GAUCHE: A Library for Gaussian Processes in Chemistry

Ryan-Rhys Griffiths[1*] **Leo Klarner**[2*] **Henry Moss**[3*] **Aditya Ravuri**[3*] **Sang Truong**[4*] **Samuel Stanton**[5*] **Gary Tom**[6,7*] **Bojana Rankovic**[8,9*] **Yuanqi Du**[10*] **Arian Jamasb**[3*]

Aryan Deshwal[11] **Julius Schwartz**[3] **Austin Tripp**[3] **Gregory Kell**[12] **Simon Frieder**[2] **Anthony Bourached**[13] **Alex J. Chan**[3] **Jacob Moss**[3] **Chengzhi Guo**[3] **Johannes Durholt**[14] **Saudamini Chaurasia**[15] **Ji Won Park**[5] **Felix Strieth-Kalthoff**[6]

Alpha A. Lee[3] **Bingqing Cheng**[16] **Alán Aspuru-Guzik**[6,7,17] **Philippe Schwaller**[8,9] **Jian Tang**[18,19,17]

[1]Meta [2]University of Oxford [3]University of Cambridge [4]Stanford University [5]Genentech [6]University of Toronto [7]Vector Institute [8]EPFL [9]NCCR Catalysis [10]Cornell University [11]Washington State University [12]King's College London [13]University College London [14]Evonik Industries AG [15]Syracuse University [16]IST Austria [17]CIFAR AI Research Chair [18]MILA Quebec AI Institute [19]HEC Montreal

[*] Equal contributions

{ryangriff123,leojklarner}@gmail.com

Abstract

We introduce GAUCHE, an open-source library for GAUssian processes in CHEmistry. Gaussian processes have long been a cornerstone of probabilistic machine learning, affording particular advantages for uncertainty quantification and Bayesian optimisation. Extending Gaussian processes to molecular representations, however, necessitates kernels defined over structured inputs such as graphs, strings and bit vectors. By providing such kernels in a modular, robust and easy-to-use framework, we seek to enable expert chemists and materials scientists to make use of state-of-the-art black-box optimization techniques. Motivated by scenarios frequently encountered in practice, we showcase applications for GAUCHE in molecular discovery, chemical reaction optimisation and protein design.

The codebase is made available at https://github.com/leojklarner/gauche.

1 Introduction

Early-stage scientific discovery is often characterised by the limited availability of high-quality experimental data [1, 2, 3], meaning that there is much knowledge to gain from targeted experiments. As such, machine learning methods that facilitate discovery in low data regimes, such as Bayesian optimisation (BO) [4, 5, 6, 7, 8, 9] and active learning (AL) [10, 11], have great potential to expedite the rate at which useful molecules, materials, chemical reactions and proteins can be discovered.

At present, Bayesian neural networks (BNNs) and deep ensembles are typically the method of choice to generate uncertainty estimates for molecular BO and AL loops [10, 12, 13, 14]. For small datasets, however, Gaussian processes (GPs) may often be a preferable and more appropriate choice [15, 16]. Furthermore, GPs possess particularly advantageous properties for BO; first, they admit exact as opposed to approximate Bayesian inference and second, few of their parameters need to be determined by hand. In the words of Sir David MacKay [17],

> "Gaussian processes are useful tools for automated tasks where fine tuning for each problem is not possible. We do not appear to sacrifice any performance for this simplicity."

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

https://doi.org/10.52202/075280-3362

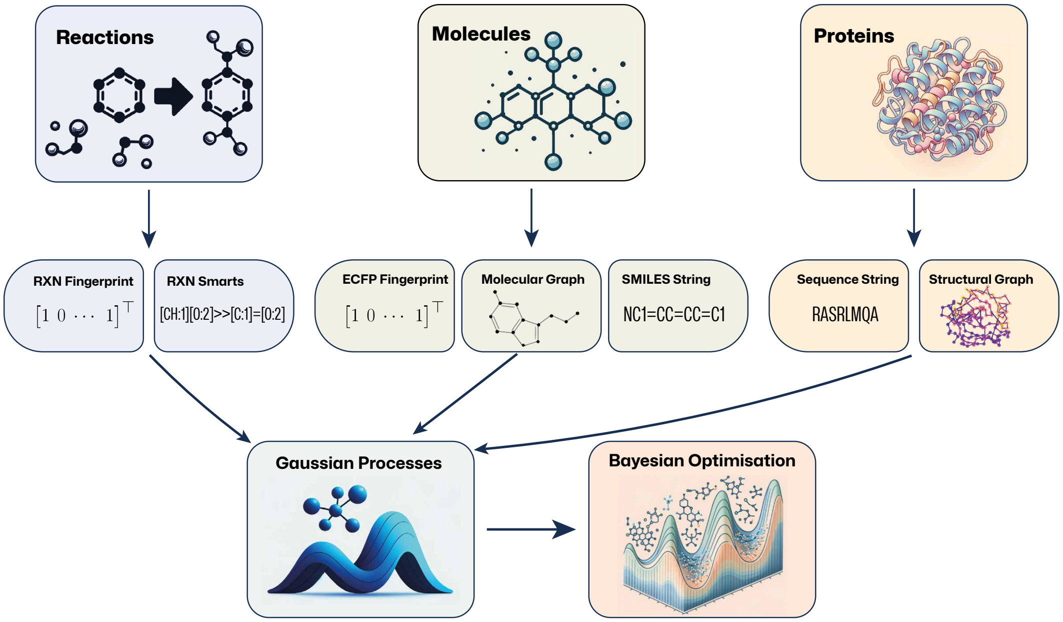


Figure 1: An overview of the applications and representations available in GAUCHE.

The iterative model refitting required in BO makes it a prime example of such an automated task. However, canonical GPs typically assume continuous input spaces of low and fixed dimensionality, hindering their application to standard molecular representations such as SMILES/SELFIES strings [18, 19, 20], topological fingerprints [21, 22, 23] and discrete graphs [24, 25].

With GAUCHE, we provide a modular, robust and easy-to-use framework to rapidly prototype GPs with 30+ GPU-accelerated string, fingerprint and graph kernels that operate on a range of molecular representations (see Figure 1). Furthermore, GAUCHE interfaces with the GPyTorch [26] and BoTorch [27] libraries and contains an extensive set of tutorial notebooks to make state-of-the-art probabilistic modelling and black-box optimization techniques more easily accessible to scientific experts in chemistry, materials science and beyond.

2 Background

We briefly recall the fundamentals of Gaussian processes and Bayesian optimisation in Sections 2.1 and 2.2, respectively, and refer the reader to [28] and [29, 30, 31] for a more comprehensive treatment.

2.1 Gaussian Processes

Notation $\mathbf{X} \in \mathbb{R}^{n \times d}$ is a design matrix of n training examples of dimension d. A given row i of the design matrix contains a training molecule's representation $\mathbf{x}_i$. A GP is specified by a mean function, $m(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$ and a covariance function $k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))]$. $K_\theta(\mathbf{X}, \mathbf{X})$ is a kernel matrix, where entries are computed by the kernel function as $[K]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ and θ represents the set of kernel hyperparameters. The GP specifies the full distribution over the function f to be modelled as

$$f(\mathbf{x}) \sim \mathcal{GP}\big(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\big).$$

Training Hyperparameters for GPs comprise kernel hyperparameters, θ, in addition to the likelihood noise, σ_y^2. These hyperparameters are chosen by optimising an objective function known as the negative log marginal likelihood (NLML)

$$\log p(\mathbf{y}|\mathbf{X}, \theta) = \underbrace{-\frac{1}{2}\mathbf{y}^\top (K_\theta(\mathbf{X}, \mathbf{X}) + \sigma_y^2 I)^{-1}\mathbf{y}}_{\text{encourages fit with data}} \underbrace{-\frac{1}{2}\log|K_\theta(\mathbf{X}, \mathbf{X}) + \sigma_y^2 I|}_{\text{controls model capacity}} - \frac{N}{2}\log(2\pi),$$

https://doi.org/10.52202/075280-3362

where $\sigma_y^2 I$ represents the variance of i.i.d. Gaussian noise on the observations $\mathbf{y}$. The NLML embodies Occam's razor for Bayesian model selection [28] in favouring models that fit the data without being overly complex.

Prediction At test locations $\mathbf{X}_*$, assuming a zero mean function obtained following the standardization of the outputs $\mathbf{y}$, the GP returns a predictive mean, $\bar{\mathbf{f}}_* = K(\mathbf{X}_*, \mathbf{X})[K(\mathbf{X}, \mathbf{X}) + \sigma_y^2 I]^{-1}\mathbf{y}$, and a predictive uncertainty $\text{cov}(\mathbf{f}_*) = K(\mathbf{X}_*, \mathbf{X}_*) - K(\mathbf{X}_*, \mathbf{X})[K(\mathbf{X}, \mathbf{X}) + \sigma_y^2 I]^{-1} K(\mathbf{X}, \mathbf{X}_*)$.

2.2 Bayesian Optimisation

In molecular discovery campaigns, we are typically interested in solving problems of the form

$$\mathbf{x}^\star = \arg\max_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}),$$

where $f(\cdot) : \mathcal{X} \to \mathbb{R}$ is an expensive black-box function over a structured input domain $\mathcal{X}$. In our setting the structured input domain consists of a set of molecular representations (graphs, strings, bit vectors) and the expensive black-box function is an experimentally determined property of interest that we wish to optimise. Bayesian optimisation (BO) [32, 33, 34, 35, 29, 36] is a data-efficient methodology for determining $\mathbf{x}^\star$. BO operates sequentially by selecting input locations at which to query the black-box function f with the aim of identifying the optimum in as few queries as possible. Evaluations are focused on promising areas of the input space as well as areas with high uncertainty—a balancing act known as the exploration/exploitation trade-off.

The two components of a BO scheme are a probabilistic surrogate model and an acquisition function. The surrogate model is typically chosen to be a GP due to its ability to maintain calibrated uncertainty estimates through exact Bayesian inference. The uncertainty estimates of the surrogate model are then leveraged by the acquisition function to propose new input locations to query. The acquisition function is a heuristic that trades off exploration and exploitation, well-known examples of which include expected improvement (EI) [33, 35] and entropy search [37, 38, 39, 40]. After the acquisition function proposes an input location, the black-box is evaluated at that location, the surrogate model is retrained and the process is repeated until a solution is obtained.

3 Molecular Representations

We review commonly used representations for molecules (Section 3.1), chemical reactions (Section 3.2) and proteins (Section 3.3), before describing the kernels that operate on them in Section 4. An overview of the representations considered by GAUCHE is provided in Figure 1.

3.1 Molecules

Graphs A molecule can be represented as an undirected, labelled graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where vertices $\mathcal{V} = \{v_1, \ldots, v_N\}$ represent the atoms of an N-atom molecule and edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ represent covalent bonds between these atoms. Additional information may be incorporated in the form of vertex and edge labels $\mathcal{L} : \mathcal{V} \times \mathcal{E} \to \Sigma_V \times \Sigma_E$ by specifying e.g. atom types or bond orders.

Fingerprints Molecular fingerprints enumerate sets or bags of subgraphs $\mathcal{G}' = (\mathcal{V}' \subseteq \mathcal{V}, \mathcal{E}' \subseteq \mathcal{E})$ of a certain type and then hash them into machine-readable bit or count vectors. Extended connectivity fingerprints (ECFPs) [21], for example, enumerate all circular subgraphs up to a pre-specified radius parameter by assigning initial numeric identifiers to each atom in a molecule and iteratively updating them based on the identifiers of their neighbours. Each level of iteration appends substructural features of increasing non-locality to an array, which is then hashed to a bit vector reflecting the presence or absence of those substructures in the molecule (see Figure 2). We choose a radius of 3 for all experiments in the main text and provide a more detailed ablation of the radius parameter in Appendix E.3.

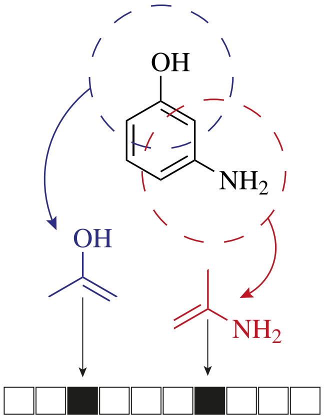


Figure 2: Visualisation of the ECFP subgraph enumeration and hashing procedures.

https://doi.org/10.52202/075280-3362

Additionally, we make use of fragment descriptors, which are count vectors in which each component indicates the count of a particular functional group present in a molecule, as well as the concatenation of the fingerprint and fragment feature vectors, a representation termed fragprints [3], which has shown strong empirical performance. Example representations of $\mathbf{x_{fp}}$ for fingerprints, $\mathbf{x_{fr}}$ for fragments and $\mathbf{x_{frp}}$ for fragprings are given as

$$\mathbf{x_{fp}} = \begin{bmatrix}1 & 0 & \cdots & 1\end{bmatrix}^\top \mathbf{x_{fr}} = \begin{bmatrix}3 & 0 & \cdots & 2\end{bmatrix}^\top \mathbf{x_{frp}} = \begin{bmatrix}1 & 0 & \cdots & 1 & 3 & 0 & \cdots & 2\end{bmatrix}^\top$$

Strings The Simplified Molecular-Input Line-Entry System (SMILES) is a text-based representation of molecules [18, 19], examples of which are given in Figure 3. Self-Referencing Embedded Strings (SELFIES) [20] is an alternative string representation to SMILES such that a bijective mapping exists between a SELFIES string and a molecule.

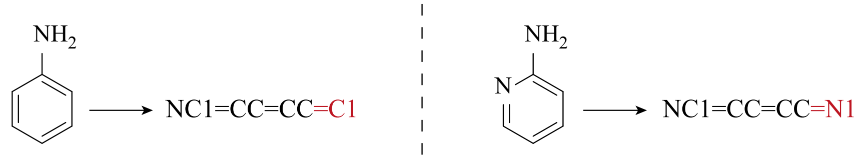


Figure 3: SMILES strings for structurally similar molecules. Similarity is encoded in the string through common contiguous subsequences (black). Local differences are highlighted in red.

3.2 Reaction Representations

Chemical reactions consist of (multiple) reactants and reagents that react to form one or more products. The choice of reactant/reagent typically constitutes a categorical design space. Taking as an example the high-throughput experiments by [41] on Buchwald-Hartwig reactions, the reaction design space consists of 15 aryl and heteroaryl halides, 4 Buchwald ligands, 3 bases, and 23 isoxazole additives.

Concatenated Molecular Representations If the number of reactants and reagents is constant, the molecular representations discussed in Section 3.1 may be used to represent them, and the vectors for the individual reaction components can be concatenated to construct a representation of the reaction [41, 42]. An additional and commonly used concatenated representation is the one-hot-encoding (OHE) of the reaction categories where bits specify which of the components in the different reactant and reagent categories is present. In the Buchwald-Hartwig example, the OHE would describe which of the aryl halides, Buchwald ligands, bases and additives are used in the reaction, resulting in a 44-dimensional bit vector [43].

Differential Reaction Fingerprints Inspired by the hand-engineered difference reaction fingerprints by [44] and [45] recently introduced the differential reaction fingerprint (DRFP). This reaction fingerprint is constructed by taking the symmetric difference of the sets containing the molecular substructures on both sides of the reaction arrow. Reagents are added to the reactants. The size of the reaction bit vector generated by DRFP is independent of the number of reaction components.

Data-Driven Reaction Fingerprints [46] described data-driven reaction fingerprints using Transformer models such as BERT [47] trained in a supervised or an unsupervised fashion on reaction SMILES. Those models can be fine-tuned on the task of interest to learn more specific reaction representations [48] (RXNFP). Similar to the DRFP, the size of the data-driven reaction fingerprints is independent of the number of reaction components.

3.3 Protein Representations

Proteins are large macromolecules that adopt complex 3D structures. They can be represented in string form describing the underlying amino acid sequence. Graphs at varying degrees of coarseness may also be used for structural representations that capture spatial and intramolecular relationships between structural elements, such as atoms, residues, secondary structures and chains. GAUCHE interfaces with Graphein [49], a library for pre-processing and computing graph representations of structural biological data, thereby enabling the application of graph kernel-based methods to protein structure. We provide experiments on protein fitness prediction in Appendix E.1.

4 Molecular Kernels

The choice of kernel is an important inductive bias for the properties of the function being modelled. A common choice for continuous input domains is the radial basis function kernel

$$k_{\text{RBF}}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(\frac{-||\mathbf{x} - \mathbf{x}'||_2^2}{2\ell^2}\right),$$

where σ_f^2 is the signal amplitude hyperparameter (vertical lengthscale) and ℓ is the (horizontal) lengthscale hyperparameter. However, in order to train GPS on the molecular representations covered in Section 3, bespoke kernel functions that are able to operate non-continuous input spaces are needed.

4.1 Fingerprint Kernels

Scalar Product Kernel The simplest kernel to operate on fingerprints is the scalar product or linear kernel defined for vectors $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^d$ as

$$k_{\text{Scalar Product}}(\mathbf{x}, \mathbf{x}') \coloneqq \sigma_f^2 \cdot \langle \mathbf{x}, \mathbf{x}' \rangle,$$

where σ_f is a scalar signal variance hyperparameter and $\langle \cdot, \cdot \rangle$ is the Euclidean inner product.

Tanimoto Kernel Introduced as a general similarity metric for binary attributes [50], the Tanimoto kernel was first used in chemoinformatics in conjunction with non-GP-based kernel methods [51]. It is defined for binary vectors $\mathbf{x}, \mathbf{x}' \in \{0,1\}^d$ for $d \geq 1$ as

$$k_{\text{Tanimoto}}(\mathbf{x}, \mathbf{x}') \coloneqq \sigma_f^2 \cdot \frac{\langle \mathbf{x}, \mathbf{x}' \rangle}{\|\mathbf{x}\|^2 + \|\mathbf{x}'\|^2 - \langle \mathbf{x}, \mathbf{x}' \rangle},$$

where $|| \cdot ||$ is the Euclidean norm.

In addition to the Tanimoto kernel, GAUCHE provides parallelisable and batch-GP-compatible implementations of 12 other bit and count vector kernels that are presented in Appendix G.

4.2 String Kernels

String kernels [52, 53] measure the similarity between strings by examining the degree to which their sub-strings differ. In GAUCHE, we implement the SMILES string kernel [54] which calculates an inner product between the occurrences of sub-strings, considering all contiguous sub-strings made from at most n characters (we set $n = 5$ in our experiments). Therefore, for the sub-string count featurisation $\phi : \mathcal{S} \to \mathbb{R}^p$, also known as a bag-of-characters representation [55], the SMILES string kernel between two strings $\mathcal{S}$ and $\mathcal{S}'$ is given by

$$k_{\text{String}}(\mathcal{S}, \mathcal{S}') \coloneqq \sigma_f^2 \cdot \langle \phi(\mathcal{S}), \phi(\mathcal{S}') \rangle.$$

More complicated string kernels do exist in the literature, for example, GAUCHE also provides an implementation of the subset string kernel [56] which allows non-contiguous matches. However, we found that the significant added computational cost of these methods did not provide improved performance over the more simple SMILES string kernel in the context of molecular data. Note that although named the SMILES string kernel, this kernel can also be applied to any other string representation of molecules e.g. SELFIES or protein sequences.

4.3 Graph Kernels

Graph kernels define a mapping $\phi_\lambda : \mathcal{G} \to \mathcal{H}$ from a graph domain $\mathcal{G}$ to a feature space $\mathcal{H}$, in which the inner product between a pair of graphs $g, g' \in \mathcal{G}$ serves as a similarity measure

$$k_{\text{Graph}}(g, g') \coloneqq \sigma_f^2 \cdot \langle \phi_\lambda(g), \phi_\lambda(g') \rangle_{\mathcal{H}},$$

where λ denotes kernel-specific hyperparameters. Depending on how ϕ_λ is defined, the kernel captures different substructural motifs and is characterised by different hyperparameters. The Weisfeiler-Lehman (WL) kernel [57], for instance, is given by the inner products of label count vectors over λ iterations of the Weisfeiler-Lehman algorithm [58].

To maximise the number of graph kernels available in GAUCHE we implemented the SIGP class, which enables PyTorch-based GPS to be trained on non-tensorial inputs with any kernel from the GraKel library [59] (see Appendix F for more details).

https://doi.org/10.52202/075280-3362

Table 1: Predictive accuracy and calibration of Gaussian process and probabilistic deep learning models across four different molecular property prediction benchmarks. RMSE (↓) and NLPD (↓) are reported as mean and standard error over 20 random 80/20 train/test splits. The best models up to statistical significance are highlighted in bold. Numerical issues were encountered with the WL kernel on the large lipophilicity dataset and the corresponding entries are left blank.

	Model	**Representation**	Photoswitch		ESOL		FreeSolv		Lipophilicity	
			RMSE	NLPD	RMSE	NLPD	RMSE	NLPD	RMSE	NLPD
Gaussian Processes	Tanimoto	fragprints	**20.9±0.7**	**0.22±0.03**	0.71±0.01	0.33±0.01	1.31±0.06	0.28±0.02	**0.67±0.01**	**0.71±0.01**
		fingerprints	23.4±0.8	0.33±0.03	1.01±0.01	0.71±0.01	1.93±0.09	0.58±0.03	0.76±0.01	0.85±0.01
		fragments	26.3±0.8	0.50±0.04	0.91±0.01	0.57±0.01	1.49±0.05	0.44±0.03	0.80±0.01	0.94±0.02
	Scalar Product	fragprints	22.5±0.7	**0.23±0.03**	0.88±0.01	0.53±0.01	1.27±0.02	0.25±0.02	0.77±0.01	0.92±0.01
		fingerprints	24.8±0.8	0.33±0.03	1.17±0.01	0.84±0.01	1.93±0.07	0.64±0.03	0.84±0.01	1.03±0.01
		fragments	36.6±1.0	0.80±0.03	1.15±0.01	0.82±0.01	1.63±0.03	0.54±0.02	0.97±0.01	0.88±0.10
	String	SMILES	24.8±0.7	0.30±0.04	**0.66±0.01**	**0.29±0.03**	1.31±0.01	**0.16±0.02**	**0.68±0.01**	**0.72±0.01**
	WL Kernel	graph	22.4±1.4	0.39±0.11	1.04±0.02	0.76±0.001	1.47±0.06	0.47±0.02	-	-
Neural Nets	FC-BNN	fragprints	**20.9±0.6**	1.63±0.44	0.88±0.01	1.70±0.11	1.39±0.03	1.41±0.38	0.75±0.01	3.82±0.12
		fingerprints	22.4±0.7	2.22±0.56	1.08±0.02	2.59±0.40	1.93±0.07	2.65±0.72	0.81±0.01	3.74±0.10
		fragments	25.8±0.7	0.69±0.09	1.03±0.01	1.93±0.28	1.48±0.02	0.89±0.12	0.87±0.01	5.52±0.23
	GNN-BNN	graph	28.5±1.2	1.00±0.13	0.88±0.01	1.70±0.11	**0.96±0.01**	1.01±0.02	0.73±0.02	1.14±0.01
	CNN Ensemble	SELFIES	26.4±1.0	4.34±0.55	**0.67±0.01**	2.91±0.14	1.29±0.04	2.24±0.21	0.75±0.01	2.60±0.06
	CNN DKL GP	SELFIES	25.1±0.8	0.48±0.05	0.94±0.04	0.90±0.15	1.41±0.11	0.33±0.04	0.91±0.01	1.46±0.03

5 Experiments

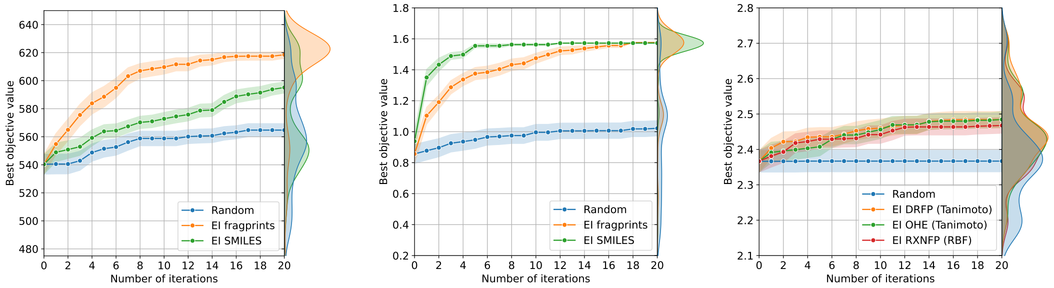
We evaluate GAUCHE on a range of regression, uncertainty quantification (UQ) and Bayesian optimisation (BO) tasks. The principle goal in conducting regression and UQ benchmarks is to gauge whether performance on these tasks may be used as a proxy for BO performance. BO is a powerful tool for automated scientific discovery but one would prefer to avoid model misspecification in the surrogate when deploying a scheme in the real world. We make use of the following datasets with experimentally determined labels:

- **Photoswitch** The labels y are the values of the E isomer $\pi - \pi^*$ transition wavelength for 392 photoswitch molecules [3].
- **ESOL** The labels y are the logarithmic aqueous solubility values for 1128 organic small molecules [60].
- **FreeSolv** The labels y are the hydration free energies for 642 molecules [41].
- **Lipophilicity** The labels y are the octanol/water distribution coefficient (log D at pH 7.4) of 4200 compounds curated from the ChEMBL database [61, 62].
- **Buchwald-Hartwig reactions** The labels y are the yields for 3955 Pd-catalysed Buchwald–Hartwig C–N cross-couplings [41].
- **Suzuki-Miyaura reactions** The labels y are the yields for 5760 Pd-catalysed Suzuki-Miyaura C-C cross-couplings [63].

5.1 Regression and Uncertainty Quantification (UQ)

Experimental setup For the regression and uncertainty quantification experiments, all datasets were randomly split into training and test sets with a ratio of 80/20. (Note that validation sets are not required for the GP models, since hyperparameters are chosen using the marginal likelihood objective on the train set). All GP models were trained using the L-BFGS-B optimiser [64] and, if not stated otherwise, the default settings in the GPyTorch and BoTorch libraries apply.

To quantify model performance, the predictive accuracy and the calibration of the predictive uncertainty estimates of the fitted models were evaluated on the held-out test set and summarised as the root-mean-square error (RMSE) and the negative log predictive density (NLPD), respectively. The mean and standard error of these metrics over 20 different random splits are reported in Table 1.



(a) Photoswitch

(b) ESOL

(c) Buchwald-Hartwig reactions

Figure 4: BO performance reporting the standard error from 50 randomly initialised trials (20 for Buchwald-Hartwig). A kernel density estimate over the trials is shown on the right axis. EI fragprints results use the Tanimoto kernel. The random search baseline is indicated in blue.

Alternative metrics to measure the quality of the predictive uncertainty estimates such as the mean standardised log loss (MSLL) and the quantile coverage error (QCE) are reported in Appendix A, while additional results for the reaction yield prediction datasets and scaffold splits are presented in Appendices B and C. We also benchmark the performance of GP models against a range of Bayesian neural network- (BNN) and deep ensemble-based methods that are detailed in Appendix D.

Results Summarising our results in Table 1, we find that Tanimoto-based GPs generally outperform Scalar Product ones in terms of RMSD and NLDP, while string kernel-based GPS often yield even better performance. We additionally note that the quality of the predictive uncertainty estimates roughly correlates with predictive accuracy in the case of GP-based models.

While deep probabilistic models attained competitive results in terms of RMSD, we found their uncertainty estimates to be consistently less reliable than those of GP-based models with discrete string kernels or shallow continuous kernels on hand-crafted features (e.g. fragprints), limiting their suitability for Bayesian optimisation and active learning. Our results suggest that for small to mid-sized molecular datasets the Tanimoto kernel combined with fragprint representations in particular is a very compelling option, with good accuracy, calibration, and runtime across all tasks.

5.2 Bayesian Optimisation

Building on these results, we employed the two best-performing kernels, namely the Tanimoto-fragprint kernel and the SMILES string kernel, to undertake Bayesian optimization BO over the photoswitch and ESOL datasets. BO is run for 20 iterations of sequential candidate selection (EI acquisition) where candidates are drawn from 95% of the dataset. The models are initialised with 5% of the dataset. In the case of the photoswitch dataset, this corresponds to just 19 molecules. The results are provided in Figure 4. In this ultra-low data setting—common to many areas of synthetic chemistry [3]—both models significantly outperform the random search baseline, highlighting the real-world use-case for such models in supporting human chemists to prioritise candidates for synthesis.

Furthermore, one may observe that BO performance is tightly coupled to regression and UQ performance. In the case of the photoswitch dataset, the better-performing Tanimoto model on regression and UQ also achieves better BO performance, while on the ESOL dataset, the string kernel performs best. Additionally, we run BO on the Buchwald-Hartwig dataset using the Tanimoto kernel for the bit-vector representations DRFP and OHE, and the RBF kernel for RXNFP. All three representations perform similarly and outperform the random search.

Finally, we also investigate the performance of GP-based models for preferential BO—a setting in which the acquisition strategy only requires rank-based preferences of candidates, as opposed to their absolute objective function values [65, 66, 67, 68]. We use a Tanimoto-fragprint kernel GP model to perform molecular Bayesian optimization on the photoswitch dataset using binary preference data alone and present the full results in Appendix E.4.

https://doi.org/10.52202/075280-3362

Table 2: An overview of existing open-source GP, BO and molecular machine learning libraries. With GAUCHE, we provide a modular, robust and easy-to-use framework that combines state-of-the-art probabilistic modelling and black-box optimization techniques with bespoke molecular representations and kernels to make them more easily accessible to the broader scientific community.

Library	Gaussian Processes	Bayesian Optimisation	Molecular Representations	Chemistry Tutorials	Graph Kernels	Bit Vector Kernels	String Kernels
GPyTorch [26]	✓	✗	✗	✗	✗	✗	✗
GPflow [69, 70]	✓	✗	✗	✗	✗	✗	✗
BoTorch [27]	✓	✓	✗	✗	✗	✗	✗
DeepChem [71]	✗	✗	✓	✓	✗	✗	✗
GraKel [59]	✗	✗	✗	✗	✓	✗	✗
FlowMO [72]	✓	✗	✓	✓	✗	✓	✓
GAUCHE (ours)	✓	✓	✓	✓	✓	✓	✓

6 Related Work

General-purpose GP and BO libraries do not cater for molecular representations. Likewise, general-purpose molecular machine learning libraries do not consider GPs and BO. Here, we review existing libraries, highlighting the niche GAUCHE fills in bridging the GP and molecular machine learning communities. The closest work to ours is FlowMO [72], which introduces a basic molecular GP library in the GPflow framework. In this project, we extend the scope of the library to a broader class of molecular representations (graphs), problem settings (BO) and applications (reaction optimisation and protein engineering). An overview of how GAUCHE fits into the existing open-source GP, BO and molecular machine learning stack is presented in Table 2.

Gaussian Process Libraries GP libraries include GPy (Python) [73], GPflow (TensorFlow) [69, 70], GPyTorch (PyTorch) [26] and GPJax (Jax) [74] while examples of recent BO libraries include BoTorch (PyTorch) [27], Dragonfly (Python) [75], HEBO (PyTorch) [76] and Trieste (Tensorflow) [77]. The aforementioned libraries do not explicitly support molecular representations. Extending them to cover molecular representations, however, requires implementations of bespoke GP kernels for bit vector, string and graph inputs together with modifications to BO schemes to consider acquisition function evaluations over a discrete set of held-out molecules, a setting commonly encountered in virtual screening [78, 79].

Molecular Machine Learning Libraries Molecular machine learning libraries include DeepChem [71], DGL-LifeSci [80] and TorchDrug [81]. DeepChem features a broad range of model implementations and tasks, while DGL-LifeSci focuses on graph neural networks. TorchDrug caters for applications including property prediction, representation learning, retrosynthesis, biomedical knowledge graph reasoning and molecule generation. However, none of the aforementioned libraries includes GP implementations. In terms of atomistic systems, DScribe [82] features, amongst other methods, the Smooth Overlap of Atomic Positions (SOAP) representation [83], which is typically used in conjunction with a GP model to learn atomistic properties. Automatic Selection And Prediction (ASAP) [84] also principally focuses on atomistic properties as well as dimensionality reduction and visualisation techniques for materials and molecules. Lastly, the Graphein library focuses on graph representations of proteins [49].

Graph Kernel Libraries Graph kernel libraries include GraKel [59], graphkit-learn [85], graphkernels [86], graph-kernels [87], pykernels (https://github.com/gmum/pykernels) and ChemoKernel [88]. The aforementioned libraries focus on CPU implementations in Python. Extending graph kernel computation to GPUs has been noted as an important direction for future research [89]. In our work, we build on the GraKel library to construct GPyTorch-based GPs that can be trained on non-tensorial, graph-structured inputs. It is worth noting that GAUCHE extends the applicability of GPU-enabled GPs to general graph-structured inputs beyond just molecules and proteins.

Molecular Bayesian Optimisation BO over molecular space can be divided into two classes. In the first class, molecules are encoded into the latent space of a variational autoencoder (VAE) [4]. BO is then performed over the continuous latent space and queried molecules are decoded back to the original space. Much work on VAE-BO has focussed on improving the synergy between the surrogate model and the VAE [90, 5, 91, 92, 93, 94, 95, 96]. One of the defining characteristics of VAE-BO is that it enables the generation of new molecular structures. In the second class of methods, BO is performed directly over the original discrete space of molecules. In this setting it is not possible to generate new structures and so a candidate set of queryable molecules is defined. The inability to generate new structures however, is not a bottleneck to molecule discovery as the principle concern is how best to explore existing candidate sets. These candidate sets are also known as molecular libraries in the virtual screening literature [97]. To date, there has been little work on BO directly over discrete molecular spaces. In [56], the authors use a string kernel GP trained on SMILES to perform BO to select from a candidate set of molecules. In [98], an optimal transport kernel GP is used for BO over molecular graphs. In [99] a surrogate based on the Nadarya-Watson estimator is defined such that the kernel density estimates are inferred using BNNs. The model is then trained on molecular descriptors. Lastly, in [100] and [101] a BNN and a sparse GP respectively are trained on fingerprint representations of molecules. In the case of the sparse GP the authors select an ArcCosine kernel. It is a longstanding aim of the GAUCHE Project to compare the efficacy of VAE-BO against vanilla BO on real-world molecule discovery tasks.

Chemical Reaction Optimisation Chemical reactions describe how reactants transform into products. Reagents (catalysts, solvents, and additives) and reaction conditions heavily impact the outcome of chemical reactions. Typically the objective is to maximise the reaction yield (the amount of product compared to the theoretical maximum) [41], in asymmetric synthesis, where the reactions could result in different enantiomers, to maximise the enantiomeric excess [102], or to minimise the E-factor, which is the ratio between waste materials and the desired product [103]. A diverse set of studies have evaluated the optimisation of chemical reactions in single and multi-objective settings [103, 104]. [105] and [106] benchmarked reaction optimisation algorithms in low-dimensional settings including reaction conditions, such as time, temperature, and concentrations. [6] suggested BO as a general tool for chemical reaction optimisation and benchmarked their approach against human experts. [107] compared the yield prediction performance of different kernels and [108] the impact of various molecular representations. In all reaction optimisation studies above, the representations of the different categories of reactants and reagents are concatenated to generate the reaction input vector, which could lead to limitations if another type of reagent is suddenly considered. Moreover, most studies concluded that simple one-hot encodings (OHE) perform at least on par with more elaborate molecular representations in the low-data regime [6, 108, 109]. In GAUCHE, we introduce reaction fingerprint kernels, based on existing reaction fingerprints [46, 45] and work independently of the number of reactant and reagent categories.

7 Limitations

One potential limitation of GAUCHE is the focus on core implementations that are likely to remain robust across as many applied problems as possible which will enable the library to have the most impact. As such, there is less of a focus on bespoke GP implementations for more targeted problems [110, 111, 112, 113]. Nonetheless, we hope that GAUCHE will function as an active development platform for such implementations.

8 Conclusion

We have introduced GAUCHE, a library for Gaussian Processes in Chemistry, with the aim of providing a user-friendly and robust library of state-of-the-art uncertainty quantification and Bayesian optimisation tools that may hopefully be deployed for screening in laboratory settings. Our aim is to maintain a lean, well-tested and up-to-date codebase and invite community-driven contributions principally as pull requests in the form of notebooks that reflect the needs and considerations that researchers come across in practice. In this fashion, we may support more advanced features without bloating the codebase and increasing maintenance requirements.

https://doi.org/10.52202/075280-3362

9 Acknowledgements

BR and PS acknowledge support from the NCCR Catalysis (grant number 180544), a National Centre of Competence in Research funded by the Swiss National Science Foundation. ARJ is funded by a Biotechnology and Biological Sciences Research Council (BBSRC) DTP studentship (BB/M011194/1). GT acknowledges the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), and the Vector Institute. FS-K is a postdoctoral fellow in the Eric and Wendy Schmidt AI in Science Postdoctoral Fellowship Program, a program by Schmidt Futures. AA-G acknowledges the generous support of Anders G. Frøseth, the Canadian Institute for Advanced Research (CIFAR), and the Canada 150 Research Chair program. LK acknowledges support from the University of Oxford's Clarendon Fund.

R-RG declares that this work was done exclusively at the University of Cambridge. Code/data was not created/accessed by Meta.

References

[1] Ying Zhang and Chen Ling. A strategy to apply machine learning to small datasets in materials science. *Npj Computational Materials*, 2018.

[2] Ryan-Rhys Griffiths, Jiachen Jiang, Douglas JK Buisson, Dan Wilkins, Luigi C Gallo, Adam Ingram, Dirk Grupe, Erin Kara, Michael L Parker, William Alston, et al. Modeling the multiwavelength variability of Mrk 335 using Gaussian processes. *The Astrophysical Journal*, 914(2):144, 2021.

[3] Ryan-Rhys Griffiths, Jake L Greenfield, Aditya R Thawani, Arian R Jamasb, Henry B Moss, Anthony Bourached, Penelope Jones, William McCorkindale, Alexander A Aldrick, Matthew J Fuchter, et al. Data-driven discovery of molecular photoswitches with multioutput Gaussian processes. *Chemical Science*, 2022.

[4] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 2018.

[5] Ryan-Rhys Griffiths and José Miguel Hernández-Lobato. Constrained Bayesian optimization for automatic chemical design using variational autoencoders. *Chemical Science*, 2020.

[6] Benjamin J Shields, Jason Stevens, Jun Li, Marvin Parasram, Farhan Damani, Jesus I Martinez Alvarado, Jacob M Janey, Ryan P Adams, and Abigail G Doyle. Bayesian reaction optimization as a tool for chemical synthesis. *Nature*, 2021.

[7] Yuanqi Du, Tianfan Fu, Jimeng Sun, and Shengchao Liu. Molgensurvey: A systematic survey in machine learning models for molecule design. *arXiv preprint arXiv:2203.14500*, 2022.

[8] Jasper Snoek, Oren Rippel, Kevin Swersky, Ryan Kiros, Nadathur Satish, Narayanan Sundaram, Mostofa Patwary, Mr Prabhat, and Ryan Adams. Scalable Bayesian optimization using deep neural networks. In *International Conference on Machine Learning*, 2015.

[9] Jost Tobias Springenberg, Aaron Klein, Stefan Falkner, and Frank Hutter. Bayesian optimization with robust Bayesian neural networks. *Advances in Neural Information Processing Systems*, 2016.

[10] Yao Zhang et al. Bayesian semi-supervised learning for uncertainty-calibrated prediction of molecular properties and active learning. *Chemical Science*, 2019.

[11] Kevin Maik Jablonka, Giriprasad Melpatti Jothiappan, Shefang Wang, Berend Smit, and Brian Yoo. Bias free multiobjective active learning for materials design and discovery. *Nature Communications*, 2021.

[12] Seongok Ryu, Yongchan Kwon, and Woo Youn Kim. A bayesian graph convolutional network for reliable prediction of molecular properties with uncertainty quantification. *Chemical Science*, 2019.

https://doi.org/10.52202/075280-3362

[13] Doyeong Hwang, Grace Lee, Hanseok Jo, Seyoul Yoon, and Seongok Ryu. A benchmark study on reliable molecular supervised learning via Bayesian learning. *arXiv preprint arXiv:2006.07021*, 2020.

[14] Gabriele Scalia, Colin A Grambow, Barbara Pernici, Yi-Pei Li, and William H Green. Evaluating scalable uncertainty estimation methods for deep learning-based molecular property prediction. *Journal of Chemical Information and Modeling*, 2020.

[15] Yoshua Bengio. *What are some Advantages of Using Gaussian Process Models vs Neural Networks?*, 2011. https://www.quora.com/What-are-some-advantages-of-using-Gaussian-Process-Models-vs-Neural-Networks.

[16] Gary Tom, Riley J Hickman, Anizet Zinzuwadia, Afshan Mohajeri, Benjamin Sanchez-Lengeling, and Alan Aspuru-Guzik. Calibration and generalizability of probabilistic models on low-data chemical datasets with dionysus. *arXiv preprint arXiv:2212.01574*, 2022.

[17] David JC MacKay, David JC Mac Kay, et al. *Information theory, inference and learning algorithms*. Cambridge University Press, 2003.

[18] Eric Anderson, Gilman D Veith, and David Weininger. SMILES, a line notation and computerized interpreter for chemical structures. *Environmental Research Laboratory*, 1987.

[19] David Weininger. SMILES, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 1988.

[20] Mario Krenn, Florian Häse, AkshatKumar Nigam, Pascal Friederich, and Alan Aspuru-Guzik. Self-referencing embedded strings (SELFIES): A 100% robust molecular string representation. *Machine Learning: Science and Technology*, 2020.

[21] David Rogers and Mathew Hahn. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 2010.

[22] Daniel Probst and Jean-Louis Reymond. A probabilistic molecular fingerprint for big data settings. *Journal of Cheminformatics*, 2018.

[23] Alice Capecchi, Daniel Probst, and Jean-Louis Reymond. One molecular fingerprint to rule them all: Drugs, biomolecules, and the metabolome. *Journal of Cheminformatics*, 2020.

[24] David K Duvenaud, Dougal Maclaurin, Jorge Iparraguirre, Rafael Bombarell, Timothy Hirzel, Alán Aspuru-Guzik, and Ryan P Adams. Convolutional networks on graphs for learning molecular fingerprints. *Advances in Neural Information Processing Systems*, 2015.

[25] Steven Kearnes, Kevin McCloskey, Marc Berndl, Vijay Pande, and Patrick Riley. Molecular graph convolutions: Moving beyond fingerprints. *Journal of Computer-Aided Molecular Design*, 2016.

[26] Jacob Gardner, Geoff Pleiss, Kilian Q Weinberger, David Bindel, and Andrew G Wilson. GPyTorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration. In *Advances in Neural Information Processing Systems*, 2018.

[27] Maximilian Balandat, Brian Karrer, Daniel Jiang, Samuel Daulton, Ben Letham, Andrew G Wilson, and Eytan Bakshy. BoTorch: A framework for efficient Monte-Carlo Bayesian optimization. *Advances in Neural Information Processing Systems*, 2020.

[28] Carl Edward Rasmussen and Zoubin Ghahramani. Occam's razor. In *Advances in Neural Information Processing Systems*, 2001.

[29] Eric Brochu, Vlad M Cora, and Nando De Freitas. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010.

[30] Bobak Shahriari, Kevin Swersky, Ziyu Wang, Ryan P Adams, and Nando de Freitas. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 2016.

https://doi.org/10.52202/075280-3362

[31] Peter I Frazier. A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

[32] Harold J Kushner. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. In *Joint Automatic Control Conference*, 1963.

[33] Jonas Močkus. On Bayesian methods for seeking the extremum. In *Optimization techniques IFIP technical conference*, 1975.

[34] A. Zhilinskas. Single-step Bayesian search method for an extremum of functions of a single variable. *Cybernetics*, 1975.

[35] Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 1998.

[36] Antoine Grosnit, Alexander I Cowen-Rivers, Rasul Tutunov, Ryan-Rhys Griffiths, Jun Wang, and Haitham Bou-Ammar. Are we forgetting about compositional optimisers in Bayesian optimisation? *arXiv preprint arXiv:2012.08240*, 2020.

[37] Philipp Hennig and Christian J Schuler. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 2012.

[38] José Miguel Hernández-Lobato, Matthew W Hoffman, and Zoubin Ghahramani. Predictive entropy search for efficient global optimization of black-box functions. *Advances in Neural Information Processing Systems*, 2014.

[39] Zi Wang and Stefanie Jegelka. Max-value entropy search for efficient Bayesian optimization. In *International Conference on Machine Learning*, 2017.

[40] Henry B Moss, David S Leslie, Javier Gonzalez, and Paul Rayson. Gibbon: General-purpose information-based Bayesian optimisation. *Journal of Machine Learning Research*, 2021.

[41] Derek T Ahneman, Jesús G Estrada, Shishi Lin, Spencer D Dreher, and Abigail G Doyle. Predicting reaction performance in C–N cross-coupling using machine learning. *Science*, 2018.

[42] Frederik Sandfort, Felix Strieth-Kalthoff, Marius Kühnemund, Christian Beecks, and Frank Glorius. A structure-based platform for predicting chemical reactivity. *Chem*, 2020.

[43] Kangway V Chuang and Michael J Keiser. Comment on "predicting reaction performance in c–n cross-coupling using machine learning". *Science*, 2018.

[44] Nadine Schneider, Daniel M Lowe, Roger A Sayle, and Gregory A Landrum. Development of a novel fingerprint for chemical reactions and its application to large-scale reaction classification and similarity. *Journal of Chemical Information and Modeling*, 2015.

[45] Daniel Probst, Philippe Schwaller, and Jean-Louis Reymond. Reaction classification and yield prediction using the differential reaction fingerprint DRFP. *Digital Discovery*, 2022.

[46] Philippe Schwaller, Daniel Probst, Alain C Vaucher, Vishnu H Nair, David Kreutter, Teodoro Laino, and Jean-Louis Reymond. Mapping the space of chemical reactions using attention-based neural networks. *Nature Machine Intelligence*, 2021.

[47] Jacob Devlin Kenton, Chang Ming-Wei, and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019.

[48] Philippe Schwaller, Alain C Vaucher, Teodoro Laino, and Jean-Louis Reymond. Prediction of chemical reaction yields using deep learning. *Machine learning: Science and Technology*, 2021.

[49] Arian R. Jamasb, Ramon Viñas, Eric J. Ma, Charlie Harris, Kexin Huang, Dominic Hall, Pietro Lió, and Tom L. Blundell. Graphein - a Python library for geometric deep learning and network analysis on protein structures and interaction networks. *bioRxiv*, 2021.

[50] John C Gower. A general coefficient of similarity and some of its properties. *Biometrics*, 1971.

[51] Liva Ralaivola, Sanjay J Swamidass, Hiroto Saigo, and Pierre Baldi. Graph kernels for chemical informatics. *Neural networks*, 2005.

[52] Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. Text classification using string kernels. *Journal of Machine Learning Research*, 2002.

[53] Nicola Cancedda, Eric Gaussier, Cyril Goutte, and Jean Michel Renders. Word sequence kernels. *Journal of Machine Learning Research*, 2003.

[54] D-S Cao, J-C Zhao, Y-N Yang, C-X Zhao, J Yan, S Liu, Q-N Hu, Q-S Xu, and Y-Z Liang. In silico toxicity prediction by support vector machine and SMILES representation-based string kernel. *SAR and QSAR in Environmental Research*, 2012.

[55] Daniel Jurafsky and James H Martin. An introduction to natural language processing, computational linguistics, and speech recognition, 2000.

[56] Henry Moss, David Leslie, Daniel Beck, Javier Gonzalez, and Paul Rayson. BOSS: Bayesian optimization over string spaces. *Advances in Neural Information Processing Systems*, 2020.

[57] Nino Shervashidze, Pascal Schweitzer, Erik Jan Van Leeuwen, Kurt Mehlhorn, and Karsten M Borgwardt. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12(9), 2011.

[58] Boris Weisfeiler and Andrei Leman. The reduction of a graph to canonical form and the algebra which appears therein. *nti, Series*, 2(9):12–16, 1968.

[59] Giannis Siglidis, Giannis Nikolentzos, Stratis Limnios, Christos Giatsidis, Konstantinos Skianis, and Michalis Vazirgiannis. Grakel: A graph kernel library in Python. *Journal of Machine Learning Research*, 2020.

[60] John S Delaney. ESOL: Estimating aqueous solubility directly from molecular structure. *Journal of Chemical Information and Computer Sciences*, 2004.

[61] Anna Gaulton, Louisa J Bellis, A Patricia Bento, Jon Chambers, Mark Davies, Anne Hersey, Yvonne Light, Shaun McGlinchey, David Michalovich, Bissan Al-Lazikani, et al. ChEMBL: A large-scale bioactivity database for drug discovery. *Nucleic Acids Research*, 2012.

[62] A Patrícia Bento, Anna Gaulton, Anne Hersey, Louisa J Bellis, Jon Chambers, Mark Davies, Felix A Krüger, Yvonne Light, Lora Mak, Shaun McGlinchey, et al. The ChEMBL bioactivity database: An update. *Nucleic Acids Research*, 2014.

[63] Damith Perera, Joseph W Tucker, Shalini Brahmbhatt, Christopher J Helal, Ashley Chong, William Farrell, Paul Richardson, and Neal W Sach. A platform for automated nanomole-scale reaction screening and micromole-scale synthesis in flow. *Science*, 2018.

[64] Dong C Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 1989.

[65] Eric Brochu, Nando Freitas, and Abhijeet Ghosh. Active preference learning with discrete choice data. *Advances in Neural Information Processing Systems*, 20, 2007.

[66] Eric Brochu. *Interactive Bayesian optimization: Learning user preferences for graphics and animation*. PhD thesis, University of British Columbia, 2010.

[67] Javier González, Zhenwen Dai, Andreas Damianou, and Neil D Lawrence. Preferential Bayesian optimization. In *International Conference on Machine Learning*, pages 1282–1291. PMLR, 2017.

[68] Zhiyuan Jerry Lin, Raul Astudillo, Peter Frazier, and Eytan Bakshy. Preference exploration for efficient Bayesian optimization with multiple outcomes. In *International Conference on Artificial Intelligence and Statistics*, pages 4235–4258. PMLR, 2022.

[69] A. G. Matthews, Mark van der Wilk, Tom Nickson, Keisuke Fujii, Alexis Boukouvalas, Pablo León-Villagrá, Zoubin Ghahramani, and James Hensman. GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research*, 2017.

[70] Mark van der Wilk, Vincent Dutordoir, ST John, Artem Artemev, Vincent Adam, and James Hensman. A framework for interdomain and multioutput Gaussian processes. *arXiv preprint arXiv:2003.01115*, 2020.

[71] Bharath Ramsundar, Peter Eastman, Patrick Walters, Vijay Pande, Karl Leswing, and Zhenqin Wu. *Deep Learning for the Life Sciences*. O'Reilly Media, 2019.

[72] Henry B. Moss and Ryan Rhys Griffiths. Gaussian process molecule property prediction with FlowMO. *arXiv preprint arXiv:2010.01118*, 2020.

[73] GPy. GPy: A Gaussian process framework in Python. `http://github.com/SheffieldML/GPy`, since 2012.

[74] Thomas Pinder and Daniel Dodd. Gpjax: A gaussian process framework in jax. *Journal of Open Source Software*, 7(75):4455, 2022.

[75] Kirthevasan Kandasamy, Karun Raju Vysyaraju, Willie Neiswanger, Biswajit Paria, Christopher R Collins, Jeff Schneider, Barnabas Poczos, and Eric P Xing. Tuning hyperparameters without grad students: Scalable and robust Bayesian optimisation with Dragonfly. *Journal of Machine Learning Research*, 2020.

[76] Alexander I Cowen-Rivers, Wenlong Lyu, Rasul Tutunov, Zhi Wang, Antoine Grosnit, Ryan Rhys Griffiths, Alexandre Max Maraval, Hao Jianye, Jun Wang, Jan Peters, et al. An empirical study of assumptions in Bayesian optimisation. *arXiv preprint arXiv:2012.03826*, 2020.

[77] Joel Berkeley, Henry B. Moss, Artem Artemev, Sergio Pascual-Diaz, Uri Granta, Hrvoje Stojic, Ivo Couckuyt, Jixiang Qing, Nasrulloh Loka, Andrei Paleyes, Sebastian W. Ober, and Victor Picheny. Trieste. *https://github.com/secondmind-labs/trieste*, 2022.

[78] Edward O Pyzer-Knapp. Using Bayesian optimization to accelerate virtual screening for the discovery of therapeutics appropriate for repurposing for COVID-19. *arXiv preprint arXiv:2005.07121*, 2020.

[79] David E Graff, Matteo Aldeghi, Joseph A Morrone, Kirk E Jordan, Edward O Pyzer-Knapp, and Connor W Coley. Self-focusing virtual screening with active design space pruning. *arXiv preprint arXiv:2205.01753*, 2022.

[80] Mufei Li, Jinjing Zhou, Jiajing Hu, Wenxuan Fan, Yangkang Zhang, Yaxin Gu, and George Karypis. DGL-LifeSci: An open-source toolkit for deep learning on graphs in life science. *ACS Omega*, 2021.

[81] Zhaocheng Zhu, Chence Shi, Zuobai Zhang, Shengchao Liu, Minghao Xu, Xinyu Yuan, Yangtian Zhang, Junkun Chen, Huiyu Cai, Jiarui Lu, Chang Ma, Runcheng Liu, Louis-Pascal Xhonneux, Meng Qu, and Jian Tang. TorchDrug: A powerful and flexible machine learning platform for drug discovery. *arXiv preprint arXiv:2202.08320*, 2022.

[82] Lauri Himanen, Marc OJ Jäger, Eiaki V Morooka, Filippo Federici Canova, Yashasvi S Ranawat, David Z Gao, Patrick Rinke, and Adam S Foster. DScribe: Library of descriptors for machine learning in materials science. *Computer Physics Communications*, 2020.

[83] Albert P Bartók, Risi Kondor, and Gábor Csányi. On representing chemical environments. *Physical Review B*, 2013.

[84] Bingqing Cheng, Ryan-Rhys Griffiths, Simon Wengert, Christian Kunkel, Tamas Stenczel, Bonan Zhu, Volker L Deringer, Noam Bernstein, Johannes T Margraf, Karsten Reuter, et al. Mapping materials and molecules. *Accounts of Chemical Research*, 2020.

[85] Linlin Jia, Benoit Gaüzère, and Paul Honeine. graphkit-learn: A Python library for graph kernels based on linear patterns. *Pattern Recognition Letters*, 2021.

[86] Mahito Sugiyama, M Elisabetta Ghisu, Felipe Llinares-López, and Karsten Borgwardt. graphkernels: R and Python packages for graph comparison. *Bioinformatics*, 2018.

https://doi.org/10.52202/075280-3362

[87] Mahito Sugiyama and Karsten Borgwardt. Halting in random walk kernels. *Advances in Neural Information Processing Systems*, 2015.

[88] Benoit Gaüzére, Luc Brun, Didier Villemin, and Myriam Brun. Graph kernels based on relevant patterns and cycle information for chemoinformatics. In *Proceedings of the 21st International Conference on Pattern Recognition*, 2012.

[89] Swarnendu Ghosh, Nibaran Das, Teresa Gonçalves, Paulo Quaresma, and Mahantapas Kundu. The journey of graph kernels through two decades. *Computer Science Review*, 2018.

[90] Ryan-Rhys Griffiths, Philippe Schwaller, and Alpha A Lee. Dataset bias in the natural sciences: A case study in chemical reaction prediction and synthesis design. *arXiv preprint arXiv:2105.02637*, 2021.

[91] Austin Tripp, Erik Daxberger, and José Miguel Hernández-Lobato. Sample-efficient optimization in the latent space of deep generative models via weighted retraining. *Advances in Neural Information Processing Systems*, 2020.

[92] Aryan Deshwal and Jana Doppa. Combining latent space and structured kernels for Bayesian optimization over combinatorial spaces. *Advances in Neural Information Processing Systems*, 2021.

[93] Antoine Grosnit, Rasul Tutunov, Alexandre Max Maraval, Ryan-Rhys Griffiths, Alexander I Cowen-Rivers, Lin Yang, Lin Zhu, Wenlong Lyu, Zhitang Chen, Jun Wang, et al. High-dimensional Bayesian optimisation with variational autoencoders and deep metric learning. *arXiv preprint arXiv:2106.03609*, 2021.

[94] Ekansh Verma and Souradip Chakraborty. Uncertainty-aware labelled augmentations for high dimensional latent space Bayesian optimization. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021.

[95] Natalie Maus, Haydn T Jones, Juston S Moore, Matt J Kusner, John Bradshaw, and Jacob R Gardner. Local latent space Bayesian optimization over structured inputs. *arXiv preprint arXiv:2201.11872*, 2022.

[96] Samuel Stanton, Wesley Maddox, Nate Gruver, Phillip Maffettone, Emily Delaney, Peyton Greenside, and Andrew Gordon Wilson. Accelerating Bayesian optimization for biological sequence design with denoising autoencoders. *arXiv preprint arXiv:2203.12742*, 2022.

[97] Edward O Pyzer-Knapp, Changwon Suh, Rafael Gómez-Bombarelli, Jorge Aguilera-Iparraguirre, and Alán Aspuru-Guzik. What is high-throughput virtual screening? A perspective from organic materials discovery. *Annual Review of Materials Research*, 2015.

[98] Ksenia Korovina, Sailun Xu, Kirthevasan Kandasamy, Willie Neiswanger, Barnabas Poczos, Jeff Schneider, and Eric Xing. ChemBO: Bayesian optimization of small organic molecules with synthesizable recommendations. In *International Conference on Artificial Intelligence and Statistics*, 2020.

[99] Florian Häse, Matteo Aldeghi, Riley J Hickman, Loïc M Roch, and Alán Aspuru-Guzik. Gryffin: An algorithm for Bayesian optimization of categorical variables informed by expert knowledge. *Applied Physics Reviews*, 2021.

[100] José Miguel Hernández-Lobato, James Requeima, Edward O Pyzer-Knapp, and Alán Aspuru-Guzik. Parallel and distributed Thompson sampling for large-scale accelerated exploration of chemical space. In *International Conference on Machine Learning*, 2017.

[101] Sattar Vakili, Henry Moss, Artem Artemev, Vincent Dutordoir, and Victor Picheny. Scalable Thompson sampling using sparse Gaussian process models. *Advances in Neural Information Processing Systems*, 2021.

[102] Andrew F Zahrt, Jeremy J Henle, Brennan T Rose, Yang Wang, William T Darrow, and Scott E Denmark. Prediction of higher-selectivity catalysts by computer-driven workflow and machine learning. *Science*, 2019.

https://doi.org/10.52202/075280-3362

[103] Artur M Schweidtmann, Adam D Clayton, Nicholas Holmes, Eric Bradford, Richard A Bourne, and Alexei A Lapkin. Machine learning meets continuous flow chemistry: Automated optimization towards the pareto front of multiple objectives. *Chemical Engineering Journal*, 2018.

[104] Pia Müller, Adam D Clayton, Jamie Manson, Samuel Riley, Oliver S May, Norman Govan, Stuart Notman, Steven V Ley, Thomas W Chamberlain, and Richard A Bourne. Automated multi-objective reaction optimisation: Which algorithm should I use? *Reaction Chemistry & Engineering*, 2022.

[105] Kobi C Felton, Jan G Rittig, and Alexei A Lapkin. Summit: Benchmarking machine learning methods for reaction optimisation. *Chemistry-Methods*, 2021.

[106] Florian Häse, Matteo Aldeghi, Riley J Hickman, Loïc M Roch, Melodie Christensen, Elena Liles, Jason E Hein, and Alán Aspuru-Guzik. Olympus: A benchmarking framework for noisy optimization and experiment planning. *Machine Learning: Science and Technology*, 2021.

[107] Alexe L Haywood, Joseph Redshaw, Magnus WD Hanson-Heine, Adam Taylor, Alex Brown, Andrew M Mason, Thomas Gärtner, and Jonathan D Hirst. Kernel methods for predicting yields of chemical reactions. *Journal of Chemical Information and Modeling*, 2021.

[108] Alexander Pomberger, Antonio Pedrina McCarthy, Ahmad Khan, Simon Sung, Connor Taylor, Matthew Gaunt, Lucy Colwell, David Walz, and Alexei Lapkin. The effect of chemical representation on active machine learning towards closed-loop optimization. *ChemRxiv*, 2022.

[109] Riley Hickman, Jurgis Ruža, Loïc Roch, Hermann Tribukait, and Alberto García-Durán. Equipping data-driven experiment planning for self-driving laboratories with semantic memory: Case studies of transfer learning in chemical reaction optimization. *ChemRxiv*, 2022.

[110] Natalie S Eyke, William H Green, and Klavs F Jensen. Iterative experimental design based on active machine learning reduces the experimental burden associated with reaction screening. *Reaction Chemistry & Engineering*, 2020.

[111] Bojana Rankovic, Ryan-Rhys Griffiths, Henry B. Moss, and Philippe Schwaller. Bayesian optimisation for additive screening and yield improvements in chemical reactions – beyond one-hot encodings. *ChemRxiv*, 2022.

[112] Ryan-Rhys Griffiths. *Gaussian Processes at Extreme Lengthscales: From Molecules to Black Holes*. PhD thesis, University of Cambridge, 2023.

[113] Henry B Moss, Sebastian W Ober, and Victor Picheny. Inducing point allocation for sparse Gaussian processes in high-throughput Bayesian optimisation. In *Artificial intelligence and statistics*, 2023.

[114] Guy W Bemis and Mark A Murcko. The properties of known drugs. 1. molecular frameworks. *Journal of medicinal chemistry*, 39(15):2887–2893, 1996.

[115] W Patrick Walters and Regina Barzilay. Applications of deep learning in molecule generation and molecular property prediction. *Accounts of Chemical Research*, 54(2):263–270, 2020.

[116] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. Strategies for pre-training graph neural networks. In *International Conference on Learning Representations*, 2019.

[117] Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. Deep kernel learning. In *Artificial intelligence and statistics*, pages 370–378. PMLR, 2016.

[118] Pavel Izmailov, Sharad Vikram, Matthew D Hoffman, and Andrew Gordon Gordon Wilson. What are bayesian neural network posteriors really like? In *International conference on machine learning*, pages 4629–4640. PMLR, 2021.

[119] Yinglu Cui, Yanchun Chen, Xinyue Liu, Saijun Dong, Yu'e Tian, Yuxin Qiao, Ruchira Mitra, Jing Han, Chunli Li, Xu Han, et al. Computational redesign of a petase for plastic biodegradation under ambient condition by the GRAPE strategy. *ACS Catalysis*, 11(3):1340–1350, 2021.

https://doi.org/10.52202/075280-3362

[120] Seongjoon Joo, Injin Cho, Hogyun Seo, Hyeoncheol Francis Son, Hye-Young Sagong, Tae Joo Shin, So Young Choi, Sang Yup Lee, and Kyung-Jin Kim. Structural insight into molecular mechanism of poly(ethylene terephthalate) degradation. *Nature Communications*, 9, 2018.

[121] Bing Liu, Lihui He, Liping Wang, Tao Li, Chang-Cheng Li, Huayi Liu, Yunzi Luo, and Rui Bao. Protein crystallography and site-direct mutagenesis analysis of the poly(ethylene terephthalate) hydrolase PETase from Ideonella sakaiensis. *ChemBioChem*, 19, 2018.

[122] Shosuke Yoshida, Kazumi Hiraga, Toshihiko Takehana, Ikuo Taniguchi, Hironao Yamaji, Yasuhito Maeda, Kiyotsuna Toyohara, Kenji Miyamoto, Yoshiharu Kimura, and Kohei Oda. A bacterium that degrades and assimilates poly (ethylene terephthalate). *Science*, 351(6278):1196–1199, 2016.

[123] Urszula Chajewska, Daphne Koller, and Ronald Parr. Making rational decisions using adaptive utility elicitation. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 363–369, 2000.

[124] Johannes Fürnkranz and Eyke Hüllermeier. Pairwise preference learning and ranking. In *Machine Learning: ECML 2003: 14th European Conference on Machine Learning, Cavtat-Dubrovnik, Croatia, September 22-26, 2003. Proceedings 14*, pages 145–156. Springer, 2003.

[125] Craig Boutilier. A POMDP formulation of preference elicitation problems. In *Eighteenth National Conference on Artificial Intelligence*, page 239–246. American Association for Artificial Intelligence, 2002.

[126] Johannes Fürnkranz and Eyke Hüllermeier. *Preference Learning*. Springer Science & Business Media, 2010.

[127] G Richard Bickerton, Gaia V Paolini, Jérémy Besnard, Sorel Muresan, and Andrew L Hopkins. Quantifying the chemical beauty of drugs. *Nature Chemistry*, 4(2):90–98, 2012.

[128] Ryan-Rhys Griffiths, Philippe Schwaller, and Alpha A Lee. Dataset bias in the natural sciences: a case study in chemical reaction prediction and synthesis design. *arXiv preprint arXiv:2105.02637*, 2021.

[129] Nathan Brown, Marco Fiscato, Marwin HS Segler, and Alain C Vaucher. GuacaMol: benchmarking models for de novo molecular design. *Journal of chemical information and modeling*, 59(3):1096–1108, 2019.

[130] Iiris Sundin, Alexey Voronov, Haoping Xiao, Kostas Papadopoulos, Esben Jannik Bjerrum, Markus Heinonen, Atanas Patronov, Samuel Kaski, and Ola Engkvist. Human-in-the-loop assisted de novo molecular design. *Journal of Cheminformatics*, 14(1):1–16, 2022.

[131] Oh-Hyeon Choung, Riccardo Vianello, Marwin Segler, Nikolaus Stiefl, and José Jiménez-Luna. Learning chemical intuition from humans in the loop. *ChemRxiv*, 2023.

[132] Wei Chu and Zoubin Ghahramani. Preference learning with Gaussian processes. In *Proceedings of the 22nd international conference on Machine learning*, pages 137–144, 2005.

[133] Raul Astudillo, Zhiyuan Jerry Lin, Eytan Bakshy, and Peter Frazier. qEUBO: A decision-theoretic acquisition function for preferential Bayesian optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 1093–1114. PMLR, 2023.

[134] Seung-Seok Choi, Sung-Hyuk Cha, Charles C Tappert, et al. A survey of binary similarity and distance measures. *Journal of systemics, cybernetics and informatics*, 8(1):43–48, 2010.

[135] Rafic Nader, Alain Bretto, Bassam Mourad, and Hassan Abbas. On the positive semi-definite property of similarity matrices. *Theoretical Computer Science*, 755:13–28, 2019.

[136] Daniel P Faith, Peter R Minchin, and Lee Belbin. Compositional dissimilarity as a robust measure of ecological distance. *Vegetatio*, 69:57–68, 1987.

[137] Stephen A Forbes. *On the local distribution of certain Illinois fishes: an essay in statistical ecology*, volume 7. Illinois State Laboratory of Natural History, 1907.

[138] Stephen A Forbes. Method of determining and measuring the associative relations of species. *Science*, 61(1585):518–524, 1925.

[139] Robert R Sokal, Peter HA Sneath, et al. Principles of numerical taxonomy. *Principles of numerical taxonomy.*, 1963.

https://doi.org/10.52202/075280-3362

Supplementary to:
GAUCHE: A Library for Gaussian Processes in Chemistry

A Uncertainty Quantification Experiments

In Table 3 we present further uncertainty quantification results using the quantile coverage error (QCE) metric. Numerical errors were encountered with the WL kernel on the large lipophilicity dataset and the corresponding entry is left blank.

Table 3: UQ benchmark. QCE values (↓) for 80/20 train/test split across 20 random trials.

GP Model		**Dataset**			
Kernel	Representation	Photoswitch	ESOL	FreeSolv	Lipophilicity
Tanimoto	fragprints	$\mathbf{0.019 \pm 0.003}$	0.023 ± 0.002	0.023 ± 0.002	0.006 ± 0.002
	fingerprints	0.023 ± 0.003	0.022 ± 0.002	0.018 ± 0.003	0.006 ± 0.001
	fragments	0.025 ± 0.005	0.012 ± 0.002	0.014 ± 0.002	0.009 ± 0.002
Scalar Product	fragprints	0.033 ± 0.006	0.010 ± 0.002	0.017 ± 0.003	0.010 ± 0.001
	fingerprints	0.036 ± 0.006	0.014 ± 0.002	0.016 ± 0.002	0.009 ± 0.001
	fragments	0.027 ± 0.004	0.012 ± 0.003	0.021 ± 0.003	0.010 ± 0.001
String	SMILES	0.024 ± 0.003	0.016 ± 0.002	0.019 ± 0.003	0.005 ± 0.001
WL Kernel (GraKel)	graph	0.025 ± 0.007	0.011 ± 0.004	0.019 ± 0.009	-

B Chemical Reaction Yield Prediction Experiments

Further regression and uncertainty quantification experiments are presented in Table 4. The differential reaction fingerprint in conjunction with the Tanimoto kernel is the best-performing reaction representation.

Table 4: Chemical reaction regression benchmark. 80/20 train/test split across 20 random trials.

GP Model		**Buchwald-Hartwig**		
Kernel	Representation	RMSE ↓	R^2 score ↑	QCE ↓
Tanimoto	OHE	7.94 ± 0.05	0.91 ± 0.001	0.011 ± 0.001
	DRFP	$\mathbf{6.48 \pm 0.45}$	$\mathbf{0.94 \pm 0.015}$	0.027 ± 0.002
Scalar Product	OHE	15.23 ± 0.052	0.69 ± 0.002	0.008 ± 0.001
	DRFP	14.63 ± 0.050	0.71 ± 0.002	0.010 ± 0.001
RBF	RXNFP	10.79 ± 0.049	0.84 ± 0.001	0.024 ± 0.001
		Suzuki-Miyaura		
Tanimoto	OHE	11.18 ± 0.036	0.83 ± 0.001	0.007 ± 0.001
	DRFP	11.46 ± 0.038	0.83 ± 0.001	0.019 ± 0.000
Scalar Product	OHE	19.91 ± 0.042	0.47 ± 0.003	0.012 ± 0.001
	DRFP	19.66 ± 0.042	0.52 ± 0.003	0.014 ± 0.001
RBF	RXNFP	13.83 ± 0.048	0.75 ± 0.002	0.007 ± 0.001

C Scaffold Split Experiments

To investigate how GP models behave on more challenging train/test splits, we have re-run parts of our experimental evaluation with 80-20 Bemis-Murcko [114] scaffold splits instead of random splits. As only the lipophilicity dataset exhibits sufficient scaffold diversity to perform this analysis (the skewness of the scaffold distribution in the others makes an 80-20 split impossible), the results in Table 5 focus on the predictive accuracy (RMSE) and calibration (NLPD) of GP models in this setting.

While this more challenging evaluation setup leads to slightly higher RMSEs and NLPDs, we note that one can observe the same trends as with random splits: Tanimoto-based GPs generally outperform Scalar Product ones, while string kernel-based GPs are better than both.

Table 5: Regression and uncertainty quantification experiments on scaffold splits.

Kernel	Representation	RMSE (↓)	NLPD (↓)
Tanimoto	Fragprints	$0.86_{\pm 0.01}$	$\mathbf{1.02_{\pm 0.04}}$
	Fingerprints	$0.88_{\pm 0.01}$	$1.12_{\pm 0.04}$
	Fragments	$0.89_{\pm 0.01}$	$2.10_{\pm 0.13}$
Scalar Product	Fragprints	$0.89_{\pm 0.01}$	$1.75_{\pm 0.08}$
	Fingerprints	$0.95_{\pm 0.01}$	$1.99_{\pm 0.09}$
	Fragments	$1.00_{\pm 0.01}$	-
String	SMILES	$\mathbf{0.82_{\pm 0.01}}$	$1.08_{\pm 0.04}$

D Deep Probabilistic Models

There is a growing body of work applying deep learning to molecular property prediction [115]. Therefore in addition to evaluating GPs with varying shallow kernel functions, we repeat the regression experiments with a range of deep Bayesian models, varying both the network architecture and the Bayesian inference procedure. We evaluate the following models:

- **FC-BNN + VI** is a fully connected neural network with a single variational inference (VI) Bayesian layer with 100 nodes, followed by the rectified linear unit (ReLU) activation, and a final output layer. Trained with early stopping.
- **GNN-BNN + VI** utilises the same network and graph features as used for the graph embeddings [116] followed by a final VI Bayesian layer. Trained with early stopping.
- **CNN DKL GP** is the same approach and architecture used by [96] to predict molecular properties for Bayesian optimisation. Using SELFIES representations, a 1D CNN encoder is shared and trained jointly through a generative masked language model (MLM) head [47] and a discriminative deep kernel GP head [117].
- **CNN Ensemble** is a deep ensemble of 1D CNN networks, also implemented by [96], where each ensemble component uses the SELFIES molecule representation and is trained independently to minimize the MSE loss. Deep ensembles have been shown to provide high-fidelity approximations of Bayesian model averages relative to alternative approaches such as Laplace approximation or VI [118].

E Further Experiments

E.1 Protein Fitness Prediction

We consider the task of protein fitness prediction where the fitness function (target label) takes the form of the melting point in degrees Celsius. We collate a dataset of 151 PETase protein sequences from values reported in [119, 120, 121]. PETases, recently discovered in 2016 [122], are a class of esterase enzymes which, via hydrolysis, catalyse the breakdown of polyethylene terephthalate (PET) plastic to monomeric mono-2-hydroxyethyl terephthalate (MHET). While PET plastics can require hundreds of years to degrade naturally, PETases are capable of degrading PET in a matter of days but their melting point is a key property of interest for deployed applications. Each sequence consists of an amino acid chain of length 290. We use the 'bag of amino acids' representation with a max n-gram value of 5 which gives rise to a count vector where each component represents the number of a given n-gram contained in the sequence. A max n-gram value of 5 means that all n-grams up to and including length 5 are included in the featurisation. We subsequently train a Tanimoto kernel GP

https://doi.org/10.52202/075280-3362

on the featurisation. The results are provided in Table 6 and indicate that the GP model obtains low generalisation error on the melting point prediction task.

Table 6: PETase melting point prediction experiment. 80/20 train/test split across 20 random trials.

GP Model		**PETase Test Set**		
Kernel	Representation	RMSE ↓	MAE ↓	R^2 score ↑
Tanimoto	Bag of Amino Acids	3.68 ± 0.13	2.54 ± 0.10	0.81 ± 0.02

E.2 Molecular Preference Learning

In many optimisation problems, while it may be challenging to define a mathematical utility function, human feedback on pairwise comparisons can be leveraged to learn a latent utility function. In the machine learning literature, this observation has inspired work on preference learning [123, 124, 125, 126] where the goal is to learn a utility function $g(\mathbf{x})$ using binary preference data $r(\mathbf{x}_1, \mathbf{x}_2)$ obtained from human feedback on inputs $\mathbf{x}$. In the case of molecular design, there are many situations in which it is difficult to articulate a utility function [127, 128, 129] which has motivated recent work on human-in-the-loop preference learning [130, 131]. In this setting, pairwise preferences $r(\mathbf{x}_1, \mathbf{x}_2)$ are collected from a human chemist by presenting with many choices over pairs of molecules. In this section, we highlight a use-case for GAUCHE in GP-based preference learning [132]. Using the photoswitch dataset as a case study, we simulate noiseless pairwise comparisons from a human chemist on the E isomer $\pi - \pi^*$ transition wavelength (latent utility function). We train a Tanimoto kernel GP on the fragprints representation of the molecules, using a probit likelihood and the Laplace approximation [132]. The results are provided in Table 7. We report the Kendall-Tau rank correlation as the preference model learns ordinal rankings in place of absolute values of the transition wavelength. The results indicate that it is possible to learn accurate models of molecular properties through binary feedback alone.

Table 7: Photoswitch preference learning experiment. 80/20 train/test split across 20 random trials with 5000 training set pairwise comparisons. Kendall-Tau rank correlation (K-T) values reported.

GP Model		**Photoswitch Dataset**	
Kernel	Representation	K-T Train ↑	K-T Test ↑
Tanimoto	Fragprints	0.95 ± 0.001	0.79 ± 0.01

E.3 Ablation over the ECFP Radius Parameter

In addition to the experimental results presented in Section 5, we have performed an ablation study over the radius parameter of extended-connectivity fingerprints (ECFPs). Specifically, we trained Tanimoto-kernel GPs on the Photoswitch dataset using a series of five increasing radius parameters. In Table 8, we report the mean and standard error of the RMSE and NLPD over 50 different 80-20 train-test-splits.

Table 8: RMSE and NLPD of Tanimoto-kernel GPs trained on the Photoswitch dataset using a series of five increasing ECFP radius parameters.

Fingerprint	RMSE (↓)	NLPD (↓)
ECFP4	**22.65±0.55**	**0.41±0.05**
ECFP6	23.50±0.55	0.47±0.05
ECFP8	24.43±0.54	0.52±0.05
ECFP10	25.17±0.54	0.56±0.04
ECFP12	25.70±0.53	0.58±0.04

Intriguingly, these results show a strong negative correlation between the fingerprint radius and predictive performance. We hypothesize that this is caused by the fact that an expanding feature

https://doi.org/10.52202/075280-3362

space leads to lower and less informative Tanimoto similarity scores, making it more difficult to train generalisable models.

E.4 Preferential Bayesian Optimisation

In many Bayesian optimisation problems, the acquisition strategy only requires rank-based preferences of candidates (as opposed to their absolute objective function values) to operate. Such an observation has motivated the field of Preferential Bayesian optimisation (PBO) [65, 66, 67, 68] where Bayesian optimisation is performed using binary preference data in lieu of the absolute values of the objective function. In Figure 5 we present the results of a Bayesian optimisation loop on the photoswitch dataset featuring a Tanimoto kernel GP surrogate with probit likelihood and the Laplace approximation [132], combined with the EUBO $-\ \zeta$ acquisition function from [68, 133]. The results indicate that it is possible to perform molecular Bayesian optimization using binary preference data alone.

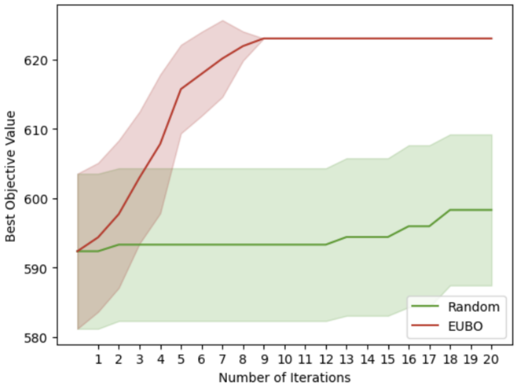


Figure 5: PBO results on the Photoswitch dataset using the EUBO $-\ \zeta$ acquisition function. 20 random trials reporting the 1.96 standard error bands. The surrogate is initialized with 98 molecules with 100 pairwise comparisons. 294 molecules are retained in a heldout set to be selected as part of the BO loop. A single molecule is selected on each iteration and an additional 100 pairwise comparisons over the augmented train set are recorded.

https://doi.org/10.52202/075280-3362

F Coding Kernels in GAUCHE

We provide an example of the class definition for the Tanimoto kernel in GAUCHE below

```
class TanimotoGP(ExactGP):
    def __init__(self, train_x, train_y, likelihood):
        super(TanimotoGP, self).__init__(train_x,
                                         train_y,
                                         likelihood)
        self.mean_module = ConstantMean()
        # We use the Tanimoto kernel to work with
        # molecular fingerprint representations
        self.covar_module = ScaleKernel(TanimotoKernel())

    def forward(self, x):
        mean_x = self.mean_module(x)
        covar_x = self.covar_module(x)
        return MultivariateNormal(mean_x, covar_x)
```

and an example definition of a black box kernel (where gradients with respect to hyperparameters and input labels are not required).

```
class WLKernel(gauche.Kernel):
    def __init__(self):
        super().__init__()
        self.kernel = grakel.kernels.WeisfeilerLehman()

    @lru_cache(maxsize=3)
    def kern(self, X):
        return tensor(self.kernel.fit_transform(X.data))

class GraphGP(gauche.SIGP):
    def __init__(self, train_x, train_y, likelihood):
        super().__init__(train_x, train_y, likelihood)
        self.mean = ConstantMean()
        self.covariance = WLKernel()

    def forward(self, X):
        # X is a gauche.Inputs instance, with X.data
        # holding a list of grakel.Graph instances.
        mean = self.mean(zeros(len(X.data), 1))
        covariance = self.covariance(X)
        return MultivariateNormal(mean, covariance)
```

Importantly, GAUCHE inherits all the facilities of GPyTorch and GraKel allowing a broad range of of models to be defined on molecular inputs such as deep GPs, multioutput GPs and heteroscedastic GPs.

G Additional Bit and Count Vector Kernels

GAUCHE provides parallelisable and batch-GP-compatible implementations of the following similarity measures from [134] that are provably symmetric and positive semi-definite [135]. All kernels are defined for binary vectors $\mathbf{x}, \mathbf{x}' \in \{0, 1\}^d$ for $d \geq 1$, where n represents the number of common zeros between $\mathbf{x}$ and $\mathbf{x}'$ and $|| \cdot ||$ is the Euclidean norm.

Braun-Blanquet Kernel

$$k_{\text{Braun-Blanquet}}(\mathbf{x}, \mathbf{x}') := \sigma_f^2 \cdot \frac{\langle \mathbf{x}, \mathbf{x}' \rangle}{\max(\|\mathbf{x}\|, \|\mathbf{x}'\|)}.$$

Dice Kernel

$$k_{\text{Dice}}(\mathbf{x}, \mathbf{x}') := \sigma_f^2 \cdot \frac{2 \cdot \langle \mathbf{x}, \mathbf{x}' \rangle}{\|\mathbf{x}\| + \|\mathbf{x}'\|}.$$

Faith Kernel [136]

$$k_{\text{Faith}}(\mathbf{x}, \mathbf{x}') := \sigma_f^2 \cdot \frac{2 \cdot \langle \mathbf{x}, \mathbf{x}' \rangle + n}{2d}.$$

Forbes Kernel [137, 138]

$$k_{\text{Forbes}}(\mathbf{x}, \mathbf{x}') := \sigma_f^2 \cdot \frac{d \cdot \langle \mathbf{x}, \mathbf{x}' \rangle}{\|\mathbf{x}\| + \|\mathbf{x}'\|}.$$

Intersection Kernel

$$k_{\text{Intersection}}(\mathbf{x}, \mathbf{x}') := \sigma_f^2 \cdot (\langle \mathbf{x}, \mathbf{x} \rangle + \langle \mathbf{x}', \mathbf{x}' \rangle).$$

MinMax Kernel

$$k_{\text{MinMax}}(\mathbf{x}, \mathbf{x}') := \sigma_f^2 \cdot \frac{\|\mathbf{x}\| + \|\mathbf{x}'\| - \|\mathbf{x} - \mathbf{x}'\|}{\|\mathbf{x}\| + \|\mathbf{x}'\| + \|\mathbf{x} - \mathbf{x}'\|}.$$

Otsuka Kernel

$$k_{\text{Otsuka}}(\mathbf{x}, \mathbf{x}') := \sigma_f^2 \cdot \frac{\langle \mathbf{x}, \mathbf{x}' \rangle}{\sqrt{\|\mathbf{x}\| + \|\mathbf{x}'\|}}.$$

Rand Kernel

$$k_{\text{Rand}}(\mathbf{x}, \mathbf{x}') := \sigma_f^2 \cdot \frac{\langle \mathbf{x}, \mathbf{x}' \rangle + n}{d}.$$

Rogers-Tanimoto Kernel

$$k_{\text{Rogers-Tanimoto}}(\mathbf{x}, \mathbf{x}') := \sigma_f^2 \cdot \frac{\langle \mathbf{x}, \mathbf{x}' \rangle + n}{2 \cdot \|\mathbf{x}\| + 2 \cdot \|\mathbf{x}'\| - 3 \cdot \langle \mathbf{x}, \mathbf{x}' \rangle + n}.$$

Russell-Rao Kernel

$$k_{\text{Russell-Rao}}(\mathbf{x}, \mathbf{x}') := \sigma_f^2 \cdot \frac{\langle \mathbf{x}, \mathbf{x}' \rangle}{d}.$$

Sorgenfrei Kernel

$$k_{\text{Sorgenfrei}}(\mathbf{x}, \mathbf{x}') := \sigma_f^2 \cdot \frac{\langle \mathbf{x}, \mathbf{x}' \rangle^2}{\|\mathbf{x}\| + \|\mathbf{x}'\|}.$$

Sokal-Sneath Kernel [139]

$$k_{\text{Sokal-Sneath}}(\mathbf{x}, \mathbf{x}') := \sigma_f^2 \cdot \frac{\langle \mathbf{x}, \mathbf{x}' \rangle}{2 \cdot \|\mathbf{x}\| + 2 \cdot \|\mathbf{x}'\| - 3 \cdot \langle \mathbf{x}, \mathbf{x}' \rangle}.$$

https://doi.org/10.52202/075280-3362