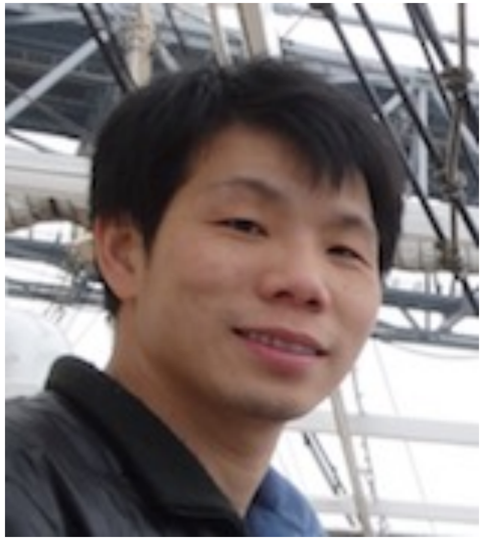
Reproducible and Efficient Benchmarks for Hyperparameter Optimization of Neural Machine Translation Systems

@ TACL 2020



Xuan Zhang

Kevin Duh

Johns Hopkins Whiting School of Engineering

Department of Computer Science,
Johns Hopkins University

Reproducible and Efficient Benchmarks for Hyperparameter Optimization of Neural Machine Translation Systems

Outline

1. Motivation

2. Introduction to Hyperparameter Optimization (HPO)

3. Contributions
   - a new HPO benchmark dataset (tabular dataset)
   - a new HPO algorithm (graph-based semi-supervised learning)

4. Summary

1. Motivation

Hyperparameter Search of NMT systems

Hyperparameters:

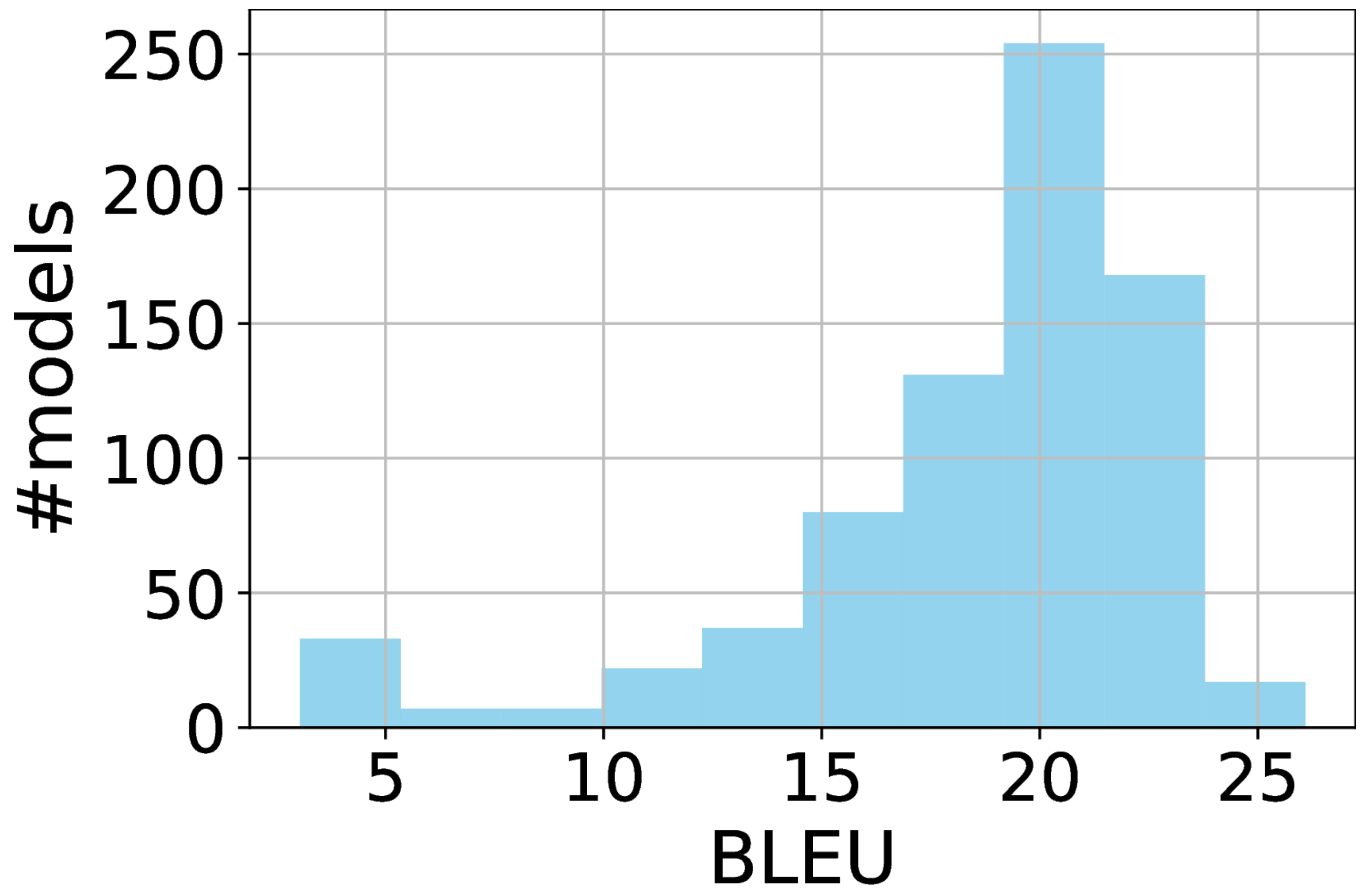
- preprocessing configurations: number of BPE symbols
- training settings: initial learning rate, warmup
- architecture designs: number of layers, embedding size,
  number of hidden units in each layer,
  number of self-attention heads

Objectives:

- training accuracy: BLEU, perplexity
- computational cost: decoding time, number of model parameters

Hyperparameter Search of NMT systems

--- Rewarded and Necessary



Challenges of HPO on NMT

- **Large search space & high computational costs for NMT training**

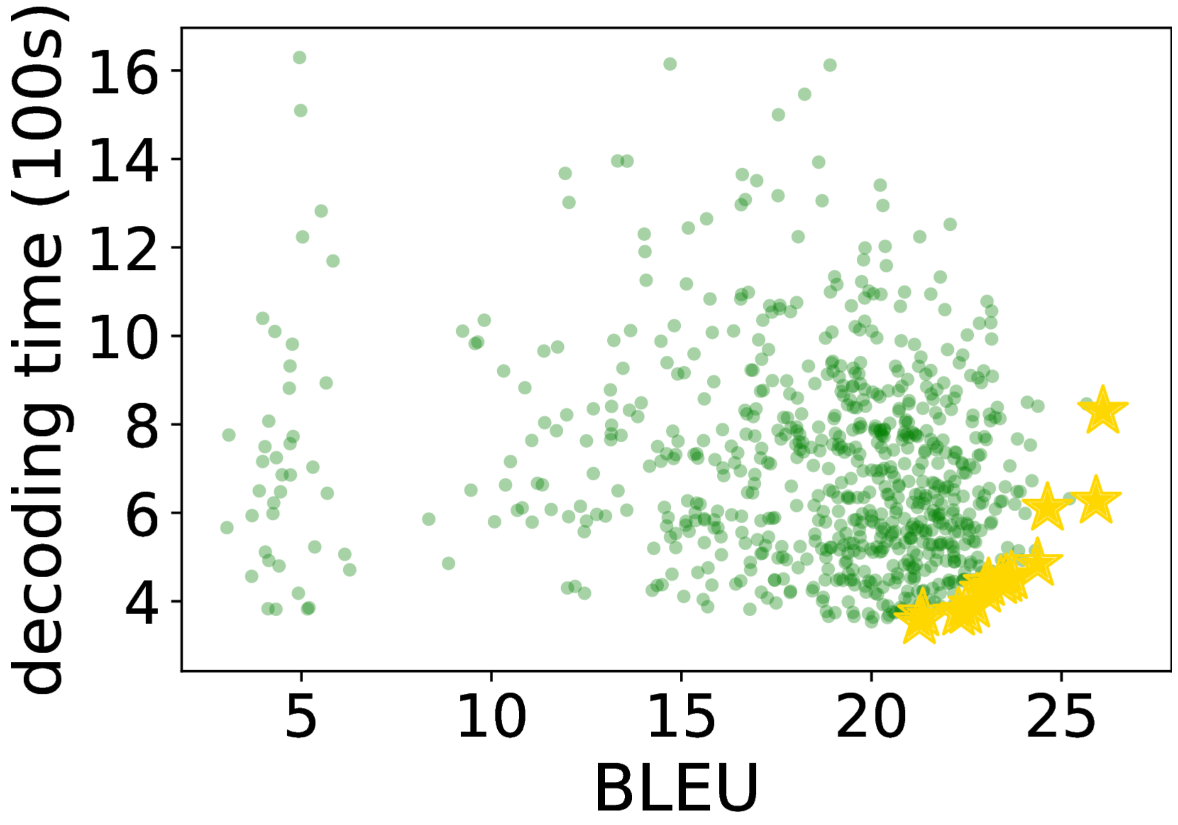
If we have 6 hyperparameters to tune, where we want to try 3 candidate values for each hyperparameter, and it takes 1 day to 1 week to train a model, then how long will it take for a grid search?

3^6 = 729 (days / weeks)

HPO is expensive to run!

Challenges of HPO on NMT

- Large search space & high computational costs for NMT training
- **Difficult to optimize multiple objectives**



Pareto-optimal system
(There does not exist a system that outperforms it on both objectives.)

Challenges of HPO on NMT

- Large search space & high computational costs for NMT training
- Difficult to optimize multiple objectives

HPO on NMT has been hardly studied.

It is prohibitively expensive to **compare** different HPO methods on NMT tasks in practice.

(This work) **HPO Benchmark Dataset on NMT**

Goal: enable **reproducible** HPO research on NMT tasks

Table-lookup benchmark procedure:

1. train an extremely large number of NMT systems with diverse hyperparameter settings and record their performance.

-> a table of (configuration, performance) pairs

2. constrain HPO methods to sample from this finite set of models.

2. Intro to HPO

HPO Problem Definition

Let

- λ be the hyperparameters of a ML algorithm with domain Λ ,
- $\mathbf{L}(\lambda, D_{train}, D_{valid})$ denote the loss of the ML algorithm, using hyperparameters λ trained on D_{train} and evaluated on D_{valid} .

The HPO problem is to find a configuration λ^* that minimizes this loss:

$$\lambda^* \in argmin_{\lambda \in \Lambda} \mathbf{L}(\lambda, D_{train}, D_{valid})$$

HPO Methods

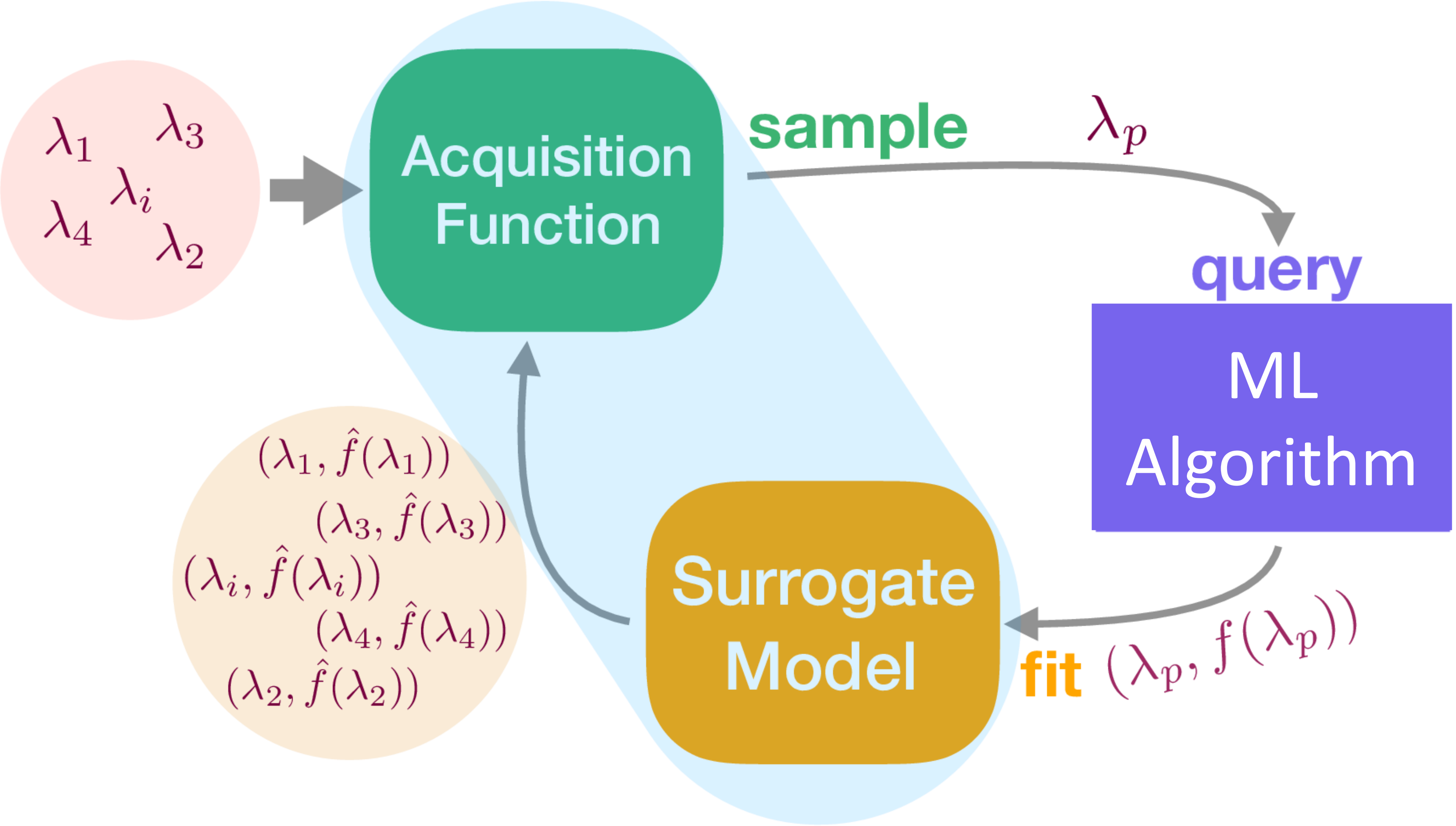
Model-Free Optimization Methods

- **Grid Search**
- **Random Search**
- **Population-based methods**
  e.g. genetic algorithms, evolutionary algorithms --- CMA-ES

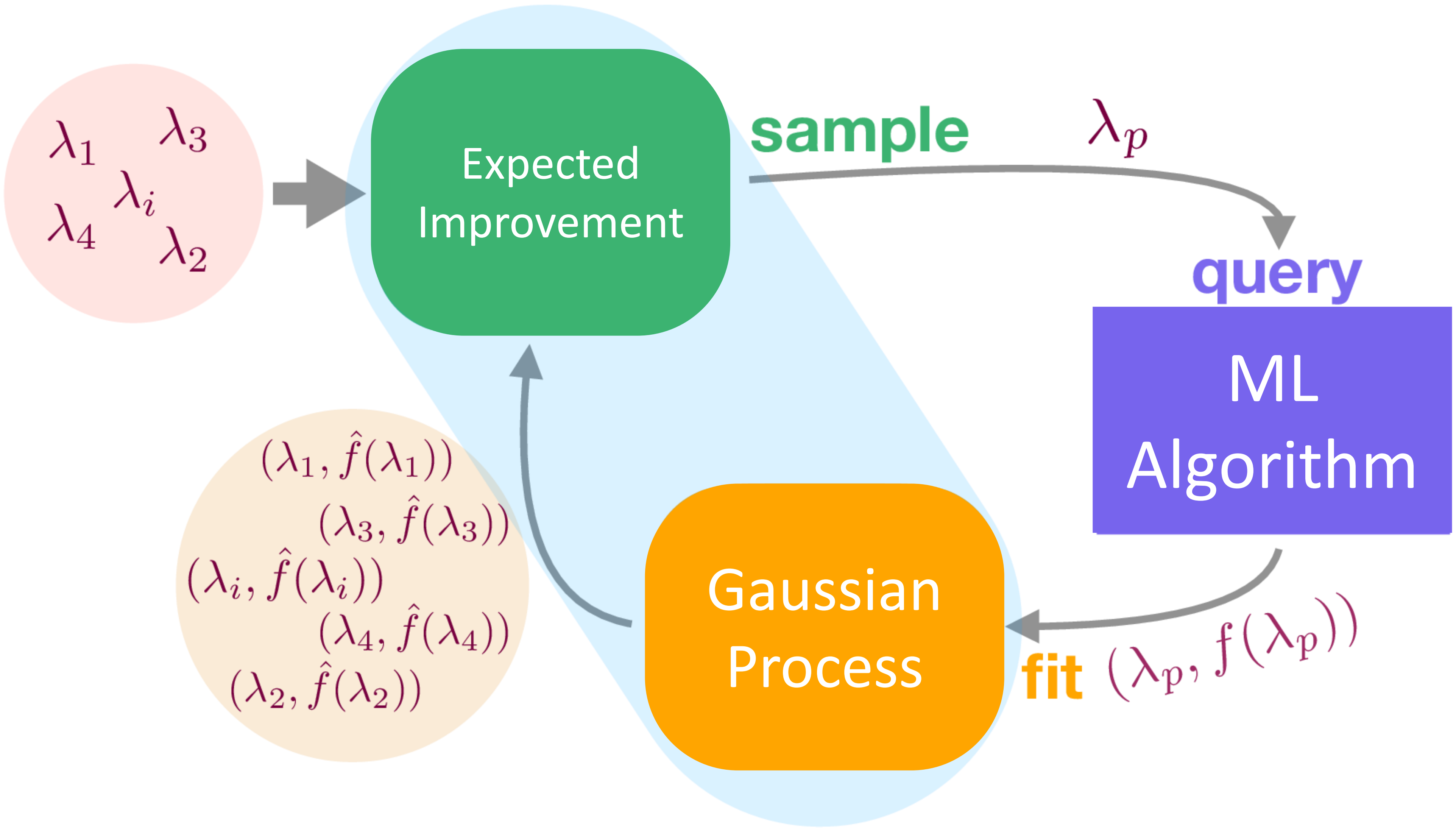
Sequential Model-Based Optimization Methods (SMBO)

- **Bayesian Optimization (BO)**
- **Tree Parzen Estimator (TPE)**

Sequential Model-Based Optimization (SMBO)



Bayesian Optimization



3. Contributions

- **a new HPO benchmark dataset**
  (tabular dataset)
- a new HPO algorithm
  (graph-based semi-supervised learning)

HPO Method Selection

One pitfall in the evaluation of HPO methods:

The ranking between HPO methods varies between tasks.

(Klein et al., 2019)

Solution:

Select HPO method based on its performance on various MT corpora.

HPO Method Selection

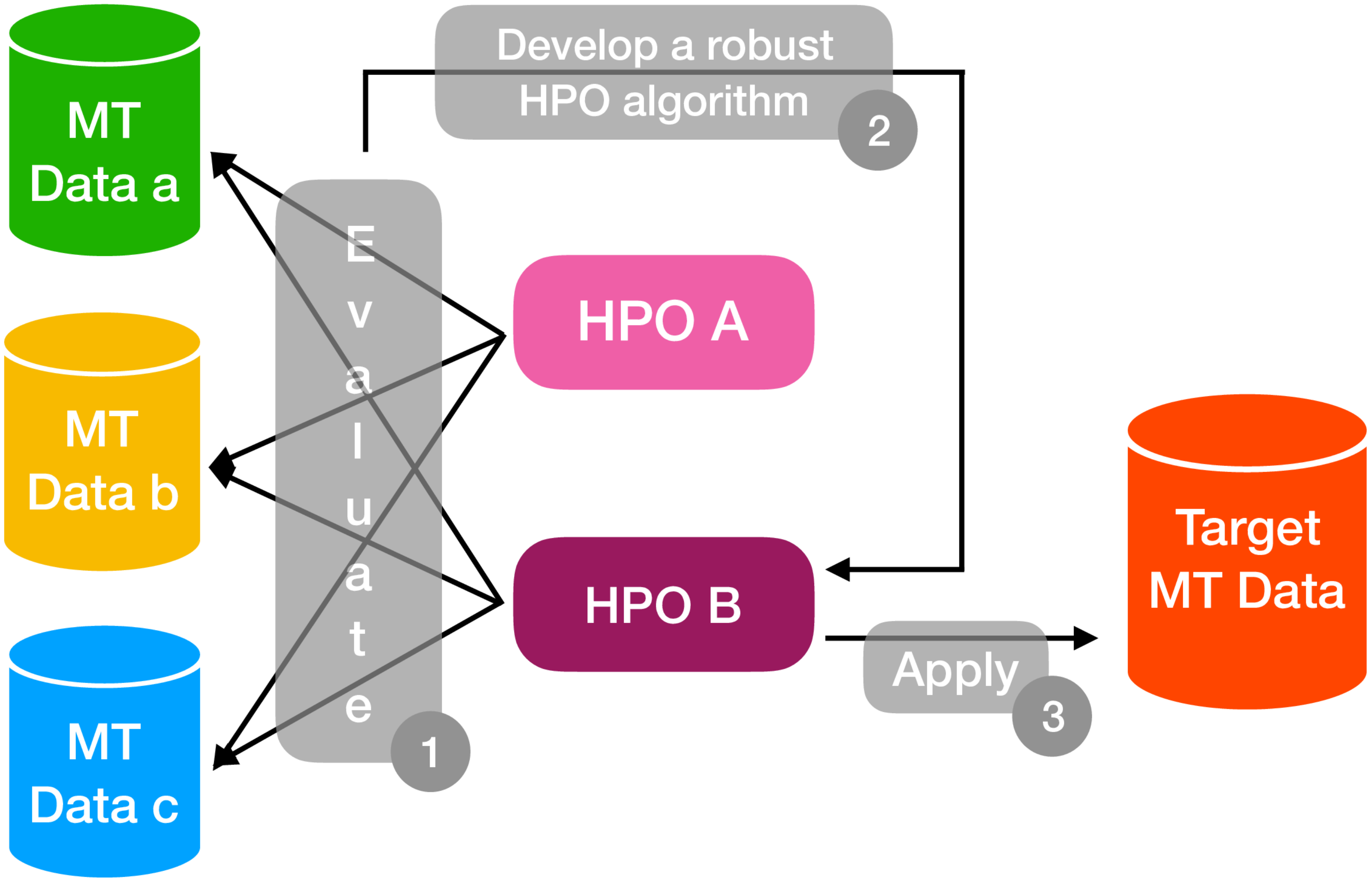


Table-Lookup HPO Datasets

- **6 MT Corpora:**
  - large resource (WMT2019 Robustness): ja-en, en-ja (4M lines)
  - mid resource (TED Talks): zh-en, ru-en (170k lines)
  - low resource: sw-en, so-en (24k lines)
- **Model:** Transformers

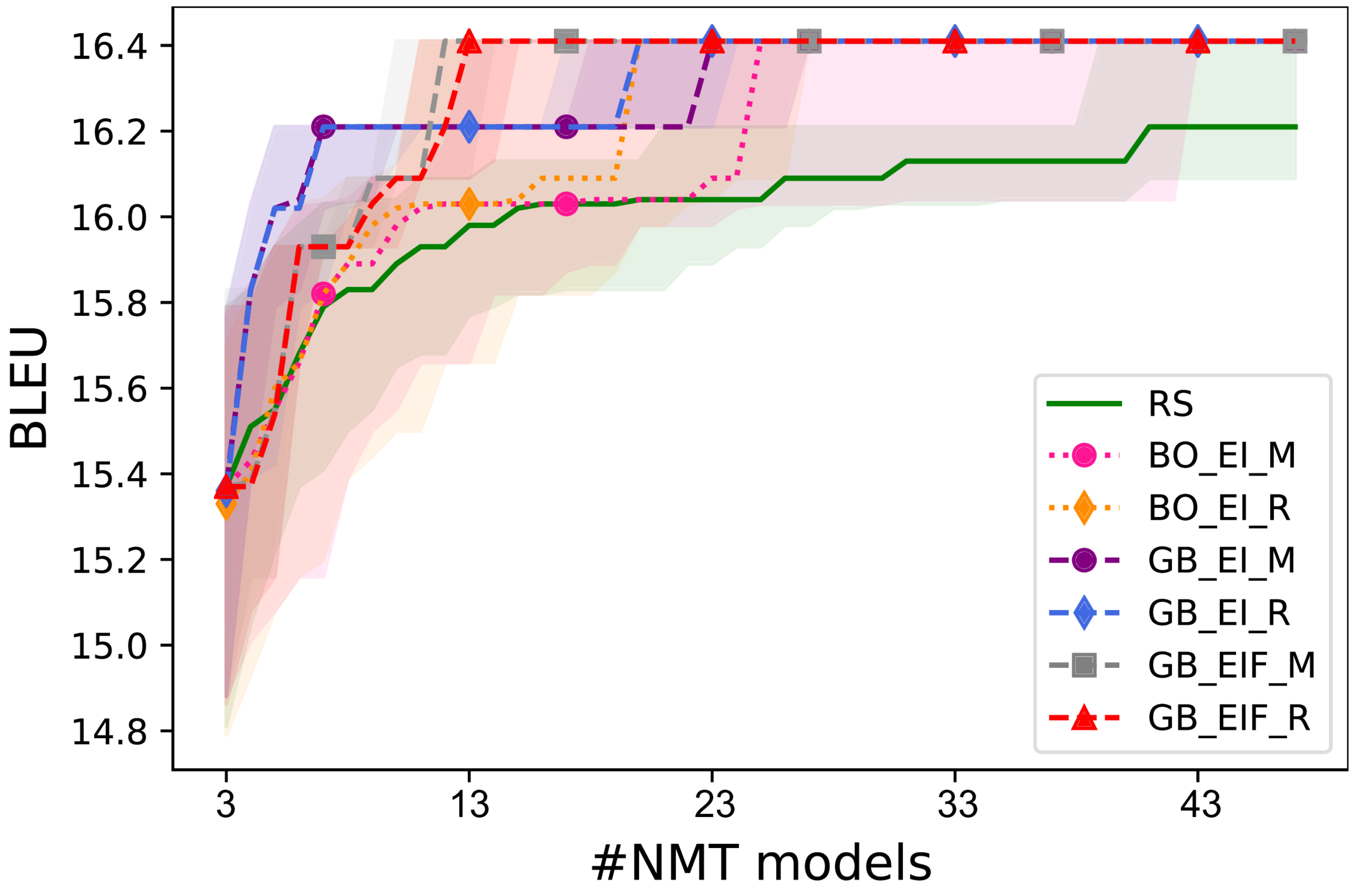
Table-Lookup HPO Datasets

- **6 MT Corpora:**
  large resource (WMT2019 Robustness): ja-en, en-ja (4M lines)
  mid resource (TED Talks): zh-en, ru-en (170k lines)
  low resource: sw-en, so-en (24k lines)
- **Model:** Transformers
- **Search Space:** 2245 Transformers (1547 GPU days)

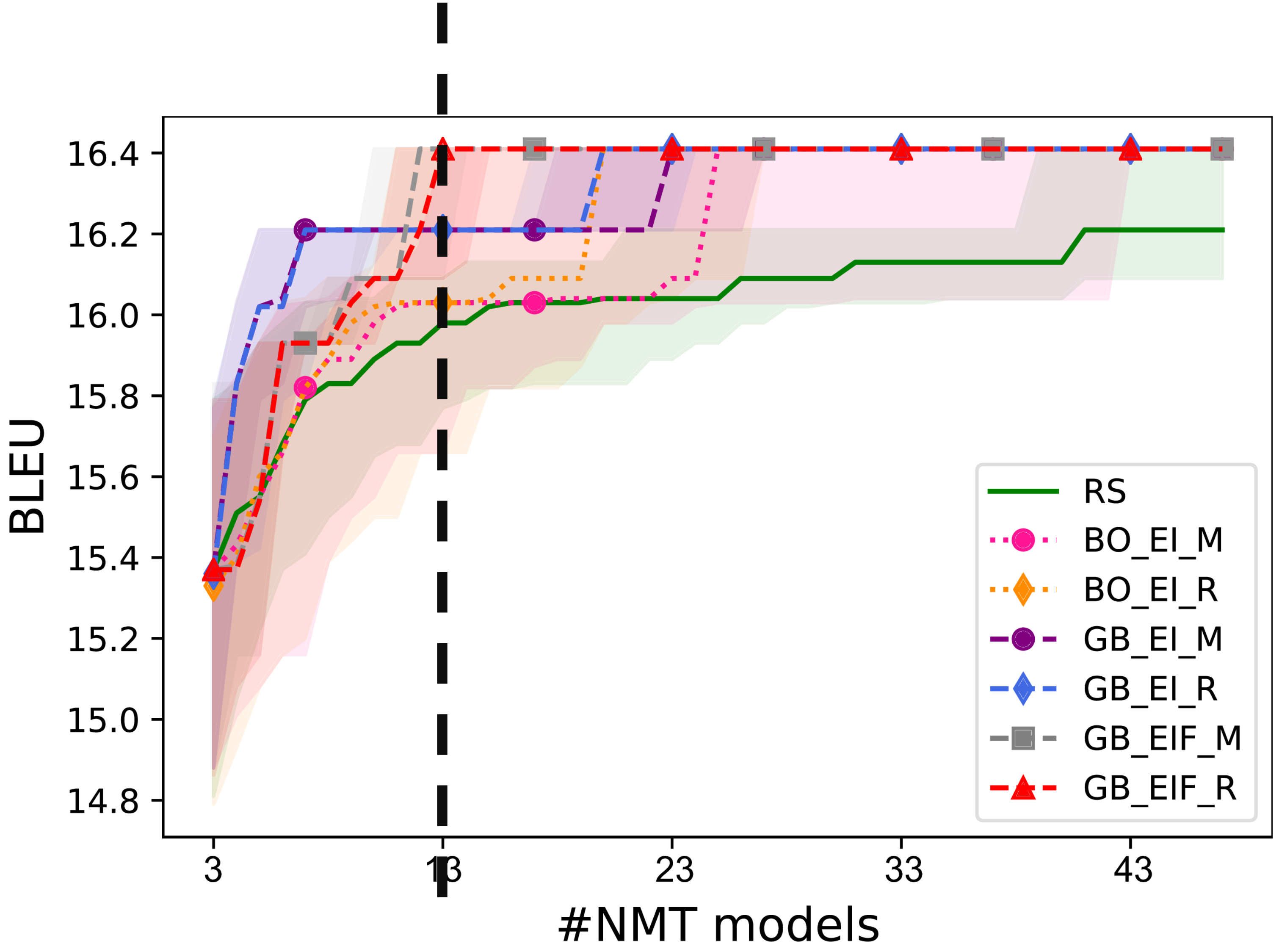
dataset	**bpe** (1k)	**#layers**	**#embed**	**#hidden**	**#att_heads**	**init_lr** (10^{-4})
zh, ru, ja, en	10, 30, 50	2, 4	256, 512, 1024	1024, 2048	8, 16	3, 6, 10
sw	1, 2, 4, 8, 16, 32	1, 2, 4, 6	256, 512, 1024	1024, 2048	8, 16	3, 6, 10
so	1, 2, 4, 8, 16, 32	1, 2, 4	256, 512, 1024	1024, 2048	8, 16	3, 6, 10

- **Objectives:** BLEU & perplexity; decoding time, #updates, GPU memory, #model parameters

Application I. HPO Method Comparison

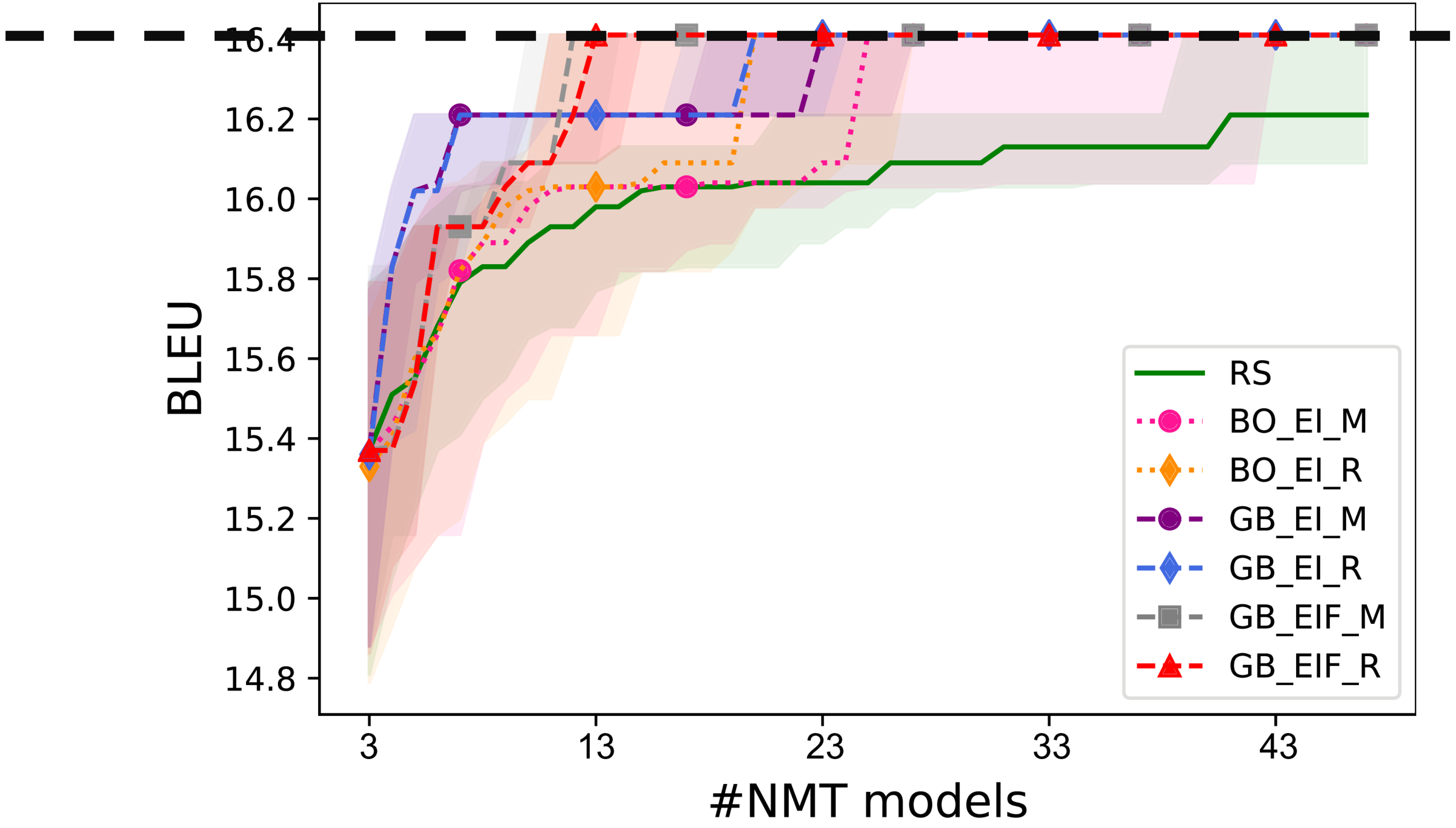


Application I. HPO Method Comparison



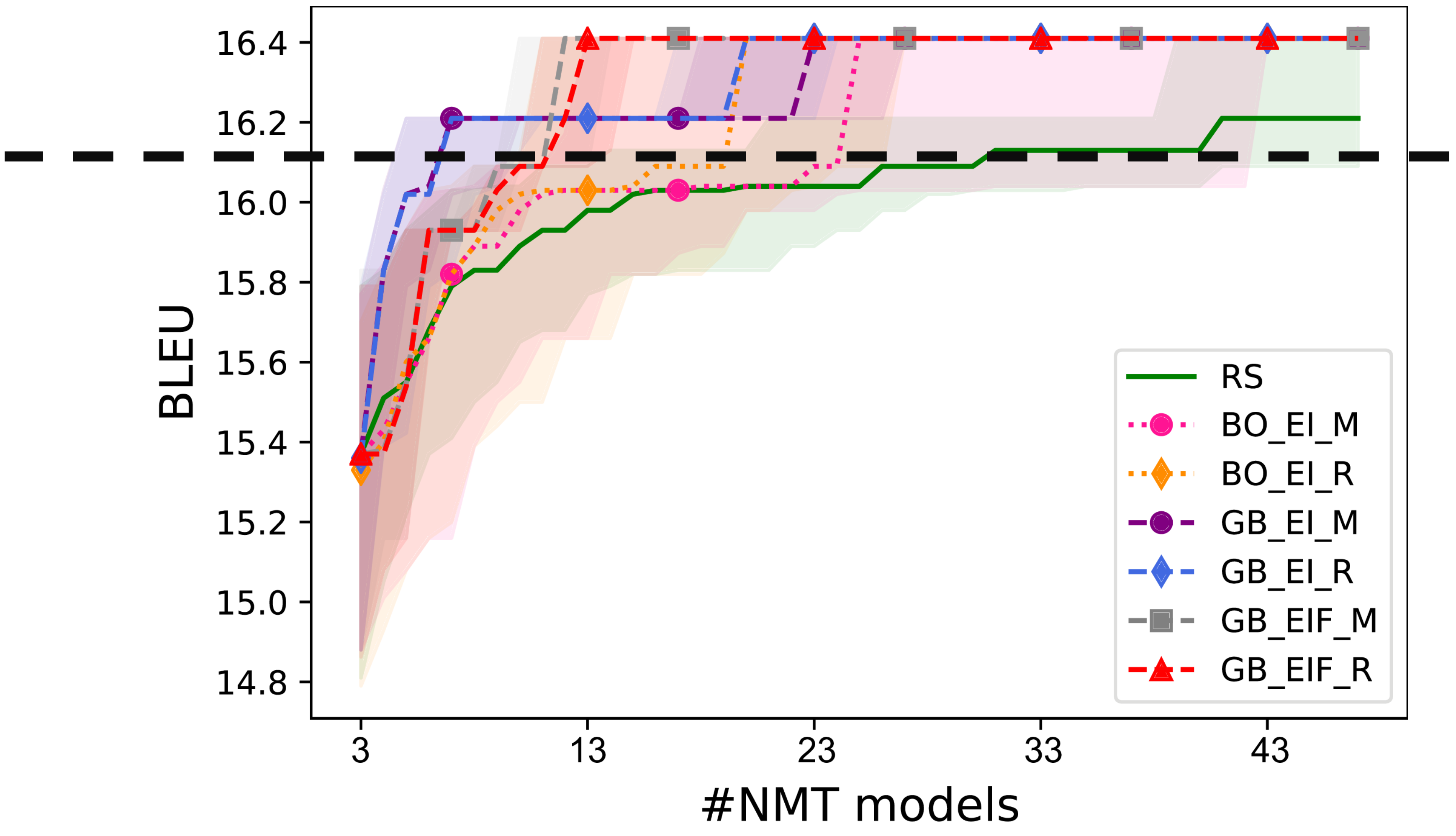
fix the budget

Application I. HPO Method Comparison



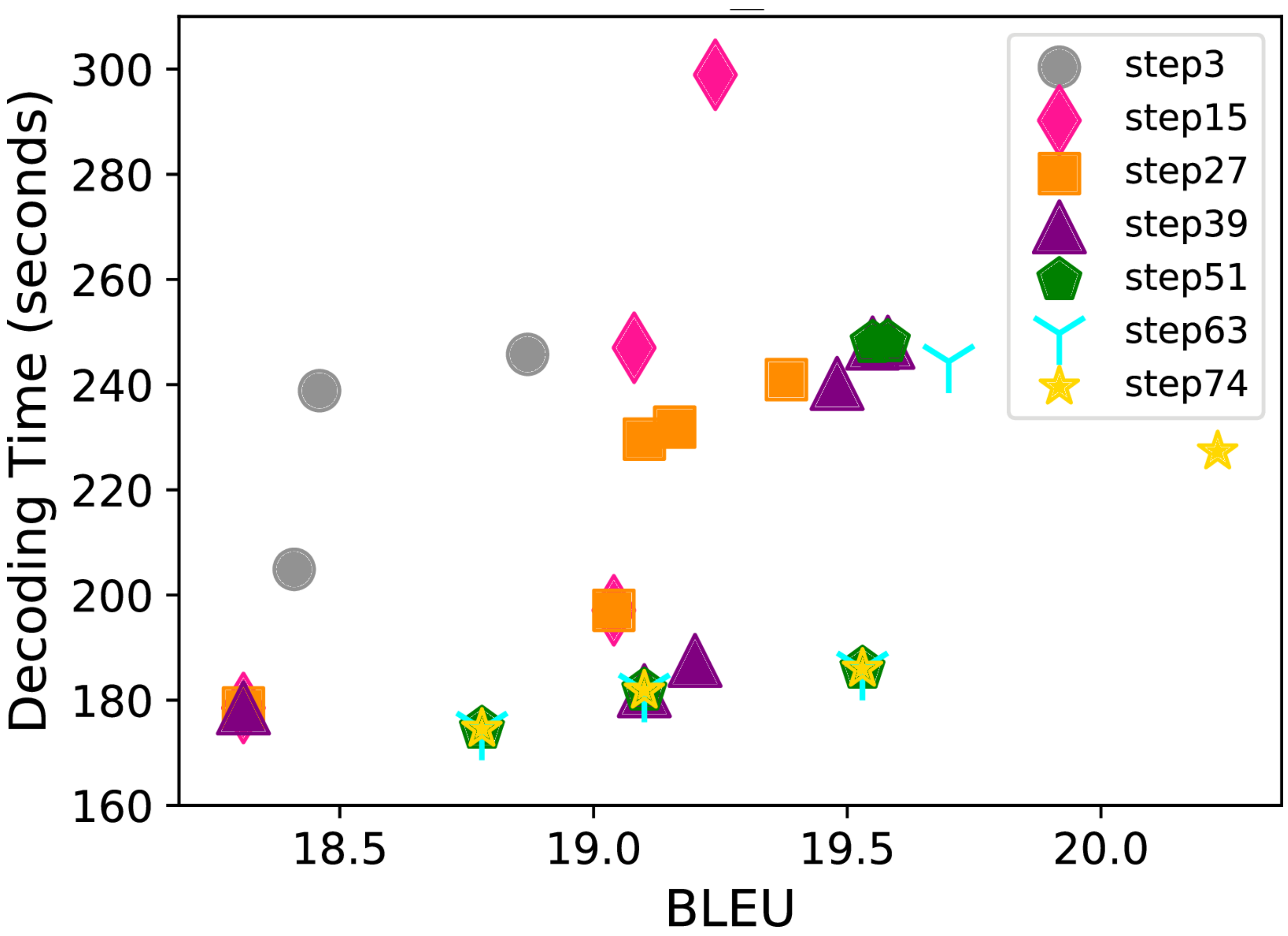
fix the target performance

Application I. HPO Method Comparison



fix the target performance

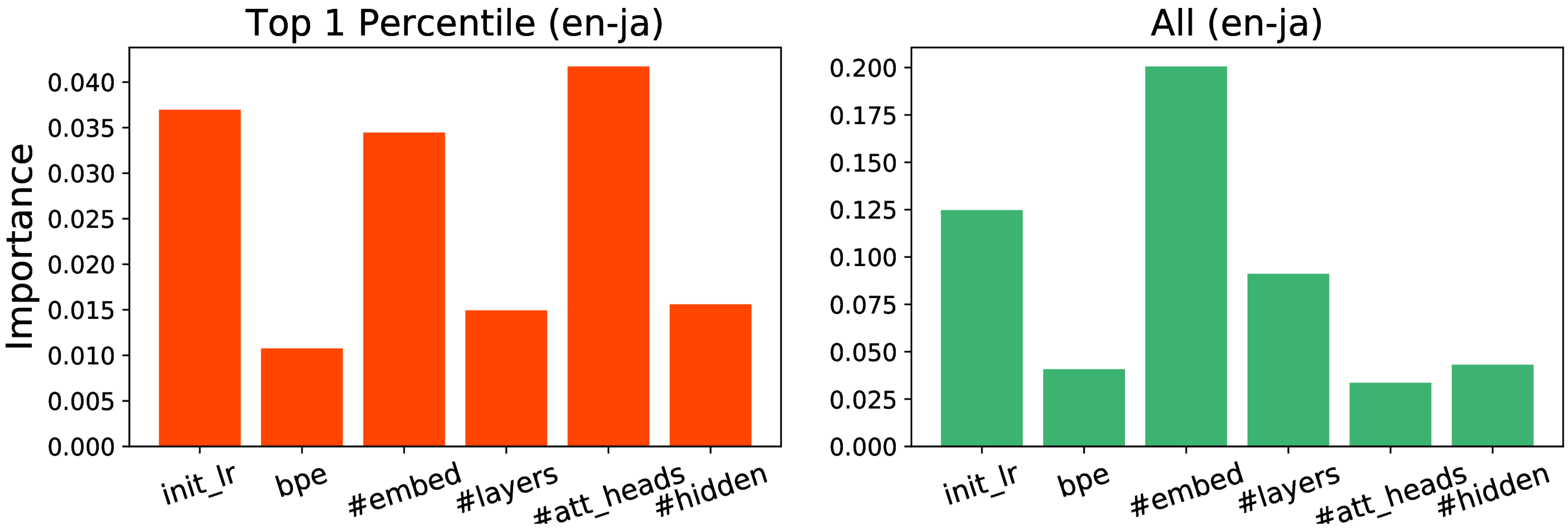
Application II. Multiobjective Optimization



Application III. Hyperparameter Analyses

Hyperparameter Importance

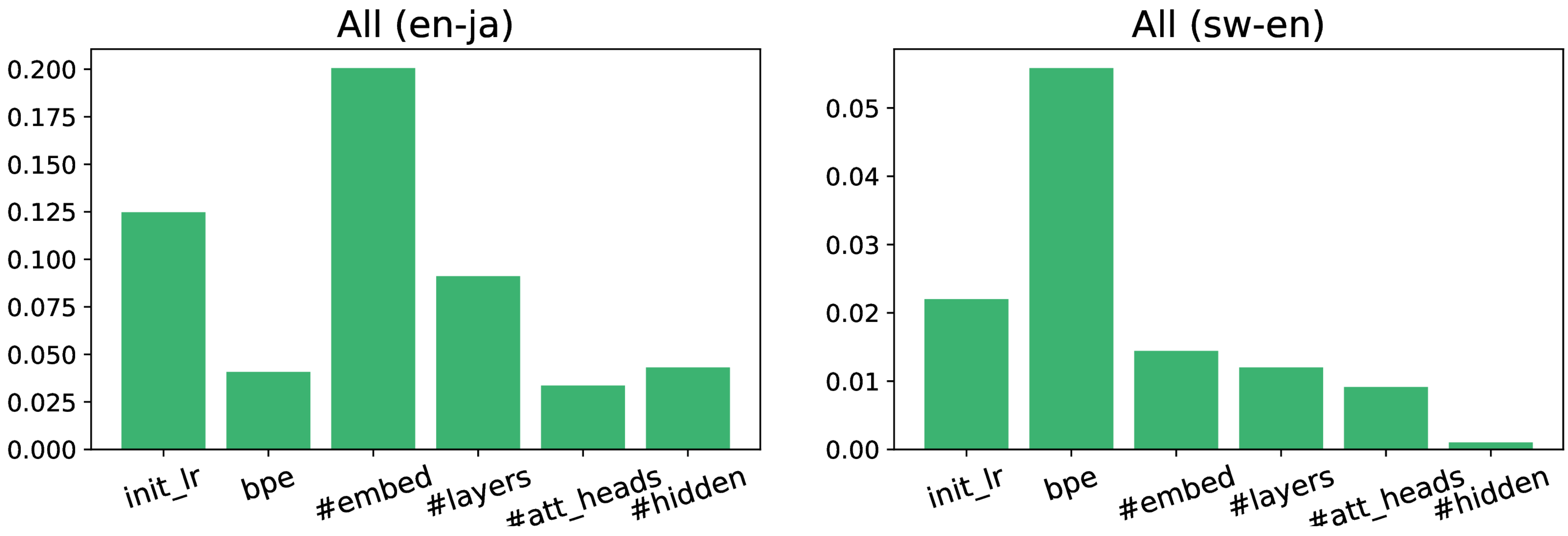
top 1 vs. all NMT models



Application III. Hyperparameter Analyses

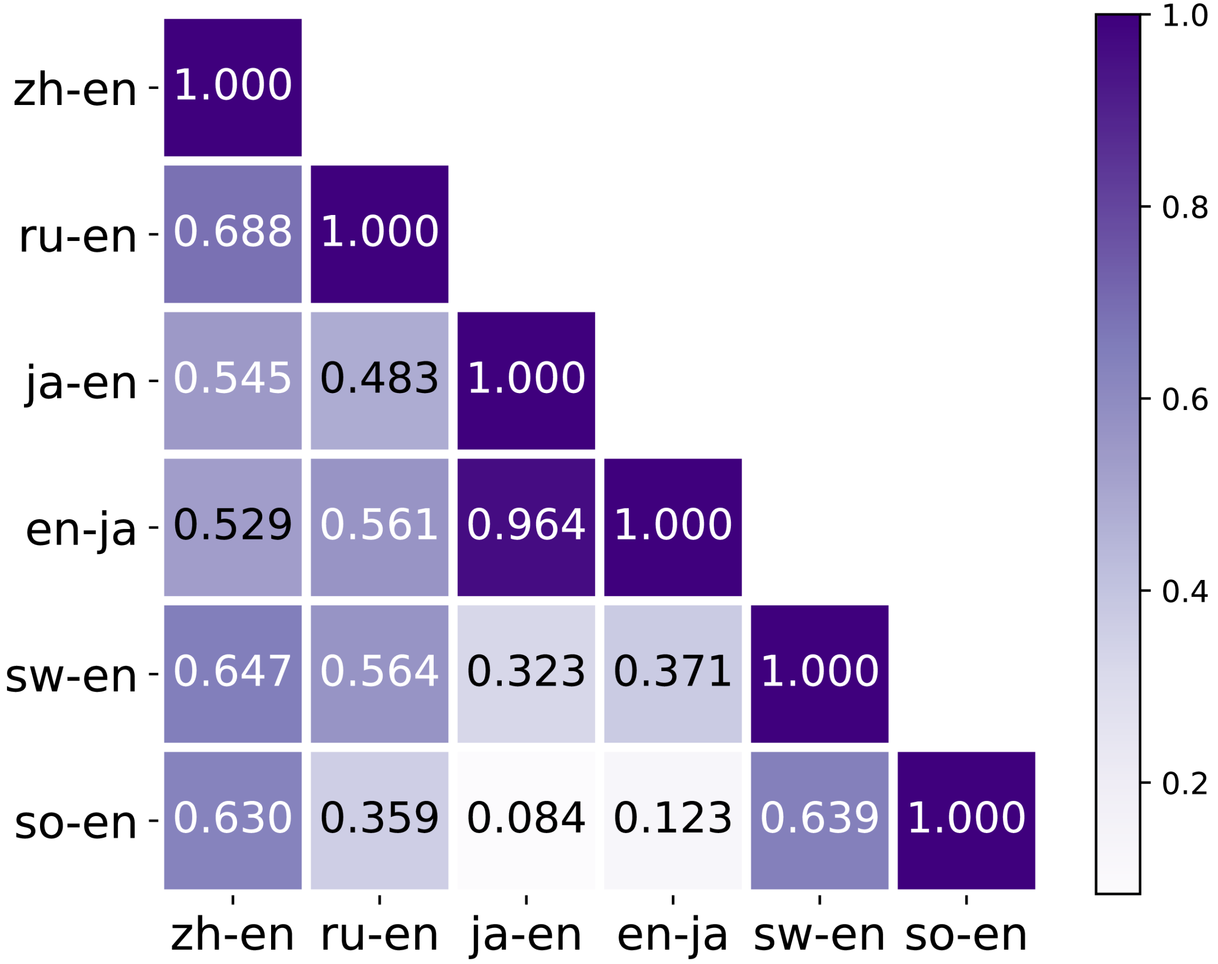
Hyperparameter Importance

en-ja vs. sw-en



Application III. Hyperparameter Analyses

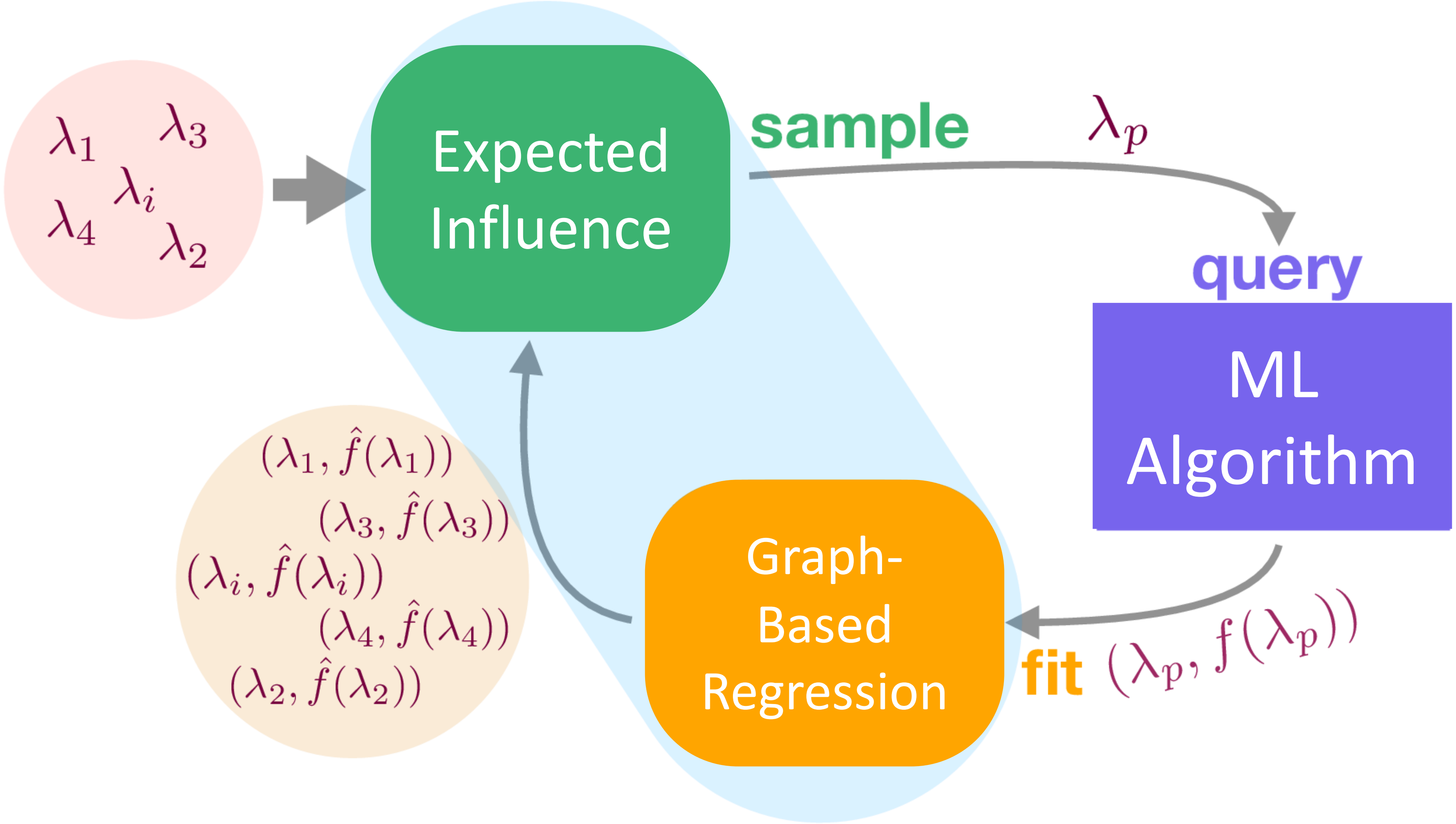
Hyperparameter Ranking Correlation



3. Contributions

- a new HPO benchmark dataset
  (tabular dataset)
- **a new HPO algorithm**
  (graph-based semi-supervised learning)

Graph-Based SMBO



Graph-Based Regression (Surrogate Model)

Let

- $G = (V, E)$ be a graph with nodes V, and edges E.
- $V = L \cup U$, L denote the labeled nodes, U the unlabeled.
- W be the edge weights.
- f be the soft labels of nodes.

Labels of U can be predicted by minimizing the energy function:

$$E(f) = \frac{1}{2}\sum_{i,j} w_{i,j}\left(f(i) - f(j)\right)^2,$$

with the constraint that $f(i), i \in L$ are true labels. (label propagation)

(this work) **Expected Influence** (Acquisition Function)

Idea:

To query a point such that, if its soft label f is observed, has the highest potential to change $f(i)$ for all the node i as we re-run label propagation through the graph.

Results:

It outperforms *expected improvement* significantly when combined with *graph-based regression*.

(this work) **Expected Influence** (Acquisition Function)

- Scale f to be within $[0, 1]$.
- If we were to query an unlabeled point k:
  - its label is 1, with prob $f(k)$
  - its label is 0, with prob $1 - f(k)$
- Include k as a newly-added "labeled" point and re-run label propagation:
  - k is added with label 1, $f^{+(\lambda_k,1)}(i)$ are the new predictions for points i
  - k is added with label 0, $f^{+(\lambda_k,0)}(i)$ are the new predictions for points i
- If k is an influencer,
  - added with label 1, $f^{+(\lambda_k,1)}(i)$ will be large for i
  - added with label 0, $1 - f^{+(\lambda_k,0)}(i)$ will be large for i

(this work) **Expected Influence** (Acquisition Function)

We want to seek a point that maximizes the expected influence score defined as the following:

$$a_{EIF}(\lambda_k) = \left(1 - f(k)\right) \sum_{i=1}^{n} (1 - f^{+(\lambda_k,0)}(i))$$

$$+ f(k) \sum_{i=1}^{n} f^{+(\lambda_k,1)}(i)$$

4. Summary

Summary

Li and Talwalkar (2019): *"Of the 12 papers published since 2018 at NeurIPS, ICML, and ICLR that introduce novel Neural Architecture Search methods, none are exactly reproducible."*

- **Our benchmarks are reproducible.**

  dataset: https://github.com/Este1le/hpo_nmt

  code: https://github.com/Este1le/gbopt

- **Our benchmarks are efficient.**

  One can perform multiple random trials of the same algorithm to test robustness.

- **Our benchmarks are effective.**

  We cover various MT corpora and a reasonable search space.

We hope our dataset can facilitate reproducible research and rigorous evaluation of HPO for complex and expensive models.

Reproducible and Efficient Benchmarks for Hyperparameter Optimization of Neural Machine Translation Systems

Q & A