12Representationfigure.1
Draft version September 20, 2024
Typeset using LaTeX default style in AASTeX631

Robust and efficient reranking in crystal structure prediction: a data driven method for real-life molecules

Supporting Information

Andrea Anelli,[1] Hanno Dietrich,[2] Philipp Ectors,[3] Frank Stowasser,[1] Tristan Bereau,[4] Marcus Neumann,[2] and Joost Van den Ende[1]

[1] *Roche Pharma Research and Early Development, Therapeutic Modalities, Roche Innovation Center Basel, F. Hoffmann-La Roche Ltd., Basel, Switzerland*

[2] *Avant-garde Materials Simulation Deutschland GmbH, Merzhausen, Germany*

[3] *Pharma Technical Development, F. Hoffmann-La Roche Ltd., Basel, Switzerland*

[4] *University of Heidelberg, Heidelberg, Germany*

1. MACHINE LEARNING

1.1. *Representation*

The generation of the used crystal structure descriptors is based on the package librascal by Musil et al. (2021) and uses its `spherical_invariants` module, which generates SOAP descriptors for the atomic environments contained in the input molecular configuration. The unit cells of each crystal are extracted from VASP inputs and parsed into ASE (Larsen et al. (2017)) formats. We use as atomic representation the power spectrum centered around each atom in the unit cell. The number of radial basis functions and spherical harmonics channels are selected based on memory constraints to $n_{\max} = 14$ and $l_{\max} = 9$ and a fixed radial cutoff of $r_{\text{cut}} = 4.5$ Å throughout each exercise. This choice is close to the optimal value across each system and is in line with literature-reported best practices in molecular crystals energy models (Anelli et al. (2018); Musil et al. (2018)). We further adopt a radial filter, as introduced by Willatt et al. (2018) to enhance the models' accuracy, with parameters fixed at rate = 1, scale = 3.5, exponent = 2, after performing a grid search optimization on fentanyl data. We employ the radial compression scheme proposed by Goscinski et al. (2021) for compressing radial information computed on the generation structures using $n_{\max} = 14$ and mapping it to $n_{\max} = 5$. For every structure, we average the power spectra across each atomic center to obtain a crystal structure feature vector. The choice of the cutoff turns out to be mostly a hyperparameter choice which can be tuned depending on the training dataset at hand, but in the presence of large molecular crystal structures, it does not have a vital effect on the accuracy of the model and the working principle of the selection algorithm. In Figure S1, we report the learning curves obtained for the fentanyl model. The training set used for the benchmark consists of the whole reranked dataset obtained by the previous iteration of the GRACE algorithm. The structures used for the SOAP calculations correspond to the TMFF optimized configurations - to mimic the minima-to-minima learning scheme proposed in the main manuscript. The energies are directly the DFT energies. The different lines correspond to SOAP descriptors built using a different radial cutoff while keeping $n_{\max} = 14$ and $l_{\max}$=10 fixed. The radial filter is still used, although starting from a 6 Å distance (thus effectively "switched off"). As one can see, the performances of the model do not change dramatically depending on the cutoff choice.

1.2. *Delta learning : from force-field structures to* DFT *minima energies*

The problem setup is such that we possess an accurate and fast-to-evaluate tailor-made force field per molecule, and we use this to screen the crystal energy landscape for the most stable molecular packings. This exercise produces a selection of initial force field minima, called S_{gen} henceforth, from which we have to distill all the configurations that, upon relaxation, lie between a desired energetic window on a more accurate reference potential energy surface (PES)

Corresponding author: Andrea Anelli
andrea.anelli@roche.com

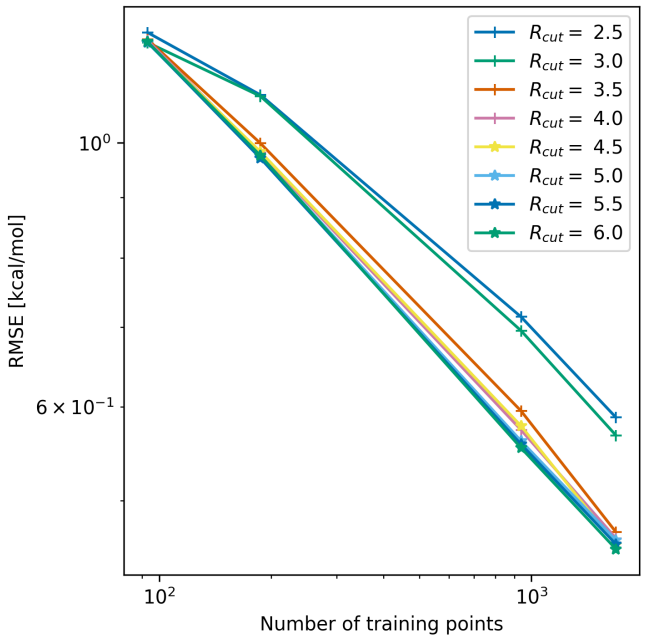

Figure 1. Learning curves for different cutoff radii SOAP descriptors (keeping $n_{\max}$ and $l_{\max}$ fixed to 14) in the fentanyl dataset. The curves are constructed using a ridge regression estimator fitted on a 10-fold cross-validation at each training set size. The performances across the different radii show a diminishing returns behavior at the onset of 4Å, as previously observed in similar molecular crystals systems Musil et al. (2018); Anelli et al. (2018)

(*i.e.* PBE-DFT + Neumann-Perrin dispersion corrections). The objective is thus to construct a predictor that maps the TMFF-optimized structure to the energy it would have once minimized with the reference level of theory (DFT + dispersion correction). To make the exercise more efficient, we resort to a delta learning scheme, where the target energy to be predicted is the energy difference between the *ab initio* optimized configuration and the starting TMFF energies. Our target is thus defined as $E^{\delta}_{\text{atomic}} = (E_{\text{DFT}} - E_{\text{TMFF}})/N_{\text{atoms}}$, where N_{atoms} simply counts the number of atoms in the unit cell. The SOAP descriptors are calculated on the molecular crystal structures contained in the S_{gen}, and create the set of generation structures' descriptors: X_{gen}.

2. REGRESSION MODEL

2.1. *Regression model and training procedure*

The regression model for the energies is based on linear regression with an L^2 norm penalty in the cost function.

$$Y_{\text{test}} = \mathbf{W}\mathbf{X}_{\text{test}} \text{ where } \mathbf{W} = (\mathbf{X}_{\text{train}}\mathbf{X}^T_{\text{train}} - \lambda\mathbf{I})^{-1}Y_{\text{train}} \tag{1}$$

The $\mathbf{I}$ matrix indicates the $N_{\text{train}} \times N_{\text{train}}$ identity matrix. The optimal weights $\mathbf{W}$ are calculated on the training set. They can be then stored to perform a prediction on a new, unseen test point and depend on the choice of the regularization parameter λ. We use the scikit-learn `ridge_CV` class from its `linear_models` to fit the model and store its weights. At each training cycle, we divide our dataset in training (80 %) and test (20 %) sets. Since the data points are ordered according to the order of sampling (*i.e.* chosen by diversity and improvement to the model), we do not reshuffle to train our models, and simply train on the first 80 % and predict on the most recent 20 %. To optimize the choice of the regularization λ we perform a grid search on a log-space spanning $[10^{-10}, 10]$ using 12 sample points (using `np.logspace(-10,1,12)`). The grid search is performed using the library `GridSearchCV` from the scikit-learn library, and it is based on the scikit-learn model class `Ridge`, with a `neg_root_mean_squared_error` loss and a 5-fold cross-validation scheme. To perform this optimization we use the training set available at each step. The best regularization (as the one which minimizes the RMSE across the folds) is then used to train a model on the training set, and thus to produce an estimator of the property to be used on the test set.

2.2. *Uncertainty by resampling*

To produce an uncertainty we adopt the committee uncertainty scheme proposed by Imbalzano et al. (2021). We describe in the following the specific implementation choices that we have followed in designing our resampling estimator for this task. Starting from our training set of dimension N, we subsample n_{res} times a fraction f_{res} of the total data without replacement, constructing an ensemble of different folds of the initial training set. We independently train a ridge regression model on each of these subsets, and obtain as a result a committee of predictors $y^i(X)_{i=1..n_{\mathrm{res}}}$.

To extract the prediction and the uncertainty from our collection of models, we simply calculate the mean and standard deviation of the models' consensus on a new test point:

$$y_{\mathrm{resampling}}(X_{\mathrm{test}}) = \frac{1}{n_{\mathrm{res}}} \sum_{i=1}^{n_{\mathrm{res}}} y^i(X_{\mathrm{test}}) \tag{2}$$

$$\sigma^2_{\mathrm{resampling}}(X_{\mathrm{test}}) = \frac{1}{n_{\mathrm{res}}-1} \sum_{i=1}^{n_{\mathrm{res}}} [y^i(X_{\mathrm{test}}) - y_{\mathrm{resampling}}(X_{\mathrm{test}})]^2 \tag{3}$$

The limit of subsampling is reached when one distorts the sample size statistics, thus obtaining a conservative measure of the uncertainty. This happens as the variance across the models is artificially compressed by the reduction of the property space they are trained on. To compensate for this uncertainty "bias", we adopt the un-biasing operation suggested in Imbalzano et al. (2021) and calibrate the models such that the uncertainty is expanded to match the error it incurs when predicting over a separate calibration set. To do this, we use the test set as a calibration set (while normally data leaks from test to training should be avoided at all costs, using the test data to calibrate the uncertainty model is necessary in the presence of extremely low data regime - a scenario in which this algorithm mostly operates) and calculate an expansion factor $\alpha_{\mathrm{resampling}}$ such that our uncertainties match an average error observed over the test set:

$$\alpha^2_{\mathrm{resampling}} = -\frac{1}{n_{\mathrm{res}}} + \frac{n_{\mathrm{res}}-3}{n_{\mathrm{res}}-1} \frac{1}{N_{\mathrm{test}}} \sum_{i=1}^{N_{\mathrm{test}}} \frac{[y_{\mathrm{ref}}(X_i) - y_{\mathrm{resampling}}(X_i)]^2}{\sigma^2_{\mathrm{resampling}}(X_i)} \tag{4}$$

The final unbiased resampling estimator is built so to reflect this expanded spread while not changing its mean as follows :

$$y^i(X_{\mathrm{test}}) = y_{\mathrm{resampling}}(X_{\mathrm{test}}) + \alpha_{\mathrm{resampling}}[y^i(X_{\mathrm{test}}) - y_{\mathrm{resampling}}(X_{\mathrm{test}})] \tag{5}$$

A crucial number in the resampling approach is the choice of the number of resampling models. As the reference Imbalzano et al. (2021) shows, the uncertainties converge to an established value with the number of resampling models chosen. Across the whole exercise, we use $n_{\mathrm{res}} = 256$ regressors for each model.

3. ACTIVE LEARNING ALGORITHM

3.1. *Problem statement*

Having prepared a regression scheme for predicting the energies of a target molecular crystal structure, we want to proceed to use it to guide the selection of which crystal structure to select for a costly energy evaluation through full geometry optimization. The objective of the prediction exercise is to ensure sampling of all the structures $X \in X_{\mathrm{gen}}$ such that their relative lattice energies $Y - Y_{\mathrm{GS}} < E_{\mathrm{window}}$, where Y_{GS} corresponds to the lowest lattice energy structure contained in the generation pool (*i.e.* the putative ground state) and E_{window} corresponds to the energy window we are willing to sample completely. Trivially, if $E_{\mathrm{window}} = 0$, this exercise becomes a relatively simpler global minima search. Increasing E_{window} the goal of the exercises shifts to finding all the structures whose energy is contained within the desired range. We decide to follow an iterative scheme, where we sample N_{batch} structures at every iteration, and then use this batch to extend our training set, obtaining an increasingly more accurate or robust model as we continue through the iterations. The exercise we want to perform acts on two parallel axes: on the one hand we must ensure a complete sampling of all the 'low enough' (*i.e.* within a desired cutoff) energy structures present in the dataset, on the other hand, we want to have an increasingly accurate machine learning model so that we can trust its prediction at each sampling iteration. In an ideal scenario, a perfect prediction model will point us with 100 % confidence to which structures to optimize to ensure complete sampling below the desired energy window in one single round sampling with a batch size corresponding to the total number of expected low energy structures. As the model has an average mean squared error RMSE_i at each sampling iteration i, we account for this uncertainty by introducing a probabilistic framework to decide which structure to select. To make sure that our model improves with each iteration, and at the same time prioritizes selections of low energy configurations, we adopt an active learning approach that combines

exploration (maximizing the robustness of the model to different crystalline packings) and exploitation (sampling low energy candidates when confident enough). The pool of structures we select from is defined at the beginning of the exercise and depends on the size of the generation set, which is a function of the complexity of the crystal structure landscape, and the energy window we want to sample. The choice of the generation set is independent of this approach and is detailed in Section 5. From this exercise perspective, we will assume to have a finite-sized pool of configurations we can sample from, called X_{gen}. With each iteration of sampling, the pool to sample from reduces in size due to the removal of newly sampled structures.

3.2. *Initializing the algorithm*

In the absence of samples from the *ab initio* potential surface, we do not possess any prior knowledge except for the force-field based ranking we start from. To initialize the construction of our model - and at the same time have an unbiased sampling of the generation pool - we choose the first batch of N_{init} candidates based on structural diversity. We choose the first number of structures that are the most distinct from the others in order to prepare the machine learning model to deal with the potentially high diversity present in the crystal structure landscape of the target molecule. To do this, we follow a farthest point sampling approach (FPS). The deterministic FPS method used in this work requires solely the knowledge of the feature vector of each of the generation pool crystal structures: X_{gen}. To select the most diverse configurations contained in such a set, it is sufficient to ensure a uniform sampling of the dataset, by selecting each new point so as to maximize a reference distance metric between all of them. In our case, we adopt a simple Euclidean distance between points and initialize the first selection starting from the lowest TMFF energy structure. The next index selected will be the i such that:

$$i \;=\; \arg\max(\min_{j} \|X_i - X_j\|^2) \tag{6}$$

The index j runs over all the already selected points. We repeat this exercise until we have sampled N_{init} structures.

3.3. *The Expected Improvement function*

Once we have optimized the first N_{init} structures, we can train a model using the first 80 % of them as training data, and the remaining as a test/calibration set. We thus have a model $M_0(X^0_{\text{train}}, y^0_{\text{train}})$ (where the superscript index corresponds to the iteration number), which outputs for each new test point X_{test} its predicted lattice energy value and the uncertainty associated with it: $\{\mu_{\text{test}}, \sigma_{\text{test}}\}$.

To find the next suitable candidate for the minimization, we will want to consider a structure whose prediction either confidently points to low energy (a so-called exploitation-driven choice) or to a potentially low energy configuration exhibiting very high uncertainty (an exploration-driven choice) - so as to stabilize the model. To quantify how "good" each structure is, we repurpose a Bayesian optimization acquisition function Mohr et al. (2022) as our Expected Improvement (EI) function which provides an easily tunable tradeoff between exploration and exploitation. The EI function associates each unlabelled point's prediction tuple (*i.e.* the pair containing the predicted target and its uncertainty) with a positive real value which is maximized either by showing low energy (within a desired window) or by having a "low enough" energy with a high uncertainty. The EI is defined in the following way:

$$\text{EI}(X_{\text{test}}) = \begin{cases} (-\mu(X_{\text{test}}) + E_{\text{best}} + E_{\text{window}})\Phi(Z) + \sigma(X_{\text{test}})\phi(Z) & \text{if } \sigma(X_{\text{test}}) > 0 \\ 0 & \text{if } \sigma(X_{\text{test}}) = 0 \end{cases} \tag{7}$$

With Z representing a ratio between the potential for stability and the uncertainty in the prediction, defined as follows:

$$\text{Z} = \begin{cases} \frac{-\mu(X_{\text{test}}) + E_{\text{best}} + E_{\text{window}}}{\sigma(X_{\text{test}})} & \text{if } \sigma(X_{\text{test}}) > 0 \\ 0 & \text{if } \sigma(X_{\text{test}}) = 0 \end{cases} \tag{8}$$

Where $\Phi(Z)$ is the cumulative distribution function and $\phi(Z)$ the probability density function of a normal distribution, E_{best} the lowest lattice energy found so far and E_{window} serves as an offset term to tune between exploration and exploitation behavior. While in the original paper Mohr et al. (2022) this term is suggested to be fine-tuned to obtain an optimal balance between the two regimes of sampling, in this work we use it to control the extension of the exploitative sampling by making it coincide with the target energy window. The advantage of this choice is that the acquisition function has no parametric dependence on the offset term and allows a simple interpretation of its values.

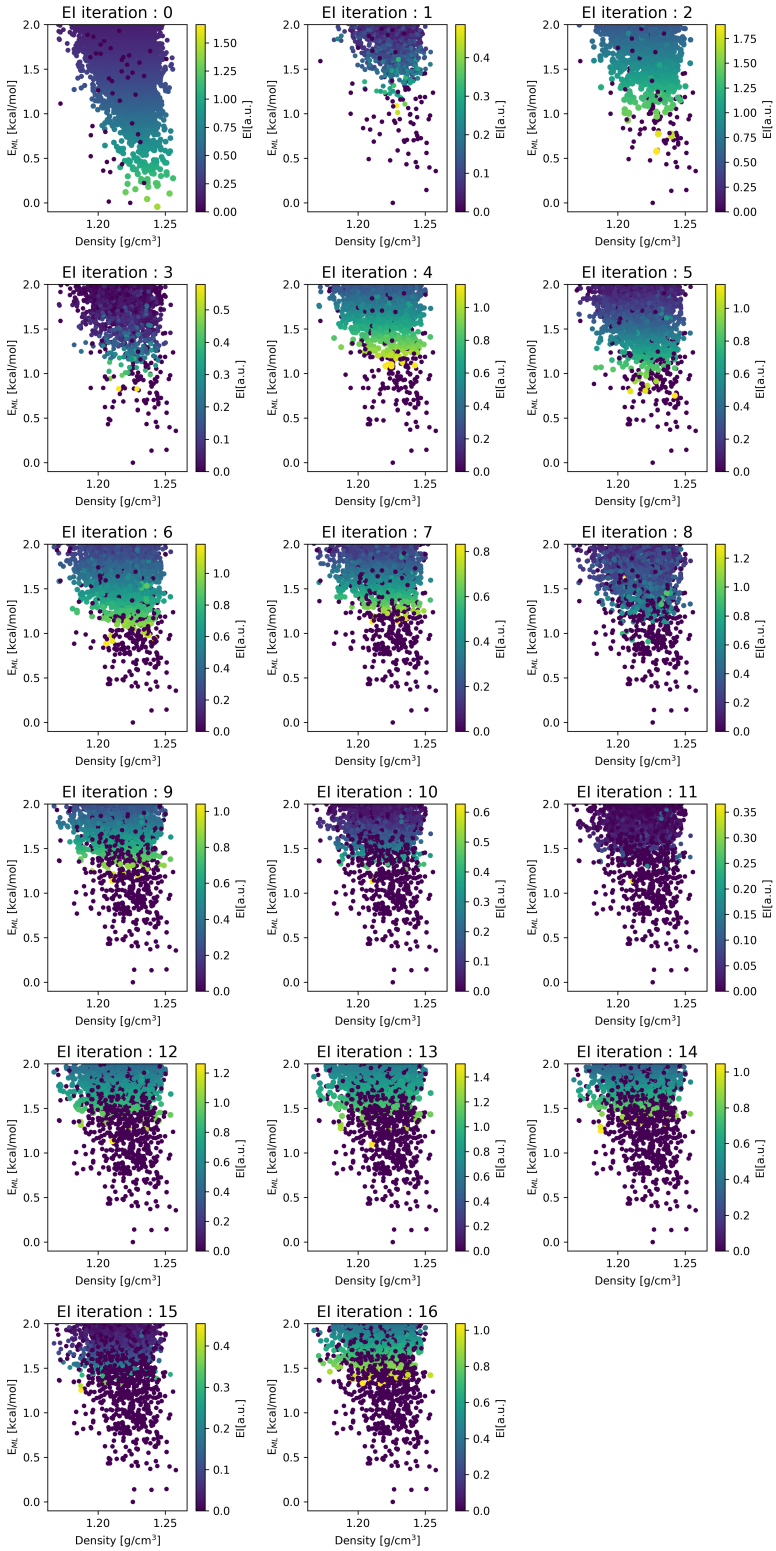

Figure 2. Values of the Expected improvement score on each of the crystal structures evaluated at the different iterations in the fentanyl exercise. The EI values reach zero when a structure has been sampled or when it has high confidence in being unstable. As the figures show, the scores follow an intuitive behavior, highlighting points of high EI values as the ones that are close to the bottom of the landscape. As the bottom of the landscape is completely sampled, the EI values get closer to the energy window threshold.

The approach proposed so far allows to assign to each unlabelled point an EI value, as shown in Figure 2. The selection of the best sample then requires just to choose the $x_{\text{best}} \in X_{\text{gen}}$ such that $x_{\text{best}} = \arg\max_i(\text{EI}(x_i); x_i \notin \{x\}_{\text{selected}})$. Whereas in theory, this method is perfectly suitable for the problem at hand, for practical reasons, it can be preferable to select a batch of candidates at once, before summoning the *ab initio* minimization routine. To address this issue, we adopt the approach proposed by Mohr et al. (2022) and construct a batch selection routine based on the Kriging-Believer method. If we consider our pool of selected candidates as $\{x\}_{\text{selected}} = \{x_0, x_1, x_2, ..., x_M\}$ at any iteration step, we select the next configuration x_{M+1} by maximization of the $x_{M+1} = \arg\max_i(\text{EI}(x_i); x_i \notin \{x\}_{\text{selected}})$. To continue selecting a new candidate, we extend the selected pool with the previously chosen x_{M+1} structure. We assign to this structure energy its predicted energy and treat it as if it were actually sampled. A new temporary selected pool is formed and a new model is trained on it. Crucially, the new selected point must be included in the training set of such a model. We then proceed to use this new temporary model to estimate the EI values for the remnant of the generation pool and select $x_{M+2} = \arg\max_i(\text{EI}(x_i); x_i \notin \{\{x\}_{\text{selected}}), x_{M+1}\}$. We repeat the previous operation to continue selecting new points until we have collected N_{batch} new indices to sample. Throughout our work we use $N_{\text{batch}} = 100$ in the minimizations offentanyl, and $N_{\text{batch}} = 50$ for the `Roche_API` dataset.

4. MACHINE LEARNING GUIDED RERANKING

The model M^i at iteration i associates to each crystal structure descriptor X an energy probability density function $p_{X_i}(E)$. By construction, it is a normal distribution centered around the mean predicted energy and with standard deviation the uncertainty in such prediction so that it can be written in the form:

$$p_{X_i} = \frac{1}{\sqrt{2\pi\sigma^2(X_i)}} \exp\left\{-\frac{(E-\mu(X_i))^2}{2\sigma^2(X_i)}\right\} \tag{9}$$

To produce an estimate of the distribution of energy values across the generation pool, we calculate the generation set energy probability density function by summing each generation crystal structure energy probability distribution : $D_{\text{total}}(E) = \sum_{i=1}^{N_{\text{gen}}} p_{X_i}(E)$ Such probability measures the distribution of the energy values within the generation pool. As the model gets more accurate, the uncertainties shrink and the energy values distribution are more precisely localized. The advantage of having such a distribution is that we can use it to measure the number of configurations that have energy lying within a desired energy range. To do this, it is natural to resort to the associated cumulative distribution function. By calculating its cumulative distribution function:

$$F_{\text{total}}(E) = P(D_{\text{total}}(E) < e) = \int_{\infty}^{e} D_{\text{total}}(e)\,\mathrm{d}e \tag{10}$$

$$\hat{F}_{\text{total}}(E) = \frac{F_{\text{total}}(E)}{\max(F_{\text{total}}(E))} \tag{11}$$

We have an increasing function that measures how likely it is to find a structure in the generation set with energy lower than an energy value E. This is exactly what we need to count the structures that would fall below our desired energy window. To calculate how many structures are expected (N_{expected}) to be present within the preset energy window (E_{window}), we compute the integral of the generation set CDF function from the lowest energy value found (offset to zero at each iteration by setting the lowest energy found so far as the reference) to E_{window}. This quantity, multiplied by the total number of structures in the set, measures what is the chance of having left a structure within E_{window} above the ground state. In parallel, we keep counting the number of structures sampled within the window as N_{measured}. The convergence probability is defined by $p_{\text{conv}} = \frac{N_{\text{measured}}}{N_{\text{expected}}}$, and once it reaches a value above p_{conv} (in these exercises set to 0.99) we consider the sampling exhaustive. Note how, by construction, such an estimator is non-monotonically tending to 1, and as a result, it depends on the number of iterations and, indirectly, on the absolute number of structures to be sampled. This trend is easily noticeable on the `Roche_API` set, where one can observe easier chances of missing a structure in the presence of very crowded energy ranges.

5. GRACE WORKFLOW

5.1. *Tailor made force fields*

A TMFF is fitted for each molecule from scratch and aims at modelling the molecular interactions in the solid. It includes bond-increment-derived atomic point charges, isotropic van der Waals interactions (modeled by the Lennard-Jones 9-6 potential), and anharmonic terms for intramolecular bonds, angles, inversions, and torsions. Van der Waals parameters are precisely fitted, including off-diagonal interactions. Electrostatic interactions are enhanced using dipole and quadrupole moments within local atomic coordinate systems. The details of the fitting operations involved in a tailor-made force field routine are reported in Ref. Neumann (2008). In ourfentanyl exercise the accuracy of the TMFF reported in the typical GRACE units is expressed in energy error per atom with σ and equal to 0.0693 kcal/mol/atom, which corresponds to a molecule error of $\sigma_{\mathrm{mol}} = \sqrt{N_{\mathrm{molecule}}}\,\sigma \sim 0.5$ kcal/mol per molecule.

5.2. *Crystal Structure Prediction*

To prevent errors in force field calculations, CSP uses a careful multi-step process. First, it generates a broad range of crystal structures using the TMFF force field. Then, it selects a subset of structures for training and testing using PBE-Neumann-Perrin energy minimizations. The training set is used in a 'backfitting' process to refine the force field parameters. Accuracy is evaluated against the test set before and after to ensure improvement. If needed, the backfitting is repeated. GRACE main operations are divided into three steps:

- Crystal Structure Generation: Monte Carlo parallel tempering using the freshly calculated TMFF to generate diverse crystal structures within a target energy range.
- Re-ranking: Structures are re-ranked with *ab initio* calculations. Statistical correlations help reduce the need for full optimizations on every structure.
- Final Optimization: Structures within the target energy range are fully optimized with a higher accuracy (or convergence setting) *ab initio* method for structural and energetic refinement.

A more detailed explanation of the GRACE's crystal structure prediction algorithm can be found at Ref. Mortazavi et al. (2019) in the supplementary information.
The generation process is carried out for each separate Z' value until a generation convergence threshold has been reached. In all our exercises we have used 0.99 for $Z' = 1$ and 0.95 for $Z' = 2$. The same principle is followed to perform the reranking, which continues until the calculated reranking convergence variable reaches the threshold. Throughout our calculations, we have used the same convergence criteria for the reranking as well and sampled the $Z' = 2$ space in two sequential rounds. The outcome of a completed CSP consists of several information and datasets, among which a collection of the output of the (1) generated TMFF structure (which we call 'generation set'), (2) the reranked dataset, which contains a subset of the generation which have undergone a minimization through DFT (which we refer to when we compare our reranking landscapes in the main manuscript) and lastly (3) a refined reranked dataset, with structures further relaxed at higher level of theory. In our case, since we mostly focused on speeding up reranking, we only refer to the first and second datasets, not discussing the refined data. The observed convergences in the fentanyl cases are represented in the following table:

Step	Generation	Reranking
$Z' = 1$	0.9902	0.992
$Z' = 2$ Part 1	0.9501	0.983
$Z' = 2$ Part 2	0.9290	0.968

Table 1. Convergence values for fentanyl crystal structure prediction

As it can be observed in Table 1, the generation convergence of the second portion of the $Z' = 2$ space did not reach the desired completion but was stopped due to having reached the timeout wall-time allowed for the calculation. Nevertheless, since we are comparing reranking approaches, this difference does not affect the outcome of the main results.

6. DETAILS ON RESULTS

6.1. *Fentanyl*

The first system under study is the fentanyl molecule. This is a challenging example as the landscape produced with `GRACE 2.7.49` is very dense, even within the target energy window of 1 kcal/mol. In Figure 4 we report the result of the reranked landscape. The difficulty in this example lies in the limited spread in energy for the configurations found to be stable, with 211 structures found lying between 0.0 and 1.15 kcal/mol, and with a large amount (109) close to the energy window border 0.9 and 1.15 kcal/mol. This makes the exercise particularly challenging from the machine learning perspective, as high accuracy is needed from the very beginning to discriminate with sufficient confidence structures within the energy window. The ML-reranker algorithm selects roughly 100 structures per round (the imprecise number is dependent on the outcome of the *ab initio* minimization, as some configurations can become unstable upon minimization, and these are then removed from the sampling). The trend of the residual probability, as well as the sampling convergence probability is non-monotonic can be seen in Figure 3. Monotonicity could be enforced

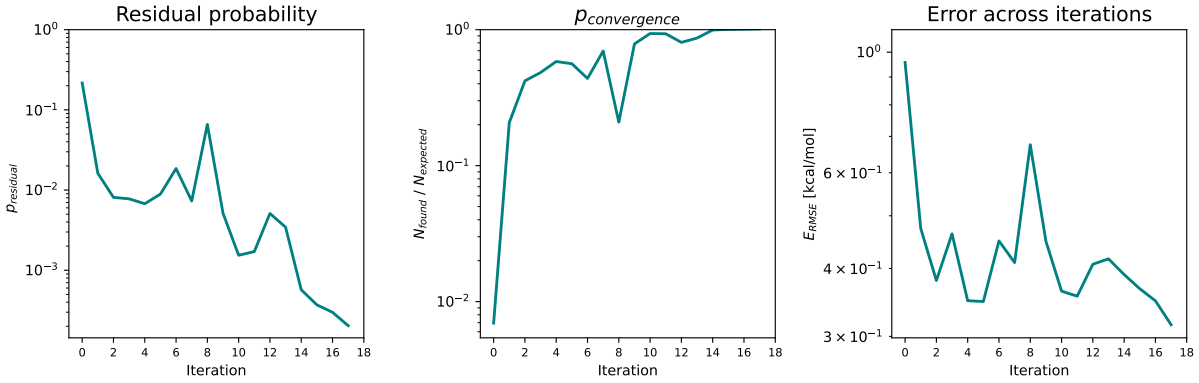

Figure 3. Left-hand side: logarithmic decay of the residual probability of finding a structure within the energy window over the different iterations. Center: calculated convergence probability across the iterations. Right-hand side: the trend of the RMSE calculated in the test sets at each iteration. It is a notable feature of this approach how the model confidence impacts heavily the estimates of the probability of convergence, as seen in the dip in iterations 8 and 12.

by carrying a weight from the selection round happening before, but we observed that keeping a momentum parameter hinders the speed of convergence. As a result, as shown in the figure 3 above one can appreciate the convergence trends across the sampling round.

Since the geometry minimization criteria between GRACE and the ML-reranker are different, the resulting optimized landscape has non-substantial differences, as can be seen in the figure below on the left. An important aspect to keep in mind is that, while GRACE includes a last step of duplicate removal, our method does not remove similar configurations - thus creating a fictitiously more populated landscape. The moment we perform a comparison based on structural similarity with GRACE, we observe limited differences between structures sampled by GRACE and our approach. In practice, only one configuration is not found by our proposed reranking, as shown in the main text, which is in line with our convergence of 0.99.

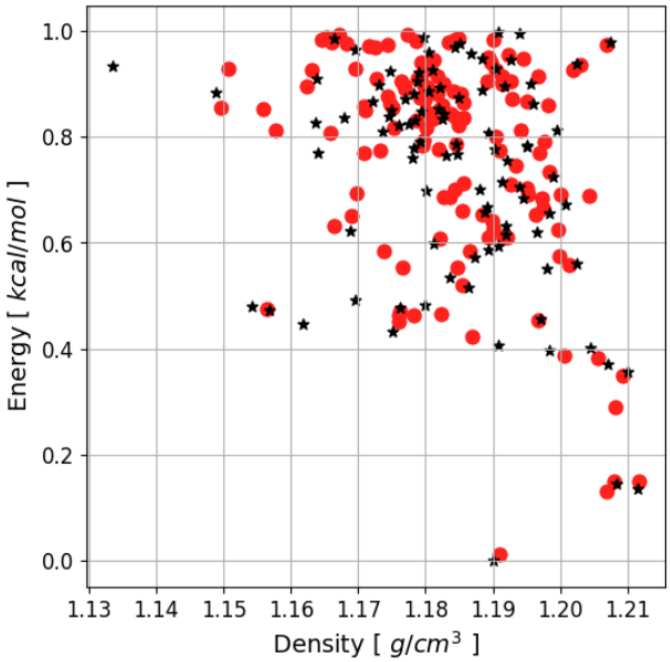

Figure 4. Crystal structure energy-density plot for the fentanyl reranking exercise, plotted only in the range below the energy window. The circles in red correspond to structures found by our ML-reranker while the black stars correspond to the landscape points found by GRACE.

6.2. *Roche proprietary compounds*

The dataset constructed using Roche proprietary data is built by collecting the most recent, large enough (> 1k reranked structures), CSP exercises we have run in-house, totaling 17 entries. Each entry corresponds to a different API candidate from the development pipeline, providing an extremely challenging dataset from both the chemistry (*e.g.* differences in chemical species) and physical behavior (more or less pronounced polymorphism). A histogram highlighting the molecular features of these 17 compounds is provided in Figure 5. The performance of the prediction of the approach across the whole dataset is shown in the manuscript in Table 1. As stated in the main text, this does not constitute a direct comparison of the two methods, as the reranking exercise performed in this exercise is radically different. It is worth noting that, in general, the efficiency of the algorithm seems to appear whenever the landscape is large enough (>1k structures to rerank). Moreover, given the wide diversity of the structures contained in this set, the roughly comparable performances across them indicate a method that is robust to the chemical structure of the API.

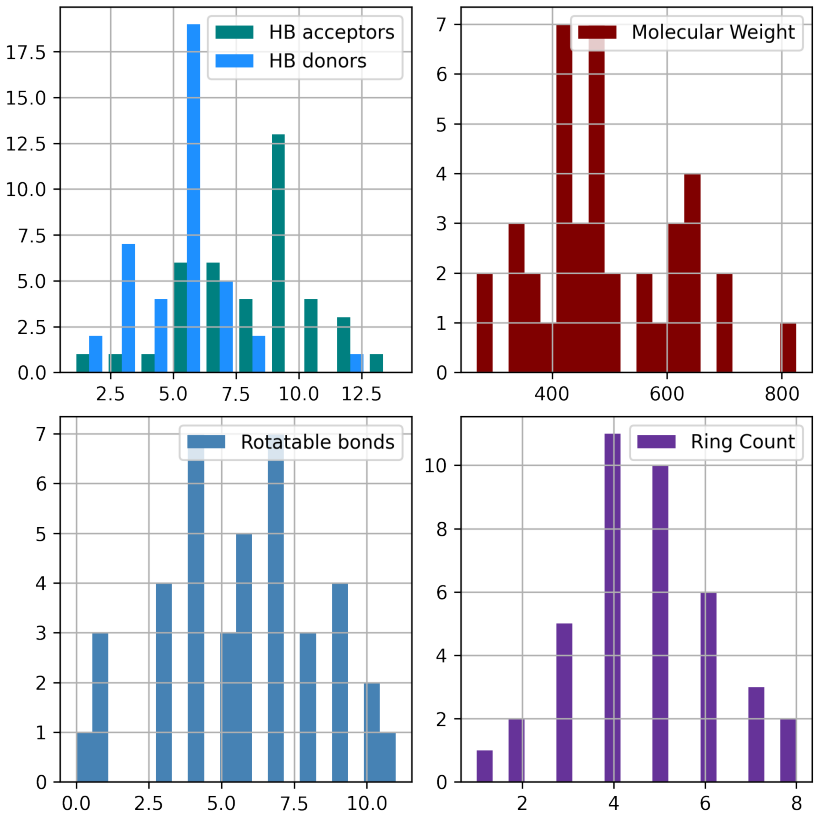

Figure 5. Four histograms representing the distribution of (top-left) Number of hydrogen bonding donors and acceptors, (top-right) molecular weight in g/mol, (bottom-right) number of rings in the compound (bottom-left) number of rotatable bonds. The distribution follows the well-known trend of growing chemical complexity in small molecule drug discovery, with a significant amount of molecules getting closer to the beyond rule of five boundaries (> 500 g/mol molecular weight), exhibiting complex hydrogen bonding capacities and structural flexibilities.

REFERENCES

Anelli, A., Engel, E. A., Pickard, C. J., & Ceriotti, M. 2018, Physical Review Materials, 2, 103804

Goscinski, A., Musil, F., Pozdnyakov, S., Nigam, J., & Ceriotti, M. 2021, The Journal of Chemical Physics, 155

Imbalzano, G., et al. 2021, The Journal of chemical physics, 154, 074102, doi: 10.1063/5.0036522

Larsen, A. H., Mortensen, J. J., Blomqvist, J., et al. 2017, Journal of Physics: Condensed Matter, 29, 273002

Mohr, B., Shmilovich, K., Kleinwächter, I. S., et al. 2022, Chemical Science, 13, 4498

Mortazavi, M., Hoja, J., Aerts, L., et al. 2019, Communications Chemistry, 2

Musil, F., De, S., Yang, J., et al. 2018, Chemical science, 9, 1289

Musil, F., Veit, M., Goscinski, A., et al. 2021, The Journal of Chemical Physics, 154

Neumann, M. A. 2008, The Journal of Physical Chemistry B, 112, 9810

Willatt, M. J., Musil, F., & Ceriotti, M. 2018, Physical Chemistry Chemical Physics, 20, 29661