Model-based Asynchronous Hyperparameter and Neural Architecture Search

Louis Tiao, Aaron Klein, Thibaut Lienart, Cedric Archambeau, Matthias Seeger

https://arxiv.org/abs/2003.10865

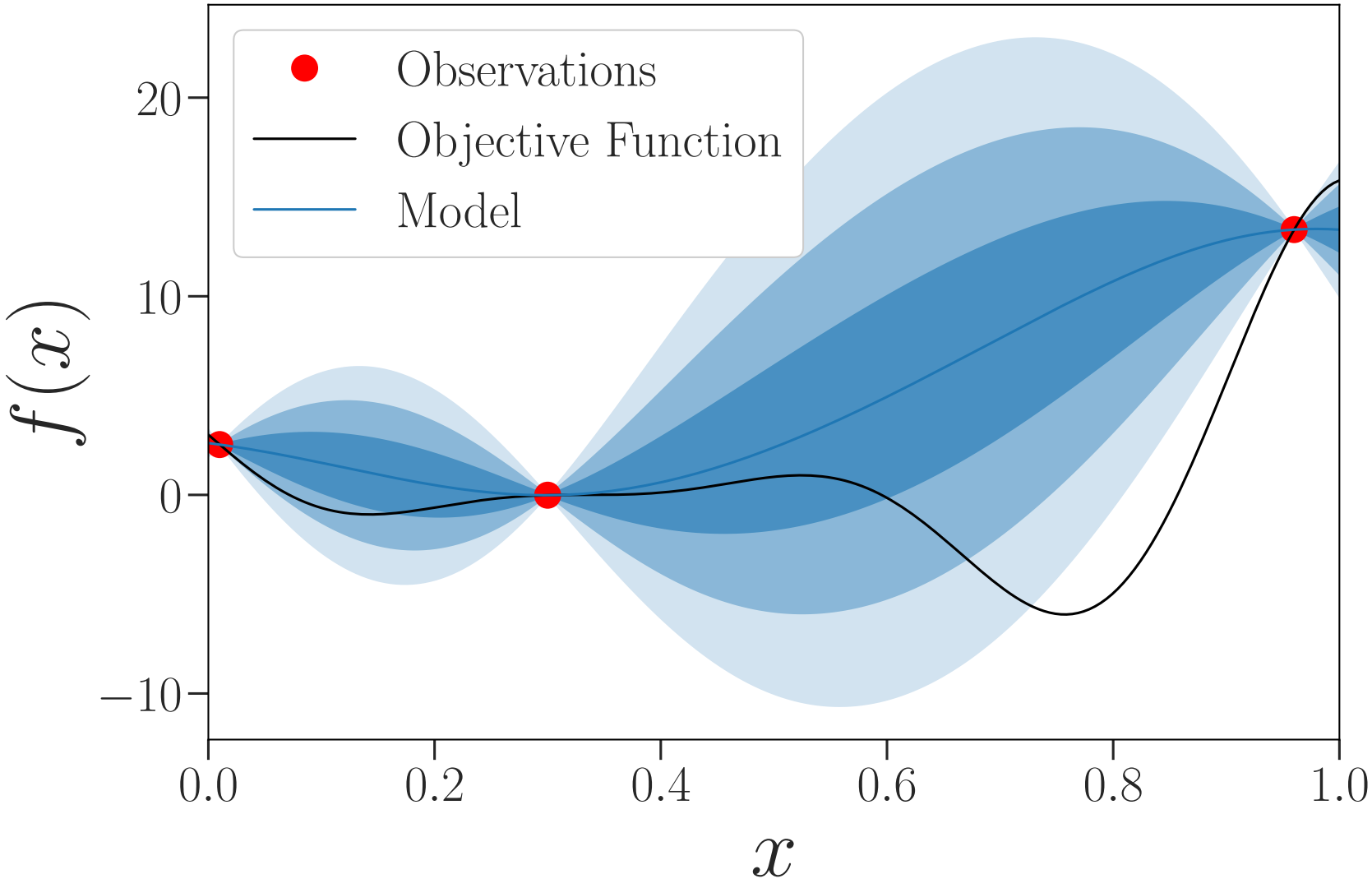
July 9, 2020

Hyperparameter Optimization

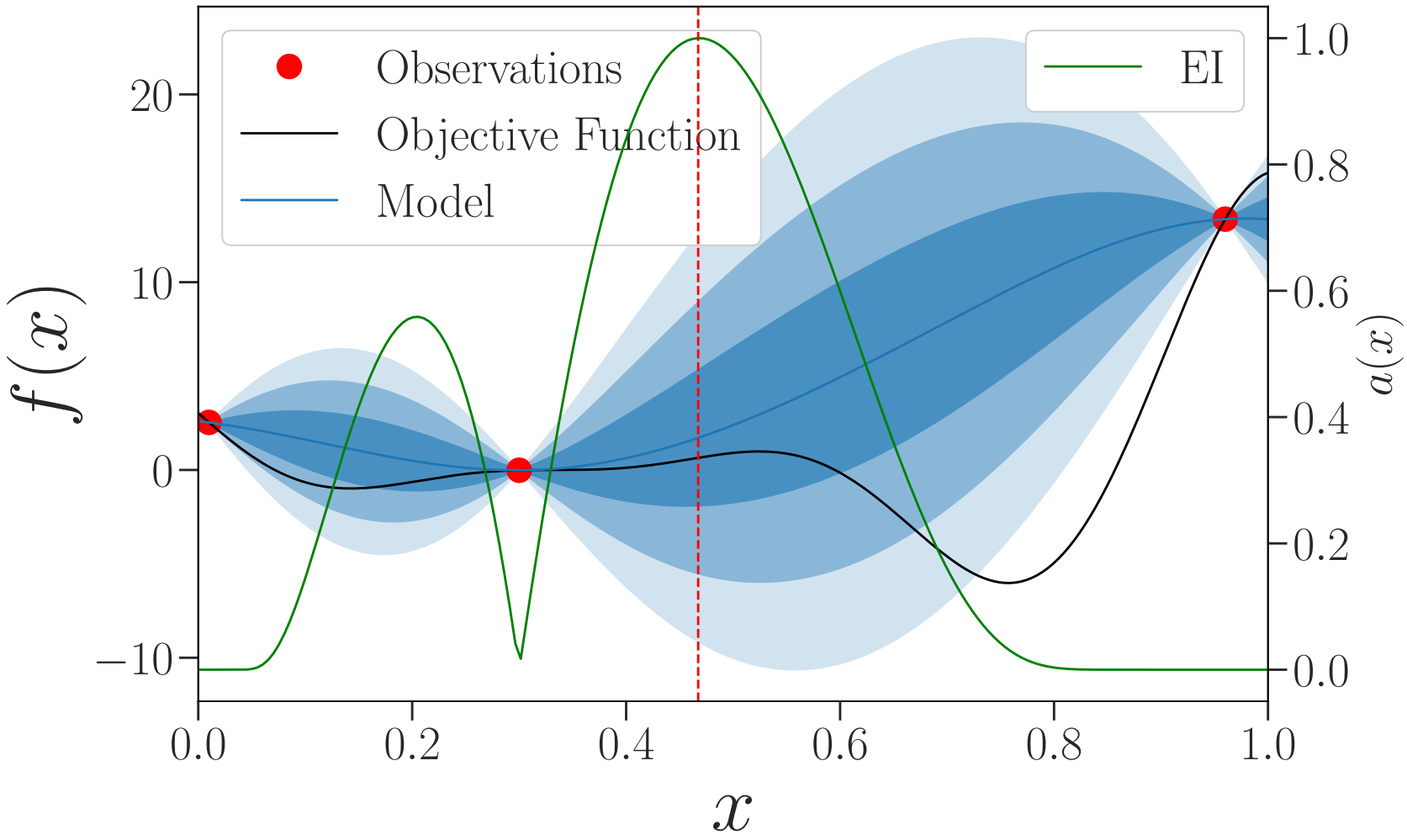
- Competitive ML methods: Many tuning parameters
  - Tuning is essential for good performance
  - Tuning needs expert knowledge and/or lots of time
  - No AutoML without automated tuning
- Trial and error?

  Can do better with **sequential decision making**

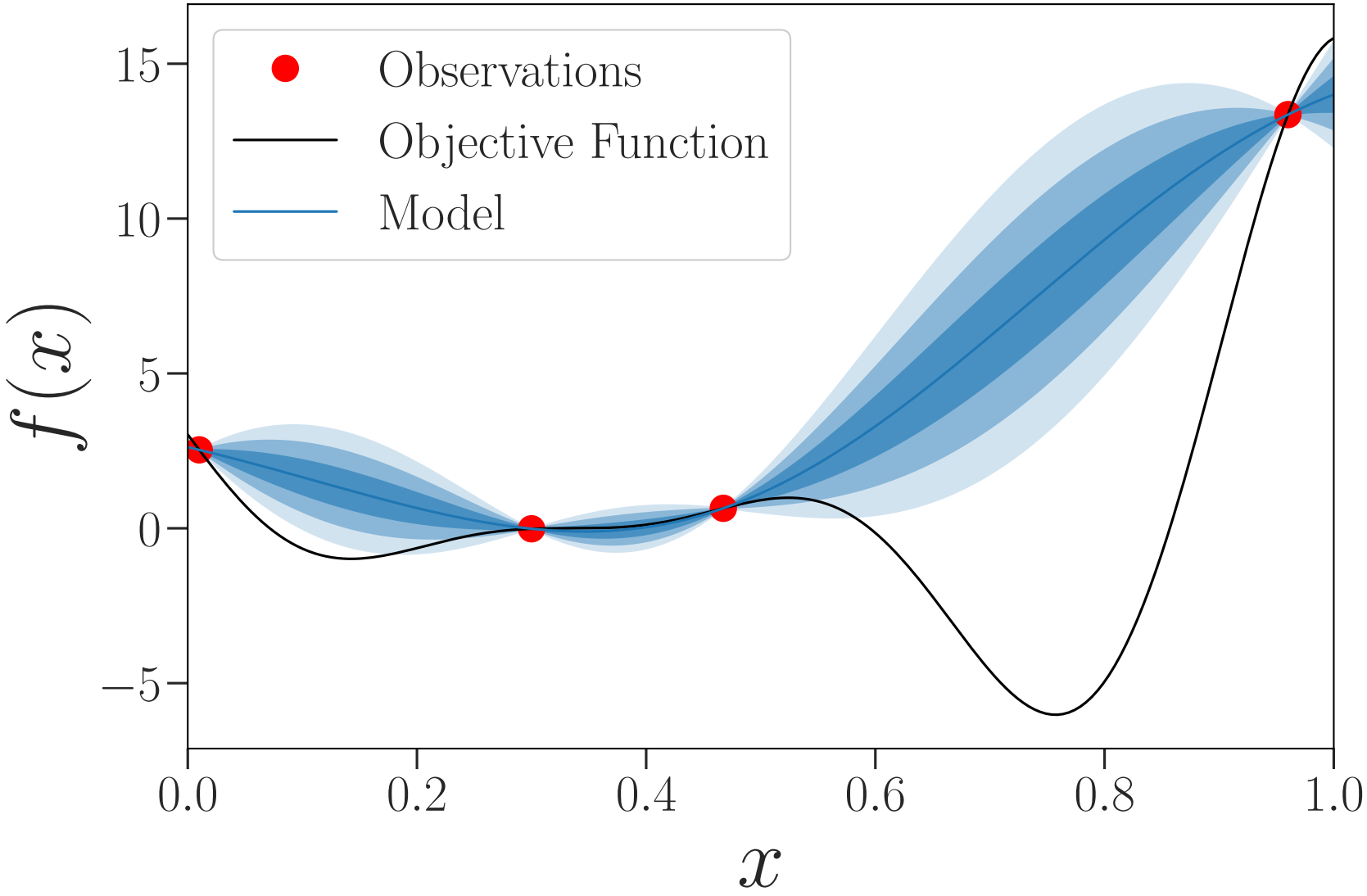
Bayesian Optimization



Bayesian Optimization



Bayesian Optimization



Bayesian Hyperparameter Optimization
can be very slow.
How can we speed it up?

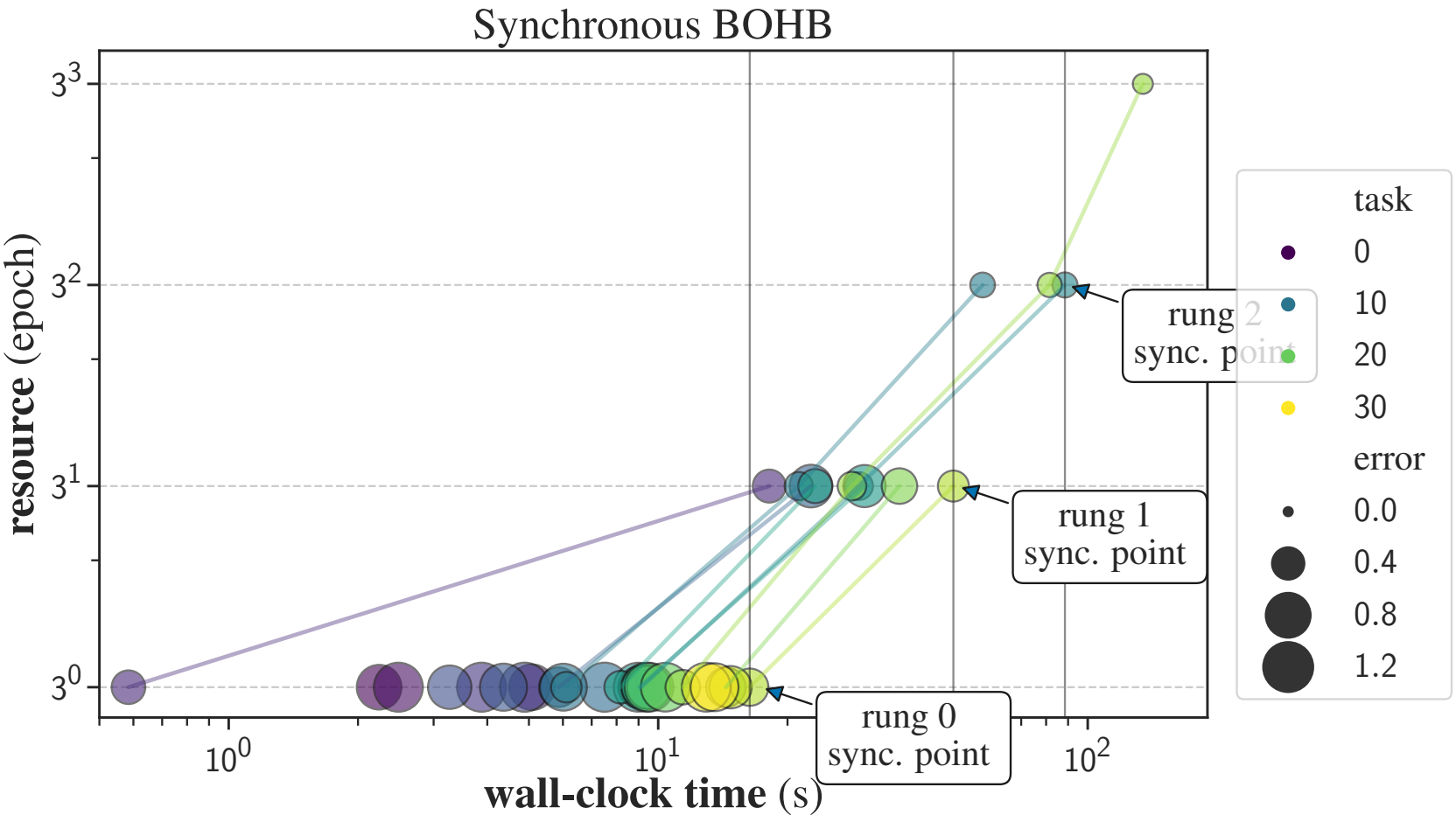
Speed Up 1: Multi-Fidelity

- $f(\boldsymbol{x}) = f(\boldsymbol{x}, r_{\text{max}} = 27)$:
  Validation error after 27 training epochs
  - Signals $f(\boldsymbol{x}, r)$, $r < 27$, for free (better for larger r)
  - Use them for resource allocation

Speed Up 1: Multi-Fidelity

- $f(\boldsymbol{x}) = f(\boldsymbol{x}, r_{\text{max}} = 27)$:
  Validation error after 27 training epochs
  - Signals $f(\boldsymbol{x}, r)$, $r < 27$, for free (better for larger r)
  - Use them for resource allocation
- Multi-fidelity (fidelity $\leftrightarrow$ resource r):
  Start many trials with low resources, then
  - Early stopping of least promising ones
  - Pause and resume (e.g., ASHA)

Successive Halving (Hyperband)



Speed Up 2: Asynchronous Parallel Scheduling

- Parallel evaluations (pool of workers)
  - Synchronous: Batch decisions, wait for all results
  - Asynchronous: Any-time decisions (pending feedback)

Speed Up 2: Asynchronous Parallel Scheduling

- Parallel evaluations (pool of workers)
  - Synchronous: Batch decisions, wait for all results
  - Asynchronous: Any-time decisions (pending feedback)
- **Take away: Asynchronous scheduling very powerful tool for model-based HPO**

Asynchronous Multi-Fidelity Scheduling

Once trial reaches a rung: Competition with other trials

- **Stopping rule**: Compete with earlier trials ≈ Median rule
  - Continue to next rung, or terminate

Asynchronous Multi-Fidelity Scheduling

Once trial reaches a rung: Competition with other trials

- **Stopping rule**: Compete with earlier trials ≈ Median rule
  - Continue to next rung, or terminate
- **Promotion rule**: Compete with earlier/later trials ASHA
  - Trials paused, not terminated
  - May resume later (more competitors)

Speeding Up Hyperparameter Optimization

1. **Multi-fidelity** (fidelity $\leftrightarrow$ resource r):
   Start many trials with low resources, then
   - Early stopping of least promising ones
   - Pause and resume (e.g., ASHA)
2. **Parallel evaluations** (pool of workers)
   - Synchronous: Batch decisions, wait for all results
   - **Asynchronous**: Any-time decisions (pending feedback)

Our Contributions

- BO + Asynchronous HB = **Asynchronous BOHB**:
  Combine asynchronous Hyperband scheduling with Bayesian optimization (joint surrogate model over resource levels)

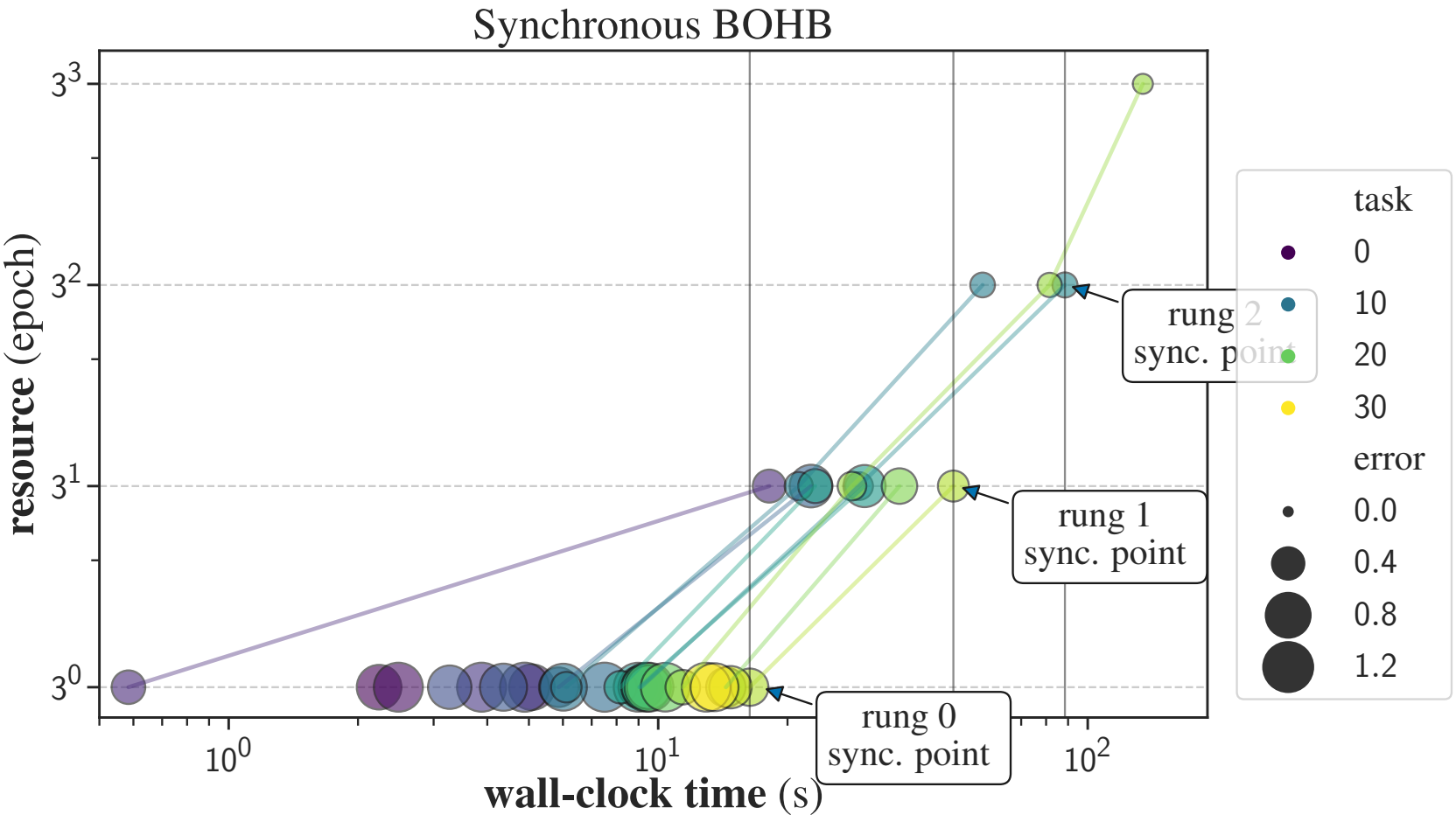
Our Contributions

- BO + Asynchronous HB = **Asynchronous BOHB**:
  Combine asynchronous Hyperband scheduling with Bayesian optimization (joint surrogate model over resource levels)
- Distributed HPO across multiple instances:
  Open-sourced in `AutoGluon` autogluon.mxnet.io

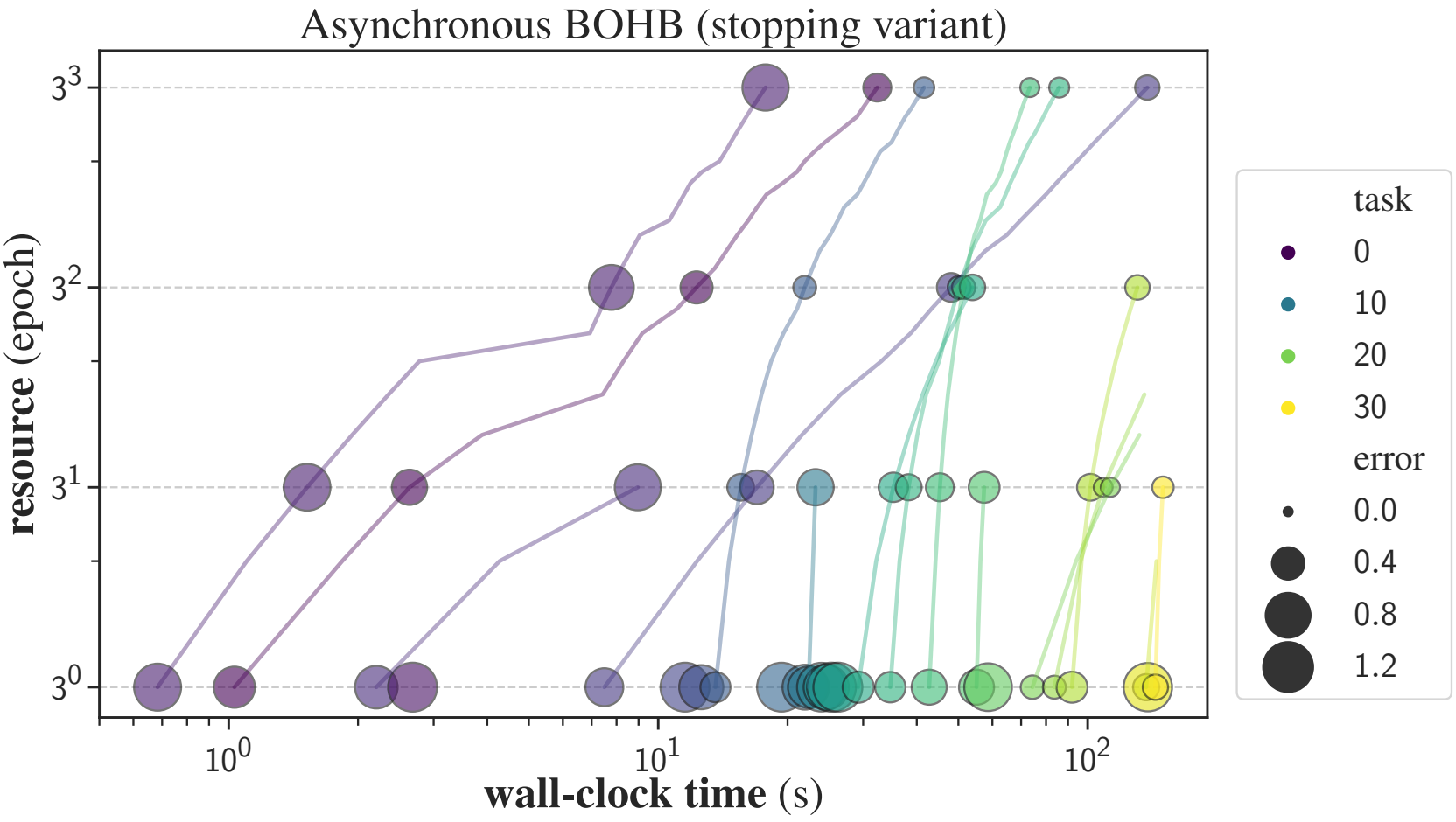
Our Contributions

- BO + Asynchronous HB = **Asynchronous BOHB**:
  Combine asynchronous Hyperband scheduling with Bayesian optimization (joint surrogate model over resource levels)
- Distributed HPO across multiple instances:
  Open-sourced in `AutoGluon` autogluon.mxnet.io
- In most expensive experiments:
  Similar performance after same wall-clock time as best baseline, using **half the computational resources**

Synchronous BOHB



Stopping-Based Asynchronous BOHB

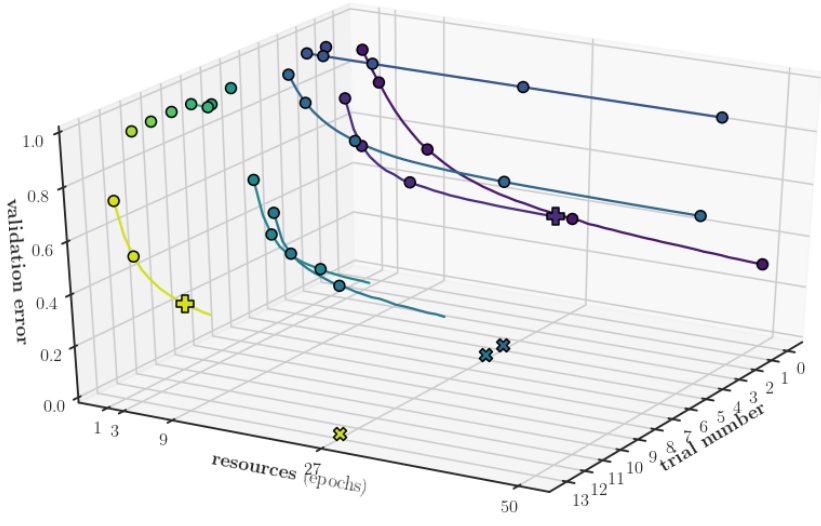


How To Select New Candidate?

We use a **joint GP model** to fit performances of hyperparameter configurations across resource levels

Legend

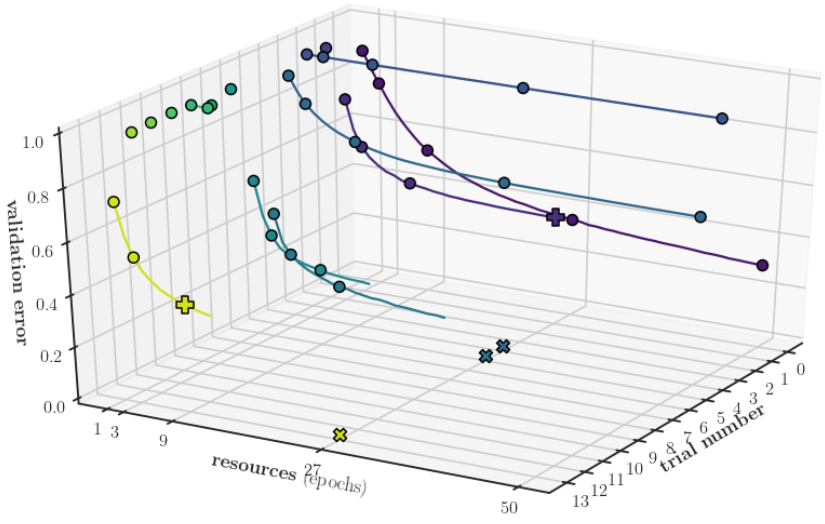
- **Circle**: observed
- **Plus**: newly observed
- **Cross**: pending evaluations



Update Surrogate Model

Step 1: Update GP with new targets (**Circle** + **Plus**) Legend:

- **Circle**: observed
- **Plus**: newly observed
- **Cross**: pending evaluations

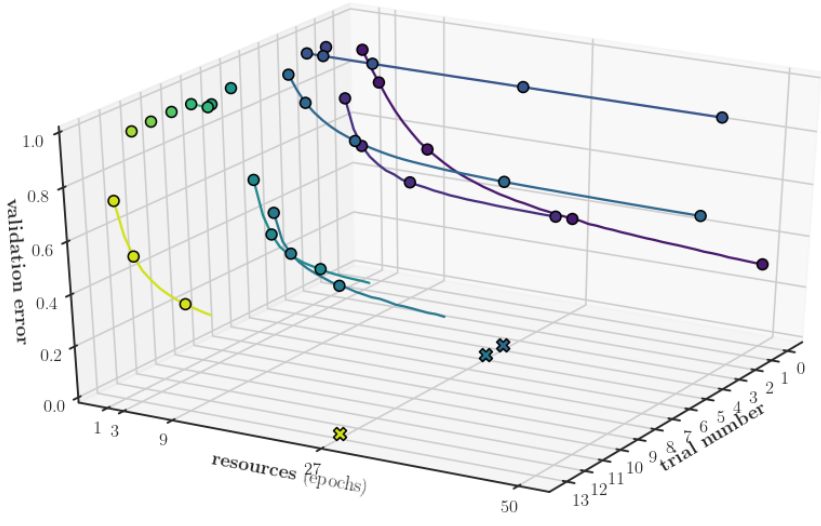


Draw Fantasy Values for Pending Evaluations

Step 2: Draw new fantasy values for all pending (**Cross** → **Diamond**)

Legend:

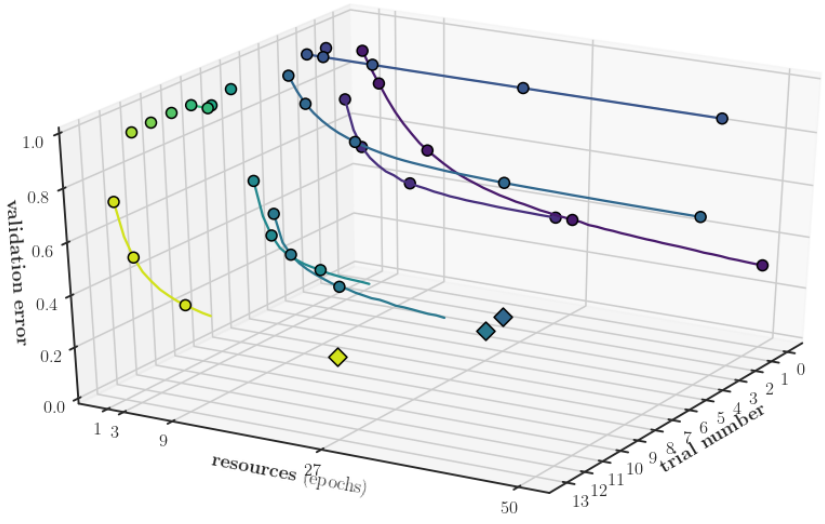
- **Circle**: observed
- **Plus**: newly observed
- **Cross**: pending evaluations



Optimize Acquisition Function

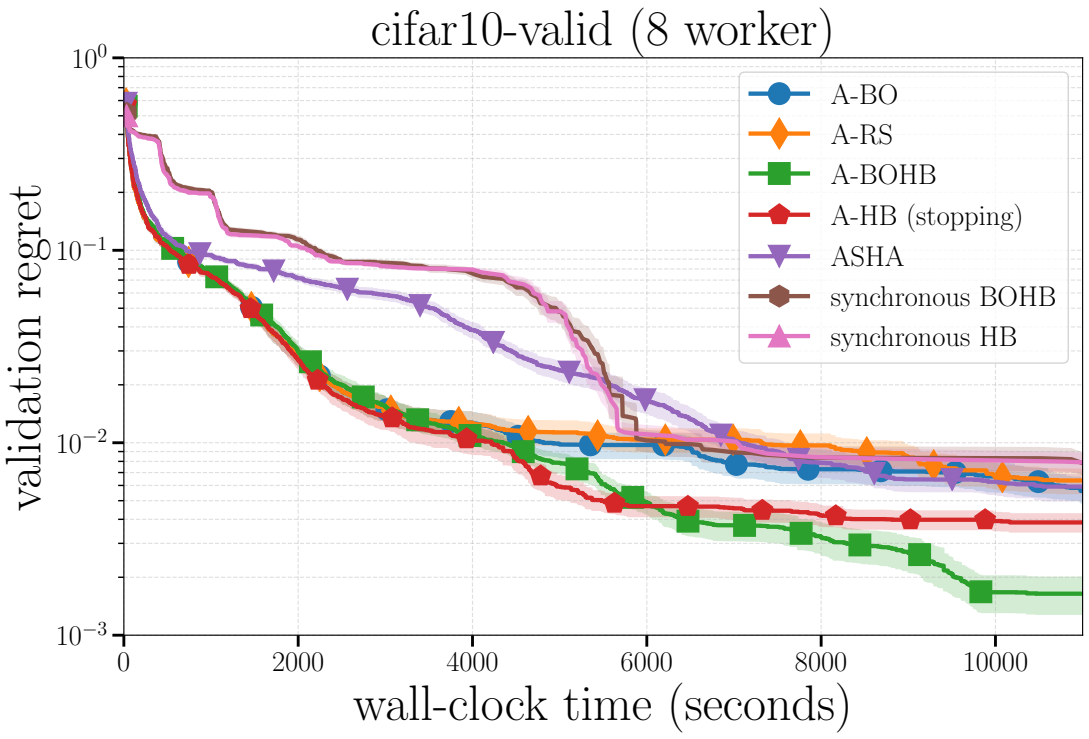
Step 3: Optimize EI on highest rung, averaging over fantasies (**Circle** + **Diamond**) Legend:

- **Circle**: observed
- **Plus**: newly observed
- **Diamond**: fantasy values



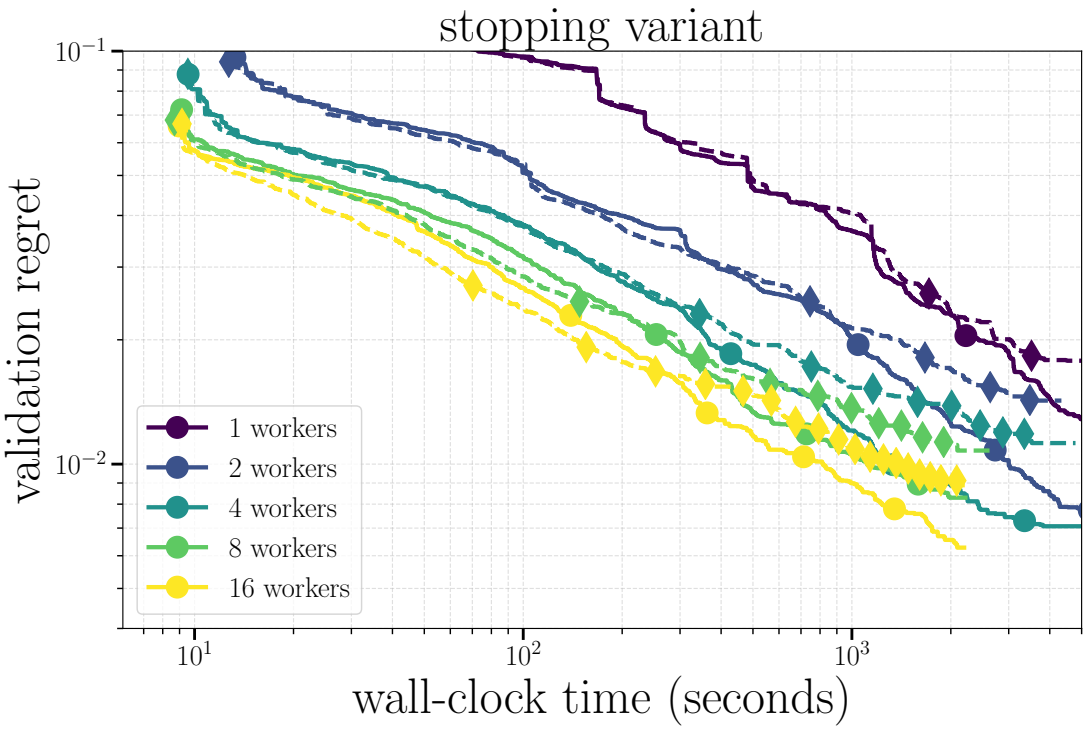
NASBench201 (CIFAR-10), 8 Workers

- Synchronous methods take a lot longer
- ASHA (pause and resume) initially too conservative
- Asynchronous stopping: BOHB outperforms HB eventually



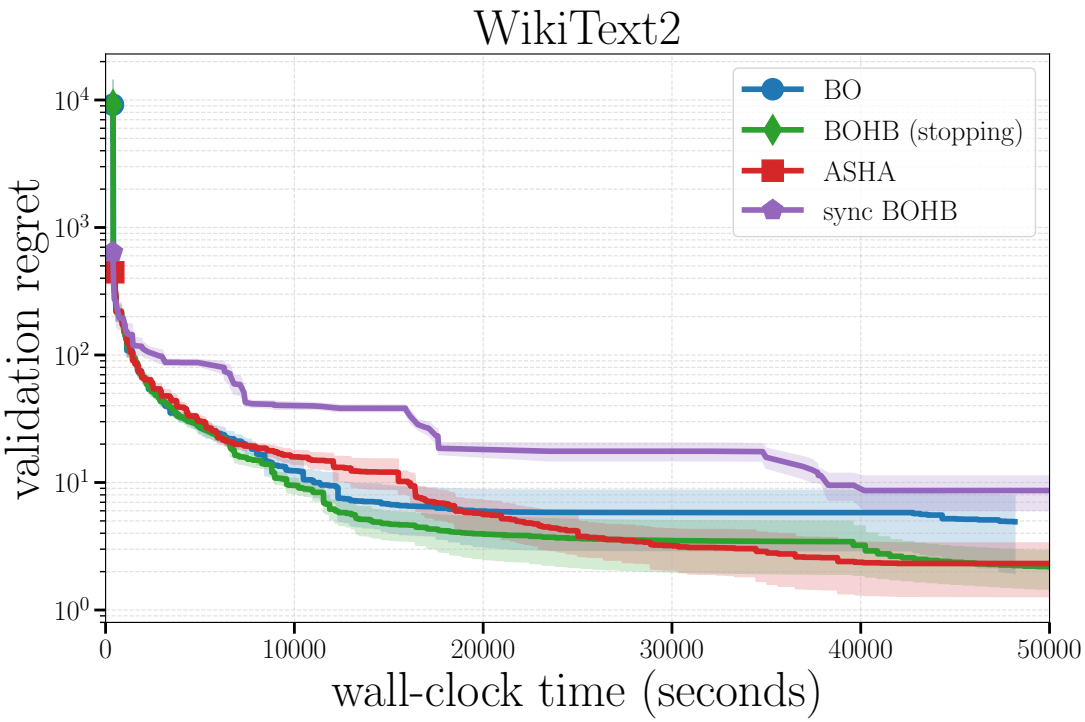
Asynchronous MLP Tuning: Scaling with Workers

- OpenML dataset "electricity"
- Speedup as expected for asynchronous HB (random search)
- Break-away point of BOHB (vs HB) earlier with more workers



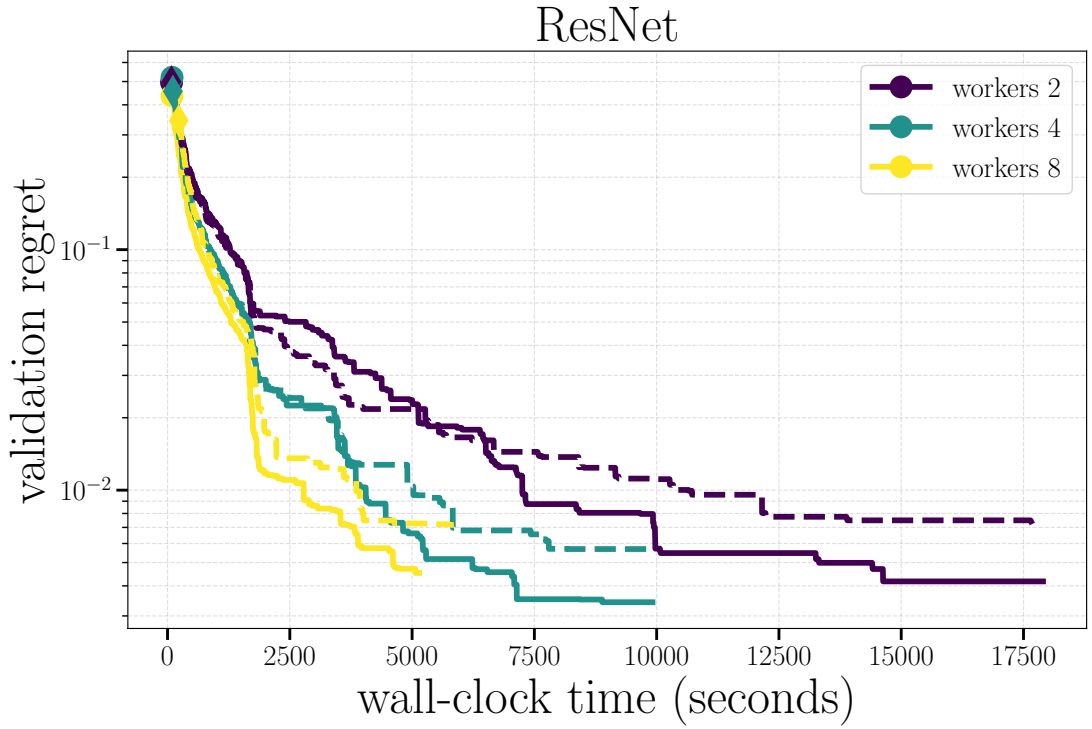
Language Modeling with LSTM, 8 Workers

- WikiText-2 dataset
- Synchronous BOHB not competitive
- ASHA (pause and resume) catches up with asynchronous BOHB (stopping) after 7 hours

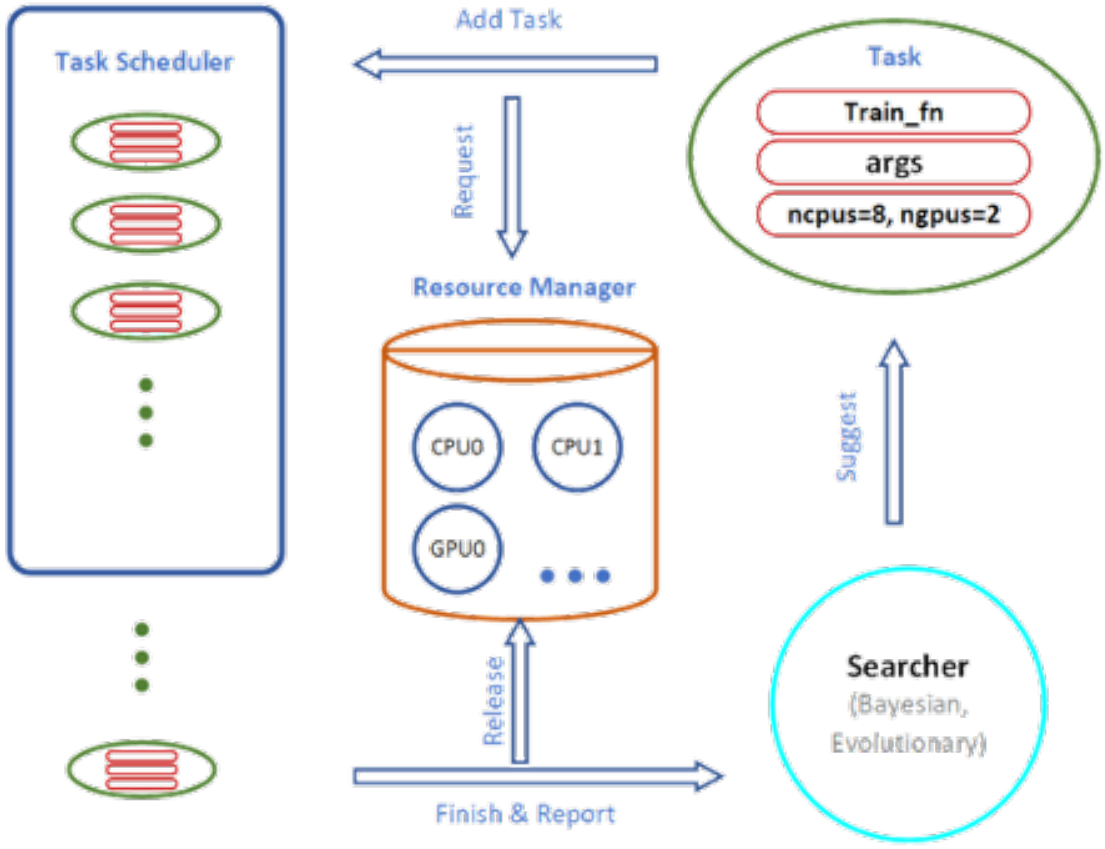


Tuning ResNet on CIFAR-10

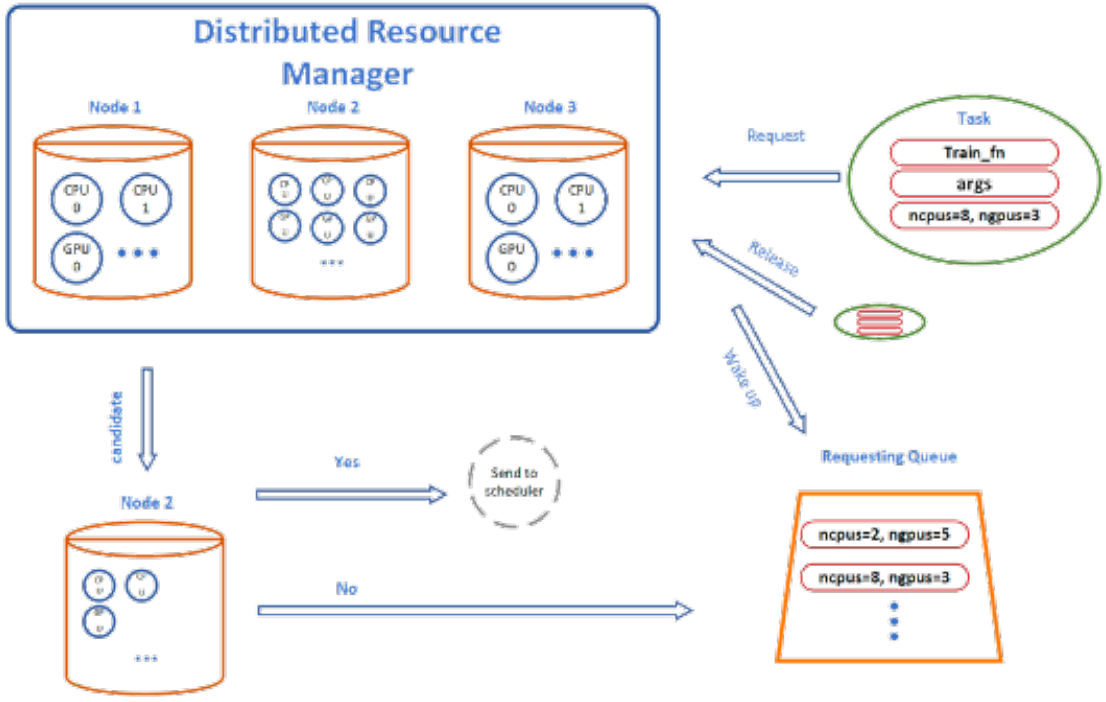
- Comparing asynchronous stopping: BOHB (solid) and HB (dashed)
- BOHB achieves competitive regret 10^{-2} at roughly same time as HB, with half the number of workers



- **Searcher**: Which trial to run next?
- **Scheduler**: What computational resource for each trial, and when?



- Modular. Easy to use and extend
- Seemless experience on multiple instances
- Automatic resource management and scheduling on remote machines



Demo Time!

Joint Gaussian Process Surrogate Model

$$f(\boldsymbol{x}, r) = \gamma e^{-\lambda r} + f_0(\boldsymbol{x}) \left(1 - \delta e^{-\lambda r}\right), \quad \lambda \sim \Gamma(\alpha, \beta),\ \delta \in [0, 1]$$

- Exponential decay model
  - $f_0(\boldsymbol{x}) \sim \mathrm{GP}(k_{\mathrm{matern}_{5/2}})$: Value for $r \to \infty$
    $\gamma + f_0(\boldsymbol{x})(1 - \delta)$: Value for $r \to 0$
  - $\delta = 0$: Additive Freeze-Thaw model

Swersky et.al., 2014

Joint Gaussian Process Surrogate Model

$$f(\boldsymbol{x}, r) = \gamma e^{-\lambda r} + f_0(\boldsymbol{x})\left(1 - \delta e^{-\lambda r}\right), \quad \lambda \sim \Gamma(\alpha, \beta),\ \delta \in [0,1]$$

- Exponential decay model
  - $f_0(\boldsymbol{x}) \sim \mathrm{GP}(k_{\mathrm{matern}_{5/2}})$: Value for $r \to \infty$
    $\gamma + f_0(\boldsymbol{x})(1-\delta)$: Value for $r \to 0$
  - $\delta = 0$: Additive Freeze-Thaw model Swersky et.al., 2014
- Baseline: Matérn $\frac{5}{2}$ ARD on $[\boldsymbol{x}^T, r]^T$, w/o warping of r
- Our results so far:
  Choice of surrogate model of minor importance