Hash Collisions in Molecular Fingerprints: Effects on Property Prediction and Bayesian Optimization

Walter Virany
University of Toronto
wvirany@cs.toronto.edu

Austin Tripp
Valence Labs
www.austintripp.ca

Abstract

Molecular fingerprinting methods use hash functions to create fixed-length vector representations of molecules. However, hash collisions cause distinct substructures to be represented with the same feature, leading to overestimates in molecular similarity calculations. We investigate whether using exact fingerprints improves accuracy compared to standard compressed fingerprints in molecular property prediction and Bayesian optimization where the underlying predictive model is a Gaussian process. We find that using exact fingerprints yields a small yet consistent improvement in predictive accuracy on five molecular property prediction benchmarks from the DOCKSTRING dataset. However, these gains did not translate to significant improvements in Bayesian optimization performance.

1 Introduction

Molecular fingerprints are vector representations that encode chemical structure information [Yang et al., 2022]. Despite substantial advances in AI, neural network representations of molecules have not been shown to robustly outperform fingerprints on molecular property prediction tasks, particularly given limited training data [Green et al., 2023, Praski et al., 2025].

Ideally, each dimension in a molecular fingerprint should represent a unique substructure in a molecule. For example, in Extended Connectivity Fingerprints (ECFPs) [Rogers and Hahn, 2010],[1] each dimension represents a circular substructure up to a given radius. However, due to the combinatorially large number of possible substructures, such fingerprints would be extremely high dimensional. Common practice is to instead use a *fixed*-length vector, usually of dimension 1024 or 2048, using hash functions to map substructure to a dimension. When there are more structures present than the length of this vector, it will inevitably result in *collisions*, where a single dimension is used to represent multiple distinct substructures. An example of this is given in Figure 1.

Intuitively, one would expect the effect of such collisions to be problematic, since it prevents machine learning algorithms from distinguishing between substructures whose effects may be different. In this paper we investigate this hypothesis quantitatively using Gaussian process (GP) models. Because GPs are kernel methods which typically use the *similarity* between fingerprints as input, the effect of hash collisions can be isolated without introducing confounding factors like a changed number of parameters. Our results in §4 show that eliminating collisions results in a small but consistent improvement on regression metrics, but has a negligible effect on Bayesian optimization (BO) performance. Given GPs' natural applicability to lab-in-the-loop active learning tasks [Moss and Griffiths, 2020, Griffiths et al., 2023], we believe these results provide important guidance to practitioners.

[1] Alternatively known as *Morgan* fingerprints. Arguably, this is the most popular type of fingerprint.

Accepted at the AI for Science workshop (NeurIPS 2025).

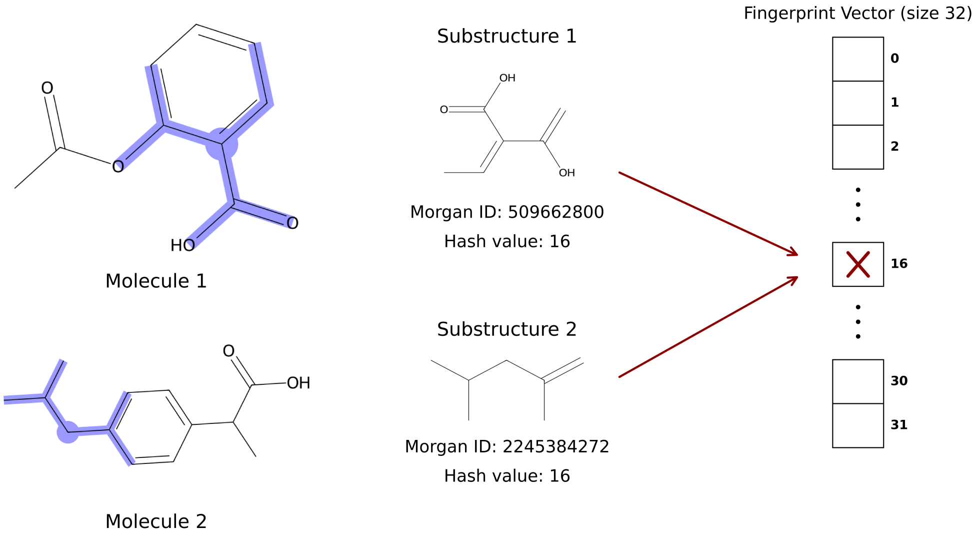

Figure 1: Example of collisions. (Left) Two structurally different molecules (SMILES strings are `CC(=O)OC1=CC=CC=C1C(=O)O` and `CC(C)CC1=CC=C(C=C1)C(C)C(=O)O`) with highlighted circular substructures of radius 2. (Middle) Highlighted substructures and their corresponding Morgan identifiers, as well as the resulting hash values after taking modulo 32 (corresponding to the fingerprint size). (Right) Fingerprint vector where both distinct substructures map to the same element given by the hash key, demonstrating how different substructures can map to the same dimension.

2 Background

2.1 Gaussian processes

GPs are a commonly used class of probabilistic models which assume that the data follows a joint multivariate Gaussian distribution. The mean of the distribution is given by a mean function $\mu : \mathcal{X} \to \mathbb{R}$, and the covariance is given by a positive-definite kernel function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, where $\mathcal{X}$ denotes some input space. A detailed treatment of GPs can be found in several textbooks [Rasmussen and Williams, 2006, Garnett, 2023]. GPs have become popular for molecular property prediction due to their efficacy in low-data regimes [Green et al., 2023, Moss and Griffiths, 2020, Griffiths et al., 2023]. A common choice of kernel function for molecular fingerprints is the *Tanimoto kernel*, which computes the Tanimoto similarity between two molecules, defined as

$$k_{\text{Tanimoto}}(\mathbf{x}, \mathbf{x}') = \frac{\sum_i \min(x_i, x_i')}{\sum_i \max(x_i, x_i')} \,, \tag{1}$$

where $\mathbf{x}, \mathbf{x}'$ are molecular fingerprint vectors. The full covariance function then takes the form

$$k(\mathbf{x}, \mathbf{x}') = a^2 \, k_{\text{Tanimoto}}(\mathbf{x}, \mathbf{x}') + \sigma_n^2 \, \delta(\mathbf{x}, \mathbf{x}') \,, \tag{2}$$

where a^2 and σ_n^2 are the amplitude and noise variance hyperparameters respectively, and $\delta(\mathbf{x}, \mathbf{x}')$ is the Kronecker delta.

2.2 Bayesian optimization

BO is a sequential optimization method for expensive black-box functions [Garnett, 2023]. BO uses a probabilistic surrogate model (commonly a GP) to approximate the unknown objective function from observed data. At each iteration, an acquisition function balances exploration of uncertain regions with exploitation of promising areas to select the next point(s) to evaluate. In molecular optimization, this corresponds to selecting molecules for expensive experimental evaluation (e.g., synthesis and testing). The acquired data is used to retrain the surrogate model, yielding a more accurate approximation of the objective function. This process is then repeated until some stopping criterion is met (e.g., budget exhausted or satisfactory molecules found).

BO heavily relies on accurate predictions of the objective function by the surrogate model. However, as we established, when using a GP surrogate model, the performance can suffer as a result of overestimates of the similarities between molecules. This paper investigates whether exact fingerprints improve GP accuracy and whether this has a substantial effect on BO performance.

3 Methods

3.1 Fingerprint representations

We evaluated three different strategies for encoding molecular fingerprints. Exact fingerprints, as previously mentioned, preserve all unique substructures without compression, resulting in variable-length representations depending on the number of unique substructures present in the molecule. In contrast, compressed fingerprints use fixed-length vectors with RDKit's default hash function, causing collisions when the number of unique substructures exceeds the vector dimension. Finally, we used the Sort&Slice method as proposed by Dablander et al. [2024], which selects the most frequently occurring substructures from a reference dataset rather than using hash-based folding. This eliminates collisions for the most common substructures, reducing the number of hash collisions overall.

For all three approaches, we used count-based ECFPs with radius 2. Count fingerprints store the frequency of each substructure, as opposed to binary fingerprints which only indicate the presence/absence of each substructure. Our rationale for this was to include as much information as possible in the compressed fingerprints, in order to not give exact fingerprints an artificial advantage. For the Sort&Slice method, we calibrated the substructure selection using the ZINC250k dataset. Thus, all three fingerprint methods compute the same underlying substructure identifiers, with differences arising solely from how these are encoded into vectors.

3.2 Gaussian process implementation

Standard GP libraries typically require fixed-dimensional input vectors, making them incompatible with exact fingerprints that have variable dimensions. To address this, we developed a custom GP implementation that works directly with RDKit fingerprint objects by computing Tanimoto similarities using the Tanimoto kernel and storing only the resulting kernel matrices. This approach enables fair comparison between exact and fixed-length fingerprints using identical GP methodology with performance differences attributable solely to the fingerprint representation and resulting kernel evaluations.

3.3 Experiment design

Datasets and tasks. We evaluated our methods on five protein-ligand docking targets from the DOCKSTRING regression benchmark [García-Ortegón et al., 2022]. DOCKSTRING provides a dataset of over 260,000 diverse molecules each docked against 58 protein targets, resulting in over 15 million docking scores. The five targets in the regression benchmark span different protein families and difficulty levels: PARP1 (enzyme, easy), F2 (protease, easy-medium), KIT (kinase, medium), ESR2 (nuclear receptor, hard), and PGR (nuclear receptor, hard). We use the predefined cluster-based train/test split to prevent data leakage from structurally similar molecules.

Molecular property prediction. To assess the effects of hash collisions in molecular fingerprints on property prediction, we evaluated GP regression performance on the aforementioned tasks for each fingerprint type. For the fixed-length fingerprint vectors (compressed and Sort&Slice), we tested the following fingerprint dimensions: 512, 1024, 2048, and 4096. Although the full training set contains approximately 220,000 molecules, we trained on a subset of 10,000 randomly sampled molecules due to the computational expense of GP training, following the same methodology outlined in García-Ortegón et al. [2022]. We evaluated predictions using R^2, mean squared error (MSE), and mean absolute error (MAE) on the predefined test set consisting of approximately 38,000 molecules. Each experiment was run under two settings of the GP hyperparameters: fixed and optimized (details on hyperparameter settings can be found in Appendix B). Results were averaged across 10 random trials for each setting.

Bayesian optimization. We also investigated the effect of hash collisions in BO. Our hypothesis was that even small differences in predictive accuracy would compound, yielding significant differences in BO outcomes over many iterations. To test this, we simulated BO on the same five DOCKSTRING targets using a GP surrogate model with fixed hyperparameters. We used the complete dataset (combining training and test sets) as the candidate pool, randomly selected 1,000 molecules from the bottom 80% of the dataset as the initial training set, and ran optimization with a budget of 1,000 iterations using expected improvement (EI) as the acquisition function. The same three fingerprint configurations previously discussed were compared, using fingerprint dimensions of 1024 and 2048 for the fixed-length fingerprints. We evaluated performance using area under the curve (AUC) of the best observed value over time, and averaged our results across 5 random trials of each experiment setting.

4 Results

4.1 Molecular property prediction

Exact fingerprints consistently outperformed compressed fingerprints across all DOCKSTRING targets under both hyperparameter settings. Table 1 shows results with optimized hyperparameters, where exact fingerprints universally outperformed the *best*-scoring compressed configurations, with improvements in R^2 scores ranging from 0.006 (F2) to 0.017 (KIT). Against Sort&Slice fingerprints, exact fingerprints showed mixed performance: outperforming on two targets (ESR2, KIT), matching performance on two targets (F2, PARP1), and being outperformed on one target (PGR), where Sort&Slice achieved an R^2 score of 0.480 vs. 0.470. Notably, PGR was both the hardest task (with the lowest overall R^2 scores) and the only target where Sort&Slice outperformed exact fingerprints, achieving its best performance at the smallest dimension tested (512) rather than the typical 4096. These same patterns were observed across all evaluation metrics (MSE and MAE) and were particularly pronounced under fixed hyperparameter settings, where exact fingerprints outperformed all alternatives on every target. Complete results are provided in Appendix C.

To assess statistical significance, we performed one-way ANOVA followed by Tukey's HSD post hoc test with type I error rate $\alpha = 0.05$, comparing exact fingerprints against each alternative. Exact fingerprints showed statistically significant performance improvements against compressed fingerprints across all targets, fingerprint sizes, and hyperparameter settings. For Sort&Slice, the performance differences observed were statistically significant on ESR2, KIT, and PGR, while differences on F2 and PARP1 were not significant.

Figure 2 demonstrates the monotonic relationship between fingerprint dimension and predictive performance for the ESR2 and KIT targets. Fixed-length fingerprints show consistent improvements in R^2 score as dimension increases from 512 to 4096, approaching exact fingerprint performance. This scaling pattern was observed for both compressed and Sort&Slice fingerprints across nearly all targets and hyperparameter settings, with the sole exception of Sort&Slice on PGR with optimized hyperparameters. These observations are consistent with the hypothesis that mitigating errors in

Table 1: GP Regression Performance with Optimized Hyperparameters. Values show mean R^2 scores ± 1 standard deviation across 10 random train/test splits. For compressed and Sort&Slice fingerprints, numbers in parentheses indicate the best-performing fingerprint dimension among those tested (512, 1024, 2048, and 4096). Bold indicates best performance for each target. An asterisk * indicates statistically significant difference from exact fingerprint based on Tukey's HSD test ($\alpha = 0.05$).

Target	Exact	Compressed	Sort&Slice
ESR2	**0.553** ± 0.003	*0.54 ± 0.003 (4096)	*0.545 ± 0.004 (4096)
F2	**0.848** ± 0.002	*0.842 ± 0.002 (4096)	**0.848** ± 0.002 (4096)
KIT	**0.745** ± 0.002	*0.728 ± 0.002 (4096)	*0.739 ± 0.003 (4096)
PARP1	**0.882** ± 0.001	*0.876 ± 0.001 (4096)	**0.882** ± 0.001 (4096)
PGR	0.470 ± 0.004	*0.457 ± 0.004 (4096)	***0.480** ± 0.003 (512)

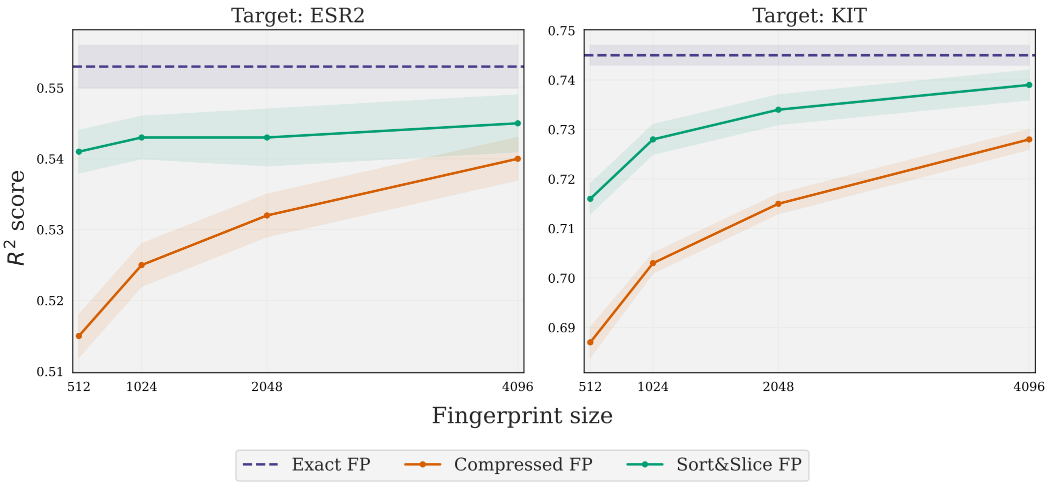

Figure 2: R^2 scores for GP regression with optimized hyperparameters as a function of fingerprint dimension on ESR2 (left) and KIT (right) targets. Exact fingerprints (purple) consistently outperform compressed fingerprints (orange) and Sort&Slice method (green). Dark lines indicate mean and shaded regions indicate ±1 standard deviation across 10 random train/test splits. Note that the dimension of exact fingerprints is not changing, but the performance is included as a horizontal line for reference.

similarity calculations leads to improved regression performance in GPs. We provide further analysis of hash collision rates and the resulting effects on similarity calculations in Appendix A.

4.2 Bayesian optimization

Despite the consistent improvements in GP regression accuracy, exact fingerprints yielded no significant improvement in BO performance. Table 2 summarizes AUC scores across all five targets, showing no clear advantage for any fingerprint configuration. In fact, exact fingerprints achieved the best performance on only one target (ESR2), while compressed and Sort&Slice methods matched or exceeded exact performance on the remaining targets.

Figure 3 shows representative BO trajectories for two DOCKSTRING targets, illustrating how all fingerprint configurations follow similar optimization paths despite their different regression accuracies. This disconnect suggests that the small improvements observed in predictive accuracy do not necessarily translate to significant differences in the sequential decision-making process that drives molecular optimization.

Table 2: Bayesian Optimization Performance (AUC scores). Values show mean ±1 standard deviation across 5 random trials. Bold indicates best performance for each target.

		Compressed		Sort&Slice	
Target	Exact	1024	2048	1024	2048
ESR2	**0.973** ± 0.004	0.965 ± 0.015	0.952 ± 0.013	0.968 ± 0.013	0.972 ± 0.006
F2	0.986 ± 0.004	**0.989** ± 0.004	**0.989** ± 0.004	0.983 ± 0.005	0.988 ± 0.004
KIT	0.973 ± 0.007	0.973 ± 0.01	0.971 ± 0.005	0.978 ± 0.005	**0.981** ± 0.005
PARP1	0.972 ± 0.004	0.970 ± 0.003	0.977 ± 0.002	0.977 ± 0.002	**0.978** ± 0.006
PGR	0.860 ± 0.023	0.868 ± 0.035	0.856 ± 0.028	0.867 ± 0.021	**0.873** ± 0.023

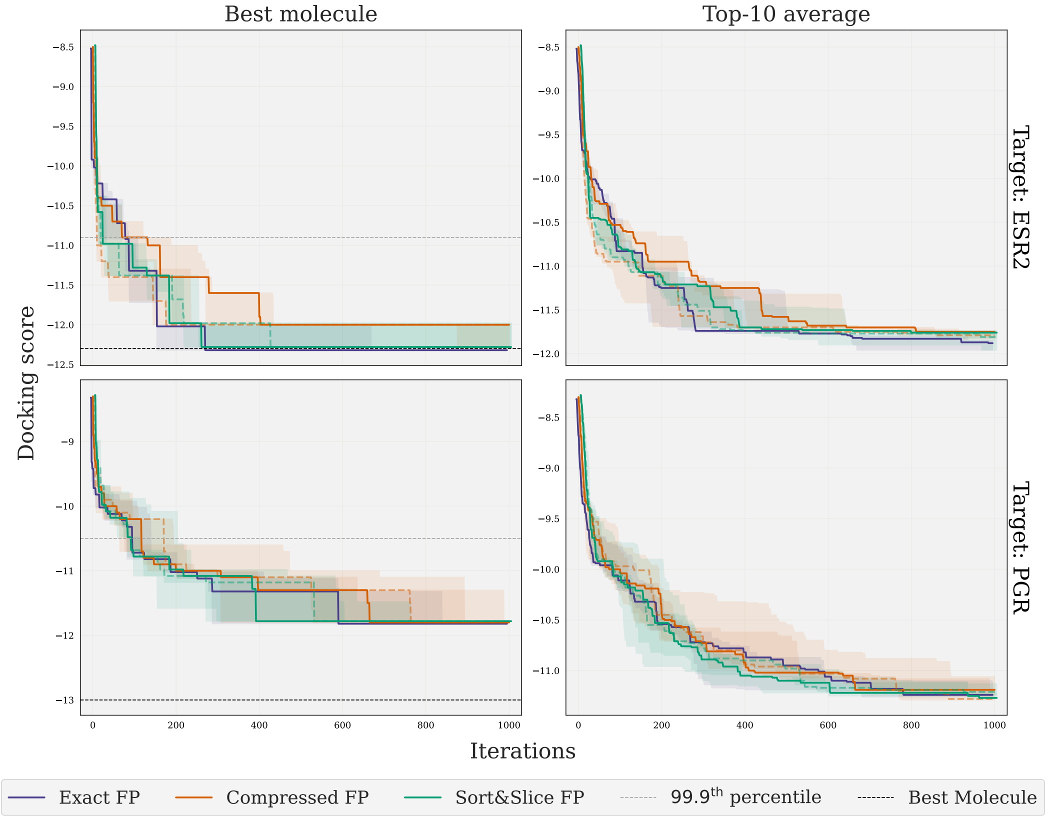

Figure 3: BO trajectories for two targets: ESR2 (top) and PGR (bottom). The first column shows the score of the best molecule at each iteration, and the second column shows the average score of the top 10 acquired molecules at each iteration. For fixed-length fingerprint types, two fingerprint sizes are shown: 1024 (dashed line) and 2048 (solid line). Dark lines indicate the median and shaded regions indicate the 1st and 3rd quartiles over 5 random trials. The dashed horizontal lines in the left-hand figures indicate the 99.9th percentile and the best possible score for each target.

5 Discussion

Our results demonstrate that exact fingerprints consistently improve GP regression performance across all DOCKSTRING targets, supporting our hypothesis that hash collisions lead to overestimated molecular similarities and thus reduced predictive accuracy. The Sort&Slice method performs comparably to exact fingerprints, suggesting that preserving the most informative substructures may be an effective alternative to eliminating all hash collisions. Interestingly, these regression improvements did not translate to enhanced Bayesian optimization performance, where all fingerprint types achieved similar AUC scores. Future work could further investigate the relationship between BO and predictive accuracy of the surrogate model. We note that, in addition to predictive performance, accurate uncertainty estimates are also a key component of successful BO procedures.

Several limitations should be acknowledged. The Sort&Slice method performance may be overestimated due to potential data leakage, as we calibrated this method using the ZINC250k dataset, which was also the dataset used for training and evaluation. Additionally, the optimization landscapes may not have been sufficiently challenging to reveal differences in BO between fingerprint types. For example, for even the hardest target (PGR), all BO procedures found top 0.1% molecules within several hundred iterations. In addition to addressing these, future work could investigate the effects of hash collisions in other types of fingerprints or different kernel methods such as SVMs. While preliminary results did show similar observations with different fingerprint types, a systematic analysis might prove valuable. Despite these limitations, our work suggests that for property prediction tasks, methods which reduce hash collisions in molecular fingerprints provide modest but consistent improvements, while their benefits for BO remain unclear.

References

Markus Dablander, Thierry Hanser, Renaud Lambiotte, and Garrett M. Morris. Sort & Slice: A Simple and Superior Alternative to Hash-Based Folding for Extended-Connectivity Fingerprints. *Journal of Cheminformatics*, 16(1):135, 2024.

Miguel García-Ortegón, Gregor N. C. Simm, Austin J. Tripp, José Miguel Hernández-Lobato, Andreas Bender, and Sergio Bacallado. DOCKSTRING: Easy Molecular Docking Yields Better Benchmarks for Ligand Design. *Journal of Chemical Information and Modeling*, 62(15):3486–3502, 2022.

Roman Garnett. *Bayesian Optimization*. Cambridge University Press, 2023.

Jacob Green, Cecilia Cabrera Diaz, Maximilian A. H. Jakobs, Andrea Dimitracopoulos, Mark van der Wilk, and Ryan D. Greenhalgh. Current Methods for Drug Property Prediction in the Real World, 2023. URL `http://arxiv.org/abs/2309.17161`. arXiv:2309.17161.

Ryan-Rhys Griffiths, Leo Klarner, Henry B. Moss, Aditya Ravuri, Sang Truong, Samuel Stanton, Gary Tom, Bojana Rankovic, Yuanqi Du, Arian Jamasb, Aryan Deshwal, Julius Schwartz, Austin Tripp, Gregory Kell, Simon Frieder, Anthony Bourached, Alex Chan, Jacob Moss, Chengzhi Guo, Johannes Durholt, Saudamini Chaurasia, Felix Strieth-Kalthoff, Alpha A. Lee, Bingqing Cheng, Alán Aspuru-Guzik, Philippe Schwaller, and Jian Tang. GAUCHE: A Library for Gaussian Processes in Chemistry, 2023. URL `http://arxiv.org/abs/2212.04450`. arXiv:2212.04450.

Henry B. Moss and Ryan-Rhys Griffiths. Gaussian Process Molecule Property Prediction with FlowMO, 2020. URL `http://arxiv.org/abs/2010.01118`. arXiv:2010.01118.

Mateusz Praski, Jakub Adamczyk, and Wojciech Czech. Benchmarking pretrained molecular embedding models for molecular representation learning. *arXiv preprint arXiv:2508.06199*, 2025.

Carl Edward Rasmussen and Christopher KI Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

David Rogers and Mathew Hahn. Extended-Connectivity Fingerprints. *Journal of Chemical Information and Modeling*, 50(5):742–754, 2010.

Jingbo Yang, Yiyang Cai, Kairui Zhao, Hongbo Xie, and Xiujie Chen. Concepts and applications of chemical fingerprint for hit and lead screening. *Drug discovery today*, 27(11):103356, 2022.

A Hash collisions analysis

To quantify the prevalence of hash collisions in molecular fingerprints, we analyzed the frequency of collisions in 10,000 pairs of molecules randomly sampled from the DOCKSTRING dataset, and the resulting effect on Tanimoto similarity calculations.

Table 3: Pairwise collision statistics and Tanimoto similarity differences for compressed fingerprints. Values are reported as the mean over 10,000 randomly sampled pairs of molecules. Overestimation is measured as the difference between compressed and exact Tanimoto similarity.

	512	**1024**	**2048**	**4096**
Pairwise collisions	4.87	2.43	1.41	0.48
Exact Tanimoto similarity	0.167	0.167	0.167	0.167
Compressed Tanimoto similarity	0.198	0.183	0.176	0.171
Overestimation	0.031	0.0152	0.008	0.004

Table 3 demonstrates the inverse relationship between fingerprint size and the number of hash collisions between pairs of molecules. As the fingerprint size increases, the number of hash collisions decreases, along with the overestimation in Tanimoto similarity. Figure 4 visualizes this systematic overestimation, where molecule pairs with more collisions tend to lie further above the diagonal line corresponding to equal similarity calculations.

B GP hyperparameter selection

We evaluated GP performance under two hyperparameter settings: fixed and optimized hyperparameters.

Fixed hyperparameters. For the fixed hyperparameter setting, we used the following values across all experiments:

- Amplitude: Set to the empirical variance of the training targets for each dataset
- Noise variance: Set to $0.01\times$ amplitude
- Mean function: Constant mean set to the empirical mean of the training targets

These values represent commonly used heuristics in GP regression and ensure consistent comparison across different fingerprint representations.

Optimized hyperparameters. For the optimized setting, we learned the hyperparameters by optimizing the marginal log-likelihood under gradient descent. Our optimization procedure is outlined as follows:

1. Initialization: Started with the same values used for fixed hyperparameters described above
2. Optimizer: Used Adam optimizer with learning rate 0.01
3. Objective: Maximized the marginal log-likelihood of the training data
4. Convergence criteria: Gradient norm below 10^{-3} or maximum 10,000 iterations reached
5. Numerical stability: Applied lower bounds on noise variance (minimum $10^{-4} \times$ target variance) to prevent numerical instability

The optimization was performed independently for each combination of fingerprint type, dataset, and random trial, allowing each configuration to find its optimal hyperparameters. This approach ensures that performance differences between fingerprint types are not confounded by suboptimal hyperparameter choices.

For the BO experiments, we used only the fixed hyperparameter setting to maintain computational efficiency during the sequential optimization process.

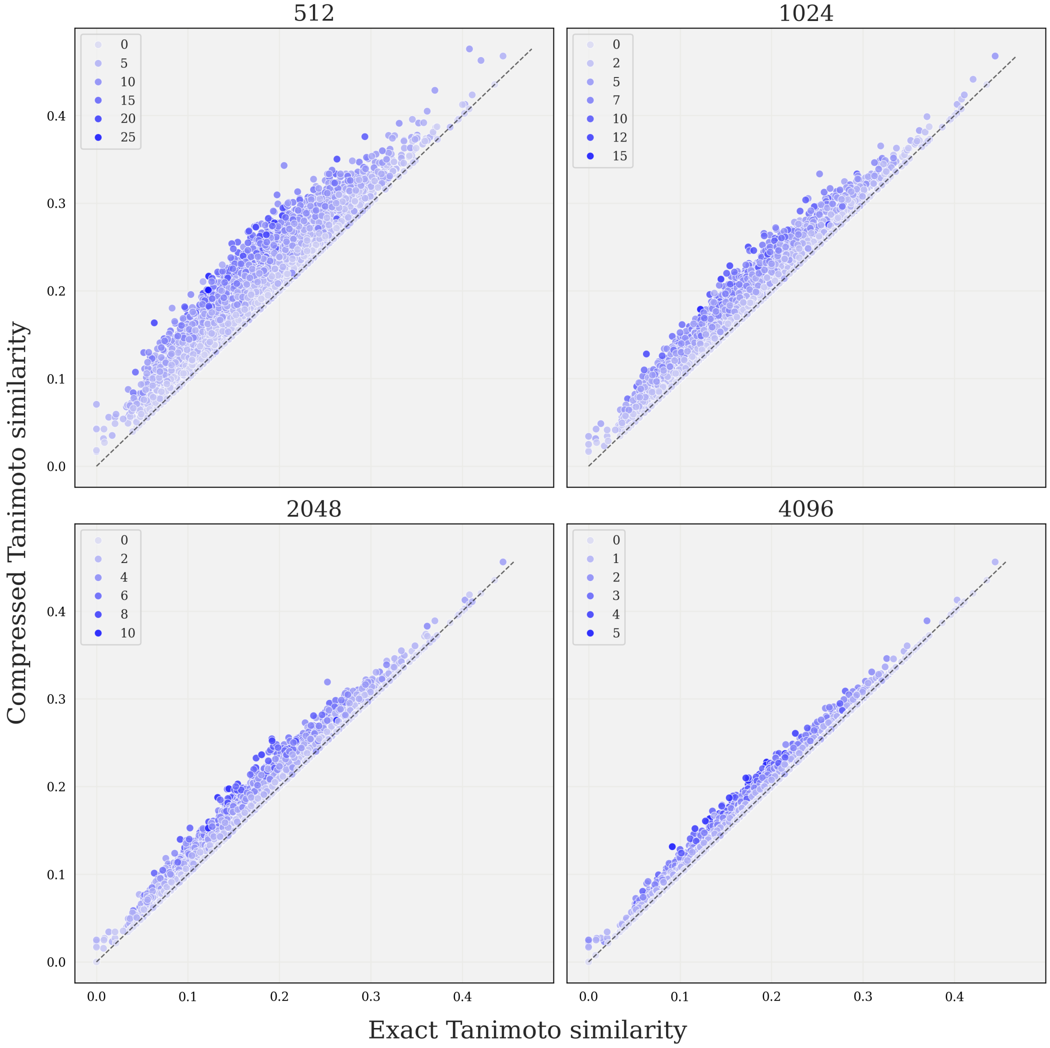

Figure 4: Tanimoto similarity in compressed vs. exact fingerprints for 10,000 pairs of molecules. Each point represents one pair, where the x-axis is the Tanimoto similarity computed between exact fingerprints, and the y-axis is the Tanimoto similarity computed between compressed fingerprints. The diagonal line corresponds to equal similarity calculations for the two fingerprints, and points above this line indicate that compression results in overestimated similarity. Shade indicates the number of hash collisions between each pair, with darker colors representing more collisions.

C Complete regression results

This section provides comprehensive regression results across all DOCKSTRING targets, fingerprint dimensions, and hyperparameter settings. All values represent mean ± 1 standard deviation across 10 random trials. Bold indicates the best performance for each target.

C.1 Optimized hyperparameters

Table 4: R^2 scores with optimized hyperparameters (↑)

Target	Exact	Compressed				Sort&Slice			
		512	1024	2048	4096	512	1024	2048	4096
ESR2	**0.553 ± 0.003**	0.515 ± 0.003	0.525 ± 0.003	0.532 ± 0.003	0.540 ± 0.003	0.541 ± 0.003	0.543 ± 0.003	0.543 ± 0.004	0.545 ± 0.004
F2	**0.848 ± 0.002**	0.83 ± 0.002	0.834 ± 0.001	0.838 ± 0.002	0.842 ± 0.002	0.843 ± 0.002	0.845 ± 0.001	0.846 ± 0.001	**0.848 ± 0.002**
KIT	**0.745 ± 0.002**	0.687 ± 0.003	0.703 ± 0.002	0.715 ± 0.002	0.728 ± 0.002	0.716 ± 0.003	0.728 ± 0.003	0.734 ± 0.003	0.739 ± 0.003
PARP1	**0.882 ± 0.001**	0.865 ± 0.001	0.869 ± 0.001	0.872 ± 0.001	0.876 ± 0.001	0.878 ± 0.001	0.879 ± 0.001	0.881 ± 0.001	**0.882 ± 0.001**
PGR	0.470 ± 0.004	0.444 ± 0.004	0.444 ± 0.004	0.449 ± 0.004	0.457 ± 0.004	**0.480 ± 0.003**	0.478 ± 0.002	0.476 ± 0.003	0.474 ± 0.003

Table 5: MSE with optimized hyperparameters (↓)

Target	Exact	Compressed				Sort&Slice			
		512	1024	2048	4096	512	1024	2048	4096
ESR2	**0.348 ± 0.002**	0.378 ± 0.002	0.37 ± 0.002	0.364 ± 0.002	0.358 ± 0.002	0.357 ± 0.002	0.356 ± 0.003	0.355 ± 0.003	0.354 ± 0.003
F2	**0.156 ± 0.002**	0.176 ± 0.002	0.171 ± 0.002	0.167 ± 0.002	0.163 ± 0.002	0.162 ± 0.002	0.16 ± 0.001	0.159 ± 0.001	0.157 ± 0.002
KIT	**0.3 ± 0.003**	0.369 ± 0.003	0.351 ± 0.002	0.336 ± 0.002	0.321 ± 0.003	0.335 ± 0.003	0.321 ± 0.003	0.314 ± 0.003	0.307 ± 0.004
PARP1	**0.185 ± 0.001**	0.213 ± 0.001	0.206 ± 0.001	0.201 ± 0.001	0.195 ± 0.001	0.192 ± 0.001	0.19 ± 0.001	0.187 ± 0.001	0.186 ± 0.001
PGR	0.494 ± 0.003	0.519 ± 0.004	0.519 ± 0.003	0.514 ± 0.003	0.506 ± 0.004	**0.485 ± 0.003**	0.487 ± 0.002	0.489 ± 0.003	0.49 ± 0.003

Table 6: MAE with optimized hyperparameters (↓)

Target	Exact	Compressed				Sort&Slice			
		512	1024	2048	4096	512	1024	2048	4096
ESR2	**0.449 ± 0.002**	0.469 ± 0.001	0.465 ± 0.001	0.461 ± 0.001	0.457 ± 0.002	0.456 ± 0.002	0.455 ± 0.002	0.455 ± 0.002	0.454 ± 0.002
F2	**0.305 ± 0.002**	0.325 ± 0.002	0.32 ± 0.001	0.316 ± 0.002	0.311 ± 0.002	0.311 ± 0.002	0.309 ± 0.001	0.308 ± 0.002	0.306 ± 0.002
KIT	**0.422 ± 0.002**	0.47 ± 0.002	0.458 ± 0.001	0.448 ± 0.001	0.437 ± 0.002	0.446 ± 0.002	0.437 ± 0.002	0.432 ± 0.002	0.427 ± 0.002
PARP1	**0.326 ± 0.001**	0.352 ± 0.001	0.346 ± 0.001	0.341 ± 0.001	0.335 ± 0.001	0.332 ± 0.001	0.33 ± 0.001	0.328 ± 0.001	0.327 ± 0.001
PGR	0.553 ± 0.002	0.568 ± 0.002	0.568 ± 0.002	0.565 ± 0.002	0.561 ± 0.002	**0.545 ± 0.002**	0.547 ± 0.002	0.549 ± 0.003	0.55 ± 0.002

C.2 Fixed hyperparameters

Table 7: R^2 scores with fixed hyperparameters (↑)

Target	Exact	Compressed				Sort&Slice			
		512	1024	2048	4096	512	1024	2048	4096
ESR2	**0.533 ± 0.004**	0.491 ± 0.003	0.498 ± 0.004	0.5 ± 0.003	0.507 ± 0.004	0.49 ± 0.004	0.501 ± 0.005	0.506 ± 0.005	0.511 ± 0.005
F2	**0.832 ± 0.003**	0.815 ± 0.003	0.818 ± 0.003	0.821 ± 0.003	0.824 ± 0.003	0.823 ± 0.003	0.826 ± 0.002	0.827 ± 0.002	0.828 ± 0.003
KIT	**0.732 ± 0.003**	0.675 ± 0.003	0.688 ± 0.002	0.699 ± 0.002	0.712 ± 0.002	0.695 ± 0.003	0.71 ± 0.003	0.719 ± 0.003	0.727 ± 0.003
PARP1	**0.867 ± 0.002**	0.852 ± 0.001	0.855 ± 0.001	0.857 ± 0.001	0.86 ± 0.002	0.863 ± 0.001	0.865 ± 0.002	0.866 ± 0.001	0.866 ± 0.002
PGR	**0.461 ± 0.005**	0.435 ± 0.005	0.434 ± 0.005	0.432 ± 0.005	0.439 ± 0.006	0.452 ± 0.005	0.456 ± 0.004	0.458 ± 0.004	0.459 ± 0.005

Table 8: MSE with fixed hyperparameters (↓)

Target	Exact	Compressed				Sort&Slice			
		512	1024	2048	4096	512	1024	2048	4096
ESR2	**0.363** ± **0.003**	0.397 ± 0.003	0.391 ± 0.003	0.389 ± 0.003	0.384 ± 0.003	0.397 ± 0.003	0.389 ± 0.004	0.385 ± 0.004	0.381 ± 0.004
F2	**0.174** ± **0.003**	0.191 ± 0.003	0.188 ± 0.003	0.185 ± 0.003	0.182 ± 0.003	0.182 ± 0.003	0.18 ± 0.003	0.179 ± 0.003	0.177 ± 0.003
KIT	**0.315** ± **0.003**	0.383 ± 0.004	0.367 ± 0.002	0.354 ± 0.002	0.339 ± 0.003	0.36 ± 0.004	0.342 ± 0.004	0.331 ± 0.004	0.322 ± 0.004
PARP1	**0.21** ± **0.003**	0.233 ± 0.002	0.228 ± 0.002	0.225 ± 0.002	0.22 ± 0.003	0.215 ± 0.002	0.213 ± 0.002	0.211 ± 0.002	**0.21** ± **0.002**
PGR	**0.503** ± **0.004**	0.527 ± 0.005	0.528 ± 0.005	0.529 ± 0.005	0.523 ± 0.005	0.511 ± 0.005	0.507 ± 0.004	0.505 ± 0.004	0.504 ± 0.004

Table 9: MAE with fixed hyperparameters (↓)

Target	Exact	Compressed				Sort&Slice			
		512	1024	2048	4096	512	1024	2048	4096
ESR2	**0.46** ± **0.002**	0.481 ± 0.002	0.478 ± 0.002	0.478 ± 0.002	0.474 ± 0.002	0.482 ± 0.002	0.477 ± 0.003	0.475 ± 0.003	0.472 ± 0.003
F2	**0.319** ± **0.002**	0.336 ± 0.002	0.333 ± 0.002	0.33 ± 0.002	0.327 ± 0.003	0.329 ± 0.002	0.327 ± 0.002	0.325 ± 0.002	0.324 ± 0.002
KIT	**0.432** ± **0.002**	0.478 ± 0.002	0.468 ± 0.002	0.46 ± 0.001	0.449 ± 0.002	0.464 ± 0.002	0.451 ± 0.002	0.444 ± 0.002	0.438 ± 0.003
PARP1	**0.342** ± **0.002**	0.366 ± 0.001	0.36 ± 0.001	0.357 ± 0.001	0.352 ± 0.002	0.351 ± 0.001	0.349 ± 0.001	0.347 ± 0.001	0.346 ± 0.001
PGR	0.556 ± 0.003	0.569 ± 0.003	0.57 ± 0.003	0.57 ± 0.003	0.567 ± 0.003	0.556 ± 0.003	**0.554** ± **0.002**	**0.554** ± **0.002**	**0.554** ± **0.003**

D Code and reproducibility

The custom GP implementation can be found at https://github.com/AustinT/tanimoto-gp. All experiments from the paper can be found at https://github.com/wvirany/molcollisions.