MODEL-BASED REINFORCEMENT LEARNING FOR BIOLOGICAL SEQUENCE DESIGN

3/12/22

Authors: Christof Angermueller, David Dohan, David Belanger,
Ramya Deshpande, Kevin Murphy, Lucy Colwell

ICLR 2020

Benyamin Jamialahmadi

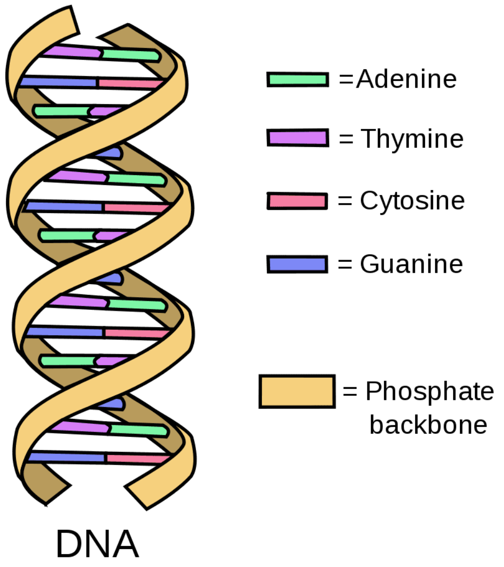
CS 885 Paper Presentation Winter 2022

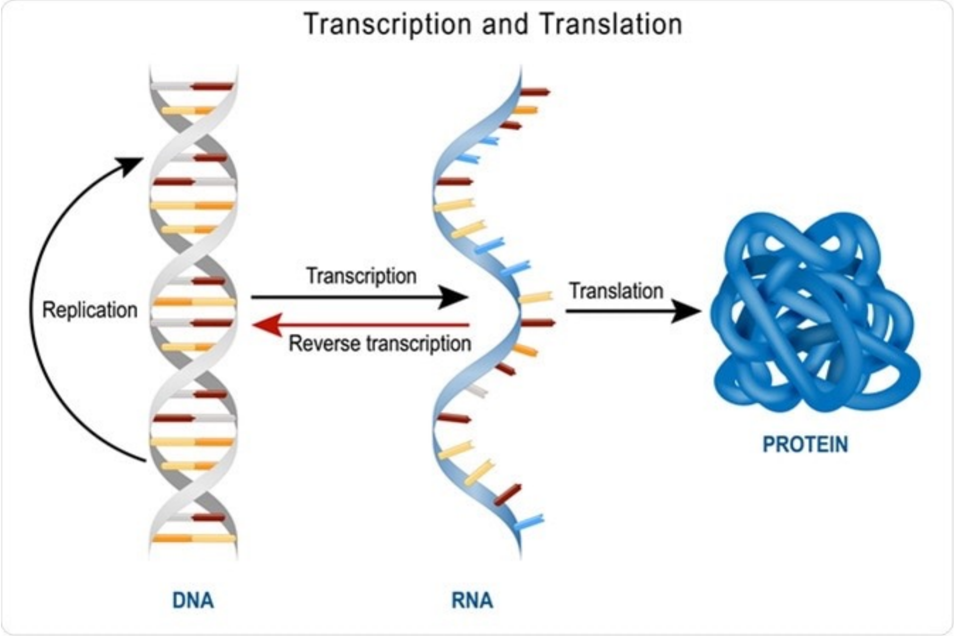
UNIVERSITY OF WATERLOO | FACULTY OF MATHEMATICS

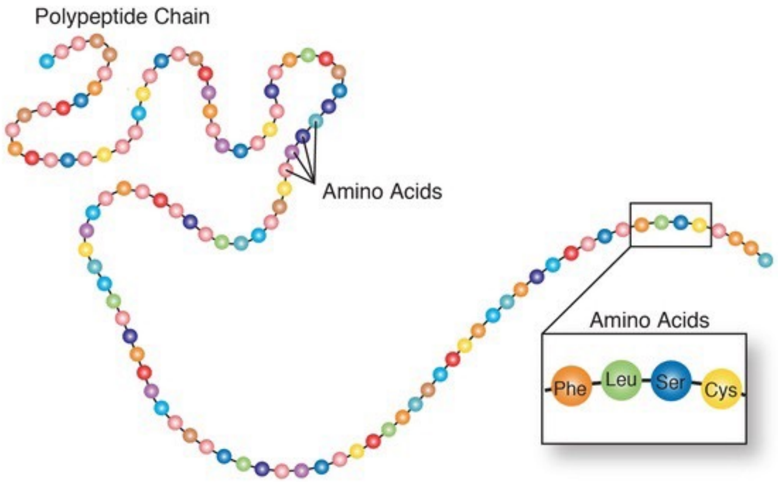
Agenda

- Introduction and Background
- Method
- Empirical Evaluation
- Conclusion

Introduction: DNA and Protein Sequences



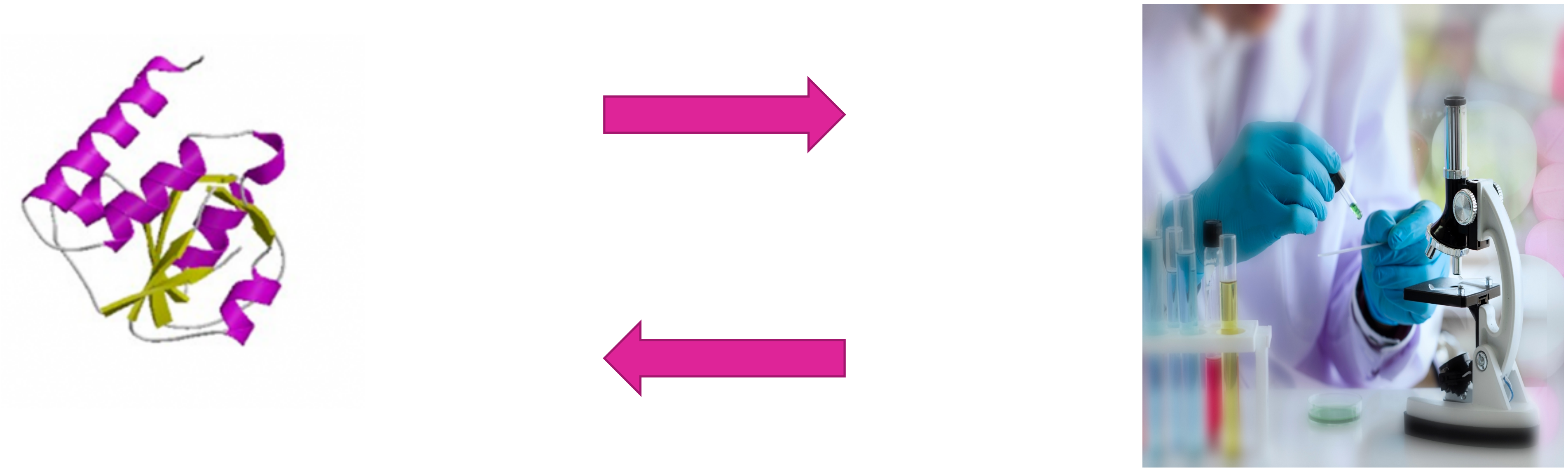




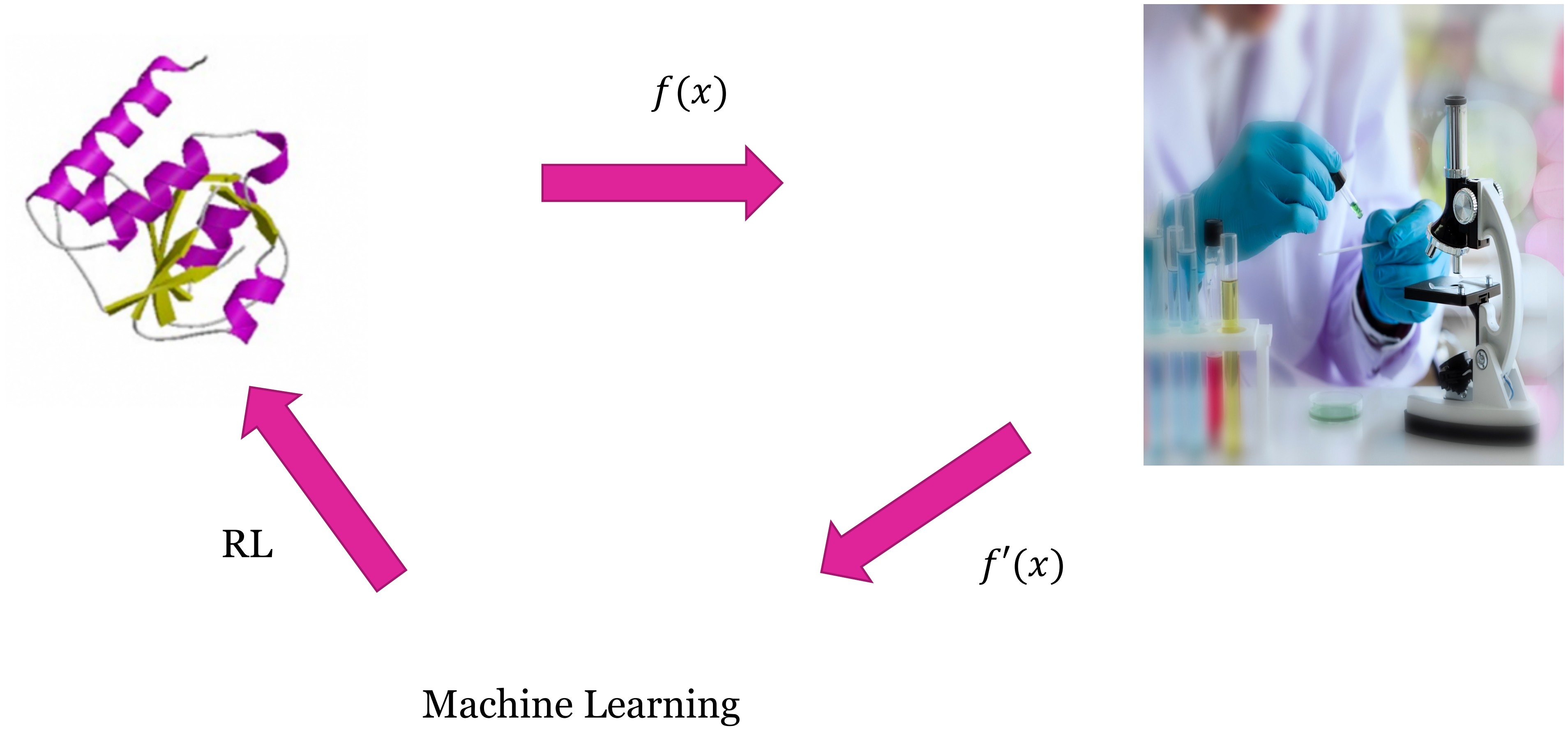
Introduction: Biological Sequence Design

- The goal of biological sequence design is to find new sequences x which optimize some oracle, typically an experimentally-measured functional property f(x).
- The current gold standard for biomolecular design is **directed evolution**, which was recently recognized with a Nobel prize (Arnold, 1998) which is a form of randomized local search.
- Wet-lab experiments are slow and expensive.

Introduction: Directed Evolution



Introduction: Directed Evolution Guided by Machine Learning



$f(x)$

$f'(x)$

RL

Machine Learning

Dyna PPO

Method: TRPO and PPO

TRPO

$$\underset{\theta}{\text{maximize}} \quad \hat{\mathbb{E}}_t\left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}\hat{A}_t\right]$$

$$\text{subject to} \quad \hat{\mathbb{E}}_t[\text{KL}[\pi_{\theta_{\text{old}}}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t)]] \le \delta.$$

PPO
with Adaptive KL Penalty Coefficient

$$\underset{\theta}{\text{maximize}}\,\hat{\mathbb{E}}_t\left[\frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}\hat{A}_t - \beta\,\text{KL}[\pi_{\theta_{\text{old}}}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t)]\right]$$

Compute $d = \hat{\mathbb{E}}_t[\text{KL}[\pi_{\theta_{\text{old}}}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t)]]$

- If $d < d_{\text{targ}}/1.5$, $\beta \leftarrow \beta/2$
- If $d > d_{\text{targ}} \times 1.5$, $\beta \leftarrow \beta \times 2$

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347, 2017.

Method: Dyna-Q

```
Dyna-Q($s$)
  Repeat
    Select and execute $a$, observe $s'$ and $r$
    Update transition: $w_T \leftarrow w_T - \alpha_T (T(s,a) - s') \nabla_{w_T} T(s,a)$
    Update reward: $w_R \leftarrow w_R - \alpha_R (R(s,a) - r) \nabla_{w_R} R(s,a)$
    $\delta \leftarrow r + \gamma \max_{a'} Q(s',a') - Q(s,a)$
    Update $Q$: $w_Q \leftarrow w_Q - \alpha_Q \delta \nabla_{w_Q} Q(s,a)$
    Repeat a few times:
      sample $\hat{s}, \hat{a}$ arbitrarily
      $\delta \leftarrow R(\hat{s},\hat{a}) + \gamma \max_{\hat{a}'} Q(T(\hat{s},\hat{a}),\hat{a}') - Q(\hat{s},\hat{a})$
      Update $Q$: $w_Q \leftarrow w_Q - \alpha_Q \delta \nabla_{w_Q} Q(\hat{s},\hat{a})$
    $s \leftarrow s'$
  Return $Q$
```

Method: Problem Formulation

- Let $f(x)$ be the function that we want to optimize.
- $x \in V^T$ a sequence of length T over a vocabulary V such as DNA nucleotides ($|V| = 4$) or amino acids ($|V| = 20$).
- Assume N experimental rounds and that B sequences can be measured per round.
- Let $D_n = \{(x, f(x))\}$ be the data acquired in round n with $|D_n| = B$.

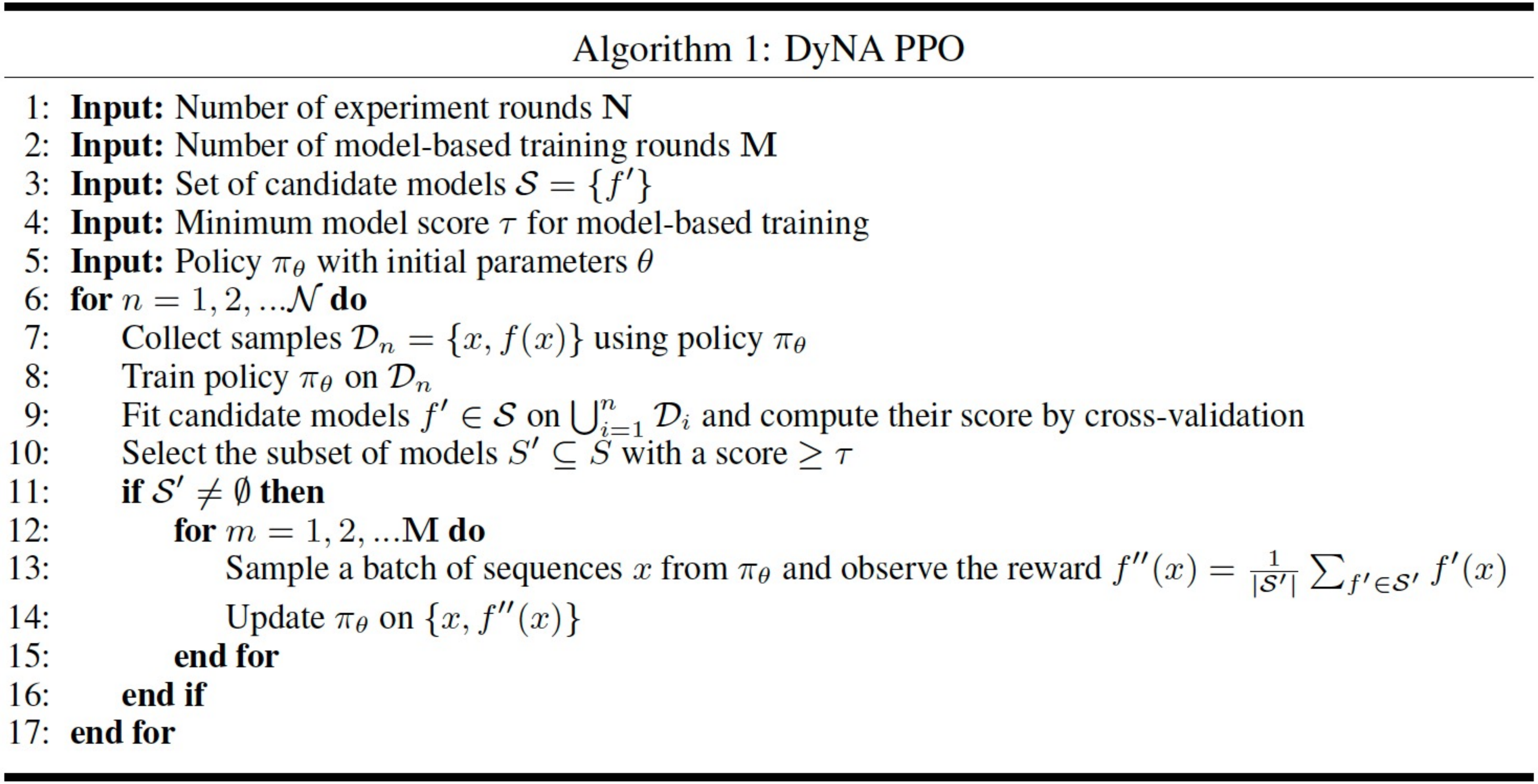
Method: State and Action Formulation

- The state and action space and reward and transition function for RL model are defined as follow:
- a_t is the token which has been chosen at timestep t, and $a_t \in V$.
- The state $s_t \ = \ a_0, \dots \ , a_{t-1}$ corresponds to the t last tokens ($S \ = \ \bigcup_{t=1 \dots T} V^t$)
- The transition function $p(s_t \ + \ 1|s_t) \ = \ s_t a_t$ is deterministic and corresponds to appending a_t to s_t.
- The reward $r(s_t, a_t)$ is zero except at the last step T, where it corresponds to the functional measurement $f(s_{T-1})$

Method: Automatic Model Tuning and Selection

- Consider a set of candidate models consist of nearest neighbor regression, Bayesian ridge regression, random forests, gradient boosting trees, Gaussian processes, and ensemble of deep neural networks.
- Automatically, tune their hyper-parameters by cross-validation.
- Evaluate models accuracy by the R^2 score and cross-validation.
- Select the models which have a R^2 score below a pre-specified threshold (τ).
- Stop model-based training as soon the the model uncertainty increases by a certain threshold.

Method: Dyna PPO



Algorithm 1: DyNA PPO

```
Input: Number of experiment rounds N
Input: Number of model-based training rounds M
Input: Set of candidate models $\mathcal{S} = \{f'\}$
Input: Minimum model score $\tau$ for model-based training
Input: Policy $\pi_\theta$ with initial parameters $\theta$
for $n = 1, 2, ...\mathcal{N}$ do
    Collect samples $\mathcal{D}_n = \{x, f(x)\}$ using policy $\pi_\theta$
    Train policy $\pi_\theta$ on $\mathcal{D}_n$
    Fit candidate models $f' \in \mathcal{S}$ on $\bigcup_{i=1}^{n} \mathcal{D}_i$ and compute their score by cross-validation
    Select the subset of models $S' \subseteq S$ with a score $\geq \tau$
    if $\mathcal{S}' \neq \emptyset$ then
        for $m = 1, 2, ...$M do
            Sample a batch of sequences $x$ from $\pi_\theta$ and observe the reward $f''(x) = \frac{1}{|\mathcal{S}'|} \sum_{f' \in \mathcal{S}'} f'(x)$
            Update $\pi_\theta$ on $\{x, f''(x)\}$
        end for
    end if
end for
```

Method: Diversity-Promoting Reward Function

In order to encourage the model to generate diverse sequences, the reward function was defined as

$$r_T = f(x) - \lambda . dens(x)$$

where $dens(x) \in \mathbb{N}^+$ is the weighted number of sequences that have been proposed in previous rounds with a distance of less than ϵ away from x, where the weight decays linearly with the distance.

They used the edit distance as distance metric and tuned the distance radius ϵ.

Expriments

They compared the performance of Dyna PPO to different existing methods in three in-silico optimization problems that were designed to simulate the behaviour of real wet-lab experiments, which were cost prohibitive for a comprehensive methodological evaluation.

Optimization performance was quantified by the cumulative maximum reward f(x) for sequences proposed up to a given round, and the area under the cumulative maximum reward curve was used to summarize one optimization trajectory as a single number.

Experiments: Optimization of Protein Contact Ising Models

Given a protein, they sought to find the amino acid sequence that minimizes the energy of the Ising model parameterized by its structure.

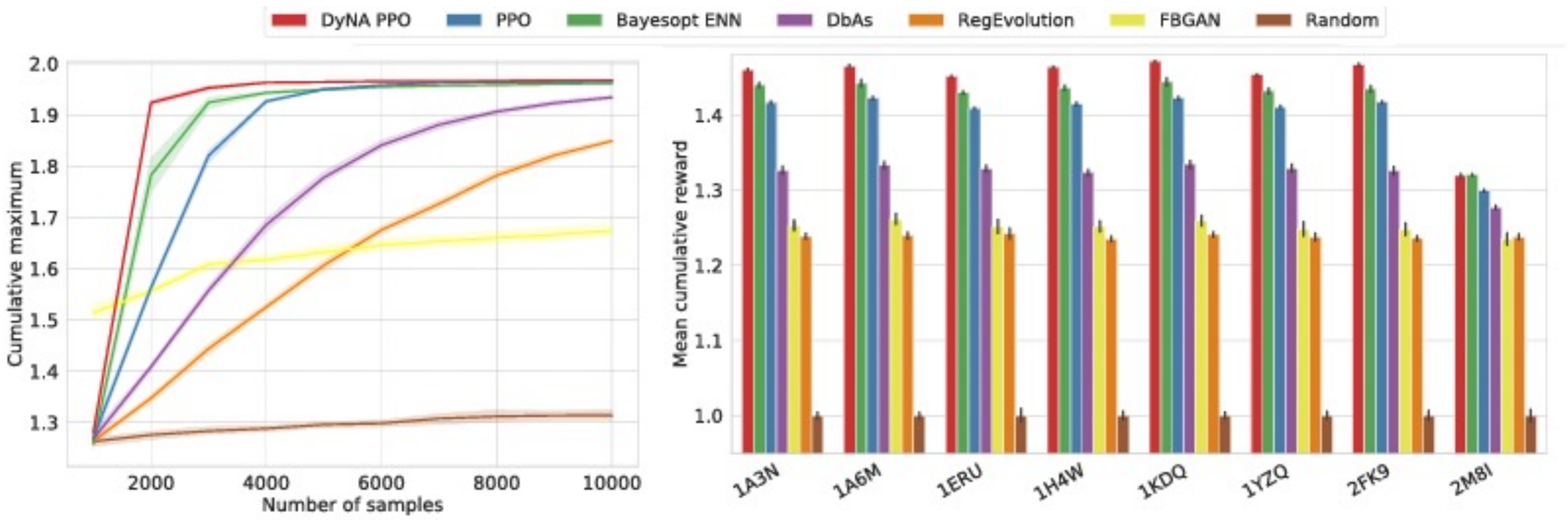


Figure 1: **Comparison of methods on optimizing the energy of protein contact Ising models**. Left: the cumulative maximum reward depending on the number of rounds for one selected protein target (1A3N). Right: the mean cumulative maximum relative to *Random* for alternative protein targets. Since $f(x)$ can be well-approximated by a model trained on few examples, model-based training (DyNA PPO) results in a clear improvement over model-free training (PPO).

Experiments: Optimization of Protein Contact Ising Models

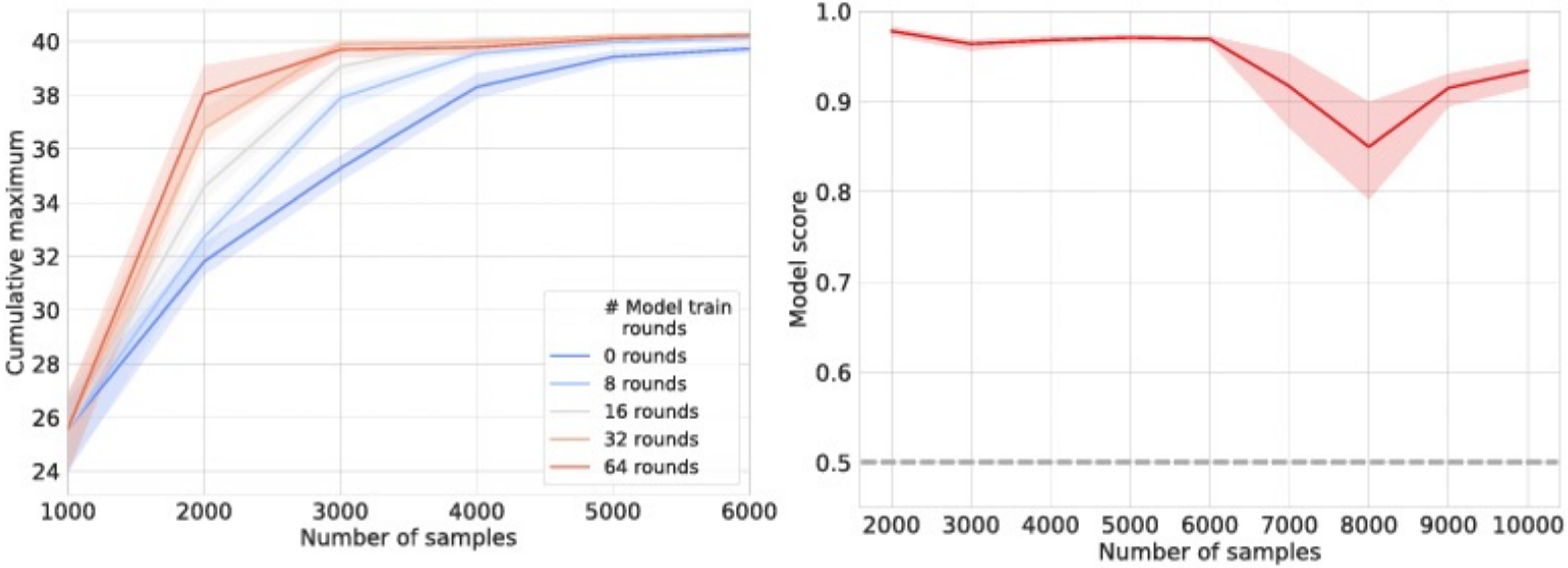
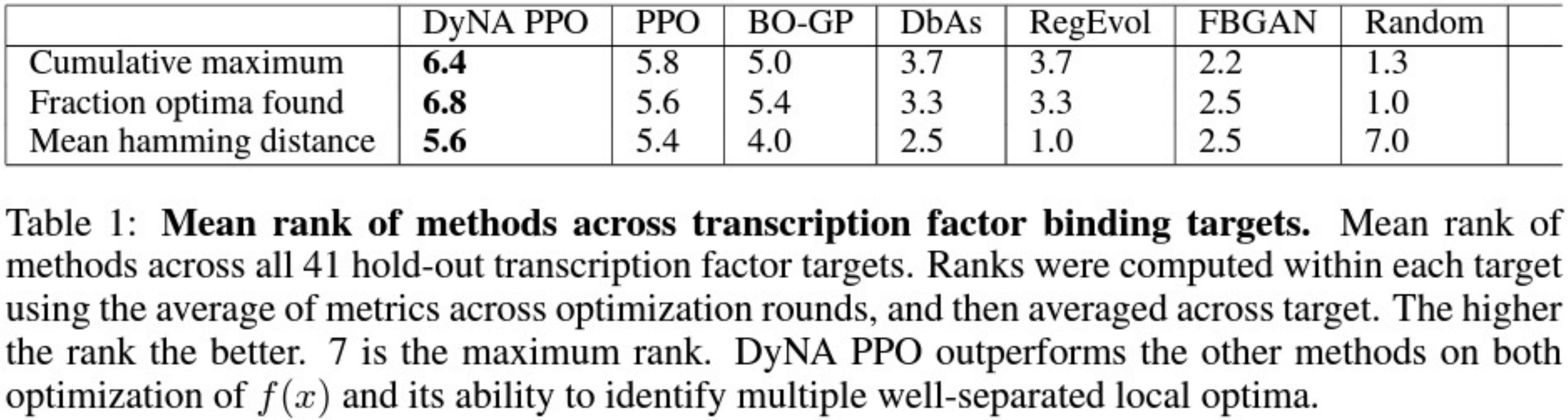


Figure 2: **Analysis of the performance of DyNA PPO on the Ising model.** Left: Performance of DyNA PPO depending on the number of inner policy optimization rounds using the surrogate model. Using 0 rounds corresponds to PPO training. Since the surrogate model is sufficiently accurate, it is useful to perform many inner loop optimization rounds before querying $f(x)$ again. Right: the R^2 of the surrogate model. Since it is always above the threshold for model-based training (0.5; dashed line), it is always used for training.

Experiments: Optimization of Transcription Factor Binding Sites

Designing length-8 DNA sequences (search space = 4^8).



	DyNA PPO	PPO	BO-GP	DbAs	RegEvol	FBGAN	Random
Cumulative maximum	**6.4**	5.8	5.0	3.7	3.7	2.2	1.3
Fraction optima found	**6.8**	5.6	5.4	3.3	3.3	2.5	1.0
Mean hamming distance	**5.6**	5.4	4.0	2.5	1.0	2.5	7.0

Table 1: **Mean rank of methods across transcription factor binding targets.** Mean rank of methods across all 41 hold-out transcription factor targets. Ranks were computed within each target using the average of metrics across optimization rounds, and then averaged across target. The higher the rank the better. 7 is the maximum rank. DyNA PPO outperforms the other methods on both optimization of $f(x)$ and its ability to identify multiple well-separated local optima.

Experiments: The Effect of Exploration Bonus

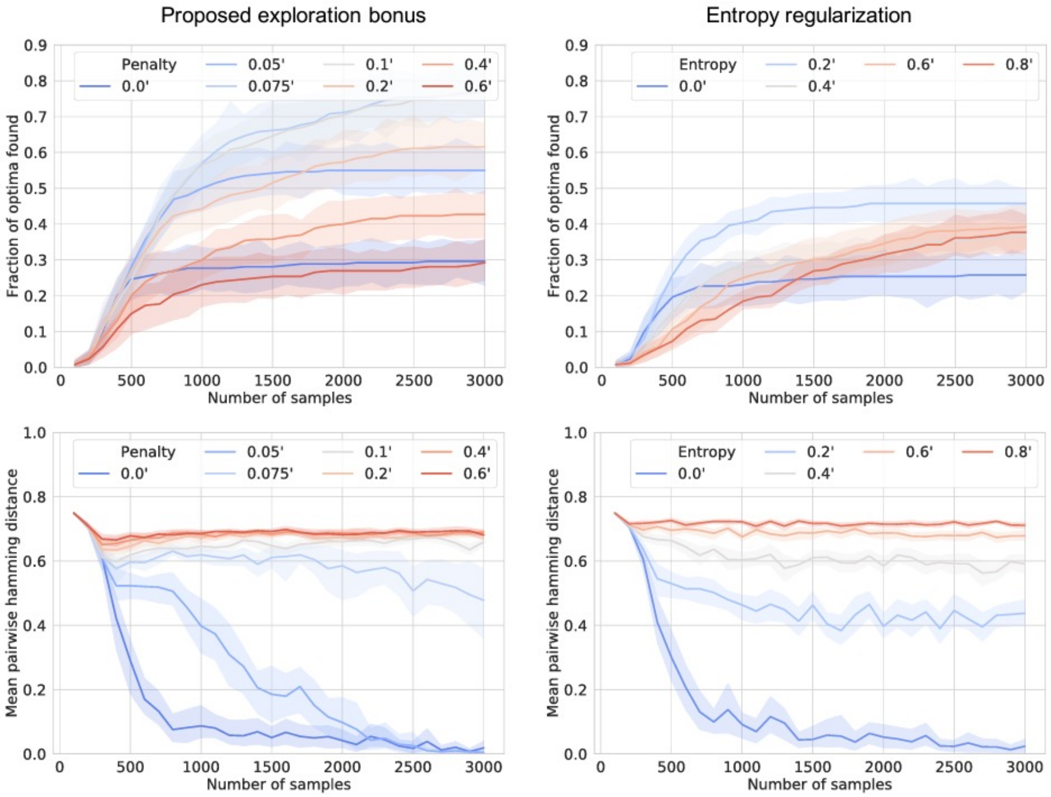


Figure 5: **Comparison of the proposed exploration bonus vs. entropy regularization on the transcription factor task**. Left: performance with exploration bonus as a function of the density penalty λ (Section 2.4). Right: performance of entropy regularization as a function of the regularization strength. The top row shows that PPO finds about 80% of local optima with a relatively mild density penalty of $\lambda = 0.1$, whereas only about 45% local optima are found when using entropy regularization. The bottom row shows that varying the density penalty enables to control the sequence diversity quantified by the mean pairwise hamming distance between sequences.

Conclusion: Contributions

In summary, the contributions of this paper are as follows:

- They provided a model-based RL algorithm, DyNA PPO, and demonstrated its effectiveness in performing sample efficient batched black-box function optimization.
- They used an automatic model tuning and selection in order to have a reliable reward function.
- They propose a visitation-based exploration bonus and showed that it is more effective than entropy-regularization in identifying multiple local optima.

Conclusion: Future Extensions

The authors suggested that the large-batch, low-round optimization setting described here may well be of general interest, and that model-based RL may be applicable in other scientific and economic domains.

UNIVERSITY OF
WATERLOO

FACULTY OF MATHEMATICS