Cost-Efficient Online Hyperparameter Optimization

Jingkang Wang*[1,2], Mengye Ren*[1,2], Ilija Bogunovic[3], Yuwen Xiong[1,2], Raquel Urtasun[1,2]

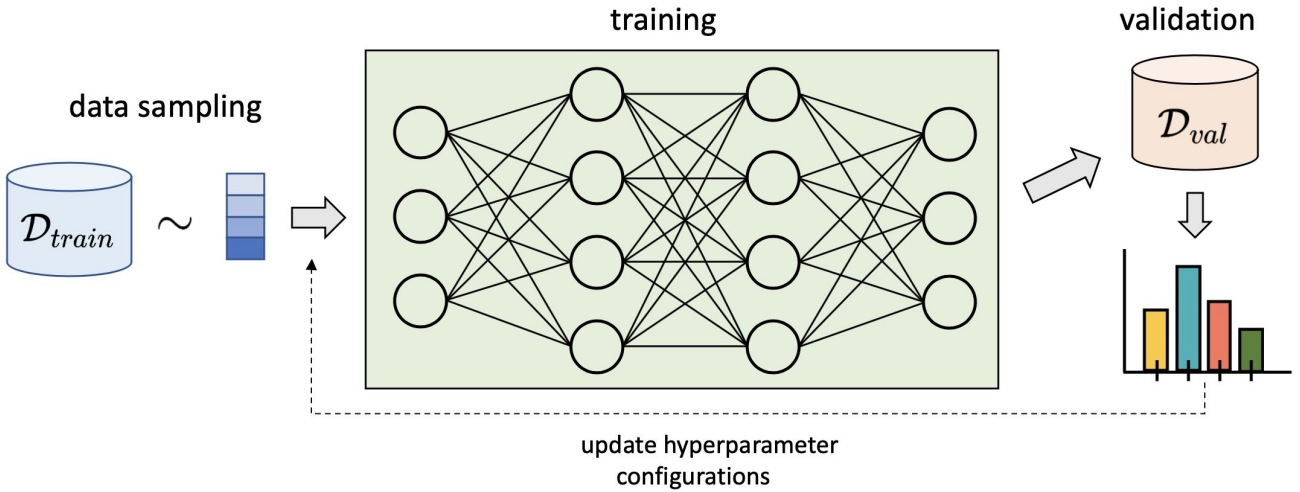
[1] Uber ATG

[2] UNIVERSITY OF TORONTO

[3] ETH zürich

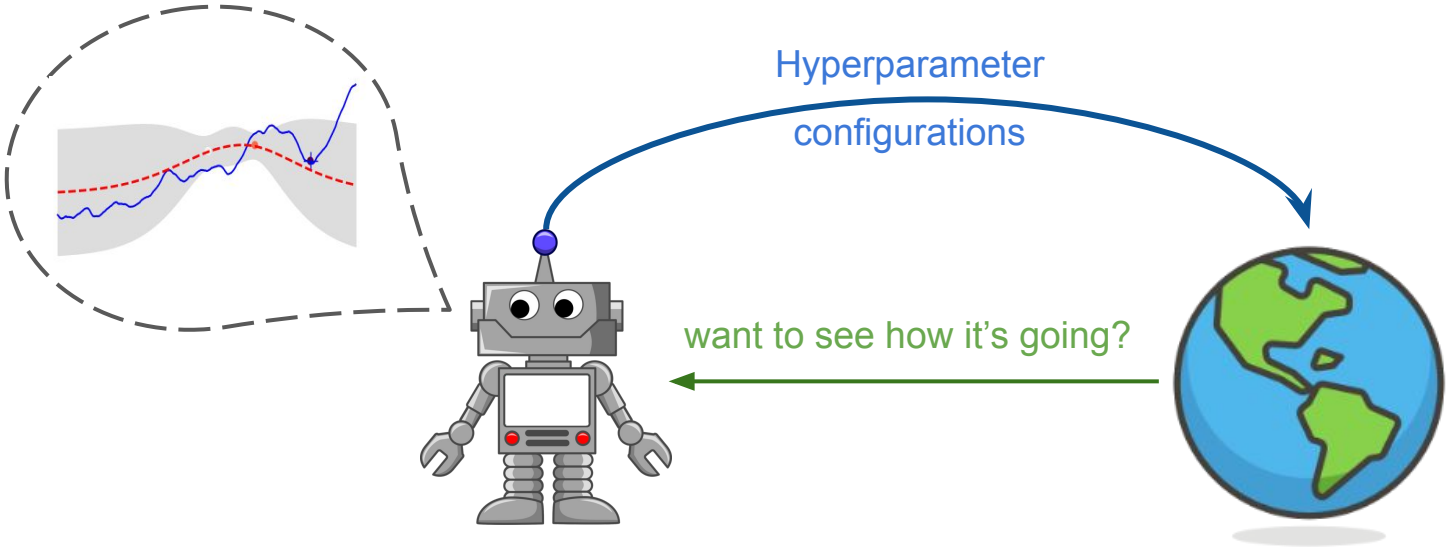
Motivation

- Online Hyperparameter Optimization (HPO) shows that one can actually tune certain hyperparameters (e.g., data augmentation, weight decay) by **constantly** evaluating on the validation set.
- Evaluating on the held-out set and taking gradient w.r.t. hyperparameters are **expensive**
- We make online HPO **efficient in the evaluation cost**



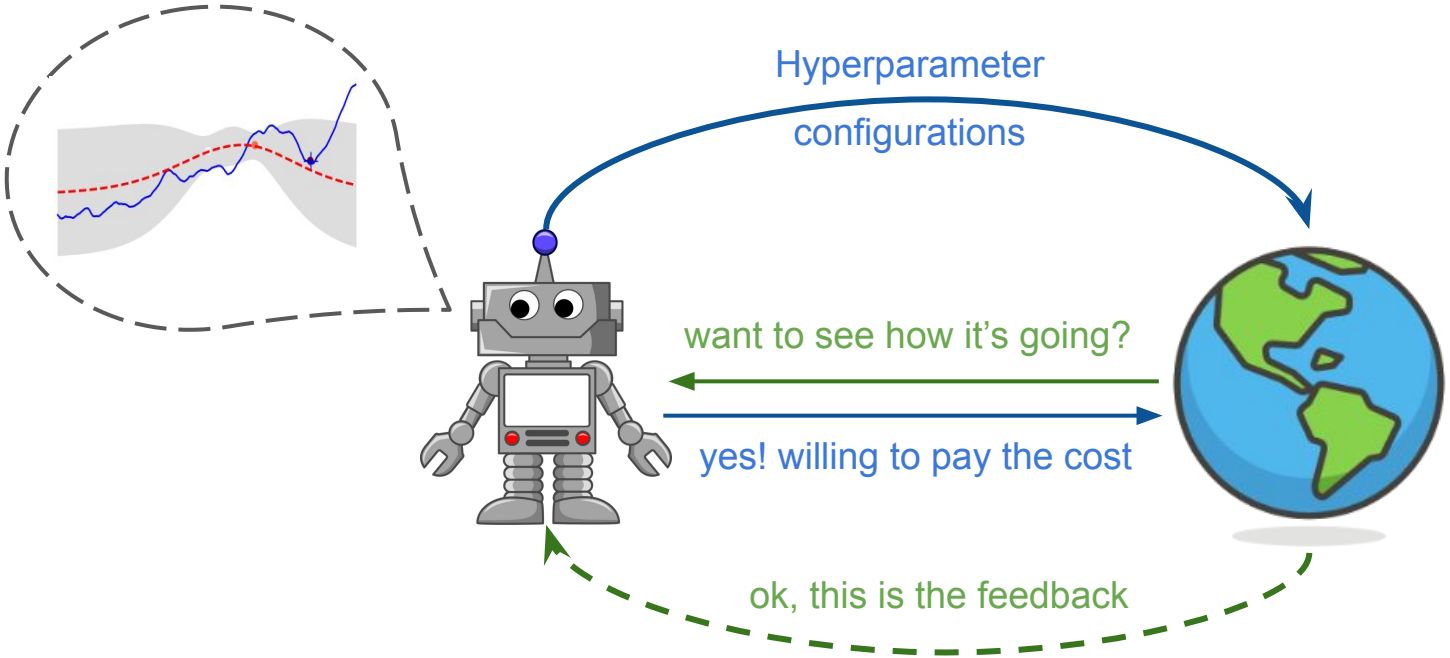
Problem Statement

- Model online HPO as a **time varying** (TV) Bayesian optimization (BO) problem with **costly feedback**
- requires the agent to **pay a cost** whenever it receives the feedback from the unknown function



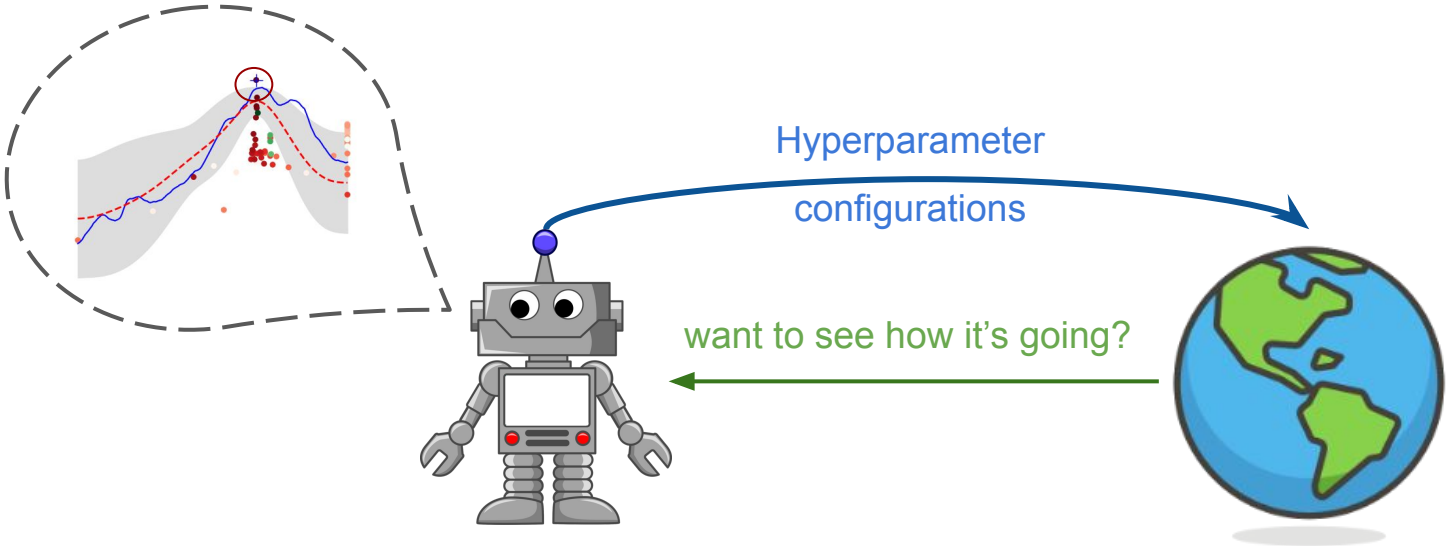
Problem Statement

- Model online HPO as a **time varying** (TV) Bayesian optimization (BO) problem with **costly feedback**
- requires the agent to **pay a cost** whenever it receives the feedback from the unknown function



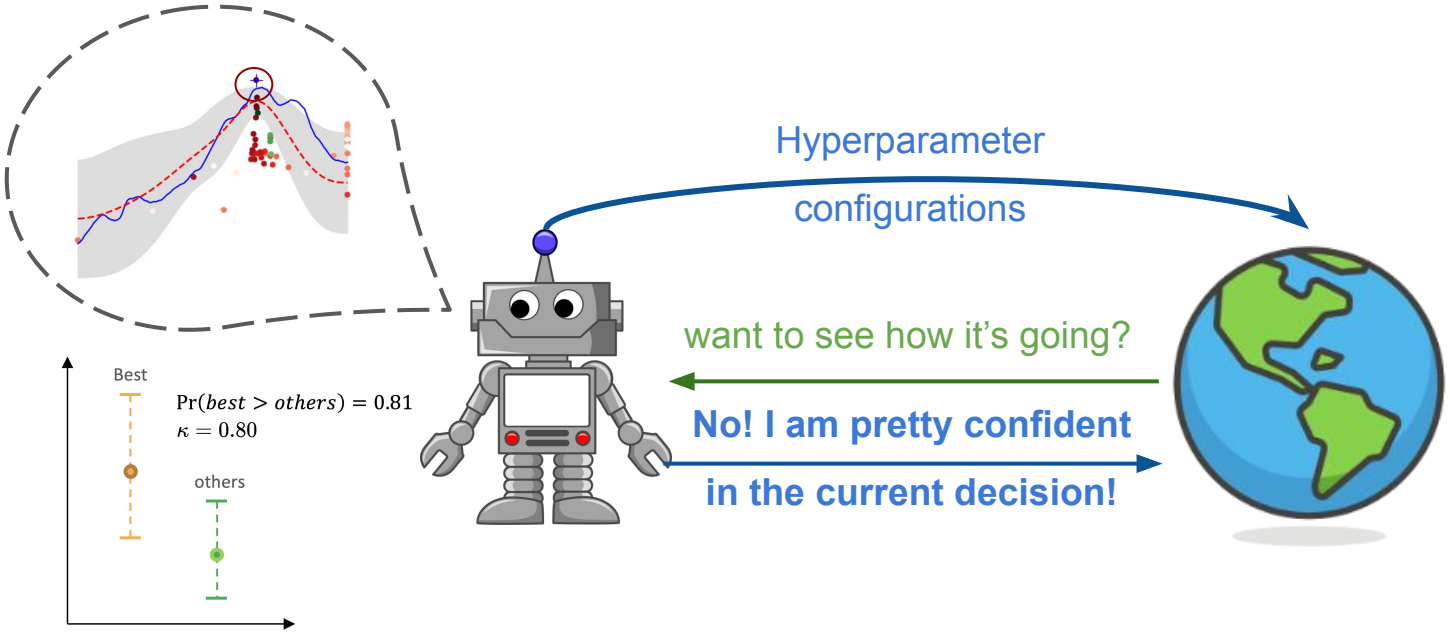
Cost-Efficient Query Strategy

- **General Idea:** skip those uninformative queries to save the query cost
- **Cost-Efficient Query Rule:**



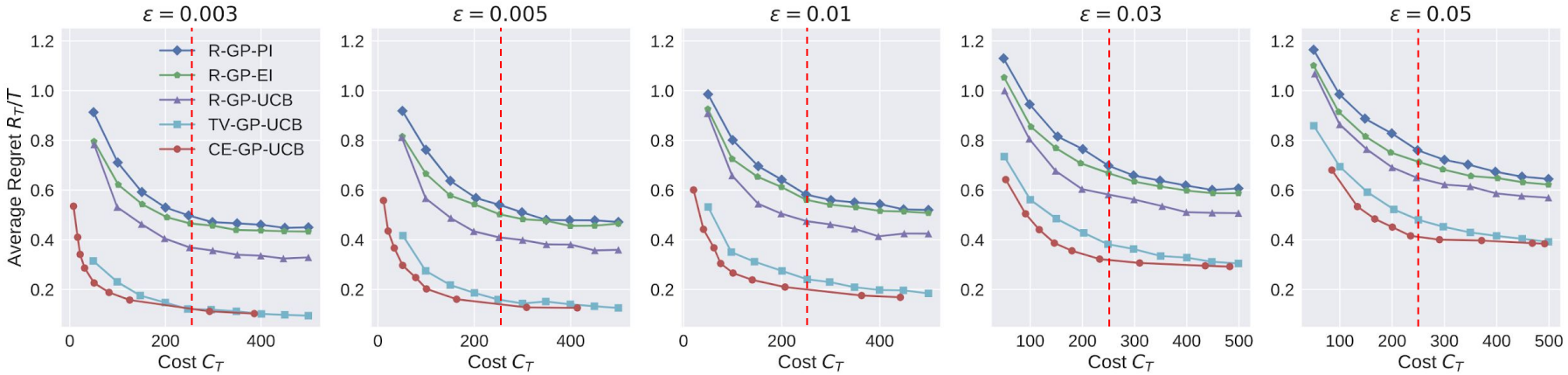
Cost-Efficient Query Strategy

- **General Idea:** skip those uninformative queries to save the query cost
- **Cost-Efficient Query Rule:**



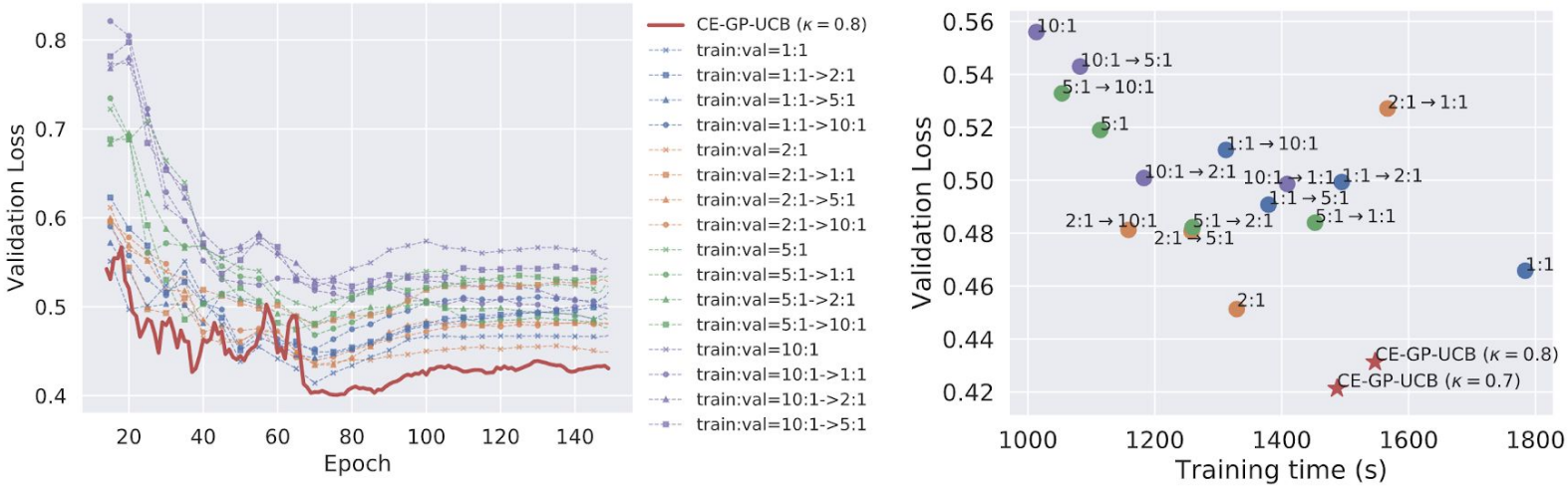
Results (synthetic data)

- **TV Bayesian Optimization**
- one-dimensional input domain $\mathcal{D} = [0, 1]$, quantized to 1000 points uniformly
- T = 500, Matern5/2 kernel, sampling noise
- forgetting rate eps: 0.003 -> 0.05



Results (self-tuning networks)

- **Tuning schedule** plays a significant role when applying STN (MacKay et al., 2019) to larger network
- VGG16 on CIFAR-10 -> Train set: 40,000 images, Val set: 10,000 images
- Modeling as a two-armed bandit problem. (1) training only; (2) tuning + training

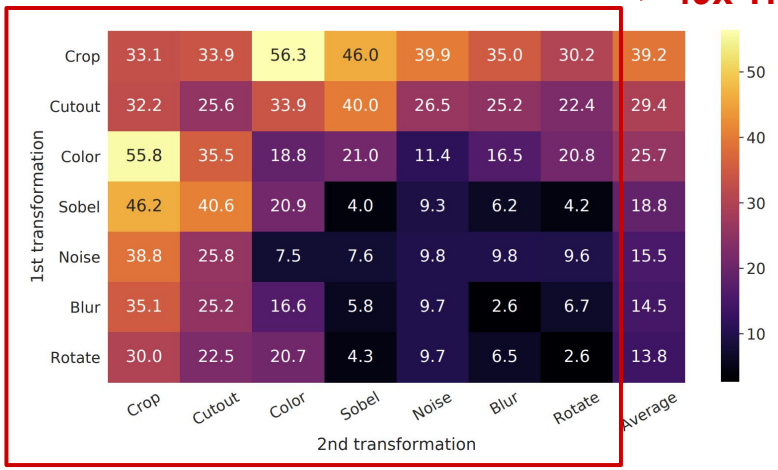


Matthew MacKay, Paul Vicol, Jon Lorraine, David Duvenaud, Roger Grosse. Self-Tuning Networks: Bilevel Optimization of Hyperparameters using Structured Best-Response Functions, ICLR 2019.

Results (unsupervised learning)

- **Eight data augmentations**: crop, cutout, flip horizontally and vertically, rotation, color distortion, gray scale, Gaussian blur.
- Baseline (fix the probability of randomly apply data augmentations as 0.5)
- Computation: grid search: **49x**, TV-GP-UCB (full): **2x**, CE-GP-UCB: **1.6x**

> 49x Time!



	Top1 (R10)	Top1 (R100)	Time
Baseline	70.91	73.20	1.00 ×
TV-GP-UCB (full)	75.14	77.95	1.97 ×
TV-GP-UCB Ber(0.6)	72.15	75.31	1.64 ×
CE-GP-UCB ($\kappa = 0.9$)	74.80	77.56	1.88 ×
CE-GP-UCB ($\kappa = 0.8$)	74.77	77.62	1.71 ×
CE-GP-UCB ($\kappa = 0.7$)	74.58	77.27	1.61 ×
Human expert [12]	75.00	77.99	-

R10/100: readout 10/100 epochs

[12]: Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations

Thank you!