Leveraging expert feedback to align proxy and ground truth rewards in goal-oriented molecular generation

Julien Martinelli[*]
Aalto University
Espoo, Finland
julien.martinelli@aalto.fi

Yasmine Nahal[*]
Aalto University
Espoo, Finland
yasmine.nahal@aalto.fi

Duong Lê
Aalto University
Espoo, Finland
duong.le@aalto.fi

Ola Engkvist
Molecular AI, Discovery Sciences, R&D
AstraZeneca
Gothenburg, Sweden
Chalmers University of Technology
Gothenburg, Sweden
ola.engkvist@astrazeneca.com

Samuel Kaski
Aalto University
Espoo, Finland
University of Manchester
Manchester, United Kingdom
samuel.kaski@aalto.fi

Abstract

Reinforcement learning has proven useful for *de novo* molecular design. Leveraging a reward function associated with a given design task allows for efficiently exploring the chemical space, thus producing relevant candidates. Nevertheless, while tasks involving optimization of drug-likeness properties such as LogP or molecular weight do enjoy a tractable and cheap-to-evaluate reward definition, more realistic objectives such as bioactivity or binding affinity do not. For such tasks, the ground truth reward is prohibitively expensive to compute and cannot be done inside a molecule generation loop, thus it is usually taken as the output of a statistical model. Such a model will act as a faulty reward signal when taken out-of-training distribution, which typically happens when exploring the chemical space, thus leading to molecules judged promising by the system, but which do not align with reality. We investigate this alignment problem through the lens of Human-In-The-Loop ML and propose a combination of two reward models independently trained on experimental data and expert feedback, with a gating process that decides which model output will be used as a reward for a given candidate. This combined system can be fine-tuned as expert feedback is acquired throughout the molecular design process, using several active learning criteria that we evaluate. In this active learning regime, our combined model demonstrates an improvement over the vanilla setting, even for noisy expert feedback.

1 Introduction

De novo drug design aims at identifying novel compounds that achieve a high level of desirability concerning given properties. Recent advances in deep generative modeling have caused a surge in research around this field, with the promise to reduce both experimental costs and time spent searching for relevant candidates. In that respect, several molecular generation methods were developed, either based on zero-shot generation (Gómez-Bombarelli *et al.*, 2018; Jin *et al.*, 2018; Verma *et al.*, 2022;

[*] These authors contributed equally.

The NeurIPS 2023 Workshop on New Frontiers of AI for Drug Discovery and Development (AI4D3 2023), New Orleans, LA, USA, 2023.

Maus *et al.*, 2022), or using autoregressive generation processes guided by a policy acquired through Reinforcement Learning (Olivecrona *et al.*, 2017; Svensson *et al.*, 2023; Jain *et al.*, 2023).

These tools have achieved promising results on toy benchmarks such as the generation of novel compounds maximizing drug-likeness properties, but fall short on more concrete and specific examples such as discovering bioactive molecules for a given therapeutic target. Except for the intrinsic difficulty of real-world optimization tasks, the main issue lies in the unavailability of ground truth labels for newly generated molecules. This means that a proxy reward has to be used. Typically, practitioners resort to a statistical model as a substitute, trained using a labeled dataset that only accounts for a microscopic fraction of the whole chemical space, the latter ranging from 10^{20} to 10^{60} entities (Polishchuk *et al.*, 2013). As more diverse candidates are being explored, a significant covariate shift occurs between molecules seen at training time and those generated. Therefore, the reward predicted for generated molecules inevitably departs from the ground truth reward, causing the system to flag as promising molecules that are not (Renz *et al.*, 2019; Gendreau *et al.*, 2023). This issue, also known as *reward hacking* (Skalse *et al.*, 2022), is illustrated in Figure 1 using `Reinvent` (Olivecrona *et al.*, 2017), an autoregressive molecular generation method. From there, it can clearly be established that even if the mean proxy reward for a random subset sampled from the learned policy keeps increasing throughout generation cycles, the ground truth or oracle reward for candidates predicted as most promising actually decreases.

To tackle this issue, recent approaches operate in a probabilistic framework but cast inference in function space rather than parameter space for improved uncertainty quantification and a more semantically meaningful way to specify knowledge of preferred parametric function mappings on unlabelled data points (Klarner *et al.*, 2023). This allows for encouraging high predictive uncertainty in unexplored regions of chemical space or specifying prior knowledge about synthetic accessibility.

In this short communication, we take another route and propose to leverage expert feedback on the generated molecules to increasingly align the proxy reward model with the ground truth as more training samples are being acquired. Such an approach is motivated by the recent successful application of Human-In-The-Loop Machine Learning in goal-oriented molecular generation problems (Sundin *et al.*, 2022). The study demonstrated that a proxy reward model for bioactive molecules against the dopamine receptor D2 (DRD2) could be learned from human feedback. The latter took the form of a categorical score capturing whether the expert liked or disliked a candidate molecule generated. Furthermore, the learned model aligned well with an oracle. Human expertise was also found to compare favorably with statistical models for predicting other molecular properties such as solubility (Boobier *et al.*, 2017).

Based on typical practitioner needs, we outline three scenarios where expert feedback can play a crucial role in learning a reward model. The first scenario involves an expert who understands the design goal but lacks access to a predefined reward function. In this case, the reward is learned directly from the expert's observations, and the generative agent aims to emulate the expert's behavior to achieve their goal. The second scenario pertains to situations where the expert is aware of the design goal and has access to a predefined reward function, often derived from experimental data. Here, the generative model can be trained independently, with the expert supervising the learning process to ensure correct behavior. Finally, the third scenario encompasses cases where the expert's knowledge of the goal is partial but can complement a proxy reward model derived from experimental data. This scenario arises in contexts where the goal is to design specific compounds, guided by both data-driven models and the expert's unique insights, with the ultimate aim of generating suitable molecules for the target of interest.

We consider the third scenario as it is the most frequent one that practitioners may encounter. To approach it, we draw inspiration from the Learning To Defer (L2D) literature (Mozannar *et al.*, 2023) that aims at training a classifier able to defer its prediction to a human expert when needed. We design the reward as a combination of models trained on both experimental and expert data and use it to guide the molecular generation cycle. Moreover, we consider iterative fine-tuning of this combined model using several active learning strategies to improve its predictions for the most promising candidates discovered during the generation process. We evaluate our approach for the task of finding bioactive candidates against DRD2 and demonstrate that in an active learning setting, our combined model provides a consistent improvement over the vanilla proxy reward model.

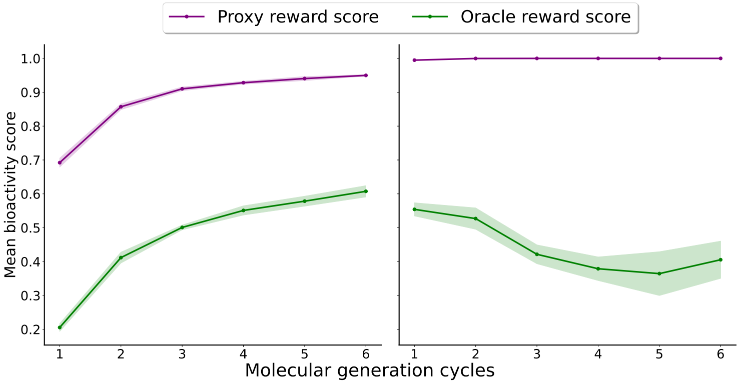

Figure 1: **The proxy reward perceived for molecules obtained across generation cycles does not align with its ground truth value.** Left: after every generation cycle, a mean reward score is computed on a batch of $M = 10000$ molecules sampled from the optimized agent using the proxy reward model. Right: mean reward score restricted to the 10% best candidates present among the batch of M molecules **according to the proxy reward model**. Mean and standard deviation obtained across 5 random seeds. For further details, see Section 3.

2 Method

We consider access to two distinct datasets $\mathcal{D}_y = \{\mathbf{x}_i, y_i\}_{i=1}^{n_y}$ and $\mathcal{D}_h = \{\mathbf{x}_j, h_j\}_{j=n_y+1}^{n_h}$, representing the molecules $\mathbf{x} \in \mathcal{X}$ together with experimental labels $y \in \mathcal{Y}$ and human labels $h \in \mathcal{Y}$ available at hand, with $\mathcal{Y} = \{0, 1\}$.

2.1 Reward model building

We introduce two classifiers, s_y and s_h, trained on $\mathcal{D}_y$ and $\mathcal{D}_h$, respectively. s_y captures the decision boundary learned based on experimental data while s_h captures the one induced by human knowledge. Each classifier is parameterized by a vector $\boldsymbol{\theta_l}, l \in \{y, h\}$, and outputs a probability $p_{\boldsymbol{\theta_l}}(\cdot)$. Next, the combined probability vector $[p_{\boldsymbol{\theta_y}}(\cdot), p_{\boldsymbol{\theta_h}}(\cdot)]$ is fed to a final *rejector* classifier s_r, parameterized by $\boldsymbol{\theta_r}$, whose task is ultimately to select which of s_y or s_h will end up predicting a given sample, effectively assessing whether we should rely on experimental or human knowledge. As such, s_r also outputs a probability $p_{\boldsymbol{\theta_r}}(\cdot)$, and at training time, we compute the *deferral indicator* $d_i \in \{0, 1\}$ for a sample $\mathbf{x}_i$ as

$$\begin{aligned} d_i :=\, &\mathbb{I}\left(\{p_{\boldsymbol{\theta_r}}(\mathbf{x}_i) > p_{\boldsymbol{\theta_y}}(\mathbf{x}_i)\} \cap \{y_i = 1\} \cap \{p_{\boldsymbol{\theta_y}}(\mathbf{x}_i) < p_{\boldsymbol{\theta_h}}(\mathbf{x}_i)\}\right) \\ &+ \mathbb{I}(\{p_{\boldsymbol{\theta_r}}(\mathbf{x}_i) > p_{\boldsymbol{\theta_y}}(\mathbf{x}_i)\} \cap \{y_i = 0\} \cap \{p_{\boldsymbol{\theta_y}}(\mathbf{x}_i) > p_{\boldsymbol{\theta_h}}(\mathbf{x}_i)\}), \end{aligned} \tag{1}$$

with $d_i = 1$ for a prediction deferred to s_h, to s_y otherwise. In effect, we are ensuring that the rejector has a higher score than the classifier trained on experimental data, and then, we ensure that the confidence of the classifier s_h trained on human data is higher than that of s_y. Here, the notion of confidence lies in the observation that for positive (resp. negative) labels, the output probability should be as close to 1 (resp. 0) as possible. The probability score as given by the global system is then

$$k_i := d_i p_{\boldsymbol{\theta_h}}(\mathbf{x}_i) + (1 - d_i) p_{\boldsymbol{\theta_y}}(\mathbf{x}_i). \tag{2}$$

Using the binary cross-entropy loss, our global loss function $\mathcal{L}$ amounts to:

$$\begin{aligned} \mathcal{L}(\boldsymbol{\theta}) =&\alpha \sum_{l \in \{y,h\}} \sum_{i=1}^{n_l} l_i \log(p_{\boldsymbol{\theta_l}}(\mathbf{x}_i)) + (1 - l_i) \log(1 - p_{\boldsymbol{\theta_l}}(\mathbf{x}_i)) \\ &+ (1 - \alpha) \sum_{i=1}^{n_y} y_i \log(k_i) + (1 - y_i) \log(1 - k_i), \end{aligned} \tag{3}$$

where $\boldsymbol{\theta} = [\boldsymbol{\theta_y}, \boldsymbol{\theta_h}, \boldsymbol{\theta_r}]$ and $\alpha \in [0, 1]$ is a hyperparameter balancing the importance of classifier accuracies with respect to the system accuracy. In practice, α is acquired by grid-search on an inde-

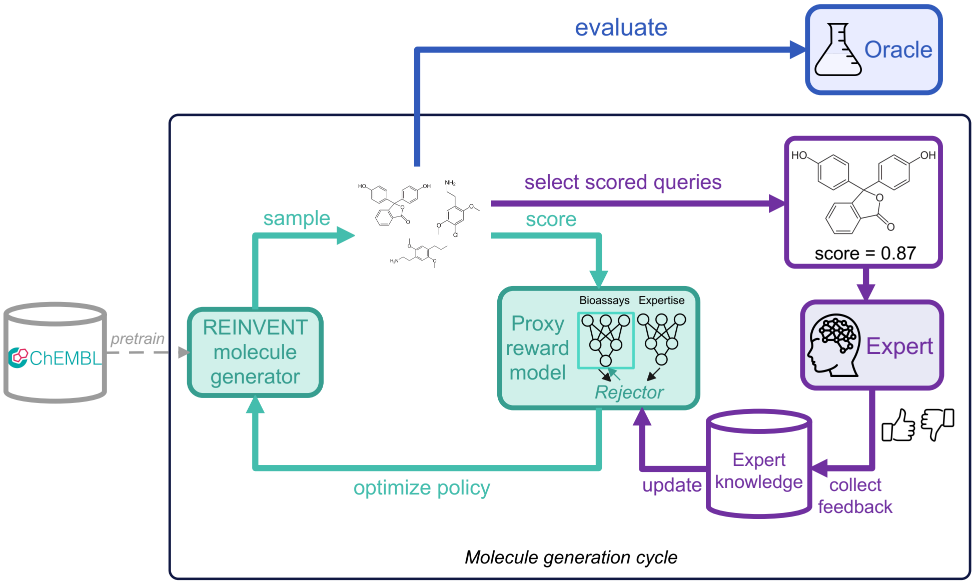

Figure 2: **Overview of the sequential experimental design pipeline for goal-oriented molecule generation.** The generation loop with `Reinvent` is shown in light green, where a Recurrent Neural Network pretrained on the ChEMBL database optimizes its policy to generate new molecules maximizing a proxy reward. The sequential experimental design is shown in purple, where the proxy reward model interacts with an expert via active query selection, collecting feedback about how much do the currently generated molecules align with the intended goal and updating its scoring strategy based on that feedback. One iteration of both loops constitute a molecule generation cycle. At the end of each generation cycle, the final set of generated molecules is evaluated by an oracle.

pendent validation set. Note that Equation 3 depends on $\boldsymbol{\theta_r}$ through d_i, involved in the computation of k_i.

Finally, during prediction time, an unseen sample $\mathbf{x}_*$ is passed through both classifiers s_y and s_h, and since we do not have access to the label required to compute the deferral indicator (Equation 1), the event $\{y_* = 1\}$ (resp. $\{y_* = 0\}$) is substituted by $\{p_{\boldsymbol{\theta_y}}(\mathbf{x}_*) > 0.5\}$ (resp. $\{p_{\boldsymbol{\theta_y}}(\mathbf{x}_*) < 0.5\}$).

2.2 Sequential experimental design

Once the model has been trained using $\mathcal{D}_y$ and $\mathcal{D}_h$, it can act as a reward for any conditional molecular generation framework. This paper focuses on `Reinvent` (Olivecrona *et al.*, 2017), a sequential generation method built on a Recurrent Neural Network tuned with a pre-specified reward function to guide chemical space exploration. Following (Sundin *et al.*, 2022), we now describe a sequential experimental design pipeline to further align the reward model with the ground truth reward (Figure 2).

At generation cycle t, `Reinvent` generates a batch of M molecules, a subset of which is shown to a human expert to acquire human data $(\mathbf{x}, h)$. These are then added to $\mathcal{D}_h$, an d one can train again the system (Equation 3) and obtain an updated reward model, which is then integrated back into `Reinvent`. Among the M generated molecules every iteration, the precise subset of m molecules shown to the expert is determined using an active learning criterion. Some desirable active learning objectives include querying the expert about the m most uncertain molecules, or the m most likely bioactive molecules, or the m most likely to be deferred.

Upon acquisition of expert labels h, an underlying assumption is that they align with ground truth labels y that would have been obtained if actual biological assays were carried, so that the reward model can align with the ground truth reward.

3 Experiments

3.1 Presentation of the task

We consider the task of generating novel molecules predicted to be active against the dopamine receptor DRD2. The aim is to quantify the discrepancy in predicted reward for generated molecules compared to the oracle reward, as the generation cycles go by. For our proposal to be successful, the combined model should catch up with the oracle reward faster than the vanilla model based on a single classifier. The training dataset for DRD2 bioactivity is taken as a subset of the Excape-db database (Sun *et al.*, 2017; Olivecrona *et al.*, 2017).

3.2 Implementation details

For classifiers s_y and s_h, we implement Multi-Layer Perceptrons, with the last layer followed by a sigmoid. The rejector s_r is taken as a logistic regression model.

In an optimal setting, the oracle reward would give the ground truth label for *any* molecule. However, ground truth labels, acquired from actual biological experiments, are only available for the training set. Thus, we create an oracle baseline by considering a Support Vector Machine trained on 20000 labeled samples, whereas the proxy reward model that will be used for scoring the molecules in `Reinvent` is only trained on 2400 labeled samples. For both proxy and oracle training, samples are represented with Extended Connectivity Fingerprint 6 (ECFP6) vectors of size 2048. SVMs, Random Forests or Boosting Trees were recently shown to better handle molecular fingerprint inputs than Deep Learning models, hence their use as an oracle model is a relevant assumption (Xia *et al.*, 2023).

Next, to a knowledgeable human expert on areas of the chemical space that differ from the experimental data, we draw inspiration from (Klarner *et al.*, 2023) and split the dataset into two clusters obtained using spectral clustering with a weighted affinity matrix corresponding to the pairwise Tanimoto similarity between ECFP6. Non-overlapping clusters then enforce diversity in the knowledge that can be gained from human or experimental data. Figure 3 confirms that the two clusters obtained are indeed well-separated through a visualization produced by UMAP dimensionality reduction (McInnes *et al.*, 2018).

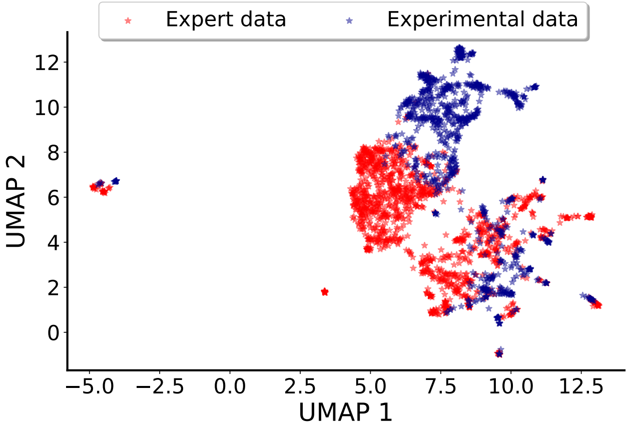

Figure 3: UMAP 2D view of the spectral clustering.

During the generation process, we acquire expert labels h on the generated molecules. For reproducibility purposes, we employ a synthetic expert using a Bernoulli model:

$$h \sim y\text{Bernoulli}(\pi) + (1 - y)\text{Bernoulli}(1 - \pi) \tag{4}$$

Where we recall that y is the ground truth bioactivity label for a molecule $\mathbf{x}$. For the training set, this corresponds to labels acquired through biological experiments, whereas for newly generated molecules, the "ground truth" y is obtained from the oracle trained on a large number of samples. A perfect expert is such that $h = y$, that is π is set to 1.

We assess the results for several active learning criteria that determine which molecules are presented to the human expert for labeling every iteration: greedy sampling, uncertainty sampling, expected predictive information gain (EPIG, Bickford Smith *et al.* (2023)), and random sampling. Detailed expressions for each criterion can be found in Supplementary Section A.

3.3 Results

At test time, both the classifier and model of expert knowledge included in the combined model have achieved high predictive performances, with F1-scores of 0.94 and 0.91 respectively on an *in-distribution* holdout test set. The combined model has achieved an F1-score of 0.95 on the same test set, with a percentage of deferral to the expert model of 27%. Most molecules being deferred were found to belong to the non-bioactive class, with predicted probabilities of them being non-bioactive by the expert model lower than that of the classifier, suggesting that the expert model is more reliable for detecting the true negatives.

In `Reinvent` , we use both the combined model and vanilla model (involving only a classifier) in separate trials as proxy reward models to guide the molecular generation towards DRD2 active candidates. The oracle reward is only used for evaluating the generated molecules. We compare generation results when using each of the reward models with and without active learning.

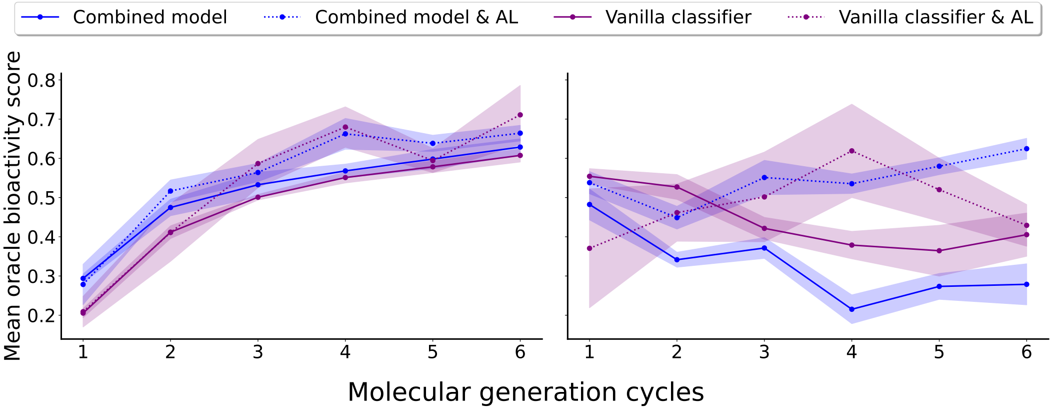

Figure 4: **Active learning jointly with a combined model yields increased oracle scores for best-scoring candidates.** Left: at the end of every generation cycle, $M = 10000$ molecules are sampled from the optimized agent using the proxy reward model. The mean oracle bioactivity score is computed. Right: Among the M generated molecules, a set composed of the 10% best candidates **according to each proxy model** is obtained, over which the mean oracle score is computed. A perfect expert is considered ($\pi = 1$), and the EPIG active learning strategy is employed.

Figure 4 reports the mean oracle reward scores computed across the molecular generation cycles for two proxy reward models. The first only relies on a classifier (purple curve), and the second on the combined model described above (blue curve). For dotted curves, active learning was carried out between each generation cycle, leading to $m = 50$ molecules being shown to the expert using the EPIG strategy every cycle. A perfect expert ($\pi = 1$) is assumed. The left panel of the figure presents the scores obtained on a batch of $M = 10000$ molecules, while the right panel considers the 10% best molecules according to the proxy reward model. From Figure 4, the obtained scores for the whole batch of 10000 molecules roughly look the same, whatever the model or active learning strategy considered. Discrepancies appear when statistics are computed over the 1000 best candidates. This number makes sense as it matches the order of magnitude of promising candidates sent for further pre-screening in pharmacological trials. For the remainder of the results section, we therefore focus on the top 1000 best candidates.

While increased performances for baselines leveraging active learning can partially be explained through the fact that these models were trained with a higher number of samples ($m = 50$ additional samples every generation cycle except the last), both active learning baselines enjoy quite different results towards the end, suggesting that the combined model benefits more from active queries.

This is further confirmed in Figure 5. The latter zooms in and considers the end of the optimization (when the sixth molecular generation cycle is over). From the top panel, it can be observed that our combined model benefits from active learning whatever the strategy employed. Quite surprisingly, random querying achieves the best performance here, highlighting the difficulties of applying active learning strategies in very high dimensional spaces (here, $d = 2048$). Next, for each model and acquisition strategy, the lower panel shows the difference in the predicted bioactivity score versus oracle score. The lower this quantity, the better the agreement between the proxy and oracle reward. Our combined model demonstrates a closer agreement, except for the uncertainty strategy.

We then analyzed the deferral behavior of the combined model before and after using active learning to fine-tune the expert model with additional expert feedback. When the combined model is not updated, the percentage of deferrals for scoring the molecules decreases considerably, from 27% to nearly 0%. Deferrals to the expert model are not useful with respect to the oracle (Figure S1**a**). As the molecule generation process progresses, most generated molecules are deferred to the classifier which predicts them as very promising but with higher disagreement with the oracle (Figure S1**b**).

After augmenting the combined model through active learning, deferrals became more useful, showcasing the improvement of the expert model in identifying the good molecules with a better alignment with the oracle (Figure S2**a**). Besides that, most molecules that were not deferred, thus scored by the classifier, were also predicted active by the oracle with a better alignment between the classifier and oracle scores. This suggests that the classifier predictions did also improve over time (Figure S2**b**). This might be because, as more expert feedback is being acquired to fine-tune the expert model, the generation evolved towards more relevant portions of the chemical space where both the expert model and classifier can predict more accurately while the rejector adapts to leverage the strengths of both expert model and classifier when scoring the molecules.

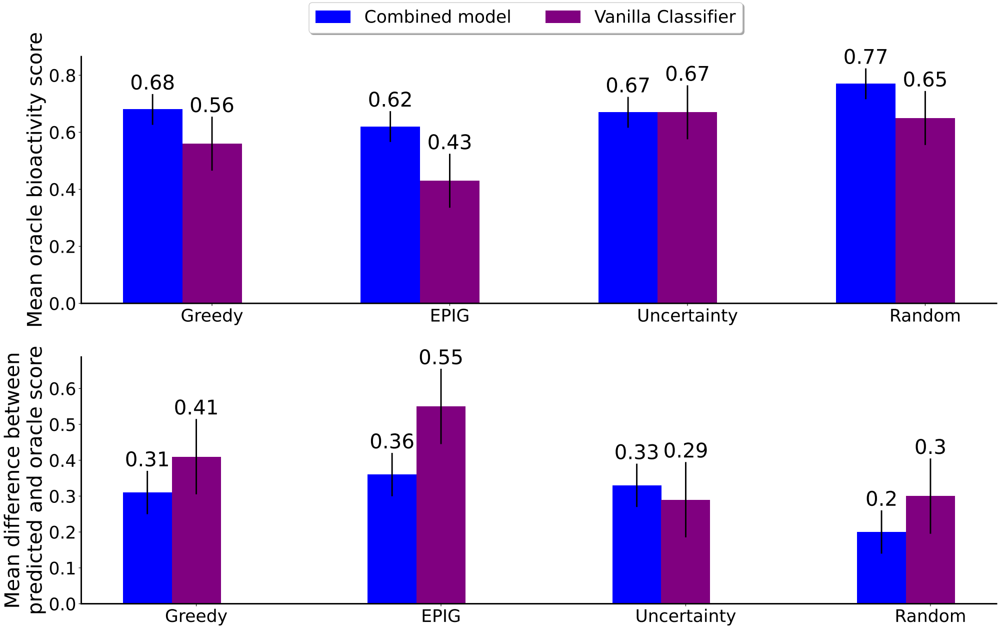

Figure 5: **The combined model provides increased oracle scores and better aligns with the oracle.** Top: at the end of the last generation cycle, $M = 10000$ molecules are sampled from the optimized agent using the proxy reward model. The mean oracle bioactivity score for the 10% best candidates from the batch is reported, according to each proxy model, and for each active learning criterion employed. Bottom: for the same setting, the mean difference between proxy reward and oracle score is reported. A perfect expert is considered ($\pi = 1$).

Lastly, the impact of noisy human feedback on performances ($\pi < 1$, Equation 4) is studied. More precisely, we consider $\pi = 0.8$ (Figure S3) $\pi = 0.5$ (Figure S4). The improvements demonstrated by our approach earlier in the perfect expert setting transfer well to that of a noisy expert. It is also worth noticing that for noisy experts, random exploration is no longer the best active learning criterion. Full trajectories along molecular generation cycles for different levels of expertise and different acquisition strategies can be found in Figure S5.

While the previous experiment considered a noisy human feedback, the initial human labels present in $\mathcal{D}_h$ before the molecular generation cycle were assumed to be noise free, i.e. for any molecule $\mathbf{x}_j$ from the initial dataset, $y_j = h_j$. We now lift this assumption, thus better capturing the fact that even for known compounds already belonging to standard molecular libraries, human-acquired labels might not be accurate. The results are provided in Figure S6. Unsurprisingly, considering noisy labels from the start negatively impacts the performances of both the vanilla classifier and the combined model, and the benefits offered by the combined approach over the vanilla one are now smaller. Interestingly, the performances reached by the combined model relying on a random expert ($\pi = 0.5$, bottom) are higher to that building on a knowledgeable expert ($\pi = 0.8$, top). This might stem from the fact that the combined system trained with $\pi = 0.5$ quickly learns not to rely on the expert, thus deferring less queries. In effect, this behavior leads to a combined system reverting back to the vanilla classifier to counteract potential errors or inconsistencies in the predicted bioactivity scores (Figure S7).

4 Conclusion

We presented a novel approach to fix proxy reward models used in goal-oriented molecular generation by leveraging expert knowledge. In an experiment aiming to generate novel molecules predicted to be active against the dopamine receptor DRD2, we showed improvements in the sense that our model led to increased oracle scores compared to a vanilla reward. The results shown are promising although remain preliminary as user studies involving real expert feedback need to be carried out in order to assess the viability of our approach. For reproducibility purposes, we designed a pipeline with a simulated expert that is used to acquire additional feedback and through which we take into account potential noise in the feedback to account for the uncertainty inherent to realistic Human-In-The-Loop scenarios.

From the methodological and modeling point of view, a number of steps could be undertaken to improve upon the current proof of concept. While we have drawn inspiration from the Learning To Defer paradigm (L2D, Mozannar *et al.* (2023)), we had to significantly depart from that setting. The reason is that L2D assumes that the human expert is always available when the prediction is deferred. This is only the case in the outer optimization loop of `Reinvent` , when already generated molecules are shown to the experts. In the inner generation loop, thousands of molecules are being assessed and so acquiring human labels for them is not an option. To circumvent this issue, we chose to integrate a model of expert knowledge, which can be used in the inner generation loop of `Reinvent`. This being said, integrating ideas of the L2D framework so that both classifiers s_y and s_h are fully aware of each other and adapt their decision boundary in that respect would certainly improve results. The L2D approach was also recently extended to the case of multiple experts (Verma *et al.*, 2023), which might represent an interesting avenue for work as well.

Finally, one might also consider other recently curated datasets on bioactivity such as the one recently introduced by Klarner *et al.* (2023) on the discovery of chemoprotective antimalarial drug candidates.

Acknowledgments and Disclosure of Funding

Julien Martinelli acknowledges the support of the Research Council of Finland under the HEALED project (grant 13342077). Yasmine Nahal acknowledges the support of the the European Union's Horizon 2020 research and innovation program under the Marie Sk lodowska-Curie Actions, grant agreement "Advanced machine learning for Innovative Drug Discovery (AIDD)" No. 956832. This work was also supported by the Academy of Finland (Flagship programme: Finnish Center for Artificial Intelligence FCAI) and ELISE Networks of Excellence Centres (EU Horizon:2020 grant agreement 951847). Finally, we also acknowledge the computational resources provided by the Aalto Science-IT Project from Computer Science IT.

References

Bickford Smith, F., Kirsch, A., Farquhar, S., Gal, Y., Foster, A., and Rainforth, T. (2023). Prediction-oriented bayesian active learning. In F. Ruiz, J. Dy, and J.-W. van de Meent, editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206.

Boobier, S., Osbourn, A., and Mitchell, J. (2017). Can human experts predict solubility better than computers? *Journal of Cheminformatics*, **9**(63).

Gendreau, P., Turk, J.-A., Drizard, N., Ribeiro da Silva, V. B., Descamps, C., and Gaston-Mathé, Y. (2023). Molecular assays simulator to unravel predictors hacking in goal-directed molecular generations. *Journal of Chemical Information and Modeling*, **63**(13).

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. (2018). Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Central Science*, **4**(2).

Jain, M., Deleu, T., Hartford, J., Liu, C.-H., Hernandez-Garcia, A., and Bengio, Y. (2023). GFlowNets for AI-driven scientific discovery. *Digital Discovery*, **2**.

Jin, W., Barzilay, R., and Jaakkola, T. (2018). Junction tree variational autoencoder for molecular graph generation. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80.

Klarner, L., Rudner, T. G. J., Reutlinger, M., Schindler, T., Morris, G. M., Deane, C., and Teh, Y. W. (2023). Drug discovery under covariate shift with domain-informed prior distributions over functions. In A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202.

Maus, N., Jones, H., Moore, J., Kusner, M. J., Bradshaw, J., and Gardner, J. (2022). Local latent space bayesian optimization over structured inputs. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35.

McInnes, L., Healy, J., Saul, N., and Großberger, L. (2018). Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, **3**(29).

Mozannar, H., Lang, H., Wei, D., Sattigeri, P., Das, S., and Sontag, D. (2023). Who should predict? exact algorithms for learning to defer to humans. In F. Ruiz, J. Dy, and J.-W. van de Meent, editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206.

Olivecrona, M., Blaschke, T., Engkvist, O., and Chen, H. (2017). Molecular de novo design through deep reinforcement learning. *Journal of Cheminformatics*, **9**(48).

Polishchuk, P., Madzhidov, T., and Varnek, A. (2013). Estimation of the size of drug-like chemical space based on gdb-17 data. *Journal of computer-aided molecular design*, **27**(8).

Renz, P., Van Rompaey, D., Wegner, J. K., Hochreiter, S., and Klambauer, G. (2019). On failure modes in molecule generation and optimization. *Drug Discovery Today: Technologies*, **32-33**.

Skalse, J., Howe, N. H. R., Krasheninnikov, D., and Krueger, D. (2022). Defining and characterizing reward hacking. *arXiv preprint arXiv:2209.13085*.

Sun, J., Jeliazkova, N., Chupakin, V., Golib-Dzib, J.-F., Engkvist, O., Carlsson, L., Wegner, J., Ceulemans, H., Georgiev, I., Jeliazkov, V., Kochev, N., Ashby, T., and Chen, H. (2017). Excape-db: An integrated large scale dataset facilitating big data analysis in chemogenomics. *Journal of Cheminformatics*, **9**(17).

Sundin, I., Voronov, A., Xiao, H., Papadopoulos, K., Bjerrum, E. J., Heinonen, M., Patronov, A., Kaski, S., and Engkvist, O. (2022). Human-in-the-loop assisted de novo molecular design. *Journal of Cheminformatics*, **14**(1).

Svensson, H. G., Tyrchan, C., Engkvist, O., and Chehreghani, M. H. (2023). Utilizing reinforcement learning for de novo drug design. *arXiv preprint arXiv:2303.17615*.

Verma, R., Barrejon, D., and Nalisnick, E. (2023). Learning to defer to multiple experts: Consistent surrogate losses, confidence calibration, and conformal ensembles. In F. Ruiz, J. Dy, and J.-W. van de Meent, editors, *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, volume 206.

Verma, Y., Kaski, S., Heinonen, M., and Garg, V. (2022). Modular flows: Differential molecular generation. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35.

Xia, J., Zhang, L., Zhu, X., and Li, S. Z. (2023). Why deep models often cannot beat non-deep counterparts on molecular property prediction? *arXiv preprint arXiv:2306.17702*.

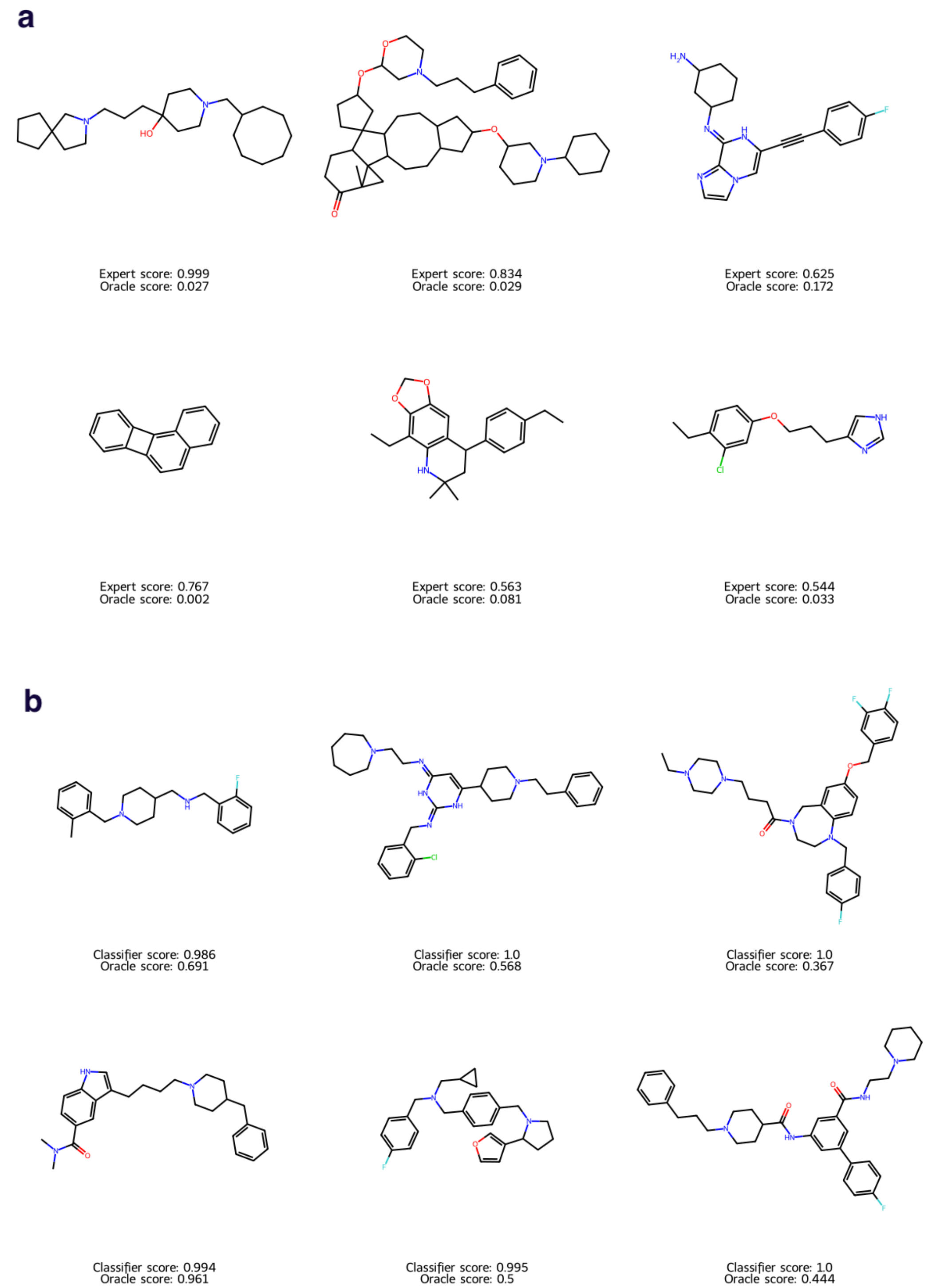

Figure S1: **(a)** Randomly sampled molecules that were deferred to the expert model when no active learning was used for combined model fine-tuning. **(b)** Randomly sampled molecules that were not deferred to the expert model, therefore scored by the classifier, when no active learning is used. Corresponding scores from the combined model (used for optimization) and the oracle are shown below.

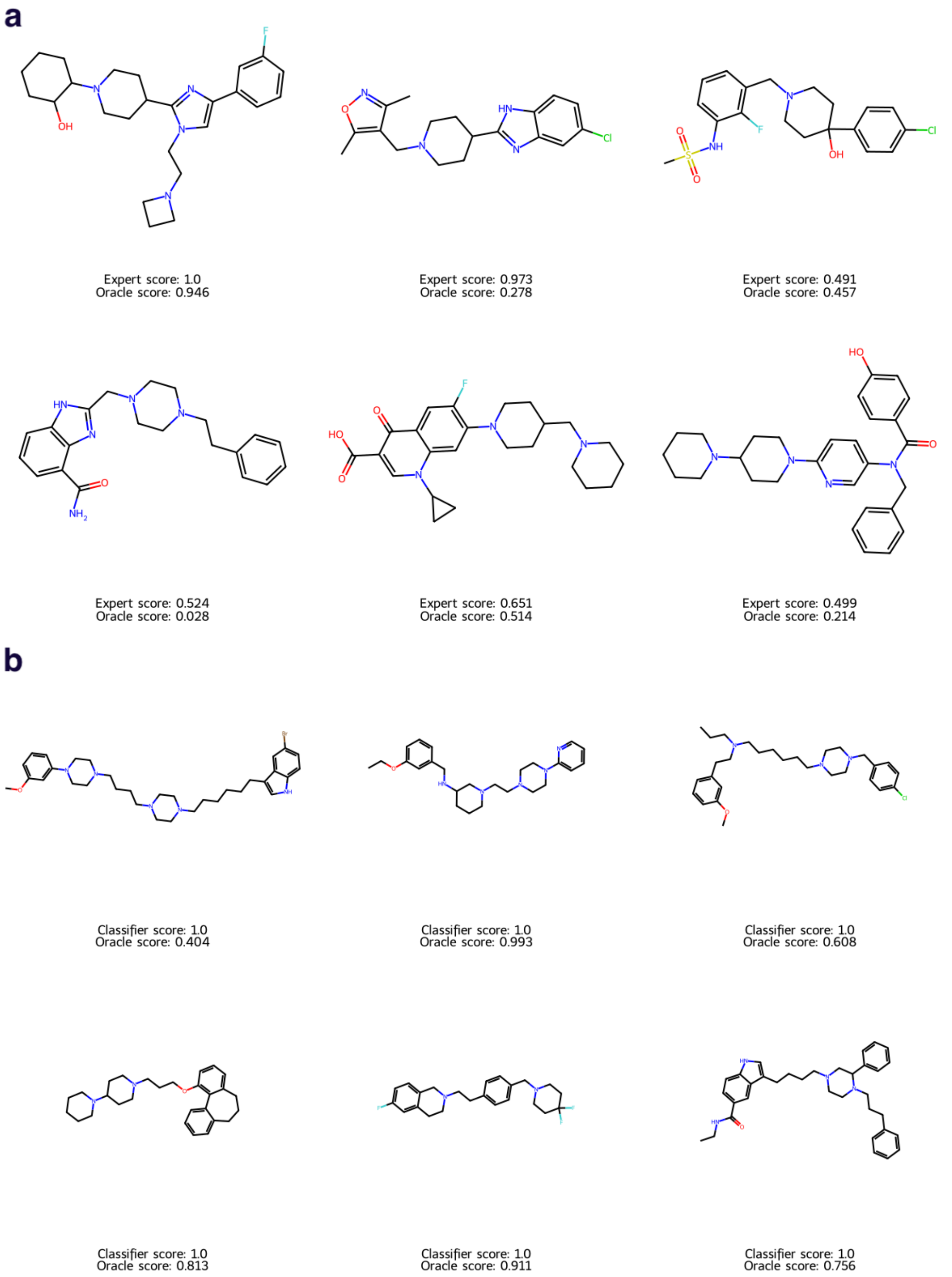

Figure S2: **(a)** Randomly sampled molecules that were deferred to the expert model when greedy sampling was used to acquire additional expert feedback ($\pi = 1$).**(b)** Randomly sampled molecules that were not deferred to the expert model, therefore scored by the classifier, when greedy sampling was used to acquire additional expert feedback ($\pi = 1$). Corresponding scores from the fine-tuned combined model (used for optimization) and the oracle are shown below.

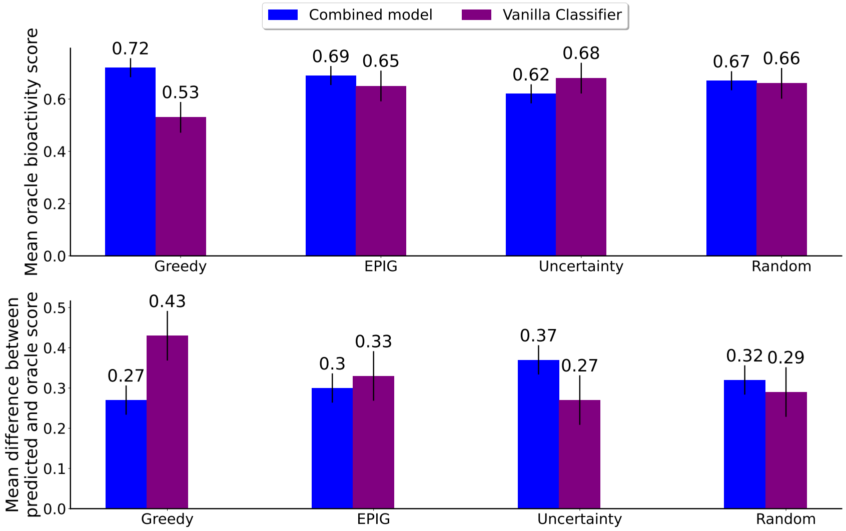

Figure S3: Results for a knowledgeable expert ($\pi = 0.8$). Top: at the end of the last generation cycle, $M = 10000$ molecules are sampled from the optimized agent using the proxy reward model. The mean oracle bioactivity score for the 10% best candidates from this batch is reported, according to each proxy reward model, depending on which active learning criterion was employed. Bottom: for the same setting, the mean difference between proxy reward and oracle score is reported.

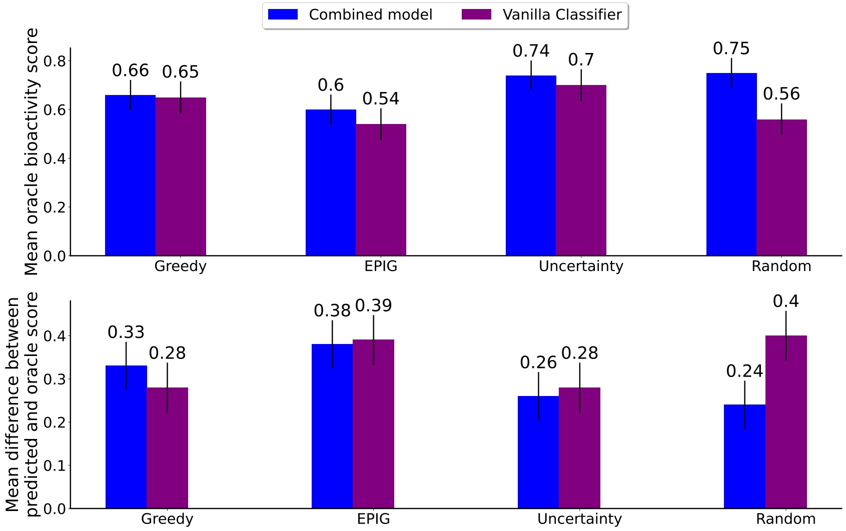

Figure S4: Results for a random expert with ($\pi = 0.5$). Top: at the end of the last generation cycle, $M = 10000$ molecules are sampled from the optimized agent using the proxy reward model. The mean oracle bioactivity score for the 10% best candidates from this batch is reported, according to each proxy reward model, depending on which active learning criterion was employed. Bottom: for the same setting, the mean difference between proxy reward and oracle score is reported.

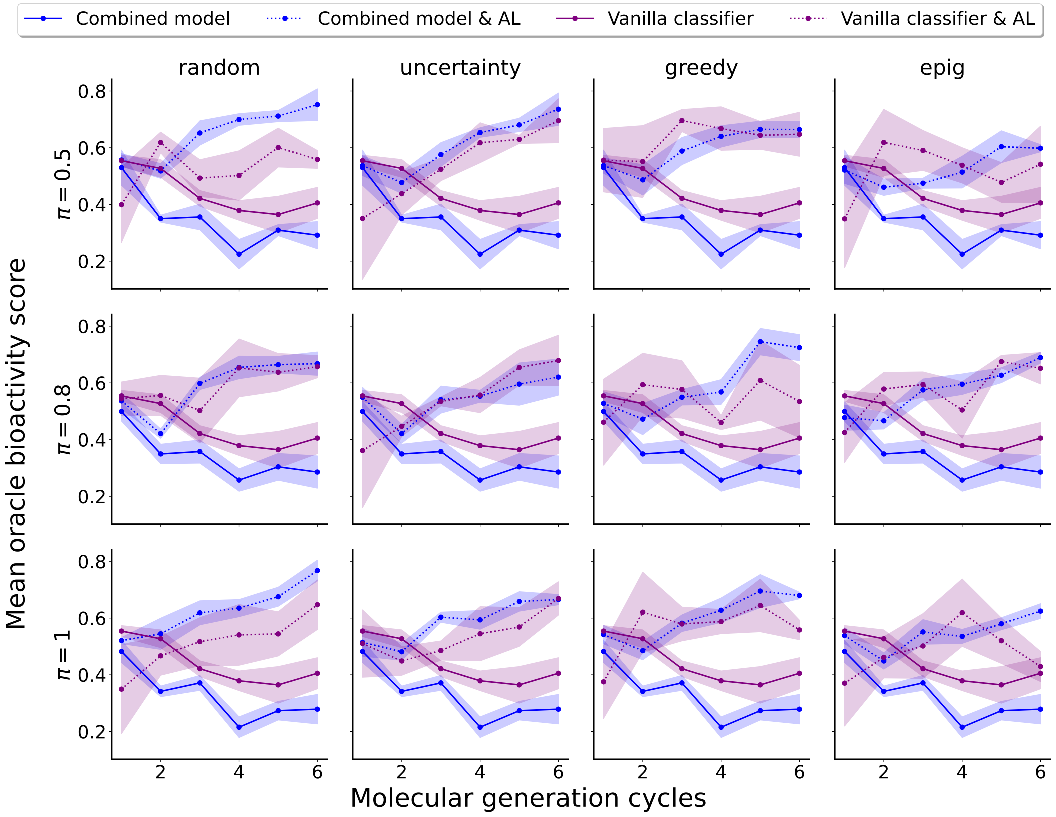

Figure S5: Mean oracle bioactivity score for both proxy reward models as a function of the level of expertise (rows) and active learning strategy chosen (columns), for the 10% best scoring molecules.

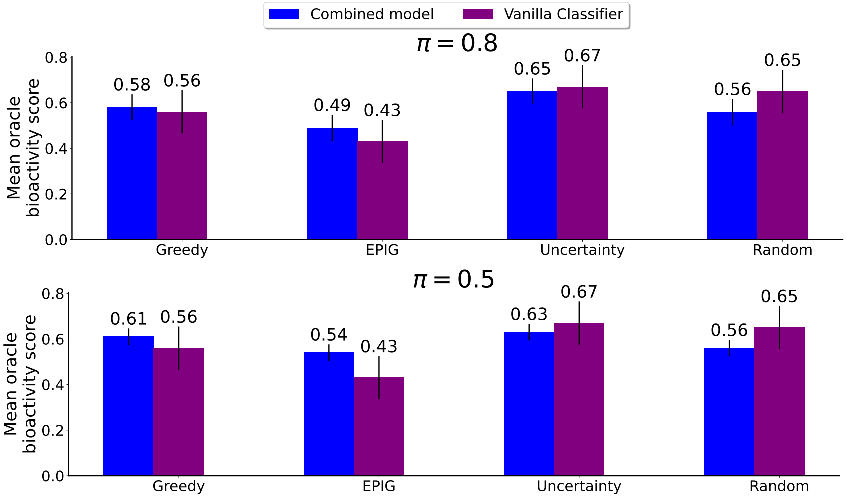

Figure S6: Results for a knowledgeable expert ($\pi = 0.8$, top) and a random expert ($\pi = 0.5$, bottom), where both the initial labels and sequentially-queried labels are noisy. At the end of the last generation cycle, $M = 10000$ molecules are sampled from the optimized agent using the proxy reward model. The mean oracle bioactivity score for the 10% best candidates from this batch is reported, according to each proxy reward model, depending on which active learning criterion was employed.

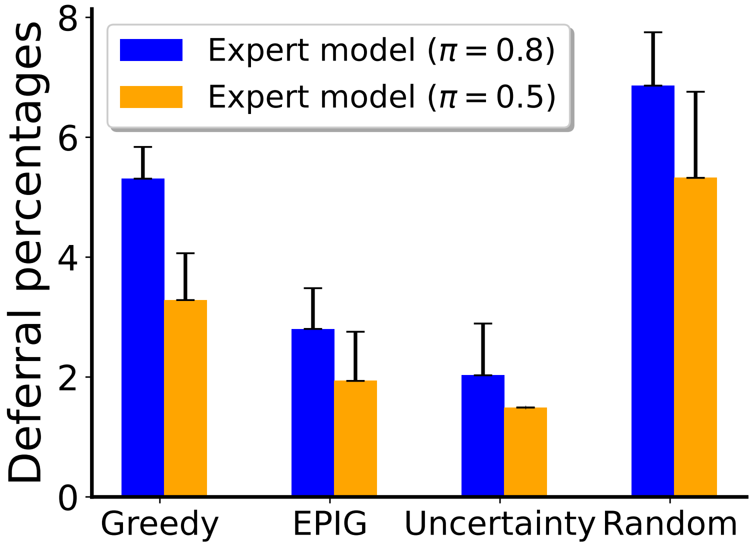

Figure S7: Percentages of deferred bioactivity score predictions to the noisy expert model at the end of the last generation cycle, where $M = 10000$ molecules are sampled from the optimized agent using the proxy reward model. Mean and standard deviations are computed across 5 runs.

A Active learning criteria

We here provide a mathematical description of the active learning criteria employed throughout the experiments. We consider a batch of Q unlabelled molecules $\mathbf{X} = \{\mathbf{x}_q\}_{q=1}^{Q}$ and a trained predictive model that outputs a score $\hat{s}(\cdot) \in [0, 1]$.

Greedy sampling.

$$\mathbf{x}_{\text{next}} = \underset{\mathbf{x} \in \mathbf{X}}{\text{argmax}} \ \hat{s}(\mathbf{x}) \tag{S1}$$

Random sampling.

$$\mathbf{x}_{\text{next}} \sim \mathcal{U}(\mathbf{X}) \tag{S2}$$

Expected Predicted Information Gain (EPIG) (Bickford Smith *et al.*, 2023) EPIG requires a probabilistic model in order to work. We follow the authors and obtain a conditional distribution $p(y|\mathbf{x})$ for a given input $\mathbf{x}$ by applying multiple forward passes to this input using $\hat{s}$, each time with different network nodes being dropped, in a random manner. This leads to different predictions for the same input. Next, EPIG focuses on reducing the predictive uncertainty over a pre-defined part of the input space. This particular choice is captured by a notion of input distribution $p_*(\mathbf{x}_*)$, yielding samples associated with labels y_* over which we want to be confident, in terms of prediction. Since we are mostly interested in increasing the true positive rate among the top high-scored molecules during a molecular generation process, we put probability mass on the 10% currently most promising molecules. Then, in a classification setting, we have (Bickford Smith *et al.*, 2023, Equation 5)

$$\begin{aligned} \mathbf{x}_{\text{next}} &= \underset{\mathbf{x} \in \mathbf{X}}{\text{argmax}} \ \text{EPIG}(\mathbf{x}) \\ &= \underset{\mathbf{x} \in \mathbf{X}}{\text{argmax}} \ \mathbb{E}_{p_*(\mathbf{x}_*)p(y,y_*|\mathbf{x},\mathbf{x}_*)}[\log p(y_*|\mathbf{x}_*, \mathbf{x}, y)] \end{aligned} \tag{S3}$$

Uncertainty sampling. As our model is deterministic, we compute uncertainty as

$$\mathbf{x}_{\text{next}} = \underset{\mathbf{x} \in \mathbf{X}}{\text{argmax}} \ \mathrm{H}[\hat{s}(\mathbf{x})] \tag{S4}$$

$$\mathrm{H}(\hat{s}(\mathbf{x})) = -(\hat{s}(\mathbf{x}) \log \hat{s}(\mathbf{x}) + (1 - \hat{s}(\mathbf{x})) \log(1 - \hat{s}(\mathbf{x})))$$

The function $z \mapsto -(z \log z + (1 - z) \log(1 - z))$ admits $z = 0.5$ as its unique maximizer, this effectively selects the sample $\mathbf{x}$ for which the associated prediction $\hat{s}(\mathbf{x})$ is the closest to 0.5.

One possible other choice would have been to follow the trick described for EPIG to cast $\hat{s}$ into a probabilistic model $p(y|\mathbf{x})$, and then use

$$\mathbf{x}_{\text{next}} = \underset{\mathbf{x} \in \mathbf{X}}{\text{argmax}} \ \mathbb{H}[p(y|\mathbf{x})] \tag{S5}$$