Faster & More Reliable Tuning of Neural Networks:
Bayesian Optimization with Importance Sampling

Setareh Ariafar, Zelda Mariet, Ehsan Elhamifar, Dana Brooks, Jennifer Dy, Jasper Snoek

Motivation

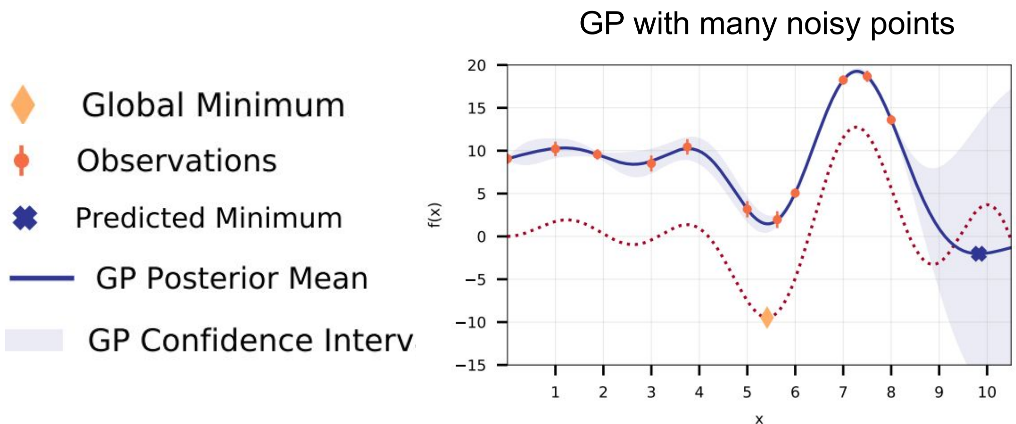
- Training neural nets → **expensive**
- Bayesian Optimization (BO) → **limited hyperparameters**
- **Low-fidelity** observations

<u>Pros</u>

- **Increased #** of explored hyperparameters via:
  - Cheap partially trained models
  - Extrapolate to fully trained models

<u>Cons</u>

- Adds to the **randomness/noise** of BO
- Challenging extrapolation



Proposed Solution

- **Decrease randomness** by using Information of each training example
- BO + **Importance Sampling** (IS)

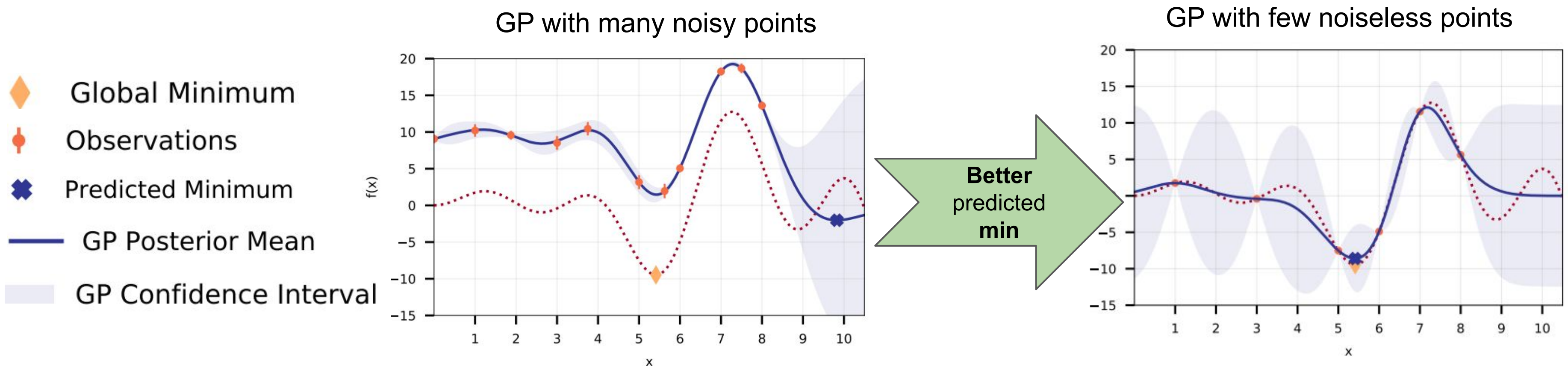
Pros

- **High-fidelity** observation
- More accurate models
- **Less # of** observations required

Cons

- Large **overhead cost** challenging

Solve via **multi-task BO** over **importance sampling design**
Learn when high-fidelity is worth the cost



Google AI Northeastern University

Proposed Solution

IS-SGD [1]

- Importance dist → **Initially similar to uniform sampling & expensive**

Start from uniform sampling
Track variance reduction
Switch to IS if variance reduction large

Select a random super-batch of size **B**
Select mini-batches with IS from super-batch

[1] Katharopoulos & Fleuret., ICML 2018

To learn the **trade-off** parameter **B** → Maximize $\alpha_n(x, B)$

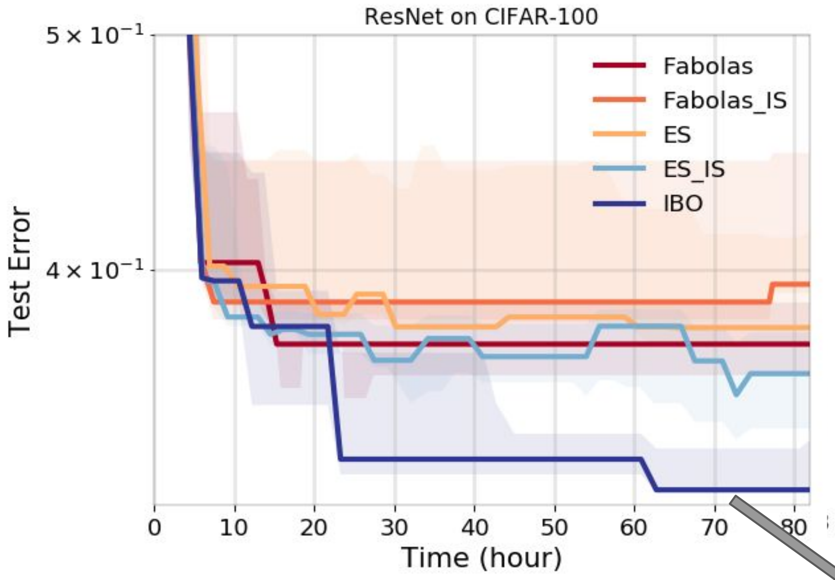
$$\alpha_n(x,B) = \frac{1}{\mu(c_n(x \mid B))}\Big[H(\mathbb{P}[x^* \mid B = |\mathbf{D}|, \mathcal{D}_n]) - \mathbb{E}_y\big[H(\mathbb{P}[x^* \mid B = |\mathbf{D}|, \mathcal{D}_n \cup \{x, B, y\}])\big]\Big],$$

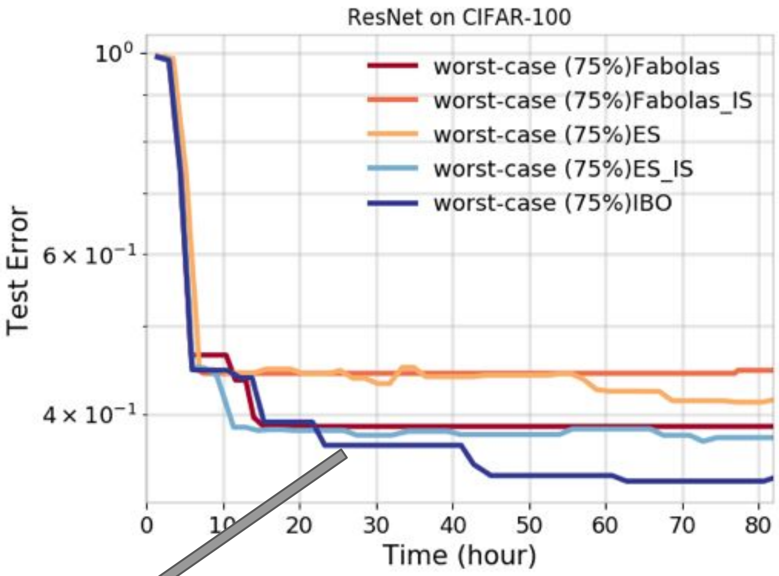
Expected training cost for x, **B**

Expected entropy reduction from training on hyperparameter x via IS-SGD routine with super-batch size **B**

Results- ResNet on CIFAR100

- Improved worst-case performance





Ours

Average Performance

Worst-case Performance