Learning to Optimize in Swarms

Yue Cao [1], Tianlong Chen [2], Zhangyang Wang [2] and Yang Shen [1]

[1]Department of Electrical and Computer Engineering, and [2]Department of Computer Science and Engineering,
Texas A&M University, College Station, TX 77843, United States.

Abstract

- **Learning to optimize** has emerged as a powerful framework for various optimization tasks.
- Current such "meta-optimizers" often learn from the space of continuous optimization algorithms that are **point-based** and **uncertainty-unaware**.
- We learn in an extended space of **both** point-based and **population-based** optimization algorithms.
- We incorporate the **Boltzmann-shaped posterior** into meta-loss to guide the search in the algorithmic space and balance the exploitation-exploration trade-off.
- Empirical results over non-convex test functions and the **protein docking** application demonstrate that this new meta-optimizer outperforms existing competitors.

Methods

- **Updating Rules:** Iterative optimization algorithms, either point-based or population-based, have a common generic expression of update formulas:

$$\boldsymbol{x}^{t+1} = \boldsymbol{x}^t + \delta \boldsymbol{x}^t$$

The update is often a function $g(\cdot)$ of the historic sample values, objective values, and gradients. For instance, in particle swarm optimization (PSO), we have

$$\begin{aligned}\delta \boldsymbol{x}_j^t &= g(\{\boldsymbol{x}_j^\tau, f(\boldsymbol{x}_j^\tau), \nabla f(\boldsymbol{x}_j^\tau)\}_{j=1,\tau=1}^{k,t}) \\ &= w\delta \boldsymbol{x}_j^{t-1} + r_1(\boldsymbol{x}_j^t - \boldsymbol{x}_j^{t*}) + r_2(\boldsymbol{x}_j^t - \boldsymbol{x}^{t*})\end{aligned}$$

In our approach, we parameterize the update rule $g(\cdot)$ through RNN, and introduce **intra-** and **inter-particle attention mechanisms**:

$$g_i(\cdot) = \mathrm{RNN}_i(\alpha_i^{\mathrm{inter}}(\{\alpha_j^{\mathrm{intra}}(\{\boldsymbol{S}_j^\tau\}_{\tau=1}^t)\}_{j=1}^k), \boldsymbol{h}_i^{t-1})$$

- **Population-based and Point-based Features:** Inspired from both point- and population-based algorithms, we choose the following four features for particle i at iteration t:
  - gradient: $\nabla f(\boldsymbol{x}_i^t)$
  - momentum: $\boldsymbol{m}_i^t = \sum_{\tau=1}^t (1-\beta)\beta^{t-1}\nabla f(\boldsymbol{x}_i^\tau)$
  - velocity: $\boldsymbol{v}_i^t = \boldsymbol{x}_i^t - \boldsymbol{x}_i^{t*}$
  - attraction: $\frac{\sum_j (e^{-\alpha d_{ij}^2}(\boldsymbol{x}_i^t - \boldsymbol{x}_j^t))}{\sum_j e^{-\alpha d_{ij}^2}}$, for all j that $f(\boldsymbol{x}_j^t) < f(\boldsymbol{x}_i^t)$. α is the hyperparameter and $d_{ij} = ||\boldsymbol{x}_i^t - \boldsymbol{x}_j^t||_2$.
- **Loss Function:** In order to balance the exploration-exploitation tradeoff, we combine the cumulative regret and the entropy of the posterior over the global optimum:

$$\ell_f(\boldsymbol{\phi}) = \sum_{t=1}^T \sum_{j=1}^k f(\boldsymbol{x}_j^t) + \lambda h(p\left(\boldsymbol{x}^* | \bigcup_{t=1}^T D_t)\right),$$

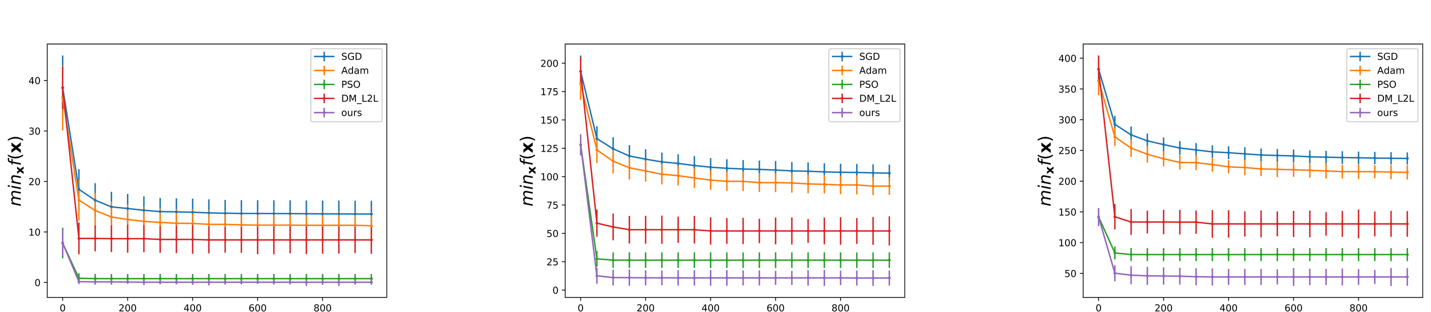
where the posterior is a **Boltzmann distribution** [3]:

$$p\left(\boldsymbol{x}^* | \bigcup_{t=1}^T D_t\right) \propto \exp(-\rho \hat{f}(\boldsymbol{x}))$$

Test Function Results

LOIS outperforms DM_LSTM [1] and hand-engineered algorithms for non-convex Rastrigin functions:

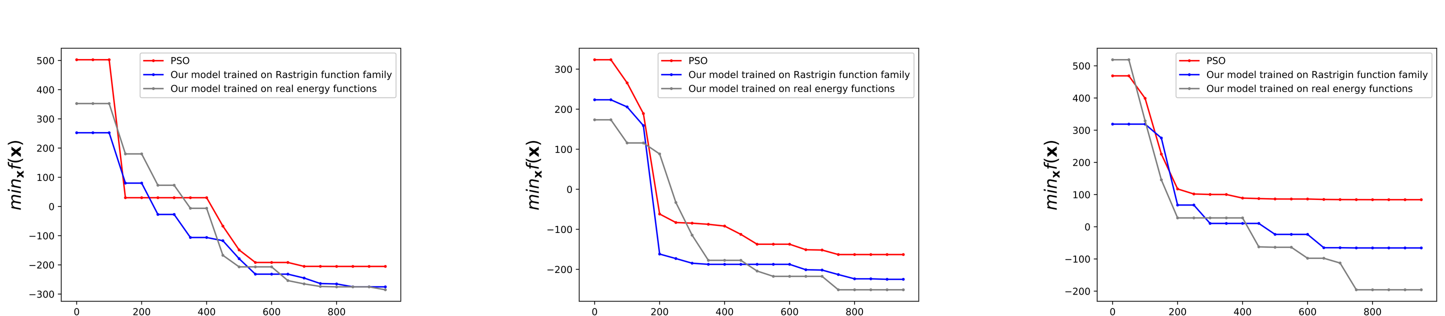
$$f(\boldsymbol{x}) = \sum_{i=1}^n x_i^2 - \sum_{i=1}^n \alpha \cos(2\pi x_i) + \alpha n$$



(a) 2D (b) 10D (c) 20D

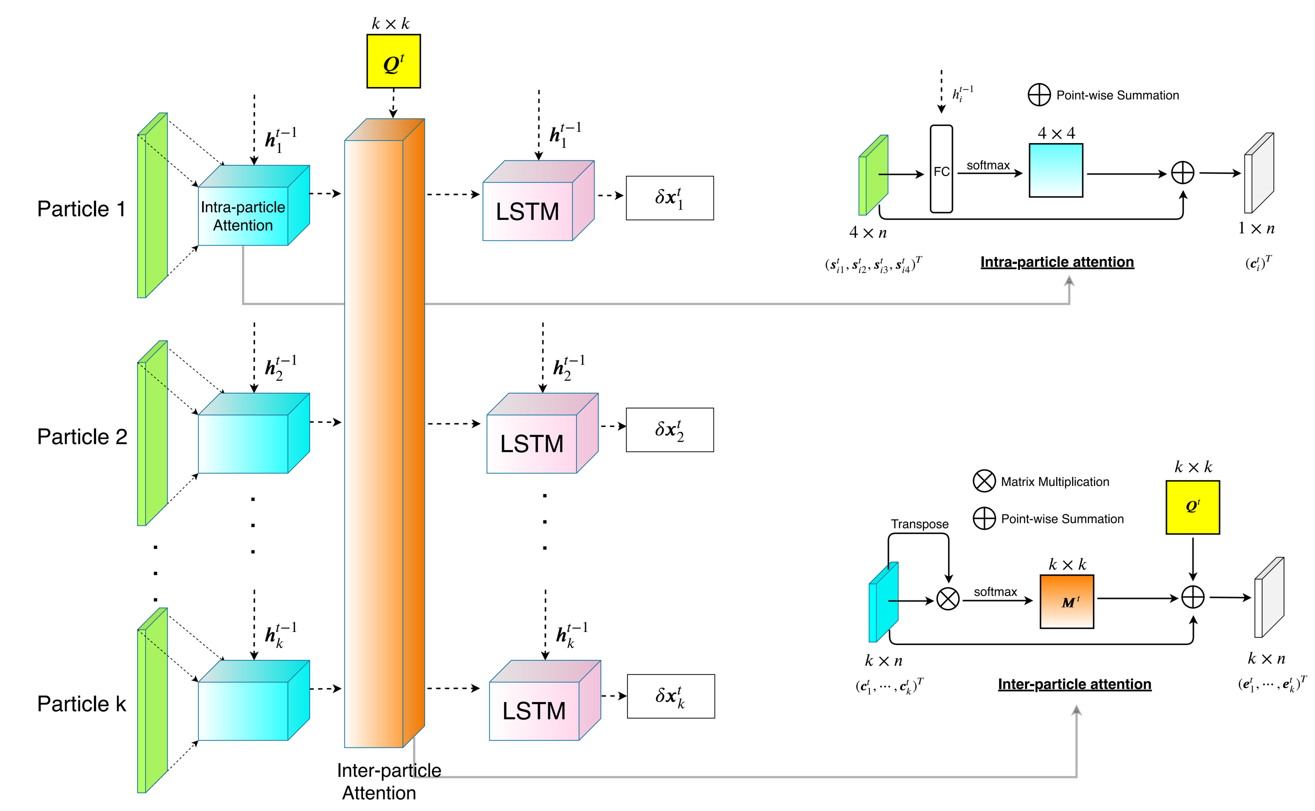
Protein Docking Results

Ab initio protein docking represents a major challenge for optimizing a noisy and costly function in a high-dimensional space [3]. We parameterize the search space as $\mathbb{R}^{12}$ as in [3]. The final $f(\boldsymbol{x})$ is fully differentiable and the search space is $\boldsymbol{x} \in \mathbb{R}^{12}$.
LOIS outperforms PSO in energy scores for three protein-protein pairs of various difficulty levels.



(a) Easy: 1AY7 (b) Medium: 2HRK (c) Difficult: 2C0L

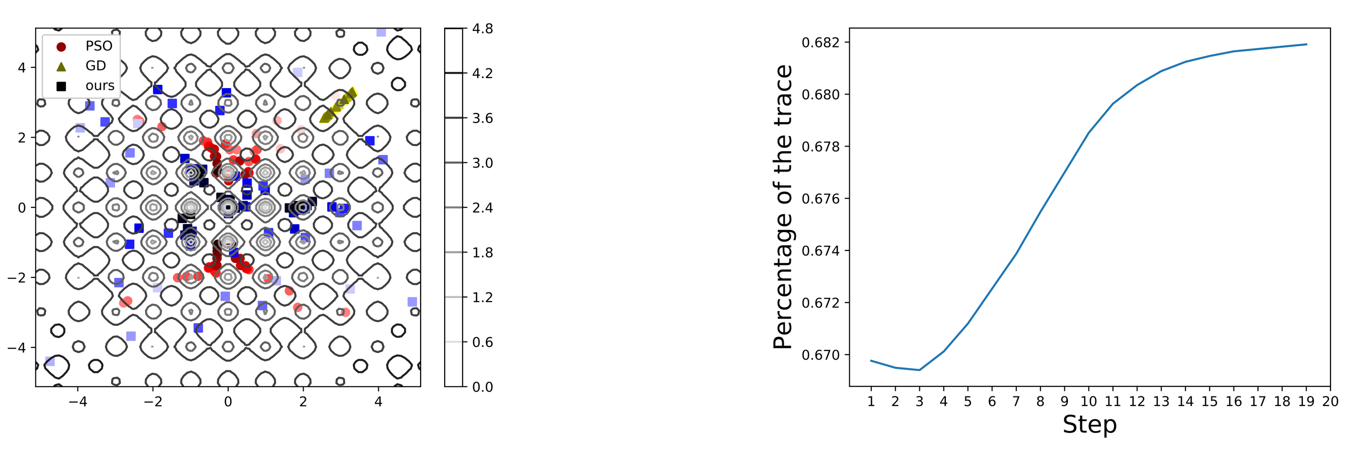
Overall architectures and attention modules



- **Intra-particle attention**:
$b_{ij}^t = \boldsymbol{v}_a^T \tanh(\boldsymbol{W}_a \boldsymbol{s}_{ij}^t + \boldsymbol{U}_a \boldsymbol{h}_{ij}^t)$, $p_{ij}^t = \frac{\exp(b_{ij}^t)}{\sum_{r=1}^4 \exp(b_{ir}^t)}$, $\boldsymbol{c}_i^t = \sum_{r=1}^4 p_{ir}^t \boldsymbol{s}_{ir}^t$
- **Inter-particle attention**: $\boldsymbol{e}_j^t = \gamma \sum_{r=1}^k m_{rj}^t q_{rj}^t \boldsymbol{c}_r^t + \boldsymbol{c}_j^t$

Interpretation Results

- The trace only accounts for 66%-69% over iterations as shown in (b). This demonstrates the importance of collaboration, a unique advantage of population-based algorithms.



(a) Paths of first 80 samples of our meta-optimizer, PSO and GD for the 2D Rastrigin function.

(b) The percentage of the trace of $\gamma \boldsymbol{Q}^t \odot \boldsymbol{M}^t + \boldsymbol{I}$ (reflecting self-impact on updates) over iteration t.

- In the first 6 iterations, the population-based features (3 & 4) contribute to the updates the most. Point-based features (1 & 2) start to play an important role later:

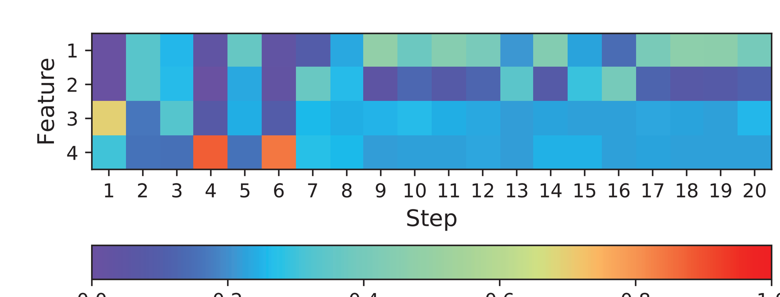


Figure: Feature distribution over the first 20 iterations for our meta-optimizer.

Ablation Study

Dimension	$\mathbf{B}_0$	$\mathbf{B}_1$	$\mathbf{B}_2$	$\mathbf{B}_3$	**Proposed**
10	55.4±13.5	48.4±10.5	40.1±9.4	20.4±6.6	12.3±5.4
20	140.4±10.2	137.4±12.7	108.4±13.4	48.5±7.1	43.0 ±9.2

Table: $\mathbf{B}_0$: the DM_LSTM baseline. $\mathbf{B}_1$: running DM_LSTM for k times and choosing the best solution . $\mathbf{B}_2$: using k independent particles, each with the two point-based features and the intra-particle attention module. $\mathbf{B}_3$: adding the two population-based features and the inter-particle attention module to $\mathbf{B}_2$. **Proposed**: adding an entropy term in meta loss to $\mathbf{B}_3$.

Acknowledgement

This work is in part supported by the National Institutes of Health (R35GM124952 to YS). Part of the computing time is provided by the Texas A&M High Performance Research Computing.

References

- [1] Marcin Andrychowicz et al. (2016). 'Learning to learn by gradient descent by gradient descent.' In Advances in Neural Information Processing Systems, pages 3981–3989.
- [2] Yue Cao, Tianlong Chen, Zhangyang Wang, Yang Shen. (2019). 'Learning to Optimize in Swarms'. NeurIPS 2019
- [3] Yue Cao and Yang Shen. (2019). 'Bayesian active learning for optimization and uncertainty quantification in protein docking.' arXiv preprint arXiv:1902.00067