Bayesian Optimization with Categorical and Continuous Variables

Dr Vu Nguyen
Machine Learning Scientist
amazon

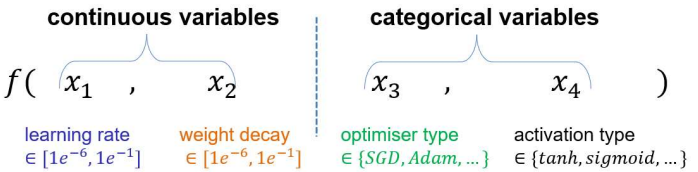
@nguyentienvu

vu-nguyen.org

Outline

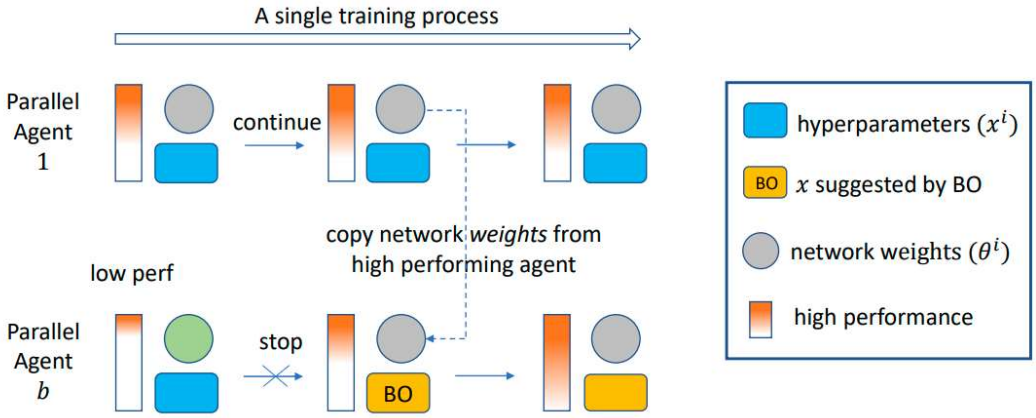
- Bayesian optimisation and Illustration

- Mixed categorical-continuous optimisation



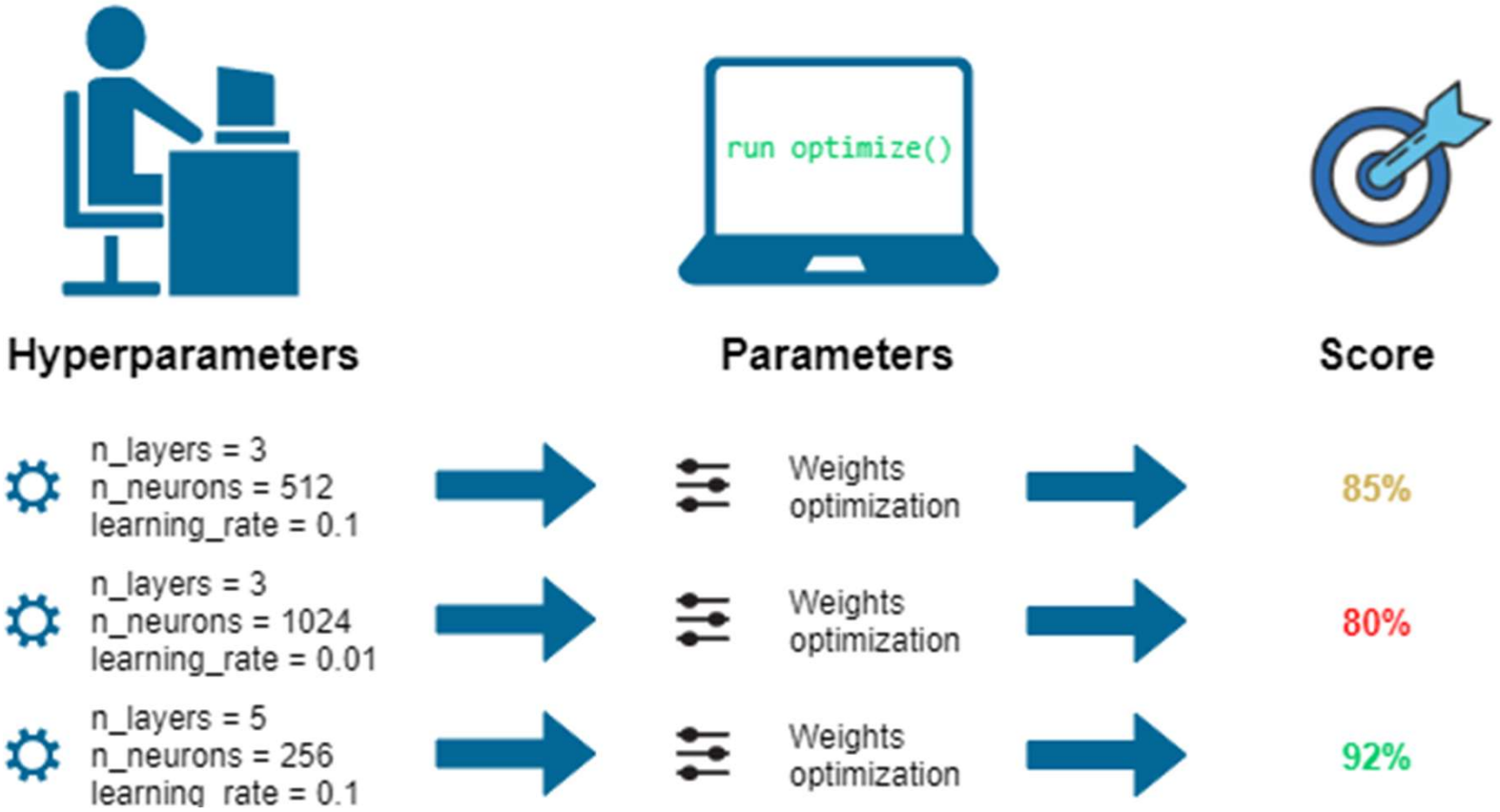
- Mixed optimization in high dimensional space

- Mixed optimization in population-based AutoRL



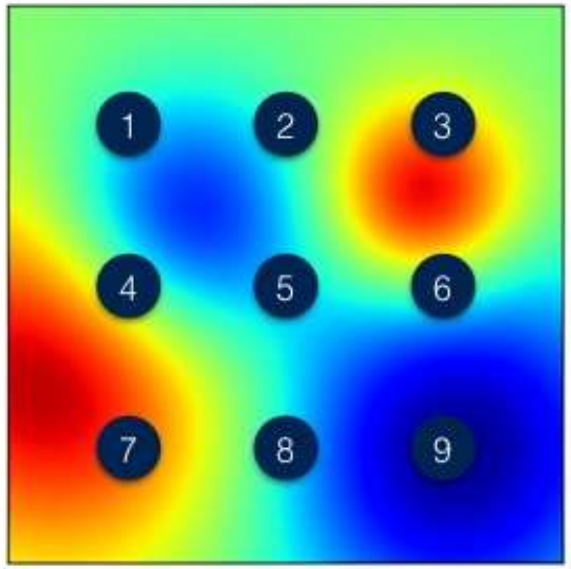
Hyperparameters Optimization

- ML algorithm's performances depend on hyperparameters.
- Finding the best hyperparameters for the highest performance

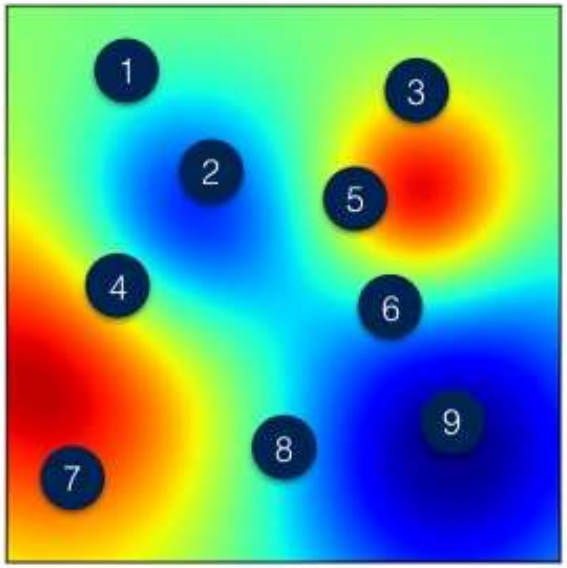


Traditional Hyperparameters Tuning

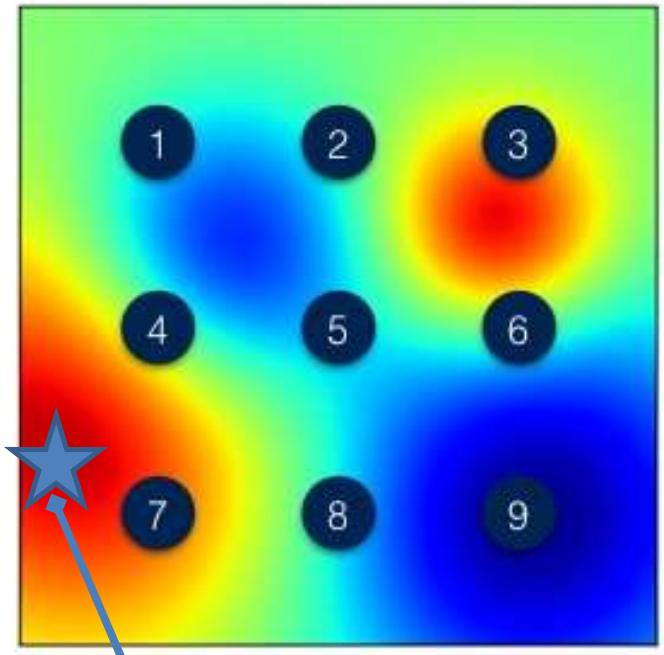
- Grid Search:
  - Create a list of values for each parameter.
  - Consider all possible combinations of these values.
  - Exhaustively evaluate the model and choose the best parameter.



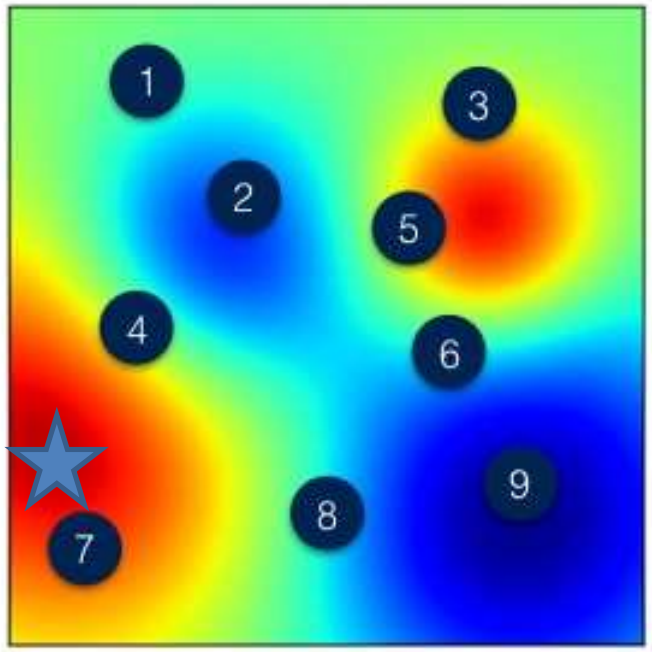
- Random Search:
  - Randomly select a parameter to evaluate.
  - Select the best parameter.



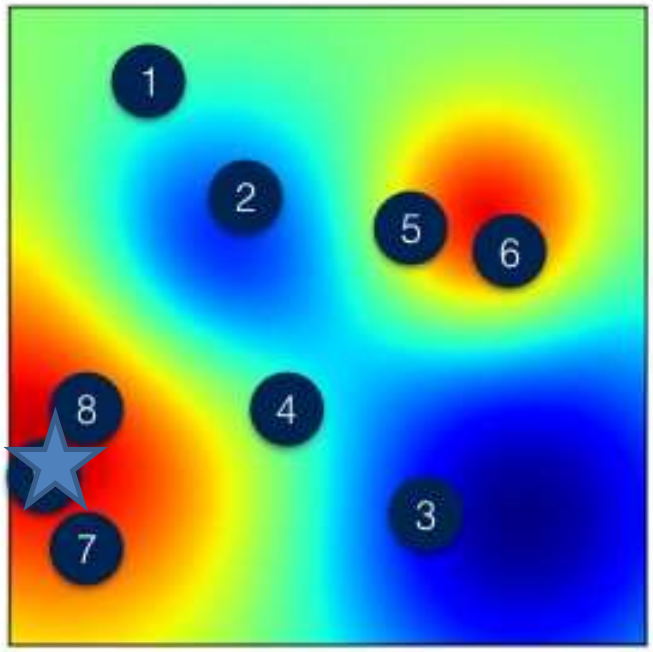
Grid vs Random vs Bayesian Optimization



Grid Search



Random Search

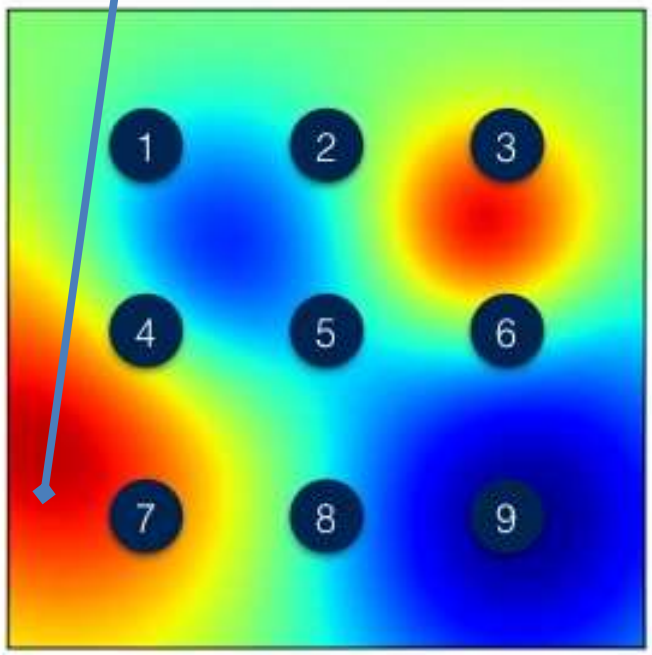


Bayesian Optimization

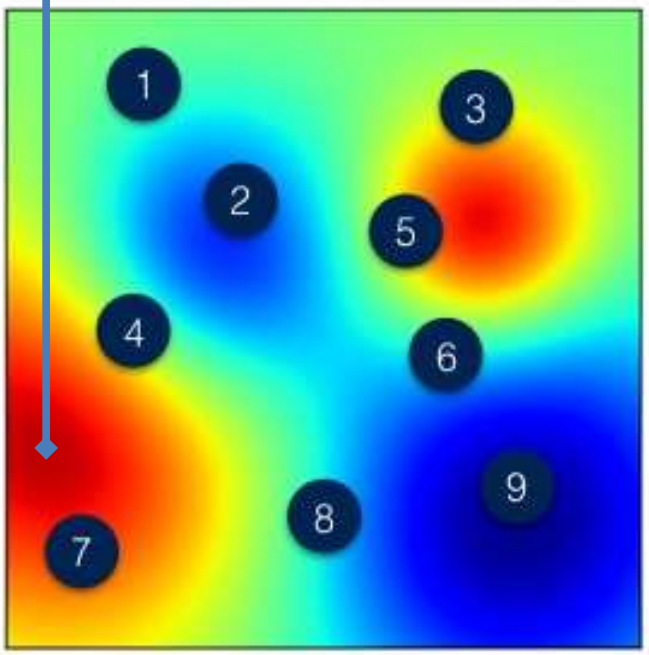
best hyperparameter

Grid vs Random vs Bayesian Optimization

Missed
optimum location

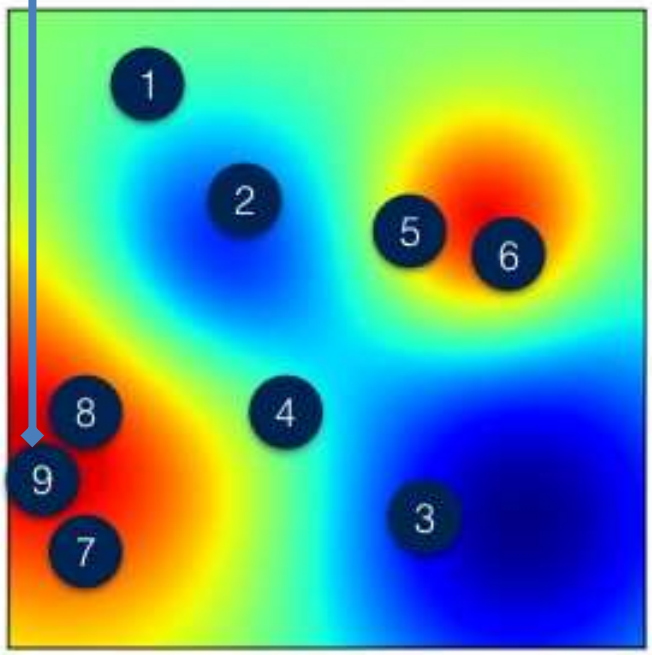


Grid Search



Missed
optimum location

Random Search

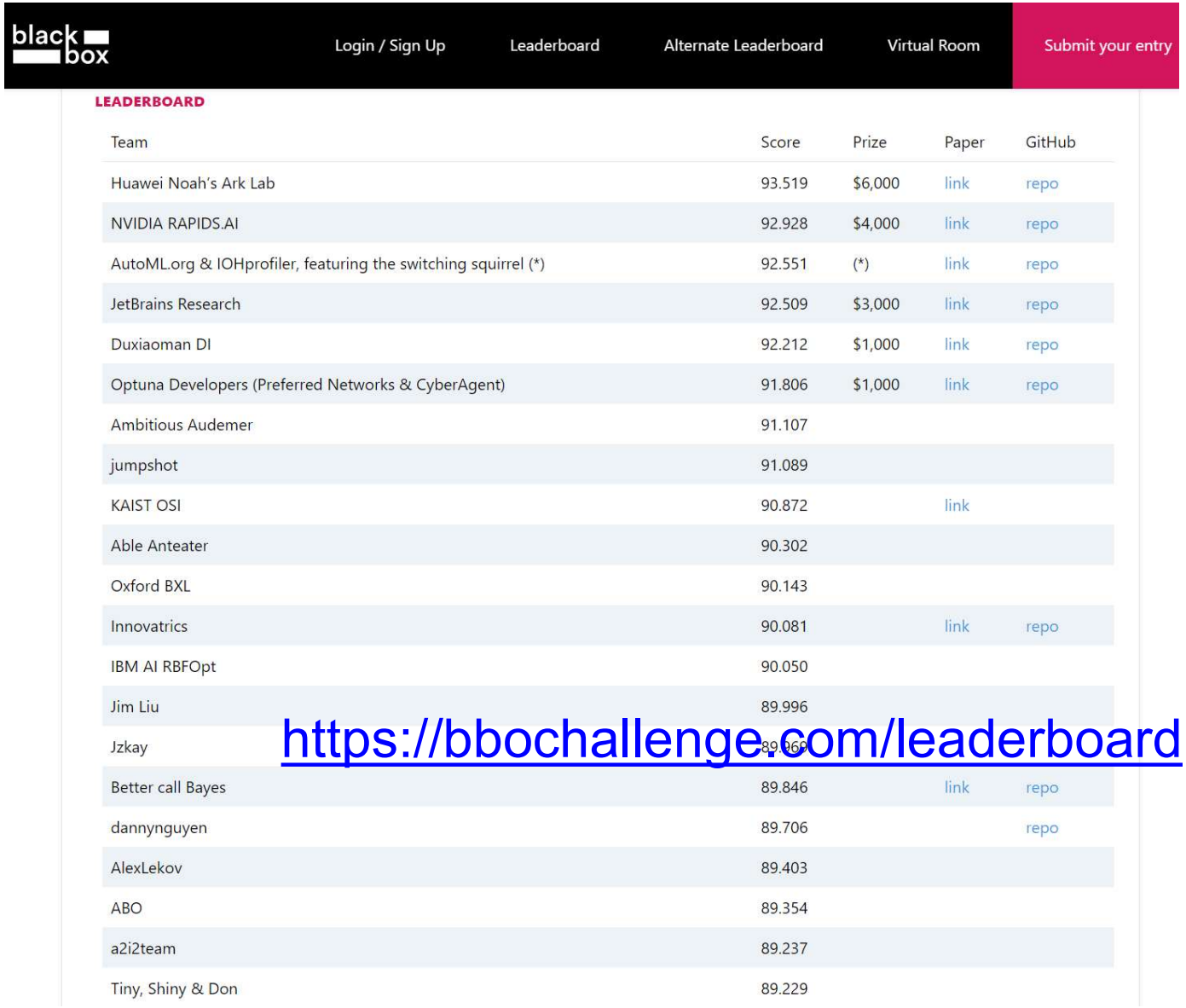


Found
optimum location

Bayesian Optimization

Blackbox optimisation competition at NeurIPS20

- The top 20 teams uses Bayes Opt.

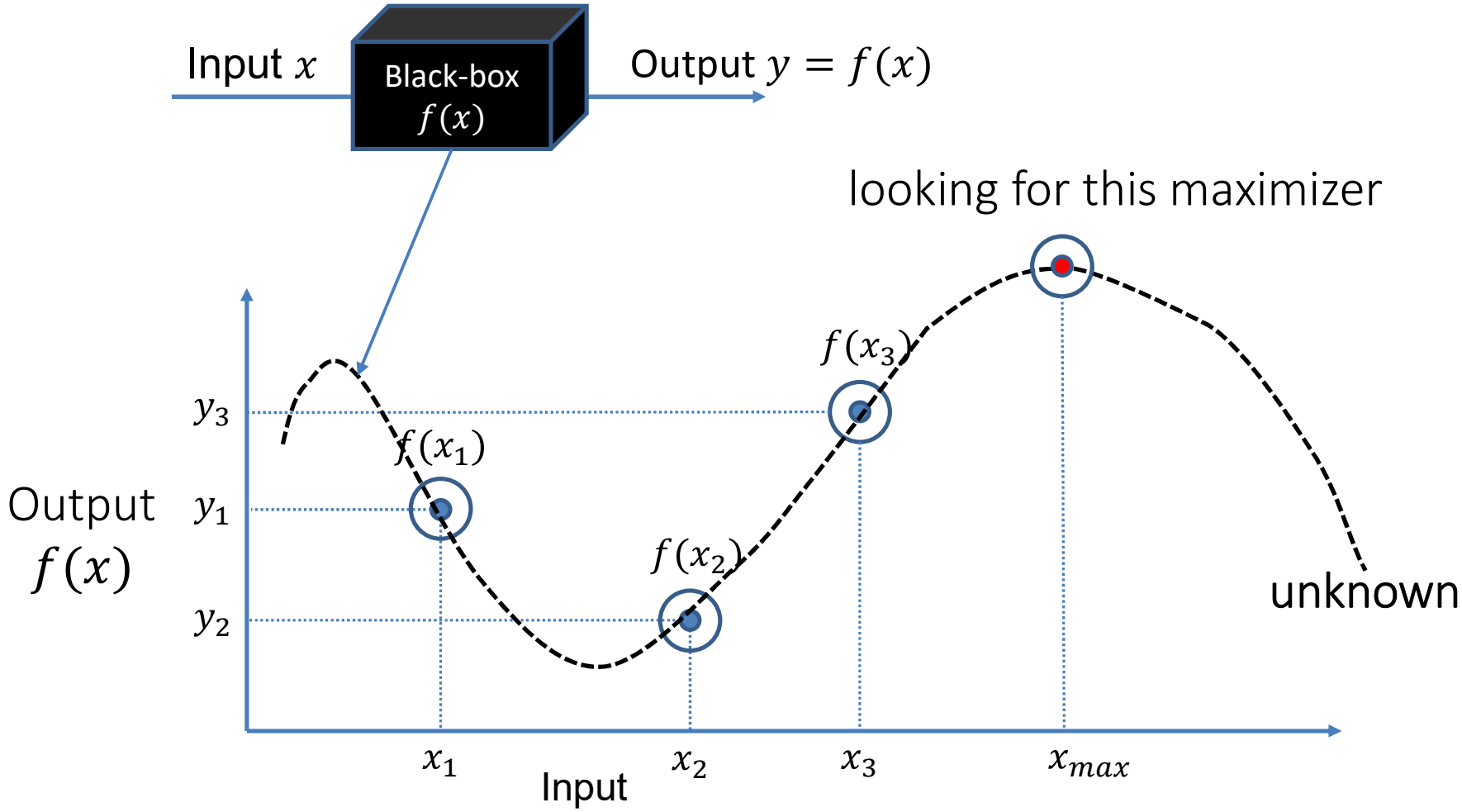


Team	Score	Prize	Paper	GitHub
Huawei Noah's Ark Lab	93.519	$6,000	link	repo
NVIDIA RAPIDS.AI	92.928	$4,000	link	repo
AutoML.org & IOHprofiler, featuring the switching squirrel (*)	92.551	(*)	link	repo
JetBrains Research	92.509	$3,000	link	repo
Duxiaoman DI	92.212	$1,000	link	repo
Optuna Developers (Preferred Networks & CyberAgent)	91.806	$1,000	link	repo
Ambitious Audemer	91.107			
jumpshot	91.089			
KAIST OSI	90.872		link	
Able Anteater	90.302			
Oxford BXL	90.143			
Innovatrics	90.081		link	repo
IBM AI RBFOpt	90.050			
Jim Liu	89.996			
Jzkay	89.969			
Better call Bayes	89.846		link	repo
dannynguyen	89.706			repo
AlexLekov	89.403			
ABO	89.354			
a2i2team	89.237			
Tiny, Shiny & Don	89.229			

https://bbochallenge.com/leaderboard

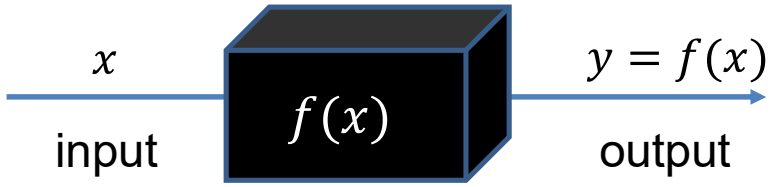
Black-box Optimization

- The relationship from x to y is through the black-box.



Properties of Black-box Function

$$f: X \in \mathcal{R}^d \rightarrow Y \in \mathcal{R}$$



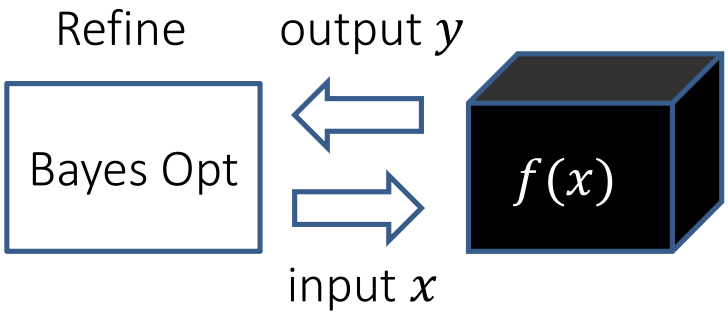
Function form is not known ~~$y = ax + b$~~

No derivative form ~~$\frac{\partial f}{\partial x} = \cdots$~~

Expensive to evaluate (in time and cost)

Nothing is known about the function, except a few evaluations $y = f(x)$

Bayesian Optimization Overview



- Make a series of evaluations $x_1, x_2, \ldots x_T$
- Find the optimum using few evaluations

Acquisition function $\alpha(x) = \boldsymbol{\mu(x)} + \kappa \times \sigma(x)$

exploit

explore

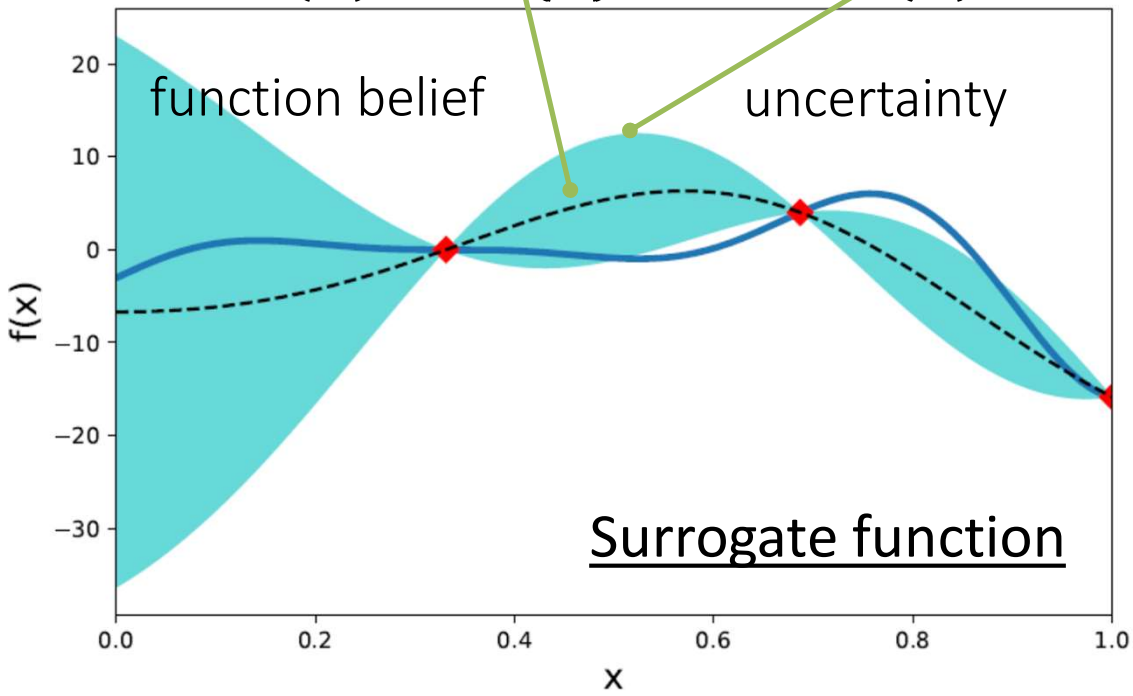


Illustration of Bayes Opt (3 points)

- Given 3 initial observations

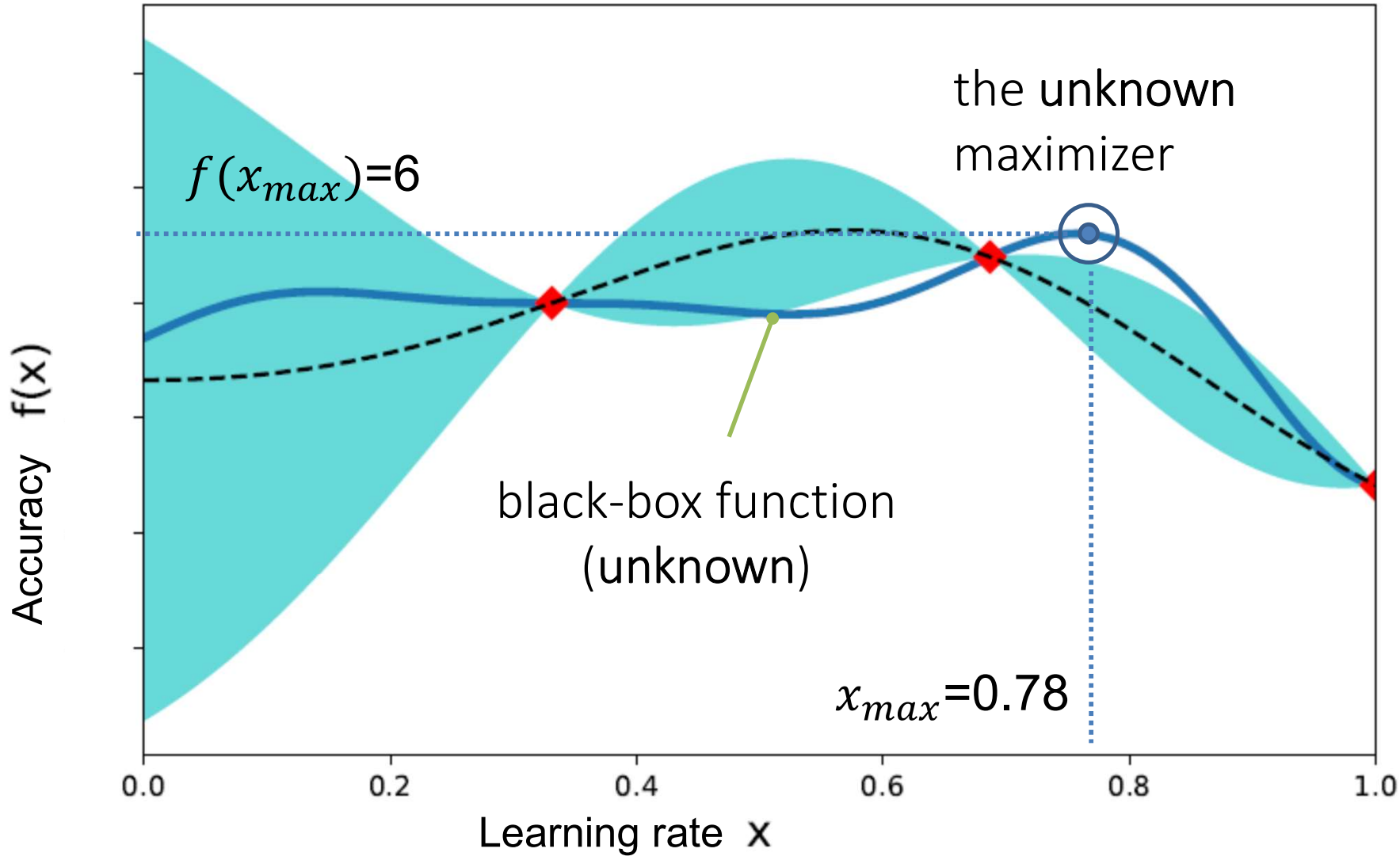


Illustration of Bayes Opt (3 points)

- Given 3 initial observations

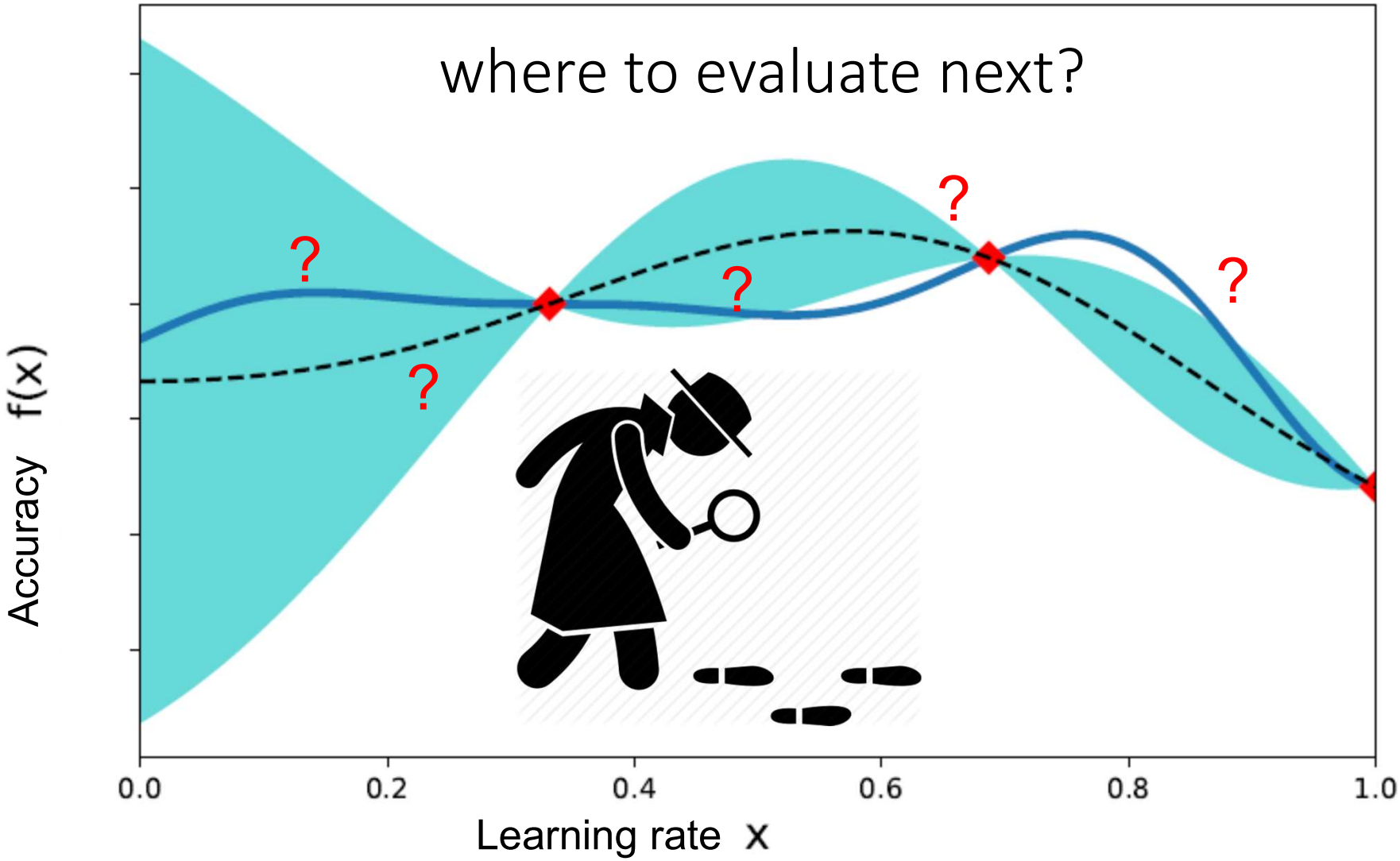


Illustration of Bayes Opt (4 points)

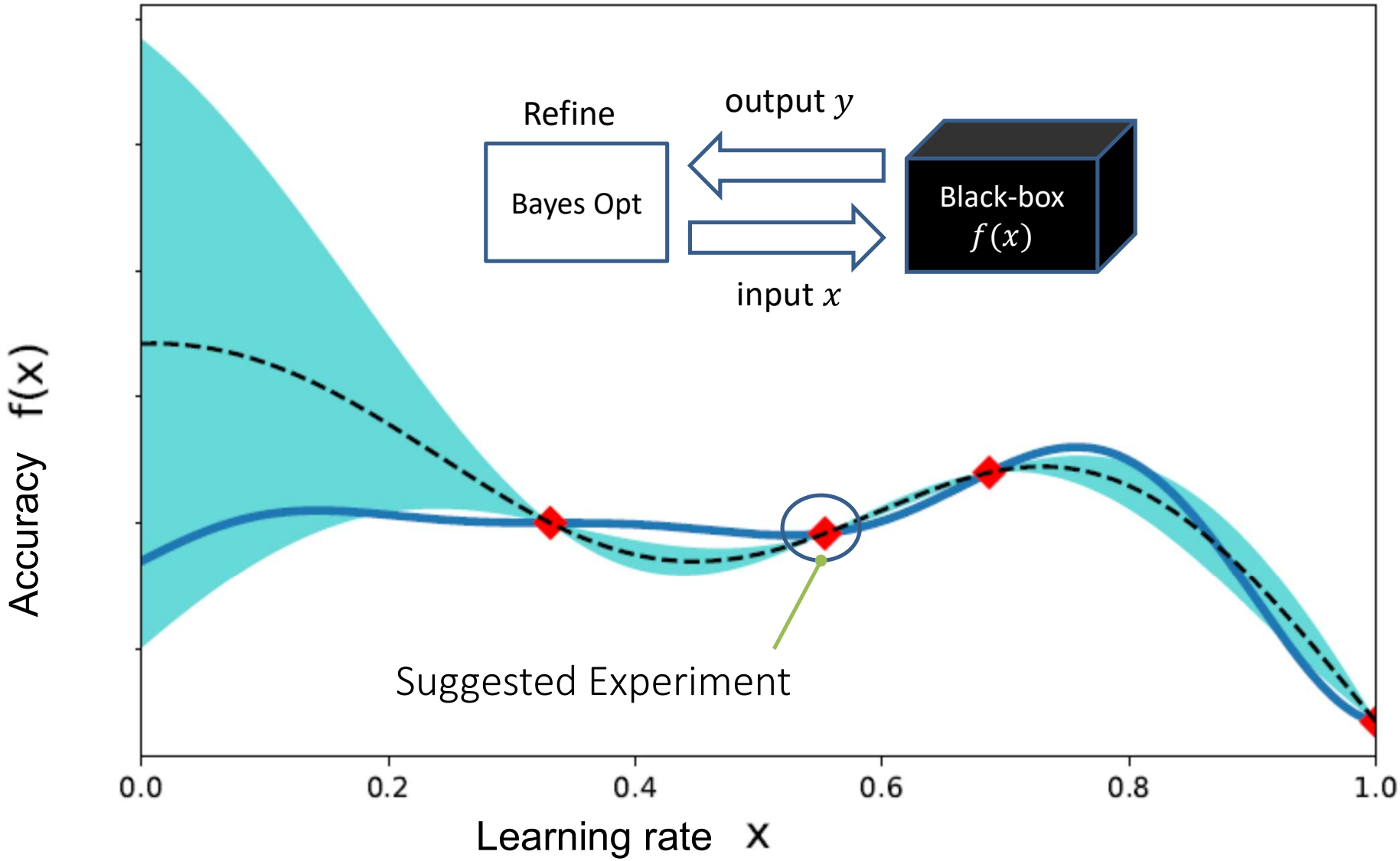


Illustration of Bayes Opt (5 points)

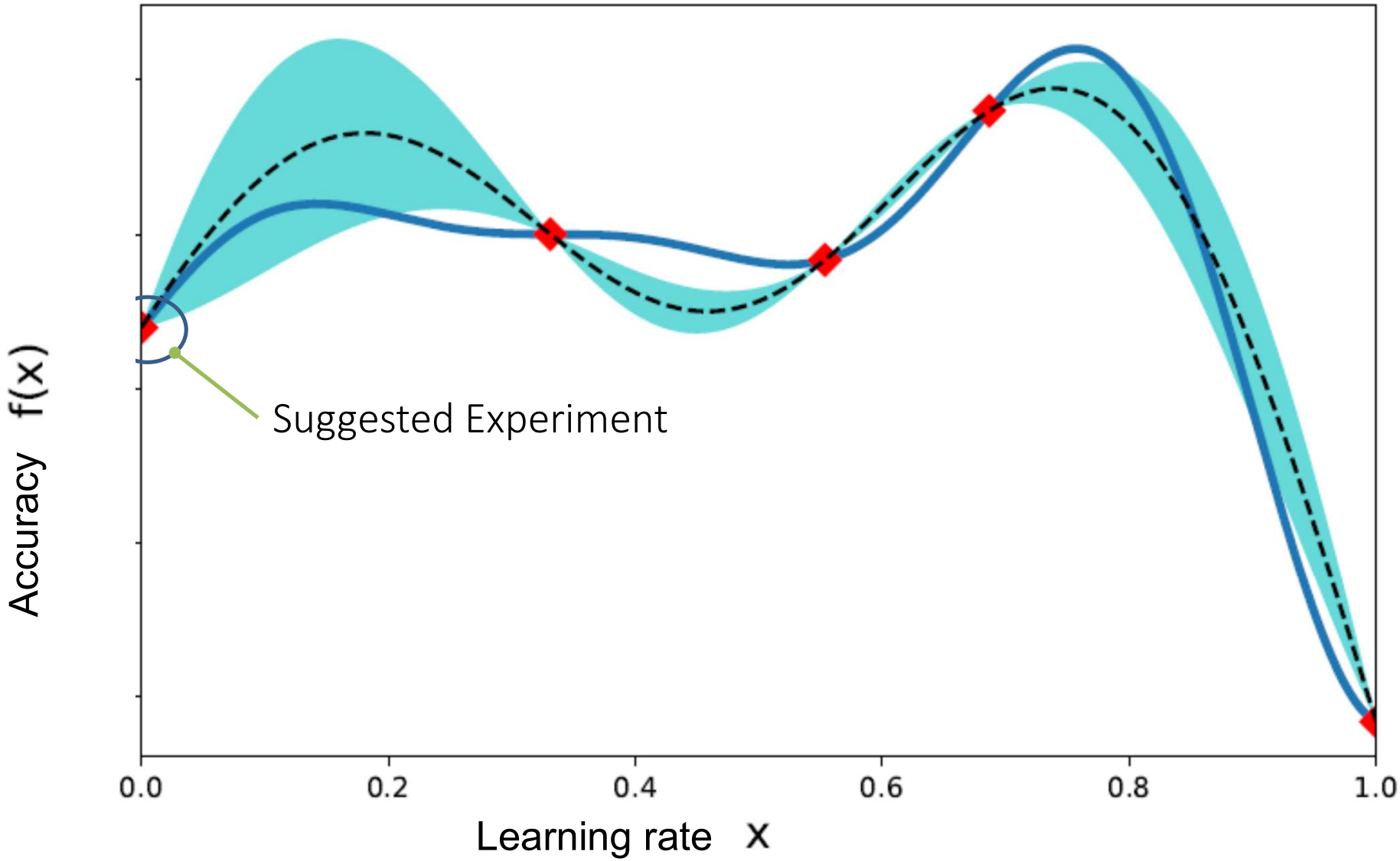


Illustration of Bayes Opt (6 points)

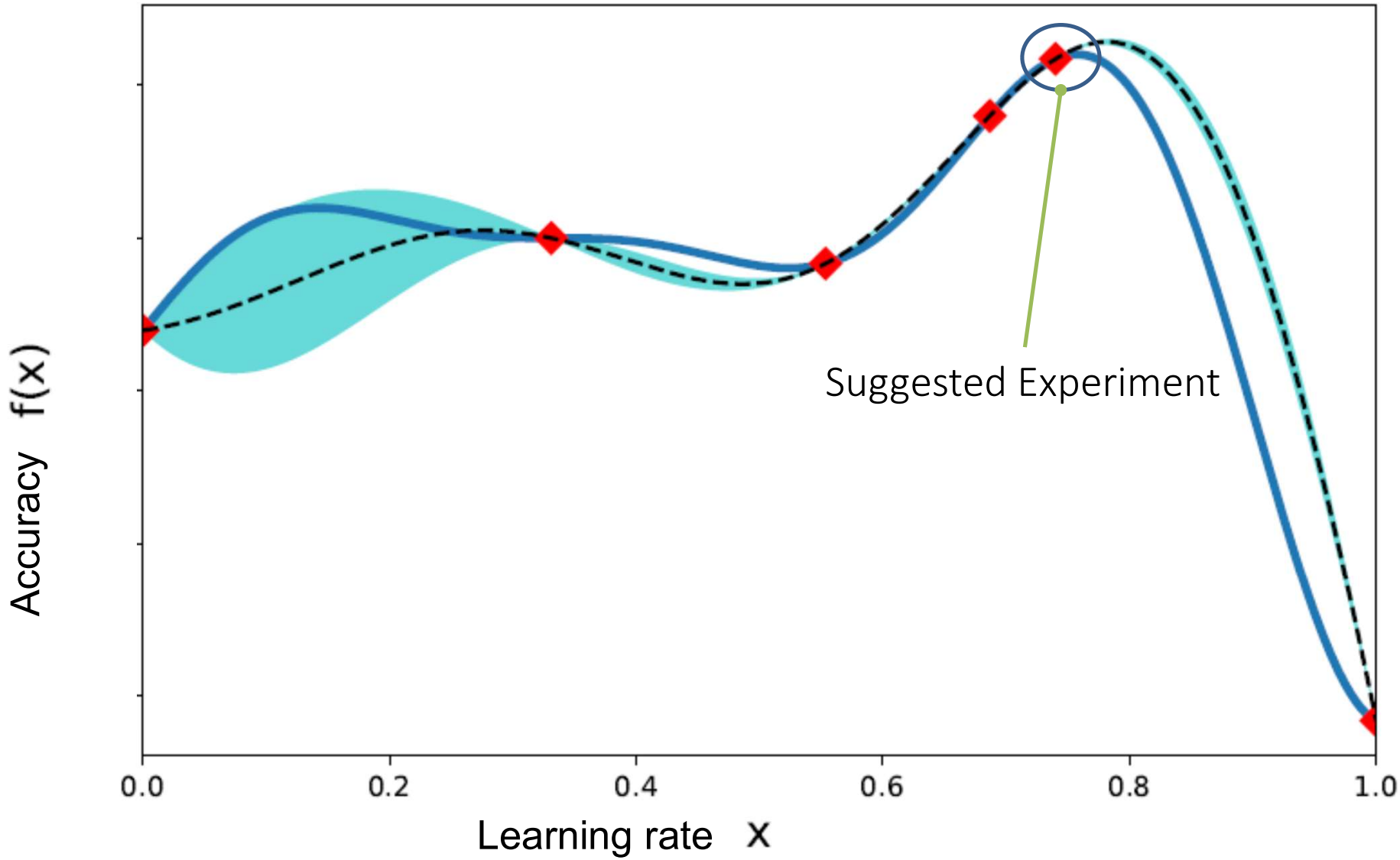


Illustration of Bayes Opt (7 points)

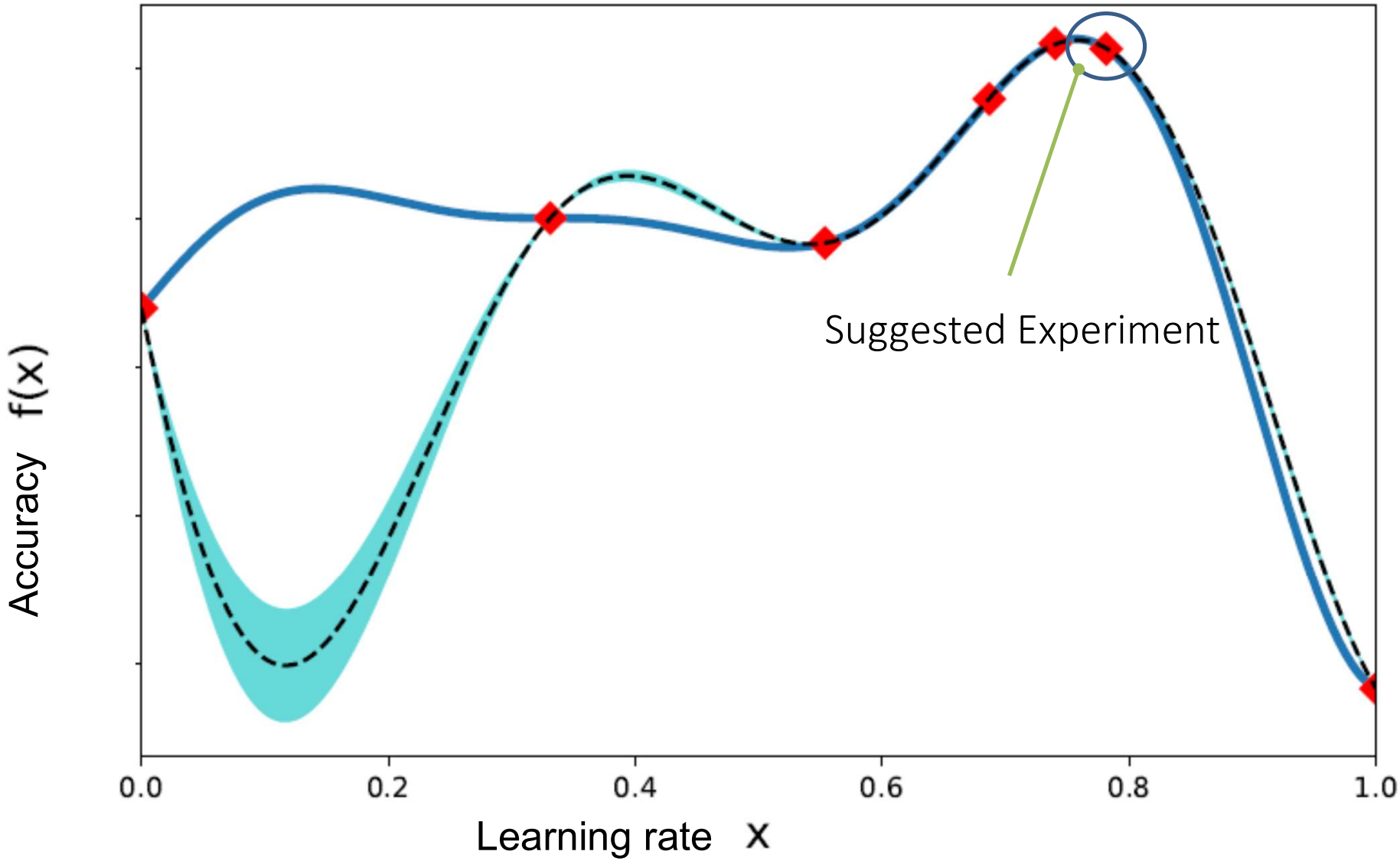
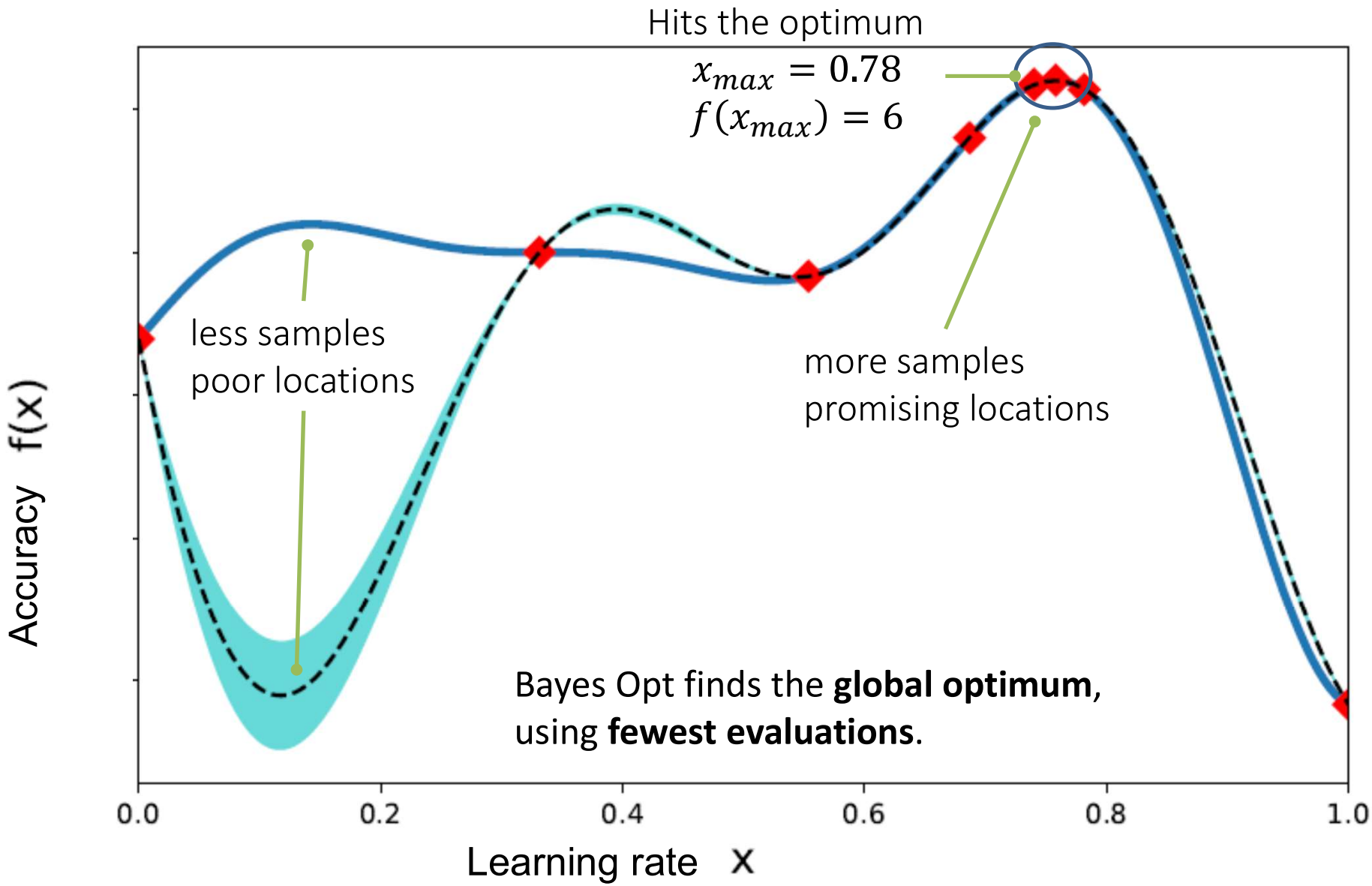
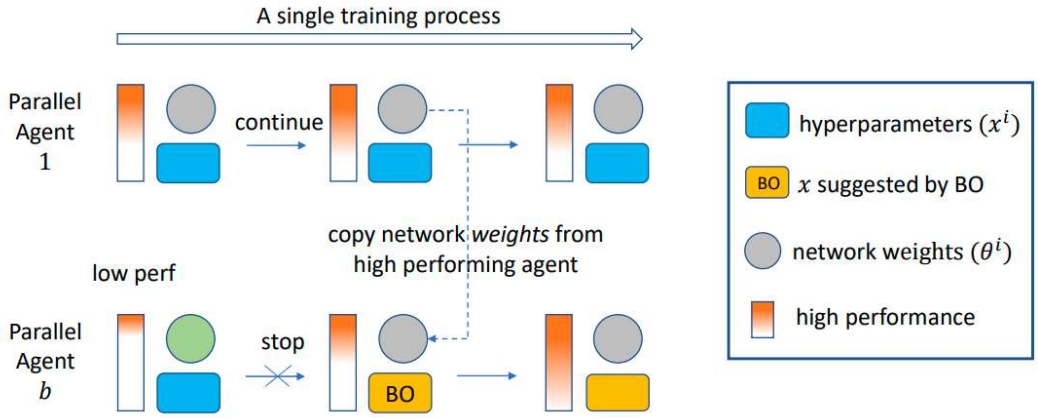


Illustration of Bayes Opt (8 points)



Outline

- Bayesian optimisation and Illustration
- Mixed categorical-continuous optimisation
- Mixed optimization in high dimensional space
- Mixed optimization in population-based AutoRL



Bayes Opt Mixed Categorical – Continuous Input

- Tuning hyperparameters for deep neural network

continuous variables | **categorical variables**

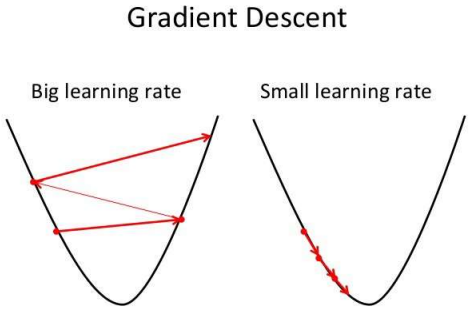
- $y = f($ x_1 , x_2 | x_3 , x_4 $)$

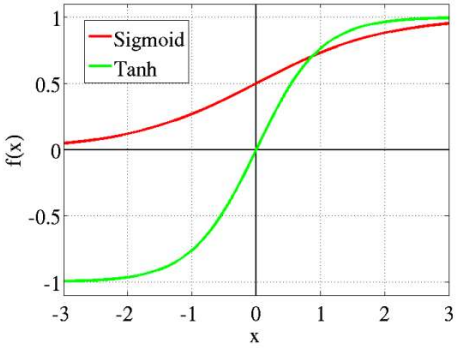
learning rate $\in [1e^{-6}, 1e^{-1}]$

weight decay $\in [1e^{-6}, 1e^{-1}]$

optimiser type $\in \{SGD, Adam, \dots\}$

activation type $\in \{tanh, sigmoid, \dots\}$





Bayes Opt Mixed Categorical – Continuous Input

- One-hot encoding:
  - Red: [1,0,0] Green: [0,1,0] Blue: [0,0,1]

- Drawbacks:
  - Make the search space *large*.

  if C =4 categories, each has $V = 5$ choices=> 20 extra dimensions.
  - Non-continuous and non-differentiable space

- Challenging in optimizing mixed-type: categorical - continuous

Bayes Opt Mixed Categorical – Continuous Input

continuous **categorical**

$$y = f(\overbrace{x_1 , x_2}, \overbrace{x_3 , x_4})$$

learning rate $\in [1e^{-6}, 1e^{-1}]$

weight decay $\in [1e^{-6}, 1e^{-1}]$

optimiser type $\in \{SGD, Adam, \dots\}$

activation type $\in \{tanh, sigmoid, \dots\}$

Bayes Opt

Multi-armed Bandit

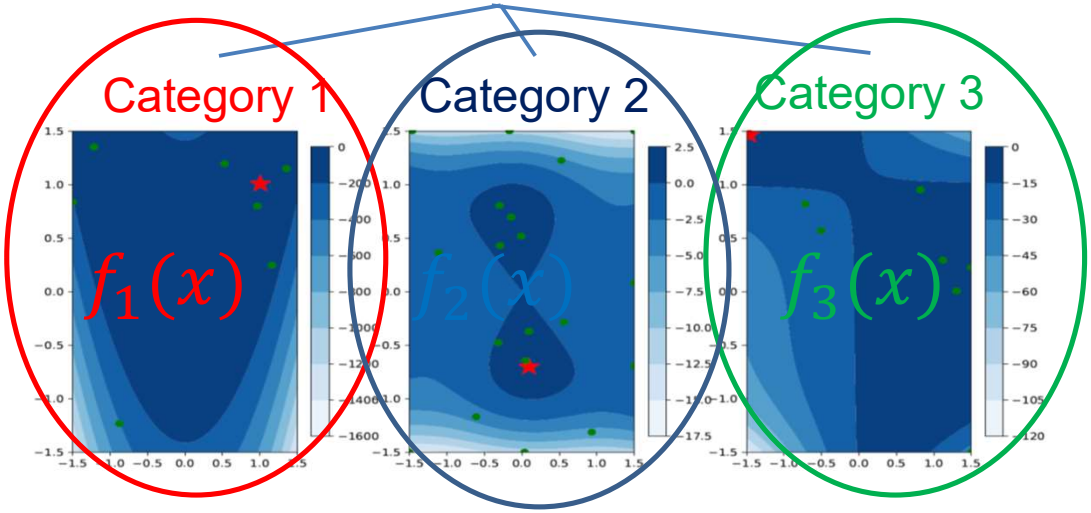
Two settings in mixed variables optimization

1. Continuous variable is specific to categorical variable

2. Continuous is sharing across categorical variables

Two settings in mixed variables optimization

1. Continuous variable is specific to categorical variable



RBF Polynomial Linear categorical

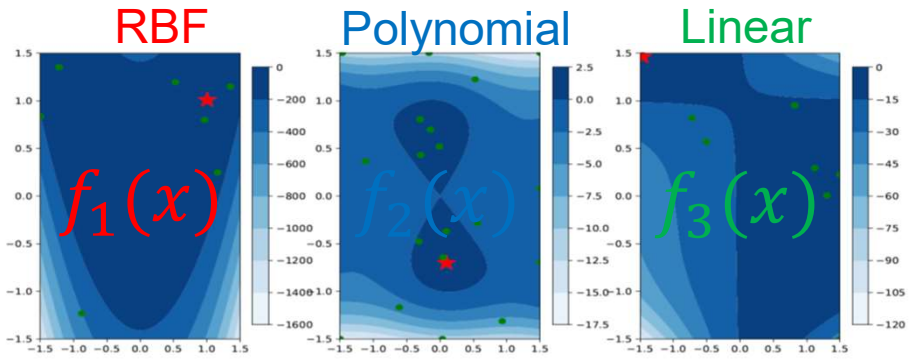
Kernel parameter is specific to the kernel type.

lengthscale degree none continuous

dimensions for the continuous: **1** or **2** ?

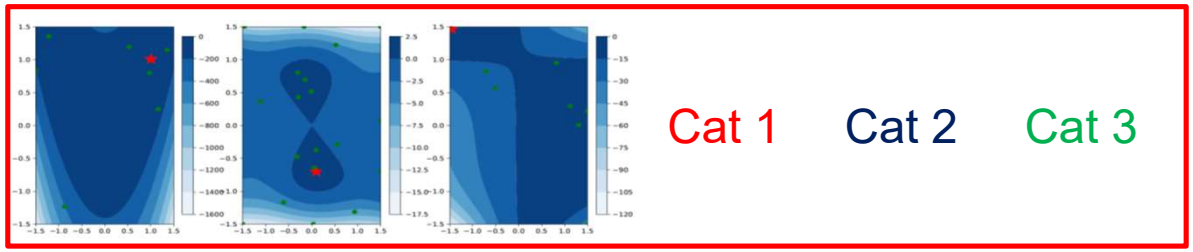
Two settings in mixed variables optimization

1. Continuous variable is specific to categorical variable



2. Continuous is sharing across categorical variables

Learning rate *in DL is sharing across* ***activation types***



continuous

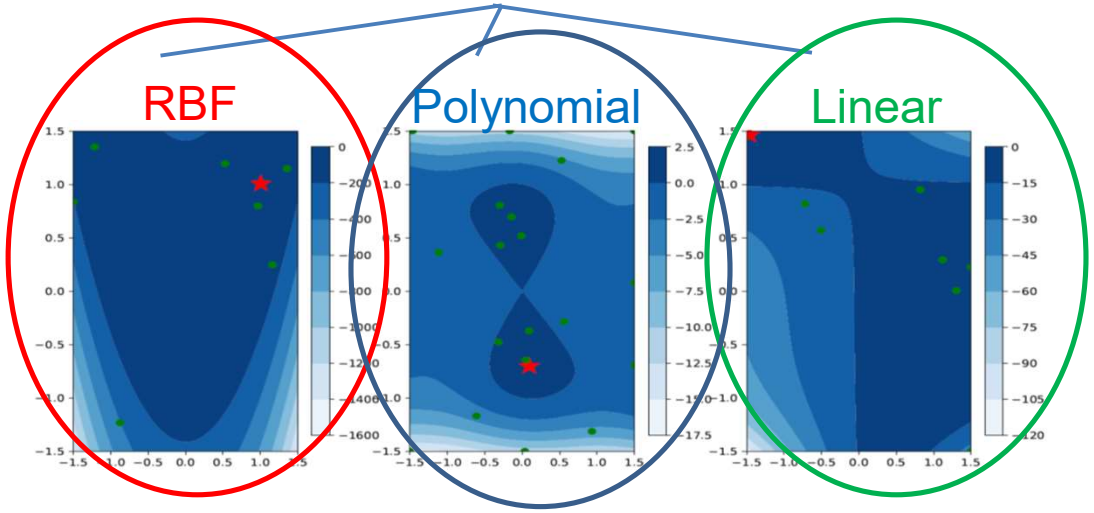
categorical

$f(x, h)$

continuous

Two settings in mixed variables optimization

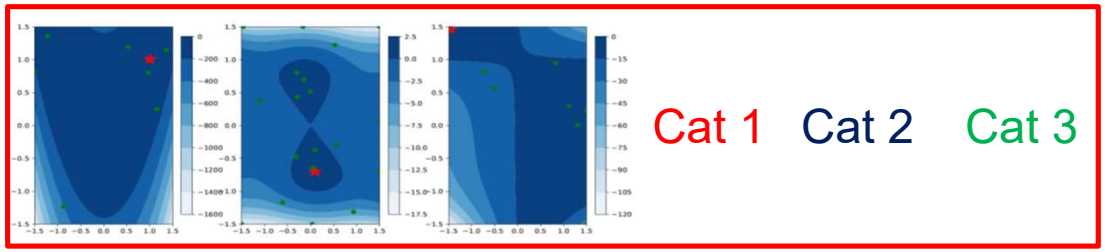
1. Continuous variable is specific to categorical variable



$f_1(x)$ $f_2(x)$ $f_3(x)$

- Independent/local functions

2. Continuous is sharing across categorical variables



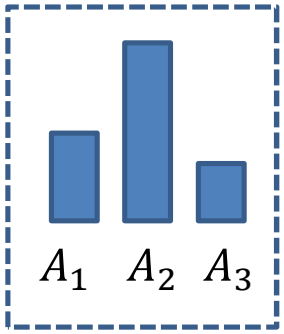
continuous

- Global function $f(x, h)$

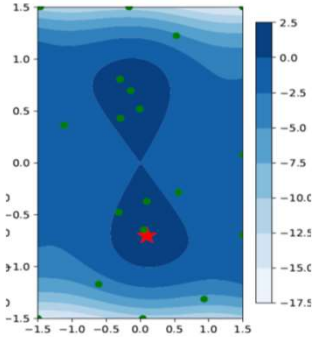
Algorithm overview

Optimize categorical $\boldsymbol{h}$

MAB



Optimize continuous $\boldsymbol{x}$ given $\boldsymbol{h}$

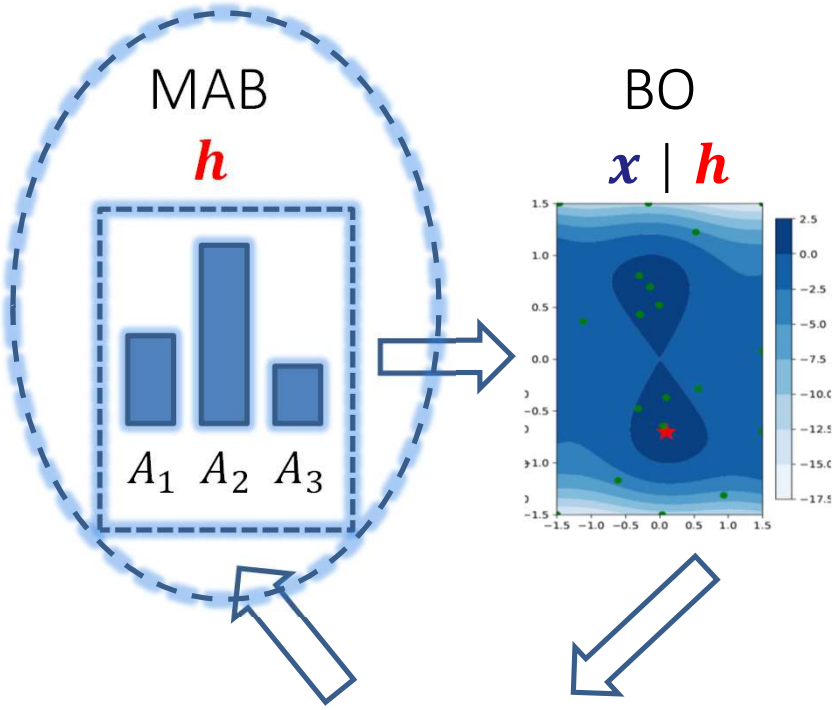


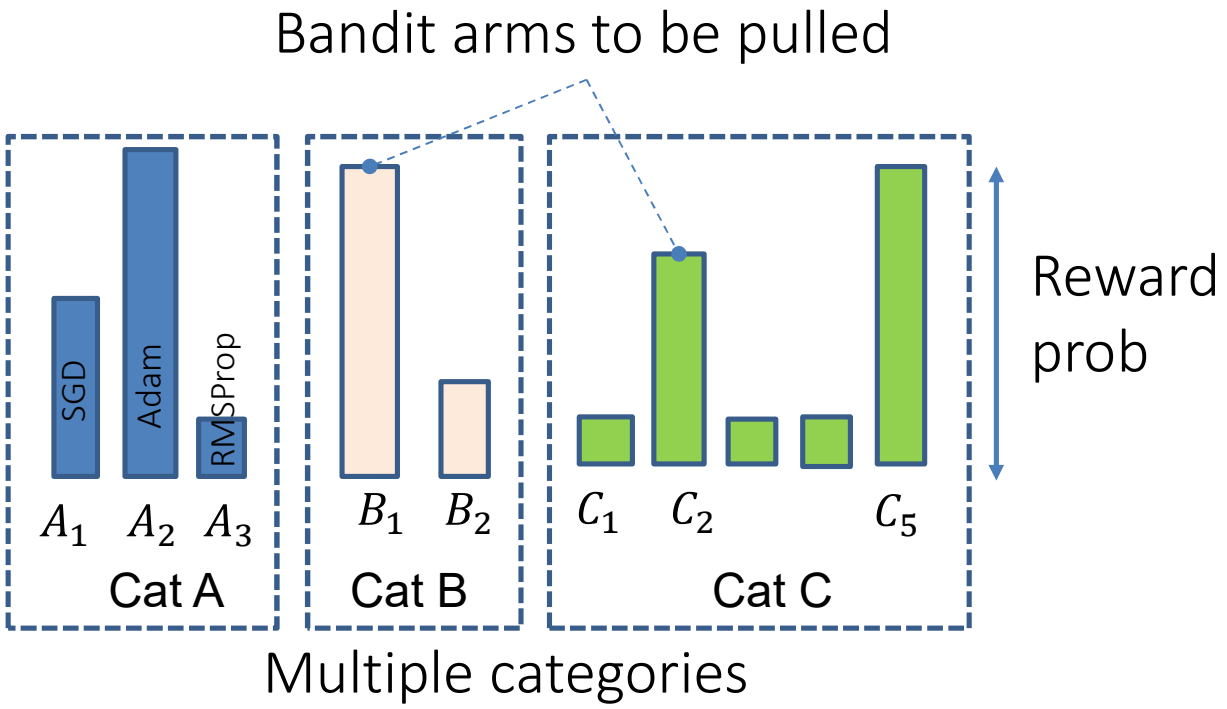
Bayes Opt

Observe the feedback

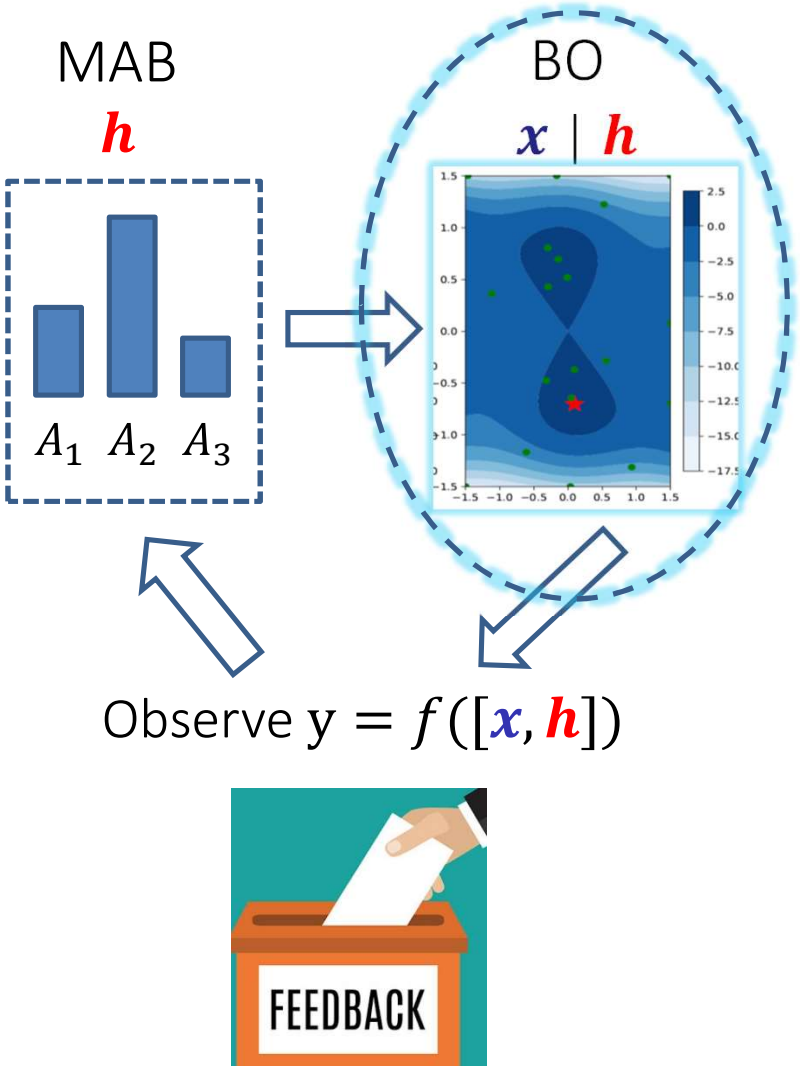
$\mathrm{y} = f([\boldsymbol{x}, \boldsymbol{h}])$

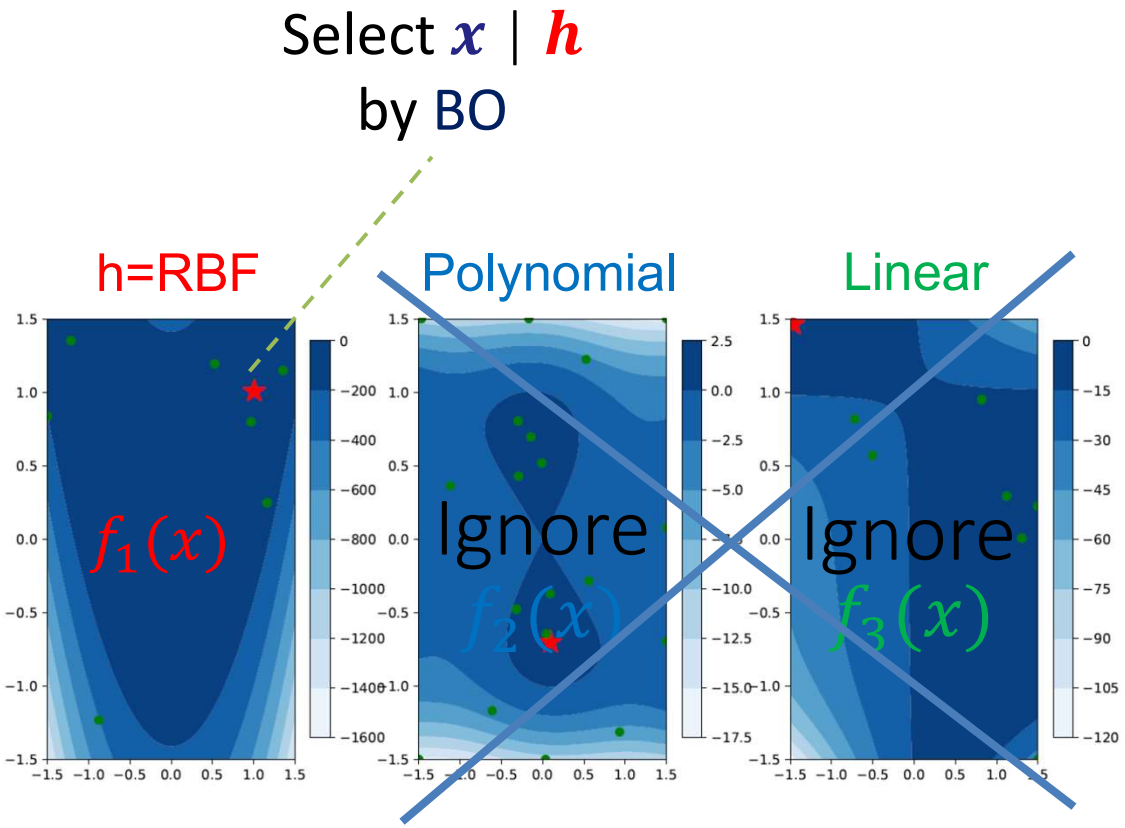
Optimize h by multi-armed bandits





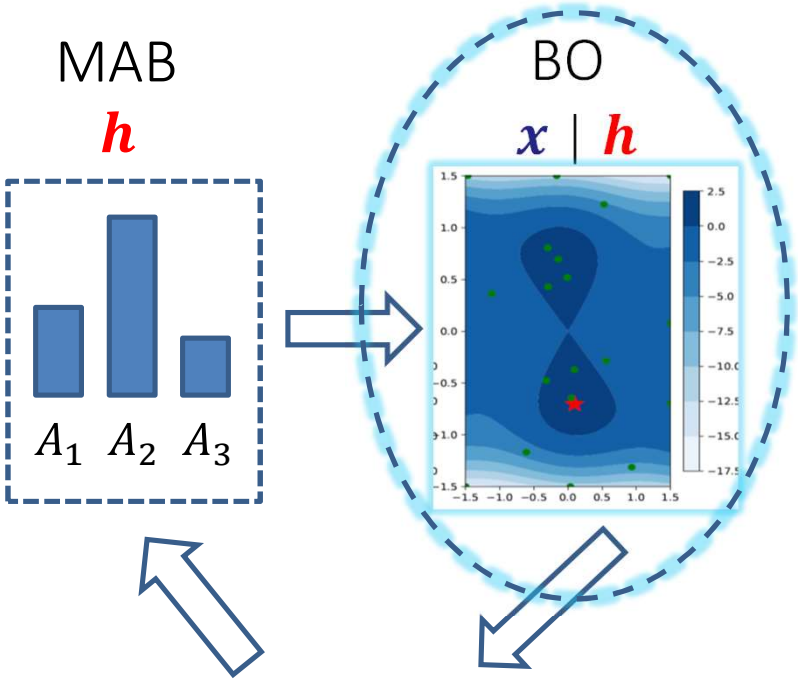
Setting (1): continuous is categorical-specific





S. Gopakumar, S. Gupta, S. Rana, V. Nguyen, & S. Venkatesh. Algorithmic assurance: An active approach to algorithmic testing using BO. *NeurIPS'18*

Setting (2): continuous shared across categories



- Using Bayes Opt to find $\boldsymbol{x}$ in a global space across all categories $\boldsymbol{h}$.
- Joint kernel for both $\boldsymbol{x}$ and $\boldsymbol{h}$
  - Additive $k(\boldsymbol{x}, \boldsymbol{x}') + k(\boldsymbol{h}, \boldsymbol{h}')$
  - Multiplicative $k(\boldsymbol{x}, \boldsymbol{x}') \times k(\boldsymbol{h}, \boldsymbol{h}')$
  - $k(z, z') = (1 - \lambda)k(\boldsymbol{x}, \boldsymbol{x}')k(\boldsymbol{h}, \boldsymbol{h}') + \lambda[k(\boldsymbol{x}, \boldsymbol{x}') + k(\boldsymbol{h}, \boldsymbol{h}')]$
  - λ can be estimated from the data.

B. Ru, A. Alvi, **V. Nguyen**, M. Osborne, & S. Roberts. Bayesian optimisation over multiple continuous and categorical inputs. *ICML'20*

Visualization of the Algorithm

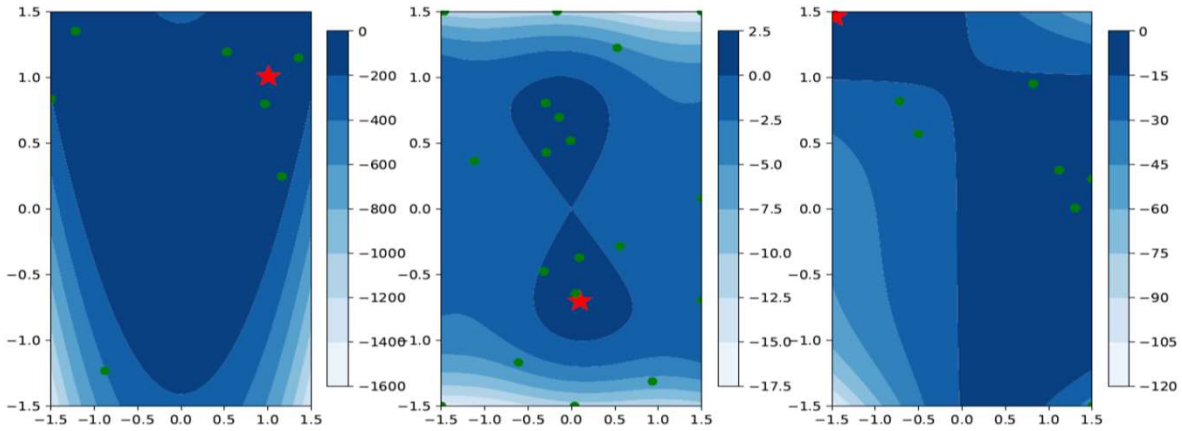
Categorical

Continuous

RBF

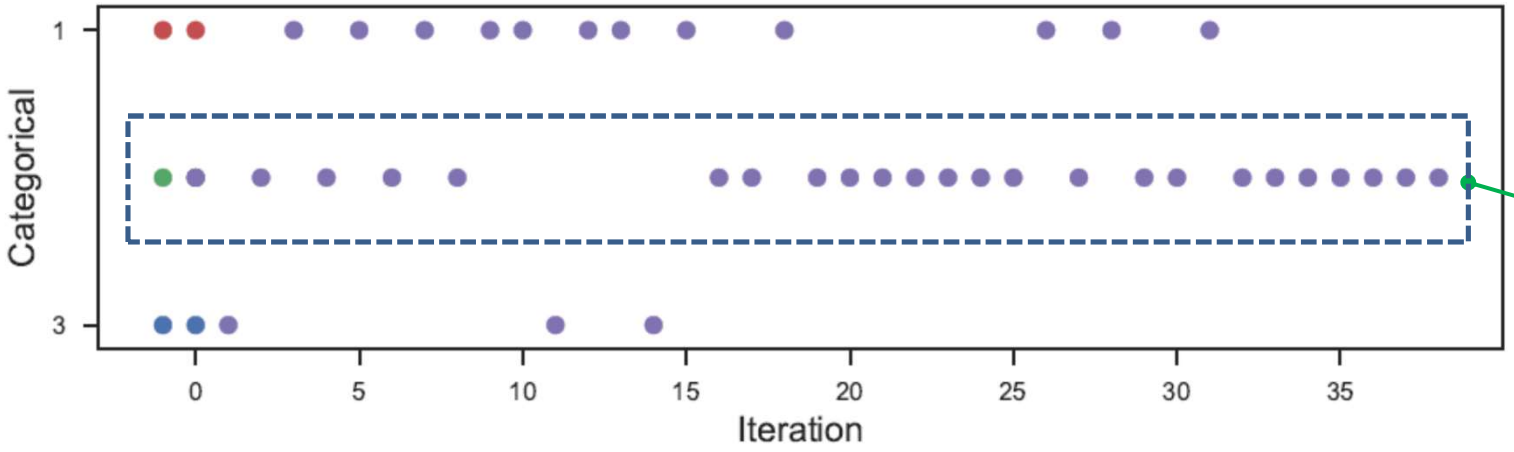
Polynomial

Linear



MAB

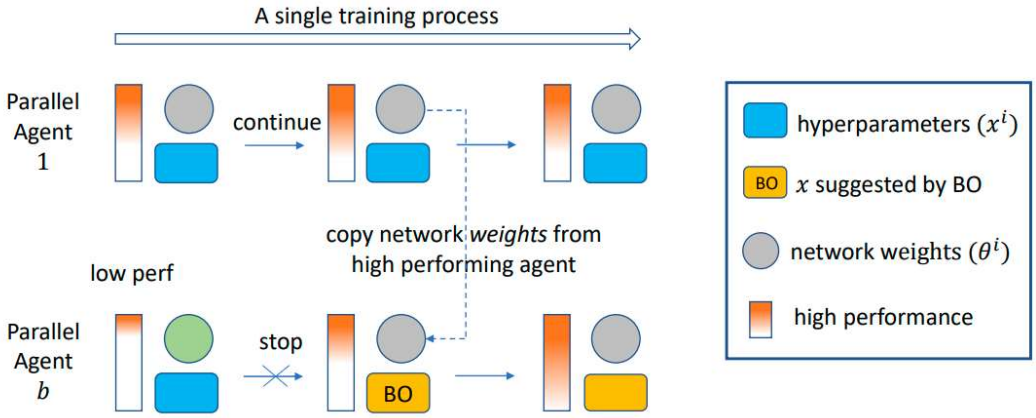
Bayes Opt



Concentrate on Polynomial with **higher (expected) value**.

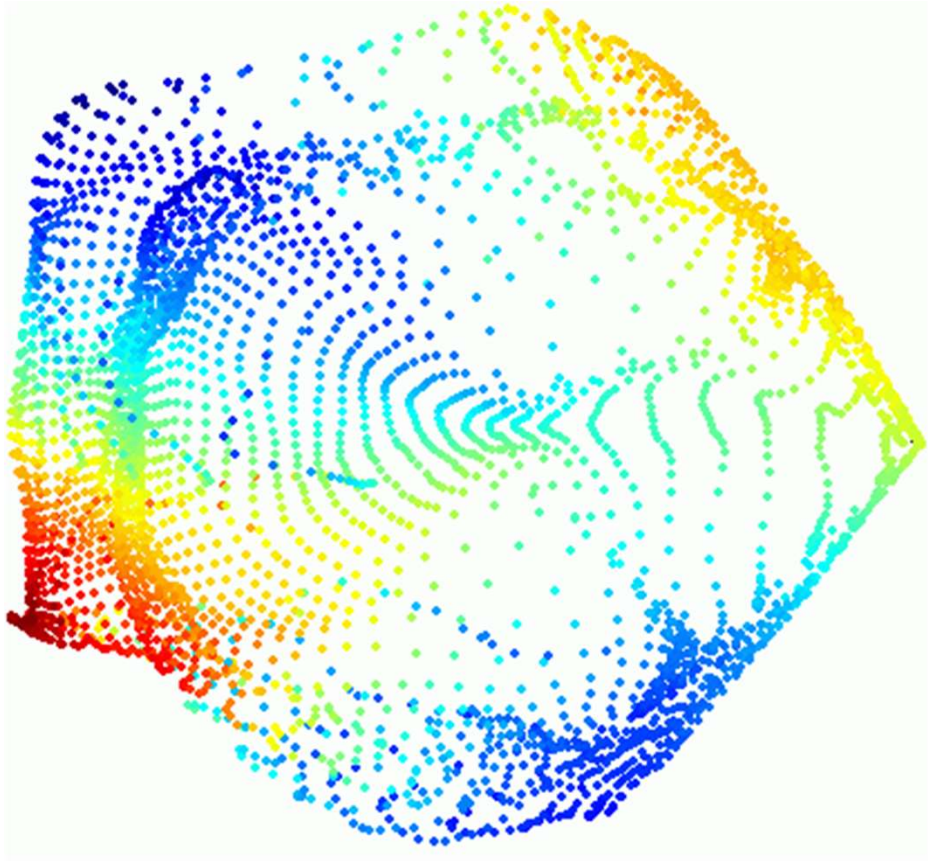
Outline

- Bayesian optimisation and Illustration
- Mixed categorical-continuous optimization
- Mixed optimization in high dimensional space
- Mixed optimization in population-based AutoRL



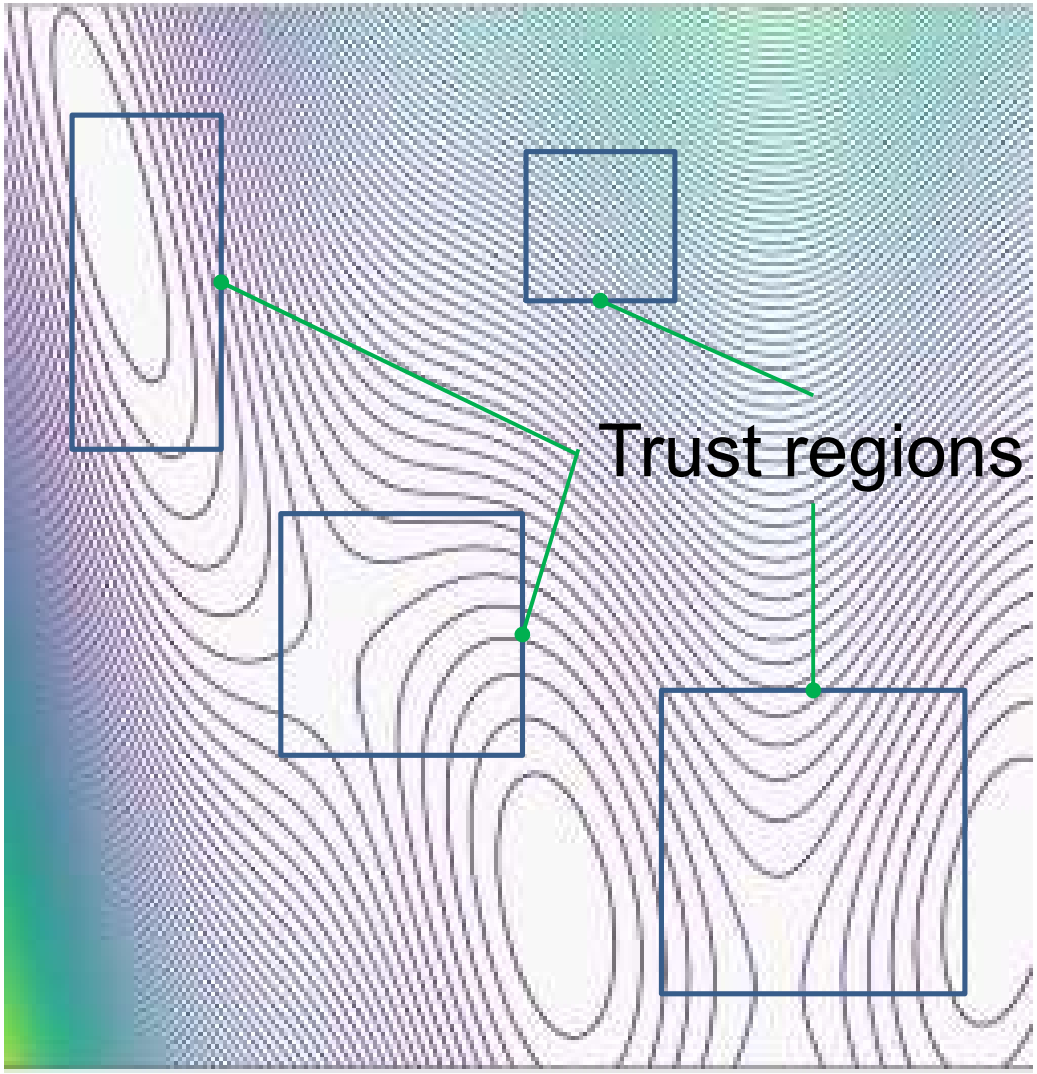
Mixed optimization with 200 dimensions ?

- High dimension causes problem for optimization.
  - Statistical challenge: the search space grows exponentially
  - Computational challenge: global optimizers fail to return an optimum within limited time and resource



Local Trust Optimization

- Build multiple trust regions
- Perform local optimization in each trust region
- The local region is narrower for the local optimizer to be successful

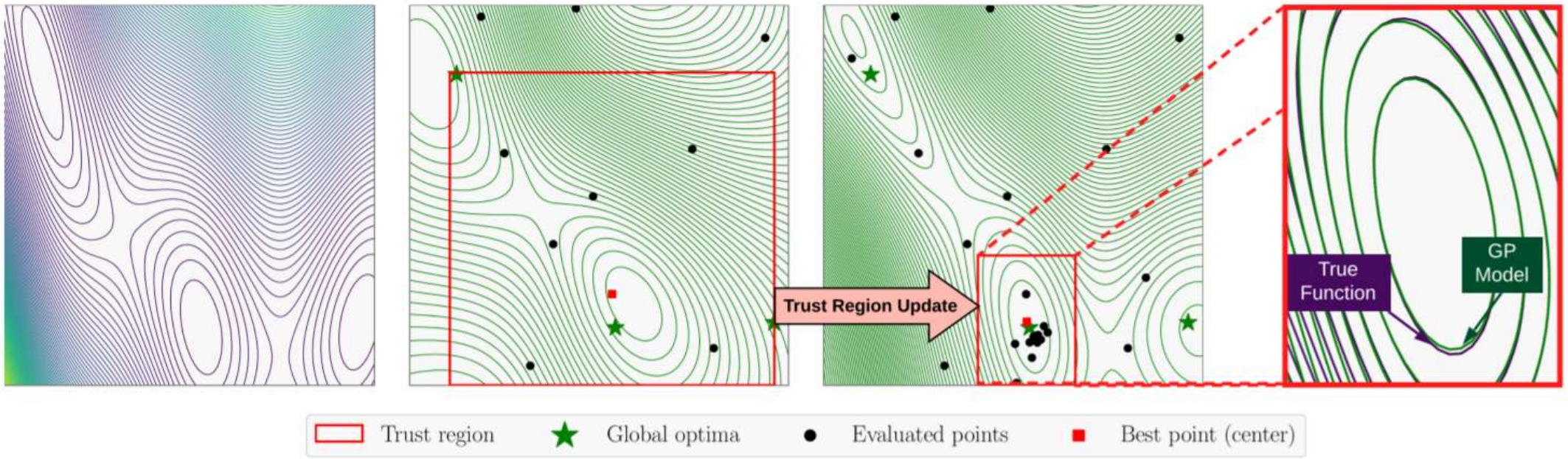


Eriksson et al. "Scalable global optimization via local Bayesian optimization." *NeurIPS'19*

Local Trust Optimization

- Iteratively
  - expand the trust region for exploration if successes
  - shrink the trust region for exploitation if failures
  - terminate and randomly jump to another region if local search is exhausted

exploration

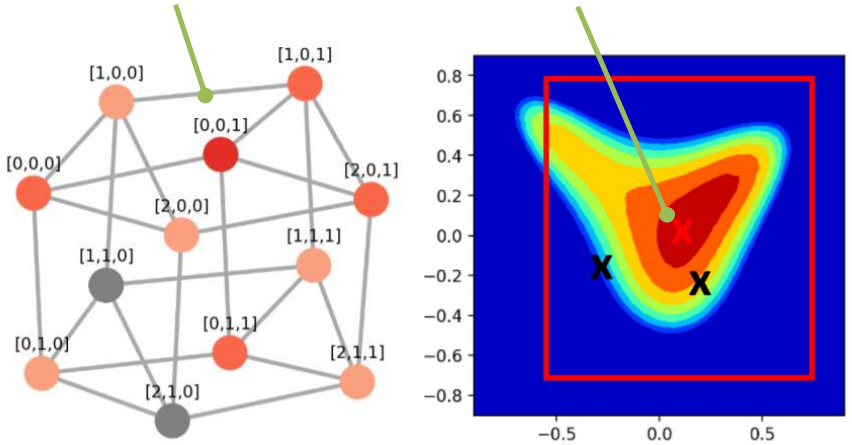


Given **limited observations**, the GP estimation is *more accurate* within a *local region*

CASMOPOLITAN

- Extending **local trust** idea for **mixed categorical-continuous** variables

  - Separate trust regions: categorical and continuous, defined from the best seen location



  - After a local search is exhausted, a new **trust region** is selected using the UCB

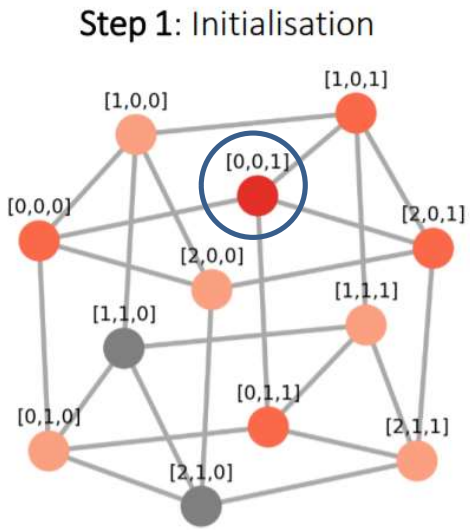
exploit promising region

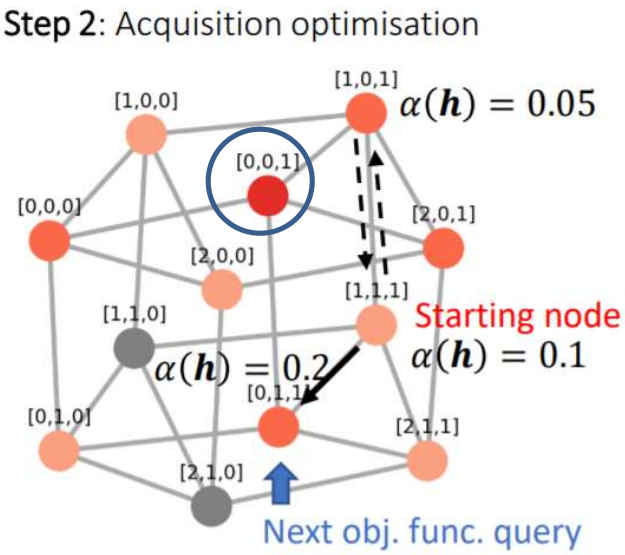
$$\mathbf{h}_i^{(0)} = \arg\max_{\mathbf{h}\in\mathcal{H}} \mu_{gl}(\mathbf{h}; D^*_{i-1}) + \sqrt{\beta_i}\sigma_{gl}(\mathbf{h}; D^*_{i-1})$$

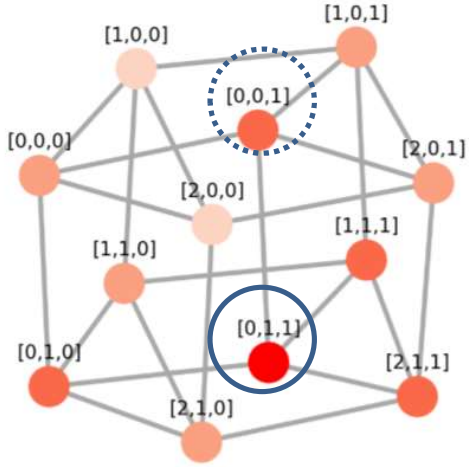
explore region with high uncertainty

X. Wan, **V. Nguyen**, H. Ha, B. Ru, C. Lu, and M. Osborne. "Think Global and Act Local: Bayes Opt over High-Dimensional Categorical and Mixed Search Spaces." *ICML'21*

Illustration of Updating Trust Region Center







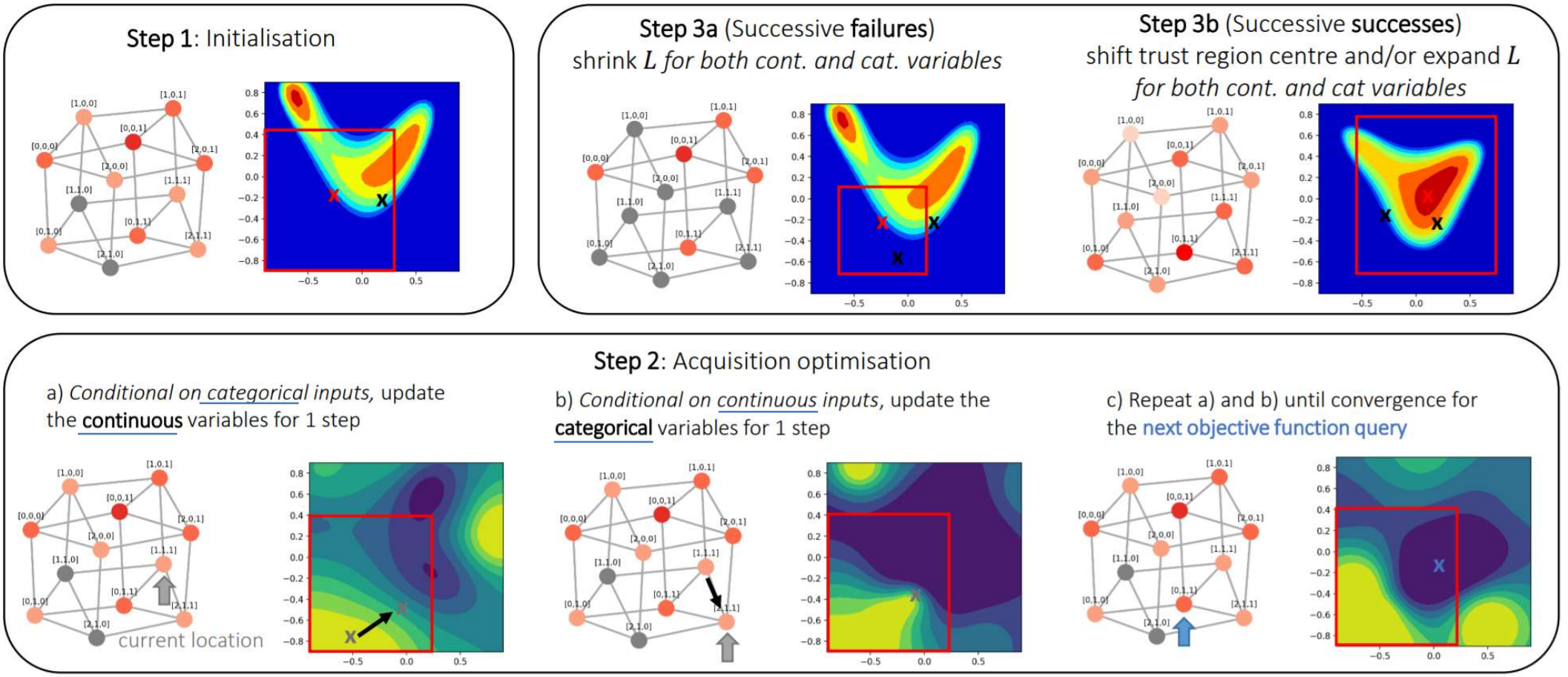
At unseen nodes:
(i) Predict the expected function value and uncertainty
(ii) Calculate acq func value

Exploit/explore to select the highest value node to query the black-box

Update trust region center
Rebuild the local trust region

Illustration of the algorithm in the mixed space

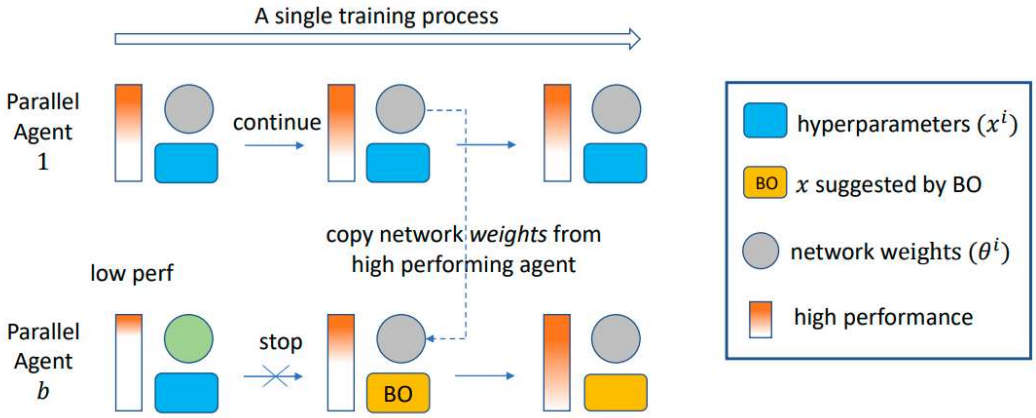
- Iteratively expand, shrink and restart the local search
- Natural extension to handle parallelism



https://github.com/xingchenwan/Casmopolitan

Outline

- Bayesian optimisation and Illustration
- Mixed categorical-continuous optimisation
- Mixed optimization in high dimensional space
- Mixed optimization in population-based AutoRL



Population Based Training (PBT)

Population Based Training of Neural Networks

Max Jaderberg Valentin Dalibard Simon Osindero Wojciech M. Czarnecki
Jeff Donahue Ali Razavi Oriol Vinyals Tim Green Iain Dunning
Karen Simonyan Chrisantha Fernando Koray Kavukcuoglu
DeepMind, London, UK

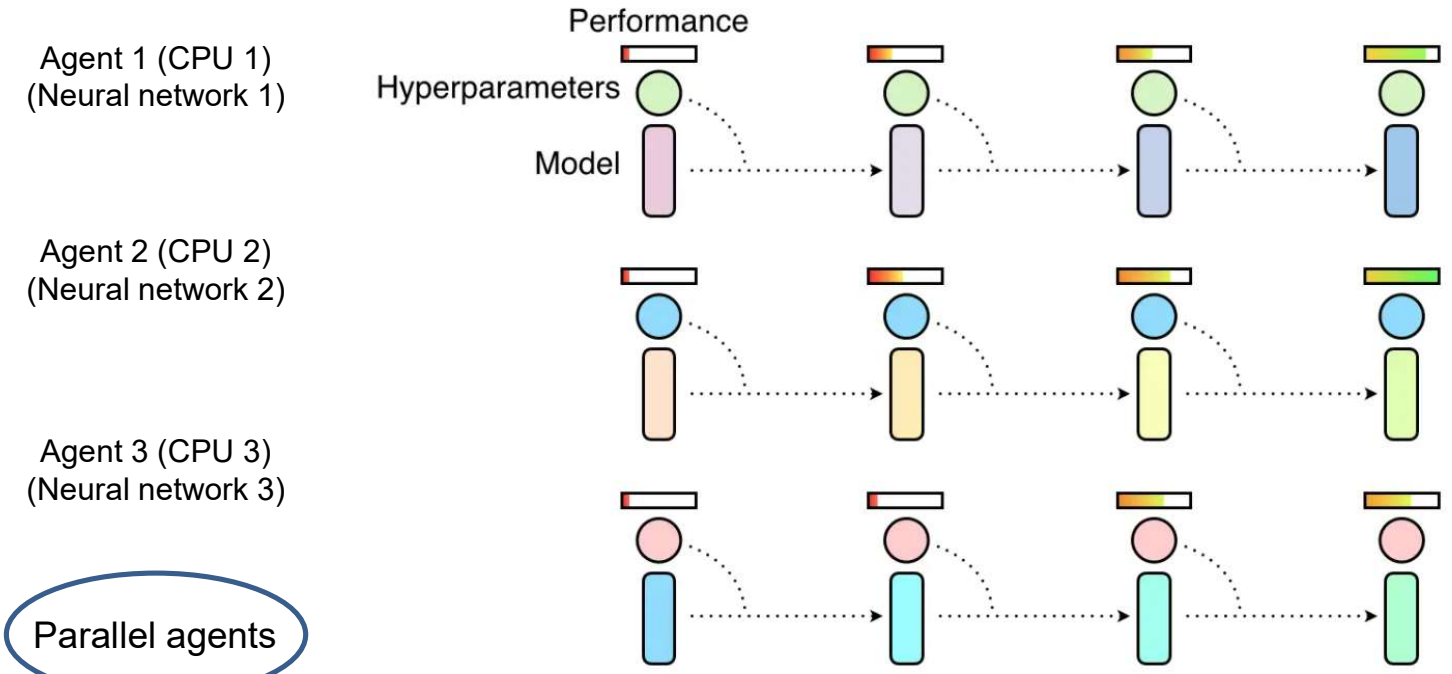
Abstract

Neural networks dominate the modern machine learning landscape, but their training and success still suffer from sensitivity to empirical choices of hyperparameters such as model architecture, loss function, and optimisation algorithm. In this work we present *Population Based Training (PBT)*, a simple asynchronous optimisation algorithm which effectively utilises a fixed computational budget to jointly optimise a population of models and their hyperparameters to maximise performance. Importantly, PBT discovers a schedule of hyperparameter settings rather than following the generally sub-optimal strategy of trying to find a single fixed set to use for the whole course of training. With just a small modification to a typical distributed hyperparameter training framework, our method allows robust and reliable training of models. We demonstrate the effectiveness of PBT on deep reinforcement learning problems, showing faster wall-clock convergence and higher final performance of agents by optimising over a suite of hyperparameters. In addition, we show the same method can be applied to supervised learning for machine translation, where PBT is used to maximise the BLEU score directly, and also to training of Generative Adversarial Networks to maximise the Inception score of generated images. In all cases PBT results in the automatic discovery of hyperparameter schedules and model selection which results in stable training and better final performance.

Human-level performance in 3D multiplayer games with population-based reinforcement learning

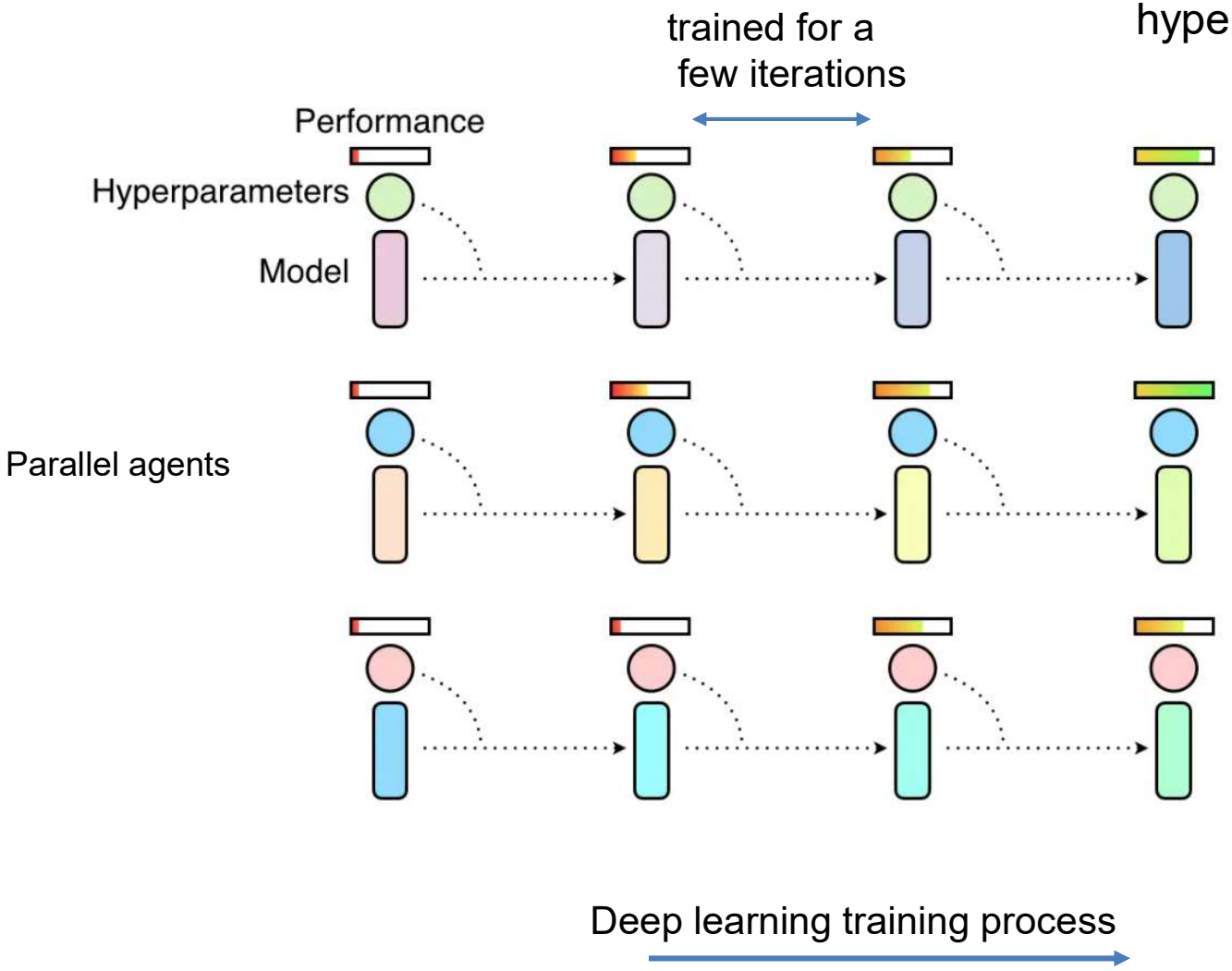
Max Jaderberg*,†, Wojciech M. Czarnecki*,†, Iain Dunning†, Luke Marris, Guy Lever, Antonio Garcia Castañed…
+ See all authors and affiliations

Science 31 May 2019:
Vol. 364, Issue 6443, pp. 859-865
DOI: 10.1126/science.aau6249



Population Based Training (PBT)

each agent receives a (different) hyperparameter to train

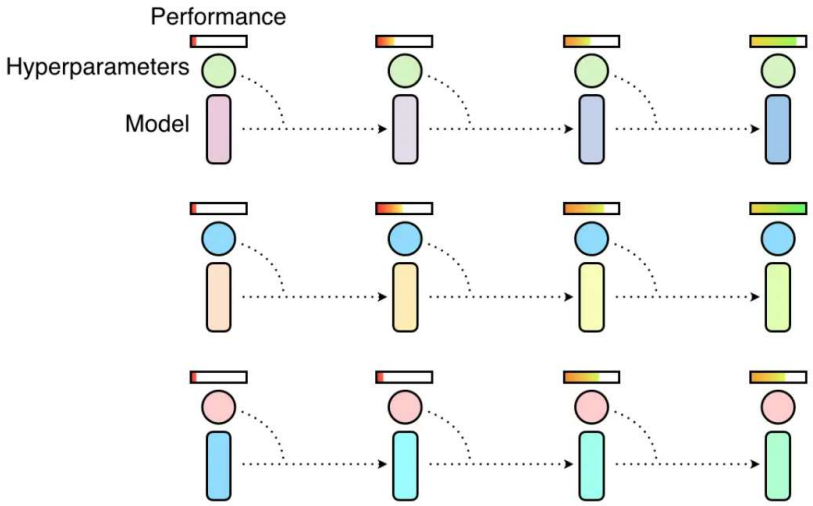


Population Based Training

Terminates poor-performing agents and restarts by:

- Copies a network weight from the best-performing agent.
- Generate a new hyperparameter by randomly perturb from the best agent.

At the end of this single training process, we obtain the “well-trained” model.



Two Key Advantages of PBT

- Learning a schedule of hyperparameters, such as selecting **large** learning rate at the beginning and **smaller** at the later stage.
  - Existing HPO: a single set of hyperparameters are fixed during training.
  - PBT: perfectly adapts the hyperparameters.

- Time efficiency:
  - Existing HPO: repeatedly evaluate the black-box with different hypers.
  - PBT: uses a single training run.

Population Based Bandit (PB2)

- PB2 is an extension of Population Based Training (PBT).
- The hyperparameters update is controlled by **time-varying Gaussian process bandit optimization**.

initial reward (constant)

x_t: hyperparameter at iteration t

$$F_T(x_T) - F_1(x_1) = F_T(x_T) - F_{T-1}(x_{T-1}) + \cdots + F_3(x_3) - F_2(x_2) + F_2(x_2) - F_1(x_1)$$

final reward

reward improvement between 2 timesteps

reward improvement between 2 timesteps

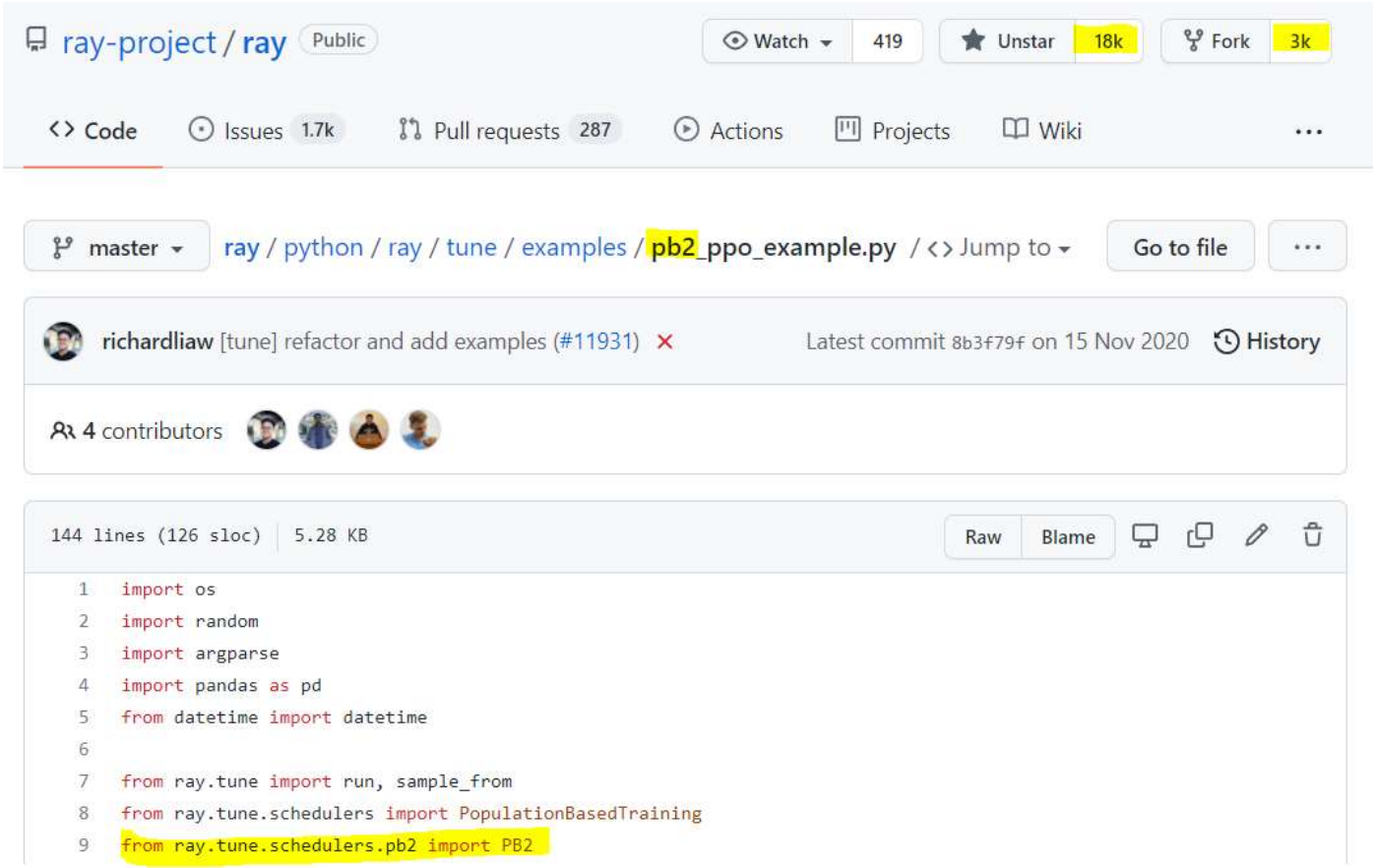
$$\max\left[F_T(x_T) - F_1(x_1)\right] = \max \sum_{t=1}^{T} F_t(x_t) - F_{t-1}(x_{t-1}) = \max \sum_{t=1}^{T} f_t(x_t) = \min \sum_{t=1}^{T} r_t(x_t)$$

Maximizing final reward = Minimizing cumulative regret

J. Parker-Holder, V. Nguyen, and S. Roberts. "Provably efficient online hyperparameter optimization with population-based bandits." *NeurIPS'20*

Population Based Bandit (PB2)

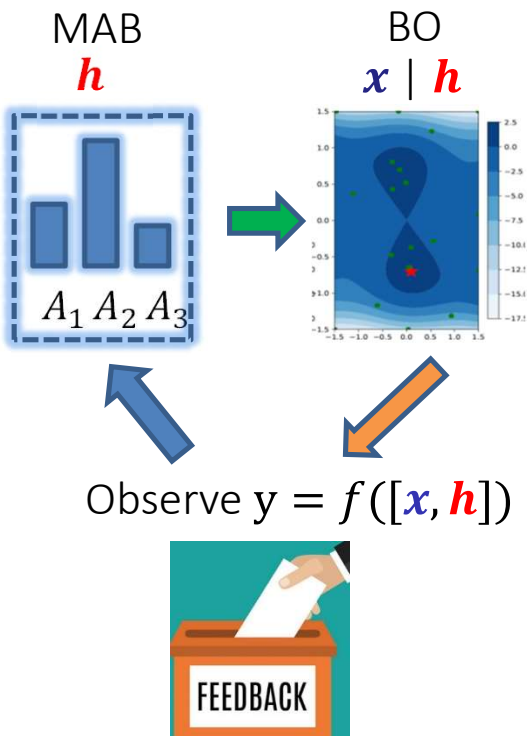
- PB2 is deployed in **Ray Tune** library (18k users).
  - Import **PB2** from **Ray** and try in a few line of codes.



J. Parker-Holder, **V. Nguyen**, and S. Roberts. "Provably efficient online hyperparameter optimization with population-based bandits." *NeurIPS'20*

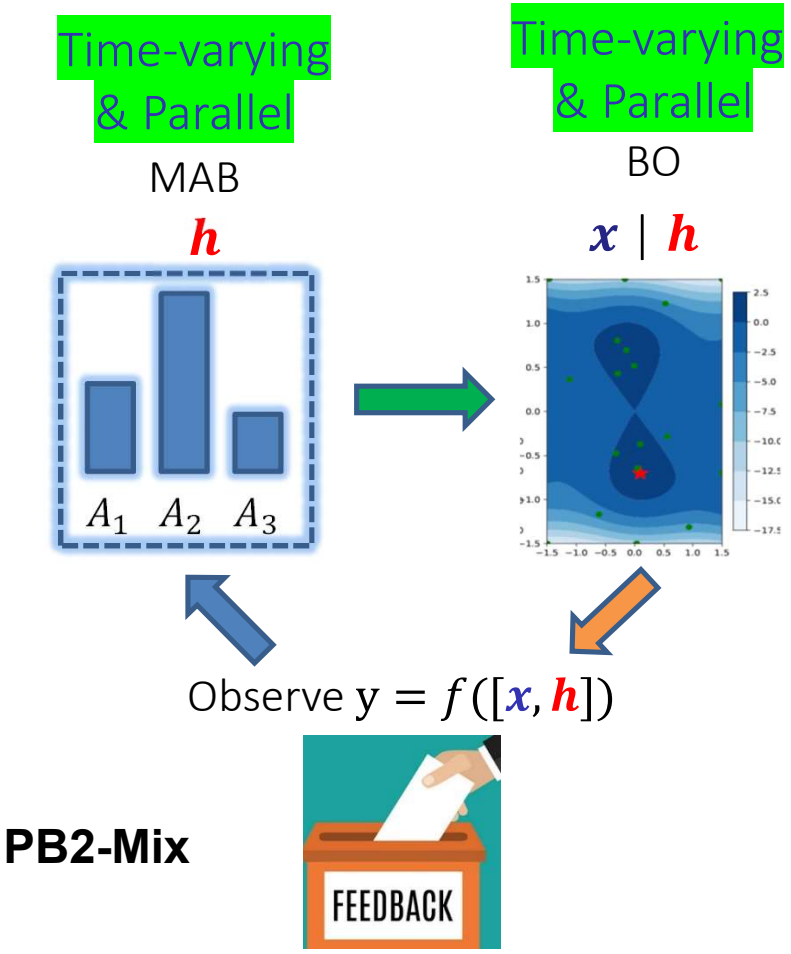
PB2-Mix

- PB2 is only applicable for continuous variables.
- PB2-Mix extends PB2 to handle mixed categ-cont variables



Existing works

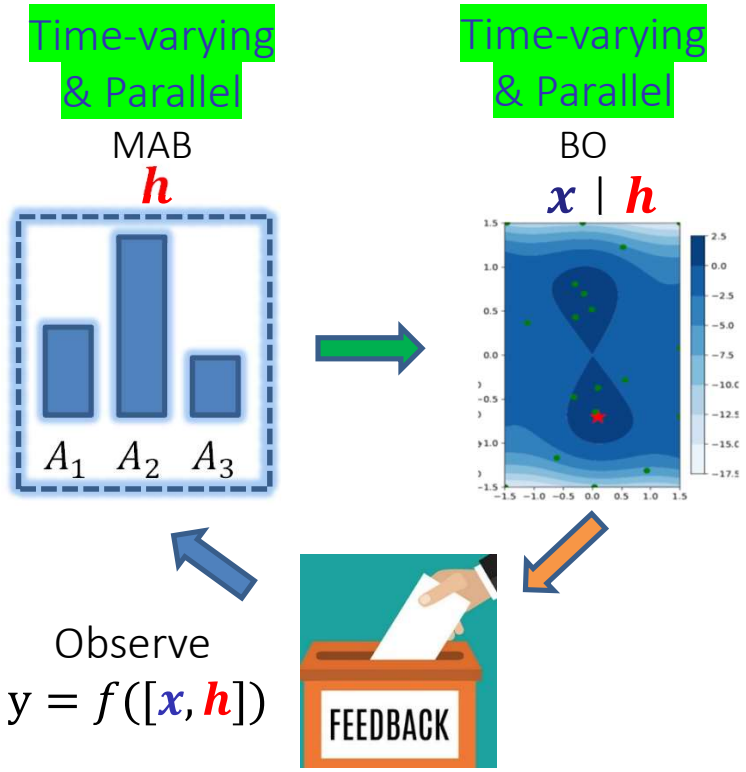
Time-varying & Parallel



PB2-Mix

J. Parker-Holder, V. Nguyen, S. Desai, and S. Roberts. "Tuning Mixed Input Hyperparameters on the Fly for Efficient Population Based AutoRL." *NeurIPS'21*

Time-varying Parallel Extension of MAB and BO



- Propose *Time-varying* and *Parallel* MAB to select categorical variables
- *Time-varying parallel* ***BO*** has been available in *PB2*

J. Parker-Holder, **V. Nguyen**, S. Desai, and S. Roberts. "Tuning Mixed Input Hyperparameters on the Fly for Efficient Population Based AutoRL." *NeurIPS'21*

TV.EXP3.M

- *Time-varying* and *Parallel* TV.EXP3.M to select categorical variable
  - EXP3 is a popular algorithm in multi-armed bandits
  - EXP3.M is the *Parallel* extension for EXP3
  - *Time-varying* extension for EXP3.M

Algorithm 5 TV.EXP3.M algorithm

Input: $\gamma = \sqrt{\frac{C\ln(C/B)}{(e-1)BT}}$, $\alpha = \frac{1}{T}$, C #categorical choice, T #max iteration, B #multiple play

Init $\omega_c = 1, \forall c = 1 \ldots C$ and denote $\eta = (\frac{1}{B} - \frac{\gamma}{C})\frac{1}{1-\gamma}$

for $t = 1$ to T **do** ← Iteratively

  if $\arg\max_{c\in[C]} \omega_c \geq \eta \sum_{c=1}^{C} \omega(c)$ **then**

    ν s.t. $\frac{\nu}{\eta} = \sum_{\omega_t(c)\geq\nu} \nu + \sum_{\omega_t(c)<\omega_t(c)} \omega_t(c)$

    Set $S_0 = (c : \omega_t(c) \geq \nu)$ and $\omega_t(S_0) = \nu$

  else

    Set $S_0 = \emptyset$

  end if

  Compute $p_t^c = B\left((1-\gamma)\frac{\omega_c}{\sum_{c=1}^{C}\omega_c} + \frac{\gamma}{C}\right), \forall c$

  $S_t = \text{DepRound}\left(B, [p_t^1 p_t^2 \ldots p_t^C]\right)$ ← Select a batch of arms (categorical variables)

  Observe the reward $g_t(c) = f(h_t = c)$ for $c \in S_t$

  $\hat{g}_t(c) = \frac{g_t(c)}{p_t^c}, \forall c \in S_t$ and $\hat{g}_t(c) = 0$ otherwise

  $\forall c \notin S_0$: update $\omega_c = \omega_c \times \exp\left(B\gamma\hat{g}_t(c)/C\right) + \frac{e\alpha}{C}\sum_{i=1}^{C}\omega_c$,

  $\forall c = S_0$: update $\omega_c = \omega_c + \frac{e\alpha}{C}\sum_{i=1}^{C}\omega_c$

end for

reward is time-varying

Theorem 1. *Set* $\alpha = \frac{1}{T}$ *and* $\gamma = \min\left\{1, \sqrt{\frac{C\ln(C/B)}{(e-1)BT}}\right\}$, *we assume the reward distributions changes at arbitrary instances, but the total number of change points is no more than* $V \ll \sqrt{T}$ *times. The expected regret of TV.EXP3.M satisfies the following sublinear bound*

$$\mathbb{E}\left[R_{TB}\right] \leq [1 + e + V]\sqrt{(e-1)\frac{CT}{B}\ln\frac{CT}{B}}.$$

Sublinear rate on the cumulative regret

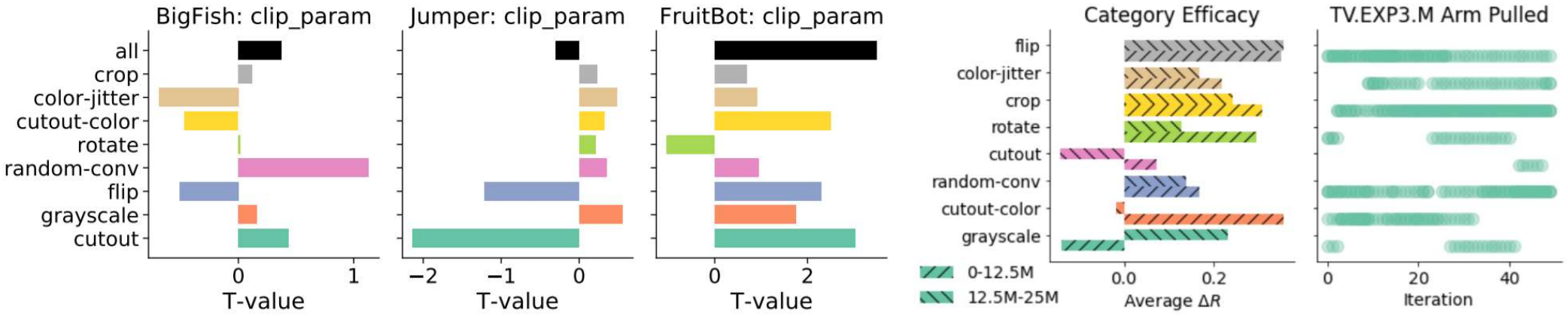
Uchiya, T., Nakamura, A. and Kudo, M., Algorithms for adversarial bandit problems with multiple plays. In *International Conference on Algorithmic Learning Theory.* 2010

J. Parker-Holder, V. Nguyen, S. Desai, and S. Roberts. "Tuning Mixed Input Hyperparameters on the Fly for Efficient Population Based AutoRL." *NeurIPS'21*

Data Augmentation and Hyper-optimization for AutoRL

- Augmentation types are categorical variables

- Hyperparameters are continuous
  - Regularization, PPO clip, learning rate and entropy exploration.



(a) Hyperparameter impact by augmentation type

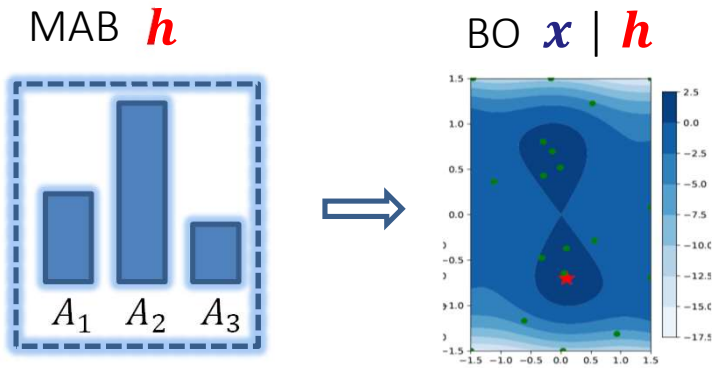
(b) Augmentation selection efficacy

J. Parker-Holder, **V. Nguyen**, S. Desai, and S. Roberts. "Tuning Mixed Input Hyperparameters on the Fly for Efficient Population Based AutoRL." *NeurIPS'21*

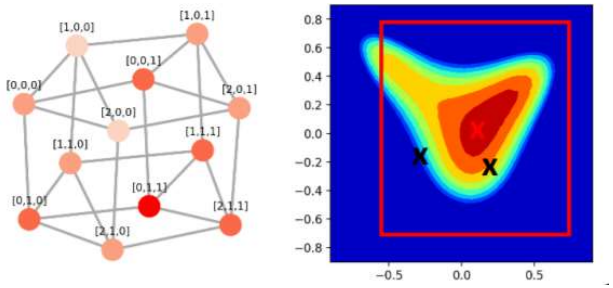
Takeaway: mixed categorical-continuous Bayes opt

$$\max f(\overbrace{x_1, x_2}^{\text{continuous}}, \overbrace{x_3, x_4}^{\text{categorical}})$$

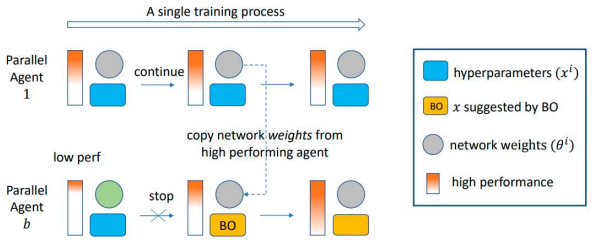
- MAB/BO for mixed optimization



- Local trust region for handling mixed optimization in high dimension



- Parallel time-varying bandit/BO for population-based AutoRL



References

- S. Gopakumar, S. Gupta, S. Rana, V. Nguyen, & S. Venkatesh. Algorithmic assurance: An active approach to algorithmic testing using BO. *NeurIPS'18*
- B. Ru, A. Alvi, A., V. Nguyen, M. Osborne, & S. Roberts. Bayesian optimisation over multiple continuous and categorical inputs. *ICML'20*
- Eriksson et al. "Scalable global optimization via local Bayesian optimization." *NeurIPS'19*
- X. Wan, V. Nguyen, H. Ha, B. Ru, C. Lu, and M. Osborne. "Think Global and Act Local: Bayes Opt over High-Dimensional Categorical and Mixed Search Spaces." *ICML'21*
- Bogunovic, I., Scarlett, J., & Cevher, V. Time-varying Gaussian process bandit optimization. *AISTATS 2016*
- Jaderberg, M., Dalibard, V., Osindero, S., Czarnecki, W.M., Donahue, J., Razavi, A., Vinyals, O., Green, T., Dunning, I., Simonyan, K. and Fernando, C. *Population based training of neural networks. arXiv preprint arXiv:1711.09846.*
- J. Parker-Holder, V. Nguyen, and S. Roberts. "Provably efficient online hyperparameter optimization with population-based bandits." *NeurIPS'20*
- Uchiya, T., Nakamura, A. and Kudo, M., Algorithms for adversarial bandit problems with multiple plays. In *International Conference on Algorithmic Learning Theory* 2010
- J. Parker-Holder, V. Nguyen, S. Desai, and S. Roberts. "Tuning Mixed Input Hyperparameters on the Fly for Efficient Population Based AutoRL." *NeurIPS'21*

Cocabo: https://github.com/rubinxin/CoCaBO_code

Casmopolitan: https://github.com/xingchenwan/Casmopolitan

PB2: https://github.com/jparkerholder/PB2

PB2-Mix: https://github.com/jparkerholder/procgen_autorl

Question and Answer

vu@ieee.org

@nguyentienvu

vu-nguyen.org

github.com/ntienvu