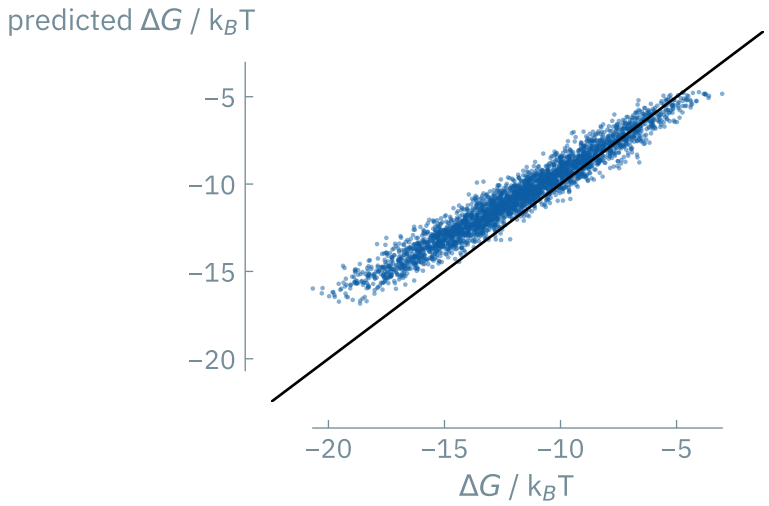
model	MAE	RMSE	MAPE	max error
GPT-3	142 ± 16	204 ± 17	0.10 ± 0.01	679 ± 64
MODNet (v0.1.12) [52]	88 ± 12	145 ± 37	0.06 ± 0.01	722 ± 277
Crabnet [48]	107 ± 19	153 ± 29	0.07 ± 0.01	477 ± 79
RF-Regex Steels [23]	91 ± 7	128 ± 10	0.06 ± 0.01	423 ± 72
Dummy	230 ± 10	301 ± 21	0.16 ± 0.00	1032 ± 59

Supplementary Note 11 Inverse design

Note that for the inverse design case studies, we focused on examples for which we can readily test the performance of the predicted materials.

11.1 Monomer sequences

Since our objectives come from computationally expensive mesoscale simulations, it was not computationally feasible for us to run the simulations for all the generated polymers. Instead, we trained our XGBoost baseline model on all the polymers in our dataset and used this model to score the generated monomer sequences. The performance of this model is shown in Supplementary Figure 66



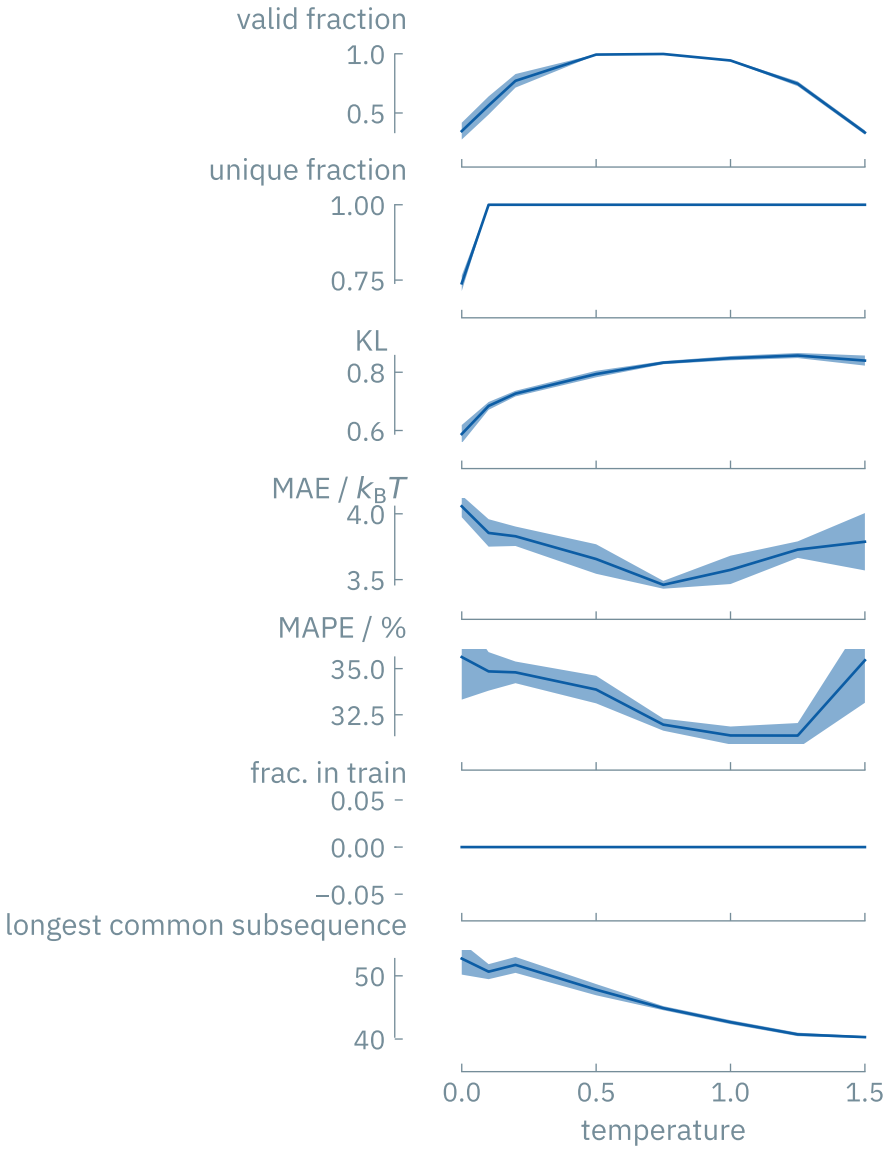
Supplementary Figure 66 | XGBoost model trained on the polymer database. Mean absolute error 1.24 $k_{\mathrm{B}}T$.

11.1.1 Random generation continuous target

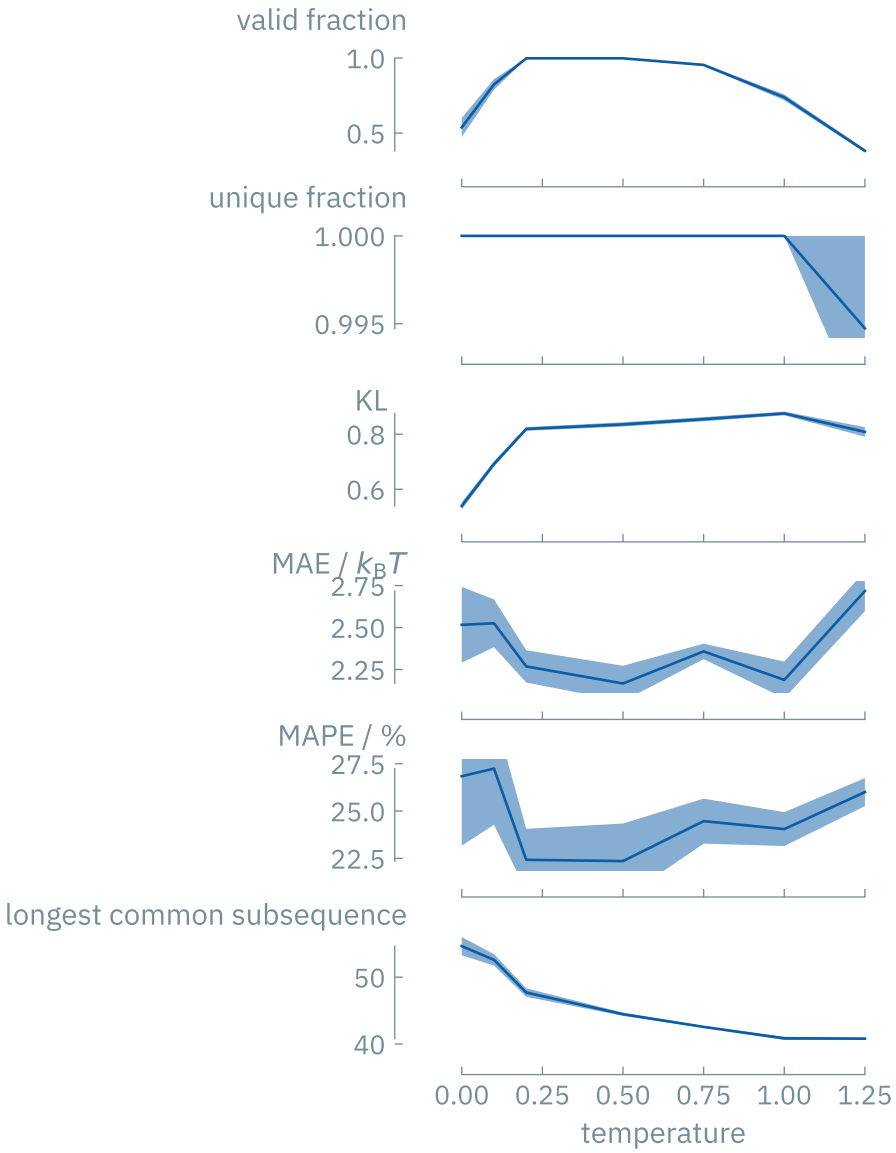
In Supplementary Figure 67 we show the performance for generating polymers with properties randomly sampled from the training distribution. We use rounded continuous numbers in the prompts.

11.1.2 Random generation binned target

Instead of using continuous numbers in the prompt, we also considered using the ordinal encoding into five bins. As a loss, we then computed the distance to the nearest bin edge.



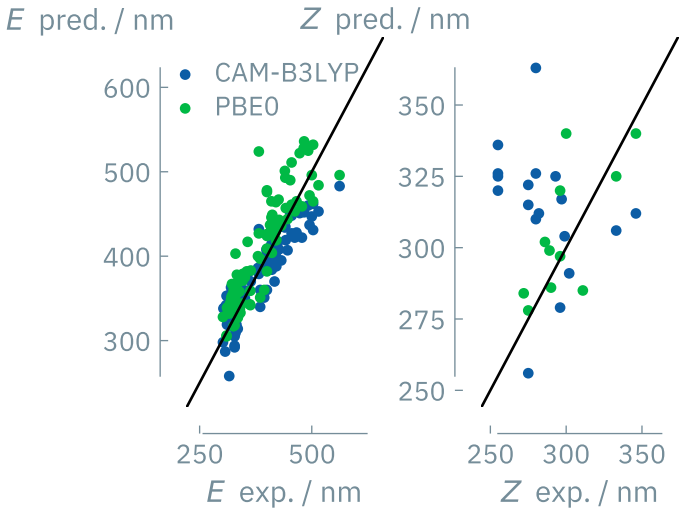
Supplementary Figure 67 | Inverse design metrics for the random generation of polymers as a function of temperature.



Supplementary Figure 68 | Inverse design metrics for the random generation of polymers as a function of temperature for ordinal design objectives (adsorption free energies). Error bands show the standard error of the mean.

11.2 Photoswitches

To investigate if we can evaluate our generative model using TDDFT, we conducted some experiments to analyze how well we can match the experimental or previously reported DFT data.

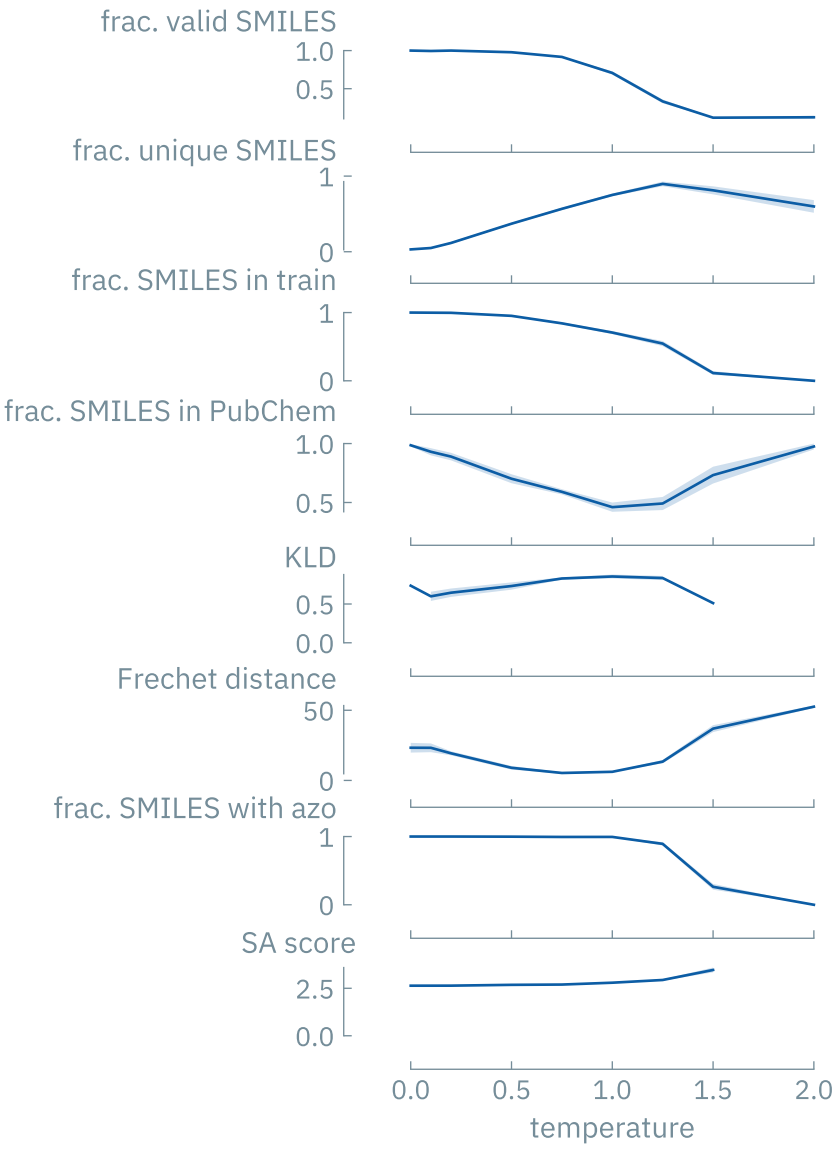


Supplementary Figure 69 | Correlation between experimental transition wavelengths and reported TD-DFT data.

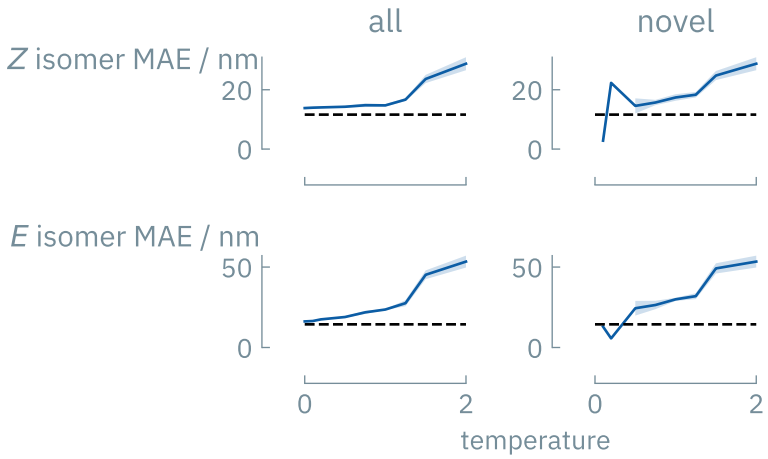
A dummy-mean baseline on the photoswitch dataset, gives a mean absolute error (MAE) of 53.82 nm for the *E* isomer $\pi - \pi^*$ transition wavelength, and one of 12.63 nm for the *Z* isomer $\pi - \pi^*$ transition wavelength.

11.2.1 Sampling from the training distribution

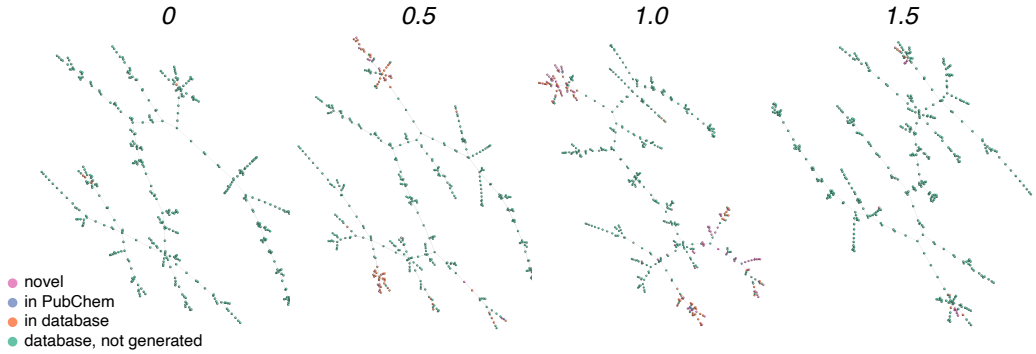
As the initial experiment, we prompted the model with transition wavelengths randomly sampled from the distribution of the dataset by Griffiths et al.[2] Supplementary Figure 70 shows the quality and the diversity of the generated molecules, Supplementary Figure 71 shows the constrain satisfaction.



Supplementary Figure 70 | SMILES generation metrics for random generation of photoswitches. Error bands show the standard error of the mean.



Supplementary Figure 71 | Constrain satisfaction for random generation of photoswitches. Error bands show the standard error of the mean.

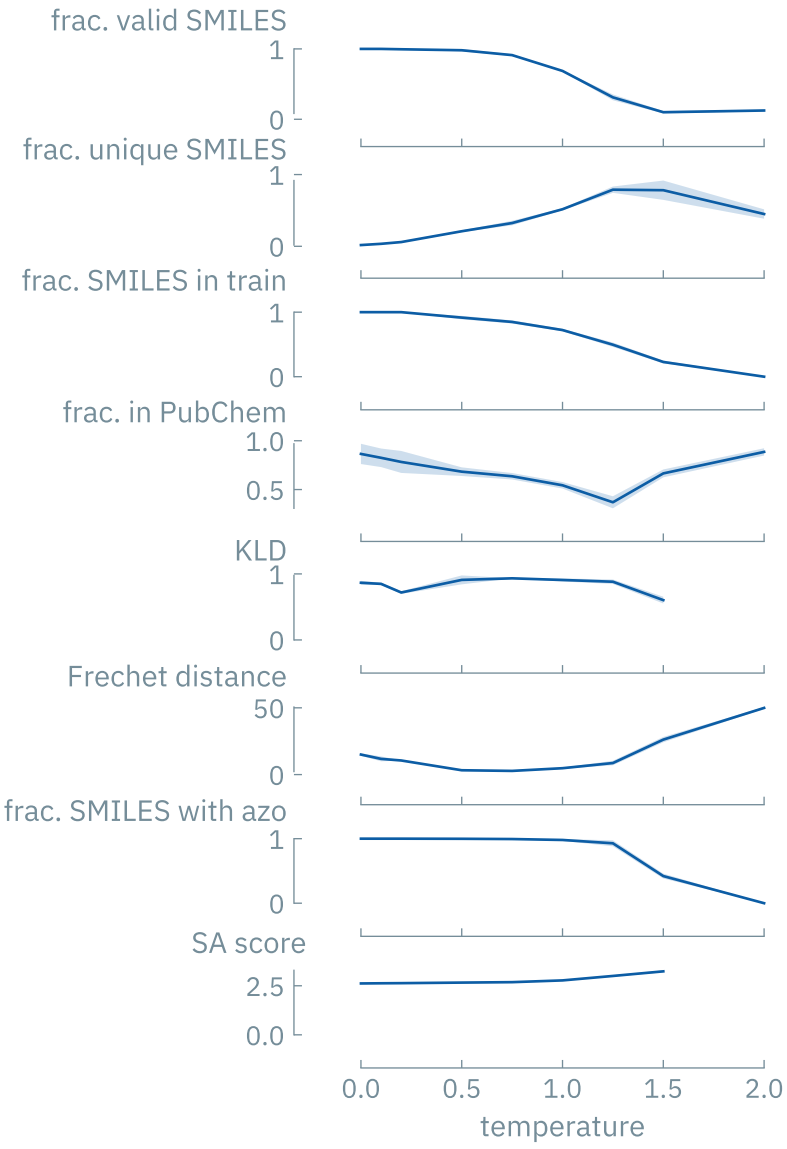


Supplementary Figure 72 | TMAP visualization of the generated photoswitches and the training set. For this visualization, we used the tree-map (TMAP) [58] algorithm on photoswitch molecules described using MinHash fingerprint (MHFP) with 2048 permutations. [59]

11.2.2 Extrapolation

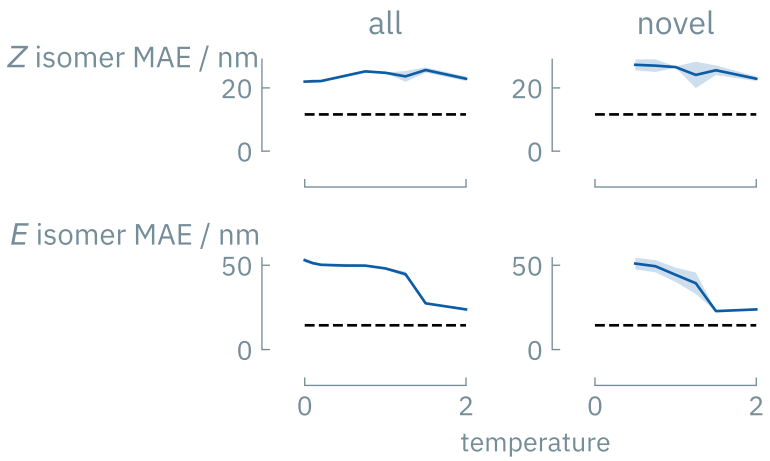
To investigate if our approach can outperform high-throughput virtual screening (HTVS), we only trained our model on photoswitches that adsorb at wavelengths below 350.00 nm. Upon inference, we queried for photoswitches with transition wavelengths above this threshold.

In Supplementary Figure 73 we show the quality of the generated SMILES, and in Supplementary Figure 74 the constrain satisfaction. Supplementary Figure 75 compares the generated distributions.



Supplementary Figure 73 | SMILES generation metrics for extrapolative generation of photoswitches. We find that low temperatures generated more valid molecules that, however, are less unique and often part of PubChem. Error bands show the standard error of the mean.

Supplementary Figure 76 gives TMAP visualizations as a function of temperature.

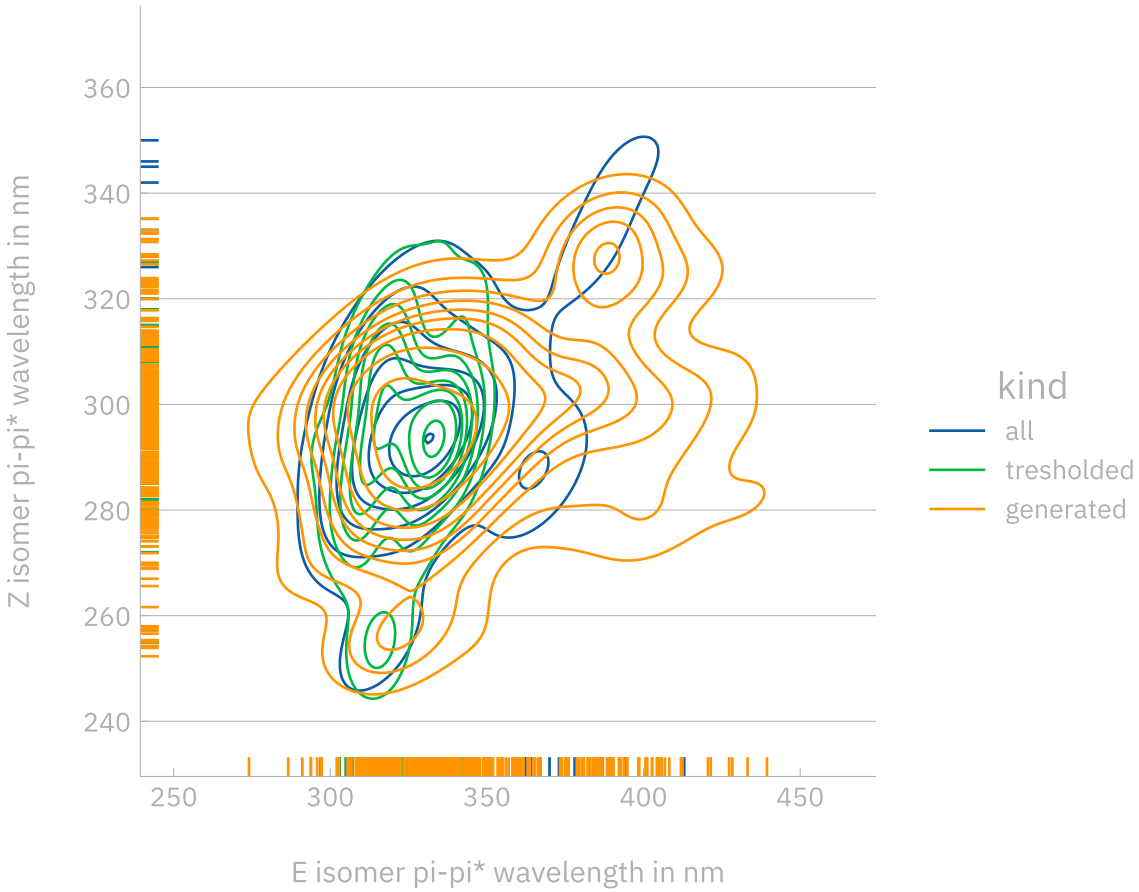


Supplementary Figure 74 | Constrain satisfaction for extrapolative generation of photoswitches. We generally find larger errors than for the random generation. "all" refers to the metrics for all generated photoswitches, including those that have been part of the training set. "novel" only includes those not part of the training set. Error bands show the standard error of the mean.

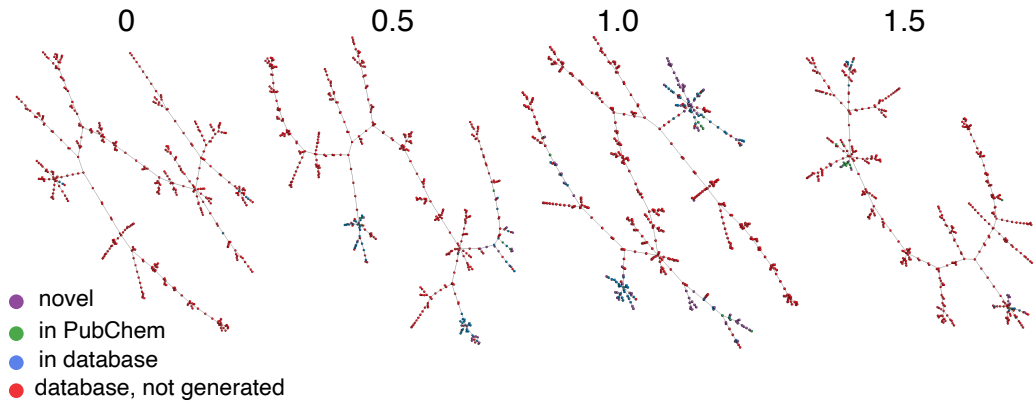
TDDFT validation To further analyze our shortlisted candidates, we performed TDDFT simulations. Briefly, we used Gaussian 16, Revision C.01[60] to perform geometry optimization followed by the computation of the singlet excited states. Following Jacquemin et al.[61] we used the PBE0 functional (PBE1PBE)[62] and 6-31G(d',p') basis set,[63,64] modeling the effect of ethanol solvent using Polarizable Continuum Model (PCM).[65] Supplementary Table 8 list the results.

Supplementary Tab. 8: **TDDFT Predicted transition wavelengths.** SMILES generated in the extrapolation experiment (training set containing only molecules with transition wavelengths small than 350.00 nm.

SMILES	E π-π^* / nm	Z π-π^* / nm
CC1=NOC(C)=C1/N=N/C2=CC=C(NC)C=C2	386.78	386.79
CC1=C(C(C)=NN1)/N=N/C2=CC=C(NC)C=C2	373.75	387.06
C[N]1C=CC=C1N=NC2=CC=CC=C2	371.45	355.75
CC1=C(C(C)=NN1)/N=N/C2=CC(NC)=CC=C2	325.23	317.49
FC1=CC=CC=C1/N=N/C2=CC=C(N)C=C2	379.68	333.06
FC1=CC=CC=C1/N=N/C2=C(N)C=CC=C2	422.21	386.60
C[N]1C=CC=C1N=NC2=CC(NC)=NN=C2	393.60	356.72
C[N]1N=CC(=C1N)N=NC2=CC=CC=C2	353.73	360.62
CC1=NOC(C)=C1/N=N/C2=C(N)C=CC=C2	401.91	376.42
CC1=C(C(C)=NN1)/N=N/C2=CC(N)C=C2	341.33	342.74
C[N]1C=CC=C1N=NC2=CC=CC=C2	355.75	293.61



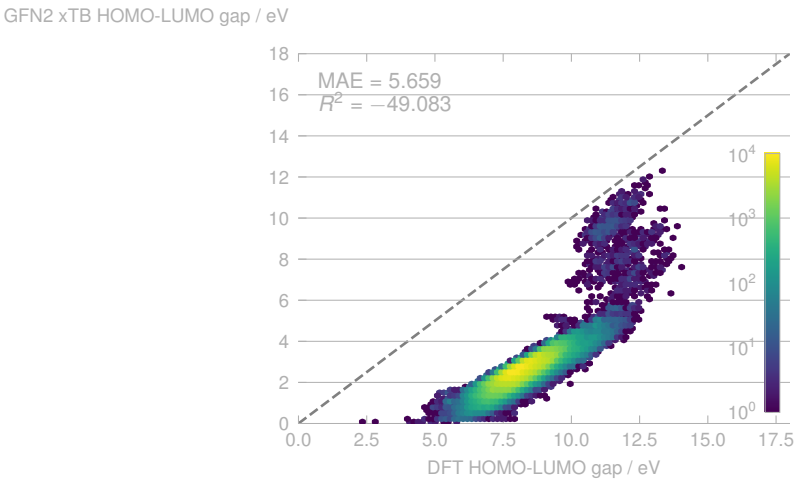
Supplementary Figure 75 | Distribution of transition wavelengths in generated photoswitch molecules, the training set, and the underlying database. For this figure, we considered 192 unique molecules generated across all tested temperatures (0–2) and noise levels. We find that even if we train the model on only a subset of the dataset (green), it can generate molecules (orange) that span a much wider range in transition wavelengths. Transition wavelengths for the generated molecules have been predicted using the GPR models reported by Griffiths et al.[2]



Supplementary Figure 76 | TMAP visualization of the generated photoswitches and the training set. For this visualization, we used the TMAP[58] algorithm on photoswitch molecules described using MHFP with 2048 permutations.[59] Some molecules generated by GPT-3 do not contain azo groups. Therefore we show in the top row only those that contain azo groups and below all generated valid molecules. We perform the analysis at different inference temperatures (columns).

11.3 HOMO-LUMO gaps

Note that we use HOMO-LUMO gaps computed using GFN2-extended tight-binding (xTB) for computational efficiency. As also reported by Isert et al.[3], there is a deviation between those gaps and those computed using DFT (Supplementary Figure 77).



Supplementary Figure 77 | DFT vs. GFN2-xTB bandgaps on the QMUGs dataset.

To measure the compliance of the generated molecules with the prompts, we generated conformers using RDKit (via our givemeconformer wrapper[66] that also performs optimization using the Merck molecular force field,[67] ranking and pruning), performed a geometry optimization using GFN2-xTB, and evaluated the HOMO-LUMO gap. Since we did not run metadynamics to generate diverse conformers, our results differ for some cases from the ones reported in QMugs.

On the QMUG dataset, a dummy mean predictor would find a MAE of 0.51 eV.

11.3.1 Shorter molecules

Many molecules in the QMugs database have many atoms. To investigate the influence of the SMILES length, we created a subset of the QMugs database in which the SMILES have less than 50 characters.

11.3.2 Precision of the prompt

In our current form of performing inverse design, we create the prompts specifying the desired HOMO-LUMO gap as a floating point number rounded to a specific number of decimal points (by default 2). To investigate the influence of this parameter, we also tested a model in which we rounded to only one decimal point.

11.3.3 Extrapolation

In order for a generative model to be more useful than HTVS, it must be able to generate molecules that lie outside the training distribution. To test GPT-3's ability to do so, we truncated the QMugs distribution to only include molecules with bandgaps smaller than 3.50 eV.

11.3.4 Biasing the generation

To investigate if we can use GPT-3 to shift the generated distribution far from the training distribution, we utilized an iterative approach:

Algorithm

Bootstrapping To start the workflow

1. Fine-tune GPT-3 in inverse setting on a random sample
2. Query from a biased distribution, with mean shifted by α with respect to the current mean
3. Evaluate n molecules

Inner loop While the goal is not achieved, do

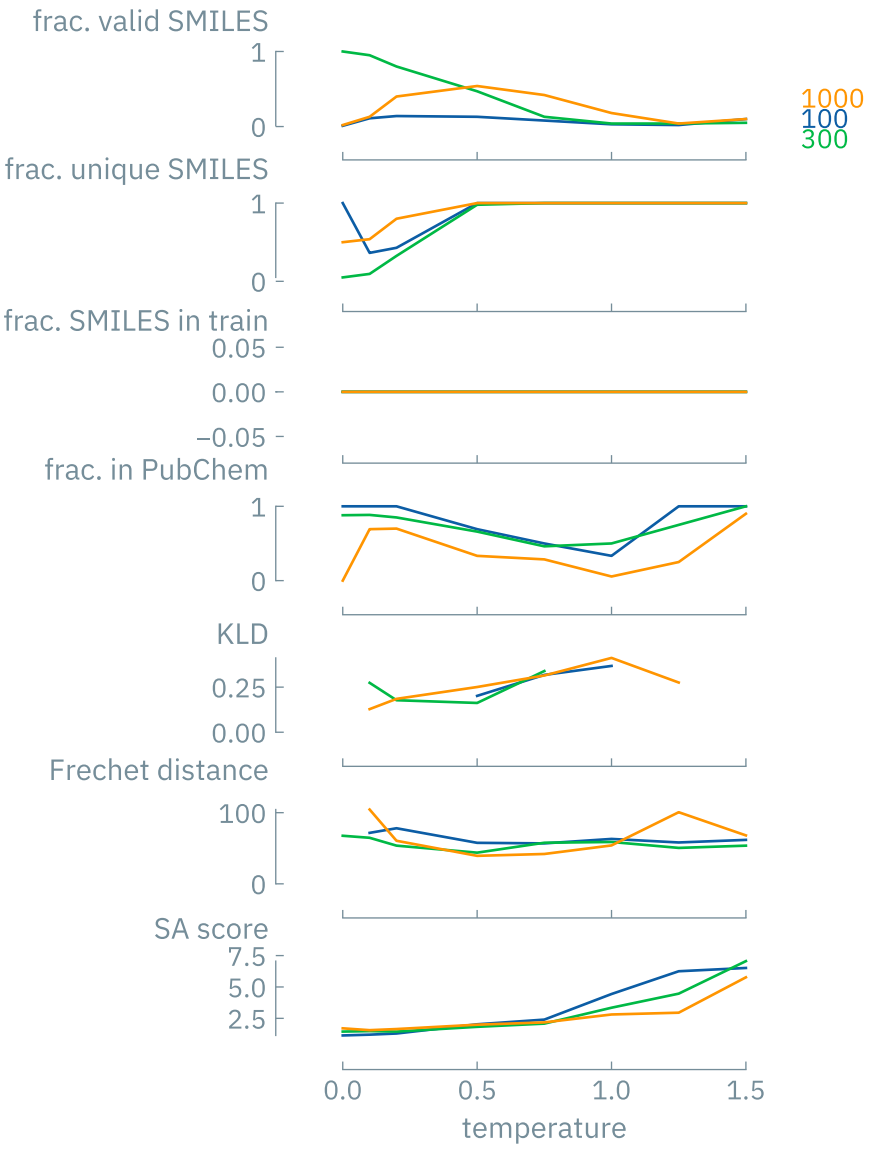
1. Combine all labeled data
2. Select a training set that is biased toward the target distribution (e.g., truncated with a lower bound)
3. Fine-tune a model
4. Query from a biased distribution, with mean shifted by α with respect to the current mean
5. Evaluate n molecules

Different batch sizes For the experiment in the main text, we tested the following number of evaluations: 2252 in the first, 1997 in the second, 370 in the third, 1875 in the fourth, and 1572 in the last.

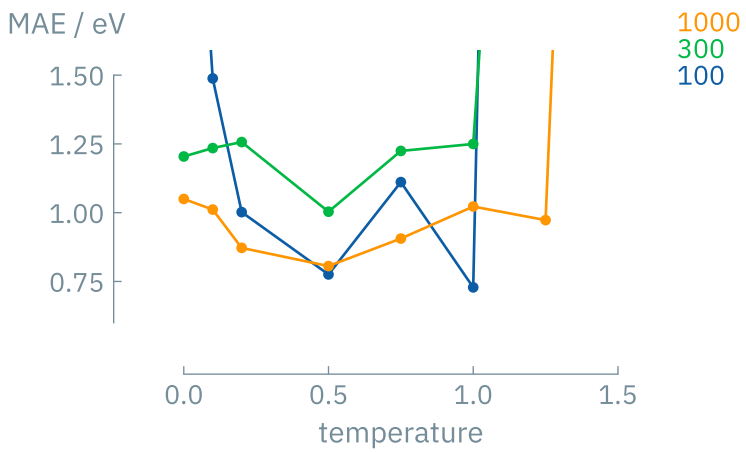
We also performed the experiment by constraining the number of molecules we evaluate using xTB to 100. Note that we queried many more molecules from the biased distribution and then *randomly* selected 100 from this distribution for evaluation using xTB and we can also successfully shift the distribution (Supplementary Figure 83). This can be improved in future work by ranking the candidates using a ML surrogate model.

11.4 Enrichment

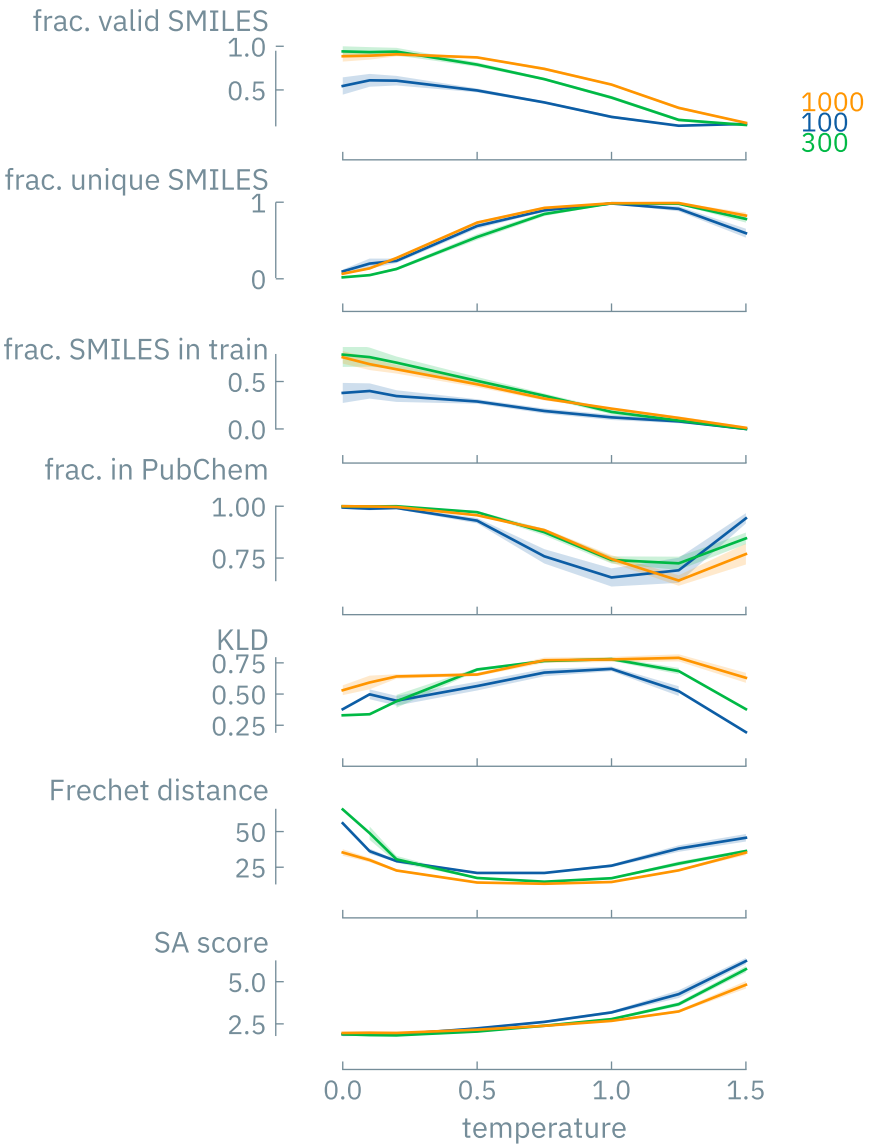
In many practical applications, one has more requirements than the correct HOMO-LUMO gap. Suppose we need a bromine-containing material with a well-defined HOMO-LUMO gap. Without additional training, we queried our model using `What is a molecule with a HOMO-LUMO gap of 3.5 eV and Br as part of the molecule?` Supplementary Figure 85 shows the results of such questions for different functional groups. We do not necessarily generate more molecules with the desired functional groups if we take the low-temperature results. However, at higher temperatures, where we allow for more creativity, we generate molecules with as much as ten times more structures with the desired functional group than the query without specifying the functional groups. As we did not train the model for this question, it is unclear whether it can recognize Br as an element and add it to a chemically meaningful place in the SMILES string. Yet it is known that these models can give meaningful answers without any training (zero-shot). It is fascinating that the GPT model can connect a SMILES string and the part of the query that Br should be part of the molecule.



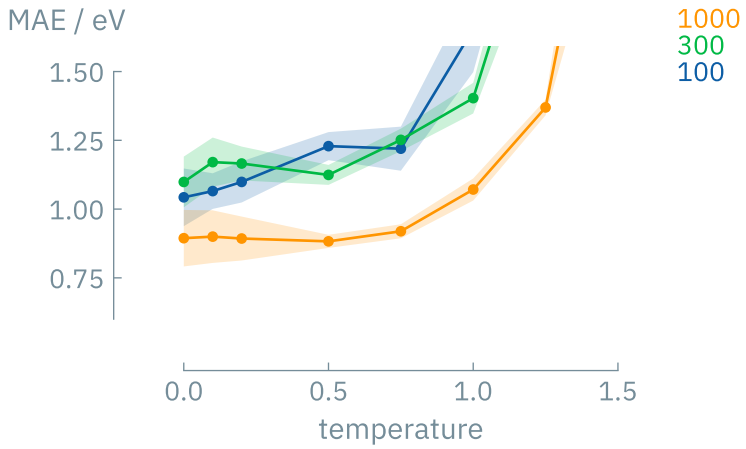
Supplementary Figure 78 | SMILES quality metrics for random generation of SMILES for an inverse model trained on the QMugs dataset. Colors indicate the training set size.



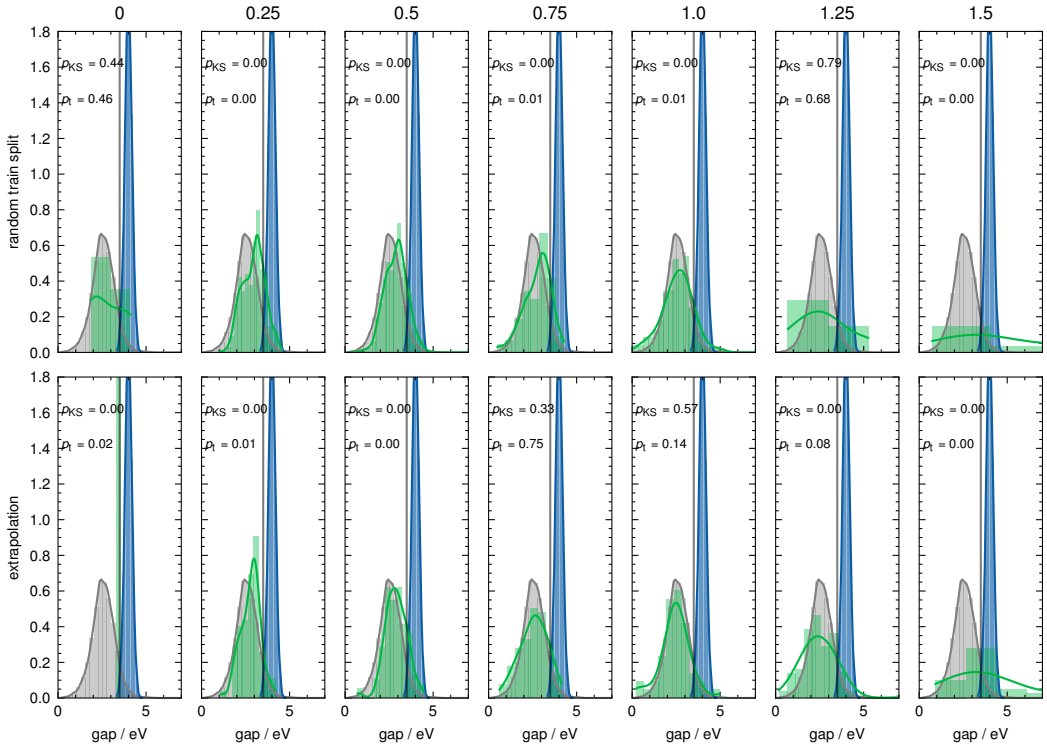
Supplementary Figure 79 | Constrain statisfaction for random generation of SMILES for an inverse model trained on the QMugs dataset. Colors indicate the training set size.



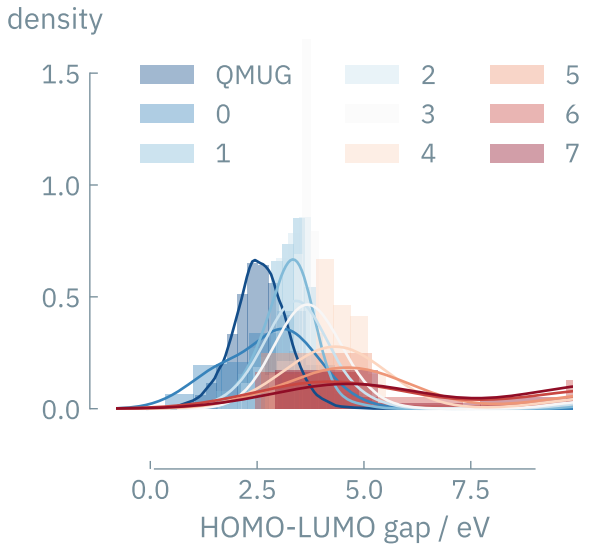
Supplementary Figure 80 | SMILES quality metrics for random generation of SMILES for an inverse model trained on the short-SMILES subset QMugs dataset. Colors indicate the training set size. Error bands show the standard error of the mean.



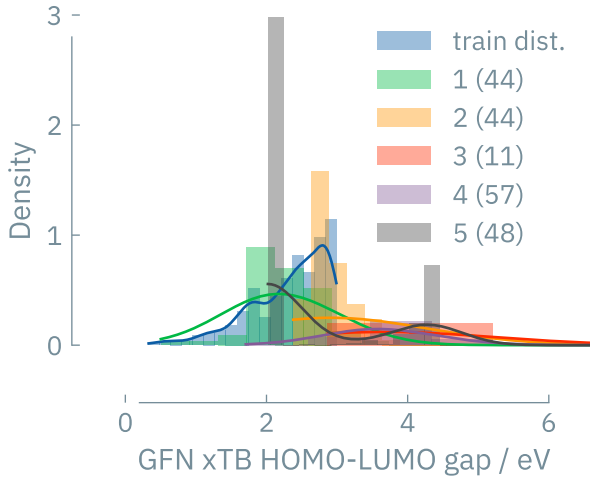
Supplementary Figure 81 | Constraint satisfaction metrics for random generation of SMILES for an inverse model trained on the short-SMILES subset QMugs dataset. Colors indicate the training set size. Error bands show the standard error of the mean.



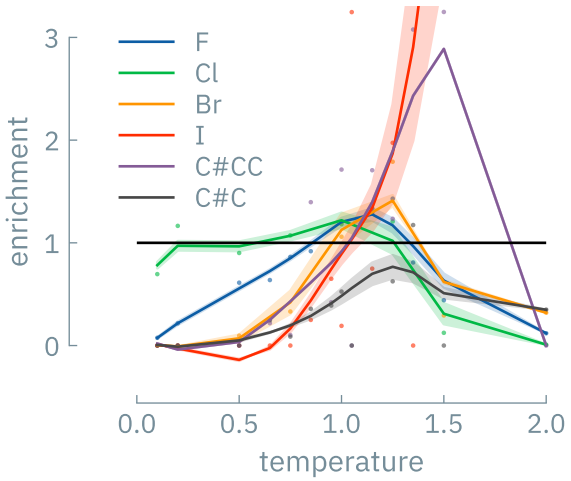
Supplementary Figure 82 | Distribution of HOMO-LUMO gaps of molecules generated by GPT-3 fine-tuned on the random and truncated training set, respectively. The vertical line indicates 3.50 eV, the threshold above which we excluded molecules from the extrapolation training set (bottom row). p-values in the inset are computed using Kolmogorov-Smirnov (p_{KS}) and t-tests (p_{t}) between the full distribution of HOMO-LUMO gaps of the QMugs database and the one of the generated molecules.



Supplementary Figure 83 | Generated distribution of HOMO-LUMO gaps for biased inverse models. For this experiment, we limited the number of molecules we evaluated using xTB to around 100. In detail, we queried xTB 101, 101, 101, 101, 101, 92, 101, and 85 times.



Supplementary Figure 84 | Generated distributions of HOMO-LUMO gaps for biased inverse models based on the `curie` base model. The numbers in the legend indicate the biasing step; the numbers in the parentheses show the number of times we called the simulation.

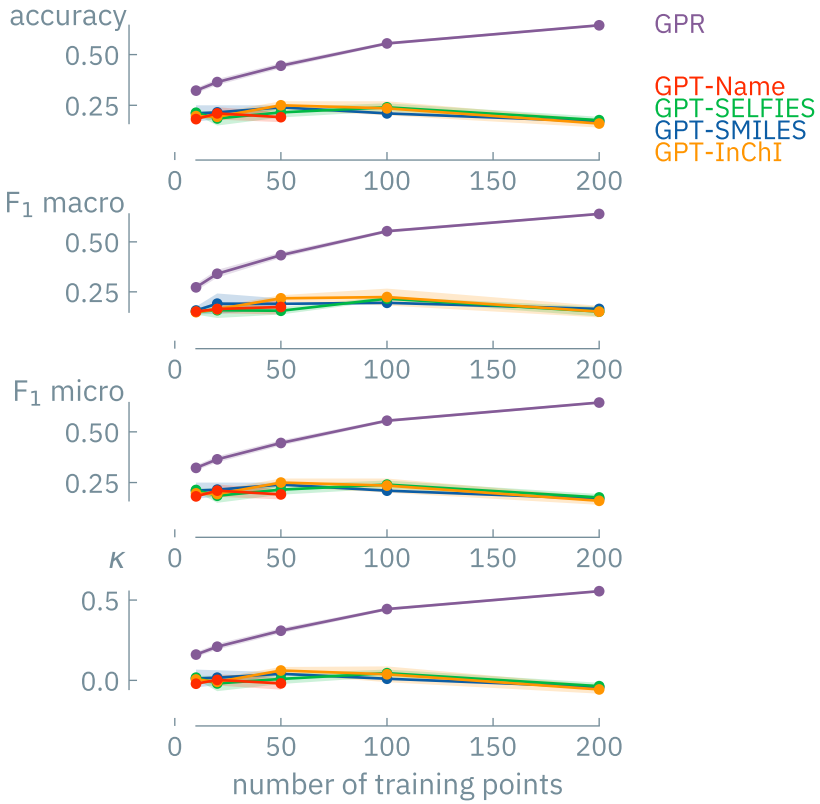


Supplementary Figure 85 | Generating molecules with desired functional groups. The figure shows the enrichment, i.e., the ratio of the fraction of functional groups in the generated molecules to their occurrence in QMugs for different functional groups as a function of temperature (see Supplementary Note 14). The first entries in the legend show the simple addition of halogens. The last two show the addition of more complex functionalities (alkyne, methyl-alkyne). For all points above the horizontal line, the generated molecules contain the desired functional group more frequently than in the QMugs distribution. For the smooth curves, we performed local polynomial regression (Gaussian kernel of width 0.25 and degree 2). Original data points are shown with dots. Errorbands obtained by fits to the lower and upper bounds obtained via the standard error of the mean.

Supplementary Note 12 Permutation test

To test if the model extracted meaningful structure-property relationships, we performed permutation tests (on the classification task on the photoswitch dataset). For this, we randomly shuffled the target column and fine-tuned and tested GPT-3 on this data.

Supplementary Figure 86 | Learning curves for binary classification for GPT-3 models fine-tuned on shuffled versions of the photoswitch dataset. Learning curve for the GPR model trained on unshuffled data shown for reference. Error bands show the standard error of the mean.



Supplementary Figure 87 | Learning curves for 5-class classification for GPT-3 models fine-tuned on shuffled versions of the photoswitch dataset. Learning curve for the GPR model trained on unshuffled data shown for reference. Error bands show the standard error of the mean.

Supplementary Note 13 Invalid inputs

ML models can confidently hallucinate. To test the behavior of our fine-tuned models, we investigated different inputs: random names, random combinations of elemental symbols as well as random combinations of letters.

13.1 Forward classification model

Supplementary Tab. 9: **Completions for a classification model trained on the photoswitch dataset when fed with inputs that are not a SMILES string.**

prompt	completion
What is the transition wavelength of Berend?	0
What is the transition wavelength of Kevin?	0
What is the transition wavelength of Philippe?	0
What is the transition wavelength of Andres?	0
What is the transition wavelength of Bus?	0
What is the transition wavelength of car?	0
What is the transition wavelength of tree?	0
What is the transition wavelength of house?	0
What is the transition wavelength of cat?	0
What is the transition wavelength of magnificent?	0
What is the transition wavelength of a Berend?	0
What is the transition wavelength of a Kevin?	0
What is the transition wavelength of a Philippe?	0
What is the transition wavelength of an Andres?	0
What is the transition wavelength of a Bus?	0
What is the transition wavelength of a car?	0

Supplementary Tab. 10: **Completions for a classification model trained on the photoswitch dataset when fed with inputs that are not a SMILES string and additional change to the prompt template.**

prompt	completion
what is the adsorption energy of Berend?	0
what is the adsorption energy of Kevin?	0
what is the adsorption energy of Philippe?	0
what is the adsorption energy of Andres?	0
what is the adsorption energy of Bus?	0
what is the adsorption energy of car?	0
what is the adsorption energy of tree?	0
what is the adsorption energy of house?	0
what is the adsorption energy of cat?	0
what is the adsorption energy of magnificent?	0
what is the adsorption energy of a Berend?	0
what is the adsorption energy of a Kevin?	0
what is the adsorption energy of a Phlippe?	0
what is the adsorption energy of an Andres?	0
what is the adsorption energy of a Bus?	0
what is the adsorption energy of a car?	0

Supplementary Tab. 11: **Completions for a classification model trained on the photoswitch dataset when fed with inputs that are not a SMILES string and not even valid words.**

prompt	completion
What is the transition wavelength of jtjei?	0
What is the transition wavelength of jytcs?	0
What is the transition wavelength of cymtv?	0
What is the transition wavelength of ntnlz?	0
What is the transition wavelength of oqeze?	0
What is the transition wavelength of linwg?	0
What is the transition wavelength of ktgje?	0
What is the transition wavelength of cjmqi?	0
What is the transition wavelength of srved?	0

Supplementary Tab. 12: **Completions for a classification model trained on the photoswitch dataset when fed with inputs that are not a SMILES string but a random combination of element symbols.**

prompt	completion
What is the transition wavelength of PFeScNaFe?	0
What is the transition wavelength of ClMnCuScBe?	0
What is the transition wavelength of GeClNaCFe?	0
What is the transition wavelength of BeCuGeZnFe?	0
What is the transition wavelength of CoGeNaMnCr?	0
What is the transition wavelength of CaFOGaNi?	0
What is the transition wavelength of NaKNeNiN?	0
What is the transition wavelength of GeNeMgMnMg?	0
What is the transition wavelength of ZnNiZnFF?	0
What is the transition wavelength of CaTiAsCoMg?	0

13.2 Inverse design model

The results for invalid inputs to an inverse design model are shown in Supplementary Table 13–Supplementary Table 15.

Supplementary Tab. 13: **Completions for an inverse design model trained on the photoswitch dataset when fed with inputs that are not a SMILES string (but valid words or names).**

prompt	completion
What is a molecule with E isomer transition wavelength Berend, Z isomer transition wavelength Berend?	CC1=C(C(C)=NN1)/N=N/C2=CC=C(C(F)(F)F)C=C2
What is a molecule with E isomer transition wavelength Kevin, Z isomer transition wavelength Kevin?	CC1=C(C(C)=NN1)/N=N/C2=CC=C(C(F)(F)F)C=C2
What is a molecule with E isomer transition wavelength Philippe, Z isomer transition wavelength Philippe?	CC1=C(C(C)=NN1)/N=N/C2=CC=C(C(F)(F)F)C=C2
What is a molecule with E isomer transition wavelength Andres, Z isomer transition wavelength Andres?	CC1=C(C(C)=NN1)/N=N/C2=CC(C(F)(F)F)=CC=C2
What is a molecule with E isomer transition wavelength Bus, Z isomer transition wavelength Bus?	CC1=C(C(C)=NN1)/N=N/C2=CC=C(Br)C=C2
What is a molecule with E isomer transition wavelength car, Z isomer transition wavelength car?	CC1=C(C(C)=NN1)/N=N/C2=CC=C(C(F)(F)F)C=C2
What is a molecule with E isomer transition wavelength tree, Z isomer transition wavelength tree?	CC1=C(C(C)=NN1)/N=N/C2=CC=C(C)C=C2
What is a molecule with E isomer transition wavelength house, Z isomer transition wavelength house?	CC1=C(C(C)=NN1)/N=N/C2=CC=C(C(F)(F)F)C=C2
What is a molecule with E isomer transition wavelength cat, Z isomer transition wavelength cat?	CC1=C(C(C)=NN1)/N=N/C2=CC=C(C(F)(F)F)C=C2
What is a molecule with E isomer transition wavelength magnificent, Z isomer transition wavelength magnificent?	CC1=C(C(C)=NN1)/N=N/C2=CC=C(C(F)(F)F)C=C2
What is a molecule with E isomer transition wavelength a Berend, Z isomer transition wavelength a Berend?	CC1=C(C(C)=NN1)/N=N/C2=CC=C(C(F)(F)F)C=C2
What is a molecule with E isomer transition wavelength a Kevin, Z isomer transition wavelength a Kevin?	CC1=C(C(C)=NN1)/N=N/C2=CC(NC(C)=O)=CC=C2
What is a molecule with E isomer transition wavelength a Philippe, Z isomer transition wavelength a Philippe?	CC1=C(C(C)=NN1)/N=N/C2=CC=C(C(F)(F)F)C=C2
What is a molecule with E isomer transition wavelength an Andres, Z isomer transition wavelength an Andres?	CC1=C(C(C)=NN1)/N=N/C2=CC=C(C(F)(F)F)C=C2
What is a molecule with E isomer transition wavelength a Bus, Z isomer transition wavelength a Bus?	CC1=C(C(C)=NN1)/N=N/C2=CC=C(Br)C=C2
What is a molecule with E isomer transition wavelength a car, Z isomer transition wavelength a car?	CC1=C(C(C)=NN1)/N=N/C2=CC=C(C(F)(F)F)C=C2

Supplementary Tab. 14: **Completions for an inverse design model trained on the photoswitch dataset when fed with inputs that are not a SMILES string (but random combinations of element symbols).**

prompt	completion
What is a molecule with E isomer transition wavelength PFeScNaFe, Z isomer transition wavelength PFeScNaFe?	CC1=C(C(C)=NN1)/N=N/C2=CC=C(C(F)(F)F)C=C2
What is a molecule with E isomer transition wavelength ClMnCuScBe, Z isomer transition wavelength ClMnCuScBe?	CC1=C(C(C)=NN1)/N=N/C2=CC=C(C(F)(F)F)C=C2
What is a molecule with E isomer transition wavelength GeClNaCFe, Z isomer transition wavelength GeClNaCFe?	CC1=C(C(C)=NN1)/N=N/C2=CC=C(C(F)(F)F)C=C2
What is a molecule with E isomer transition wavelength BeCuGeZnFe, Z isomer transition wavelength BeCuGeZnFe?	CC1=C(C(C)=NN1)/N=N/C2=CC=C(C(F)(F)F)C=C2
What is a molecule with E isomer transition wavelength CoGeNaMnCr, Z isomer transition wavelength CoGeNaMnCr?	CC1=C(C(C)=NN1)/N=N/C2=CC=C(C(F)(F)F)C=C2
What is a molecule with E isomer transition wavelength CaFOGaNi, Z isomer transition wavelength CaFOGaNi?	CC1=C(C(C)=NN1)/N=N/C2=CC(C(F)(F)F)=CC=C2
What is a molecule with E isomer transition wavelength NaKNeNiN, Z isomer transition wavelength NaKNeNiN?	CC1=C(C(C)=NN1)/N=N/C2=CC(C(F)(F)F)=CC=C2
What is a molecule with E isomer transition wavelength GeNeMgMnMg, Z isomer transition wavelength GeNeMgMnMg?	CC1=C(C(C)=NN1)/N=N/C2=CC=C(C(F)(F)F)C=C2
What is a molecule with E isomer transition wavelength ZnNiZnFF, Z isomer transition wavelength ZnNiZnFF?	CC1=C(C(C)=NN1)/N=N/C2=CC=C(C)C=C2
What is a molecule with E isomer transition wavelength CaTiAsCoMg, Z isomer transition wavelength CaTiAsCoMg?	CC1=C(C(C)=NN1)/N=N/C2=CC(C(F)(F)F)=CC=C2

Supplementary Tab. 15: **Completions for an inverse design model trained on the photoswitch dataset when fed with inputs that are not a SMILES string (but random combinations of random letters).**

prompt	completion
What is a molecule with E isomer transition wavelength jtjei, Z isomer transition wavelength jtjei?	CC1=C(C(C)=NN1)/N=N/C2=CC(C(F)(F)F)=CC=C2
What is a molecule with E isomer transition wavelength jytcs, Z isomer transition wavelength jytcs?	CC1=C(C(C)=NN1)/N=N/C2=CC(C(F)(F)F)=CC=C2
What is a molecule with E isomer transition wavelength cymtv, Z isomer transition wavelength cymtv?	CC1=C(C(C)=NN1)/N=N/C2=CC(C(F)(F)F)=CC=C2
What is a molecule with E isomer transition wavelength ntnlz, Z isomer transition wavelength ntnlz?	CC1=C(C(C)=NN1)/N=N/C2=CC(C(O)=O)=CC=C2
What is a molecule with E isomer transition wavelength oqeze, Z isomer transition wavelength oqeze?	CC1=C(C(C)=NN1)/N=N/C2=CC(C(F)(F)F)=CC=C2
What is a molecule with E isomer transition wavelength linwg, Z isomer transition wavelength linwg?	CC1=C(C(C)=NN1)/N=N/C2=CC=C(C(F)(F)F)C=C2
What is a molecule with E isomer transition wavelength ktgje, Z isomer transition wavelength ktgje?	CC1=C(C(C)=NN1)/N=N/C2=CC(C(F)(F)F)=CC=C2
What is a molecule with E isomer transition wavelength cjmqi, Z isomer transition wavelength cjmqi?	CC1=C(C(C)=NN1)/N=N/C2=CC(C(O)=O)=CC=C2
What is a molecule with E isomer transition wavelength srved, Z isomer transition wavelength srved?	CC1=C(C(C)=NN1)/N=N/C2=CC=C(C(F)(F)F)C=C2
What is a molecule with E isomer transition wavelength xjkgl, Z isomer transition wavelength xjkgl?	CC1=C(C(C)=NN1)/N=N/C2=CC(C(F)(F)F)=CC=C2

Supplementary Note 14 Novel tasks

We also investigated how our fine-tuned inverse design models can perform zero-shot tasks.

For this, we added statements such as "and F as part of the molecule?" to the end of the prompt and measured if the generated molecules were enriched with the specified functional group. To quantify potential changes in distribution, we randomly sampled from the distribution of properties on the training dataset and then computed the enrichment factor. We queried a model fine-tuned on 1000 randomly sampled molecules from QMugs.

$$\text{enrichment factor} = \frac{\text{fraction in valid SMILES in generated batch}}{\text{fraction in training set}} \tag{1}$$

Note that we do use an indicator function that only considers if a functional group is a molecule or not to compute the fractions. We do not consider the count of a given functional group in a molecule.

References

1. Crespi, S.; Simeth, N. A.; König, B. Heteroaryl azo dyes as molecular photoswitches. *Nat Rev Chem* **2019**, *3*, 133–146.

2. 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. *Chem. Sci.* **2022**, *13*, 13541–13551.

3. Isert, C.; Atz, K.; Jiménez-Luna, J.; Schneider, G. QMugs, quantum mechanical properties of drug-like molecules. *Sci Data* **2022**, *9*.

4. Savjani, K. T.; Gajjar, A. K.; Savjani, J. K. Drug Solubility: Importance and Enhancement Techniques. *ISRN Pharmaceutics Pharmaceutics* **2012**, *2012*, 1–10.

5. Walters, P. Predicting Aqueous Solubility - It's Harder Than It Looks. https://practicalcheminformatics.blogspot.com/2018/09/predicting-aqueous-solubility-its.html, last accessed 06 Feb 2023.

6. Delaney, J. S. ESOL: Estimating Aqueous Solubility Directly from Molecular Structure. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 1000–1005.

7. Mitchell, J. B. O. DLS-100 Solubility Dataset. 2017; https://risweb.st-andrews.ac.uk:443/portal/en/datasets/dls100-solubility-dataset(3a3a5abc-8458-4924-8e6c-b804347605e8).html.

8. Kearnes, S.; McCloskey, K.; Berndl, M.; Pande, V.; Riley, P. Molecular graph convolutions: moving beyond fingerprints. *J Comput Aided Mol Des* **2016**, *30*, 595–608.

9. Duvenaud, D. K.; Maclaurin, D.; Iparraguirre, J.; Bombarell, R.; Hirzel, T.; Aspuru-Guzik, A.; Adams, R. P. Convolutional Networks on Graphs for Learning Molecular Fingerprints. Advances in Neural Information Processing Systems. 2015.

10. Mobley, D. L.; Guthrie, J. P. FreeSolv: a database of experimental and calculated hydration free energies, with input files. *J. Comput. Aided Mol. Des.* **2014**, *28*, 711–720.

11. Waring, M. J. Lipophilicity in drug discovery. *Expert Opinion on Drug Discovery* **2010**, *5*, 235–248.

12. Lyu, H.; Ji, Z.; Wuttke, S.; Yaghi, O. M. Digital Reticular Chemistry. *Chem* **2020**, *6*, 2219–2241.

13. Yaghi, O. M. Reticular chemistry in all dimensions. *ACS Cent. Sci.* **2019**, *5*, 1295–1300.

14. 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.

15. Lin, L.-C.; Berger, A. H.; Martin, R. L.; Kim, J.; Swisher, J. A.; Jariwala, K.; Rycroft, C. H.; Bhown, A. S.; Deem, M. W.; Haranczyk, M.; Smit, B. In silico screening of carbon-capture materials. *Nature Mater* **2012**, *11*, 633–641.

16. Moosavi, S. M.; Novotny, B. Á.; Ongari, D.; Moubarak, E.; Asgari, M.; Özge Kadioglu,; Charalambous, C.; Ortega-Guerrero, A.; Farmahini, A. H.; Sarkisov, L.; Garcia, S.; Noé, F.; Smit, B. A data-science approach to predict the heat capacity of nanoporous materials. *Nat. Mater.* **2022**, *21*, 1419–1425.

17. Burtch, N. C.; Jasuja, H.; Walton, K. S. Water Stability and Adsorption in Metal–Organic Frameworks. *Chem. Rev.* **2014**, *114*, 10575–10612.

18. Jablonka, K. M.; Jothiappan, G. M.; Wang, S.; Smit, B.; Yoo, B. Bias free multiobjective active learning for materials design and discovery. *Nat. Commun.* **2021**, *12*.

19. Yeh, J.-W.; Chen, S.-K.; Lin, S.-J.; Gan, J.-Y.; Chin, T.-S.; Shun, T.-T.; Tsau, C.-H.; Chang, S.-Y. Nanostructured High-Entropy Alloys with Multiple Principal Elements: Novel Alloy Design Concepts and Outcomes. *Adv. Eng. Mater.* **2004**, *6*, 299–303.

20. Cantor, B.; Chang, I.; Knight, P.; Vincent, A. Microstructural development in equiatomic multicomponent alloys. *Materials Science and Engineering: A* **2004**, *375-377*, 213–218.

21. George, E. P.; Raabe, D.; Ritchie, R. O. High-entropy alloys. *Nat Rev Mater* **2019**, *4*, 515–534.

22. Pei, Z.; Yin, J.; Hawk, J. A.; Alman, D. E.; Gao, M. C. Machine-learning informed prediction of high-entropy solid solution formation: Beyond the Hume-Rothery rules. *npj Comput Mater* **2020**, *6*.

23. Dunn, A.; Wang, Q.; Ganose, A.; Dopp, D.; Jain, A. Benchmarking materials property prediction methods: the Matbench test set and Automatminer reference algorithm. *npj Comput Mater* **2020**, *6*.

24. Kawazoe, Y., Yu, J.-Z., Tsai, A.-P., Masumoto, T., Eds. *Nonequilibrium phase diagrams of ternary amorphous alloys*; Landolt-Börnstein: Numerical Data and Functional Relationships in Science and Technology - New Series; Springer: New York, NY, 2006.

25. Zhuo, Y.; Tehrani, A. M.; Brgoch, J. Predicting the Band Gaps of Inorganic Solids by Machine Learning. *J. Phys. Chem. Lett.* **2018**, *9*, 1668–1673.

26. Pomberger, A.; McCarthy, A. A. P.; Khan, A.; Sung, S.; Taylor, C. J.; Gaunt, M. J.; Colwell, L.; Walz, D.; Lapkin, A. A. The effect of chemical representation on active machine learning towards closed-loop optimization. *React. Chem. Eng.* **2022**, *7*, 1368–1379.

27. Shields, B. J.; Stevens, J.; Li, J.; Parasram, M.; Damani, F.; Alvarado, J. I. M.; Janey, J. M.; Adams, R. P.; Doyle, A. G. Bayesian reaction optimization as a tool for chemical synthesis. *Nature* **2021**, *590*, 89–96.

28. Hickman, R.; Ruža, J.; Roch, L.; Tribukait, H.; García-Durán, A. Equipping data-driven experiment planning for Self-driving Laboratories with semantic memory: case studies of transfer learning in chemical reaction optimization. **2022**,

29. Ahneman, D. T.; Estrada, J. G.; Lin, S.; Dreher, S. D.; Doyle, A. G. Predicting reaction performance in C–N cross-coupling using machine learning. *Science* **2018**, *360*, 186–190.

30. Perera, D.; Tucker, J. W.; Brahmbhatt, S.; Helal, C. J.; Chong, A.; Farrell, W.; Richardson, P.; Sach, N. W. A platform for automated nanomole-scale reaction screening and micromole-scale synthesis in flow. *Science* **2018**, *359*, 429–434.

31. 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 preprint: Arxiv-2206.06565. 2022.

32. Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 31–36.

33. Krenn, M.; Häse, F.; Nigam, A.; Friederich, P.; Aspuru-Guzik, A. Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. *Mach. Learn.: Sci. Technol.* **2020**, *1*, 045024.

34. Krenn, M. et al. SELFIES and the future of molecular string representations. *Patterns* **2022**, *3*, 100588.

35. Lu, J.; Xia, S.; Lu, J.; Zhang, Y. Dataset Construction to Explore Chemical Space with 3D Geometry and Deep Learning. *Journal of Chemical Information and Modeling* **2021**, *61*, 1095–1104.

36. Ho, J.; Tumkaya, T.; Aryal, S.; Choi, H.; Claridge-Chang, A. Moving beyond P values: data analysis with estimation graphics. *Nature methods* **2019**, *16*, 565–566.

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

38. Lester, B.; Al-Rfou, R.; Constant, N. The Power of Scale for Parameter-Efficient Prompt Tuning. Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021. 2021; pp 3045–3059.

39. Singhal, K. et al. Large Language Models Encode Clinical Knowledge. *arXiv preprint arXiv: Arxiv-2212.13138* **2022**,

40. Wang, Y.; Wang, J.; Cao, Z.; Farimani, A. B. Molecular contrastive learning of representations via graph neural networks. *Nat. Mach. Intell.* **2022**, *4*, 279–287.

41. Hollmann, N.; Müller, S.; Eggensperger, K.; Hutter, F. TabPFN: A Transformer That Solves Small Tabular Classification Problems in a Second. arXiv preprint arXiv: Arxiv-2207.01848. 2022.

42. Griffiths, R.-R.; Klarner, L.; Moss, H.; Ravuri, A.; Truong, S. T.; Rankovic, B.; Du, Y.; Jamasb, A. R.; Schwartz, J.; Tripp, A., et al. GAUCHE: A Library for Gaussian Processes in Chemistry. ICML 2022 2nd AI for Science Workshop.

43. Nagasawa, S.; Al-Naamani, E.; Saeki, A. Computer-Aided Screening of Conjugated Polymers for Organic Solar Cell: Classification by Random Forest. *J. Phys. Chem. Lett.* **2018**, *9*, 2639–2646.

44. Mayr, A.; Klambauer, G.; Unterthiner, T.; Hochreiter, S. DeepTox: Toxicity Prediction using Deep Learning. *Front. Environ. Sci.* **2016**, *3*.

45. Huang, R.; Xia, M.; Nguyen, D.-T.; Zhao, T.; Sakamuru, S.; Zhao, J.; Shahane, S. A.; Rossoshek, A.; Simeonov, A. Tox21Challenge to Build Predictive Models of Nuclear Receptor and Stress Response Pathways as Mediated by Exposure to Environmental Chemicals and Drugs. *Front. Environ. Sci.* **2016**, *3*.

46. Moosavi, S. M.; Nandy, A.; Jablonka, K. M.; Ongari, D.; Janet, J. P.; Boyd, P. G.; Lee, Y.; Smit, B.; Kulik, H. J. Understanding the diversity of the metal-organic framework ecosystem. *Nat. Commun.* **2020**, *11*.

47. 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.

48. Wang, A. Y.-T.; Kauwe, S. K.; Murdock, R. J.; Sparks, T. D. Compositionally restricted attention-based network for materials property predictions. *npj Computational Materials* **2021**, *7*, 77.

49. Batra, R.; Chen, C.; Evans, T. G.; Walton, K. S.; Ramprasad, R. Prediction of water stability of metal–organic frameworks using machine learning. *Nat. Mach. Intell.* **2020**, *2*, 704–710.

50. Probst, D.; Schwaller, P.; Reymond, J.-L. Reaction classification and yield prediction using the differential reaction fingerprint DRFP. *Digital Discovery* **2022**, *1*, 91–97.

51. Schwaller, P.; Probst, D.; Vaucher, A. C.; Nair, V. H.; Kreutter, D.; Laino, T.; Reymond, J.-L. Mapping the space of chemical reactions using attention-based neural networks. *Nature Machine Intelligence* **2021**, *3*, 144–152.

52. Breuck, P.-P. D.; Evans, M. L.; Rignanese, G.-M. Robust model benchmarking and bias-imbalance in data-driven materials science: a case study on MODNet. *J. Phys.: Condens. Matter* **2021**, *33*, 404002.

53. Faber, F.; Lindmaa, A.; von Lilienfeld, O. A.; Armiento, R. Crystal structure representations for machine learning models of formation energies. *International Journal of Quantum Chemistry* **2015**, *115*, 1094–1101.

54. Breiman, L. *Machine Learning* **2001**, *45*, 5–32.

55. Ward, L.; Agrawal, A.; Choudhary, A.; Wolverton, C. A general-purpose machine learning framework for predicting properties of inorganic materials. *npj Computational Materials* **2016**, *2*.

56. Biderman, S.; Schoelkopf, H.; Anthony, Q. G.; Bradley, H.; O'Brien, K.; Hallahan, E.; Khan, M. A.; Purohit, S.; Prashanth, U. S.; Raff, E., et al. Pythia: A suite for analyzing large language models across training and scaling. International Conference on Machine Learning. 2023; pp 2397–2430.

57. Eriksson, D.; Jankowiak, M. High-dimensional Bayesian optimization with sparse axis-aligned subspaces. Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence, UAI 2021, Virtual Event, 27-30 July 2021. 2021; pp 493–503.

58. Probst, D.; Reymond, J.-L. Visualization of very large high-dimensional data sets as minimum spanning trees. *J. Cheminform.* **2020**, *12*.

59. Probst, D.; Reymond, J.-L. A probabilistic molecular fingerprint for big data settings. *J. Cheminform.* **2018**, *10*.

60. Frisch, M. J. et al. Gaussian˜16 Revision C.01. 2016; Gaussian Inc. Wallingford CT.

61. Jacquemin, D.; Preat, J.; Perpète, E. A.; Vercauteren, D. P.; André, J.-M.; Ciofini, I.; Adamo, C. Absorption spectra of azobenzenes simulated with time-dependent density functional theory. *Int. J. Quantum Chem.* **2011**, *111*, 4224–4240.

62. Adamo, C.; Barone, V. Toward reliable density functional methods without adjustable parameters: The PBE0 model. *J. Chem. Phys.* **1999**, *110*, 6158–6170.

63. Petersson, G. A.; Bennett, A.; Tensfeldt, T. G.; Al-Laham, M. A.; Shirley, W. A.; Mantzaris, J. A complete basis set model chemistry. I. The total energies of closed-shell atoms and hydrides of the first-row elements. *J. Chem. Phys.* **1988**, *89*, 2193–2218.

64. Petersson, G. A.; Al-Laham, M. A. A complete basis set model chemistry. II. Open-shell systems and the total energies of the first-row atoms. *J. Chem. Phys.* **1991**, *94*, 6081–6090.

65. Tomasi, J.; Mennucci, B.; Cammi, R. Quantum Mechanical Continuum Solvation Models. *Chem. Rev.* **2005**, *105*, 2999–3094.

66. Jablonka, K. M. givemeconformer. 2022; https://github.com/kjappelbaum/givemeconformer.

67. Halgren, T. A. Merck molecular force field. I. Basis, form, scope, parameterization, and performance of MMFF94. *J. Comput. Chem.* **1996**, *17*, 490–519.

Glossary

AUC area under the curve. 49

BO Bayesian optimization. 6

COF covalent-organic framework. 34

DFT density functional theory. 3, 71, 79, 105

ECFP extended connectivity fingerprint. 25

EFG extended functional group. 9

GNN graph neural network. 4, 12

GPR Gaussian process regression. 10, 12, 43, 77, 88, 89

GPT-3 generative pre-trained transformer model 3. 3, 5, 9, 10, 13–18, 20, 21, 38, 39, 41, 44, 45, 47, 49, 52, 57, 67, 78, 80, 88

HEA high entropy alloy. 5

HOMO highest occupied molecular orbital. 3, 22, 79

HTVS high-throughput virtual screening. 75, 80

InChI international chemical identifier. 55

IUPAC International Union for Pure and Applied Chemistry. 9, 13, 55

LLM large language model. 9, 10

LUMO lowest unoccupied molecular orbital. 3, 22, 79

MAE mean absolute error. 71, 79

MHFP MinHash fingerprint. 74, 78

ML machine learning. 5, 6, 40, 80, 90

MOF metal-organic framework. 4, 5, 34

NLP natural language processing. 43

OPV organic photovoltaics. 4, 25, 26, 63

PCE power conversion efficiency. 4, 25, 26

PCM Polarizable Continuum Model. 76

PFN Prior-Data Fitted Network. 12, 105

PPD posterior predictive distribution. 12

RF random forest. 4, 25, 30

ROC receiver operating curve. 49

SELFIES self-referencing embedded strings. 55

SMILES simplified molecular-input line-entry system. 7, 12, 43, 55

TabPFN tabular PFN. 12, 35

TDDFT time-dependent DFT. 3, 71, 76

TMAP tree-map. 74, 75, 78

xTB extended tight-binding. 79, 80, 86