

## 546 A Surrogate Models and Acquisition Function

547 In this appendix, we provide additional details about the surrogate models used in our active learning  
548 experiments, as well as about the acquisition function.

### 549 A.1 Gaussian Processes

550 In our multi-fidelity active learning experiments, we model the posterior of  $f$  given  $x$ ,  $m$  and  
551 the current data set  $\mathcal{D}_j$ , assuming that observations are perturbed by noise from a normal distri-  
552 bution  $\mathcal{N}(0, \sigma^2)$ . The assumption of normally distributed noise with constant variance is widely  
553 used in the BO literature. Consider a set of  $n$  points  $z_{(1:n)} = (x_1, m_1), (x_2, m_2), \dots, (x_n, m_n)$   
554 with observed values  $y_{(1:n)} = y_1, y_2, \dots, y_n$ . We can then use Gaussian Processes such that  
555  $f|z_{(1:n)}, y_{(1:n)} \sim GP(\mu_n, K_n)$  with mean  $\mu_n$  and covariance function or kernel  $K_n$  evaluated at  
556 point  $z = (x, m)$  as

$$\begin{aligned} \mu_n(x) &= \mu(x) + K(x, x_{1:n})(K(x_{1:n}, x_{1:n}) + \sigma^2 I)^{-1}(y_{1:n} - \mu(x_{1:n})) \\ K_n(x_1, x_2) &= K((x_1, x_2) - K(x_1, x_{1:n})(K(x_{1:n}, x_{1:n}) + \sigma^2 I)^{-1}K(x_{1:n}, x_2)). \end{aligned}$$

558 We adapt the multi-fidelity kernel as proposed in [68], such that the kernel function of the GP is

$$K(z_1, z_2) = K_1(x_1, x_2) \times K_2(m_1, m_2),$$

559 where  $K_1(\cdot, \cdot)$  is a square-exponential kernel and

$$K_2(m_1, m_2) = c + (1 - m_1)^{1+\delta}(1 - m_2)^{(1+\delta)},$$

560 where  $c, \delta > 0$  are hyper-parameters.

### 561 A.2 Deep Kernel Learning

562 While for the synthetic (simpler) tasks we use exact GPs, for the benchmark tasks we implement  
563 deep kernel learning [DKL; 67]. In DKL, the inputs are transformed by

$$k(x_i, x_j | \theta) \rightarrow k(g(x_i, w), g(x_j, w) | \theta, w),$$

564 where the non-linear mapping  $g(x, w)$  is a low-dimensional continuous embedding, learnt via a deep  
565 neural network—a transformer in our tasks. To scale the GP to large datasets, we implement the  
566 stochastic variational GP based on the greedy inducing point method [13]. We adopt the deep kernel  
567 learning experimental setup from [61].

### 568 A.3 Acquisition Function

569 Max Value Entropy Search (MES) [65] is an information-theoretic acquisition function. The stan-  
570 dard, single-fidelity MES seeks to query the objective function at locations that reduce our current  
571 uncertainty in the maximum value of  $f^*$ . It aims to maximise the mutual information between the  
572 value of the objective function  $f$  when choosing point  $x$  and the maximum of the objective function,  
573  $f^*$ . This contrasts with previously proposed entropy search (ES) criterion, which instead considers  
574 the  $\arg \max$  of the objective function. The MES criterion is defined as follows:

$$\alpha = I(f^*; y | \mathcal{D}_j) = H(y | \mathcal{D}_j) - \mathbb{E}_{f^*} [H(y | \mathcal{D}_j, f^*) | \mathcal{D}_j],$$

575 where  $y$  is the outcome of experiment  $x$  and  $\mathcal{D}_j$  is the data set at the  $j^{th}$  active learning iteration. and  
576  $H(Y) = \mathbb{E}_Y [-\log(p(Y))]$  is the differential entropy of random variable  $Y$ .

577 The information gain is defined as the reduction in entropy of  $y$  provided by knowing the maximal  
578 value  $f^*$

$$IG(y, m | \mathcal{D}_n) = H(y | \mathcal{D}_n) - H(y | f^* < m, \mathcal{D}_n)$$

579 It follows that the MES acquisition function can be expressed in terms of  $IG$ :

$$\alpha = \mathbb{E}_{m \sim f^*} [IG_n(y, m | \mathcal{D}_n)],$$

where  $y \sim \mathcal{N}(\mu_A, \sigma_A)$ ,  $f(x) \sim \mathcal{N}(\mu_B, \sigma_B)$  and the difference between  $y$  and  $f(x)$  is just independent Gaussian noise.

Replacing the maximisation of an intractable quantity with the maximisation of a lower bound is a well-established strategy. Instead of attempting to evaluate the intractable quantity,  $IG$ , we evaluate its lower bound  $IG^{Approx}$ .

Thus, the acquisition function becomes

$$\alpha = \frac{1}{\mathcal{M}} \sum_{m \in \mathcal{M}} IG^{Approx}(y, m | \mathcal{D}_n)$$

$$IG^{Approx} = \frac{1}{2} \log |R| - \frac{1}{2\mathcal{M}} \sum_{m \in \mathcal{M}} \log(1 - \rho^2 \frac{\phi(\gamma(m))}{\Phi(\gamma(m))} [\gamma(m) + \frac{\phi(\gamma(m))}{\Phi(\gamma(m))}]),$$

where  $\phi$  and  $\Phi$  are the standard normal cumulative distribution and probability density functions (as arising from the expression for the differential entropy of a truncated Gaussian),  $\gamma(m) = \frac{m - \mu_n(x)}{\sigma_n(x)}$  and  $R$  is the correlation matrix with elements  $R_{i,j} = \frac{\Sigma_{i,j}^y}{\Sigma_{i,i}^y \Sigma_{j,j}^y}$ .

This construction is called the General Information-Based Bayesian Optimisation (GIBBON) acquisition function [43].

**Multi-Fidelity Formulation** Let the maximum of the highest fidelity function  $f_M$  (when  $M$  different fidelities are available to querying) be  $f_M^*$ . We obtain a pair  $(x, m)$  which maximally gains information of the optimal value  $f^*$  of the highest fidelity per unit cost.

$$\alpha(x, m) = \frac{1}{\lambda_m} I(f_M^*; f_m | \mathcal{D}_j). \quad (3)$$

where  $\lambda_m$  is the cost of the oracle at fidelity  $m$ .

## B Experimental Details

This appendix presents the details about the experiments presented in the main Section 4. First, we provide general details about all tasks and then present details specific to each task in separate sections.

**Policy model architecture** For all tasks, the architecture of the forward policy model to train GFlowNet is multi-layer perceptron with 2 hidden layers and 2048 units per layer. The backward policy model was set to share the parameters with the forward model, except for the parameters of the output layer, which were trained. We use LeakyReLU as our activation function as in [6]. All models are trained with the Adam optimiser [35].

**GFlowNet reward** As discussed in Section 3.2, the reward for GFlowNet in our multi-fidelity GFlowNet algorithm is the acquisition function max entropy search (MES) (and its multi-fidelity variant) for all experiments. In order to increase the relative reward of higher values of the acquisition function, we transform the MES value  $\alpha$  with the the reward function  $R(\alpha) = \frac{\alpha \times \rho^{j-1}}{\beta}$ , where  $j$  is the active learning round.

**Budget and initial data set** For each task, we assign a total budget  $B = \gamma \times C$ , where  $C$  is the cost if the highest fidelity oracle. The cost of the lower fidelity oracles are as specified in Section 4. The initial data sets were constructed according to an initial budget. Depending on this initial budget, we set the number of training data points for the single-fidelity baseline, as well as the number of points from each oracle in the multi-fidelity experiments. Task specific information is summarized in Table 2. The initial data set is split into train-validation in the ratio of 9:1 for all tasks.

All the above details for all tasks are summarized in Table 1 and Table 2. Our models are implemented in pytorch [47], and rely on botorch [3] and GPytorch [].

Table 1: Implementation details of multi-fidelity experiments of all tasks

Task	Surrogate	$\beta$	$\rho$	$\gamma$
Branin	Exact GP	1	1	300
Hartmann 6D	Exact GP	1e-2	1	100
DNA	DKL	1e-5	1.5	5120
Antimicrobial Peptides	DKL	1e-5	1	320
Molecules	DKL	1e-6	1.5	1280

Table 2: Oracle costs, initial budget and initial training data sets. Oracles are indexed by increasing level of fidelity.

Task	Oracle costs			Budget	Initial training examples			
	$m = 1$	$m = 2$	$m = M$		SF	Multi-Fidelity		
						$m = 1$	$m = 2$	$m = M$
Branin	0.01	0.1	1	42	4	20	20	2
Hartmann 6D	0.125	0.25	1	200	25	80	40	5
AMP	–	0.2	20	2500	50	2000	2000	10
DNA	0.5	0.5	50	1600	80	–	3000	50
Molecules	1	3	7	1050	150	700	68	16

## 617 B.1 Branin

618 The Branin function is evaluated in the domain  $[-5, 10] \times [0, 15]$  using the following expression:

$$f(x) = (x_2 - \frac{-1.25x_1^2}{\pi^2} + \frac{5x_1}{\pi} - 6)^2 + (10 - \frac{5}{4\pi})\cos(x_1) + 10.$$

619 This corresponds to the modification introduced in [58]. As lower fidelity functions, we used the  
 620 expressions from [50], which involve non-linear transformations of the true function as well as shifts  
 621 and non-uniform scalings. The functions, indexed by increasing level of fidelity, are the following:

$$f_1(x) = f_2(1.2(x + 2)) - 3x_2 + 1$$

622

$$f_2(x) = 10\sqrt{f(x - 2)} + 2(x_1 - 0.5) - 3(3x_2 - 1) - 1$$

623 On the discrete set of points obtained after mapping the continuous domain to a  $100 \times 100$  grid, the  
 624 explained variance of the oracles are 0.1125, 0.825 and 1, in increasing level of fidelity.

625 We use the `botorch` implementation of an exact multi-fidelity gaussian process as described in A.1  
 626 for regression. The active learning batch size  $B$  is 30 in the Branin task.

## 627 B.2 Hartmann 6D

628 The true Hartmann function is given by

$$f(x) = \sum_{i=1}^4 \alpha_i \exp(-\sum_{j=1}^3 A_{ij}(x_j - P_{ij})^2),$$

629 where  $\alpha = [1.0, 1.2, 3.0, 3.2]$  and  $A, P \in R^{4 \times 6}$  are the following fixed matrices:

$$A = \begin{bmatrix} 10 & 3 & 17 & 3.5 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 1 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 1 \end{bmatrix}$$

$$P = 10^{-4} \times \begin{bmatrix} 3689 & 1170 & 267 \\ 4699 & 4387 & 7470 \\ 1091 & 8732 & 5547 \\ 381 & 5743 & 8828 \end{bmatrix}$$

To simulate the lower fidelities, we modify  $\alpha$  to  $\alpha(m)$  where  $\alpha(m) = \alpha + (M - m)\delta$  where  $\delta = [0.01, -0.01, -0.1, 0.1]$  and  $M = 3$ . The domain is  $X = [0, 1]^6$ . This implementation was adopted from [32]. For the surrogate, we use the same exact multi-fidelity GP implementation as of Branin. The active learning batch size is 10.

### B.3 DNA

We conduct experiments using a two-oracle setup with costs 20 and 0.2 for the high and low fidelity oracles, respectively. As objective function (highest fidelity oracle) we used the free energy of the secondary structure of DNA sequences obtained via the software NUPACK [72], setting the temperature at 310 K. The free energy can be seen as a proxy of the stability of the sequences. To construct a lower fidelity oracle, we train a transformer with 8 layers, 1024 hidden units per layer and 16 heads. We sampled 1 million random sequences for the training set. The explained variance of the transformer model is 0.8. For the probabilistic surrogate, we implement deep kernel learning. The parameters of the surrogate as well as other details about the DNA task are provided in Table 5.

Table 3: Deep Kernel hyperparameters for the DNA and Antimicrobial tasks

	Hyperparameter	Value
Architecture	Num. of Layers	8
	Num. of Heads	8
	Latent dimension	64
	GP likelihood variance init	0.25
	GP lengthscale prior	$\mathcal{N}(0.7, 0.01)$
	Num. of inducing points (SVGP head)	64
Optimisation	Batch Size	128
	Learning Rate	1e-3
	Adam EMA params ( $\beta_1, \beta_2$ )	(0., 1e-2)
	Max Num. of Epochs	512
	Early stopping patience (Num. of Epochs)	15
	Early stopping holdout ratio	0.1

### B.4 Antimicrobial Peptides

We construct a three-oracle setup where the oracle corresponding to the highest fidelity is trained on data from DBAASP [51]. Each antimicrobial peptide belongs to a group. For the lower fidelity oracles, we divide the datapoints into two halves such that the set of groups present in one half is mutually exclusive of the other. This simulated a setup wherein the lower fidelity oracles specialised only in a specific subregion of the entire sample space. The configurations of the oracle models are present in Table 4. We assigned a cost of 50 to the highest fidelity oracle and 0.5 to both lower fidelity oracles. The explained variance of the oracles was 0.1435, 0.099 and 1 respectively. As for DNA, we implement deep kernel learning with hyperparameters in Table 5.

### B.5 Small Molecules

We implement the oracles using RDKit 2023.03 [1] and the semiempirical quantum chemistry package xTB. We use GFN2-xTB [4] method for the single point calculation of ionization potential and electron affinity with empirical correction terms. In the cheapest oracle, we consider one conformer obtained by RDKit with its geometry optimised via force-field (MMFF94[25]), this geometry is used to calculate (vertical) IP/EA. In the second fidelity, we consider two conformers obtained by RDKit, and take the lowest energy conformer after optimisation by MMFF94, and further optimise it via GFN2-xTB to

Table 4: Oracles for the antimicrobial peptides task (indexed by increasing level of fidelity)

Fidelity	# Training points	Model	# Layers	# Hidden units	Epoch
1	3447	MLP	2	512	51
2	3348	MLP	2	512	51
3	6795	MLP	2	1024	101

Table 5: Deep Kernel hyperparameters for the molecular tasks

	Hyperparameter	Value
Architecture	Num. of Layers	8
	Num. of Heads	8
	Latent dimension	32
	GP likelihood variance init	0.25
	GP lengthscale prior	$\mathcal{N}(0.7, 0.01)$
	Num. of inducing points (SVGP head)	64
Optimisation	Batch Size	128
	Learning Rate	1e-3
	Adam EMA params ( $\beta_1, \beta_2$ )	(0., 1e-2)
	Max Num. of Epochs	512
	Early stopping patience (Num. of Epochs)	15
	Early stopping holdout ratio	0.1

obtain the ground state geometry; this remains a vertical IP/EA calculation. In the highest fidelity oracle, we consider four conformers obtained by RDKit, and take the lowest energy conformer after optimisation by MMFF94, and further optimise it via GFN2-xTB; the corresponding ion is then optimised by GFN2-xTB, and the adiabatic energy difference is obtained via total electronic energy. The fidelities are based on the fact that vertical IP/EA approximates that of adiabatic ones (to varying degrees, depending on the molecule). On a test dataset of 1400 molecules of length<64, the explained variance of the oracles is 0.1359, 0.279, 1 and 0.79, 0.86, 1 for the EA and IP tasks respectively.

We note that this is proof-of-concept and hence we do not conduct a full search of conformers, and nor do we use Density Functional Theory calculations, but the highest fidelity oracle has a good correlation with experiments [44].

In the environment for GFN, we consider a set of SELFIES vocabularies containing aliphatic and aromatic carbon, boron, nitrogen, oxygen, fluorine, sulfur, phosphorous, chlorine, and bromine, subject to standard valency rules. We do not consider synthesizability in this study and we note it may negatively impact GFN as unphysical molecules could produce false results for the semiempirical oracle.

## C Metrics

In this section, we provide additional details about the metrics used for the evaluation of the proposed MF-GFN as well as the baselines.

**Mean Top- $K$  Score** We adapt this metric from [6]. At the end of an active learning round, we sample  $N(x, m)$  candidates and then select the top- $K$  candidates ( $K \ll N$ ) according to the acquisition function value. In the experiments, we score these top- $K$  candidates with the corresponding oracle. In the figures, we report the mean score according to the highest-fidelity oracle.

**Mean Diverse Top- $K$  Score** This is a version of the previous metric by which we restricts the selection of the  $K$  candidates to examples that are diverse between each other. We use similarity measures (*vide infra*) such that we sample the top- $K$  candidates where each candidate is at most similar to each other by a certain threshold. For antimicrobial peptides, the sequence identity threshold is 0.35; for DNA aptamers, the sequence identity threshold is 0.60; for molecules, the Tanimoto similarity distance threshold is 0.35.

**Diversity** In order to measure the diversity of a set of candidates, we use the similarity index with the following details for each of the tasks:

- **DNA aptamers:** The similarity measure is calculated by the mean of pairwise sequence identity between a set of DNA sequences. We utilize global alignment with Needleman-Wunsch algorithm and standard nucleotide substitution matrix, as calculated by biotite package [37].

- **Antimicrobial peptides:** The similarity measure is calculated by the mean of pairwise sequence identity between a set of peptide sequences. We utilize global alignment with Needleman-Wunsch algorithm and BLOSUM62 substitution matrix, as calculated by biotite package [37].
- **Molecules:** The similarity measure is calculated by the mean of pairwise Tanimoto distances between a set of molecules. Tanimoto distances are calculated from Morgan Fingerprints (radius of two, size of 2048 bits) as implemented in RDKit package [1].

## D Additional Results

### D.1 Impact of Oracle Costs

As discussed in Section 3.2, a multi-fidelity acquisition function like the one we use (defined in Eq. (2)) is a cost cost-adjusted utility function. Therefore, the cost of each oracle has a crucial impact in the values of the acquisition function. In our tasks with small molecules (Section 4.4.3), we used oracles with costs proportional to their computational demands and we observed that multi-fidelity active learning largely outperformed single-fidelity active learning. However, depending on the set of costs of the oracles, the advantage of multi-fidelity methods can significantly diminish.

In order to analyse the impact of the oracle costs in the performance of MF-GFN, we run several experiments on the DNA task (Section 4.4.1), which consists of two oracles, with various sets of oracles. In particular, besides the costs used in the experiments presented in the main paper, with costs (0.2, 20) for the lowest- and highest-fidelity oracles, we trained with costs (1, 20) and (10, 20). We present the results in Fig. 4. We can indeed confirm that the advantage of MF-GFN over SF-GFN decreases as the cost of the lowest-fidelity oracle becomes closer to the cost of the highest-fidelity oracle. However, it is remarkable that even with a ratio of costs as small as 1 : 2, MG-GFN still outperforms not SF-GFN but also MF-PPO in terms of cost effectiveness, without diversity being negatively impacted. It is important to note that practical scenarios of scientific discovery, the cost of lower fidelity oracles is orders of magnitude small than the cost of the most accurate oracles, since the latter correspond to wet-lab experiments or expensive computer simulations.

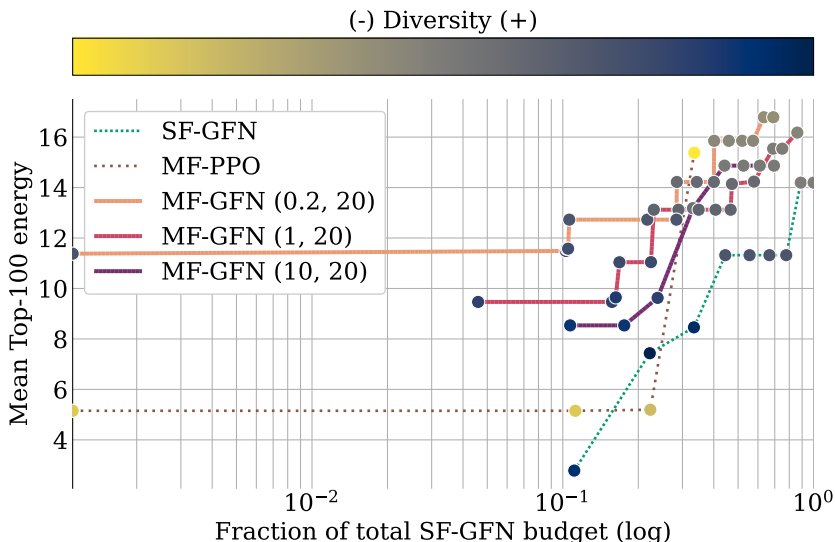


Figure 4: Analysis of the impact of the oracle costs on the performance of MG-GFN. We observe that the advantage over SF-GFN and MF-PPO (0.2, 20) decreases as the cost of the lower fidelity oracle becomes closer to the cost of the highest fidelity oracle. Nonetheless, even with a cost ratio of 1 : 2 MF-GFN displays remarkable performance with respect to other methods.

## 719 D.2 Energy of Diverse Top- $K$

720 In this section we complement the results presented in Section 4 with the mean diverse top- $K$  scores,  
 721 as defined in Appendix C. This metric combines that mean top- $K$  score and the measure of diversity.  
 722 Figure 5 shows the results on the DNA, AMP and the molecular tasks.

723 The results with this metric allow us to further confirm that multi-fidelity active learning with  
 724 GFlowNets is able to discover sets of diverse candidates with high mean scores, as is sought in  
 725 many scientific discovery applications. In contrast, methods that do not encourage diversity such as  
 726 RL-based algorithms (MF-PPO) obtain comparatively much lower results with this metric.

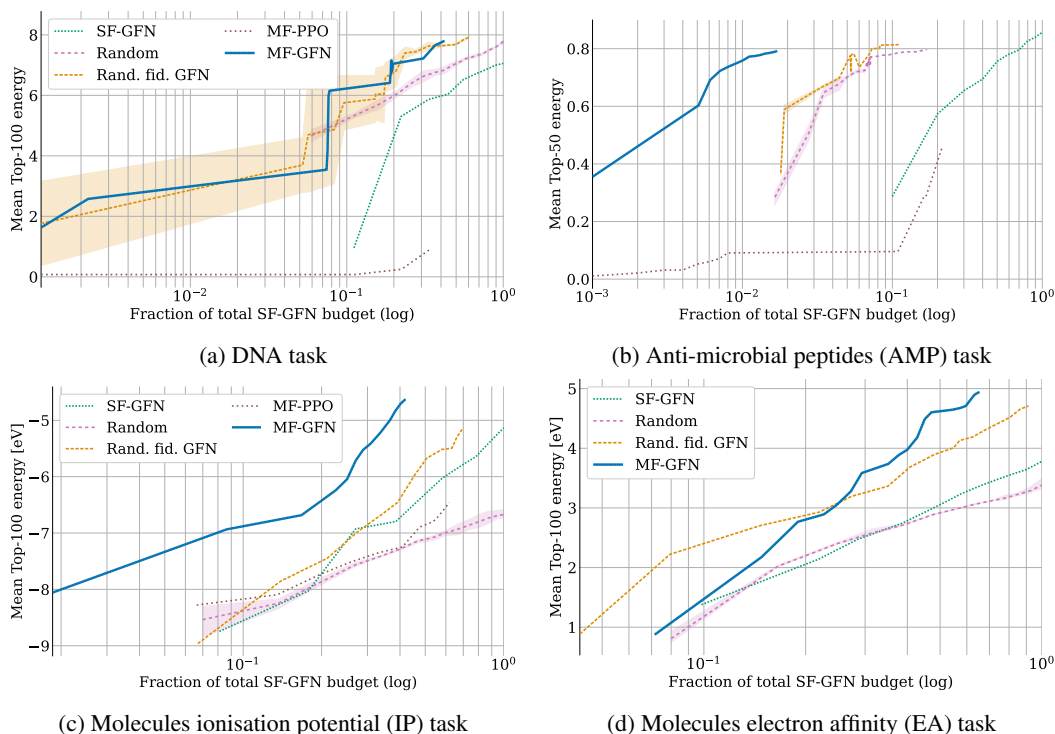


Figure 5: Mean scores (energy) of diverse top- $K$  candidates on the DNA (top left), AMP (top right) and molecular (bottom) tasks. The mean energy is computed across the top- $K$  examples at each active learning round that also satisfy the criteria of diversity. Consistent with the diversity metrics observed in Fig. 2, we here see that GFlowNet-based methods, and especially MF-GFN, obtain good results according to this metric, while MF-PPO achieves comparatively much lower mean energy.