# META-LEARNING FOR UNSUPERVISED OUTLIER DETECTION WITH OPTIMAL TRANSPORT

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Automated machine learning has been widely researched and adopted in the field of supervised classification and regression, but progress in unsupervised settings has been limited. We propose a novel approach to automate outlier detection based on meta-learning from previous datasets with outliers. Our premise is that the selection of the optimal outlier detection technique depends on the inherent properties of the data distribution. We leverage optimal transport in particular, to find the dataset with the most similar underlying distribution, and then apply the outlier detection techniques that proved to work best for that data distribution. We evaluate the robustness of our approach and find that it outperforms the state of the art methods in unsupervised outlier detection. This approach can also be easily generalized to automate other unsupervised settings.

## 1 INTRODUCTION

Outlier detection(OD) is the process of identifying data points that are significantly different from the rest of the data. These data points can be caused by errors in the data collection process, incorrect values, or unusual events. Outlier detection can be used to improve the quality of the data or to find unusual events that could be interesting to different business and scientific domains . The term ”outlier detection” can be interchangeably used with ”anomaly detection”. For consistency, we will use the term ”outlier detection” in this paper. Outlier detection has multiple applications such as medicine (Chow & keung Tse, 1990; Ma et al., 2021b), chemistry (Egan & Morgan, 1998) and molecular biology (Cho & Eo, 2016). Outlier detection has been a particularly hard problem. A number of Outlier detection algorithms have been introduced in the last two decades (Aggarwal, 2013). Unsupervised outlier detection is a very challenging task with no universally good model which works optimally on every task (Campos et al., 2015).

AutoML (Hutter et al., 2019) has shown reliable performance and benefits in model selection and hyperparameter optimization (Hutter et al., 2019; Feurer et al., 2015; Thornton et al., 2013). The research in Automated machine learning has been highly focused on supervised machine learning where we can focus on the performance on the hold-out dataset to define an optimization metric for the search algorithm which finds the optimal algorithms by iterating over the search space. This setting is very reliable (Feurer et al., 2015) but the research on unsupervised setting is rather limited. In recent years frameworks like MetaOD (Zhao et al., 2021) have appeared which attempt to solve automated outlier detection via meta-learning (Vanschoren, 2018).

In this work we propose an automated framework for unsupervised machine learning tasks **LOTUS**(Learning to learn with Optimal Transport for Unsupervised Scenarios), which leverages meta-learning (Vanschoren, 2018) and optimal transport distances (Peyré & Cuturi, 2019; Scetbon & Cuturi, 2022). In this work we use LOTUS to perform model selection on a given unsupervised outlier detection task. In summary, we make the following 4 contributions:

- **A Meta-learner for outlier detection**: We propose **LOTUS**: Learning to learn with Optimal Transport for Unsupervised Scenarios, an optimal transport based meta-learner which recommends an optimal outlier detection algorithm based on a historical collection of datasets and models in a zero-shot learning scenario. Our solution can be used in cold start settings for model selection on unsupervised outlier detection.

- **Experiments and results:** We empirically evaluate LOTUS in combination with existing state of the art methods. We demonstrate the robustness of our approach against existing state of the art meta-learners and learners.
- **Open source:** We open-source the code for LOTUS for researchers to use and reproduce our experiments. Our tools can be extended with new datasets and algorithms.

## 2 Background

This section describes related work regarding Automated Machine learning for unsupervised outlier detection, optimal transport and meta-learning.

### 2.1 AUTOML FOR OUTLIER DETECTION

AutoML (Hutter et al., 2019) for unsupervised outlier detection is an extremely hard problem due lack of an optimization metric to perform algorithm selection. One can argue that the use of internal metrics like Excess-Mass (Goix, 2016), Mass-Volume (Goix, 2016) and IREOS (Marques et al., 2015) can make algorithm selection possible. Ma et al. (2021a) shows in their experiments that these internal metrics are computationally very expensive and do not scale well for large datasets. This makes it unfeasible to use these metrics in AutoML tools for most real world scenarios.

There has been recent research on AutoML for outlier detection. PyODDS and MetaOD (Li et al., 2020; Zhao et al., 2021) are among the few tools which have been shown to automate outlier detection.

To the best of our knowledge MetaOD (Zhao et al., 2021) is the current state of the art meta-learner for model selection on outlier detection tasks for tabular data. MetaOD uses meta-learning as a recommendation engine using landmark meta-features and model based meta-features with collaborative filtering (Stern et al., 2010) to perform model selection for a given task.

### 2.2 META LEARNING

Meta-learning or *learning to learn* in AutoML (Vanschoren) is the study of learning from historical performances of machine learning models on a variety of tasks and using this knowledge to find better models for new tasks. Meta-learning can help to speed up the model selection process and find better architectures. Meta-learning is often proposed as a solution to *cold start problem*, by initializing the hyperparameters or search space for the AutoML algorithm. This is often called *warm-starting* for AutoML.

**Meta-learning in existing AutoML tools:** Different AutoML tools use different meta-learning schemes to solve this cold start problem. AutoSklearn-2.0 (Feurer et al., 2020) learns pipeline portfolios, MetaOD (Zhao et al., 2021) trains a collaborative filtering based algorithm (Stern et al., 2010) with landmark-based and model-based metafeatures (Castiello et al., 2005), FLAML (Wang et al., 2021) uses in-built meta-learned defaults for warm starting. MetaBu (Rakotoarison et al., 2022) uses Fused Gromov Wasserstein with proximal gradient method on landmark meta-features for warm-starting AutoSklearn (More discussion about LOTUS vs MetaBu is provided in the section 2.4.2).

### 2.3 OPTIMAL TRANSPORT AND DATASET DISTANCES

Optimal transport(OT) theory deals with the problem of finding an optimal transport map between two probability measures, often on different metric spaces. It is closely related to Monge's problem (Villani, 2008), in which one searches for the optimal transport map between two given measures.

An Optimal transport problem consists of minimizing the cost of transporting mass from one distribution to another. For cost function(ground metric) between pair of points, we calculate the cost matrix $C$ with dimensionality $n \times m$, the OT problem minimizes the loss function $L_c(P) := \langle C, P \rangle$ w.r.t a coupling matrix $P$. Most common approach with practitioners is to use a regularized approach which is computationally more efficient $L_c^{\epsilon}(P) := \langle C, P \rangle + \epsilon r(P)$ where r is negative entropy sinkhorn algorithm (Cuturi, 2013) which is computationally more efficient. A discrete OT problem can be defined with two finite pointclouds, $\{x^{(i)}\}_{i=1}^{n}$ ,$\{y^{(j)}\}_{j=1}^{m}, x^{(i)}, y^{(j)} \in \mathbb{R}^d$, which can be described as two empirical distributions: $\mu := \sum_{i=1}^{n} a_i \delta_{x^{(i)}}, \nu := \sum_{j=1}^{m} b_j \delta_{y^{(j)}}$. Here $a$ and $b$ are the probability vectors of size $n$ and $m$. In this work we are interested in the the Gromov Wasserstein(GW) distance between these two discrete probability distributions. Gromomv Wasserstein allows us to match points taken within different metric spaces. This problem can be written as a function of $(a, A), (b, B)$ between our distributions $A$ and $B$ (Villani, 2008; Scetbon et al., 2022):

$$\text{GW}((a, A), (b, B)) = \min_{P \in \Pi_{a,b}} \mathcal{Q}_{A,B}(P) \tag{1}$$

where $\Pi_{a,b} := \{P \in \mathbb{R}_{+}^{n \times m} | P\mathbf{1}_m = a, P^T \mathbf{1}_n = b\}$

the energy $\mathcal{Q}_{A,B}$ is a quadratic function of $P$ which can be described as

$$\mathcal{Q}_{A,B}(P) := \sum_{i,j,i',j'} (A_{i,i'} - B_{j,j'})^2 P_{i,j} P_{i',j'} \tag{2}$$

In this work we are interested in the Entropic Gromov Wasserstein cost (Peyré et al., 2016):

$$\text{GW}_{\varepsilon}((a, A), (b, B)) = \min_{P \in \Pi_{a,b}} \mathcal{Q}_{A,B}(P) -_{\varepsilon} H(P) \tag{3}$$

where $GW_{\epsilon}$ is the Entropic Gromov Wasserstein cost between our distributions $A$ and $B$, and $\varepsilon H(P)$ is the shannon entropy. The problem with Gromov Wasserstein is that it is NP-hard and the entropic approximation of GW still has cubic complexity. To speed up the computations and use it in a realistic AutoML settings we use the Low-Rank Gromov Wasserstein (GW-LR) approximation (Scetbon et al., 2021; Scetbon & Cuturi, 2022; Scetbon et al., 2022), which reduces the computational cost from cubic to linear time. Scetbon et al. (2022) consider the GW problem with low-rank couplings, linked by a common marginal $g$. Therefore, the set of possible transport plans is restricted to those adopting the factorization of the form $P_r = Qdiag(1/g)R^T$. In this form $Q$ and $R$ are thin matrices with dimensionality of $n \times r$, $r \times m$ respectively and $g$ is a $r-$ dimensional probability vector. The GW-LR distance is be described as:

$$\text{GW-LR}^{(r)}((a, A), (b, B)) := \min_{(Q,R,g) \in \mathcal{C}_{a,b,r}} \mathcal{Q}_{A,B}(Qdiag(1/g)R^T) \tag{4}$$

Our primary inspiration for LOTUS comes from two different works.

1. Alvarez-Melis & Fusi (2020) proposes optimal transport dataset distance(OTDD) which uses optimal transport to learn a mapping over the joint feature and label spaces. Alvarez-Melis & Fusi (2020) proposed that optimal transport distances can be used as a similarity metric between different datasets from different domains and subdomains.
2. Work of Nies et al. (2021) argues that optimal transport measures can be used as a correlation measure between two random variables via transport dependency.

There have been other studies exploring the space of dataset and task similarity with distance measures. Gao & Chaudhari (2021) proposes “coupled transfer distance” which utilises optimal transport distances as a transfer learning distance metric. Achille et al. (2021) explores connections between Deep Learning, Complexity Theory, and Information Theory through their proposed asymmetric distance on tasks.

### 2.4 RELATED WORKS

In this section we will discuss the difference between closest approaches to LOTUS which are MetaOD and MetaBu. We have also added Table 1 to show how LOTUS differs from other meta-learning approaches.

#### 2.4.1 LOTUS VS METAOD

LOTUS and MetaOD solve the same problem of model selection problem for unsupervised outlier detection. The major difference in LOTUS and MetaOD is meta-feature generation. LOTUS aims to

| Technique | Meta-learning approach | Unsupervised Tasks | Use |
|---|---|---|---|
| MetaOD (Zhao et al., 2021) | Metafeatures + CF | Outlier detection only | model selection |
| MetaBu (Rakotoarison et al., 2022) | Supervised metafeatures +FusedGW | ✗ | warm-starting |
| AutoSklearn 2.0 (Feurer et al., 2020) | Pipeline Portfolios | ✗ | warm-starting |
| FLAML (Wang et al., 2021) | Built-in metafeatures | ✗ | warm-starting |
| **LOTUS (Ours)** | Transformation+GWLR | ✓ | model selection |

Table 1: Comparison of different meta-learning frameworks

capture the similarity of the given source and target representations via optimal transport. MetaOD captures similarity with a combination of landmark-features and model-based features and uses a rank-based criteria called discounted cumulative gain for model selection. MetaOD also uses stochastic algorithms such as Isolation Forest and LODA for model-based meta-feature generation which means that the absolute dataset similarity and ranking can differ based on the number of runs. Our approach generalises better than MetaOD as well for different unsupervised tasks as it aims to find similar dataset independent of task, whereas MetaOD's similarity is highly coupled with the task of outlier detection.

### 2.4.2 LOTUS VS METABU

MetaBu (Rakotoarison et al., 2022) was proposed as a solution to cold start problem in supervised learning scenario. Rakotoarison et al. (2022) uses Fused-Gromov-Wasserstein distance with multi dimensional scaling (Cox & Cox, 2008) by first extracting meta-features from the target representation and source representation and proximal gradient method (Xu et al., 2020). LOTUS is a solution for unsupervised setting whereas MetaBu relies on landmark features from PyMFE (Alcobaça et al., 2020) which are more reliable for datasets with labels. Similar to MetaOD, **MetaBu setting is limited to only one task (supervised classification) as it relies on landmark-features which require labels.** MetaBu is used for warm starting not selecting the best pipline in a zero shot setting.

## 3 Methodology

We introduce LOTUS, Learning to learn with Optimal Transport for Unsupervised Scenario. LOTUS meta-learns how well different unsupervised algorithms work on prior *labeled* datasets. These can be datasets where the correct labels are known, or proxy tasks. For instance, for outlier detection we can use extremely imbalanced classification tasks where examples of the smallest class are considered outliers. More formally, we require:

- A collection of $n$ prior labeled datasets $\mathcal{D}_{meta} = \{D_1, ..., D_n\}$ with test and train splits such that $D_i = (X_i^{train}, y_i^{train})$.
- A collection of $n$ optimized algorithms $A_i^*$ with associated tuned hyperparameters $\lambda_i^*$ (Use of $*$ indicates tuned version of model/hyperparameters)for every dataset in $\mathcal{D}_{meta}$; $\mathcal{A} = \{A^*_{\lambda_1^*}, ..., A^*_{\lambda_n^*}\}$

### 3.1 META-TRAINING

In this section, we formally describe the problem of model selection for unsupervised outlier detection.
**Problem Statement:** Given a new dataset without any labels, our meta-learner needs to selects an optimal algorithm with associated hyperparameters from a collection of previously evaluated pipeline. In this setting, we cannot optimize the given model for the dataset as there are no given labels. This problem becomes from a Combined model selection and hyperparameter optimization problem to a *zero-shot model reccomendation problem.*
Given a new input dataset (i.e., detection task) $D_{new} = (X_{new})$ without any labels, Select a model $A^*_{\lambda^*} \in \mathcal{A}$ to employ on $X_{new}$. Where $A^*_{\lambda^*}$ is a optimal model with tuned hyperparameters $\lambda^*$ for a similar dataset to $X_{new}$.

**Problem Formulation:** A Combined model selection and hyperparameter optimization problem (Thornton et al., 2013) for a supervised learning task is as follows:

**Algorithm 1** Pseudocode for LOTUS

```
Inputs: $D_{new}, \mathcal{D}_{meta}, \mathcal{A}$
while $D_i \in \mathcal{D}_{meta}$ do
    $\mathcal{O}_i \leftarrow \psi(\phi(D_{new}, D_i))$                ▷ Distance calculation
$s \leftarrow \operatorname*{argmin}_i \{\mathcal{O}_1, ..., \mathcal{O}_n\}$     ▷ Retrieval of most similar dataset
$A^*_{\lambda^*_{new}} \leftarrow A^*_{\lambda^*_s}$                       ▷ Model Selection
```

In equation 5, $A^*_{\lambda^*}$ is an optimal combination of learning algorithm from search space $A$ with associated hyperparameter space $\Lambda_A$ over $k$ cross validation folds of dataset $D$ where $D = \{X, y\}$ with training and validation splits. $L$ is our evaluation measure.

$$A^*_{\lambda^*} = \operatorname*{argmin}_{\substack{\forall A^j \in \boldsymbol{A} \\ \forall \lambda \in \boldsymbol{\Lambda_A}}} \frac{1}{k} \sum_{f=1}^{k} L\left(A^j_\lambda, \{\boldsymbol{X}^{train}_f, \boldsymbol{y}^{train}_f\}, \{\boldsymbol{X}^{val}_f, \boldsymbol{y}^{val}_f\}\right) \tag{5}$$

The CASH problem from equation 5 relies on the validation split to optimise for the optimal configuration. However, in unsupervised outlier detection scenario the learning algorithm does not have access to labels but the AutoML framework does. We do not perform k-fold CV as is not useful in this setting. Our modified CASH formulation to select the optimal unsupervised algorithm with access to labels is as follows:

$$A^*_{\lambda^*} = \operatorname*{argmin}_{\substack{\forall A^j \in \boldsymbol{A} \\ \forall \lambda \in \boldsymbol{\Lambda_A}}} L\left(A^j_\lambda, \{\boldsymbol{X}^{train}\}\{\boldsymbol{y}^{train}\}\right) \tag{6}$$

**GAMAOD:** For meta-training in this work we develop GAMAOD as a solution to populate our meta-data. GAMAOD is an extension to popular AutoML tool GAMA (Gijsbers & Vanschoren, 2021). GAMA is a general AutoML framework which allows researchers to integrate different search spaces and search algorithms for model selection easily. More details about GAMAOD architecture are provided in Appendix A.3

### 3.2 META-TESTING

Our premise is that, if a prior dataset exists that is very similar to the new dataset, then its optimal algorithms will likely work well on the new dataset. We consider two datasets similar if they have the same underlying data distribution, which we measure using Optimal Transport. We first require a transformation function $\phi$, the purpose of transformation function is to simply make input data compatible to the distance function, for example if it is raw image data then a typical transformation function is normalization of pixels, if it is dirty data then a transformation function can be a mix of encoders and scalers. Next, we calculate the dataset similarity $\mathcal{O}$ based on some distance metric $\psi$ in equation 7.

$$\mathcal{O} = \psi(\phi(D_a), \phi(D_b)) \tag{7}$$

Because our distributions lie on different metric spaces, and we require computationally efficient similarity estimates, we adopt the Low Rank Gromov-Wasserstein distance from equation 4 on these transformed distributions, as summarized in equation 8, where $r$ is the selected rank.

$$\mathcal{O} = \text{GW-LR}^{(r)}(\phi(D_a), \phi(D_b)) \tag{8}$$

The most similar prior dataset $D_s \in \mathcal{D}_{meta}$ is then the dataset with the smallest distance to the new dataset $D_{new}$. LOTUS then assigns the optimal configuration from $\mathcal{A}$: $A^*_{\lambda^*_{new}} = A^*_{\lambda^*_s}$ where $A^*_{\lambda^*_s}$ is predicted as the optimal configuration for $D_{new}$. Meta-testing is summarized in Algorithm 1. LOTUS framework with meta-training and meta-testing is shown in Figure 1

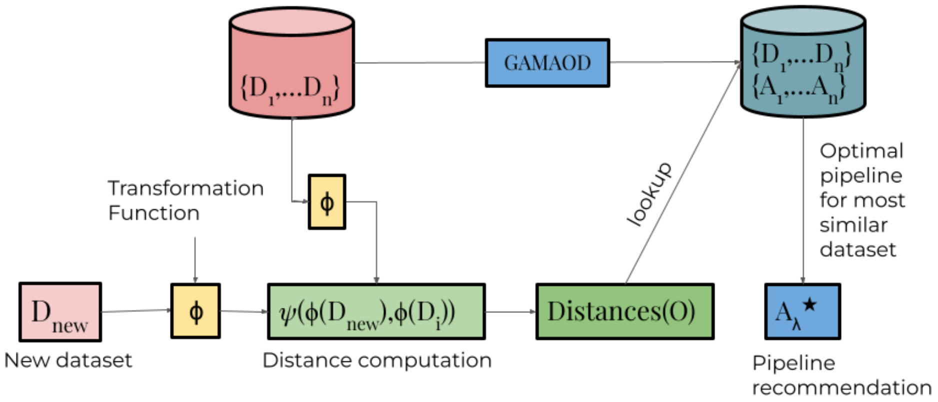


Figure 1: An overview of LOTUS

## 4 EXPERIMENTS ON ADBENCH

For our experiments, we use ADBench (Han et al., 2022) and retrieve all tabular datasets. This collection consists of 46 datasets. As we do not have access to multiple benchmarks we use a leave-one-out strategy for the evaluation of our system, i.e., we take out one dataset at a time from ADBench and use only the other datasets in the meta-data. This ensures independent meta-training on the following datasets. We compare our approach against 7 outlier detection algorithms available in PyOD (Zhao et al., 2019) and the current state of the art meta-learner for outlier detection MetaOD (Zhao et al., 2021). From PyOD we compare our approach with the following algorithms: IForest (Liu et al., 2008), ABOD (Kriegel et al., 2008), OCSVM (Schölkopf et al., 1999), LODA (Pevný, 2015), KNN (Angiulli & Pizzuti, 2002; Ramaswamy et al., 2000), HBOS (Goldstein & Dengel, 2012).

For experimental consistency, we use the same search space in our experiments as MetaOD (A.3) to ensure a fair comparison. We use an asynchronous evolutionary algorithm to iterate over the search space and return the optimal pipeline.

## 5 RESULTS AND DISCUSSION

### 5.1 EXPERIMENTAL RESULTS

We use the Bayesian Wilcoxon signed-rank test (or ROPE test, Benavoli et al. (2017; 2014)) to analyze the results of our experiments. ROPE defines an interval wherein the differences in model performance are considered equivalent to the null value. Using this test allows us to compare model performances in a more practical sense. We set the ROPE value to 1% for our experiments. We use the baycomp library (Benavoli et al., 2017) to run and visualize the analyses.

#### 5.1.1 LOTUS VS METAOD

For pairwise comparison of LOTUS and MetaOD, we use the Bayesian Wilcoxon signed-rank test (or ROPE test Benavoli et al. (2017; 2014)). We use AUC as our performance measure and set the ROPE value to 1%.[1] Results are shown in Figure 2. We find that, based on experiments over the 46

[1] We use the baycomp library Benavoli et al. (2017) to run and visualize the analysis

| Estimator name | p(LOTUS) | p(rope) | p(Estimator) |
|---|---|---|---|
| IForest | 0.99954 | 0.0 | 0.00046 |
| ABOD | 1.0 | 0.0 | 0.0 |
| OCSVM | 1.0 | 0.0 | 0.0 |
| LODA | 1.0 | 0.0 | 0.0 |
| KNN | 1.0 | 0.0 | 0.0 |
| HBOS | 0.99982 | 0.0 | 0.00018 |
| COF | 1.0 | 0.0 | 0.0 |
| LOF | 1.0 | 0.0 | 0.0 |

Table 2: Rope testing results with LOTUS vs PyOD estimators with rope=1%

datasets, there is a 74.0 % probability that LOTUS will outperform MetaOD. Since $p(LOTUS) > p(MetaOD)$ LOTUS proves to be more robust. We show the per-dataset performances in Appendix A.1.

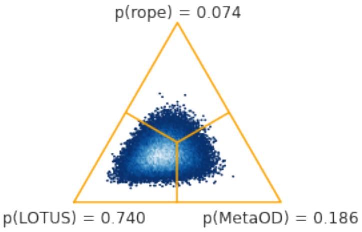


Figure 2: ROPE test LOTUS vs MetaOD.

### 5.1.2 LOTUS VS INDIVIDUAL METHODS

The results of the ROPE test comparing LOTUS with individual outlier detection techniques are shown in Table 2. LOTUS proves to be significantly better than other techniques, with default parameters. In this case $P(LOTUS) >> P(Estimator)$. We also include the critical difference plot of LOTUS vs PyOD estimators in Figure 3, again showing that it performs significantly better. The detailed experimental results are reported in appendix A.1 table 4 and Figure 4.

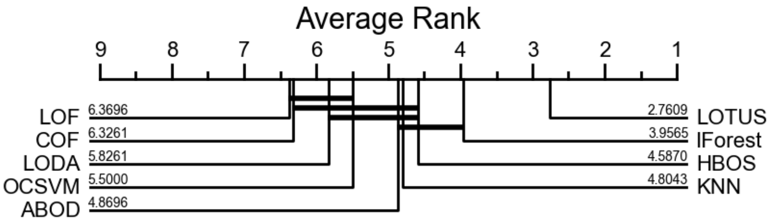
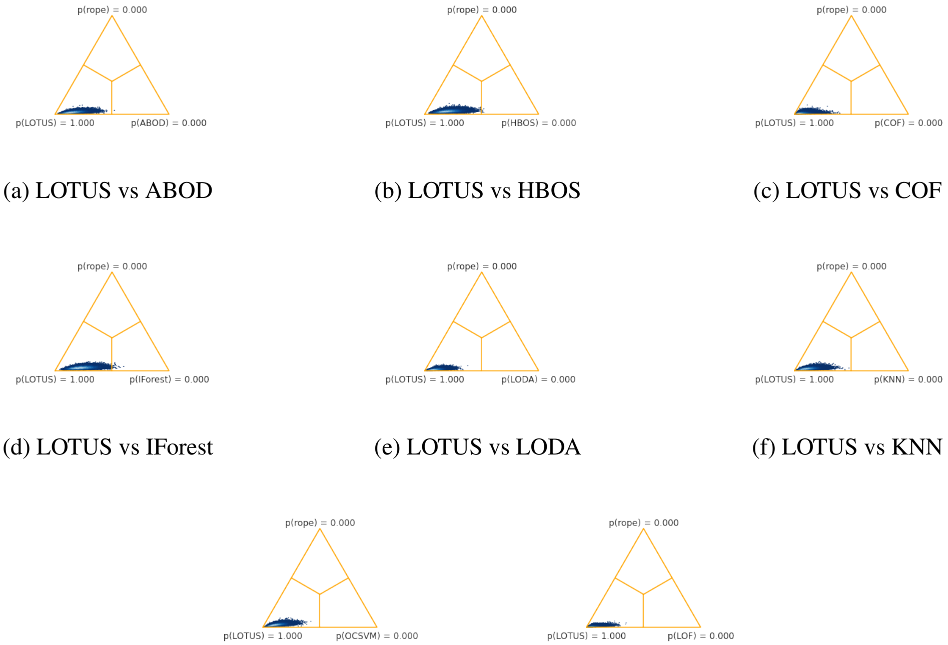


Figure 3: Comparison of average rank (lower is better) of methods w.r.t. performance across datasets in ADBench.



(a) LOTUS vs ABOD

(b) LOTUS vs HBOS

(c) LOTUS vs COF

(d) LOTUS vs IForest

(e) LOTUS vs LODA

(f) LOTUS vs KNN

(g) LOTUS vs OCSVM

(h) LOTUS vs LOF

Figure 4: ROPE test result of LOTUS vs (a) ABOD (b) HBOS (c) COF (d) IForest (e) LODA (f) KNN (g) OCSVM (h) LOF

### 5.2 USING OPTIMAL TRANSPORT DISTANCES AS A SIMILARITY MEASURE

In our experiments, we show that LOTUS is more robust and better than current state of the art meta-learner MetaOD for unsupervised outlier detection tasks and other outlier detection algorithms in default configuration.

In our method we experimentally show that using optimal transport distances like GW-LR is a feasible approach for dataset similarity and meta-learning. We would like to emphasize that this similarity measure should only be used as a relative similarity measure, for e.g. in our case where we use this similarity measure to find the most similar dataset from a collection of datasets in $\mathcal{D}_{meta}$. To estimate to what degree datasets are similar Nies et al. (2021) proposes optimal transport based correlation measures that can be leveraged. Our approach assumes that Wasserstein distances can capture intrinsic properties of datasets and can capture the similarity between them, Alvarez-Melis & Fusi (2020) also proposes their approach with optimal transport distances to provide some sort of distance between dataset.

## 6 CONCLUSION AND FUTURE WORK

Model selection for unsupervised outlier detection is a challenging task. We do not have efficient internal metrics for evaluating an algorithm without ground truth. In this work, we proposed a new meta-learner: **LOTUS**, which uses optimal transport distances to capture the similarity between datasets and uses that similarity measure to recommend pipelines from a meta-data. Through our experiments, we demonstrate that LOTUS outperforms MetaOD and other built-in estimators in PyOD. The LOTUS approach also enables researchers to use a simplified meta-learning framework as compared to other landmark and model-based meta-features methods where meta-features are highly specialized according to the domain. LOTUS comes with its own set of limitations as follows:

1. LOTUS depends on the quality of meta-data, i.e. range of datasets and algorithms in our case. In the worst case scenario, if there are no similar datasets in the $\mathcal{D}_{meta}$, LOTUS can recommend a dataset which is not sufficiently similar to new dataset. On the other hand, it is expected to improve as more benchmarks and datasets with different properties become available.
2. The computation cost of GW-LR on really large datasets can still be very high. In these cases we recommend using stratified sampling or random sampling depending on the nature of dataset and problem.
3. Tuning rank of GW-LR can be tricky. Low rank can result in faster computation but high loss and high rank can result in less efficient algorithm. Scetbon et al. (2022) states an experiment where they study the affect of rank of GW-LR. This rank can also be tuned by minimizing the loss between GW and GW-LR.

Despite the limitations we believe that our approach can be easily extended as a meta-learner to perform model selection in other unsupervised machine learning tasks as well. These include clustering, distance metric learning, density estimation and covariance estimation. This approach can also be used as a meta-learner to warm-start neural architecture search(NAS) problems.

## 7 REPRODUCIBILTY STATEMENT

We opensource both LOTUS and GAMAOD with hyperparameters used for this experiment. We also provide scripts which can be used to perform these experiment for just one dataset without making the meta-data first(not reccomended). We aim to provide modularity to researchers therefore we users them to save and retrieve meta-data in whatever format they want. More information about reproducing our experiments can be found in the README.md of the supplementary code repository. To reproduce LOTUS for other tasks and dataset, users are simply required to change the datasets and algorithms in meta-data. The approach works out of the box for other scenarios. While reproducing the experiments, the results can differ due to stochasticity of few algorithms.

## REFERENCES

Alessandro Achille, Giovanni Paolini, Glen Mbeng, and Stefano Soatto. The information complexity of learning tasks, their structure and their distance. *Information and Inference: A Journal of the IMA*, 10(1):51–72, 01 2021. ISSN 2049-8772. doi: 10.1093/imaiai/iaaa033. URL https://doi.org/10.1093/imaiai/iaaa033.

Charu C. Aggarwal. Outlier analysis. In *Springer New York*, 2013.

Edesio Alcobaça, Felipe Siqueira, Adriano Rivolli, Luís P. F. Garcia, Jefferson T. Oliva, and André C. P. L. F. de Carvalho. Mfe: Towards reproducible meta-feature extraction. *Journal of Machine Learning Research*, 21(111):1–5, 2020. URL http://jmlr.org/papers/v21/19-348.html.

David Alvarez-Melis and Nicoló Fusi. Geometric dataset distances via optimal transport. *ArXiv*, abs/2002.02923, 2020.

Fabrizio Angiulli and Clara Pizzuti. Fast outlier detection in high dimensional spaces. In *PKDD*, 2002.

Alessio Benavoli, Giorgio Corani, Francesca Mangili, Marco Zaffalon, and Fabrizio Ruggeri. A bayesian wilcoxon signed-rank test based on the dirichlet process. In Eric P. Xing and Tony Jebara (eds.), *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 1026–1034, Bejing, China, 22–24 Jun 2014. PMLR. URL https://proceedings.mlr.press/v32/benavoli14.html.

Alessio Benavoli, Giorgio Corani, Janez Demšar, and Marco Zaffalon. Time for a change: a tutorial for comparing multiple classifiers through bayesian analysis. *Journal of Machine Learning Research*, 18(77):1–36, 2017. URL http://jmlr.org/papers/v18/16-305.html.

Guilherme Oliveira Campos, Arthur Zimek, Jörg Sander, Ricardo J. G. B. Campello, Barbora Micenková, Erich Schubert, Ira Assent, and Michael E. Houle. On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data Mining and Knowledge Discovery*, 30:891–927, 2015.

Ciro Castiello, Giovanna Castellano, and Anna Maria Fanelli. Meta-data: Characterization of input features for meta-learning. In Vicenç Torra, Yasuo Narukawa, and Sadaaki Miyamoto (eds.), *Modeling Decisions for Artificial Intelligence*, pp. 457–468, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg. ISBN 978-3-540-31883-5.

HyungJun Cho and Soo-Heang Eo. Outlier detection for mass spectrometric data. *Methods in molecular biology*, 1362:91–102, 2016.

Shein-Chung Chow and Siu keung Tse. Outlier detection in bioavailability/bioequivalence studies. *Statistics in medicine*, 9 5:549–58, 1990.

Michael A. A. Cox and Trevor F. Cox. *Multidimensional Scaling*, pp. 315–347. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008. ISBN 978-3-540-33037-0. doi: 10.1007/978-3-540-33037-0_14. URL `https://doi.org/10.1007/978-3-540-33037-0_14`.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *NIPS*, 2013.

Marco Cuturi, Laetitia Meng-Papaxanthos, Yingtao Tian, Charlotte Bunne, Geoff Davis, and Olivier Teboul. Optimal transport tools (ott): A jax toolbox for all things wasserstein. *ArXiv*, abs/2201.12324, 2022.

William J. Egan and Stephen L. Morgan. Outlier detection in multivariate analytical chemical data. *Analytical chemistry*, 70 11:2372–9, 1998.

Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. Efficient and robust automated machine learning. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL `https://proceedings.neurips.cc/paper/2015/file/11d0e6287202fced83f79975ec59a3a6-Paper.pdf`.

Matthias Feurer, Katharina Eggensperger, Stefan Falkner, Marius Lindauer, and Frank Hutter. Auto-sklearn 2.0: Hands-free automl via meta-learning. *arXiv:2007.04074 [cs.LG]*, 2020.

Yansong Gao and Pratik Chaudhari. An information-geometric distance on the space of tasks. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 3553–3563. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/gao21a.html`.

Pieter Gijsbers and Joaquin Vanschoren. Gama: A general automated machine learning assistant. In Yuxiao Dong, Georgiana Ifrim, Dunja Mladenić, Craig Saunders, and Sofie Van Hoecke (eds.), *Machine Learning and Knowledge Discovery in Databases. Applied Data Science and Demo Track*, pp. 560–564, Cham, 2021. Springer International Publishing. ISBN 978-3-030-67670-4.

Nicolas Goix. How to evaluate the quality of unsupervised anomaly detection algorithms?, 2016. URL `https://arxiv.org/abs/1607.01152`.

Markus Goldstein and Andreas R. Dengel. Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm. 2012.

Songqiao Han, Xiyang Hu, Hailiang Huang, Minqi Jiang, and Yue Zhao. ADBench: Anomaly detection benchmark. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL `https://openreview.net/forum?id=foA_SFQ9zo0`.

Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren. Automated machine learning: Methods, systems, challenges. *Automated Machine Learning*, 2019.

Hans-Peter Kriegel, Matthias Schubert, and Arthur Zimek. Angle-based outlier detection in high-dimensional data. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 444–452, 2008. ISBN 9781605581934. doi: 10.1145/1401890.1401946.

Yuening Li, Daochen Zha, Na Zou, and Xia Hu. Pyodds: An end-to-end outlier detection system with automated machine learning. *Companion Proceedings of the Web Conference 2020*, 2020.

Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. *2008 Eighth IEEE International Conference on Data Mining*, pp. 413–422, 2008.

Martin Q. Ma, Yue Zhao, Xiaorong Zhang, and Leman Akoglu. A large-scale study on unsupervised outlier model selection: Do internal strategies suffice? *CoRR*, abs/2104.01422, 2021a. URL `https://arxiv.org/abs/2104.01422`.

Zhiwei Ma, Daniel S. Reich, Sarah Dembling, Jeff H. Duyn, and Alan P. Koretsky. Outlier detection in multimodal mri identifies rare individual phenotypes among 20,000 brains. *bioRxiv*, 2021b.

Henrique O. Marques, Ricardo J. G. B. Campello, Arthur Zimek, and Jorg Sander. On the internal evaluation of unsupervised outlier detection. In *Proceedings of the 27th International Conference on Scientific and Statistical Database Management*, SSDBM '15, New York, NY, USA, 2015. Association for Computing Machinery. ISBN 9781450337090. doi: 10.1145/2791347.2791352. URL `https://doi.org/10.1145/2791347.2791352`.

Thomas Giacomo Nies, Thomas Staudt, and Axel Munk. Transport dependency: Optimal transport based dependency measures. 2021.

Tomás Pevný. Loda: Lightweight on-line detector of anomalies. *Machine Learning*, 102:275–304, 2015.

Gabriel Peyré and Marco Cuturi. Computational optimal transport. *Found. Trends Mach. Learn.*, 11:355–607, 2019.

Gabriel Peyré, Marco Cuturi, and Justin Solomon. Gromov-wasserstein averaging of kernel and distance matrices. In Maria Florina Balcan and Kilian Q. Weinberger (eds.), *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 2664–2672, New York, New York, USA, 20–22 Jun 2016. PMLR. URL `https://proceedings.mlr.press/v48/peyre16.html`.

Herilalaina Rakotoarison, Louisot Milijaona, Andry RASOANAIVO, Michele Sebag, and Marc Schoenauer. Learning meta-features for autoML. In *International Conference on Learning Representations*, 2022. URL `https://openreview.net/forum?id=DTkEfj0Ygb8`.

Sridhar Ramaswamy, Rajeev Rastogi, and Kyuseok Shim. Efficient algorithms for mining outliers from large data sets. *SIGMOD Rec.*, 29(2):427–438, may 2000. ISSN 0163-5808. doi: 10.1145/335191.335437. URL `https://doi.org/10.1145/335191.335437`.

Meyer Scetbon and Marco Cuturi. Low-rank optimal transport: Approximation, statistics and debiasing. *NeurIPS 2022*, abs/2205.12365, 2022.

Meyer Scetbon, Marco Cuturi, and Gabriel Peyré. Low-rank sinkhorn factorization. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9344–9354. PMLR, 18–24 Jul 2021. URL `https://proceedings.mlr.press/v139/scetbon21a.html`.

Meyer Scetbon, Gabriel Peyré, and Marco Cuturi. Linear-time gromov Wasserstein distances using low rank couplings and costs. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 19347–19365. PMLR, 17–23 Jul 2022. URL `https://proceedings.mlr.press/v162/scetbon22b.html`.

Bernhard Schölkopf, Robert C. Williamson, Alex Smola, John Shawe-Taylor, and John C. Platt. Support vector method for novelty detection. In *NIPS*, 1999.

David Stern, Ralf Herbrich, Thore Graepel, Horst Samulowitz, Luca Pulina, and Armando Tacchella. Collaborative expert portfolio management. In *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence AAAI-10 (to appear)*, July 2010. URL `https://www.microsoft.com/en-us/research/publication/collaborative-expert-portfolio-management/`.

Chris Thornton, Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. Auto-WEKA: Combined selection and hyperparameter optimization of classification algorithms. In *19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 847–855, 2013. ISBN 9781450321747. doi: 10.1145/2487575.2487629.

Joaquin Vanschoren. Meta-learning. pp. 39–68.

Joaquin Vanschoren. Meta-learning: A survey. *ArXiv*, abs/1810.03548, 2018.

Cédric Villani. Optimal transport: Old and new. 2008.

Chi Wang, Qingyun Wu, Markus Weimer, and Erkang Zhu. Flaml: A fast and lightweight automl library. In *MLSys*, 2021.

Hongteng Xu, Dixin Luo, Ricardo Henao, Svati Shah, and Lawrence Carin. Learning autoencoders with relational regularization. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 10576–10586. PMLR, 13–18 Jul 2020. URL `https://proceedings.mlr.press/v119/xu20e.html`.

Yue Zhao, Zain Nasrullah, and Zheng Li. Pyod: A python toolbox for scalable outlier detection. *J. Mach. Learn. Res.*, 20:96:1–96:7, 2019.

Yue Zhao, Ryan Rossi, and Leman Akoglu. Automatic unsupervised outlier model selection. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 4489–4502. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper/2021/file/23c894276a2c5a16470e6a31f4618d73-Paper.pdf`.

| Dataset | LOTUS | MetaOD |
|---|---|---|
| 19_landsat | **0.7902** | 0.5931 |
| 25_musk | **0.9895** | 0.9655 |
| 24_mnist | **1.0000** | 1.0000 |
| 32_shuttle | **0.9216** | **0.9163** |
| 23_mammography | **0.6434** | 0.6477 |
| 42_WBC | 0.8521 | **0.8655** |
| 15_Hepatitis | **0.9353** | **0.9353** |
| 43_WDBC | 0.8548 | **0.9671** |
| 12_fault | **0.9246** | 0.9043 |
| 10_cover | **0.9463** | **0.9436** |
| 34_smtp | 0.2744 | **0.5212** |
| 11_donors | **0.8064** | **0.8049** |
| 29_Pima | **0.8804** | 0.7197 |
| 37_Stamps | **0.9275** | **0.9339** |
| 44_Wilt | **0.7765** | 0.5327 |
| 40_vowels | 0.8491 | **0.9355** |
| 8_celeba | **0.9908** | 0.9906 |
| 1_ALOI | **0.8954** | 0.8957 |
| 30_satellite | **0.8913** | 0.7890 |
| 26_optdigits | **0.9996** | 0.9997 |
| 2_annthyroid | **0.8472** | 0.8445 |
| 41_Waveform | **0.9758** | 0.9413 |
| 28_pendigits | 0.8597 | **0.9265** |
| 4_breastw | **0.7466** | 0.7438 |
| 21_Lymphography | 0.9441 | **0.9861** |
| 20_letter | 0.9701 | **0.9891** |
| 39_vertebral | 0.7634 | **0.8424** |
| 47_yeast | **0.9089** | **0.9097** |
| 3_backdoor | **1.0000** | 1.0000 |
| 13_fraud | **0.9646** | 0.8904 |
| 45_wine | **0.9841** | 0.9481 |
| 22_magic.gamma | **0.9322** | 0.8122 |
| 9_census | 0.9819 | **1.0000** |
| 7_Cardiotocography | **0.9392** | **0.9378** |
| 35_SpamBase | **0.9446** | 0.9015 |
| 46_WPBC | 0.7811 | **0.8088** |
| 36_speech | **1.0000** | 0.4344 |
| 6_cardio | **0.9794** | **0.9793** |
| 31_satimage-2 | **0.9552** | 0.8100 |
| 18_Ionosphere | 0.8072 | **0.8338** |
| 27_PageBlocks | 0.7164 | **0.7668** |
| 5_campaign | **0.9922** | **0.9996** |

Table 3: AUC scores of MetaOD vs LOTUS on ADBench

# A Appendix

## A.1 PERFORMANCES

Table 3 contains the performances of LOTUS and MetaOD on 42 datasets, **we had to remove 4 datasets from this experiment because MetaOD returned invalid models for these datasets(i.e. models with invalid values)**. Scores are in bold where AUC of LOTUS $>$ MetaOD or differ by less than a %. The dataset names are as they were in ADBench (Han et al., 2022).

Table 4 reports the auc scores over datasets from ADBench. The bold number shows scores where LOTUS is better than **all** other estimators in PyOD.

| Dataset | IForest | ABOD | OCSVM | LODA | KNN | HBOS | COF | LOF | LOTUS |
|---|---|---|---|---|---|---|---|---|---|
| 44_Wilt | 0.4719 | 0.5682 | 0.3013 | 0.4082 | 0.4720 | 0.2814 | 0.5442 | 0.4742 | **0.7765** |
| 6_cardio | 0.9437 | 0.4985 | 0.9396 | 0.8927 | 0.7415 | 0.8653 | 0.5445 | 0.6283 | **0.9794** |
| 43_WDBC | 0.9872 | 0.987241 | 0.9896 | 0.9875 | 0.9603 | **0.9989** | 0.7710 | 0.7231 | 0.8548 |
| 4_breastw | 0.9763 | 0.9763 | 0.7786 | **0.9819** | 0.9473 | 0.9693 | 0.3813 | 0.3283 | 0.7466 |
| 42_WBC | **0.9935** | **0.9935** | **0.9941** | **0.9959** | 0.9119 | 0.5708 | **0.9916** | 0.7547 | 0.8521 |
| 47_yeast | 0.4310 | 0.4171 | 0.4483 | 0.4925 | 0.4136 | 0.4100 | 0.4286 | 0.4718 | **0.9089** |
| 45_wine | 0.7352 | 0.7352 | 0.6816 | 0.9231 | 0.4712 | 0.8917 | 0.4122 | 0.3491 | **0.9841** |
| 5_campaign | 0.692549 | 0.6429 | 0.6455 | 0.5664 | 0.6968 | 0.7713 | 0.5645 | 0.5569 | **0.9922** |
| 46_WPBC | 0.5224 | 0.5224 | 0.4759 | 0.5621 | 0.4191 | 0.5552 | 0.4951 | 0.4862 | **0.7811** |
| 7_Cardiotocography | 0.7524 | 0.5394 | 0.8104 | 0.7859 | 0.5825 | 0.6233 | 0.5725 | 0.6119 | **0.9392** |
| 8_celeba | 0.7578 | 0.7578 | 0.761861 | 0.7182 | 0.6322 | 0.8059 | 0.3935 | 0.4354 | **0.9908** |
| 9_census | 0.5981 | 0.5981 | 0.523211 | 0.3255 | 0.6506 | 0.6333 | 0.4132 | 0.4371 | **0.9819** |
| 39_vertebral | 0.3777 | 0.3777 | 0.4273 | 0.2844 | 0.4171 | 0.2823 | 0.3219 | 0.4285 | **0.7634** |
| 41_Waveform | 0.6697 | 0.6981 | 0.4744 | 0.6112 | 0.7821 | 0.6397 | 0.8041 | 0.7760 | **0.9758** |
| 38_thyroid | **0.9796** | **0.9796** | 0.8677 | 0.6995 | 0.9511 | 0.9528 | 0.8719 | 0.8404 | 0.7910 |
| 40_vowels | 0.7083 | 0.9567 | 0.5327 | 0.6559 | **0.9717** | 0.6461 | 0.8497 | 0.9530 | 0.8491 |
| 3_backdoor | 0.7343 | 0.7343 | 0.8022 | 0.7089 | 0.7386 | 0.6654 | 0.7289 | 0.7464 | **1.00** |
| 32_shuttle | 0.9962 | 0.6187 | 0.9874 | 0.9510 | 0.6785 | 0.9949 | 0.5576 | 0.5374 | 0.9216 |
| 31_satimage-2 | 0.9968 | 0.7626 | 0.9835 | 0.9871 | 0.9098 | 0.9859 | 0.4513 | 0.4362 | 0.9552 |
| 26_optdigits | 0.7714 | 0.5255 | 0.5272 | 0.6234 | 0.3981 | 0.8528 | 0.4236 | 0.5701 | **0.99** |
| 1_ALOI | 0.5018 | 0.6095 | 0.5328 | 0.5495 | 0.5556 | 0.4780 | 0.6355 | 0.6296 | **0.8954** |
| 35_SpamBase | 0.6570 | 0.3907 | 0.5205 | 0.2739 | 0.5153 | 0.651507 | 0.4164 | 0.4152 | **0.9446** |
| 36_speech | 0.4699 | 0.7294 | 0.4620 | 0.4485 | 0.4731 | 0.4763 | 0.5531 | 0.4863 | **1.00** |
| 34_smtp | 0.6968 | 0.6702 | 0.0180 | 0.3721 | 0.7445 | 0.8786 | 0.8906 | 0.7185 | 0.2744 |
| 22_magic.gamma | 0.7044 | 0.7991 | 0.5942 | 0.6359 | 0.8232 | 0.6817 | 0.6635 | 0.6684 | **0.9322** |
| 23_mammography | 0.8594 | 0.8594 | 0.854704 | 0.814810 | 0.8596 | 0.8717 | 0.7920 | 0.7647 | 0.6434 |
| 24_mnist | 0.7944 | 0.7503 | 0.8347 | 0.7435 | 0.8282 | 0.6190 | 0.7333 | 0.6986 | **1.00** |
| 20_letter | 0.5815 | 0.8808 | 0.4851 | 0.6274 | 0.8671 | 0.5405 | 0.8297 | 0.8330 | **0.9701** |
| 30_satellite | 0.7077 | 0.5380 | 0.6054 | 0.6092 | 0.6460 | 0.7681 | 0.5569 | 0.5241 | **0.8913** |
| 19_landsat | 0.4955 | 0.50 | 0.3740 | 0.3823 | 0.5771 | 0.5567 | 0.5420 | 0.5268 | **0.7902** |
| 37_Stamps | 0.9095 | 0.9095 | 0.8782 | **0.9445** | 0.7464 | 0.9285 | 0.6363 | 0.5249 | 0.9275 |
| 18_Ionosphere | 0.8678 | 0.8678 | 0.765359 | 0.8583 | 0.8622 | 0.6674 | 0.8504 | **0.9209** | 0.8072 |
| 21_Lymphography | 0.9970 | 0.9970 | 0.9935 | 0.6675 | 0.5128 | **0.9950** | 0.9343 | 0.7045 | 0.9441 |
| 25_musk | 0.9999 | 0.0859 | 0.8186 | 0.9590 | 0.7011 | 1.00 | 0.4003 | 0.7045 | 0.9895 |
| 17_InternetAds | 0.7004 | 0.6733 | 0.7100 | 0.5808 | 0.7123 | 0.7043 | 0.6939 | 0.6760 | **1.00** |
| 16_http | 1.00 | 1.00 | 0.9953 | 0.00 | 0.0013 | 0.9946 | 0.5831 | 0.2536 | 0.7106 |
| 15_Hepatitis | 0.7427 | 0.7427 | 0.7222 | 0.7728 | 0.4678 | 0.8132 | 0.4253 | 0.3346 | **0.9353** |
| 14_glass | 0.8184 | 0.8184 | 0.4592 | 0.6322 | 0.7407 | 0.7917 | **0.8826** | 0.5756 | 0.8374 |
| 13_fraud | 0.9340 | 0.9415 | 0.9143 | 0.7511 | 0.9163 | 0.9411 | 0.9145 | 0.9345 | **0.9646** |
| 11_donors | 0.7942 | 0.7942 | 0.7234 | 0.2607 | 0.8299 | 0.7639 | 0.7202 | 0.5693 | **0.8064** |
| 12_fault | 0.5714 | 0.6764 | 0.4944 | 0.4360 | 0.7130 | 0.4792 | 0.6121 | 0.6088 | **0.9246** |
| 2_annthyroid | 0.8249 | 0.8249 | 0.6060 | 0.3058 | 0.7302 | 0.6915 | 0.7048 | 0.7060 | **0.8472** |
| 27_PageBlocks | 0.8896 | 0.6844 | 0.8926 | 0.7532 | 0.7699 | 0.7886 | 0.6732 | 0.7019 | 0.7164 |
| 28_pendigits | 0.9497 | 0.6730 | 0.9386 | 0.9511 | 0.7058 | 0.9211 | 0.4756 | 0.4501 | 0.8597 |
| 29_Pima | 0.6600 | 0.6600 | 0.5801 | 0.6061 | 0.6856 | 0.7135 | 0.5667 | 0.5777 | **0.8804** |
| 10_cover | 0.9143 | 0.7676 | 0.8864 | 0.8668 | 0.8997 | 0.7952 | 0.8702 | 0.8802 | **0.9463** |

Table 4: AUC Scores: LOTUS vs PyOD estimators with default configuration

### A.2 BASLINES

The 8 baslines estimators and frameworks are listed below with brief description from PyOD's (Zhao et al., 2019) documentation for reference here:

1. **MetaOD**: MetaOD is the first automated tool for outlier detection. MetaOD use collaborative filtering, landmark and model based meta-features to recommend the model for given task.
2. **IForest**: IsolationForest 'isolates' observations by randomly selecting a feature and then randomly selecting a split value between the maximum and minimum values of the selected feature.
3. **LOF**:The anomaly score of each sample is called Local Outlier Factor. It measures the local deviation of density of a given sample with respect to its neighbors. It is local in that the anomaly score depends on how isolated the object is with respect to the surrounding neighborhood. More precisely, locality is given by k-nearest neighbors, whose distance is used to estimate the local density. By comparing the local density of a sample to the local densities of its neighbors, one can identify samples that have a substantially lower density than their neighbors. These are considered outliers.
4. **ABOD**:For an observation, the variance of its weighted cosine scores to all neighbors could be viewed as the outlying score.
5. **HBOS**: Histogram- based outlier detection assumes the feature independence and calculates the degree of outlier by building histograms.
6. **KNN**: kNN class for outlier detection. For an observation, its distance to its kth nearest neighbor could be viewed as the outlying score.
7. **COF**: Connectivity-Based Outlier Factor uses the ratio of average chaining distance of data point and the average of average chaining distance of k nearest neighbor of the data point, as the outlier score for observations.
8. **LODA**: Lightweight on-line detector of anomalies detects anomalies in a dataset by computing the likelihood of data points using an ensemble of one-dimensional histograms.
9. **OCSVM**: One class support vector machines unsupervised outlier Detection. Estimate the support of a high-dimensional distribution.

### A.3 LOTUS+GAMAOD SEARCH SPACE AND METAOD REPRODUCIBILITY

We implement the same searchspace as MetaOD's github repository for a fair comparison [2], MetaOD also uses all the existing datasets from ADbench. We believe that we have fairly evaluated MetaOD against out baseline. We believe that our Benchmark setting was more challenging than the one evaluated in Zhao et al. (2021) where it take child and parent datasets. [3]

### A.4 ARCHITECTURE

An overview of GAMAOD system can be found in Figure 5. GAMAOD is build on top of GAMA. We replace search space from scikit-learn estimators to PyOD estimators. For evaluation of the pipeline we avoid cross-validation and tune the models on the AUC score. These models are then used in meta-data

[2] https://github.com/yzhao062/MetaOD/blob/master/metaod/models/base_detectors.py

[3] https://github.com/yzhao062/MetaOD/blob/2a8ed2761468d2f8ee2cd8194ce36b0f817576d1/metaod/models/train_metaod.py#L44

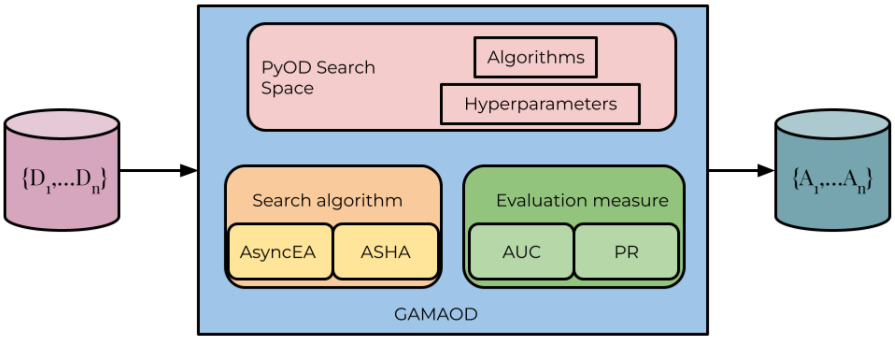


Figure 5: An overview of GAMAOD

### A.5 Experimental Implementation

**Implementation details:** We use Independent Component Analysis(ICA) from scikit-learn as our transformation function $\phi$. We use OTT-JAX library (Cuturi et al., 2022) to implement Low Rank Gromov Wassersstein distance. For this experiment, we set the rank parameter of Low Rank Gromov Wasserstein to 6. The model selection phase of LOTUS in our experiments is as follows: First the datasets are transformed via ICA and then converted into JAX pointclouds geometry objects [4] and then we turn these distributions into a quadratic regularized optimal transport problem. We input this quadratic problem to our Gromov Wasserstein Low Rank solver which returns us the distance(cost) between two datasets. When a new dataset is given to LOTUS, the pipeline corresponding to the dataset with the lowest distance(except the new dataset itself) is chosen from the optimal pipeline database.