Bayesian Modeling and Uncertainty Quantification for Learning to Optimize: What, Why, and How

Yuning You[1], Yue Cao[1], Tianlong Chen[2], Zhangyang Wang[2], Yang Shen[1]
[1]Texas A&M University, [2]University of Texas at Austin

Background

- Computational models of many real-world applications involve optimizing **non-convex** objective functions.
- Being able to quantify solution **uncertainty** (UQ) provides calibration of the solution quality and usefulness.
- Inconspicuous attention was paid to the uncertainty arising from the **optimizer** (why).
- ***Question***: What defines the optimizer uncertainty? How to enable optimizer UQ during optimization?

Technical Approaches

- Notations. Decision variable $\boldsymbol{x}$, objective $f(\boldsymbol{x})$, gradient $\nabla f(\boldsymbol{x}_t)$ at time t, trajectories $\boldsymbol{z}_t$ up to time t, ϕ-parametrized optimizer $g(\boldsymbol{z}_t; \phi)$.
- Optimization via iterative algorithms:

$$\min_{\phi} \sum_{t=1}^{T} w_t f(\mathbf{x}_t), \quad \text{with} \quad \mathbf{x}_{t+1} = \mathbf{x}_t - g(\mathbf{z}_t; \phi),\ t = 0, \ldots, T-1,$$

- We characterize **optimization uncertainty** as:

Definition 1 (Optimizer Uncertainty) *Let $\mathcal{G}$ be the algorithmic space, where each point $g \in \mathcal{G}$ is an optimizer (omitting ϕ parameterization). We assume that*

1. *g has a prior distribution $p(g)$;*
2. *Its likelihood can be interpreted as $p(\mathbf{z}_t|\mathbf{z}_{t_0}, g) = \prod_{i=t_0+1}^{t} p(\mathbf{x}_i|\mathbf{z}_{i-1}, g)$, $\forall t_0 < t$.*

- Thus, posterior via Bayes theorem: $p(g|\mathbf{z}_t) \propto p(g) \prod_{i=1}^{t} p(\mathbf{x}_i|\mathbf{z}_{i-1}, g)$.
- Prior works use hyper-parameters to parametrize classical optimizers, while it covers a relatively restricted optimizer space.
- We leverage **learning to optimize** techniques (L2O) [1,2] for parametrization, which provides a more comprehensive space coverage.
- We next develop **end-to-end** training pipeline with variational inference (UA-L2O), by optimizing:

$$-\,\mathrm{KL}[q(\phi;\boldsymbol{\theta})||p(\phi|\mathbf{z}_T)] = -\mathbb{E}_{\phi\sim q(\phi;\boldsymbol{\theta})} \sum_{t=1}^{T} f(\mathbf{x}_t^{\phi}) - \mathrm{KL}[q(\phi;\boldsymbol{\theta})||p(\phi)].$$

References

[1] Marcin Andrychowicz et al., "Learning to learn by gradient descent by gradient descent", NeurIPS'16. [2] Ke Li & Jitendra Malik, "Learning to Optimize", ICLR'17.

Experiments

- We first evaluate on the **test function** benchmarks of Rastrigin, Ackley and Griewank (Figure 3).
- Results echo our conjecture that, UA-L2O owns better uncertainty awareness, and it could sometimes improve optimization performances.
- We next apply to an application of **data privacy attack**, which critically needs UQ (Table 1).
- We observe UQ performance of UA-L2O stands out against competitors, although not as dominant as it does to competitors in test functions. The superior optimization performance could benefit from its calibration (Figure 4).
- We lastly examine using a bioinformatics application of **protein docking** (Figure 6). Advantages on both UQ and optimization are shown.

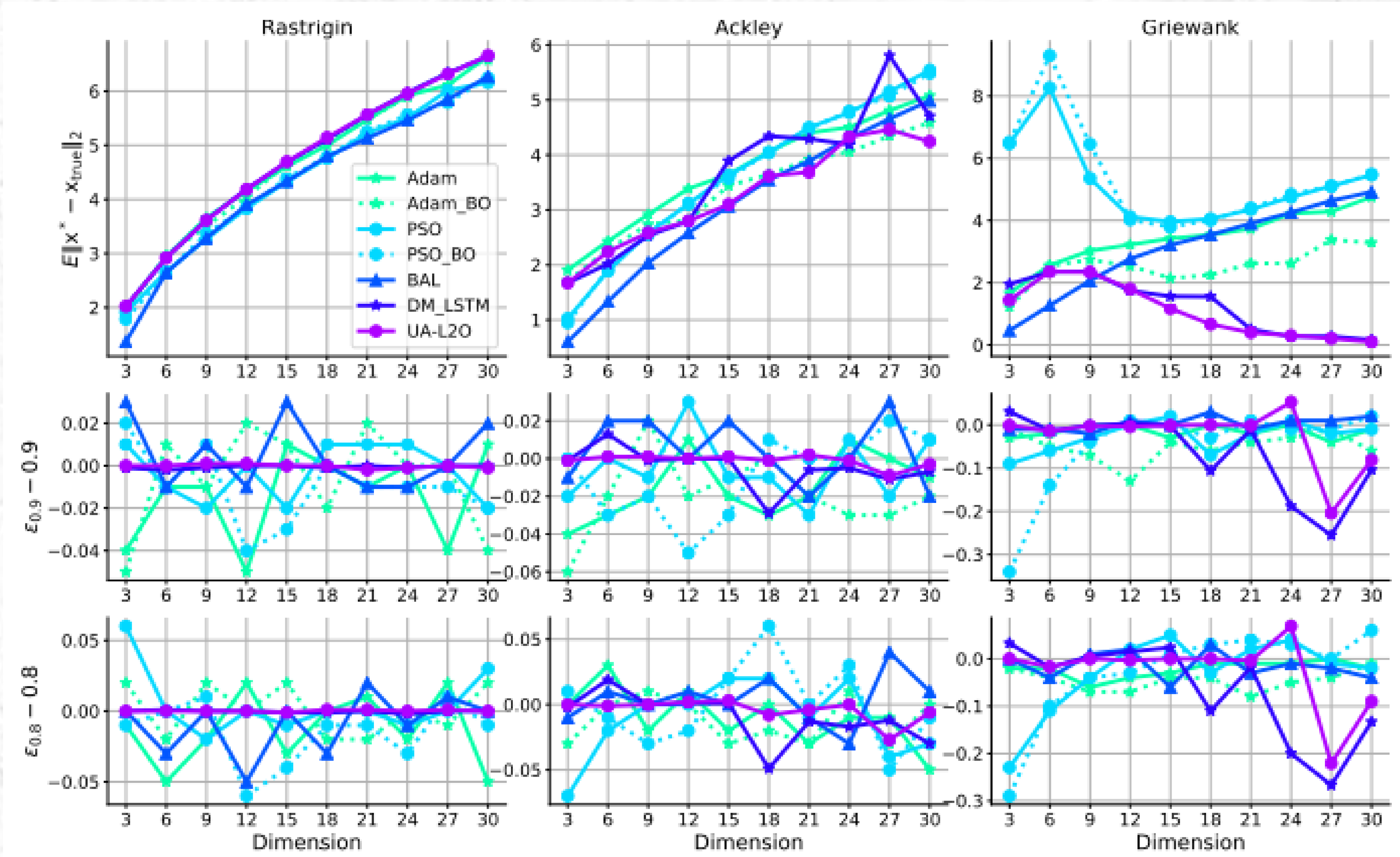

Figure 3: Optimization and uncertainty performance of different methods in three non-convex test functions. Different column represents different functions, and each row stands for: (i) 1st row is for the non-intended optimization performance, the lower the better; (ii) 2nd & 3rd rows, the most important metric for the intended uncertainty calibration, are for the precision of the estimated confidence, lower values indicating more accurate posterior estimation. The corresponding confidence intervals are shown in Appendix C.

Table 1: Optimization and uncertainty performance of different methods in genetic data privacy attack.

Method	$\mathbb{E}\|\mathbf{x}^* - \mathbf{x}_{\text{true}}\|_2$	$\|\epsilon_{0.9} - 0.9\|$	$\|\epsilon_{0.8} - 0.8\|$	$r_{0.9}$	$r_{0.8}$
Adam	0.39	0.90	0.80	0.08	0.06
Adam_BO	0.38	0.90	0.80	0.08	0.05
Adam_lr_Ensemble	0.35	0.90	0.80	0.01	0.009
Adam_Noisy_Gradient	0.57	0.90	0.80	0.01	0.01
PSO	0.54	0.08	0.10	0.54	0.34
PSO_BO	0.53	0.06	0.15	0.48	0.42
BAL	0.52	0.90	0.80	0.01	0.01
DM_LSTM	0.34	0.09	0.77	0.09	0.02
UA-L2O	**0.30**	0.25	0.27	0.06	0.04

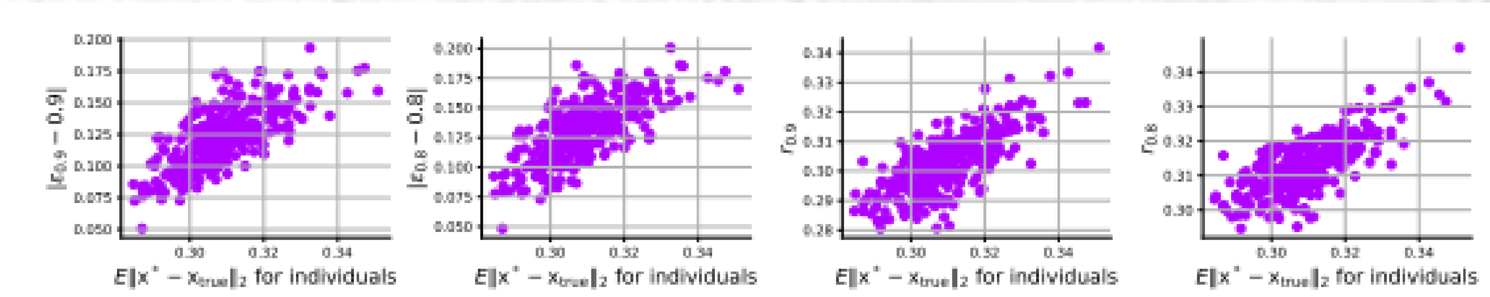

Figure 4: Optimization performance versus UQ results ($\epsilon_{0.9}, \epsilon_{0.8}, r_{0.9}$ and $r_{0.8}$) of UA-L2O for 318 test samples in data privacy attack.

Table 2: Optimization and uncertainty performance of BAL and UA-L2O in protein docking.

Target PDB model (docking difficulty)	$\mathbb{E}\|\mathbf{x}^* - \mathbf{x}_{\text{true}}\|_2$ (Å) BAL	UA-L2O	$\mathbb{E}\|\mathbf{x}^* - \mathbf{x}_{\text{true}}\|_2 \in [\text{lb}_{0.9}, \text{ub}_{0.9}]$? BAL	UA-L2O	$\text{ub}_{0.9} - \text{lb}_{0.9}$ (Å) BAL	UA-L2O
1AHW_3 (easy)	1.89	**1.11**	No	Yes	2.20	0.79
1AK4_7 (easy)	2.45	**1.13**	Yes	Yes	1.93	1.11
3CPH_7 (medium)	3.89	**3.11**	No	No	1.70	1.62
1HE8_3 (medium)	3.05	**1.42**	Yes	Yes	2.24	1.32
1JMO_4 (difficult)	**1.45**	1.87	No	No	2.90	0.67

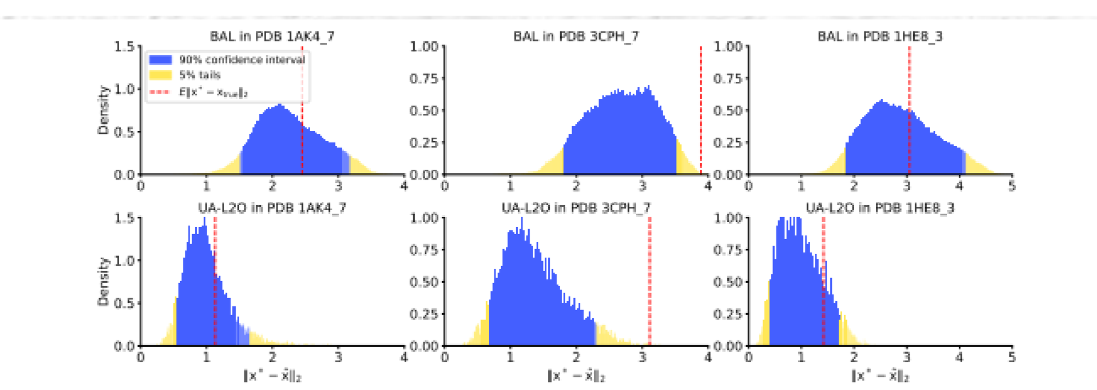

Figure 6: Estimated posterior distributions, confidence intervals and ground truth solutions for cases 1AK4_7, 3CPH_7 and 1HE8_3 in protein docking.

Contact: {yuning.you,cyppsp,yshen}@tamu.edu, {tianlong.chen,atlaswang}@utexas.edu