Blackbox Optimization using LSTMs

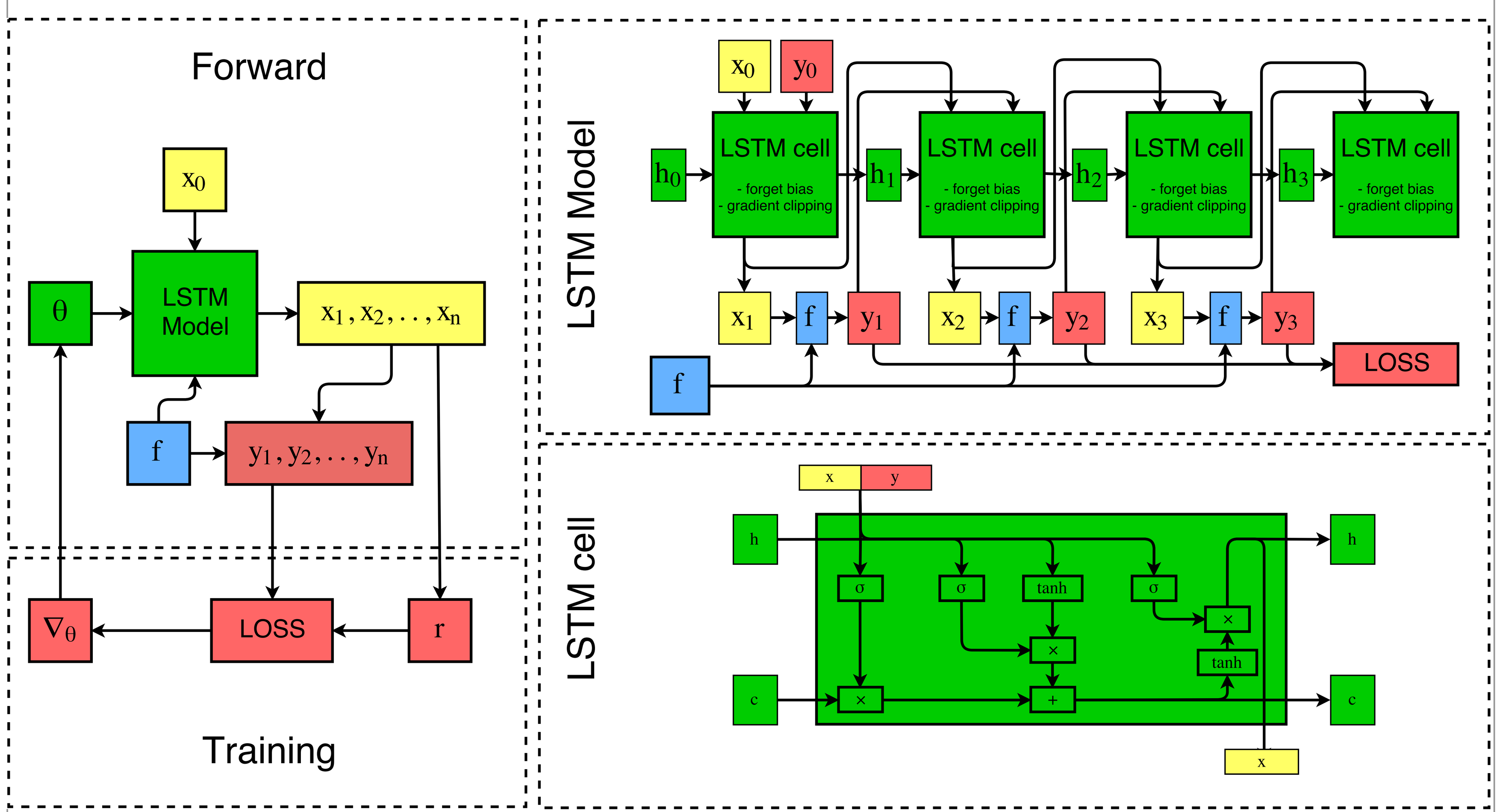
Felix Sattler, Pattarawat Chormai, Raphael Holca-Lamarre

Summary

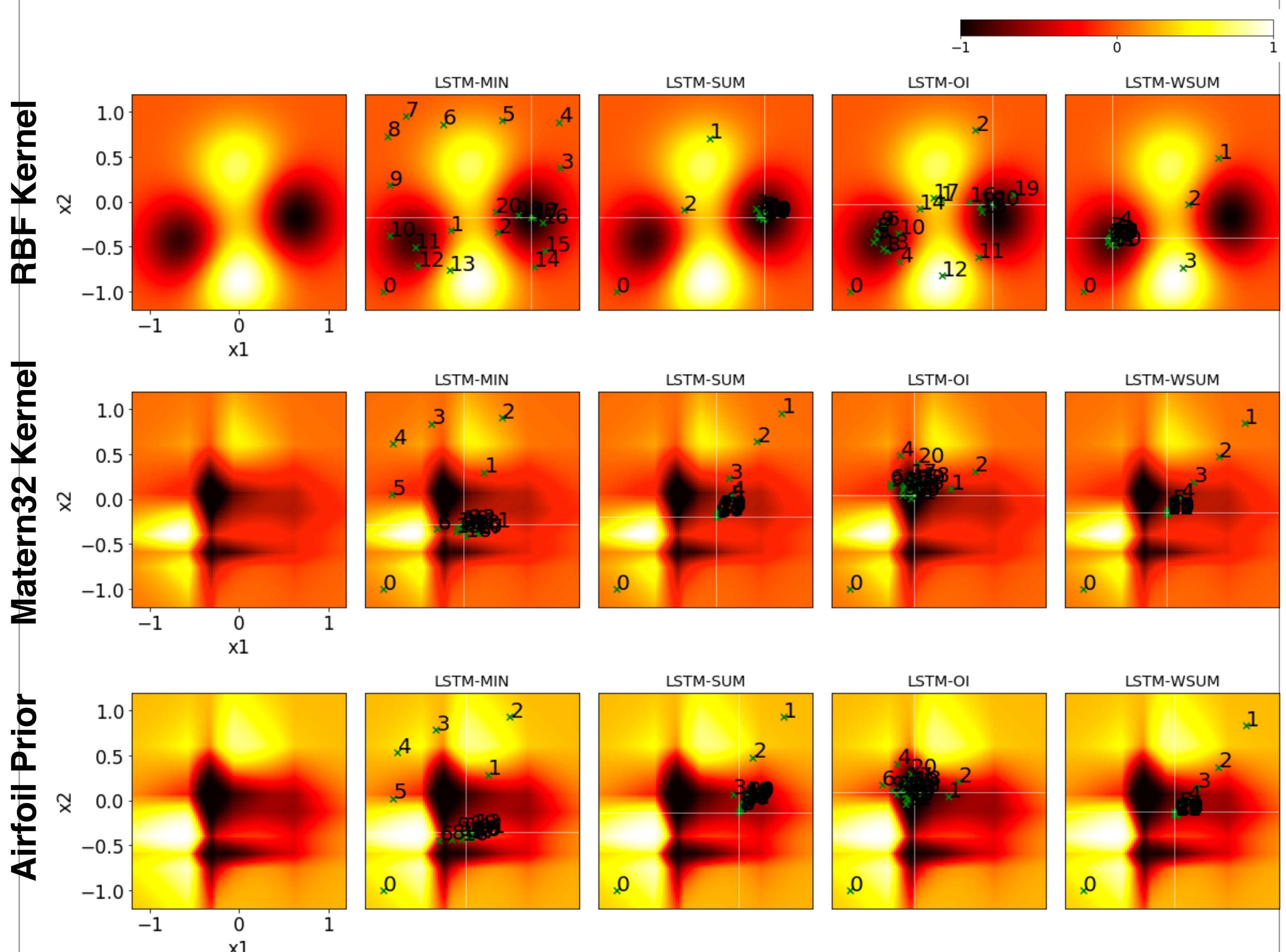
We extend a new, learning-based, approach to global Black Box optimization, introduced by Chen et al. [1]. The idea is to train a LSTM[2] to output a sequence of sample points that seek to minimize a given objective function.

Our experiments show that the learned model is able to generalize to a variety of synthetic Black-Box functions, as well as to the real world problem of airfoil optimization. Moreover, our model performs comparable to state-of-the-art Black-Box optimization algorithms w.r.t. the minimum function value found and outperforms them w.r.t. computation time by a wide margin.

Model



Training data and loss functions



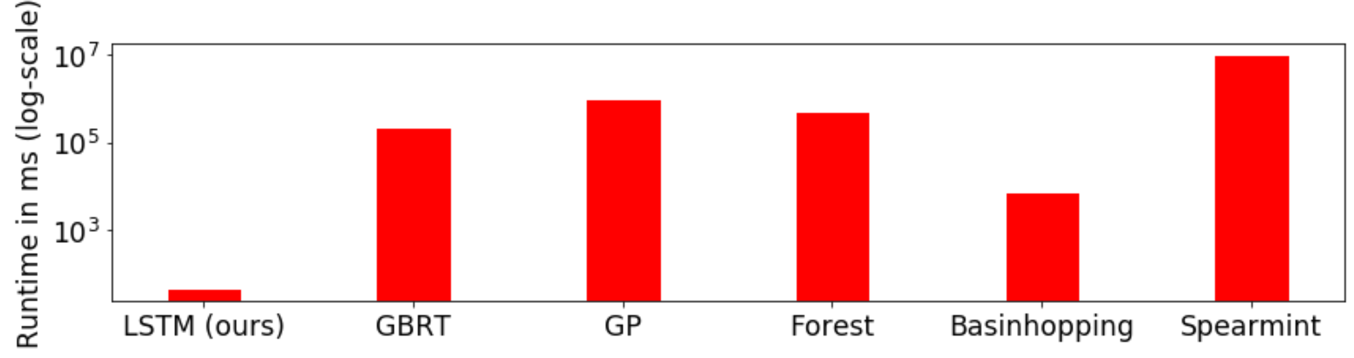
$L_{\text{MIN}}(\theta) = \min_{k=1,\ldots,n} f(x_k)$

$L_{\text{SUM}}(\theta) = \sum_{k=1}^{n} f(x_k)$

$L_{\text{OI}}(\theta) = \sum_{k=1}^{n} \max(0, f(x_k) - \min_{i<k} f(x_i))$

$L_{\text{WSUM}}(\theta) = \sum_{k=1}^{n} \frac{1}{2^{n-k}} f(x_k)$

Computation time

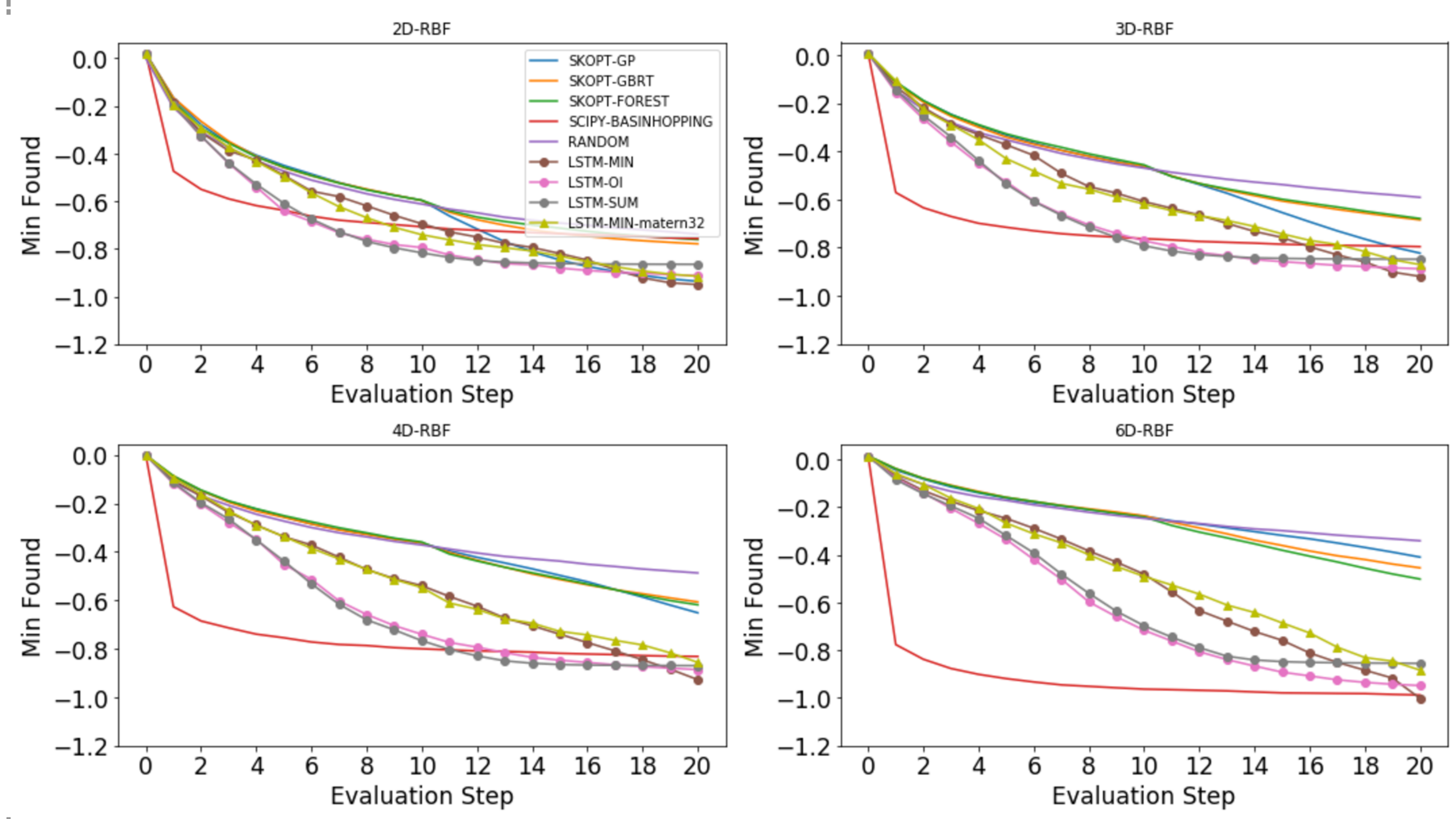


- Time taken in seconds by our approach and other optimization algorithms to propose 20 sampling points for 100 different 2D objective functions. Our method is up to significantly faster than the others. Noting that, training time for our method is approximately 30 minutes.

References

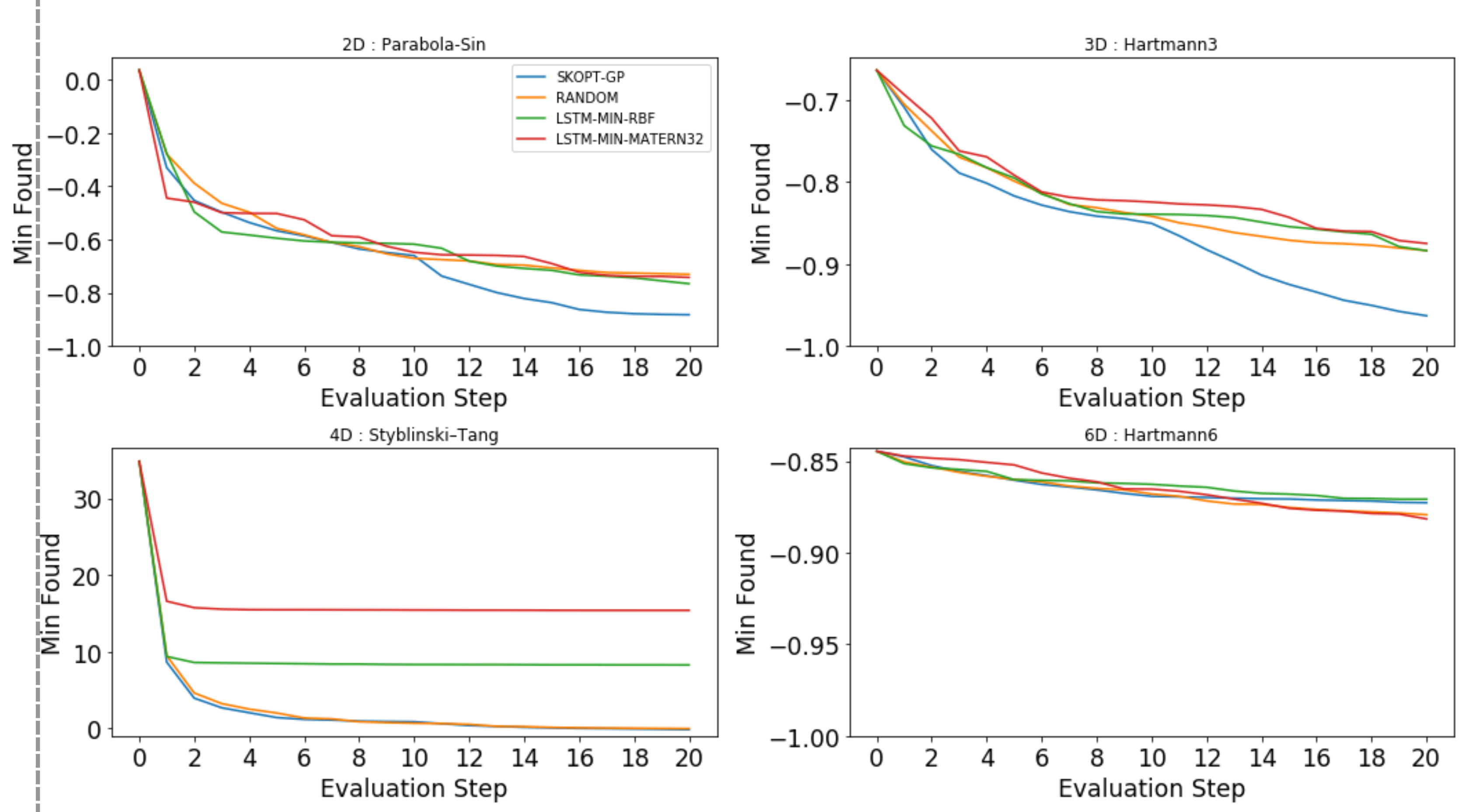
[1] Chen, Yutian, et al. "Learning to Learn without Gradient Descent by Gradient Descent."
[2] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.

Results on Test Functions



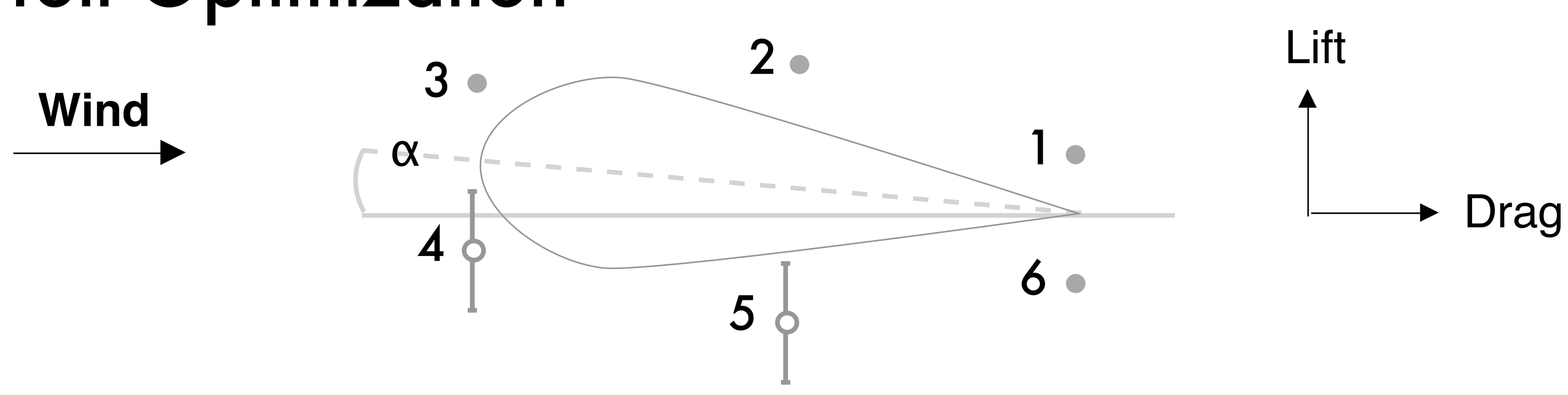
- We compare the performance of our method to other commonly used black-box optimizers on functions samples from a gaussian process prior (average taken over 2000 test functions).

Results on Benchmark Functions

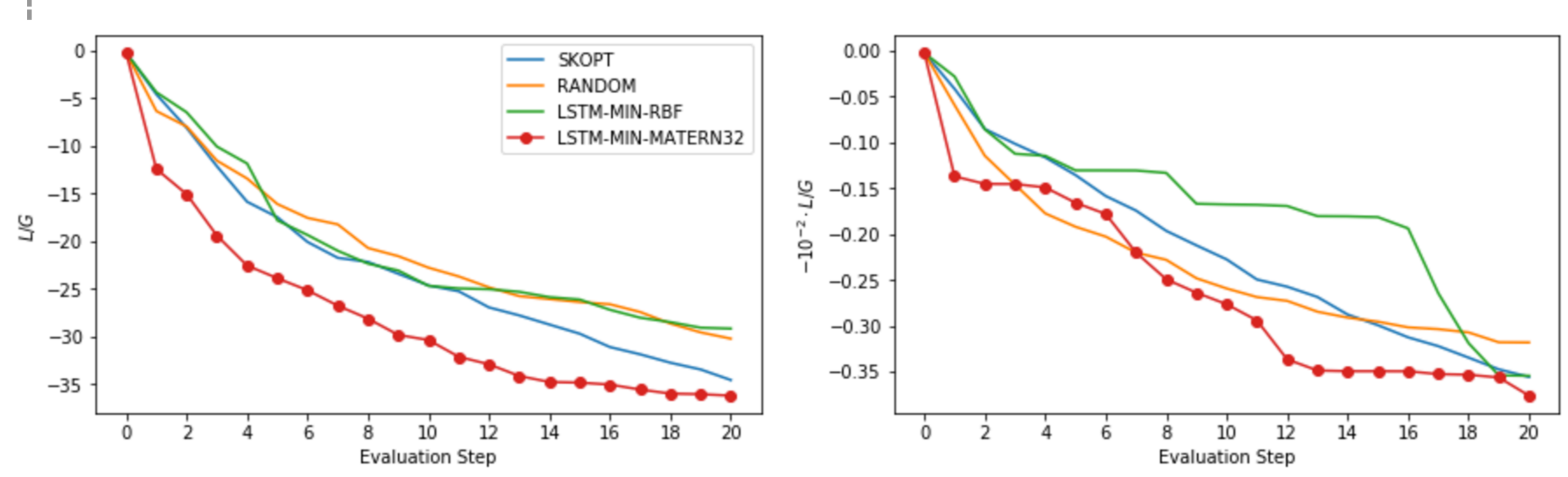


- some explanation

Airfoil Optimization



Goal : $\max \frac{L}{G} \approx \min -\frac{L}{G}$



- Results on an Airfoil shape optimization problem. Average over 100 different combinations of angle of attack and pairs of control points.

Conclusions

- Our model learns by itself to trade-off exploration and exploitation during sampling; we can guide its behavior by the choice of the loss function we use during training
- Our model performs well also in high dimensions, and inference time scales only linearly with the dimension
- Our approach requires the objective function to be somewhat normalized; this however is no restriction if rough upper and lower bounds

Acknowledgement NVIDIA