

Figure 1. The core Inspyra loop. The Inference Engine shows generated compounds to the user and learns their preferences from their responses. This is then passed to the AI, to make new, optimised, compounds.

Pairing AI with human expertise

We present a novel AI drug generation system, designed to include human oversight as a core part of the generative process. Rather than random compound generation, or generation with a predefined goal, we use a unique Inference Engine algorithm to quickly learn a user's parameter preferences, then pass a vector of these to the generative AI.

This coupling of a powerful AI model to human intuition and understanding enables precisely targeted compound generation with on-the-fly user defined objectives, and benefits from human expert knowledge and experience, avoiding costly retraining and greatly reducing the time wasted on generating and filtering out non-viable compounds.

Generative AI

We use a transformer-based AI¹, trained on pairs of related compounds encoded as Selfies strings², and parameter vectors. Given a starting compound and a set of parameter differences, the generative model outputs a closely related compound with parameters closely matching the desired changes.

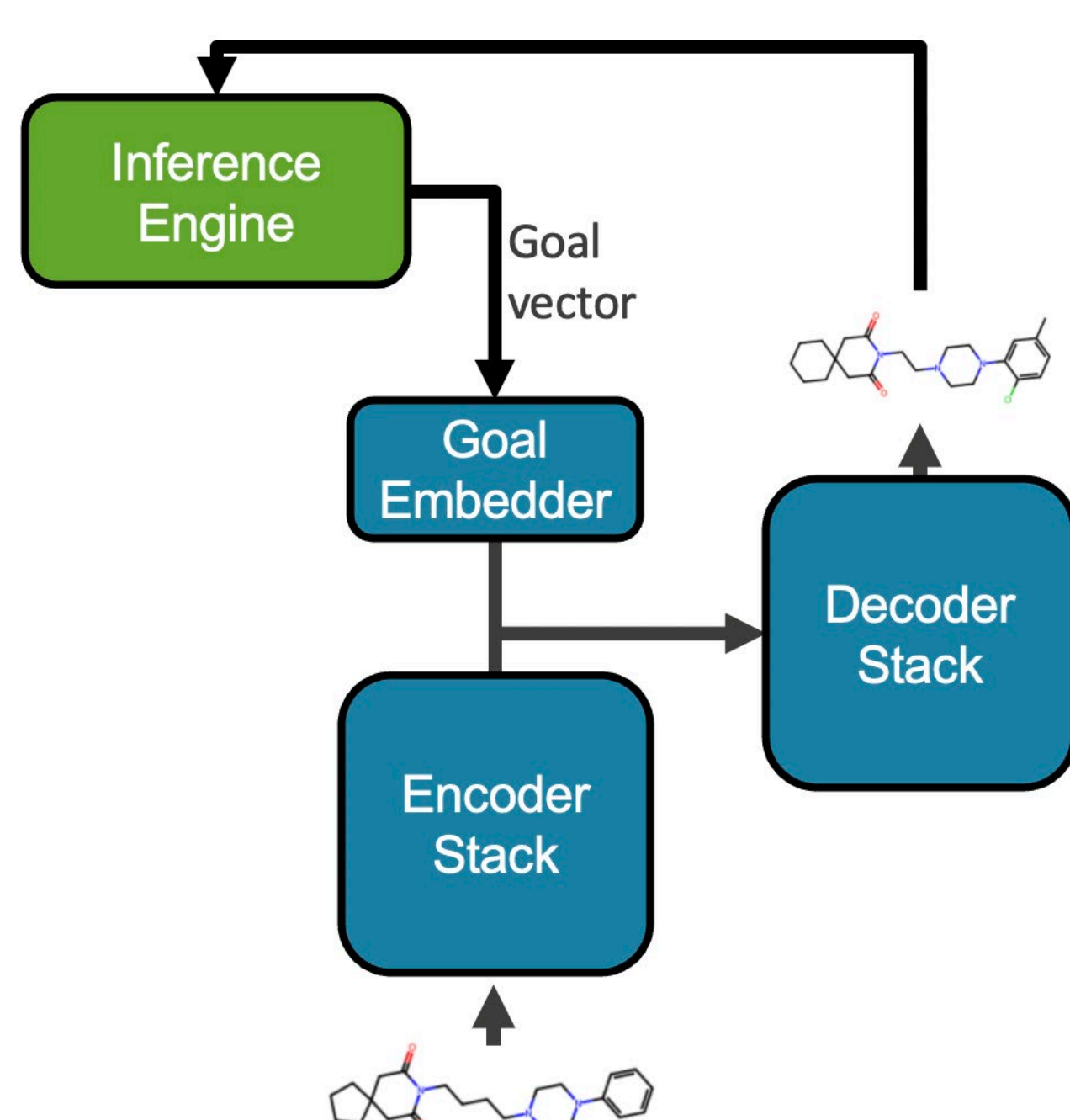


Figure 2. Transformer model architecture. We use a standard transformer with an additional goal embedding block, which converts a parameter goal vector to the higher dimensionality of the encoded starting compound.

We modify the standard transformer architecture slightly to allow seamless input of a vector of desired parameter changes, referred to as the goal vector, without conversion to tokens. The goal vector is output from the inference engine, and then passed through one or more fully-connected neural network layers to upscale the vector to match the internal dimensionality of the generative AI. This embedded goal vector is then concatenated on to the front of the encoded compound sequence.

Vector goals vs token goals

The more conventional method for passing a goal to a transformer model is by converting numerical parameter goals to a discrete set of tokens³, which are then appended to the compound string. However, this means that the model has to effectively re-learn the numerical meaning of each token, the number of tokens in the model vocabulary must greatly increase (and some will appear only rarely in the training set), and the total length of the string must increase with the number of parameters.

By instead keeping our desired parameter changes as a vector, we can completely remove the need to the model to re-learn the numerical values. This simplifies and stabilises the training process, resulting in equivalent performance with a lower training cost, and enables us to optimise for many more parameters simultaneously. Additionally, by appending the embedded goal vector after encoding the starting compound, we preserve a unique compound embedding which can be used for other modelling purposes.

Model performance

Our goal driven model gives state of the art molecular optimisation performance, successfully optimising for multiple parameters simultaneously. Fig. 3 shows the optimisation results for 6 QSAR parameters, plotting the generated compound properties vs target properties. The shaded region shows the uncertainty on the QSAR model estimate of each parameter.

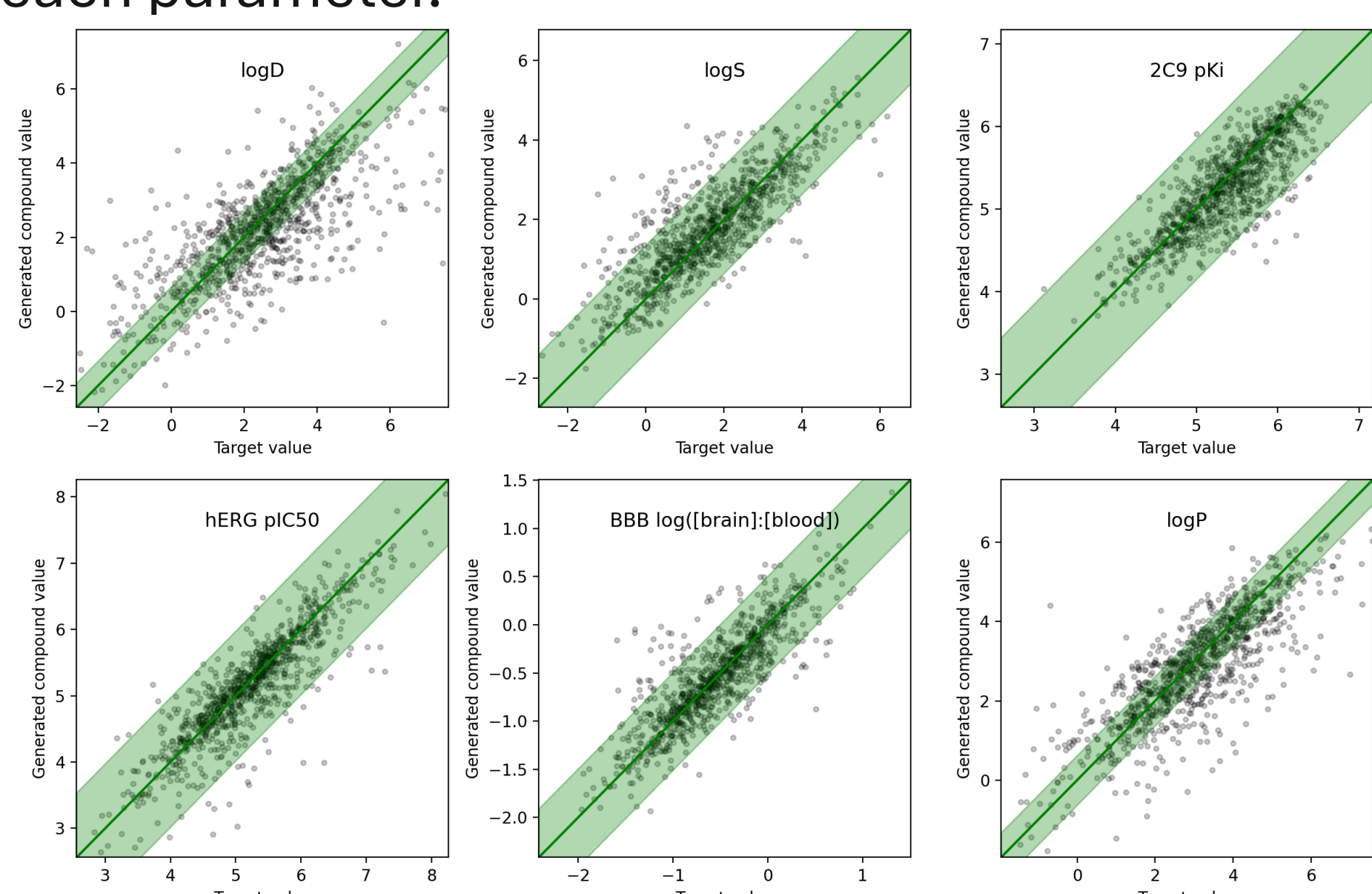


Figure 3. Generated compound property values as a function of target property value. Shaded regions show the StarDrop model uncertainty on each parameter.

Inference engine

The aim of the inference engine model is to learn, within a small number of interactions (typically around 10), an approximation of the user's desired compound properties within a complex multi-dimensional parameter space.

The model probabilistically selects compounds, based on an internal estimate of the probability of a positive response, to show to the user. The user can then respond positively or negatively, and that response is used to refine the model's likelihood estimates.

Our unique algorithm evaluates the probability of a positive user response based on a combination of local information (how has the user responded to similar compounds) and global information (what combinations of parameters does the user like) to provide a detailed multi-dimensional probability map.

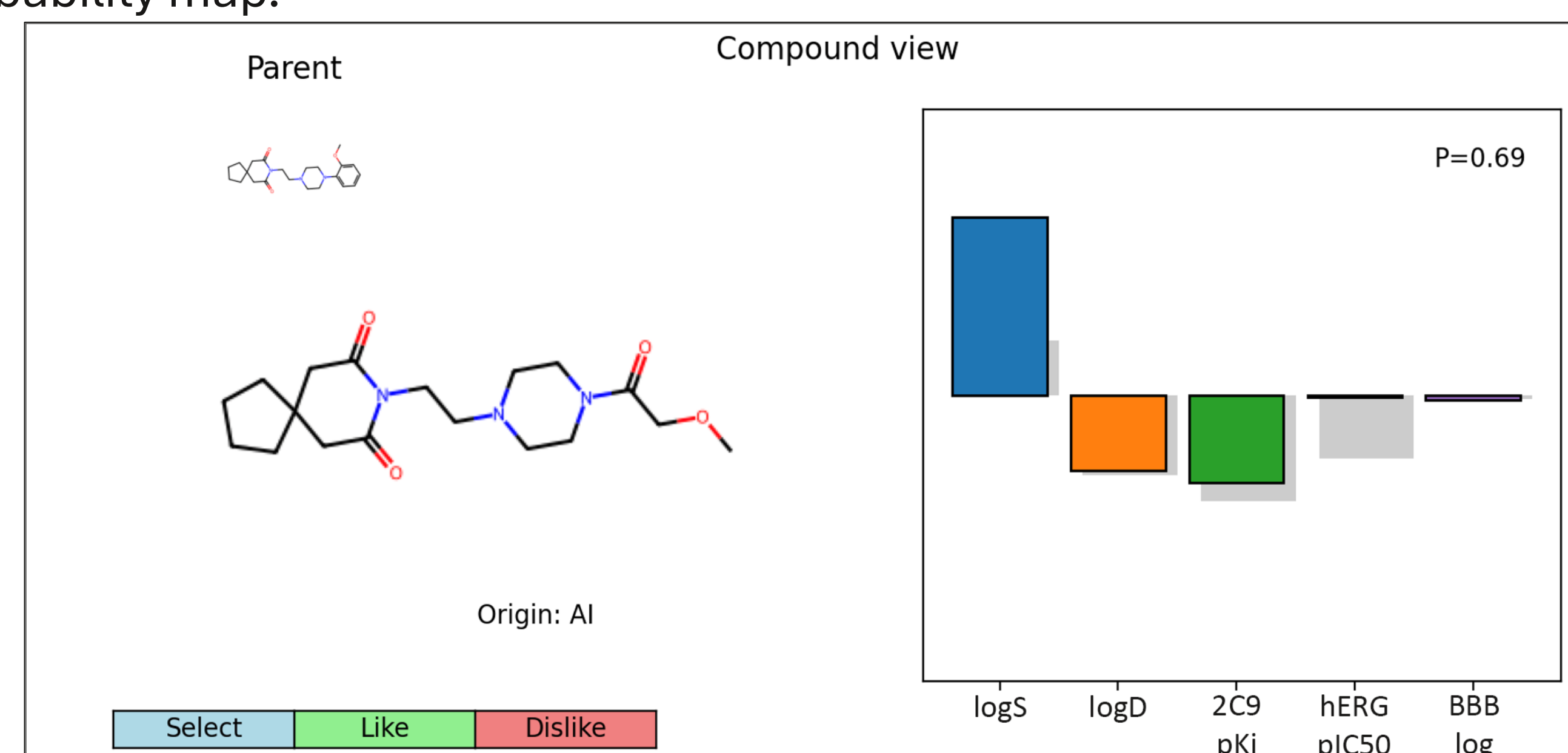


Figure 4. Basic interface for the Inference Engine. The user is presented with an AI generated compound, generated from a parent compound (either user supplied or a previous AI generation). The user may then respond positively or negatively, and may optionally add the compound to their own dataset. The plot on the right shows the parameter profile of the compound, with the grey shaded bars showing the profile of the parent.

Filtering AI compounds

After a few iterations, the inference engine converges on a reasonable approximation of a user's parameter and chemical space preferences. This can be used to select new compounds from the pool of AI generated compounds to show to the user, ensuring that they only see compounds relevant to their goals. This is particularly useful when it comes to parameters that the AI model has not yet been trained on, and effectively does not know about. The inference engine can be used to optimise for these parameters without needing to retrain the AI.

Informing AI generation

The same probabilistic compound selection can be used to choose the parent compounds for the next generation of compounds. Similarly, the global information (e.g. "the user wants compounds with high logS") is used to set the goal vector sent to the AI model, guiding the compound generation towards those values. The flexibility inherent in this approach allows users to rapidly explore compounds with improved parameter profiles around their starting compounds, avoiding costly retraining.

References

- [1] Vaswani et al., 2017, In *Proceedings of the 31st International Conference on Neural Information Processing Systems*
- [2] Mario Krenn et al., 2020, *Mach. Learn.: Sci. Technol.* 1 045024
- [3] Irwin, R et al., 2021, *Chemistry. Mach. Learn. Sci. Technol.*



optibrium.com/
inspyra