Conservative Objective Models For Offline Model-Based Optimization

Brandon Trabucco[1] * , Aviral Kumar[1] * , Xinyang Geng[1] , Sergey Levine[1,2]

[1]UC Berkeley [2] Google AI Research (* Equal Contribution)

Problem Statement

We propose **Conservative Objective Models**, a simple method for Offline Model-Based Optimization (MBO) that learns a robust model of a black-box score function using supervised learning.

The goal of MBO is to find an optimal design $\mathbf{x}^*$ that maximizes the value of an unknown black-box score function.

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} f(\mathbf{x})$$

Protein Designs | Robot Designs | Neural Net Designs

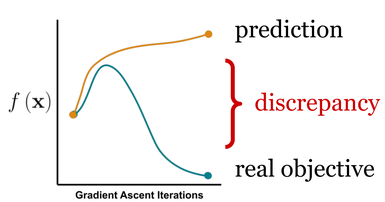
Offline means that a fixed dataset of black-box function evaluations is provided, and no new function evaluations may be collected.

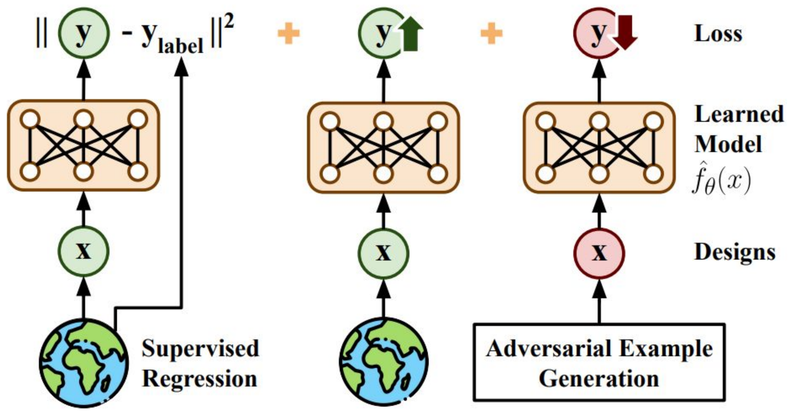
Overestimation in Offline MBO

Even if we can train accurate predictive models, optimizing them is hard. If we directly optimize $\hat{f}(\mathbf{x})$, the optimizer can get **"fooled"** by erroneous predictions of the model and produce adversarial inputs.

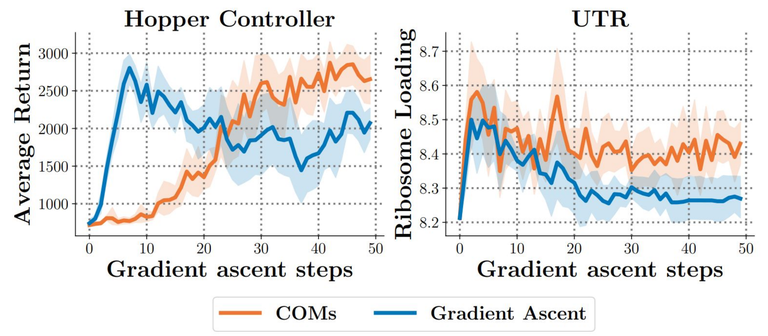
original

MBO output





COMs Are Stable Optimizers



- COMs reach solutions that remain at higher performance for longer, indicating that COMs are less sensitive to varying numbers of gradient ascent steps (a hyperparameter) performed during optimization.
- The naive gradient ascent optimizer using a single predictive model tends to produce degenerate solutions, and performance under the real objective eventually collapses to a low number.

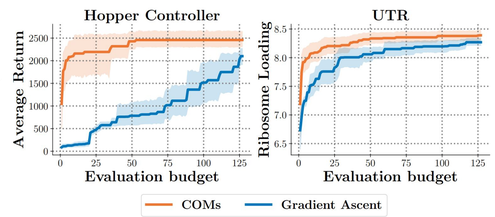
Experimental Results

	GFP	TF Bind 8	UTR	# Optimal	Norm. avg. perf.
$\mathcal{D}$ **(best)**	0.789	0.439	0.593		
Auto. CbAS	**0.865 ± 0.000**	0.910 ± 0.044	0.691 ± 0.012	1 / 7	0.687
CbAS	**0.865 ± 0.000**	0.927 ± 0.051	**0.694 ± 0.010**	3 / 7	0.699
BO-qEI	0.254 ± 0.352	0.798 ± 0.083	0.684 ± 0.000	0 / 7	0.629
CMA-ES	0.054 ± 0.002	0.953 ± 0.022	**0.707 ± 0.014**	2 / 7	0.674
Grad.	0.864 ± 0.001	**0.977 ± 0.025**	**0.695 ± 0.013**	3 / 7	0.750
Grad. Min	0.864 ± 0.000	**0.984 ± 0.012**	**0.696 ± 0.009**	3 / 7	0.829
Grad. Mean	0.864 ± 0.000	**0.986 ± 0.012**	0.693 ± 0.010	2 / 7	0.852
MINs	**0.865 ± 0.001**	0.905 ± 0.052	**0.697 ± 0.010**	**4 / 7**	0.745
REINFORCE	**0.865 ± 0.000**	0.948 ± 0.028	0.688 ± 0.010	1 / 7	0.541
COMs (Ours)	*0.864 ± 0.000*	*0.945 ± 0.033*	**0.699 ± 0.011**	**4 / 7**	**0.985**

	Superconductor	Ant Morphology	D'Kitty Morphology	Hopper Controller	
$\mathcal{D}$ **(best)**	0.399	0.565	0.884	1.0	
Auto. CbAS	0.421 ± 0.045	0.882 ± 0.045	0.906 ± 0.006	0.137 ± 0.005	
CbAS	**0.503 ± 0.069**	0.876 ± 0.031	0.892 ± 0.008	0.141 ± 0.012	
BO-qEI	0.402 ± 0.034	0.819 ± 0.000	0.896 ± 0.000	0.550 ± 0.118	
CMA-ES	0.465 ± 0.024	**1.214 ± 0.732**	0.724 ± 0.001	0.604 ± 0.215	
Grad.	**0.518 ± 0.024**	0.293 ± 0.023	0.874 ± 0.022	1.035 ± 0.482	
Grad. Min	**0.506 ± 0.009**	0.479 ± 0.064	0.889 ± 0.011	1.391 ± 0.589	
Grad. Mean	**0.499 ± 0.017**	0.445 ± 0.080	0.892 ± 0.011	1.586 ± 0.454	
MINs	0.469 ± 0.023	**0.913 ± 0.036**	**0.945 ± 0.012**	0.424 ± 0.166	
REINFORCE	0.481 ± 0.013	0.266 ± 0.032	0.562 ± 0.196	-0.020 ± 0.067	
COMs (Ours)	0.439 ± 0.033	**0.944 ± 0.016**	**0.949 ± 0.015**	**2.056 ± 0.314**	

- COMs mitigate finding adversarial solutions to optimization problems by learning conservative predictive models.
- COMs are **simple** to tune, require training only a single predictive model of the objective, and **shared hyperparameters** uniformly across all discrete and continuous tasks respectively.
- COMs are **16% better** than the next best Offline MBO method.

COMs are resilient to small evaluation budgets, and are nearly invariant down to budgets of size **50**, consistently producing high performing $\mathbf{x}^*$.

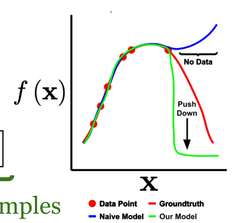


Conservatism Fixes Overestimation

COMs learn a model that underestimates the actual value of unseen designs, preventing the optimizer from being **"fooled"**.

$$\mathcal{L} = \mathbb{E}\left[(\hat{f}_\theta(\mathbf{x}) - y)^2\right] + \alpha \mathbb{E}\left[\hat{f}_\theta(\hat{\mathbf{x}}_T) - \hat{f}_\theta(\mathbf{x})\right]$$

Penalty On Adversarial Examples



COMs Algorithm

- **COMs** operates in two phases: first optimizing the learned model with conservative regression, then optimizing to get $\mathbf{x}^*$.
- During phase one, we sample initial $\mathbf{x}$ from the dataset, and obtain solutions $\mathbf{x}^*$ using standard gradient ascent. The model is trained to underestimate the actual value of these solutions $\mathbf{x}^*$.
- During phase two, we sample the most promising designs in the training set, and optimize them for T_{eval} iterations.

Future Work & Open Problems

- COMs could be further improved by incorporating generative models that explicitly model the data manifold during optimization.
- When using COMs, especially on small datasets, "overfitting" is an issue, and mitigating it can improve the reliability of Offline MBO.
- A deeper understanding of how neural network models extrapolate could help explain why and how adversarial examples are found, and could result in more powerful optimization schemes.