7th ICML Workshop on Automated Machine Learning (2020)

Multi-Source Unsupervised Hyperparameter Optimization

M. Nomura[*] NOMURA_MASAHIRO@CYBERAGENT.CO.JP
CyberAgent, Inc.

Y. Saito[*] SAITO.Y.BJ@M.TITECH.AC.JP
Tokyo Institute of Technology

Abstract

How can we conduct efficient hyperparameter optimization for a completely new task? In this work, we consider a novel setting, where we search for the optimal hyperparameters for a target task of interest using only unlabeled target task and 'somewhat relevant' source task datasets. In this setting, it is critical to estimate the ground-truth target task objective using only the available information. We propose estimators to unbiasedly approximate the ground-truth with a desirable variance property. Building on these estimators, we provide a general and tractable hyperparameter optimization procedure for our setting. The experimental evaluations demonstrate that the proposed framework broadens the applications of automated hyperparameter optimization.

1. Introduction

Hyperparameter optimization (HPO) has been a pivotal part of machine learning (ML) and contributed to achieving a good performance in a wide range of tasks (Feurer and Hutter, 2019). For example, it is widely acknowledged that the performance of deep neural networks depends greatly on the configuration of the hyperparameters (Dacrema et al., 2019; Henderson et al., 2018; Lucic et al., 2018). HPO is a special case of a black-box function optimization problem, where the input is a set of hyperparameters and the output is a validation score. Among the black-box optimization methods, adaptive algorithms, such as Bayesian optimization (BO) (Snoek et al., 2012; Frazier, 2018) have shown superior empirical performance compared with traditional algorithms, such as grid search or random search (Frazier, 2018).

A critical assumption in HPO is the **availability of an accurate validation score** (, which is often denoted as f (Frazier, 2018)). However, in reality, there are many cases where we cannot access the ground-truth f of the task of interest (referred to as target task hereinafter). For example, in display advertising, predicting the effectiveness of each advertisement, i.e., *click-through rates* (CTR), is important for showing relevant advertisements to users. Therefore, it is necessary to conduct HPO before a new advertisement campaign starts. However, for new advertisements that have not yet been displayed to users, one cannot use labeled data to conduct HPO. In this case, the standard HPO procedure is infeasible, as one cannot utilize the labeled target task data and the true validation score of the ML model under consideration.

In this work, we address the infeasibility issue of HPO when the labels of the target task are unavailable. To formulate the situation, we first introduce a novel HPO setting called

∗. equal contribution

multi-source unsupervised hyperparameter optimization (MSU-HPO). In MSU-HPO, it is assumed that we do not have the labeled data for a target task. However, we do have the data for some source tasks with a different distribution from the target task. It is natural to assume that we have access to multiple source tasks in most practical settings. In the display advertising example, several labeled datasets of old advertisements that have already been deployed are often available, which we can use as labeled source task datasets. To the best of our knowledge, no HPO approach exists that can address a situation without labeled target task data, despite its significance and possibility for applications.

A problem with MSU-HPO is that the ground-truth for the target task objective f is inaccessible, and one cannot directly apply the standard HPO procedure. Thus, it is critical to approximate f using only the available data. To this end, we propose two estimators, enabling the evaluation of the ML models on the target task using only the unlabeled target and labeled source datasets. Our estimators are general and can be used in combination with any common black-box optimization technique, such as Gaussian process-based BO (Snoek et al., 2012) and tree-structured parzen estimator (Bergstra et al., 2011). Through theoretical analysis, we show that the proposed estimators can unbiasedly approximate the target task objective, one of which achieves a desirable variance property by selecting useful source tasks based on a task divergence measure. Using the proposed estimators, we present a general and computationally inexpensive HPO procedure for the MSU-HPO setting. Finally, we demonstrate that our estimators work properly through numerical experiments with synthetic and real-world datasets.

We summarize the related literature and our main contributions in Appendix A.

2. Problem Setting

Let $\mathcal{X} \subset \mathbb{R}^d$ be the d-dimensional input space and $\mathcal{Y} \subset \mathbb{R}$ be the real-valued output space. We use $p_T(x, y)$ to denote the joint probability density function of the input and output variables $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$. The objective of MSU-HPO is to find the best set of hyperparameters θ with respect to the target distribution:

$$\theta^{opt} = \arg\max_{\theta \in \Theta} f_T(\theta) \tag{1}$$

where Θ is a pre-defined hyperparameter search space and $f_T(\theta)$ is the target task objective, which is defined as the negative generalization error over the target distribution:

$$f_T(\theta) = -\mathbb{E}_{(X,Y)\sim\mathcal{P}_T}\left[L(h_\theta(X), Y)\right] \tag{2}$$

where $L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}_{\geq 0}$ is a bounded loss function such as the zero-one loss and $h_\theta : \mathcal{X} \to \mathcal{Y}$ is an arbitrary machine learning model that predicts the output values using the input vectors with a set of hyperparameters $\theta \in \Theta$.

In a standard hyperparameter optimization setting (Bergstra et al., 2011; Feurer and Hutter, 2019; Snoek et al., 2012), labeled i.i.d. validation samples $\{x_i, y_i\}_{i=1}^{n_T} \sim p_T$ are available, and one can easily estimate the target objective in Eq. (2) by the following empirical mean:

$$\hat{f}_T(\theta; \mathcal{D}'_T) = -\frac{1}{n_T}\sum_{i=1}^{n_T} L(h_\theta(x_i), y_i) \tag{3}$$

where $\mathcal{D}'_T$ is any size n_T of the i.i.d. labeled samples from the target task distribution.

Then, a hyperparameter optimization is conducted directly using the estimated target function in Eq. (3) as a replacement for the ground-truth target objective $f_T(\theta)$ in Eq. (2).

In contrast, under the MSU-HPO setting, labels of the target task are assumed to be unobservable; we can use only **unlabeled** target validation samples denoted as $\mathcal{D}_T = \{x_i\}_{i=1}^{n_T}$ hereinafter. Instead, we assume the availability of the multiple *source task* datasets which is denoted as $\{\mathcal{D}_{S^j}\}_{j=1}^{N_S}$ where j is a source task index and N_S denotes the number of source tasks. Each source task dataset is defined as the i.i.d. **labeled** samples: $\mathcal{D}_{S^j} = \{x_i^j, y_i^j\}_{i=1}^{n_{S^j}} \sim p_{S^j}$ where $p_{S^j}(x, y)$ is a joint probability density function that characterizes the source task j.

Regarding the target and source distributions, we make the following assumptions.

***Assumption* 1.** *Source tasks have support for the target task, i.e.,* $p_T(x) > 0 \Rightarrow p_{S^j}(x) > 0,\ \forall x \in \mathcal{X},\ \forall j \in \{1, \ldots, N_S\}$.

***Assumption* 2.** *Conditional output distributions remain the same between the target and all of the source tasks, i.e.,* $p_T(y|x) = p_{S^j}(y|x),\ \forall j \in \{1, \ldots, N_S\}$.

One critical difficulty of the MSU-HPO setting is that the simple approximation using the empirical mean in Eq. (3) is infeasible, as the labeled target dataset is unavailable. Therefore, it is essential to accurately estimate the target task objective function $f_T(\theta)$ using only an unlabeled target dataset and labeled multiple source datasets.

3. Method

3.1 Unbiased Objective Estimator

A natural first candidate way to approximate the target task objective function is to use the *importance weighting* technique. To define our proposed estimator, we first formally introduce the density ratio between the target task distribution and the source task distribution below.

***Definiton* 1.** *(Density Ratio) For any* $(x, y) \in \mathcal{X} \times \mathcal{Y}$ *with a positive source density* $p_{S^j}(x, y) > 0$*, the density ratio between the target and a source task distributions is*

$$0 \leq w_{S^j}(x, y) = \frac{p_T(x, y)}{p_{S^j}(x, y)} = \frac{p_T(x)}{p_{S^j}(x)} = w_{S^j}(x) \leq C \tag{4}$$

where C *is a positive constant. The equalities are derived from Assumption 2.*

Using the above density ratio function, we define an estimator for the target task objective function called the *unbiased estimator*.

***Definiton* 2.** *(Unbiased Estimator) For a given set of hyperparameter* $\theta \in \Theta$*, the unbiased estimator for the target task objective function is defined as*

$$\hat{f}_{UB}\left(\theta; \{\mathcal{D}_{S^j}\}_{j=1}^{N_S}\right) = -\frac{1}{n}\sum_{j=1}^{N_S}\sum_{i=1}^{n_{S^j}} w_{S^j}(x_i^j) \cdot L(h_\theta(x_i^j), y_i^j) \tag{5}$$

where UB stands for unbiased, $n = \sum_{j=1}^{N_S} n_{S^j}$ *is the total sample size of the source tasks,* $\mathcal{D}_{S^j}$ *is any sample size* n_{S^j} *of the i.i.d. samples from the distribution of source task* j.

The estimator in Eq. (5) is an application of the *importance weighted cross-validation (Sugiyama et al., 2007)* to the multiple-source task setting and can easily be shown to be statistically unbiased for the ground-truth target task objective function in Eq. (2), i.e., $\mathbb{E}\left[\hat{f}_{UB}\left(\theta; \{\mathcal{D}_{S^j}\}_{j=1}^{N_S}\right)\right] = f_T(\theta)$.

We also characterize the variance of the unbiased estimator.

$$\mathbb{V}\left(\hat{f}_{UB}\left(\theta; \{\mathcal{D}_{S^j}\}_{j=1}^{N_S}\right)\right) = \frac{1}{n^2}\sum_{j=1}^{N_S} n_{S^j}\left(\mathbb{E}_{(X,Y)\sim p_{S^j}}\left[w_{S^j}^2(X)\cdot L^2(h_\theta(X),Y)\right] - (f_T(\theta))^2\right) \tag{6}$$

As stated above, the unbiased estimator is a valid approach for approximating a target task objective when multiple source task datasets have distributions different from that of the target task. The problem is that its variance depends on the square value of the density ratio function, which can be huge when there is a source task with a distribution that is dissimilar to that of the target task.

3.2 Variance Reduced Objective Estimator

To address this variance issue of the unbiased estimator, we define a *divergence measure* between the two tasks below.

***Definiton* 3.** *(Task Divergence Measure) The divergence between a source task distribution p_{S^j} where $j \in \{1, \dots N_S\}$ and the target task distribution p_T is defined as*

$$Div\left(T \,||\, S^j\right) = \mathbb{E}_{(X,Y)\sim p_{S^j}}\left[w_{S^j}^2(X)\cdot L^2(h_\theta(X),Y)\right] - (f_T(\theta))^2 \tag{7}$$

This task divergence measure is large when the corresponding source distribution deviates significantly from the target task distribution. Building on this divergence measure, we define the following estimator for the target task objective.

***Definiton* 4.** *(Variance Reduced Estimator) For a given set of hyperparameters $\theta \in \Theta$, the variance reduced estimator for the target task objective function is defined as*

$$\hat{f}_{VR}\left(\theta; \{\mathcal{D}_{S^j}\}_{j=1}^{N_S}\right) = -\sum_{j=1}^{N_S}\lambda_j^\star \sum_{i=1}^{n_{S^j}} w(x_i^j)\cdot L(h_\theta(x_i^j), y_i^j) \tag{8}$$

where VR stands for variance reduced, $\mathcal{D}_{S^j}$ is any sample size n_{S^j} of the i.i.d. samples from the distribution of source task j. $\lambda_j^\star$ is a weight for source task j, which is defined as

$$\lambda_j^\star = \left(Div\left(T \,||\, S^j\right)\sum_{j=1}^{N_S}\frac{n_{S^j}}{Div\left(T \,||\, S^j\right)}\right)^{-1}$$

Note that, for all $j \in \{1, \dots N_S\}$, $\lambda_j^\star \geq 0$ and $\sum_{j=1}^{N_S}\lambda_j^\star n_{S^j} = 1$.

The variance reduced estimator in Eq. (8) is also statistically unbiased for the ground-truth target task objective in Eq. (2), i.e., $\mathbb{E}\left[\hat{f}_{VR}\left(\theta; \{\mathcal{D}_{S^j}\}_{j=1}^{N_S}\right)\right] = f_T(\theta)$.

Then, we demonstrate that the variance reduced estimator in Eq. (8) is optimal in the sense that any other convex combination of a set of weights $\boldsymbol{\lambda} = \{\lambda_1, \dots \lambda_{N_S}\}$ that satisfies the unbiasedness for the target task objective function does not provide a smaller variance.

Algorithm 1 Hyperparameter optimization procedure under the MSU-HPO setting

Input: unlabeled target task dataset $\mathcal{D}_T = \{x_i\}_{i=1}^{n_T}$; labeled source task datasets $\{\mathcal{D}_{S^j} = \{x_i^j, y_i^j\}_{i=1}^{n_{S^j}}\}_{j=1}^{N_S}$; hyperparameter search space Θ; a machine learning model h_θ; a target task objective estimator $\hat{f}$, a hyperparameter optimization algorithm **OPT**

1: **for** $j \in \{1, \ldots, N_S\}$ **do**
2: Split $\mathcal{D}_{S^j}$ into three folds $\mathcal{D}_{S^j}^{density}$, $\mathcal{D}_{S^j}^{train}$, and $\mathcal{D}_{S^j}^{val}$
3: Estimate density ratio $w_{S^j}(\cdot)$ with $\mathcal{D}_T$ and $\mathcal{D}_{S^j}^{density}$
4: **end for**
5: Optimize the hyperparameter $\theta \in \Theta$ of h_θ with **OPT** by setting $\hat{f}(\theta; \{\mathcal{D}_{S^j}^{val}\}_{j=1}^{N_S})$ as its objective (the model parameter of h_θ is obtained by optimizing $\hat{f}(\theta; \{\mathcal{D}_{S^j}^{train}\}_{j=1}^{N_S})$)
6: **return** $h_{\theta^\star}$ (where $\theta^\star$ is the output of **OPT**)

***Theorem* 1.** *(Variance Optimality) For any given set of weights* $\boldsymbol{\lambda} = \{\lambda_1, \ldots \lambda_{N_S}\}$ *that satisfies* $\lambda_j \geq 0$ *and* $\sum_{j=1}^{N_S} \lambda_j n_{S^j} = 1$ *for all* $j \in \{1, \ldots N_S\}$*, the following inequality holds*

$$\mathbb{V}\left(\hat{f}_{VR}\left(\theta; \{\mathcal{D}_{S^j}\}_{j=1}^{N_S}\right)\right) \leq \mathbb{V}\left(\hat{f}_{\boldsymbol{\lambda}}\left(\theta; \{\mathcal{D}_{S^j}\}_{j=1}^{N_S}\right)\right)$$

where $\hat{f}_{\boldsymbol{\lambda}}(\theta; \{\mathcal{D}_{S^j}\}_{j=1}^{N_S}) = -\sum_{j=1}^{N_S} \lambda_j \sum_{i=1}^{n_{S^j}} w(x_i^j) \cdot L(h_\theta(x_i^j), y_i^j)$*. See Appendix B for the proof.*

Theorem 1 suggests that the variance reduced estimator achieves a desirable finite sample variance property by weighting each source task based on its divergence to the target task.

We summarize the high-level HPO procedure in Algorithm 1. We provide some details of our procedure in Appendix C. We also present the regret analysis in Appendix D.

4. Experiment

4.1 Setup

We investigate the behavior of our method using Parkinson's telemonitoring dataset[1] (Tsanas et al., 2009), which consists of voice measurements collected by using a telemonitoring device for 42 patients with Parkinson disease. Each patient has about 150 recordings characterized by a feature with 17 dimensions. The goal is to predict the Parkinson disease symptom score for each recording. We use *support vector machine* (SVM) implemented in *scikit-learn* (Pedregosa et al., 2011) as a ML model and tune the kernel coefficient $\gamma \in [10^{-6}, 10^6]$ of its RBF kernel and regularization parameter $C \in [10^{-6}, 10^6]$ using HPO methods.

To create the MSU-HPO setting, we treat each patient as a task. We select one patient as a target task and regard the remaining patients as multiple source tasks. Then, the experimental procedure is as follows. (1) Tune hyperparameters of SVM by a HPO method using the unlabeled target task and labeled source tasks, (2) Split the original target task data into 70% training set and 30% test set, (3) Train the tuned SVM model using the training set of the target task, (4) Predict the symptom score on the test set of the target task, (5) Calculate *mean-absolute-error* (MAE) of the prediction and regard it as the performance

1. We present the result with a synthetic data in Appendix E

of the MSU-HPO method under consideration, (6) Repeat the above steps 10 times with different random seeds and report the mean, standard error, and worst-case performances over the simulations.

To the best of our knowledge, there is no existing HPO method for MSU-HPO, and thus, we use the following possible estimators as baselines: (i) **Naive**: this estimator uses the concatenation of source tasks to calculate a validation score and ignores the distributional shift, (ii) **Oracle**: this estimator uses the labeled target task to calculate a validation score. Thus, this is infeasible in MSU-HPO, and we report its performance as a reference.

Table 1: Comparing different MSU-HPO methods

Estimators	Mean	Standard error	Worst Case
Naive	2.6183	**0.1325**	3.2700
Unbiased (ours)	1.3564	0.4591	3.8028
Variance reduced (ours)	**1.0870**	0.3507	**3.0596**
Oracle (reference)	0.0563	0.0015	0.0648

Notes: The table present the prediction performance of SVM tuned by MSU-HPO with each estimator. The bold fonts represent the best performance among estimators using only the unlabeled target task and labeled source task datasets. The mean, standard error, and worst-case performances are induced by running the simulations 10 times with different random seeds.

4.2 Results

Table 1 shows the prediction performance (MAE) of SVM tuned by setting each estimator as $\hat{f}$ in Algorithm 1. Note that we use GP-UCB (Srinivas et al., 2010) as **OPT** for all estimators.

First, the proposed estimators significantly outperform the naive estimator in mean performance because they can correctly address the distributional difference among patients. However, the unbiased estimator underperforms the naive one in the worst-case performance and has the largest standard error, because it suffers from the variance and instability issues. On the other hand, the variance reduced method performs the best in the mean and worst-case performances and has a smaller standard error than the unbiased method. This is because it can increase the stability by down-weighting harmful data when calculating a validation score, as discussed in our theoretical analysis. Finally, the performance of the oracle method indicates that there is room for further improvement, even though our methods largely outperform the naive method.

5. Conclusion

We explored a novel problem setting, MSU-HPO, with the goal of enabling HPO for a new task of interest (target task). To this end, we proposed two estimators to approximate the target task objective function using only available data. In particular, the variance reduced estimator achieves *variance optimality* building the *task divergence measure*. The empirical evaluation demonstrated that it helps us determine useful hyperparameters, even when the labels of the target task are unusable.

References

James S Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for Hyper-Parameter Optimization. In *Advances in neural information processing systems*, pages 2546–2554, 2011.

Edwin V Bonilla, Kian M Chai, and Christopher Williams. Multi-task Gaussian Process Prediction. In *Advances in neural information processing systems*, pages 153–160, 2008.

Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. Are we really making much progress? a worrying analysis of recent neural recommendation approaches. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 101–109, 2019.

Víctor Elvira, Luca Martino, David Luengo, and Mónica F Bugallo. Efficient multiple importance sampling estimators. *IEEE Signal Processing Letters*, 22(10):1757–1761, 2015.

Matthias Feurer and Frank Hutter. Hyperparameter Optimization. In *Automated Machine Learning*, pages 3–33. 2019.

Matthias Feurer, Jost Tobias Springenberg, and Frank Hutter. Initializing Bayesian Hyperparameter Optimization via Meta-learning. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

Matthias Feurer, Benjamin Letham, and Eytan Bakshy. Scalable Meta-Learning for Bayesian Optimization using Ranking-Weighted Gaussian Process Ensembles. In *AutoML Workshop at ICML*, 2018.

Peter I Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep Reinforcement Learning that Matters. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Takafumi Kanamori, Shohei Hido, and Masashi Sugiyama. A least-squares approach to direct importance estimation. *Journal of Machine Learning Research*, 10(Jul):1391–1445, 2009.

Ho Chung Law, Peilin Zhao, Leung Sing Chan, Junzhou Huang, and Dino Sejdinovic. Hyperparameter learning via distributional transfer. In *Advances in Neural Information Processing Systems*, pages 6801–6812, 2019.

Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are Gans Created Equal? A Large-Scale Study. In *Advances in neural information processing systems*, pages 700–709, 2018.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12: 2825–2830, 2011.

Valerio Perrone, Rodolphe Jenatton, Matthias W Seeger, and Cédric Archambeau. Scalable Hyperparameter Transfer Learning. In *Advances in Neural Information Processing Systems*, pages 6845–6855, 2018.

Anil Ramachandran, Sunil Gupta, Santu Rana, and Svetha Venkatesh. Information-theoretic Transfer Learning framework for Bayesian Optimisation. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 827–842, 2018.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

Jost Tobias Springenberg, Aaron Klein, Stefan Falkner, and Frank Hutter. Bayesian Optimization with Robust Bayesian Neural Networks. In *Advances in Neural Information Processing Systems*, pages 4134–4142, 2016.

Niranjan Srinivas, Andreas Krause, Sham Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: no regret and experimental design. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, pages 1015–1022, 2010.

Masashi Sugiyama, Matthias Krauledat, and Klaus-Robert MÃžller. Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research*, 8 (May):985–1005, 2007.

Kevin Swersky, Jasper Snoek, and Ryan P Adams. Multi-Task Bayesian Optimization. In *Advances in neural information processing systems*, pages 2004–2012, 2013.

Athanasios Tsanas, Max A Little, Patrick E McSharry, and Lorraine O Ramig. Accurate telemonitoring of parkinson's disease progression by noninvasive speech tests. *IEEE transactions on Biomedical Engineering*, 57(4):884–893, 2009.

Joaquin Vanschoren. Meta-Learning. In *Automated Machine Learning*, pages 35–61. 2019.

Makoto Yamada, Taiji Suzuki, Takafumi Kanamori, Hirotaka Hachiya, and Masashi Sugiyama. Relative density-ratio estimation for robust distribution comparison. In *Advances in neural information processing systems*, pages 594–602, 2011.

Kaichao You, Ximei Wang, Mingsheng Long, and Michael Jordan. Towards accurate model selection in deep unsupervised domain adaptation. In *International Conference on Machine Learning*, pages 7124–7133, 2019.

Erheng Zhong, Wei Fan, Qiang Yang, Olivier Verscheure, and Jiangtao Ren. Cross validation framework to choose amongst models and datasets for transfer learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 547–562. Springer, 2010.

Appendix A. Summary of Related Work and Contributions

We summarize the related literature and our main contributions below.

A.1 Related Work

A typical HPO setting is to find a better set of hyperparameters using a labeled target task of interest. As faster convergence is an essential performance metric of the HPO methods, the research community is moving on to the *multi-source* or *transfer* settings for which there are some previously solved related source tasks. By combining the additional source task information and the labeled target task dataset, it has been shown that one can improve the hyperparameter search efficiency, and thus reach a better solution with fewer evaluations (Bonilla et al., 2008; Feurer et al., 2018, 2015; Perrone et al., 2018; Ramachandran et al., 2018; Springenberg et al., 2016; Swersky et al., 2013; Vanschoren, 2019). A critical difference between the multi-source (or transfer) HPOs and our MSU-HPO settings is the **existence of labels for the target task**. Previous studies assume that analysts can utilize labeled target data, but as discussed above, this is often unavailable, and thus, these methods are infeasible. This work enables the use of any HPO method in the absence of a labeled target dataset in a theoretically grounded manner for the first time.

Another related field is the *model evaluation in covariate shift* literature, whose objective is to evaluate the performance of the ML models of the target task using only a relevant **single** source dataset (Sugiyama et al., 2007; You et al., 2019; Zhong et al., 2010). These studies build on the *importance sampling* (IS) method (Elvira et al., 2015; Sugiyama et al., 2007) to obtain an unbiased estimate of ground-truth model performances. While our proposed methods are also based on IS, a major difference is that we assume that there are multiple source datasets with different distributions. We will demonstrate that with the multi-source setting, the previous IS method can fail, and propose an estimator satisfying the optimal variance property. Moreover, as these methods are specific to *model evaluation*, the connection between the IS-based estimation techniques and the automated HPO methods has not yet been explored. Consequently, we are the first to empirically evaluate the possible combination of these fields.

A.2 Contributions

The contributions of this work can be summarized as follows:

- We formulate a novel and highly practical HPO setting, MSU-HPO.
- We propose two unbiased estimators for the ground-truth validation score calculable with the available data. Additionally, we demonstrate that one of them achieves optimal finite variance among a reasonable class of unbiased estimators.
- We describe a flexible and computationally tractable HPO procedure building on the proposed estimators.
- We empirically demonstrate that the proposed procedure works favorably in situations where a labeled target dataset is not available. Furthermore, our empirical results

suggest a new possible connection between the adaptive HPO and IS-based unbiased estimation techniques.

These theoretical and empirical findings provide ML practitioners with guidelines on how to optimize the hyperparameters of their ML models, even in situations where they do not have the labels of the target task.

Appendix B. Omited Proofs

B.1 Derivation of Unbiasedness

We first define a general class of unbiased estimators called $\boldsymbol{\lambda}$-*unbiased estimator* that includes the unbiased and variance reduced estimators as special cases.

***Definiton* 5.** *($\boldsymbol{\lambda}$-unbiased Estimator) When a set of weights $\boldsymbol{\lambda} = \{\lambda_1, \dots \lambda_{N_S}\}$ that satisfies $\lambda_j \geq 0$ and $\sum_{j=1}^{N_S} \lambda_j n_{S^j} = 1$ for all $j \in \{1, \dots N_S\}$ is given, the $\boldsymbol{\lambda}$-unbiased estimator for the target task objective function is*

$$\hat{f}_{\boldsymbol{\lambda}}\left(\theta; \{\mathcal{D}_{S^j}\}_{j=1}^{N_S}\right) = -\sum_{j=1}^{N_S} \lambda_j \sum_{i=1}^{n_{S^j}} w_{S^j}(x_i^j) \cdot L(h_\theta(x_i^j), y_i^j) \tag{9}$$

when $\lambda_j = n_{S^j}/N$, it is the unbiased estimator in Eq. (5). In contrast, it is the variance reduced estimator in Eq. (8) when $\lambda_j = \lambda_j^\star$

Then we show that the $\boldsymbol{\lambda}$-unbiased estimator is statistically unbiased for the target task function.

Proof. By the linearity of the expectation operator,

$$\begin{aligned}
\mathbb{E}\left[\hat{f}_{\boldsymbol{\lambda}}\left(\theta; \{\mathcal{D}_{S^j}\}_{j=1}^{N_S}\right)\right] &= -\sum_{j=1}^{N_S} \lambda_j \sum_{i=1}^{n_{S^j}} \mathbb{E}_{(X,Y)\sim p_{S^j}}\left[w_{S^j}(X) \cdot L(h_\theta(X), Y)\right] \\
&= -\sum_{j=1}^{N_S} \lambda_j \sum_{i=1}^{n_{S^j}} \mathbb{E}_{(X,Y)\sim p_{S^j}}\left[\frac{p_T(X,Y)}{p_{S^j}(X,Y)} \cdot L(h_\theta(X), Y)\right] \\
&= \sum_{j=1}^{N_S} \lambda_j \sum_{i=1}^{n_{S^j}} -\mathbb{E}_{(X,Y)\sim p_T}\left[L(h_\theta(X), Y)\right] \\
&= \sum_{j=1}^{N_S} \lambda_j \sum_{i=1}^{n_{S^j}} f_T(\theta) \\
&= \left(\sum_{j=1}^{N_S} \lambda_j n_{S^j}\right) \cdot f_T(\theta) \\
&= f_T(\theta)
\end{aligned}$$

Thus, the unbiased estimator in Eq. (5) and the variance reduced estimator in Eq. (8) are both statistically unbiased for the ground truth target task objective function in Eq. (2). □

B.2 Derivation of Eq. (6)

Proof. The variance can be represented as follows because samples are independent

$$\begin{aligned}\mathbb{V}\left(\hat{f}_{UB}\left(\theta;\{\mathcal{D}_{S^j}\}_{j=1}^{N_S}\right)\right) &= \frac{1}{n^2}\sum_{j=1}^{N_S}\sum_{i=1}^{n_{Sj}}\mathbb{V}\left(w_{S^j}(X)\cdot L(h_\theta(X),Y)\right)\\ &= \frac{1}{n^2}\sum_{j=1}^{N_S} n_{S^j}\cdot\mathbb{V}\left(w_{S^j}(X)\cdot L(h_\theta(X),Y)\right)\end{aligned}$$

$\mathbb{V}\left(w_{S^j}(X)\cdot L(h_\theta(X),Y)\right)$ is decomposed as

$$\mathbb{V}\left(w_{S^j}(X)\cdot L(h_\theta(X),Y)\right) = \mathbb{E}_{(X,Y)\sim p_{S^j}}\left[w_{S^j}^2(X)\cdot L^2(h_\theta(X),Y)\right] - \left(\mathbb{E}_{(X,Y)\sim p_{S^j}}\left[w_{S^j}(X)\cdot L(h_\theta(X),Y)\right]\right)^2$$

From the unbiasedness property, $\mathbb{E}_{(X,Y)\sim p_{S^j}}\left[w_{S^j}(X)\cdot L(h_\theta(X),Y)\right] = f_T(\theta)$. Then, we now have

$$\mathbb{V}\left(w_{S^j}(X)\cdot L(h_\theta(X),Y)\right) = \mathbb{E}_{(X,Y)\sim p_{S^j}}\left[w_{S^j}^2(X)\cdot L^2(h_\theta(X),Y)\right] - \left(f_T(\theta)\right)^2$$

□

B.3 Proof of Theorem 1

By following the same logic flow as in Section A.2, the variance of the $\boldsymbol{\lambda}$-unbiased estimator in Eq. (9) is

$$\begin{aligned}\mathbb{V}\left(\hat{f}_{\boldsymbol{\lambda}}\left(\theta;\{\mathcal{D}_{S^j}\}_{j=1}^{N_S}\right)\right) &= \sum_{j=1}^{N_S}\lambda_j^2 n_{S^j}\left(\mathbb{E}_{(X,Y)\sim p_{S^j}}\left[w_{S^j}^2(X)\cdot L^2(h_\theta(X),Y)\right] - \left(f_T(\theta)\right)^2\right)\\ &= \sum_{j=1}^{N_S}\lambda_j^2 n_{S^j}\cdot Div\left(T\,||\,S^j\right) \qquad (10)\end{aligned}$$

Thus, by replacing λ_j for $\left(Div\left(T\,||\,S^j\right)\sum_{j=1}^{N_S}\frac{n_{Sj}}{Div(T\,||\,S^j)}\right)^{-1}$, we have

$$\begin{aligned}\mathbb{V}\left(\hat{f}_{\boldsymbol{\lambda}}\left(\theta;\{\mathcal{D}_{S^j}\}_{j=1}^{N_S}\right)\right) &= \sum_{j=1}^{N_S}\left(\sum_{j=1}^{N_S}\frac{n_{S^j}}{Div\left(T\,||\,S^j\right)}\right)^{-2} n_{S^j}\cdot Div\left(T\,||\,S^j\right)\\ &= \sum_{j=1}^{N_S}\frac{n_{S^j}Div\left(T\,||\,S^j\right)}{(Div\left(T\,||\,S^j\right))^2(\sum_{j=1}^{N_S}\frac{n_{Sj}}{Div(T\,||\,S^j)})^2}\\ &= \left(\sum_{j=1}^{N_S}\frac{n_{S^j}}{Div\left(T\,||\,S^j\right)}\right)\left(\sum_{j=1}^{N_S}\frac{n_{S^j}}{Div\left(T\,||\,S^j\right)}\right)^{-2}\\ &= \left(\sum_{j=1}^{N_S}\frac{n_{S^j}}{Div\left(T\,||\,S^j\right)}\right)^{-1}\end{aligned}$$

Algorithm 2 Hyperparameter optimization procedure under the MSU-HPO setting

Input: unlabeled target task dataset $\mathcal{D}_T = \{x_i\}_{i=1}^{n_T}$; labeled source task datasets $\{\mathcal{D}_{S^j} = \{x_i^j, y_i^j\}_{i=1}^{n_{Sj}}\}_{j=1}^{N_S}$; hyperparameter search space Θ; a machine learning model h_θ; a target task objective estimator $\hat{f}$, a hyperparameter optimization algorithm **OPT**

Output: the optimized set of hyperparameters $\theta^\star \in \Theta$

1: **for** $j \in \{1, \ldots, N_S\}$ **do**
2: Split $\mathcal{D}_{S^j}$ into three folds $\mathcal{D}_{S^j}^{density}$, $\mathcal{D}_{S^j}^{train}$, and $\mathcal{D}_{S^j}^{val}$
3: Estimate density ratio $w_{S^j}(\cdot)$ by LSIF with $\mathcal{D}_T$ and $\mathcal{D}_{S^j}^{density}$
4: **end for**
5: Optimize the hyperparameter $\theta \in \Theta$ of h_θ with **OPT** by setting $\hat{f}(\theta; \{\mathcal{D}_{S^j}^{val}\}_{j=1}^{N_S})$ as its objective
6: (the model parameter of h_θ is obtained by optimizing $\hat{f}(\theta; \{\mathcal{D}_{S^j}^{train}\}_{j=1}^{N_S})$)
7: **return** $h_{\theta^\star}$ (where $\theta^\star$ is the output of **OPT**)

Moreover, for any set of weights $\boldsymbol{\lambda} = \{\lambda_1, \ldots \lambda_{N_S}\}$, we obtain the following variance optimality using the Cauchy-Schwarz inequality.

$$\left(\sum_{j=1}^{N_S} \lambda_j^2 n_{S^j} \cdot Div\left(T \,||\, S^j\right)\right)\left(\sum_{j=1}^{N_S} \frac{n_{S^j}}{Div\left(T \,||\, S^j\right)}\right) \geq \left(\sum_{j=1}^{N_S} \lambda_j n_{S^j}\right)^2 = 1$$

$$\Longrightarrow \left(\sum_{j=1}^{N_S} \lambda_j^2 n_{S^j} \cdot Div\left(T \,||\, S^j\right)\right) \geq \left(\sum_{j=1}^{N_S} \frac{n_{S^j}}{Div\left(T \,||\, S^j\right)}\right)^{-1}$$

$$\Longrightarrow \mathbb{V}\left(\hat{f}_{\boldsymbol{\lambda}}\left(\theta; \{\mathcal{D}_{S^j}\}_{j=1}^{N_S}\right)\right) \geq \mathbb{V}\left(\hat{f}_{VR}\left(\theta; \{\mathcal{D}_{S^j}\}_{j=1}^{N_S}\right)\right)$$

Appendix C. Hyperparameter Optimization Procedure

We describe several detailed components of the hyperparameter optimization procedure in the MSU-HPO setting.

Density Ratio Estimation: In general, density ratio functions between the target and source tasks are unavailable, and thus, should be estimated beforehand. To estimate this parameter, we employ the *least-squares importance fitting* procedure (Kanamori et al., 2009; Yamada et al., 2011), which suggests directly minimizing the following squared error for the true density ratio function:

$$\hat{s} = \underset{s \in \mathcal{S}}{\arg\min}\, \mathbb{E}_{p_{S^j}}\left[\left(w(X) - s(X)\right)^2\right] = \underset{s \in \mathcal{S}}{\arg\min}\, \left[\frac{1}{2}\mathbb{E}_{p_{S^j}}\left[s^2(X)\right] - \mathbb{E}_{p_T}[s(X)]\right] \tag{11}$$

where $\mathcal{S}$ is a class of measurable functions $s : \mathcal{X} \to \mathbb{R}_{\geq 0}$. It should be noted that the empirical version of Eq. (11) is calculable with unlabeled target and source task datasets.

Task Divergence Estimation: To utilize the variance reduced estimator, the task divergence measure $Div\left(T \,||\, S^j\right)$ in Eq. (7) needs to be estimated from the available data.

Algorithm 3 Bayesian Optimization under the MSU-HPO setting

Input: unlabeled target task dataset $\mathcal{D}_T = \{x_i\}_{i=1}^{n_T}$; labeled source task datasets $\{\mathcal{D}_{S^j} = \{x_i^j, y_i^j\}_{i=1}^{n_{S^j}}\}_{j=1}^{N_S}$; hyperparameter search space Θ; a machine learning model h_θ; a target task objective estiamtor $\hat{f}$, limit B, acquisition function $\alpha(\cdot)$

Output: the optimized set of hyperparameters $\theta^\star \in \Theta$

Set $\mathcal{A}_0 \leftarrow \emptyset$
for $j \in \{1, \ldots, N_S\}$ **do**
  Split $\mathcal{D}_{S^j}$ into three folds $\mathcal{D}_{S^j}^{density}$, $\mathcal{D}_{S^j}^{train}$, and $\mathcal{D}_{S^j}^{val}$
  Estimate density ratio $w_{S^j}(\cdot)$ by LSIF with $\mathcal{D}_T$ and $\mathcal{D}_{S^j}^{density}$
end for
for $t = 1, 2, \ldots, B$ **do**
  Select $\theta_t = \arg\max_{\theta \in \Theta} \alpha(\theta \,|\, \mathcal{A}_{t-1})$
  Train h_{θ_t} by optimizing $\hat{f}(\theta; \{\mathcal{D}_{S^j}^{train}\}_{j=1}^{N_S})$ and obtain a trained model h_θ^*
  Evaluate h_θ^* and obtain a validation score $z_t = \hat{f}(\theta; \{\mathcal{D}_{S^j}^{val}\}_{j=1}^{N_S})$
  $\mathcal{A}_t \leftarrow \mathcal{A}_{t-1} \cup \{(\theta_t, z_t)\}$
end for
$t^\star = \arg\max_t \{z_1, \ldots z_B\}$
return $h_{\theta^\star}$ (where $\theta^\star = \theta_{t^\star}$)

This can be done using the following empirical mean.

$$\widehat{Div}\left(T \,||\, S^j\right) = \frac{1}{n_{S^j}} \sum_{i=1}^{n_{S^j}} \left(w(x_i^j) \cdot L(h_\theta(x_i^j), y_i^j)\right)^2 - \left(\frac{1}{n_{S^j}} \sum_{i=1}^{n_{S^j}} w(x_i^j) \cdot L(h_\theta(x_i^j), y_i^j)\right)^2 \quad (12)$$

How to train h_θ?: To evaluate the validation score of $\theta \in \Theta$, the model parameters of h_θ should be optimized by the supervised learning procedure. However, in the MSU-HPO setting, the labeled target task dataset is unavailable, and direct training of h_θ is infeasible. Therefore, we suggest splitting the labeled source task datasets $\{\mathcal{D}_{S^j}\}$ into the training $\{\mathcal{D}_{S^j}^{train}\}$ and validation $\{\mathcal{D}_{S^j}^{val}\}$ sets. Then, we can train h_θ using the training set as follows:

$$h_\theta^* = \arg\max_{h_\theta \in \mathcal{H}_\theta} \hat{f}\left(\theta; \{\mathcal{D}_{S^j}^{train}\}_{j=1}^{N_S}\right)$$

where $\hat{f}$ is an estimator for the target task objective function such as the unbiased and variance reduced estimators, and $\mathcal{H}_\theta$ is a hypothesis space defined by a set of hyperparameters $\theta \in \Theta$.

This training procedure enables us to obtain the model parameters of h_θ as if it were trained on the labeled target task dataset. In addition, it is sufficient to train h_θ only once to evaluate $\theta \in \Theta$;the proposed procedure is computationally inexpensive compared with other methods in task transfer settings such as distBO (Law et al., 2019).

Algorithm 2 describes the high-level hyperparameter optimization procedure which allows any black-box optimization method to be used. In addition, Algorithm 3 describes the

hyperparameter optimization procedure under the MSU-HPO setting with the popular Bayesian optimization method.

Appendix D. Regret Analysis

In this section, we analyze the regret bound under the MSU-HPO setting. We define a *regret* as

$$r_B^n = f(\theta^*) - f(\hat{\theta}_B^*),$$

where $f : \Theta \to \mathbb{R}$ is the ground-truth target task objective, $n = \sum_{j=1}^{N_S} n_{S^j}$ is the total sample size among source tasks, B is the total number of evaluation rounds, $\theta^* = \arg\max_\theta f(\theta)$, and $\hat{\theta}_B^* = \arg\max_{\theta \in \{\theta_1, \cdots, \theta_B\}} \hat{f}_n(\theta)$ where $\hat{f}_n : \Theta \to \mathbb{R}$ is a target task objective approximated by any estimator (e.g., the unbiased estimator and the variance reduced estimator).

To bound the regret above, we first decompose it into the following terms:

$$\begin{aligned} r_B^N &= f(\theta^*) - f(\hat{\theta}_B^*) \\ &= (f(\theta^*) - \hat{f}_n(\hat{\theta}^*)) + \hat{f}_n(\hat{\theta}^*) - (f(\hat{\theta}_B^*) - \hat{f}_n(\hat{\theta}_B^*)) - \hat{f}_n(\hat{\theta}_B^*) \\ &= \underbrace{(\hat{f}_n(\hat{\theta}^*) - \hat{f}_n(\hat{\theta}_B^*))}_{(A)} + \underbrace{(f(\hat{\theta}_B^*) - \hat{f}_n(\hat{\theta}_B^*))}_{(B)} + \underbrace{(f(\theta^*) - \hat{f}_n(\hat{\theta}^*))}_{(C)}, \end{aligned} \tag{13}$$

where $\hat{\theta}^* = \arg\max_{\theta \in \Theta} \hat{f}_N(\theta)$.

The term (A) represents the regret obtained by optimizing the approximated target task objective $\hat{f}_n$. The term (B) represents the difference of a function value between the true objective f and the approximated objective $\hat{f}_n$ at $\hat{\theta}_B^*$, which is the solution obtained by the optimization for the approximated objective. The term (C) represents the difference between the maximum value for the true objective f and the maximum value for the approximated objective $\hat{f}_n$. Note that the term (A) depends on a optimizer, not an estimator; in contrast, the term (B) and (C) depend on only an estimator.

We first show the following two lemmas which is used to bound the regret.

***Lemma* 2.** *The following inequality holds with a probability of at least* $1 - \delta$, $\delta \in (0, 1)$

$$(f(\hat{\theta}_B^*) - \hat{f}_n(\hat{\theta}_B^*)) \leq \sqrt{\mathbb{V}(\hat{f}_n(\hat{\theta}_B^*))/\delta}.$$

Proof. By Chebyshev's inequality, we have

$$\begin{aligned} \mathbb{P}\{\hat{f}_N(\hat{\theta}_B^*) - f(\hat{\theta}_B^*) \geq c\} &\leq \mathbb{P}\{|\hat{f}_N(\hat{\theta}_B^*) - f(\hat{\theta}_B^*)| \geq c\} \\ &\leq \mathbb{V}(\hat{f}_N(\hat{\theta}_B^*))/c^2. \end{aligned}$$

Putting the RHS as δ and solving it for c completes the proof. □

***Lemma* 3.** *The following inequality holds with a probability of at least* $1 - \delta$, $\delta \in (0, 1)$

$$|f(\theta^*) - \hat{f}_n(\hat{\theta}^*)| \leq \sqrt{2(\mathbb{V}(\hat{f}_n(\theta^*)) + \mathbb{V}(\hat{f}_n(\hat{\theta}^*)))/\delta}.$$

Proof. By Chebyshev's inequality, we have

$$\begin{aligned}
&\mathbb{P}\{f(\theta^*) - \hat{f}_n(\hat{\theta}^*) \geq c\} \\
&\leq \mathbb{P}\{|f(\theta^*) - \hat{f}_n(\hat{\theta}^*)| \geq c\} \\
&\leq \mathbb{P}\{|f(\theta^*) - \hat{f}_n(\theta^*)| \geq c \cup |f(\hat{\theta}^*) - \hat{f}_n(\hat{\theta}^*)| \geq c\} \\
&\leq \mathbb{P}\{|f(\theta^*) - \hat{f}_n(\theta^*)| \geq c\} + \mathbb{P}\{|f(\hat{\theta}^*) - \hat{f}_n(\hat{\theta}^*)| \geq c\} \\
&\leq \frac{1}{c^2}(\mathbb{V}(\hat{f}_n(\theta^*)) + \mathbb{V}(\hat{f}_n(\hat{\theta}^*))).
\end{aligned}$$

Putting the RHS as δ and solving it for c completes the proof. □

***Theorem* 4.** *(Regret Bound on the MSU-HPO setting) When the λ-unbiased estimator with an arbitrary set of weights $\boldsymbol{\lambda}$ is used as $\hat{f}(\theta, ; \{\mathcal{D}^{val}_{S^j}\}_{j=1}^{N_S})$, the following regret bound holds with a probability of at least $1-\delta$, $\delta \in (0,1)$,*

$$r_B^N \leq R_n + \sqrt{2\mathbb{V}(\hat{f}_n(\hat{\theta}^*_B))/\delta} + \sqrt{2(\mathbb{V}(\hat{f}_n(\theta^*)) + \mathbb{V}(\hat{f}_n(\hat{\theta}^*)))/\delta}, \tag{14}$$

where $R_n = (\hat{f}_n(\hat{\theta}^*) - \hat{f}_n(\hat{\theta}^*_B))$.

Proof. Putting Lemma 2 to the term (B) in Eq. 13 and Lemma 3 to the term (B) in Eq. 13 complete the proof. □

Remark. When an estimator is the proposed unbiased estimator or variance reduced esitmator, the variance $\mathbb{V}(\hat{f}_n(\cdot))$ is $o(n)$; the second term and third term in Eq.(14) is to be no-regret with respect to n: $\lim_{n\to\infty}(\sqrt{2\mathbb{V}(\hat{f}_n(\hat{\theta}^*_B))/\delta} + \sqrt{2(\mathbb{V}(\hat{f}_n(\theta^*)) + \mathbb{V}(\hat{f}_n(\hat{\theta}^*)))/\delta})/n = 0$. That is, when we use a no-regret optimizer with respect to B, such as GP-UCB (Srinivas et al., 2010), the regret overall is to be no-regret[2] with respect to n and B.

Appendix E. Experimental Results with a Toy Problem

E.1 Setup

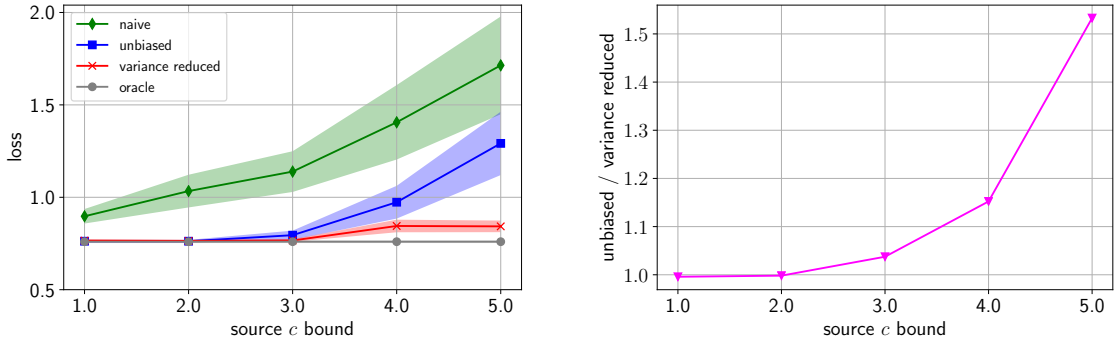
We consider a 1-dimensional regression problem with the MSU-HPO setting. The generative process of the dataset in this experiment is as follows:

$$\mu^i \sim \mathcal{U}(-c_i, c_i), \;\; \{x_l^i\}_{l=1}^n \mid \mu^i \overset{i.i.d.}{\sim} \mathcal{N}(\mu^i, 1), \;\; \{y_l^i\}_{l=1}^n \mid \{x_l^i\}_{l=1}^n \overset{i.i.d.}{\sim} \{\mathcal{N}(0.7x_l^i + 0.3, 1)\}_{l=1}^n,$$

where $\mathcal{U}$ is the uniform distribution, $\mathcal{N}$ denotes the normal distribution, and $c_i \in \mathbb{R}$ is a prior parameter that characterizes the marginal input distribution ($p(x)$) of task i. The objective function f is given by:

$$f(\theta; \mathcal{D}_i) = \frac{1}{N}\sum_{l=1}^{N} L(\theta, y_i), \;\; L(\theta, y_i) = (\theta - y_i)^2/2. \tag{15}$$

2. In this case, we need some assumption on the objective function. For example, the regret bound of GP-UCB depends on a RKHS norm or a smoothness of a kernel of the objective function.



(a) Comparing all methods

(b) Comapring unbiased and variance reduced

Figure 1: Results of the experiment on synthetic toy problems over 30 runs.

The optimal solution for this experiment is $\theta = n^{-1} \sum_{l=1}^{n} y_l$.

As discussed in our theoretical analysis, when $p(x)$ of the source task and the target task differs significantly, the performance of the variance reduced estimator is better than that of the unbiased estimator. To demonstrate this, we set c_i separately for the source ($c_i^S \in \{1.0, 2.0, \cdots, 5.0\}, i \in \{1, \cdots, N_S\}$) and the target tasks ($c^T = 1.0$). That is, the source and target distributions are similar when $c_i^S = 1.0 (= c^T)$; in contrast, the source and target distributions are quite different when $c_i^S = 5.0$. Finally, we set $N^S = 2$ and $n = 1000$.

E.2 Results

Figure 1 shows the results of the experiment on the toy problem over 30 runs with different random seeds. Figure 1 (a) indicates that the proposed unbiased and variance reduced estimators significantly outperform the naive method in all settings. This is because our estimators can unbiasedly approximate the target task objective by considering the distributional shift, while the naive method cannot. Moreover, this figure highlights the advantage of unbiasedness when the distributions of the target and source tasks diverge largely (i.e., when c_i^S is large.). Next, we compare the performance of the unbiased and variance reduced estimator in Figure 1 (b). This reports the performance of the unbiased estimator relative to the variance reduced one with varying values of c. The result indicates that the advantages of using the variance reduced estimator over the unbiased one are further strengthened when there is a large divergence between the target and source task distributions. Finally, as shown in Figure 1 (a), the variance reduced estimator achieves almost the same performance as the upper bound without using the labels of the target task, and this suggests the importance of the *variance optimality* proven in Theorem 1.