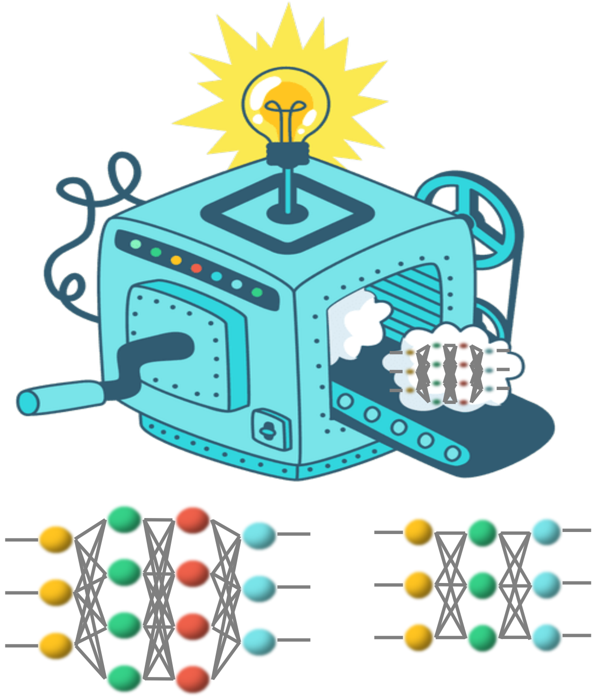
Reproducible and Efficient Benchmarks for Hyperparameter Optimization of Neural Machine Translation Systems

Xuan Zhang   Kevin Duh



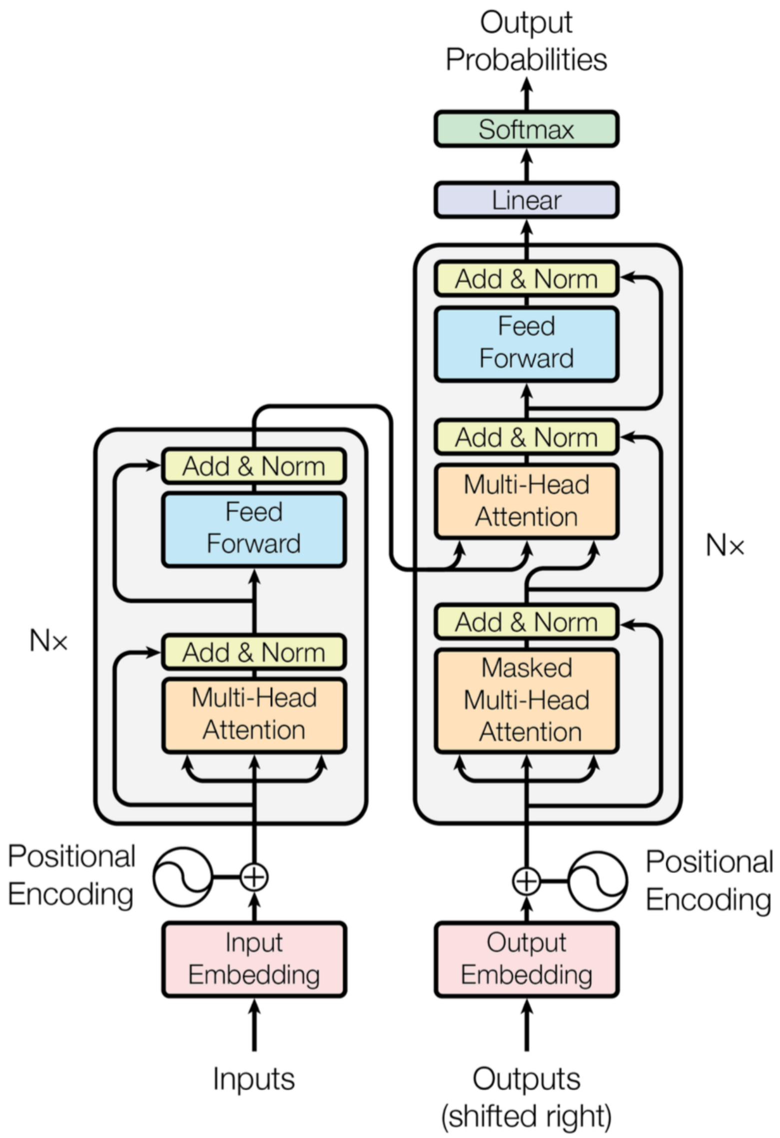
JOHNS HOPKINS
WHITING SCHOOL of ENGINEERING

EMNLP 2020

Machine Translation

Architecture Hyperparameters:
- #layer
- #units/layer
- #embed

Training Hyperparameters:
- optimizer type
- learning rate
- batch size



The Transformer model architecture.[1]

Objectives

Training Accuracy:
- BLEU
- perplexity

Computational Cost:
- inference speed
- model size

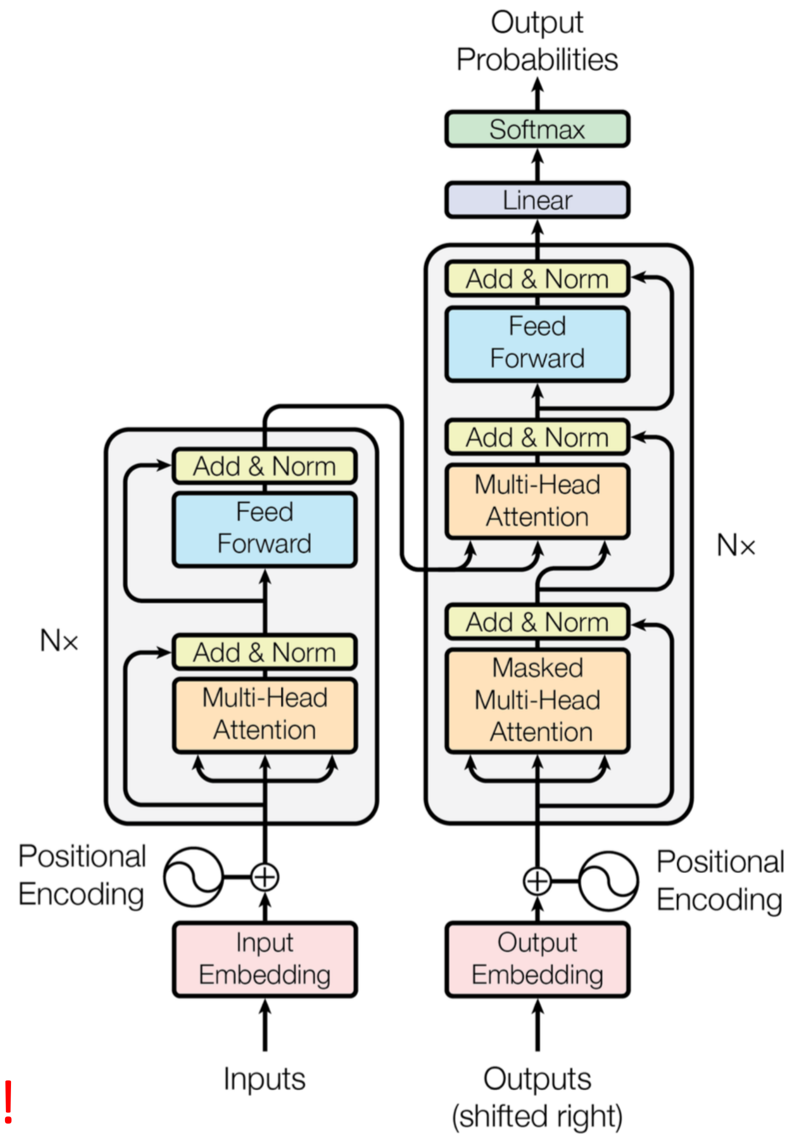
[1]Vaswani, Ashish, et al, "Attention is all you need." Advances in neural information processing systems. 2017.

Machine Translation

<u>Architecture Hyperparameters:</u>
- #layer
- #units/layer
- #embed

<u>Training Hyperparameters:</u>
- optimizer type
- learning rate
- batch size

⚠ Exponential explosion of choices!



The Transformer model architecture.[1]

Objectives

<u>Training Accuracy:</u>
- BLEU
- perplexity

<u>Computational Cost:</u>
- inference speed
- model size

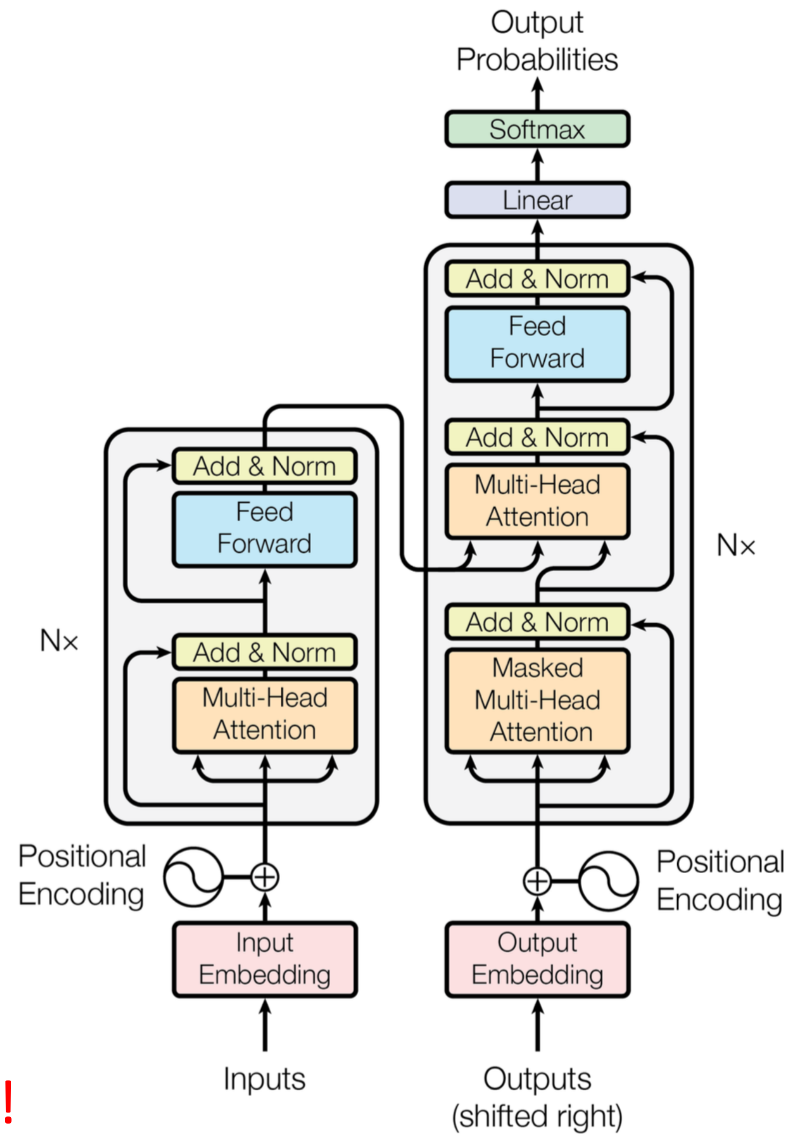
[1]Vaswani, Ashish, et al, "Attention is all you need." Advances in neural information processing systems. 2017.

Machine Translation

Architecture Hyperparameters:
- #layer
- #units/layer
- #embed

Training Hyperparameters:
- optimizer type
- learning rate
- batch size

⚠ Exponential explosion of choices!



The Transformer model architecture.[1]

Objectives

Training Accuracy:
- BLEU
- perplexity

Computational Cost:
- inference speed
- model size

⚠ Difficult to optimize multiple objectives!

[1]Vaswani, Ashish, et al, "Attention is all you need." Advances in neural information processing systems. 2017.

Machine Translation

⚠️ Require significant computational resources to train!

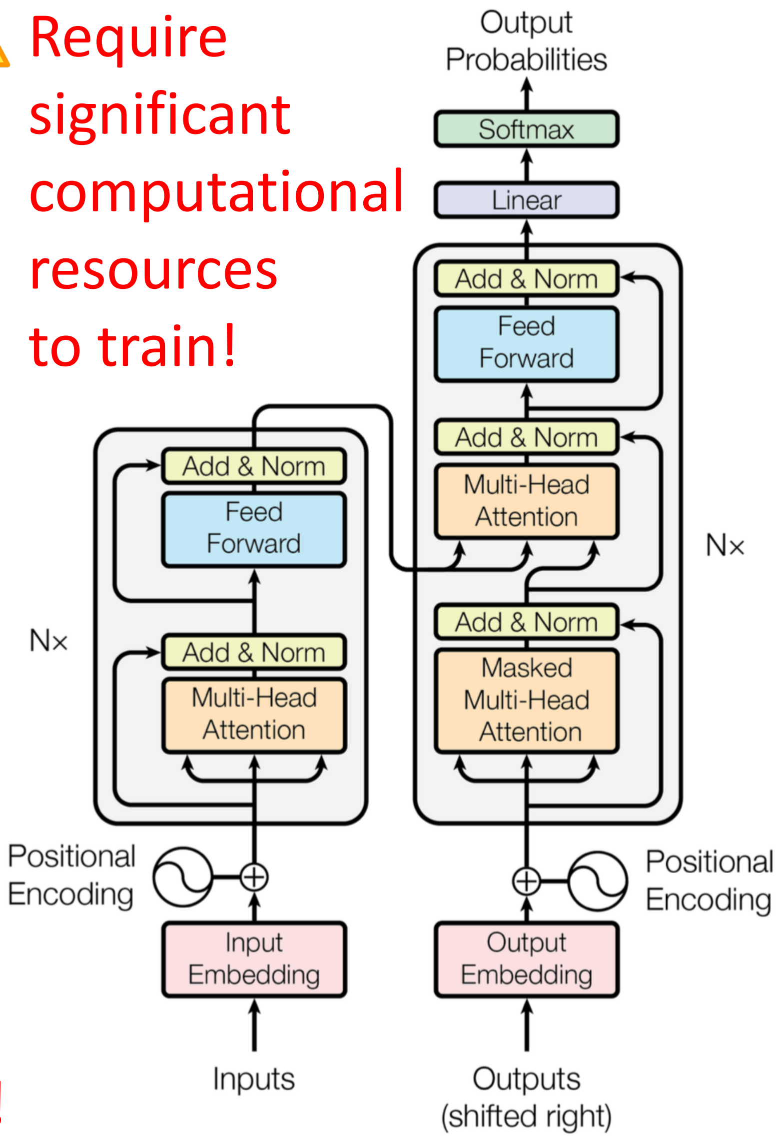
Architecture Hyperparameters:

- #layer
- #units/layer
- #embed

Training Hyperparameters:

- optimizer type
- learning rate
- batch size

⚠️ Exponential explosion of choices!



The Transformer model architecture.[1]

Objectives

Training Accuracy:

- BLEU
- perplexity

Computational Cost:

- inference speed
- model size

⚠️ Difficult to optimize multiple objectives!

[1]Vaswani, Ashish, et al, "Attention is all you need." Advances in neural information processing systems. 2017.

Hyperparameter Optimization (HPO)

Definition

HPO allows to **automatically** find good hyperparameter settings.

Let

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

The HPO problem is to find a configuration $\boldsymbol{\lambda}^*$ that minimizes this loss:

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

Hyperparameter Optimization (HPO)

Algorithms

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
- Tree Parzen Estimator

Hyperparameter Optimization (HPO)

Algorithms

Model-Free Optimization Methods:

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

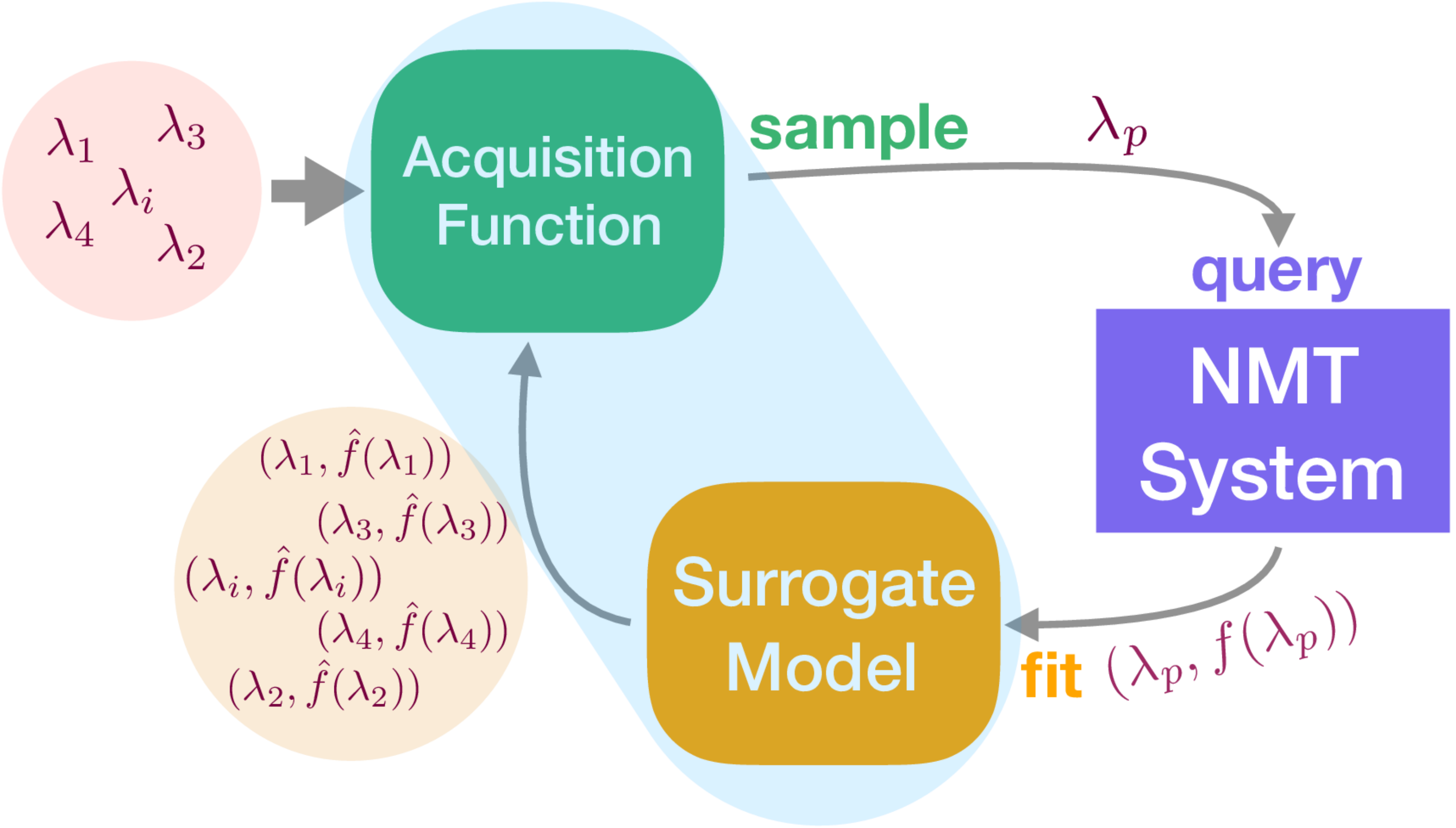
Sequential Model-Based Optimization Methods (SMBO):

This talk

- Bayesian Optimization
- Tree Parzen Estimator

Hyperparameter Optimization (HPO)

SMBO Framework

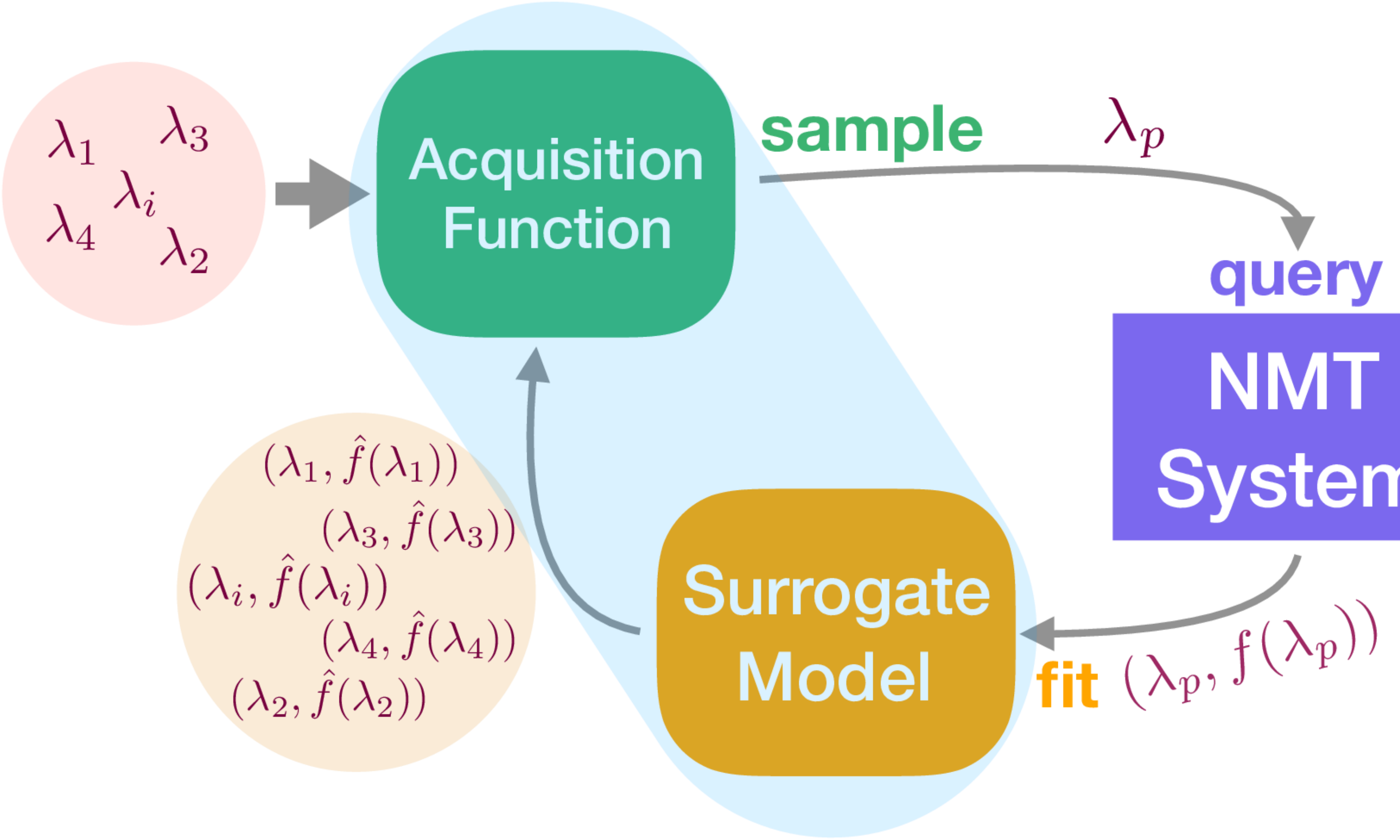


λ: hyperparameter configuration

$f(\lambda)$: model performance (e.g. BLEU)

Hyperparameter Optimization (HPO)

SMBO Framework



λ: hyperparameter configuration

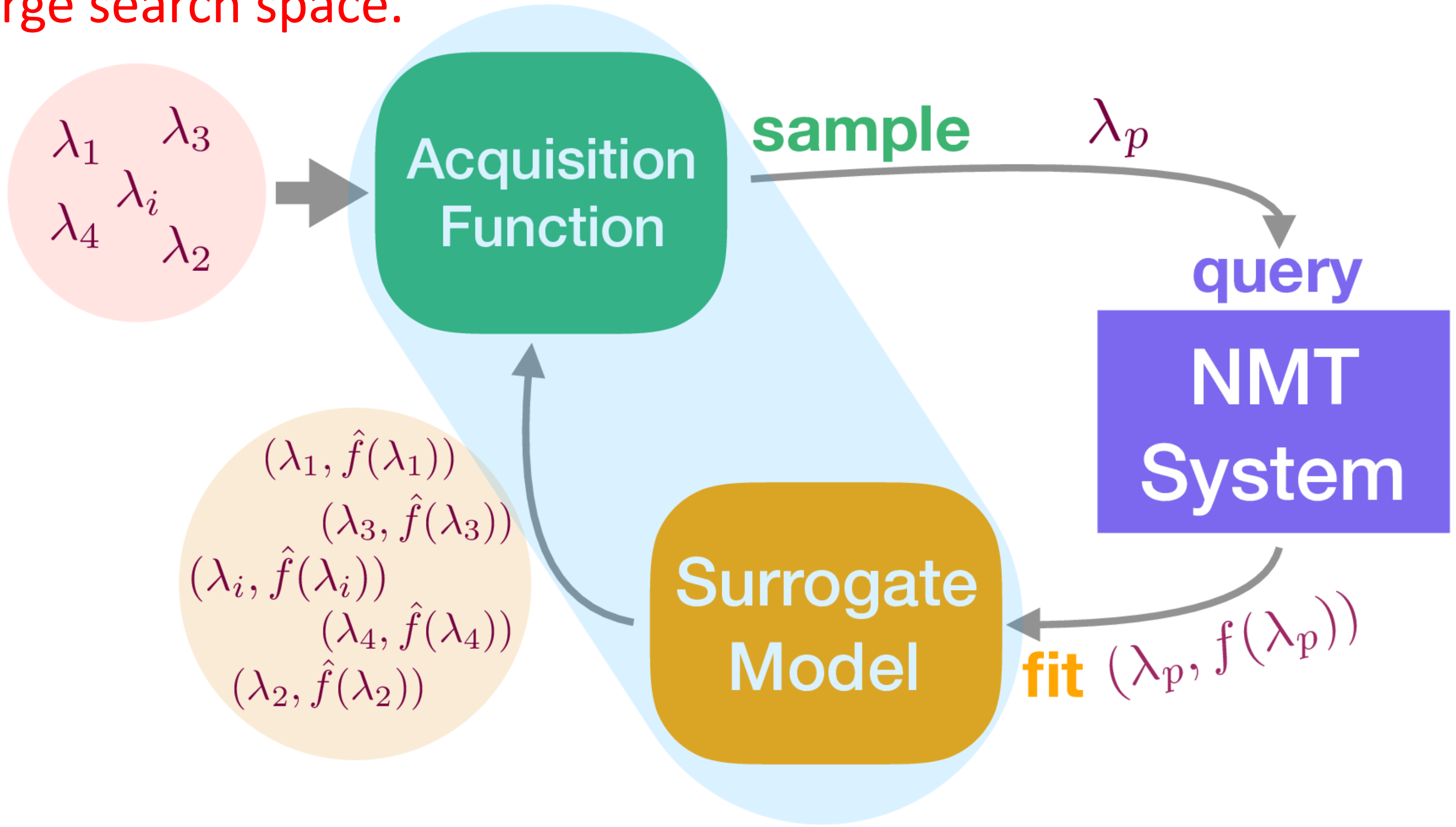
$f(\lambda)$: model performance (e.g. BLEU)

⚠ Each model takes days to weeks to train.

Hyperparameter Optimization (HPO)

SMBO Framework

⚠ Large search space.



λ: hyperparameter configuration

$f(\lambda)$: model performance (e.g. BLEU)

⚠ Each model takes days to weeks to train.

Hyperparameter Optimization

Challenges

- HPO algorithms are expensive. It is not feasible to run too many experiments and compare different HPO algorithms on NMT tasks in practice.

- Li and Talwalkar (2019)[1]: *"Of the 12 papers published since 2018 at NeurIPS, ICML, and ICLR that introduce novel Neural Architecture Search methods, none are exactly reproducible."*

[1]Li, Liam and Talwarkar, Ameet, "Random search and reproducibility for neural architecture search." ICML workshop on automated machine learning. 2019.

Goal

Enable **reproducible** Hyperparameter Optimization (HPO) research on Neural Machine Translation (NMT) tasks.

Contributions

– 01 –

DATASET

We release a benchmark dataset for comparing HPO methods on NMT models.

– 02 –

BENCHMARKS

We benchmark the performance of several HPO methods on both single-objective and multiobjective optimization on our dataset.

– 03 –

ALGORITHM

We propose a novel graph-based HPO method.

Contributions

– 01 –

DATASET

We release a benchmark dataset for comparing HPO methods on NMT models.

This talk

– 02 –

BENCHMARKS

We benchmark the performance of several HPO methods on both single-objective and multiobjective optimization on our dataset.

– 03 –

ALGORITHM

We propose a novel graph-based HPO method.

Dataset

Table-Lookup Framework

Procedure:

1. Train a large number of NMT systems with diverse **hyperparameter configurations** and record their **performance**.
2. Constrain HPO methods to sample from this finite set of models.

Dataset

Table-Lookup Framework

Procedure:

1. Train a large number of NMT systems with diverse **hyperparameter configurations** and record their **performance**.
2. Constrain HPO methods to sample from this finite set of models.

Benefits:

1. Allows HPO developers to simply lookup the performance of NMT systems without training them.
2. Reproducible and efficient HPO experiments.

Limitations:

Table needs to be large enough to cover the hyperparameter space.

Dataset

Specification

- **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 (IARPA MATERIAL): sw-en, so-en (24k lines)
- **Search Space:** 2245 Transformers (1547 GPU days)

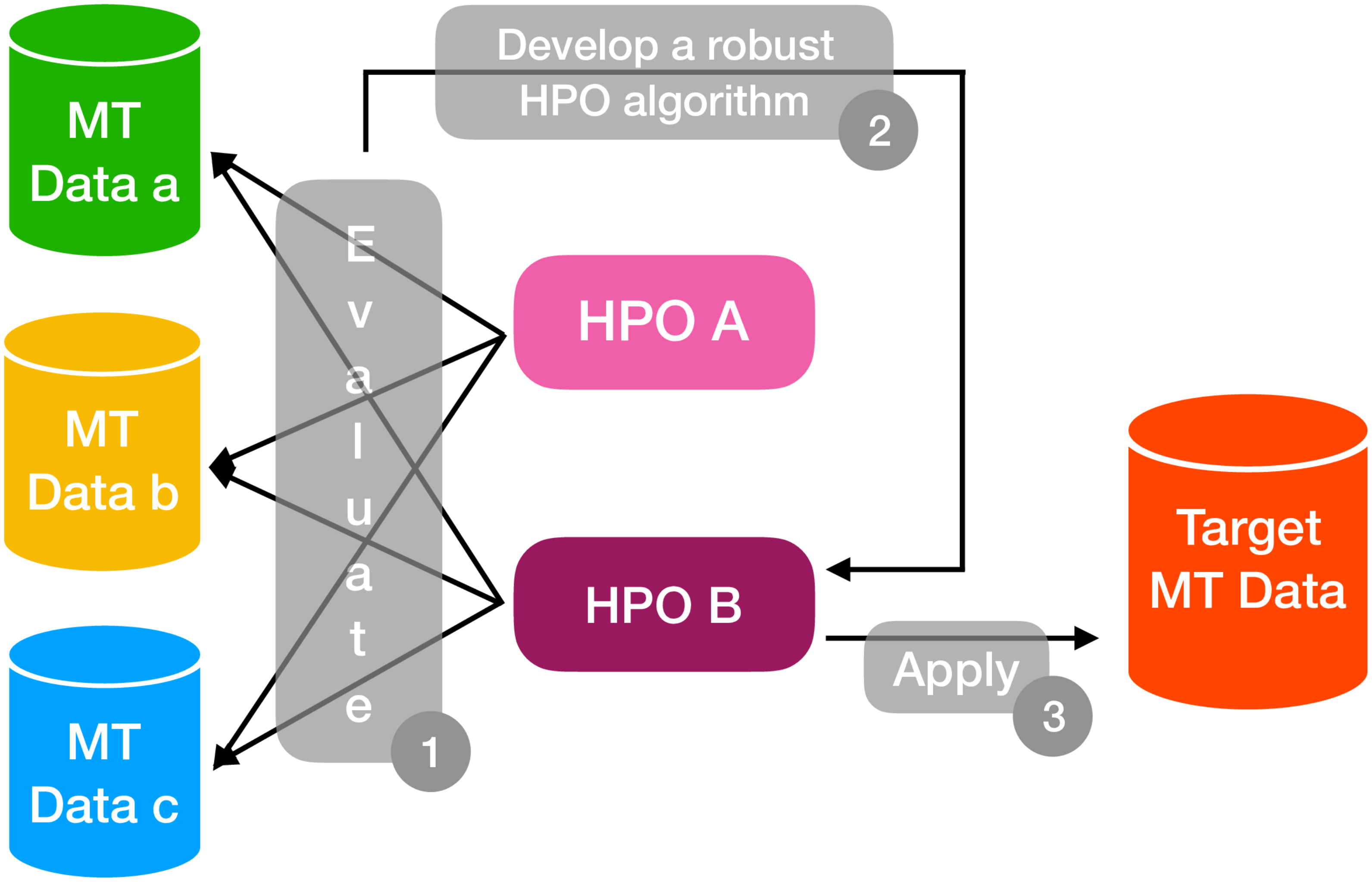
dataset	bpe (1k)	#layers	#embed	#hidden	#att_heads	init_lr (10^{-4})
zh, ru, ja, en	10, 30, 50	2, 4	256, 512, 1024	1024, 2048	8, 16	3, 6, 10
sw	1, 2, 4, 8, 16, 32	1, 2, 4, 6	256, 512, 1024	1024, 2048	8, 16	3, 6, 10
so	1, 2, 4, 8, 16, 32	1, 2, 4	256, 512, 1024	1024, 2048	8, 16	3, 6, 10

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

Dataset

Application

HPO Algorithm Selection

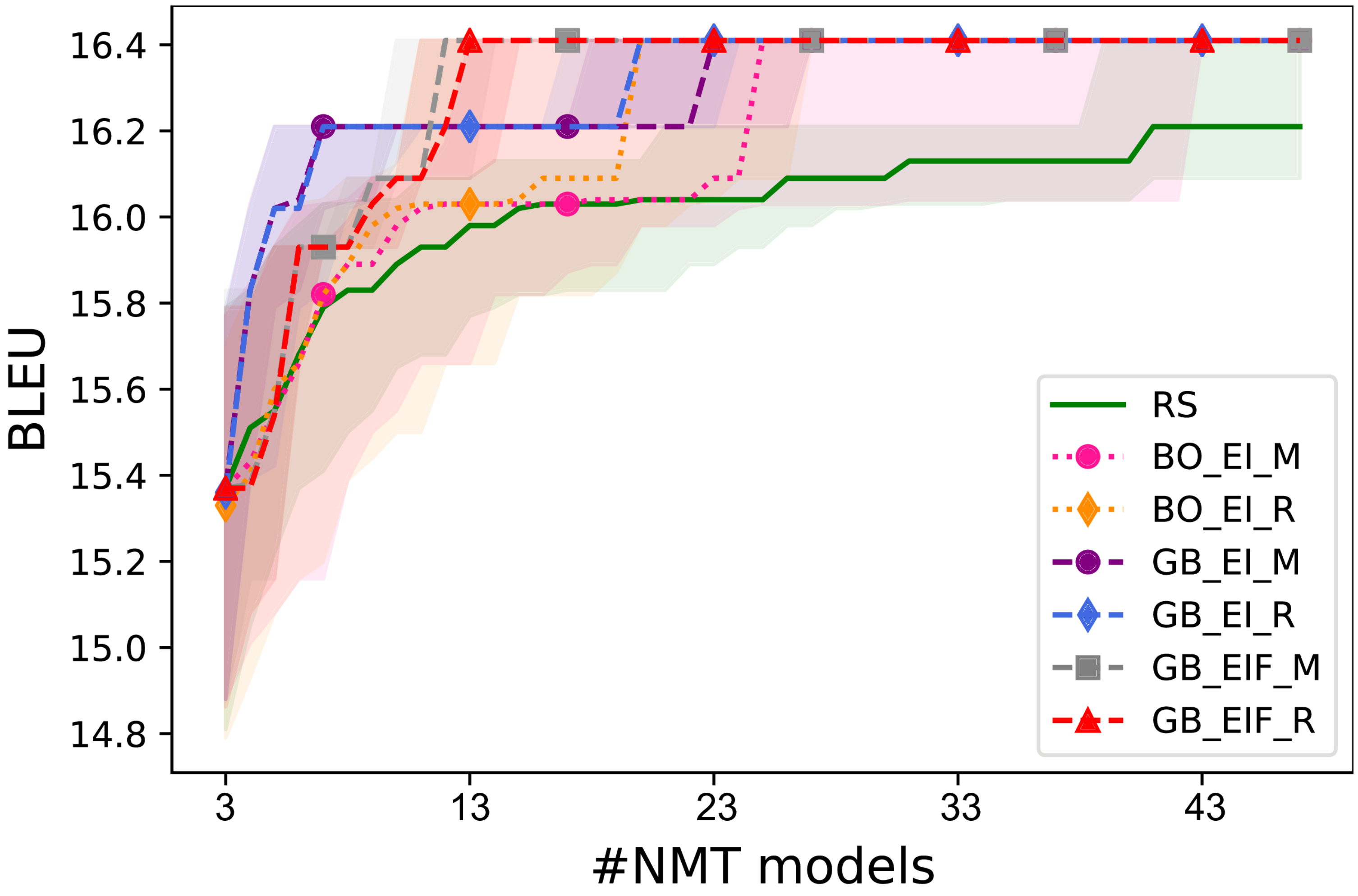


Dataset

Application

HPO Algorithm Selection

Single-objective Optimization



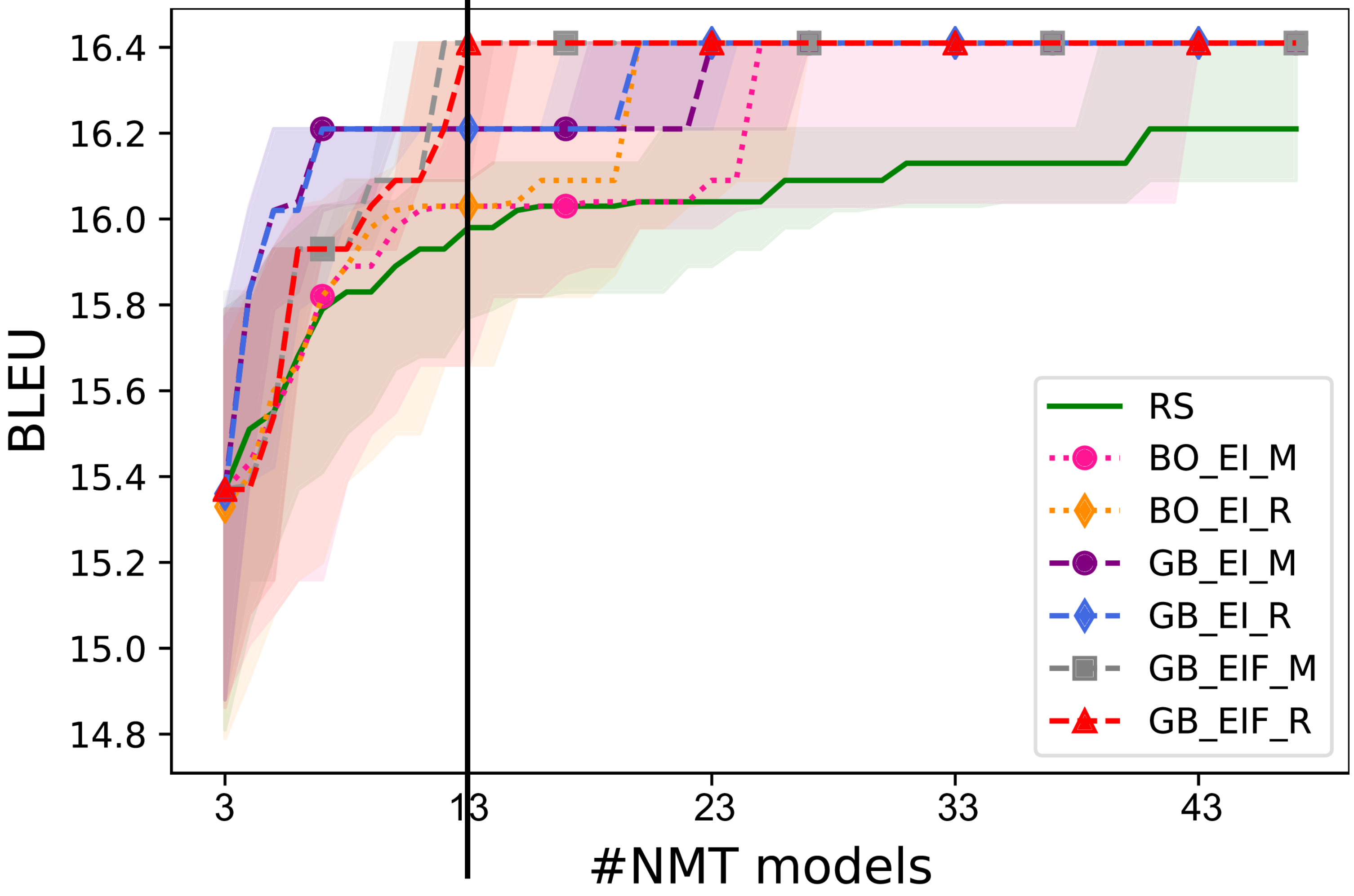
Dataset

Application

HPO Algorithm Selection

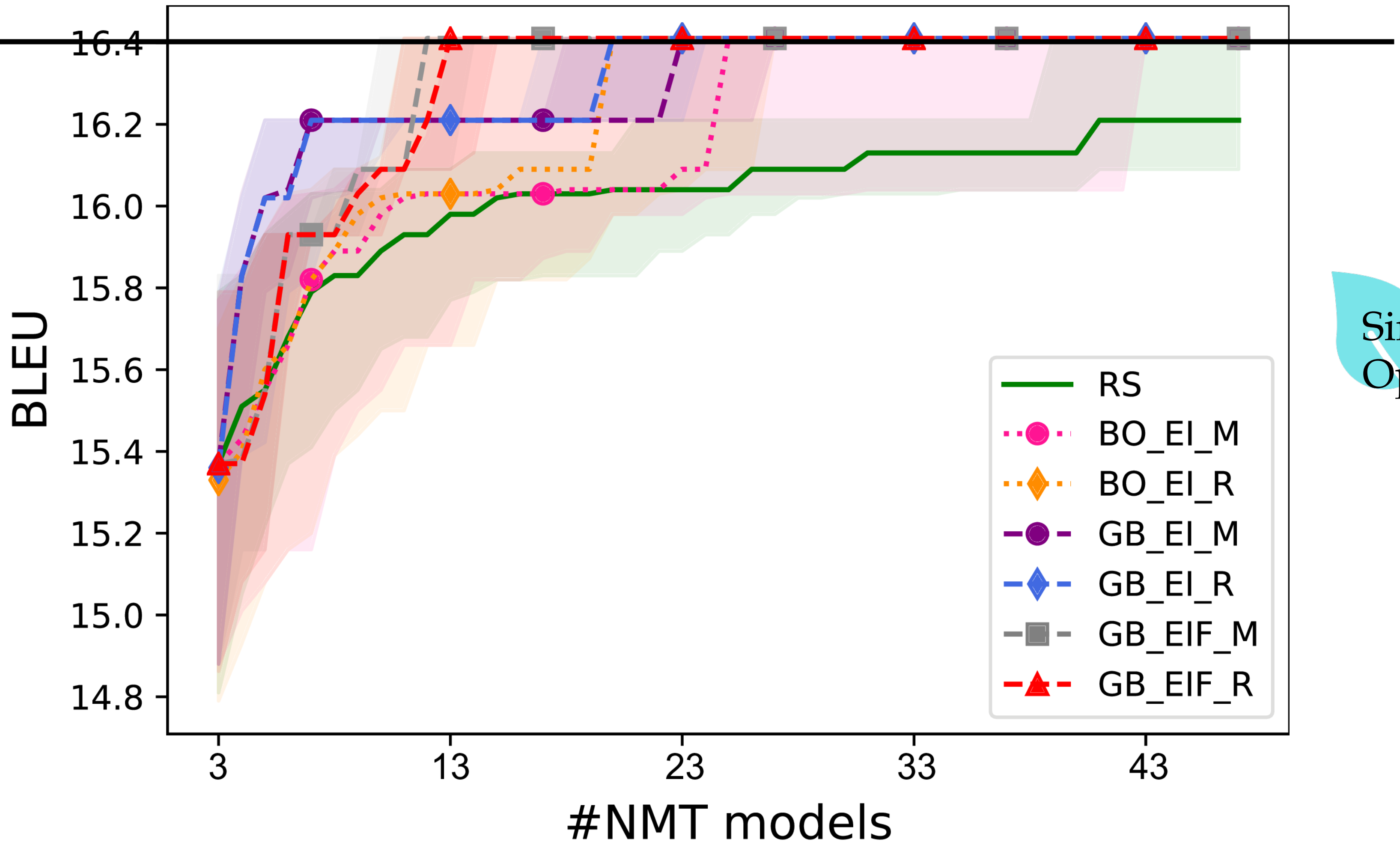
Single-objective Optimization

fix the budget



Dataset

fix the target BLEU

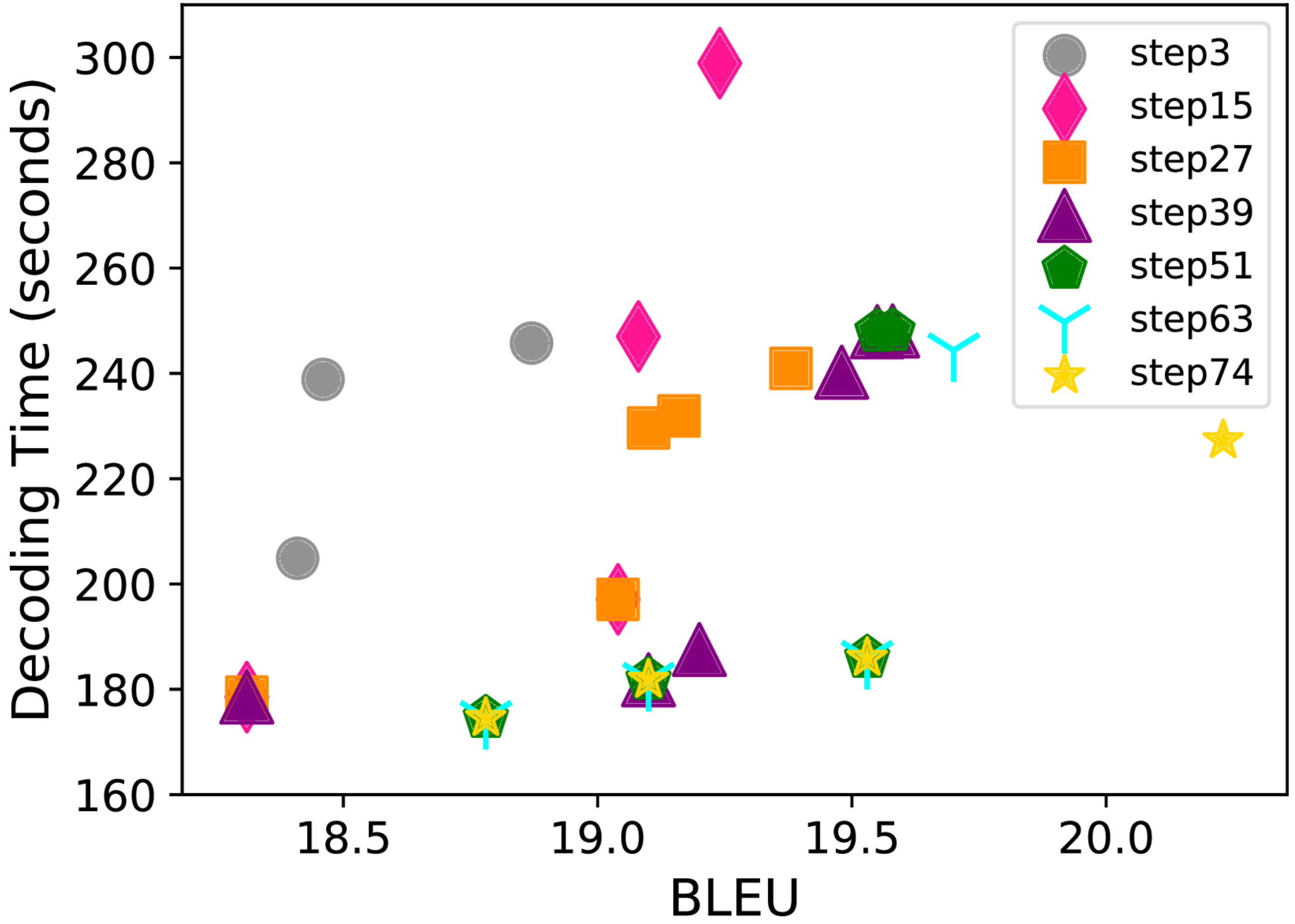


Application

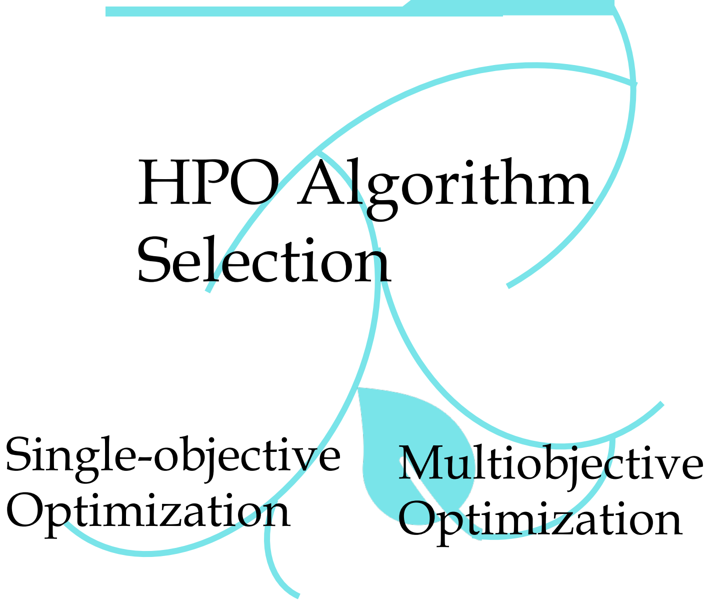
HPO Algorithm Selection

Single-objective Optimization

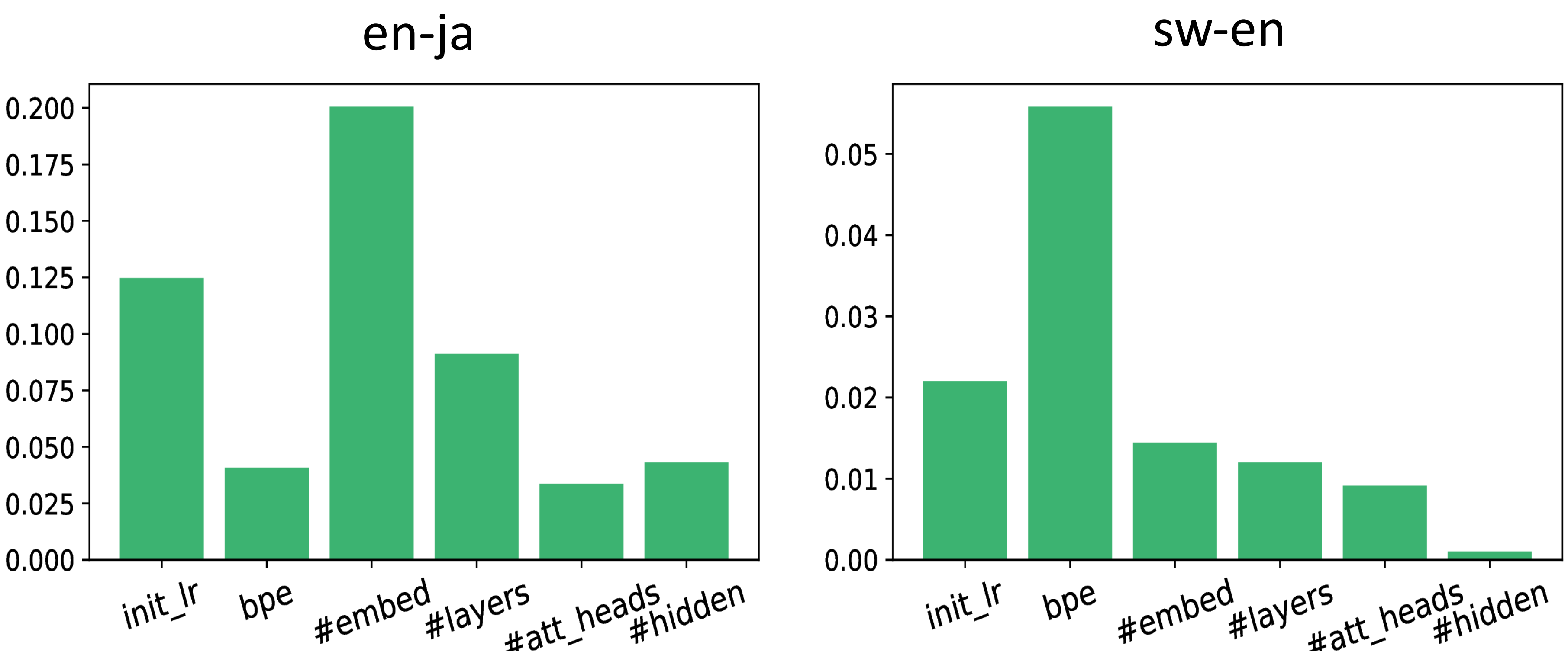
Dataset



Application



Dataset



Hyperparameter Importance

Application

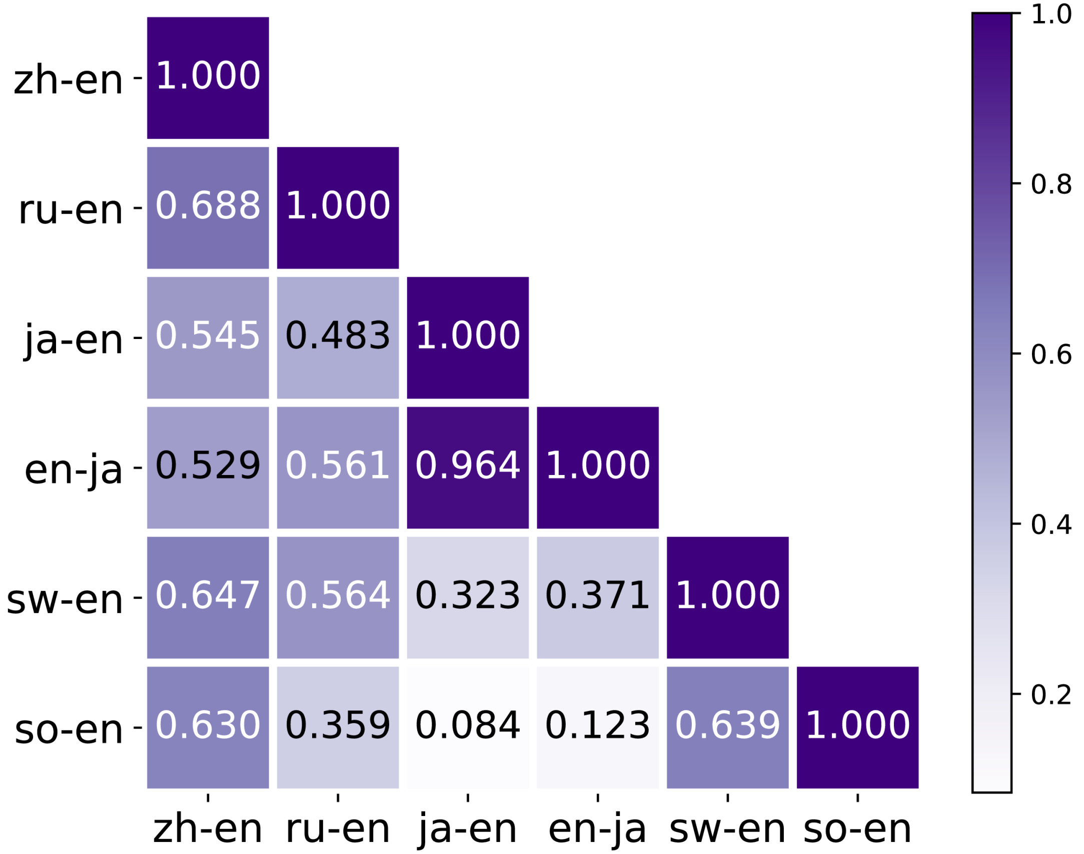
HPO Algorithm Selection

Single-objective Optimization

Multiobjective Optimization

Hyperparameter Analyses

Dataset



Model Ranking Correlation

Application

HPO Algorithm Selection

Single-objective Optimization

Multiobjective Optimization

Hyperparameter Analyses

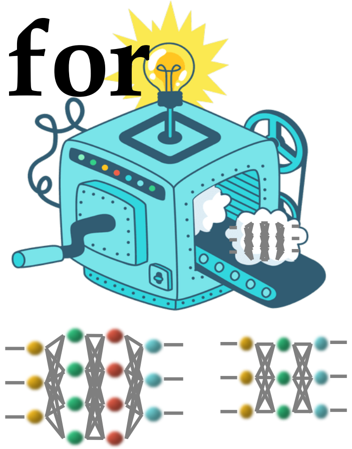
Summary

We provide a tabular dataset for comparing HPO methods on NMT models.

- **Our benchmarks are reproducible.**
  Dataset and code are publicly available.
- **Our benchmarks are efficient.**
  One can perform multiple random trials of the same algorithm to test robustness.

Feel free to utilize our dataset to develop your new HPO methods.

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



PAPER

https://www.mitpressjournals.org/doi/pdf/10.1162/tacl_a_00322

DATASET

https://github.com/Este1le/hpo_nmt

CODE

https://github.com/Este1le/gbopt