GAUCHE: A Library for Gaussian Processes in Chemistry

Anonymous Authors[1]

Abstract

We introduce GAUCHE, a 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 chemical representations however is nontrivial, necessitating kernels defined over structured inputs such as graphs, strings and bit vectors. By defining such kernels in GAUCHE, we seek to open the door to powerful tools for uncertainty quantification and Bayesian optimisation in chemistry. Motivated by scenarios frequently encountered in experimental chemistry, we showcase applications for GAUCHE in molecule discovery, chemical reaction optimisation and protein engineering.

1. Introduction

Early-stage scientific discovery is typically characterised by the small data regime due to the limited availability of high-quality experimental data (Zhang & Ling, 2018; Thawani et al., 2020). Much of the novelty of discovery relies on the fact that there is much knowledge to gain in the small data regime. By contrast, in the big data regime, discovery offers diminishing returns as much of the knowledge about the space of interest has already been acquired. As such, machine learning methodologies that facilitate search in small data regimes such as Bayesian optimisation (BO) (Gómez-Bombarelli et al., 2018; Griffiths & Hernández-Lobato, 2020; Shields et al., 2021) and active learning (AL) (Zhang et al., 2019; Jablonka et al., 2021) have great potential to expedite the rate at which performant molecules, molecular materials, chemical reactions and proteins are discovered.

To date in molecular machine learning, Bayesian neural networks (BNNs) have been the surrogate of choice to produce the uncertainty estimates that underpin BO and AL (Ryu et al., 2019; Zhang et al., 2019; Hwang et al., 2020; Scalia et al., 2020). For small datasets, however, deep neural networks are often not the model of choice. Notably, certain deep learning experts have voiced a preference for Gaussian processes (GPs) in the small data regime (Bengio, 2011). Furthermore, for BO, GPs possess particularly advantageous properties; 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 (MacKay et al., 2003),

> "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."

The iterative model refitting required in BO makes it a prime example of such an automated task. Although BNN surrogates have been trialled for BO (Snoek et al., 2015; Springenberg et al., 2016), GPs remain the model of choice as evidenced by the results of the recent NeurIPS Black-Box Optimisation Competition (Turner et al., 2021).

Training GPs on molecular inputs is non-trivial. Canonical applications of GPs assume continuous input spaces of low and fixed dimensionality. The most popular molecular input representations are SMILES/SELFIES strings (Anderson et al., 1987; Weininger, 1988; Krenn et al., 2020), fingerprints (Rogers & Hahn, 2010; Probst & Reymond, 2018; Capecchi et al., 2020) and graphs (Duvenaud et al., 2015; Kearnes et al., 2016). Each of these input representations poses problems for GPs. SMILES strings have variable lengths, fingerprints are high-dimensional and sparse bit vectors, while graphs are also a form of non-continuous input. To construct a GP framework over molecules, GAUCHE provides GPU-based implementations of kernels that operate on molecular inputs, including string, fingerprint and graph kernels. Furthermore, GAUCHE includes support for protein and chemical reaction representations and interfaces with the GPyTorch (Gardner et al., 2018) and BoTorch (Balandat et al., 2020) libraries to facilitate usage for advanced probabilistic modelling and BO.

Concretely, our contributions may be summarised as:

1. We propose a GP framework for molecules, chemical reactions and proteins.
2. We provide an open-source, GPU-enabled library building on GPyTorch (Gardner et al., 2018), BoTorch (Balandat et al., 2020) and RDKit (Landrum, 2013).

3. We extend the use of black box graph kernels (from GraKel, Siglidis et al. (2020)) to GP regression via a GPyTorch interface, along with a limited set of graph kernels implemented in native GPyTorch to enable optimisation of the graph kernel hyperparameters under the marginal likelihood.

4. We conduct benchmark experiments evaluating the utility of the GP framework on regression, uncertainty quantification and BO tasks.

GAUCHE is made available at the following anonymised link and includes tutorials to guide users through the tasks considered in this paper.

2. Background

We summarise the background on Gaussian processes, Bayesian optimisation, common molecular representations and how GP kernels may be extended to cater for them.

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 $[K]_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$. θ represents the set of kernel hyperparameters. The GP specifies the full distribution over the function f to be modelled as

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

Prediction: At test locations $\mathbf{X}_*$ 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 $\mathrm{cov}(\mathbf{f}_*) = K(\mathbf{X}_*, \mathbf{X}_*) - K(\mathbf{X}_*, \mathbf{X})[K(\mathbf{X}, \mathbf{X}) + \sigma_y^2 I]^{-1} K(\mathbf{X}, \mathbf{X}_*)$.

Kernel Functions: The choice of kernel function 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

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

where σ_f^2 is the signal amplitude hyperparameter (vertical lengthscale) and ℓ is the (horizontal) lengthscale hyperparameter. The symbol θ, introduced previously, is used to represent the set of kernel hyperparameters. For molecules, bespoke kernel functions will need to be defined for structured input spaces.

GP Training: Hyperparameters for Gaussian processes 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)

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

$I\sigma_y^2$ represents the variance of i.i.d. Gaussian noise on the observations $\mathbf{y}$. The NLML embodies Occam's razor for Bayesian model selection (Rasmussen & Ghahramani, 2001) in favouring models that fit the data without being overly complex.

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 example setting the structured input domain consists of a set of molecular representations (graphs, strings, bit vectors) and the expensive black-box function is a property of interest for a given molecule that we wish to optimise. Bayesian optimisation (BO) (Kushner, 1963; Močkus, 1975; Zhilinskas, 1975; Jones et al., 1998; Brochu et al., 2010; Grosnit et al., 2020) 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. This procedure involves the exploration/exploitation tradeoff in the sense that exploiting knowledge about the function to propose promising locations competes with the desire to learn more about the function in unobserved locations.

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) (Močkus, 1975; Jones et al., 1998) and entropy search (Hennig & Schuler, 2012; Hernández-Lobato et al., 2014; Wang & Jegelka, 2017; Moss et al., 2021). After the

acquisition function proposes an input location, the black-box is evaluated at that location, the surrogate model is retrained and the process repeats *ad libitum* until a solution is obtained. Systematic reviews of the BO literature include (Brochu et al., 2010; Shahriari et al., 2016; Frazier, 2018)

2.3. Molecular Representations

We review here the three main categories of molecular representations before describing the kernels that operate on them in section 3.

Graphs: Molecules may be represented as an undirected, labeled 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 $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ represent covalent bonds between these atoms. Additional information may be incorporated in the form of vertex and edge labels $\mathcal{L} : \mathcal{V} \times \mathcal{E} \to \Sigma_V \times \Sigma_E$, with common label spaces including attributes such as atom types (i.e. hydrogen, carbon) as vertex labels and bond orders (i.e. single, double) as edge labels.

Fingerprints: Molecular fingerprints were first introduced for chemical database substructure searching (Christie et al., 1993) but were later repurposed for similarity searching (Johnson & Maggiora, 1990), clustering (McGregor & Pallai, 1997) and classification (Breiman et al., 2017). Extended Connectivity FingerPrints (ECFP) (Rogers & Hahn, 2010) were introduced as part of the Pipeline project (Hassan et al., 2006) with the explicit goal of capturing features relevant for molecular property prediction (Xia et al., 2004). ECFP fingerprints operate by assigning initial numeric identifiers to each atom in a molecule. These identifiers are subsequently updated in an iterative fashion based on the identifiers of their neighbours. The number of iterations corresponds to half the *diameter* of the fingerprint and the naming convention reflects this. For example, ECFP6 fingerprints have a diameter of 6, meaning that 3 iterations of atom identifier reassignment are performed. Each level of iteration appends substructural features of increasing non-locality to an array and the array is then hashed to a bit vector reflecting the presence of absence of those substructures in the molecule.

For property prediction applications a radius of 3 or 4 is recommended. We use a radius of 3 for all experiments in the paper. Additionally we make use of fragment descriptors which are count vectors, each component of which indicates the number of a certain functional group present in a molecule. For example row 1 of the count vector could be an integer representing the number of aliphatic hydroxl groups present in the molecule. We make use of both fingerprint and fragment features computed using RDKit (Landrum, 2013) as well as the concatenation of the fingerprint and fragment feature vectors, a representation termed fragprints (Thawani et al., 2020) which has shown strong empirical performance. Example representations $\mathbf{x_f}$ for fingerprints and $\mathbf{x_{fr}}$ for fragments are given as

$$\mathbf{x_f} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 1 \end{bmatrix}, \quad \mathbf{x_{fr}} = \begin{bmatrix} 3 \\ 0 \\ \vdots \\ 2 \end{bmatrix}.$$

Strings: The Simplified Molecular-Input Line-Entry System (SMILES) is a text-based representation of molecules (Anderson et al., 1987; Weininger, 1988), examples of which are given in Figure 1. Self-Referencing Embedded Strings (SELFIES) (Krenn et al., 2020) is an alternative string representation to SMILES such that a bijective mapping exists between a SELFIES string and a molecule.

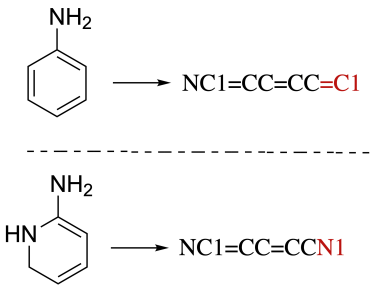


Figure 1: SMILES strings for structurally similar molecules. Similarity is encoded in the string through common contiguous subsequences (black). Local differences are highlighted in red.

2.4. Reaction Representations

Chemical reactions consist of multiple reactants and reagents that transform into one or more products. The reactants and reagents can often be categorised into different types. Taking the high-throughput experiments by (Ahneman et al., 2018) on Buchwald-Hartwig reactions as an example, 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 reactant and reagent categories is constant, the molecular representations discussed above may be used to encode the selected reactants and reagents, and the vectors for the individual reaction components can be concatenated to build the reaction representation (Ahneman et al., 2018; Sandfort et al., 2020). 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 (Chuang & Keiser, 2018).

Differential reaction fingerprints: Inspired by the hand-engineered difference reaction fingerprints by Schneider et al. (2015), Probst et al. (2022) 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: Schwaller et al. (2021a) described data-driven reaction fingerprints using Transformer models (e.g. BERT (Devlin et al., 2018)) 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 (Schwaller et al., 2021b) (RXNFP). Similar to the DRFP, the size of the data-driven reaction fingerprints is independent of the number of reaction components.

2.5. Protein Representations

Proteins are large macromolecules that adopt complex 3D structures. Proteins can be represented in string form describing the underlying amino acid sequence. Graphs at varying degrees of coarseness may 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 (Jamasb et al., 2021), 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.

3. Molecular Kernels

Here we introduce examples of the classes of GAUCHE kernel designed to operate on the molecular representations introduced in section 2.

3.1. 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

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

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

Tanimoto Kernel: First introduced as a general similarity metric for binary attributes (Gower, 1971), the Tanimoto kernel was first used in chemoinformatics in conjunction with non-GP-based kernel methods (Ralaivola et al., 2005). It is defined for binary vectors $\mathbf{x}, \mathbf{x}' \in \{0, 1\}^d$ for $d \geq 1$ as

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

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

3.2. String Kernels

String kernels (Lodhi et al., 2002; Cancedda et al., 2003) measure the similarity between strings by examining the degree at which their sub-strings differ. In GAUCHE, we implement the SMILES string kernel (Cao et al., 2012) 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 (Jurafsky & Martin, 2000)), the SMILES string kernel between two strings $\mathcal{S}$ and $\mathcal{S}'$ is given by

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

Although more complicated string kernels do exist in the literature, for example those that allow non-contiguous matches (Moss et al., 2020a), we found that the significant extra 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.

3.3. Graph Kernels

Graph Kernels: Graph kernel methods $\phi_\lambda : \mathcal{G} \to \mathcal{H}$ map elements from a graph domain $\mathcal{G}$ to a reproducing kernel Hilbert space (RKHS) $\mathcal{H}$, in which an inner product between a pair of graphs $g, g' \in \mathcal{G}$ is derived as a measure of similarity

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

where λ denotes kernel-specific hyperarameters and σ^2 is a scale factor. Depending on how ϕ_λ is defined (Nikolentzos

et al., 2021), the kernel considers different substructural motifs and is characterised by different hyperparameters.

Frequently-employed approaches include the random walk kernel (Vishwanathan et al., 2010), given by a geometric series over the count of matching random walks of increasing length with coefficient λ, and the Weisfeiler-Lehman kernel (Shervashidze et al., 2011), given by the inner products of label count vectors over λ iterations of the Weisfeiler-Lehman algorithm.

Graph Embedding: Pretrained graph neural networks (GNNs) (Hu et al., 2019) may also be used to embed molecular graphs in a vector space. Since the GNN is trained on a large amount of data, the representation it produces has the potential to be a more expressive method to encode a molecule (Note: this assumes access to a large pool of in-domain data). Given a vector representation from a pretrained GNN model, we may apply any GP kernel for continuous input spaces, such as the RBF kernel.

3.4. 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.

4. Experiments

We evaluate GAUCHE (available at anonymized link) on regression, uncertainty quantification (UQ) and BO. 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:

The Photoswitch Dataset: (Thawani et al., 2020): The labels, y are the experimentally-determined values of the E isomer $\pi - \pi^*$ transition wavelength for 392 photoswitch molecules.

ESOL: (Delaney, 2004): The labels y are the experimentally-determined logarithmic aqueous solubility values for 1128 organic small molecules.

FreeSolv: (Mobley & Guthrie, 2014): The labels y are the experimentally-determined hydration free energies for 642 molecules.

Lipophilicity: The labels y are the experimentally-determined octanol/water distribution coefficient (log D at pH 7.4) of 4200 compounds curated from the ChEMBL database (Gaulton et al., 2012; Bento et al., 2014).

Buchwald-Hartwig reactions: (Ahneman et al., 2018): The labels y are the experimentally-determined yields for 3955 Pd-catalysed Buchwald–Hartwig C–N cross-couplings.

Suzuki-Miyaura reactions: (Perera et al., 2018): The labels y are the experimentally-determined yields for 5760 Pd-catalysed Suzuki-Miyaura C-C cross-couplings.

4.1. Regression

The regression results for molecular property prediction are reported in Table 1 and for reaction yield prediction in Table A1 of Appendix A. The datasets are split in a train/test ratio of 80/20 (note that validation sets are not required for the GP models since training uses the marginal likelihood objective). Errorbars represent the standard error across 20 random initialisations. All GP models are trained using the L-BFGS-B optimiser (Liu & Nocedal,

1989). If not mentioned, default settings in the GPyTorch and BoTorch libraries apply. For the SELFIES representation, some molecules could not be featurised and corresponding entries are left blank. The results of Table A1 indicate that the best choice of representation (and hence the choice of kernel) is task-dependent.

4.2. Uncertainty Quantification (UQ)

To quantify the quality of the uncertainty estimates we use three metrics, the negative log predictive density (NLPD), the mean standardised log loss (MSLL) and the quantile coverage error (QCE). We provide the NLPD results in Table 2 and defer the MSLL and QCE results to Appendix B. One trend to note is that uncertainty estimate quality is roughly correlated with regression performance.

4.3. Bayesian Optimisation

We take forward two of the best-performing kernels, the Tanimoto-fragprint kernel and the bag of SMILES kernel to undertake BO over the photoswitch and ESOL datasets. Random search is used as a baseline. BO is run for 20 iterations of sequential candidate selection (EI acquisition) where candidates are drawn from 95% of the dataset. The results are provided in Figure 2. The models are initialised with 5% of the dataset. In the case of the photoswitch dataset this corresponds to just 19 molecules. In this ultra-low data setting, common to many areas of synthetic chemistry (Thawani et al., 2020) both models outperform random search, highlighting the real-world use-case for such models in supporting human chemists 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 relatively better BO performance. Additionally, we report results on the Buchwald-Hartwig reaction dataset.

5. Related Work

General-purpose GP and Bayesian optimisation libraries do not specifically cater for molecular representations. Likewise, general-purpose molecular machine learning libraries do not specifically consider GPs and Bayesian optimisation. 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 (Moss & Griffiths, 2020), which introduces a molecular GP library in the GPflow framework. It is on this project which we build, extending the scope of the library to a broader class of molecular representations (graphs), problem settings (Bayesian optimisation) and applications (reaction optimisation and protein engineering).

Gaussian Process Libraries: GP libraries include GPy (Python) (GPy, since 2012), GPflow (TensorFlow) (Matthews et al., 2017; van der Wilk et al., 2020) and GPyTorch (PyTorch) (Gardner et al., 2018) while examples of recent Bayesian optimisation libraries include BoTorch (PyTorch) (Balandat et al., 2020), Dragonfly (Python) (Kandasamy et al., 2020) and HEBO (PyTorch) (Cowen-Rivers et al., 2020). The aforementioned libraries do not explicitly support molecular representations. Extension to cover molecular representations however is nontrivial, requiring implementations of bespoke GP kernels for bit vector, string and graph inputs together with modifications to Bayesian optimisation schemes to consider acquisition function evaluations over a discrete set of heldout molecules, a setting commonly encountered in virtual screening campaigns (Pyzer-Knapp, 2020; Graff et al., 2022).

Molecular Machine Learning Libraries: Molecular machine learning libraries include DeepChem (Ramsundar et al., 2019), DGL-LifeSci (Li et al., 2021) and TorchDrug (Zhu et al., 2022). 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.

GP implementations are not included, however, in the aforementioned libraries. In terms of atomistic systems, DScribe (Himanen et al., 2020) features, amongst other methods, the Smooth Overlap of Atomic Positions (SOAP) representation (Bartók et al., 2013) which is typically used in conjunction with a GP model to learn atomistic properties. Automatic Selection And Prediction (ASAP) (Cheng et al., 2020) also principally focusses on atomistic properties as well as dimensionality reduction and visualisation techniques for materials and molecules. Lastly, the Graphein library focusses on graph representations of proteins (Jamasb et al., 2021).

Graph Kernel Libraries: Graph kernel libraries include GraKel (Siglidis et al., 2020), graphkit-learn (Jia et al., 2021), graphkernels (Sugiyama et al., 2018), graph-kernels (Sugiyama & Borgwardt, 2015), pykernels (https://github.com/gmum/pykernels) and ChemoKernel (Gaüzére et al., 2012). 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 (Ghosh et al., 2018). In our work, we build on the GraKel library by interfacing it with GPyTorch, facilitating GP regression with GPU computation. Furthermore, we enable the graph kernel hyperparameters to be learned through the marginal likelihood objective as

Table 1: Molecular property prediction regression benchmark. RMSE values for 80/20 train/test split across 20 random trials.

GP Model		Dataset			
Kernel	Representation	Photoswitch	ESOL	FreeSolv	Lipophilicity
Tanimoto	fragprints	$\mathbf{20.9 \pm 0.7}$	0.71 ± 0.01	1.31 ± 0.06	$\mathbf{0.67 \pm 0.01}$
	fingerprints	23.4 ± 0.8	1.01 ± 0.01	1.93 ± 0.09	0.76 ± 0.01
	fragments	26.3 ± 0.8	0.91 ± 0.01	1.49 ± 0.05	0.80 ± 0.01
Scalar Product	fragprints	22.5 ± 0.7	0.88 ± 0.01	$\mathbf{1.27 \pm 0.02}$	0.77 ± 0.01
	fingerprints	24.8 ± 0.8	1.17 ± 0.01	1.93 ± 0.07	0.84 ± 0.01
	fragments	36.6 ± 1.0	1.15 ± 0.01	1.63 ± 0.03	$0.97. \pm 0.01$
String	SELFIES	24.9 ± 0.6	-	-	-
	SMILES	24.8 ± 0.7	$\mathbf{0.66 \pm 0.01}$	1.31 ± 0.01	$\mathbf{0.68 \pm 0.01}$
WL Kernel (GraKel)	graph	22.4 ± 1.4	1.04 ± 0.02	1.47 ± 0.06	0.74 ± 0.05

Table 2: UQ benchmark. NLPD values for 80/20 train/test split across 20 random trials.

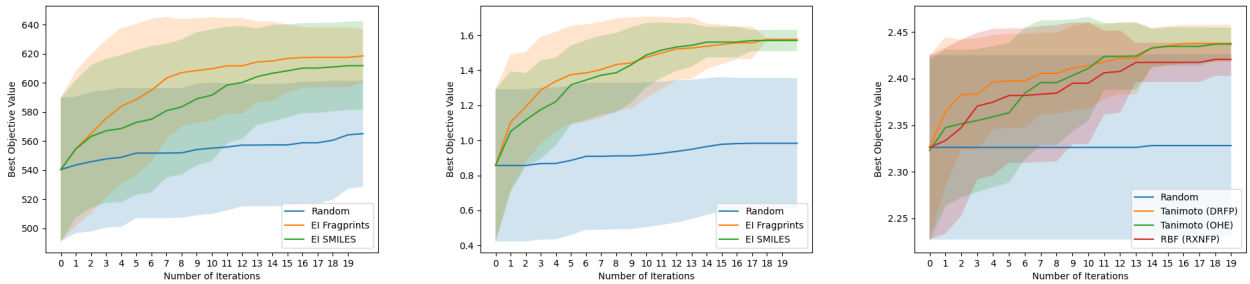
GP Model		Dataset			
Kernel	Representation	Photoswitch	ESOL	FreeSolv	Lipophilicity
Tanimoto	fragprints	$\mathbf{0.22 \pm 0.03}$	0.33 ± 0.01	0.28 ± 0.02	$\mathbf{0.71 \pm 0.01}$
	fingerprints	0.33 ± 0.03	0.71 ± 0.01	0.58 ± 0.03	0.85 ± 0.01
	fragments	0.50 ± 0.04	0.57 ± 0.01	0.44 ± 0.03	0.94 ± 0.02
Scalar Product	fragprints	$\mathbf{0.23 \pm 0.03}$	0.53 ± 0.01	0.25 ± 0.02	0.92 ± 0.01
	fingerprints	0.33 ± 0.03	0.84 ± 0.01	0.64 ± 0.03	1.03 ± 0.01
	fragments	0.80 ± 0.03	0.82 ± 0.01	0.54 ± 0.02	0.88 ± 0.10
String	SELFIES	0.37 ± 0.04	-	-	-
	SMILES	0.30 ± 0.04	$\mathbf{0.29 \pm 0.03}$	$\mathbf{0.16 \pm 0.02}$	$\mathbf{0.72 \pm 0.01}$
WL Kernel (GraKel)	graph	0.39 ± 0.11	0.76 ± 0.001	0.47 ± 0.02	-

opposed to being pre-specified and fixed upfront.

Molecular Bayesian Optimisation: BO over molecular space can be divided into two classes of methods. In the first class, molecules are encoded into the latent space of a variational autoencoder (VAE) (Gómez-Bombarelli et al., 2018). 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 (Griffiths et al., 2021; Griffiths & Hernández-Lobato, 2020; Tripp et al., 2020; Deshwal & Doppa, 2021; Grosnit et al., 2021; Verma & Chakraborty, 2021; Maus et al., 2022; Stanton et al., 2022). 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 (Pyzer-Knapp et al., 2015).

To date, there has been little work on BO directly over discrete molecular spaces. In Moss et al. (2020a), the authors use a string kernel GP trained on SMILES to perform BO to select from a candidate set of molecules. In Korovina et al. (2020), an optimal transport kernel GP is used for BO over molecular graphs. In Häse et al. (2021a) a surrogate based on the Nadarya-Watson estimator is defined such that



(a) Photoswitch (b) ESOL (c) Buchwald-Hartwig reactions

Figure 2: Bayesian optimisation performance. Standard error from 50 random initialisations, 20 for Buchwald-Hartwig.

the kernel density estimates are inferred using BNNs. The model is then trained on molecular descriptors. Lastly, in Hernández-Lobato et al. (2017) and Vakili et al. (2021) 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 long term 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 reactant molecules transform into product molecules. 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) (Ahneman et al., 2018), in asymmetric synthesis, where the reactions could result in different enantiomers, to maximise the enantiomeric excess (Zahrt et al., 2019), or to minimise the E-factor, which is the ratio between waste materials and the desired product (Schweidtmann et al., 2018).

A diverse set of studies have evaluated the optimisation of chemical reactions in single and multi-objective settings (Schweidtmann et al., 2018; Müller et al., 2022). Felton et al. (2021) and Häse et al. (2021b) benchmarked reaction optimisation algorithms in low-dimensional settings including reaction conditions, such as time, temperature, and concentrations. Shields et al. (2021) suggested BO as a general tool for chemical reaction optimisation and benchmarked their approach against human experts. Haywood et al. (2021) compared the yield prediction performance of different kernels and Pomberger et al. (2022) 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 (Shields et al., 2021; Pomberger et al., 2022; Hickman et al., 2022). In GAUCHE, we introduce reaction fingerprint kernels, based on existing reaction fingerprints (Schwaller et al., 2021a; Probst et al., 2022) and work independently of the number of reactant and reagent categories.

6. Conclusions and Future Work

We have introduced GAUCHE, a library for GAUssian Processes in CHEmistry with the aim of providing tools for uncertainty quantification and Bayesian optimisation that may hopefully be deployed for screening in laboratory settings. In future work, we seek to:

1. Expand the range of GP kernels we currently consider, most notably to include *deep kernels* based on GNN embeddings.
2. Perform more extensive benchmarking for uncertainty quantification and active learning against models such as BNNs.
3. Exploit the benefits of our autodiff framework to facilitate the learning of graph kernel hyperparameters through the GP marginal likelihood.
4. Broaden the application domains considered by GAUCHE to include examples in protein engineering.
5. Investigate more sophisticated GP-based optimization and active learning loops in chemistry applications (Eyke et al., 2020), like applying ideas from batch (González et al., 2016), multi-task (Swersky et al., 2013), multi-fidelity (Moss et al., 2020c), multi-objective (Daulton et al., 2020), controllable experimental noise (Moss et al., 2020b), or quantile (Torossian et al., 2020) optimisation.

References

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

Anderson, E., Veith, G. D., and Weininger, D. SMILES, a line notation and computerized interpreter for chemical structures. *Environmental Research Laboratory*, 1987.

Balandat, M., Karrer, B., Jiang, D., Daulton, S., Letham, B., Wilson, A. G., and Bakshy, E. BoTorch: A framework for efficient Monte-Carlo Bayesian optimization. *Advances in Neural Information Processing Systems*, 2020.

Bartók, A. P., Kondor, R., and Csányi, G. On representing chemical environments. *Physical Review B*, 2013.

Bengio, Y. *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.

Bento, A. P., Gaulton, A., Hersey, A., Bellis, L. J., Chambers, J., Davies, M., Krüger, F. A., Light, Y., Mak, L., McGlinchey, S., et al. The ChEMBL bioactivity database: An update. *Nucleic Acids Research*, 2014.

Breiman, L., Friedman, J. H., Olshen, R. A., and Stone, C. J. *Classification and regression trees*. Routledge, 2017.

Brochu, E., Cora, V. M., and De Freitas, N. 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.

Cancedda, N., Gaussier, E., Goutte, C., and Renders, J. M. Word sequence kernels. *Journal of Machine Learning Research*, 2003.

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

Capecchi, A., Probst, D., and Reymond, J.-L. One molecular fingerprint to rule them all: Drugs, biomolecules, and the metabolome. *Journal of Cheminformatics*, 2020.

Cheng, B., Griffiths, R.-R., Wengert, S., Kunkel, C., Stenczel, T., Zhu, B., Deringer, V. L., Bernstein, N., Margraf, J. T., Reuter, K., et al. Mapping materials and molecules. *Accounts of Chemical Research*, 2020.

Christie, B. D., Leland, B. A., and Nourse, J. G. Structure searching in chemical databases by direct lookup methods. *Journal of Chemical Information and Computer Sciences*, 1993.

Chuang, K. V. and Keiser, M. J. Comment on "predicting reaction performance in c–n cross-coupling using machine learning". *Science*, 2018.

Cowen-Rivers, A. I., Lyu, W., Tutunov, R., Wang, Z., Grosnit, A., Griffiths, R. R., Maraval, A. M., Jianye, H., Wang, J., Peters, J., et al. An empirical study of assumptions in Bayesian optimisation. *arXiv preprint arXiv:2012.03826*, 2020.

Daulton, S., Balandat, M., and Bakshy, E. Differentiable expected hypervolume improvement for parallel multi-objective Bayesian optimization. *Advances in Neural Information Processing Systems*, 2020.

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

Deshwal, A. and Doppa, J. Combining latent space and structured kernels for Bayesian optimization over combinatorial spaces. *Advances in Neural Information Processing Systems*, 2021.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Duvenaud, D. K., Maclaurin, D., Iparraguirre, J., Bombarell, R., Hirzel, T., Aspuru-Guzik, A., and Adams, R. P. Convolutional networks on graphs for learning molecular fingerprints. *Advances in Neural Information Processing Systems*, 2015.

Eyke, N. S., Green, W. H., and Jensen, K. F. Iterative experimental design based on active machine learning reduces the experimental burden associated with reaction screening. *Reaction Chemistry & Engineering*, 2020.

Felton, K. C., Rittig, J. G., and Lapkin, A. A. Summit: Benchmarking machine learning methods for reaction optimisation. *Chemistry-Methods*, 2021.

Frazier, P. I. A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

Gardner, J., Pleiss, G., Weinberger, K. Q., Bindel, D., and Wilson, A. G. GPyTorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration. In *Advances in Neural Information Processing Systems*, 2018.

Gaulton, A., Bellis, L. J., Bento, A. P., Chambers, J., Davies, M., Hersey, A., Light, Y., McGlinchey, S., Michalovich, D., Al-Lazikani, B., et al. ChEMBL: A large-scale bioactivity database for drug discovery. *Nucleic Acids Research*, 2012.

Gaüzére, B., Brun, L., Villemin, D., and Brun, M. Graph kernels based on relevant patterns and cycle information for chemoinformatics. In *Proceedings of the 21st International Conference on Pattern Recognition*, 2012.

Ghosh, S., Das, N., Gonçalves, T., Quaresma, P., and Kundu, M. The journey of graph kernels through two decades. *Computer Science Review*, 2018.

Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P., and Aspuru-Guzik, A. Automatic chemical design using a data-driven continuous representation of molecules. *ACS Central Science*, 2018.

González, J., Dai, Z., Hennig, P., and Lawrence, N. Batch Bayesian optimization via local penalization. In *Artificial Intelligence and Statistics*, 2016.

Gower, J. C. A general coefficient of similarity and some of its properties. *Biometrics*, 1971.

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

Graff, D. E., Aldeghi, M., Morrone, J. A., Jordan, K. E., Pyzer-Knapp, E. O., and Coley, C. W. Self-focusing virtual screening with active design space pruning. *arXiv preprint arXiv:2205.01753*, 2022.

Griffiths, R.-R. and Hernández-Lobato, J. M. Constrained Bayesian optimization for automatic chemical design using variational autoencoders. *Chemical Science*, 2020.

Griffiths, R.-R., Schwaller, P., and Lee, A. A. Dataset bias in the natural sciences: A case study in chemical reaction prediction and synthesis design. *arXiv preprint arXiv:2105.02637*, 2021.

Grosnit, A., Cowen-Rivers, A. I., Tutunov, R., Griffiths, R.-R., Wang, J., and Bou-Ammar, H. Are we forgetting about compositional optimisers in Bayesian optimisation? *arXiv preprint arXiv:2012.08240*, 2020.

Grosnit, A., Tutunov, R., Maraval, A. M., Griffiths, R.-R., Cowen-Rivers, A. I., Yang, L., Zhu, L., Lyu, W., Chen, Z., Wang, J., et al. High-dimensional Bayesian optimisation with variational autoencoders and deep metric learning. *arXiv preprint arXiv:2106.03609*, 2021.

Häse, F., Aldeghi, M., Hickman, R. J., Roch, L. M., and Aspuru-Guzik, A. Gryffin: An algorithm for Bayesian optimization of categorical variables informed by expert knowledge. *Applied Physics Reviews*, 2021a.

Häse, F., Aldeghi, M., Hickman, R. J., Roch, L. M., Christensen, M., Liles, E., Hein, J. E., and Aspuru-Guzik, A. Olympus: A benchmarking framework for noisy optimization and experiment planning. *Machine Learning: Science and Technology*, 2021b.

Hassan, M., Brown, R. D., Varma-O'Brien, S., and Rogers, D. Cheminformatics analysis and learning in a data pipelining environment. *Molecular Diversity*, 2006.

Haywood, A. L., Redshaw, J., Hanson-Heine, M. W., Taylor, A., Brown, A., Mason, A. M., Gärtner, T., and Hirst, J. D. Kernel methods for predicting yields of chemical reactions. *Journal of Chemical Information and Modeling*, 2021.

Hennig, P. and Schuler, C. J. Entropy search for information-efficient global optimization. *Journal of Machine Learning Research*, 2012.

Hernández-Lobato, J. M., Hoffman, M. W., and Ghahramani, Z. Predictive entropy search for efficient global optimization of black-box functions. *Advances in Neural Information Processing Systems*, 2014.

Hernández-Lobato, J. M., Requeima, J., Pyzer-Knapp, E. O., and Aspuru-Guzik, A. Parallel and distributed Thompson sampling for large-scale accelerated exploration of chemical space. In *International Conference on Machine Learning*, 2017.

Hickman, R., Ruža, J., Roch, L., Tribukait, H., and García-Durán, A. Equipping data-driven experiment planning for self-driving laboratories with semantic memory: Case studies of transfer learning in chemical reaction optimization. *ChemRxiv*, 2022.

Himanen, L., Jäger, M. O., Morooka, E. V., Canova, F. F., Ranawat, Y. S., Gao, D. Z., Rinke, P., and Foster, A. S. DScribe: Library of descriptors for machine learning in materials science. *Computer Physics Communications*, 2020.

Hu, W., Liu, B., Gomes, J., Zitnik, M., Liang, P., Pande, V., and Leskovec, J. Strategies for pre-training graph neural networks. In *International Conference on Learning Representations*, 2019.

Hwang, D., Lee, G., Jo, H., Yoon, S., and Ryu, S. A benchmark study on reliable molecular supervised learning via Bayesian learning. *arXiv preprint arXiv:2006.07021*, 2020.

Jablonka, K. M., Jothiappan, G. M., Wang, S., Smit, B., and Yoo, B. Bias free multiobjective active learning for materials design and discovery. *Nature Communications*, 2021.

Jamasb, A. R., Viñas, R., Ma, E. J., Harris, C., Huang, K., Hall, D., Lió, P., and Blundell, T. L. Graphein - a Python library for geometric deep learning and network analysis on protein structures and interaction networks. *bioRxiv*, 2021.

Jia, L., Gaüzère, B., and Honeine, P. graphkit-learn: A Python library for graph kernels based on linear patterns. *Pattern Recognition Letters*, 2021.

Johnson, M. A. and Maggiora, G. M. *Concepts and applications of molecular similarity*. Wiley, 1990.

Jones, D. R., Schonlau, M., and Welch, W. J. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 1998.

Jurafsky, D. and Martin, J. H. An introduction to natural language processing, computational linguistics, and speech recognition, 2000.

Kandasamy, K., Vysyaraju, K. R., Neiswanger, W., Paria, B., Collins, C. R., Schneider, J., Poczos, B., and Xing, E. P. Tuning hyperparameters without grad students: Scalable and robust Bayesian optimisation with Dragonfly. *Journal of Machine Learning Research*, 2020.

Kearnes, S., McCloskey, K., Berndl, M., Pande, V., and Riley, P. Molecular graph convolutions: Moving beyond fingerprints. *Journal of Computer-Aided Molecular Design*, 2016.

Korovina, K., Xu, S., Kandasamy, K., Neiswanger, W., Poczos, B., Schneider, J., and Xing, E. ChemBO: Bayesian optimization of small organic molecules with synthesizable recommendations. In *International Conference on Artificial Intelligence and Statistics*, 2020.

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

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

Landrum, G. RDKit: A software suite for cheminformatics, computational chemistry, and predictive modeling, 2013.

Li, M., Zhou, J., Hu, J., Fan, W., Zhang, Y., Gu, Y., and Karypis, G. DGL-LifeSci: An open-source toolkit for deep learning on graphs in life science. *ACS Omega*, 2021.

Liu, D. C. and Nocedal, J. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 1989.

Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N., and Watkins, C. Text classification using string kernels. *Journal of Machine Learning Research*, 2002.

MacKay, D. J., Mac Kay, D. J., et al. *Information theory, inference and learning algorithms*. Cambridge University Press, 2003.

Matthews, A. G., van der Wilk, M., Nickson, T., Fujii, K., Boukouvalas, A., León-Villagrá, P., Ghahramani, Z., and Hensman, J. GPflow: A Gaussian process library using TensorFlow. *Journal of Machine Learning Research*, 2017.

Maus, N., Jones, H. T., Moore, J. S., Kusner, M. J., Bradshaw, J., and Gardner, J. R. Local latent space Bayesian optimization over structured inputs. *arXiv preprint arXiv:2201.11872*, 2022.

McGregor, M. J. and Pallai, P. V. Clustering of large databases of compounds: using the MDL "keys" as structural descriptors. *Journal of Chemical Information and Computer Sciences*, 1997.

Mobley, D. L. and Guthrie, J. P. FreeSolv: A database of experimental and calculated hydration free energies, with input files. *Journal of Computer-Aided Molecular Design*, 2014.

Močkus, J. On Bayesian methods for seeking the extremum. In *Optimization techniques IFIP technical conference*, 1975.

Moss, H., Leslie, D., Beck, D., Gonzalez, J., and Rayson, P. BOSS: Bayesian optimization over string spaces. *Advances in Neural Information Processing Systems*, 2020a.

Moss, H. B. and Griffiths, R. R. Gaussian process molecule property prediction with FlowMO. *arXiv preprint arXiv:2010.01118*, 2020.

Moss, H. B., Leslie, D. S., and Rayson, P. BOSH: Bayesian optimization by sampling hierarchically. *arXiv preprint arXiv:2007.00939*, 2020b.

Moss, H. B., Leslie, D. S., and Rayson, P. MUMBO: Multi-task max-value Bayesian optimization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2020c.

Moss, H. B., Leslie, D. S., Gonzalez, J., and Rayson, P. Gibbon: General-purpose information-based Bayesian optimisation. *Journal of Machine Learning Research*, 2021.

Müller, P., Clayton, A. D., Manson, J., Riley, S., May, O. S., Govan, N., Notman, S., Ley, S. V., Chamberlain, T. W., and Bourne, R. A. Automated multi-objective reaction

optimisation: Which algorithm should I use? *Reaction Chemistry & Engineering*, 2022.

Nikolentzos, G., Siglidis, G., and Vazirgiannis, M. Graph kernels: A survey. *Journal of Artificial Intelligence Research*, 2021.

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

Pomberger, A., Pedrina McCarthy, A., Khan, A., Sung, S., Taylor, C., Gaunt, M., Colwell, L., Walz, D., and Lapkin, A. The effect of chemical representation on active machine learning towards closed-loop optimization. *ChemRxiv*, 2022.

Probst, D. and Reymond, J.-L. A probabilistic molecular fingerprint for big data settings. *Journal of Cheminformatics*, 2018.

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

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

Pyzer-Knapp, E. O., Suh, C., Gómez-Bombarelli, R., Aguilera-Iparraguirre, J., and Aspuru-Guzik, A. What is high-throughput virtual screening? A perspective from organic materials discovery. *Annual Review of Materials Research*, 2015.

Ralaivola, L., Swamidass, S. J., Saigo, H., and Baldi, P. Graph kernels for chemical informatics. *Neural networks*, 2005.

Ramsundar, B., Eastman, P., Walters, P., Pande, V., Leswing, K., and Wu, Z. *Deep Learning for the Life Sciences*. O'Reilly Media, 2019.

Rasmussen, C. E. and Ghahramani, Z. Occam's razor. In *Advances in Neural Information Processing Systems*, 2001.

Rogers, D. and Hahn, M. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling*, 2010.

Ryu, S., Kwon, Y., and Kim, W. Y. A bayesian graph convolutional network for reliable prediction of molecular properties with uncertainty quantification. *Chemical Science*, 2019.

Sandfort, F., Strieth-Kalthoff, F., Kühnemund, M., Beecks, C., and Glorius, F. A structure-based platform for predicting chemical reactivity. *Chem*, 2020.

Scalia, G., Grambow, C. A., Pernici, B., Li, Y.-P., and Green, W. H. Evaluating scalable uncertainty estimation methods for deep learning-based molecular property prediction. *Journal of Chemical Information and Modeling*, 2020.

Schneider, N., Lowe, D. M., Sayle, R. A., and Landrum, G. A. 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.

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

Schwaller, P., Vaucher, A. C., Laino, T., and Reymond, J.-L. Prediction of chemical reaction yields using deep learning. *Machine learning: Science and Technology*, 2021b.

Schweidtmann, A. M., Clayton, A. D., Holmes, N., Bradford, E., Bourne, R. A., and Lapkin, A. A. Machine learning meets continuous flow chemistry: Automated optimization towards the pareto front of multiple objectives. *Chemical Engineering Journal*, 2018.

Shahriari, B., Swersky, K., Wang, Z., Adams, R. P., and de Freitas, N. Taking the human out of the loop: A review of Bayesian optimization. *Proceedings of the IEEE*, 2016.

Shervashidze, N., Schweitzer, P., Van Leeuwen, E. J., Mehlhorn, K., and Borgwardt, K. M. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12(9), 2011.

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

Siglidis, G., Nikolentzos, G., Limnios, S., Giatsidis, C., Skianis, K., and Vazirgiannis, M. Grakel: A graph kernel library in Python. *Journal of Machine Learning Research*, 2020.

Snoek, J., Rippel, O., Swersky, K., Kiros, R., Satish, N., Sundaram, N., Patwary, M., Prabhat, M., and Adams, R. Scalable Bayesian optimization using deep neural networks. In *International Conference on Machine Learning*, 2015.

Springenberg, J. T., Klein, A., Falkner, S., and Hutter, F. Bayesian optimization with robust Bayesian neural networks. *Advances in Neural Information Processing Systems*, 2016.

Stanton, S., Maddox, W., Gruver, N., Maffettone, P., Delaney, E., Greenside, P., and Wilson, A. G. Accelerating Bayesian optimization for biological sequence design with denoising autoencoders. *arXiv preprint arXiv:2203.12742*, 2022.

Sugiyama, M. and Borgwardt, K. Halting in random walk kernels. *Advances in Neural Information Processing Systems*, 2015.

Sugiyama, M., Ghisu, M. E., Llinares-López, F., and Borgwardt, K. graphkernels: R and Python packages for graph comparison. *Bioinformatics*, 2018.

Swersky, K., Snoek, J., and Adams, R. P. Multi-task Bayesian optimization. *Advances in Neural Information Processing Systems*, 2013.

Thawani, A. R., Griffiths, R.-R., Jamasb, A., Bourached, A., Jones, P., McCorkindale, W., Aldrick, A. A., and Lee, A. A. The photoswitch dataset: A molecular machine learning benchmark for the advancement of synthetic chemistry. *arXiv preprint arXiv:2008.03226*, 2020.

Torossian, L., Picheny, V., and Durrande, N. Bayesian quantile and expectile optimisation. *arXiv preprint arXiv:2001.04833*, 2020.

Tripp, A., Daxberger, E., and Hernández-Lobato, J. M. Sample-efficient optimization in the latent space of deep generative models via weighted retraining. *Advances in Neural Information Processing Systems*, 2020.

Turner, R., Eriksson, D., McCourt, M., Kiili, J., Laaksonen, E., Xu, Z., and Guyon, I. Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020. In *NeurIPS 2020 Competition and Demonstration Track*, 2021.

Vakili, S., Moss, H., Artemev, A., Dutordoir, V., and Picheny, V. Scalable Thompson sampling using sparse Gaussian process models. *Advances in Neural Information Processing Systems*, 2021.

van der Wilk, M., Dutordoir, V., John, S., Artemev, A., Adam, V., and Hensman, J. A framework for interdomain and multioutput Gaussian processes. *arXiv preprint arXiv:2003.01115*, 2020.

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

Vishwanathan, S. V. N., Schraudolph, N. N., Kondor, R., and Borgwardt, K. M. Graph kernels. *Journal of Machine Learning Research*, 2010.

Wang, Z. and Jegelka, S. Max-value entropy search for efficient Bayesian optimization. In *International Conference on Machine Learning*, 2017.

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

Xia, X., Maliski, E. G., Gallant, P., and Rogers, D. Classification of kinase inhibitors using a Bayesian model. *Journal of Medicinal Chemistry*, 2004.

Zahrt, A. F., Henle, J. J., Rose, B. T., Wang, Y., Darrow, W. T., and Denmark, S. E. Prediction of higher-selectivity catalysts by computer-driven workflow and machine learning. *Science*, 2019.

Zhang, Y. and Ling, C. A strategy to apply machine learning to small datasets in materials science. *Npj Computational Materials*, 2018.

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

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

Zhu, Z., Shi, C., Zhang, Z., Liu, S., Xu, M., Yuan, X., Zhang, Y., Chen, J., Cai, H., Lu, J., Ma, C., Liu, R., Xhonneux, L.-P., Qu, M., and Tang, J. TorchDrug: A powerful and flexible machine learning platform for drug discovery. *arXiv preprint arXiv:2202.08320*, 2022.

A. Chemical Reaction Yield Prediction Experiments

Further regression and uncertainty quantification experiments are presented in Table A1. The differential reaction fingerprint in conjunction with the Tanimoto kernel is the best-performing reaction representation.

Table A1: Chemical reaction regression benchmark. 80/20 train/test split across 20 random trials.

GP Model		**Buchwald-Hartwig**			
Kernel	Representation	RMSE ↓	R^2 score ↑	MSLL ↓	QCE ↓
Tanimoto	OHE	7.94 ± 0.05	0.91 ± 0.001	-0.06 ± 0.002	0.011 ± 0.001
	DRFP	$\mathbf{6.48 \pm 0.45}$	$\mathbf{0.94 \pm 0.015}$	$\mathbf{-0.15 \pm 0.07}$	0.027 ± 0.002
Scalar Product	OHE	15.23 ± 0.052	0.69 ± 0.002	0.57 ± 0.002	0.008 ± 0.001
	DRFP	14.63 ± 0.050	0.71 ± 0.002	0.55 ± 0.002	0.010 ± 0.001
RBF	RXNFP	10.79 ± 0.049	0.84 ± 0.001	0.37 ± 0.005	0.024 ± 0.001
		Suzuki-Miyaura			
Tanimoto	OHE	11.18 ± 0.036	0.83 ± 0.001	0.23 ± 0.001	0.007 ± 0.001
	DRFP	11.46 ± 0.038	0.83 ± 0.001	0.25 ± 0.006	0.019 ± 0.000
Scalar Product	OHE	19.91 ± 0.042	0.47 ± 0.003	0.82 ± 0.001	0.012 ± 0.001
	DRFP	19.66 ± 0.042	0.52 ± 0.003	0.81 ± 0.001	0.014 ± 0.001
RBF	RXNFP	13.83 ± 0.048	0.75 ± 0.002	0.50 ± 0.001	0.007 ± 0.001

B. Uncertainty Quantification Experiments

In Table B2 and Table B3 we present further uncertainty quantification metrics. Numerical errors were encountered with the WL kernel on the large lipophilicity dataset which invalidated the results and so the corresponding entry is left blank. The native random walk kernel was discontinued (for the time being) due to poor performance!

Table B2: UQ Benchmark. MSLL Values (↓) for 80/20 Train/Test Split.

GP Model		**Dataset**			
Kernel	Representation	Photoswitch	ESOL	FreeSolv	Lipophilicity
Tanimoto	fragprints	$\mathbf{0.06 \pm 0.01}$	0.17 ± 0.04	0.16 ± 0.02	$\mathbf{0.50 \pm 0.006}$
	fingerprints	0.16 ± 0.01	0.55 ± 0.01	0.42 ± 0.02	0.63 ± 0.004
	fragments	0.27 ± 0.01	0.34 ± 0.04	0.24 ± 0.02	0.72 ± 0.003
Scalar Product	fragprints	0.03 ± 0.01	0.32 ± 0.004	0.06 ± 0.01	0.67 ± 0.003
	fingerprints	0.11 ± 0.01	0.64 ± 0.006	0.41 ± 0.02	0.79 ± 0.003
	fragments	0.56 ± 0.01	0.58 ± 0.005	0.29 ± 0.01	0.94 ± 0.003
String	SELFIES	0.13 ± 0.01	-	-	-
	SMILES	$\mathbf{0.08 \pm 0.02}$	$\mathbf{0.03 \pm 0.005}$	$\mathbf{0.03 \pm 0.02}$	$\mathbf{0.52 \pm 0.002}$
WL Kernel (GraKel)	graph	0.14 ± 0.03	0.54 ± 0.01	0.26 ± 0.01	-

Table B3: 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	SELFIES	0.031 ± 0.006	-	-	-
	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	0.066 ± 0.014