Multi-objective Tree-structured Parzen Estimator Meets Meta-learning

Shuhei Watanabe, Noor Awad, Masaki Onishi, Frank Hutter
{ watanabs | awad | fh }@cs.uni-freiburg.de
Department of Computer Science, University of Freiburg, Germany

Summary

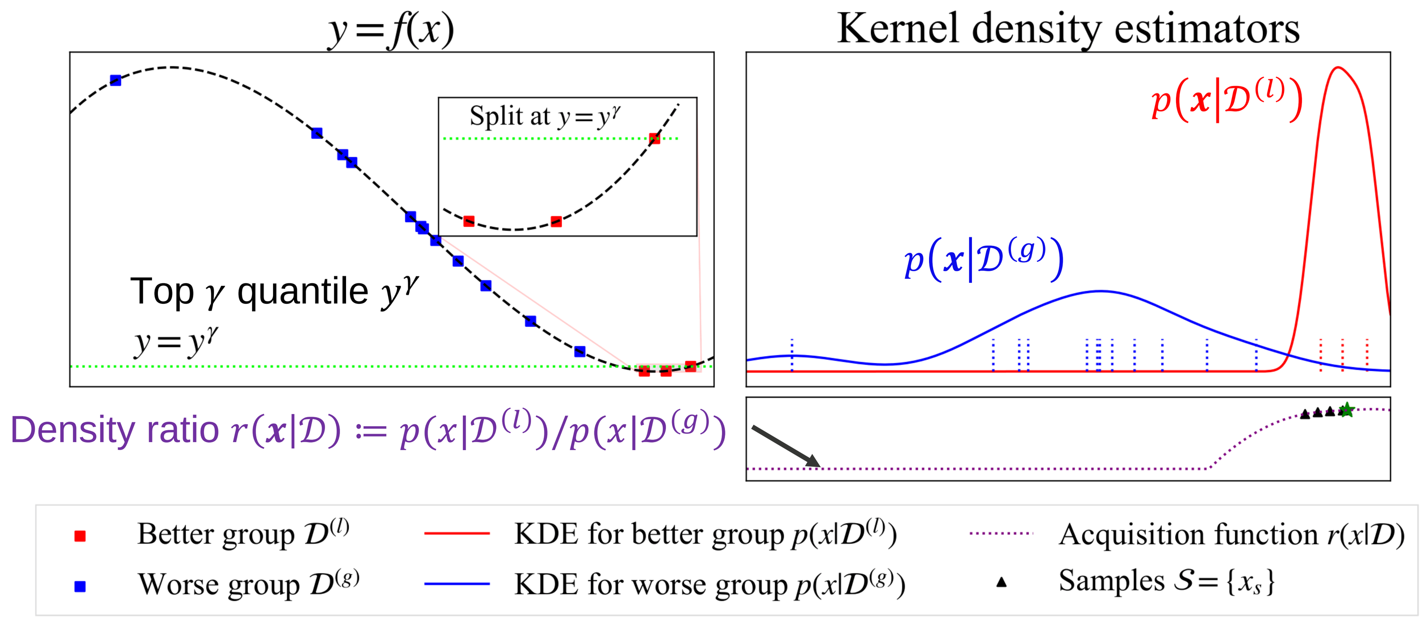
- Propose a meta-learning method for **tree-structured Parzen estimator** (**TPE**), which uses the density ratio of good and bad groups
- Show the **acquisition function** (**AF**) of meta-learning settings is achieved via the density ratio of the joint distributions on task and hyperparameters under conditional shift
- Define the task similarity using the generalized intersection over union to model the joint distributions
- Integrate ϵ-greedy algorithm to pick the next configuration and dimension reduction to stabilize the task similarity measure
- Demonstrate that our method yields good solutions with smaller budget

Tree-structured Parzen estimator (TPE)

- Assume we **minimize** $y = f(\boldsymbol{x})$ and have a set of observations $\mathcal{D} := \{(\boldsymbol{x}_n, y_n)\}_{n=1}^N$
- Define a **lower group** $\mathcal{D}^{(l)}$ as **top-γ quantile** and a **greater group** $\mathcal{D}^{(g)}$ as **the rest**
- Build kernel density estimators (KDEs) using $\mathcal{D}^{(l)}$ and $\mathcal{D}^{(g)}$ ($N^{(l)} := |\mathcal{D}^{(l)}|, N^{(g)} := |\mathcal{D}^{(g)}|$):

$$p(\boldsymbol{x}|\mathcal{D}^{(l)}) = \frac{1}{N^{(l)}} \sum_{\boldsymbol{x}_n \in \mathcal{D}^{(l)}} k(\boldsymbol{x}, \boldsymbol{x}_n), p(\boldsymbol{x}|\mathcal{D}^{(g)}) = \frac{1}{N^{(g)}} \sum_{\boldsymbol{x}_n \in \mathcal{D}^{(g)}} k(\boldsymbol{x}, \boldsymbol{x}_n)$$

- At each iteration, pick the configuration with the best **density ratio** $p(\boldsymbol{x}|\mathcal{D}^{(l)})/p(\boldsymbol{x}|\mathcal{D}^{(g)})$
- To extend it to multi-objective (MO) settings, all we need is a sorting algorithm
- As **our meta-learning method** can be easily **generalized with MO settings** using **a sorting algorithm for MO** settings, describe single objective case



Task-conditioned acquisition function for TPE

- Using the conditional shift, which assumes that the ratio $N^{(l)}/N$ is same for all tasks, the task conditioned AF is computed as:

$$r(\boldsymbol{x}|t, \mathcal{D}_1, \mathcal{D}_2, \dots, \mathcal{D}_T) := r(\boldsymbol{x}|t, \boldsymbol{D}) = \underbrace{\frac{p(\boldsymbol{x}, t|\boldsymbol{D}^{(l)})}{p(\boldsymbol{x}, t|\boldsymbol{D}^{(g)})}}_{\text{Density ratio of joint distributions}}$$

- The joint distributions are computed as:

$$p(\boldsymbol{x}, t|\boldsymbol{D}') = \frac{1}{N'_{\mathrm{all}}} \sum_{m=1}^{T} \sum_{n=1}^{N'_m} \underbrace{\boxed{k_t(t, t_m)}}_{\text{Task kernel to compute}} k_x(\boldsymbol{x}, \underbrace{\boldsymbol{x}_{m,n}}_{\text{The } n\text{-th } \boldsymbol{x} \text{ in } \mathcal{D}_m})$$

- Compute the task kernel using the task similarity, which will be explained below:

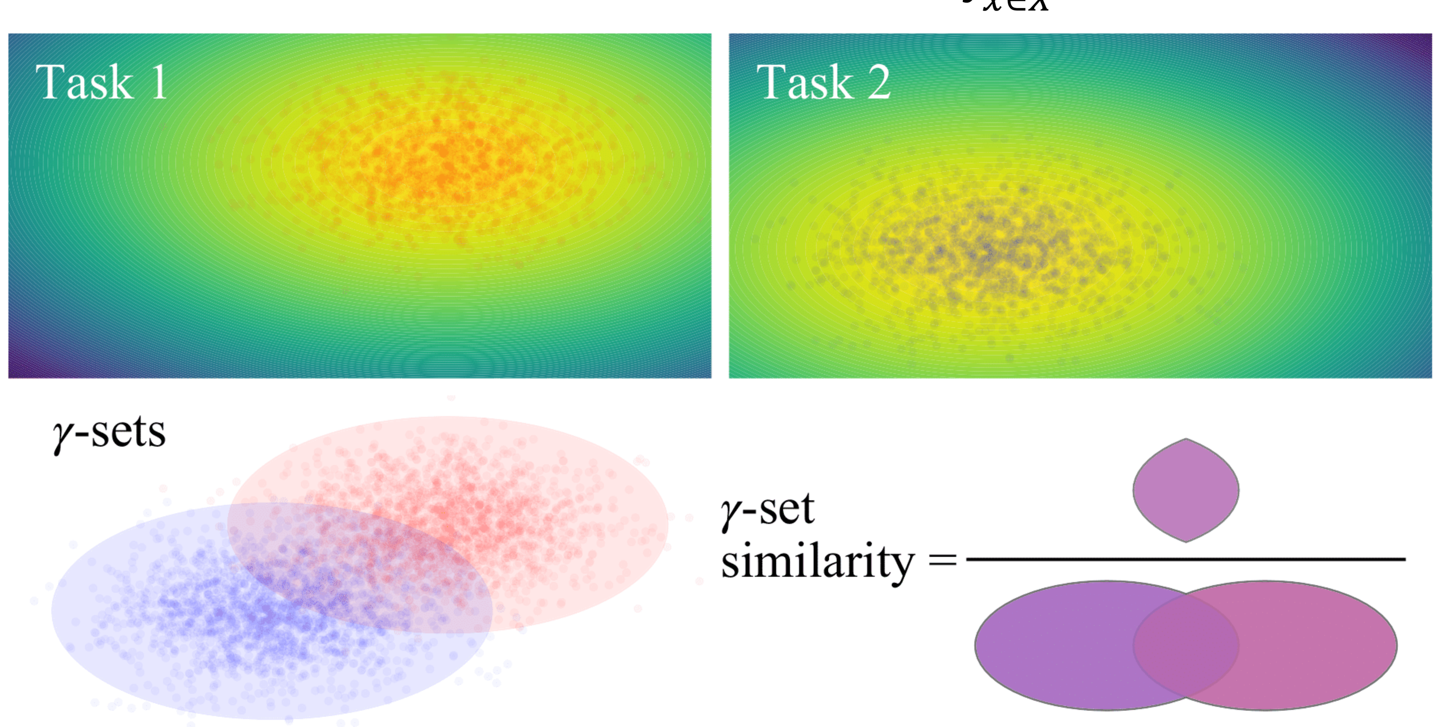
$$k_t(t_i, t_j) = \begin{cases} \frac{1}{T} \hat{s}\left(\mathcal{D}_i^{(l)}, \mathcal{D}_j^{(l)}\right) & (\text{if } i \neq j) \\ 1 - \frac{1}{T} \sum_{k \neq i} \hat{s}\left(\mathcal{D}_i^{(l)}, \mathcal{D}_k^{(l)}\right) & (\text{otherwise}) \end{cases}$$

Task similarity

- Define the task similarity as the overlap of the top-γ quantile sets (γ-sets) $\mathcal{D}_i^{(l)}, \mathcal{D}_j^{(l)}$
- Show the task similarity is computed using $p_i := p(x|\mathcal{D}_i^{(l)}), p_j := p(x|\mathcal{D}_j^{(l)})$ as follows:

$$\hat{s}\left(\mathcal{D}_i^{(l)}, \mathcal{D}_j^{(l)}\right) = \frac{1 - d_{\mathrm{tv}}(p_i, p_j)}{1 + d_{\mathrm{tv}}(p_i, p_j)}$$

$$\text{where } d_{\mathrm{tv}}(p_i, p_j) := \int_{x \in X} |p_i(x) - p_j(x)| dx$$

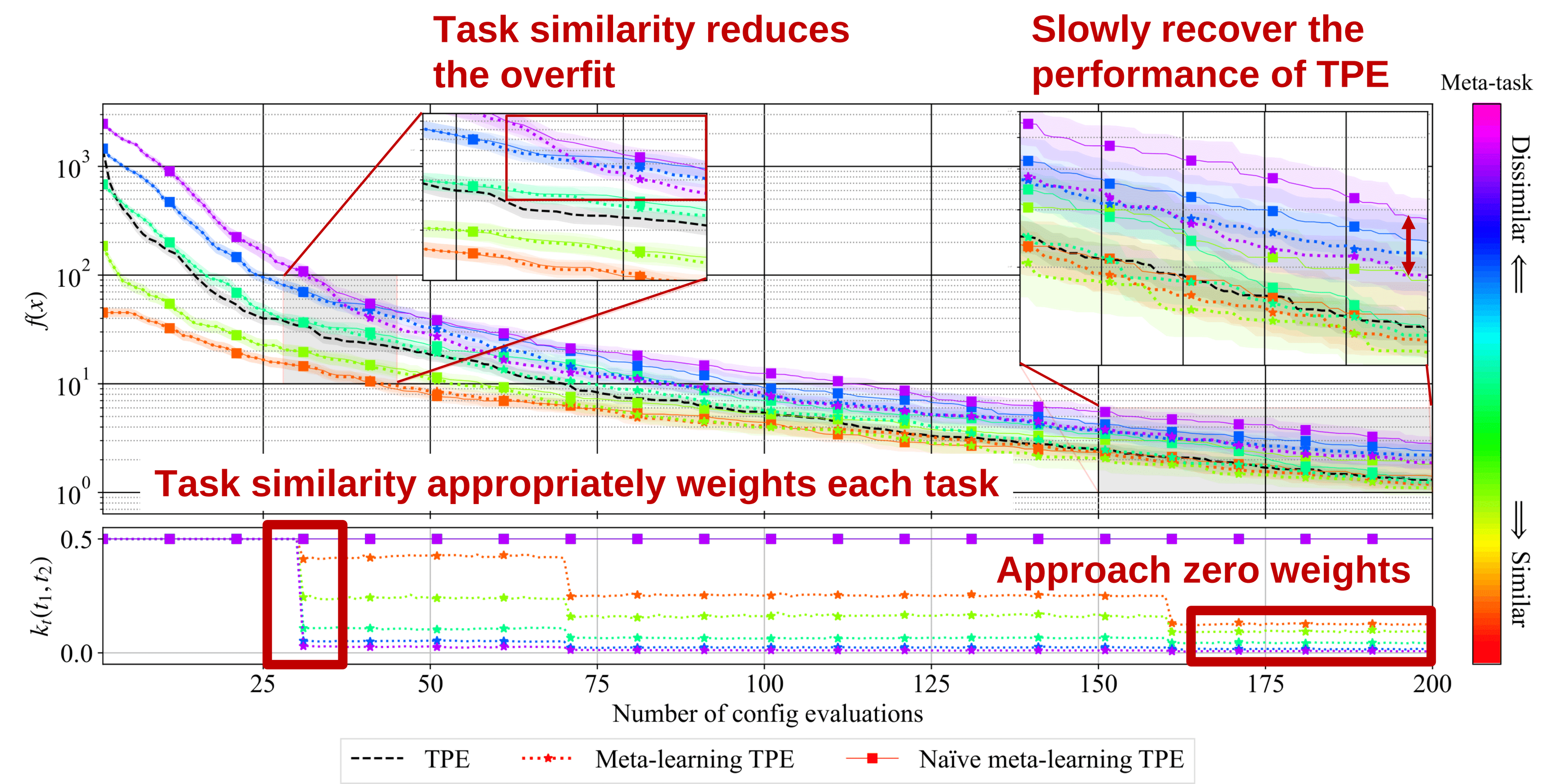


The effect of the task similarity

- First modify the task similarity to solve the following two issues:
  1. Prone to be zero for high dimensions (Curse of dimensionality)
  2. No guarantee of $\mathcal{D}^{(l)}$ to be the γ-set

- **Solution I**: Dimension reduction by ANOVA
- **Solution II**: ϵ-greedy algorithm for the optimization of AF
- Those solutions stabilize the task similarity approximation
- **Theoretical guarantee to recover the performance of TPE** for infinite budget

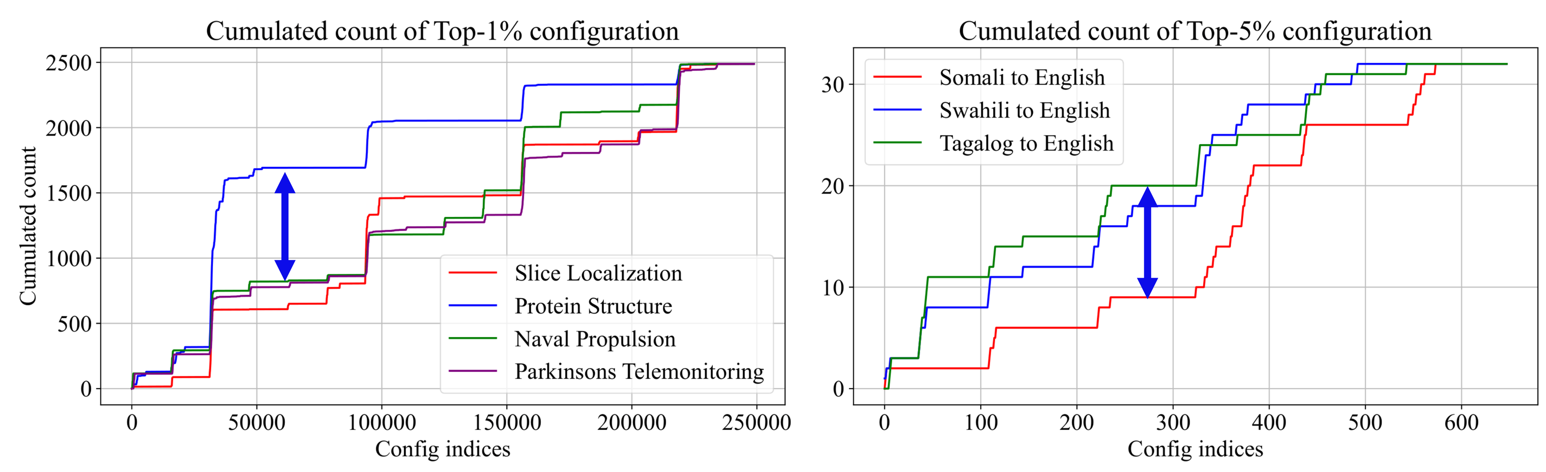
- Compare our method with naïve meta-learning, uses the same weights for each task



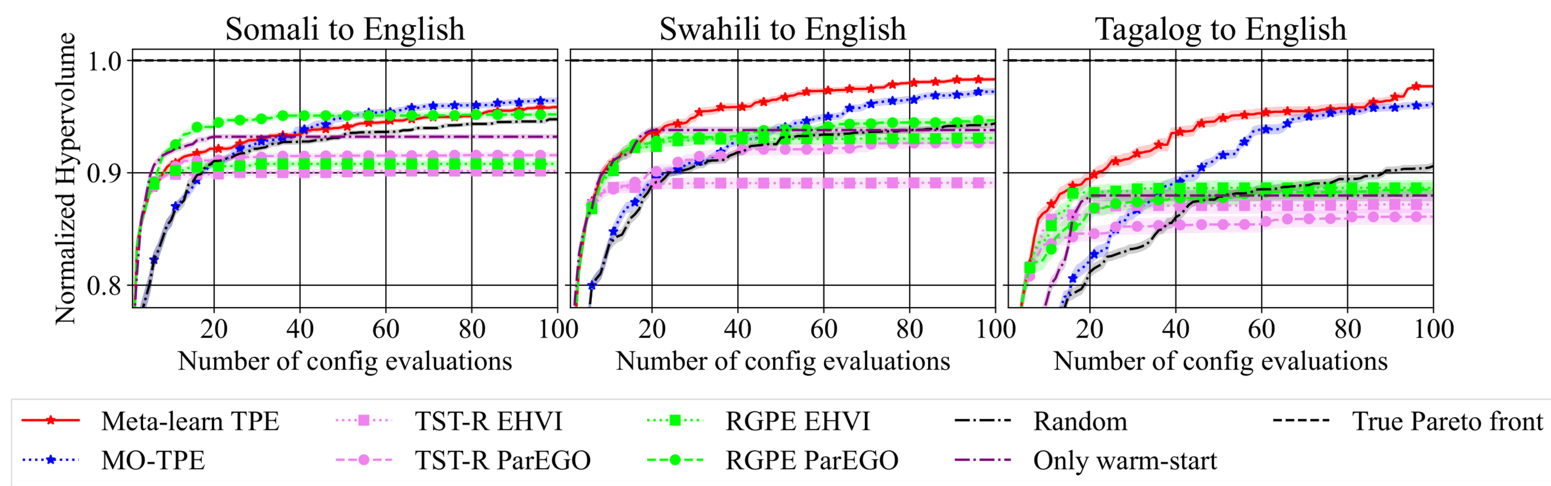
Experiments on tabular benchmarks

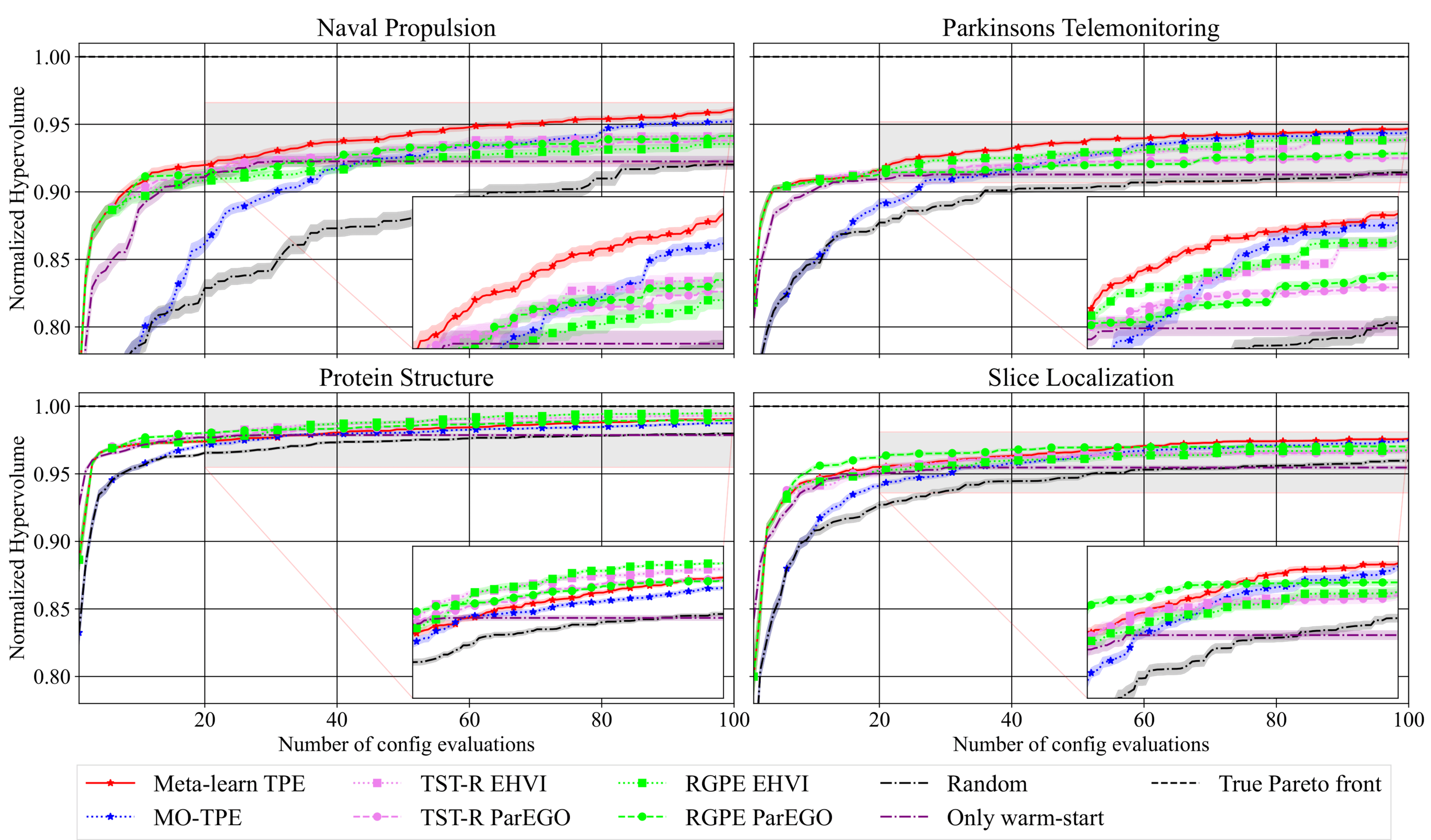
- **Summary of our propositions**
  - **Proposition I** (task-conditioned AF + task similarity + task kernel)
    1. Allow TPE to jointly model multiple tasks
    2. Reduce the contributions from unrelated tasks
  - **Proposition II** (ϵ-greedy algorithm + dimension reduction by f-ANOVA)
    3. Stabilize the task similarity approximation
    4. Guarantee the TPE performance for infinite budget
- **Setup**
  - 7 benchmarks: HPOlib (4 datasets), NMT-Bench (3 datasets)
  - 50 random configurations from the other datasets for meta-learning
  - 2 objectives: performance metric and runtime
  - 20 different random seeds

Somali-to-English and Protein structure are dissimilar to other tasks



- **Results**
  0. Our method won **the first prize in AutoML2022: Multiobjective Hyperparameter Optimization for Transformers**, which used NMT-Bench
  1. Outperform state-of-the-art meta-learning Bayesian optimization methods
  2. **Exhibit slow-start in Somali-to-English**, which is dissimilar to other tasks
  3. **Not obvious slow-start in Protein-structure**, which is dissimilar to other tasks
  4. Imply that the dimension reduction scheduling could be better-tuned although our method recovers the performance of TPE if the budget is abundant





github.com/nabenabe0928/meta-learn-tpe