DRAKES Report

Fine-tuned models tend to have faster inference times compared to guidance based inference techniques. Fine tuning in **D**irect **R**eward b**A**c**K**propagation with gumb**E**l **S**oftmax trick (DRAKES) is reinforcement learning based. Classifier-free fine-tuning constructs conditional generative models which are applicable in reinforcement learning by conditioning on high reward values. However, studies have shown that this method of conditioning in continuous diffusion is suboptimal because high-reward samples are rare.

The pretrained model in DRAKES is a discrete diffusion model based on a continuous-time Markov chain (CTMC). In the forward process, we gradually mask an amino acid sequence so that it becomes completely masked. At each iteration the current state of each amino acid either becomes masked or stays the same by some state change probabilities. Once it is masked, it stays masked. In the forward process, we change the state by some categorical distribution and in the reverse process we predict the state changes that were made at each time step.

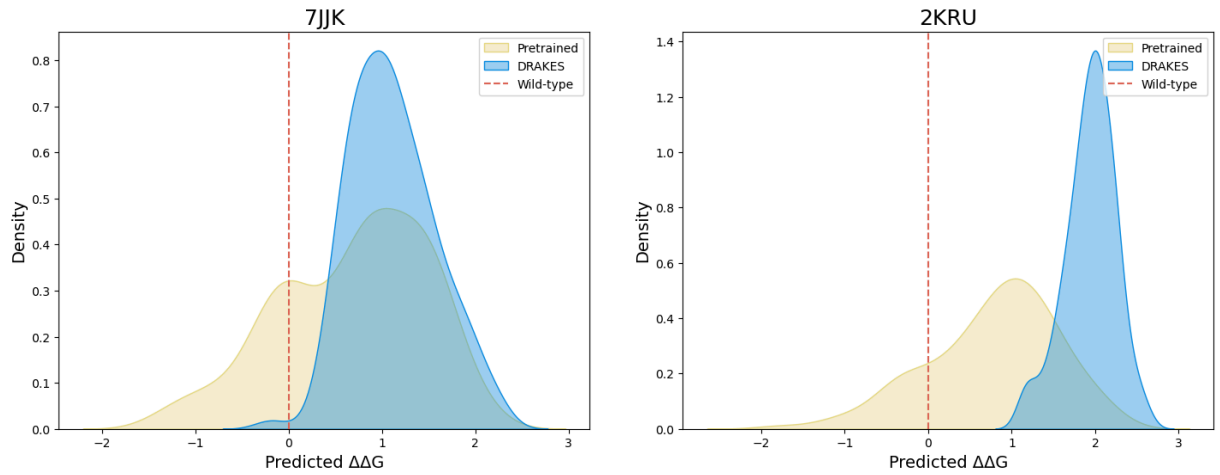
Once we have formed the pretrained model, which represents the probability distribution of proteins, we fine-tune to optimize for some reward. We also penalize the objective with a KL term so that our fine-tuned distribution still produces natural-like samples. This creates a reinforcement learning problem: maximized future reward with penalized KL divergence. This problem is solved by iteratively sampling via the reverse process and updating the model according to the objective function. The objective function is approximated here to allow for a gradient to be taken during optimization.

During the sampling step of the iterative fine-tuning, we wish to sample from a categorical distribution, representing the state transition. To make this step differentiable with respect to the model parameters, we reduce the sampling step to a Gumbel-softmax operation with some temperature parameter. As the temperature approaches zero, we converge to a sample from the exact categorical distribution. After sampling, we back propagate via stochastic gradient descent.

The algorithm is evaluated by comparing its performance on DNA sequences and protein sequences with baselines. In this report, we focus on protein generation evaluation. For this evaluation, we are given a pre-trained inverse folding model that generates sequences conditioned on their 3D structure. The reward oracles are trained on a protein stability dataset which includes stability measurements (Gibbs free energy change - ΔG) for variants of particular proteins. ΔG is the free energy difference between the folded and unfolded state of the protein. If this value is more negative, the folded protein is more stable. By finding the difference between the ΔG of the wild-type proteins and the ΔG of their variants, we train the reward oracles to predict the $\Delta\Delta G$ between a variant protein and its corresponding wild-type. Two oracles are trained, one for the fine-tuning algorithm and one for final evaluation.

The metrics used to compare the algorithms are predicted-ΔΔG and self-consistency RMSD (scRMSD). The ΔΔG values are predicted via our reward oracle trained for evaluation. The scRMSD values are computed by first predicting the structures of the generated sequences with ESMFold and then calculating the RMSD relative to the wild-type structure. A successful protein generation is defined by if the ΔΔG is greater than 0 and the scRMSD is less than 2. Below we compare results, using a seed of 0, between the pretrained model and DRAKES.

Method	Predicted ΔΔG (median)	ΔΔG > 0 (%)	scRMSD (median)	scRMSD < 2 (%)	Success Rate (%)
Pre-trained	-0.507	36.7	**0.834**	90.0	**34.4**
DRAKES	**1.084**	**86.6**	0.913	**92.4**	**79.4**



We want a model that generates proteins with positive ΔΔG as this means the proteins will be more stable than the wild-type proteins. We also want the proteins to have a lower scRMSD since we want their folded structures to be similar to the wild-type proteins. The table shows that DRAKES tends to generate more stable proteins than the wild-type while remaining structurally similar enough to the wild-type proteins since the success rate is much higher than the pre-trained model. The density distributions of the generated proteins are conditioned on the backbone structure of the 7JJK and 2KRU proteins. For reference, the wild-type protein is displayed at a ΔΔG of 0.We again see that the sequences generated by DRAKES appear to be more stable than the pretrained model and the wild-type baseline as they tend to have higher ΔΔG and thus better stability.