ACTIVMETAL: Algorithm Recommendation with Active Meta Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

We present an *active meta learning* approach to model selection or algorithm recommendation. We adopt the point of view "collaborative filtering" recommender systems in which the problem is brought back to a missing data problem: given a sparsely populated matrix of performances of algorithms on given tasks, predict missing performances; more particularly, predict which algorithm will perform best on a new dataset (empty row). In this work, we propose and study an active learning version of the recommender algorithm CofiRank algorithm and compare it with baseline methods. Our benchmark involves three real-world datasets (from StatLog, OpenML, and AutoML) and artificial data. Our results indicate that CofiRank rapidly finds well performing algorithms on new datasets at reasonable computational cost.

Keywords: Model Selection, Recommender, Active Meta Learning.

1 Introduction

While Machine Learning and Artificial Intelligence are taking momentum in many application areas ranging from computer vision to chat bots, selecting the best algorithm applicable to a novel task still requires human intelligence. The field of AutoML (Automatic Machine Learning), aiming at automatically selecting best suited algorithms and hyper-parameters for a given task, is currently drawing a lot of attention. Progress in AutoML has been stimulated by the organization of challenges such as the AutoML challenge series[1]. Among the winning AutoML approaches are AUTOWEKA and AUTO-SKLEARN, developed by the Freiburg team (6; 5; 7) (more in Section 2). These approaches, taking inspiration from Bayesian optimization (4), alternatively learn an inexpensive estimate of model performance on the current dataset, and use this estimate to reduce the number of model candidates to be trained and tested using the usual expensive cross-validation procedure. A novel ingredient of AUTOSKLEARN, referred to as "meta-learning", takes in charge the initialization of the Bayesian optimization process, with a predictor using "meta-features" describing the datasets. Meta-learning reportedly yields significant improvements over random initializations.

Another approach targeting AutoML is based on recommender systems (RS), popularized by the Netflix challenge (2). RS approaches seek the item best suited to a given user, based on historical user-item interactions and user preferences. By analogy (15) proposed first to treat algorithm selection as a recommender problem in which datasets "prefer" algorithms solving their task with better performance. Along this line, the "Algorithm Recommender System" ALORS (9), combines a

[1] http://automl.chalearn.org

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

recommender system and an estimate of model performance based on predefined meta-features, to achieve AutoML (more in Section 2).

In this paper, we propose an *active meta-learning* approach inspired by AUTOSKLEARN and ALORS. Formally (Section 3), given a matrix of historical algorithm performance on datasets, we aim at finding *as fast as possible* the model with best performance on a new dataset. The originality compared to the former approaches lies in the coupled search for the meta-features describing the dataset, the model performance based on these meta-features, and the selection of a candidate model to be trained and tested on the dataset.

This paper is organized as follows: After briefly reviewing the SOTA in Section 2, we formalize our problem setting in Section 3. We then describe the benchmark data in Section 4 and provide an empirical validation of the approach in Section 5. While the validation considers only the "classical" machine learning settings, it must be emphasized that the proposed approach does not preclude of any type of tasks or algorithms, hence is applicable to a broader range of problems.

2 State of the art

It is notorious that the success of model search techniques can be dramatically improved by a careful initialization. In AUTOSKLEARN, the search is improved by a sophisticated initialization using a form of transfer learning (10) called "meta-learning". The meta-data samples include all the datasets of openml.org (12) (a platform which allows to systematically run algorithms on datasets). Systematically launching AUTOSKLEARN on each dataset yields the best (or near best) models associated with each dataset.

Independently, each dataset is described using so-called meta-features. Meta-features are generally of two kinds: i) simple statistics of the dataset such as number of training examples, number of features, fraction of missing values, presence of categorical variables, etc.; ii) performance on the current dataset of "landmark algorithms", namely a well-chosen set of algorithms that can be trained and tested with moderate computational effort such as one nearest neighbor (1NN) or decision trees.

When considering a new dataset, AUTOSKLEARN first determines its nearest neighbors in the meta-feature space, and initializes the search using the best models associated with these neighbors. Other meta-learning formalisms, not considered further in this paper, are based on learning an estimate of the model performance from meta-features (11), or learning to predict the best performing algorithm, as a multi-class classification problem (17).

The delicate issue is to control the cost of the initialization step: considering many landmark algorithms comes with an expected benefit (a better initialization of the search), and with a cost (the computational cost of running the landmarks).

As said, recommender systems (RS) aim at selecting the item best suited to a particular user, given a community of users, a set of items and some historical data of past interactions of the users with the items, referred to as "collaborative matrix" (16; 3), denoted S (for "score") in this paper. As first noted by (15), algorithm selection can be formalized as a recommender problem, by considering that a dataset "likes better" the algorithms with best performances on this dataset. Along this line, one proceeds by i) estimating all algorithm performances on this dataset (without actually evaluating them by training and testing); and ii) recommending the algorithm(s) with best estimated performance on this dataset.

The merits of RS approaches regarding algorithm selection are twofold. Firstly, **RS approaches are frugal** (like other methods, e.g. co-clustering). RS proceeds by estimating the value associated with each (user, item) pair − here, the performance associated with each (algorithm, dataset) − from a tiny fraction of the (user, item) ratings, under the assumption that the collaborative matrix is of low rank k. More precisely the (usually sparse) matrix S of dimensions (p, N) is approximated by UV', with U a (p, k) matrix and V a (N, k) matrix, such that $\langle U_{i,\cdot}, V_{j,\cdot} \rangle$ is close to $S_{i,j}$ for all pairs i, j (e.g. using maximum margin matrix factorization in (14)). U (respectively V) is referred to as latent representation of the users (resp. the items). In the model selection context, RS approaches are thus frugal: they can operate even when the performance of a model on a dataset is known on a tiny fraction of the (model, dataset) pairs. Secondly, most-recent **RS approaches are ranking methods.** Estimating algorithm performance is a harder problem than ranking them in order of merit. A second benefit of RS is that they can rank items conditionally to a given user. The CofiRank algorithm (18)

accordingly considers the rank matrix (replacing $S_{i,j}$ with the rank of item j among all items user i has rated) and minimizes the Normalized Discounted Cumulative Gain (NDCG) in which correctness in higher ranked items is more important. As optimizing NDCG is non-convex, CofiRank thus instead optimizes a convex upper-bound of NDCG.

In counterpart for these merits, mainstream RS is not directly applicable to AutoML, as it focuses on recommending items to known users (warm-start recommendation). Quite the contrary, AutoML is concerned with recommending items (models) to new users (new datasets), a problem referred to as cold-start recommendation (13; 8). This drawback is addressed in the general purpose ALORS system (9), where external meta-features are used to estimate the latent representation $\hat{U}_.$ of the current dataset; this estimated latent representation is used together with the latent representation of any model to estimate the model performance (as $\langle \hat{U}_., V_j \rangle$) and select the model with best estimated performance. The novel active meta-learning approach presented in this paper proposes a different approach to warm start, not requiring external meta-features: Previously evaluated algorithm scores are themselves used as meta-features (see Section 3).

3 Problem setting and algorithms

We define the **active meta-learning problem** in a **collaborative filtering recommender** setting as follows:

GIVEN:

- An ensemble of **datasets** (or tasks) $\mathcal{D}$ of elements d (not necessarily finite);
- A finite ensemble of n **algorithms** (or machine learning models) $\mathcal{A}$ of elements $a_j, j = 1, \cdots, N$;
- A **scoring program** $\mathcal{S}(d, a)$ calculating the performance (score) of algorithm a on dataset d (*e.g.* by cross-validation). Without loss of generality we will assume that **the larger $\mathcal{S}(d, a)$, the better**. The evaluation of $\mathcal{S}(d, a)$ can be computationally expensive, hence we want to limit the number of times S is invoked.
- A **training matrix** S, consisting of p lines (corresponding to example datasets $d_i, i = 1, \cdots p$ drawn from $\mathcal{D}$) and n columns (corresponding to all algorithms in $\mathcal{A}$), whose elements are calculated as $S_{ij} = \mathcal{S}(d_i, a_j)$, but may contain missing values (denoted as NaN).
- A **new test dataset** $d_t \in \mathcal{D}$, NOT part of training matrix S. This setting can easily be generalized to test matrices with more than one line.

GOAL: Find "as quickly as possible" $j_* = argmax_j(S(d_t, a_j))$.

For the purpose of this paper "as quickly as possible" shall mean by evaluating as few values of $S(d_t, a_j), j = 1, \cdots, n$ as possible. More generally, it could mean minimizing the total computational time, if there is a variance in execution time of $S(d_t, a_j)$ depending on datasets and algorithms. However, because we rely in our experimental section on archival data without information of execution time, we reserve this refinement for future studies. Additionally, we assume that the computational cost of our meta-learning algorithm (excluding the evaluations of $\mathcal{S}$) is negligible compared to the evaluations of $\mathcal{S}$, which has been verified in practice.

In our setting, we reach our goal iteratively, in an **Active Meta Learning** manner (ACTIVMETAL), see Algorithm 1. The variants that we compare differ in the choices of INITIALIZATIONSCHEME(S) and SELECTNEXT$(S, \mathbf{t})$, as described in Algorithms 2-5: Given a new dataset (an empty line), we need to initialize it with one or more algorithm performances, this initialization is done by INITIALIZATIONSCHEME(S) and is indispensable to fire CofiRank. Algorithms 2-5 show 2 initialization methods: randperm in Algorithm 2 (the first algorithm is selected at random) and median in Algorithms 3-5 (the algorithms are sorted by their median over all datasets in training matrix S and the one with highest median is selected as the first algorithm to evaluate). Once we have evaluated the first algorithm, the next algorithms can be chosen with or without active learning, this is done by SELECTNEXT$(S, \mathbf{t})$: Algorithm 2-3 without active meta learning select next algorithms at random or according to median over training datasets, i.e. the knowledge from evaluated algorithms on the new dataset is not taken into account; Algorithm 4-5 run CofiRank for active meta learning, which, initialized with performances of evaluated algorithm, returns a ranking of algorithms on the new dataset. The difference is that in Alg. 4 we run CofiRank for each selection of next algorithm, i.e.

CofiRank is initialized with more and more known values. In Alg. 5 CofiRank is run only once at the beginning, initialized with 3 landmark values.

Algorithm 1 ACTIVMETAL

procedure ACTIVMETAL($\mathcal{A}$, $\mathcal{S}$, S, d_t, n_{max})
  $n \leftarrow size(\text{S}, 2)$ ▷ Number of algorithms to be evaluated on d_t
  $\mathbf{t} \leftarrow$ NaNvector(n) ▷ Algorithm scores on d_t are initialized w. missing values
  $j_+ \leftarrow$INITIALIZATIONSCHEME(S) ▷ Initial algorithm $a_{j+} \in \mathcal{A}$ is selected
  while $n < n_{max}$ **do**
    $\mathbf{t}[j_+] \leftarrow \mathcal{S}(d_t, a_{j+})$ ▷ Complete **t** w. one more prediction score of a_{j+} on d_t
    $j_+ =$ SELECTNEXT(S, $\mathbf{t}$)
    $n \leftarrow$length(notNaN(**t**)) ▷ number of algorithms evaluated on d_t
  return j_+

Algorithm 2 Random

procedure INITIALIZATIONSCHEME(S)
  $\mathbf{r} \leftarrow$randperm(size(S, 2)) ▷ Replaced by something more clever elsewhere
  return $j_+ \leftarrow$ argmax($\mathbf{r}$)
procedure SELECTNEXT(S,$\mathbf{t}$)
  $\mathbf{evaluated} \leftarrow$notNaN($\mathbf{t}$)
  $\mathbf{r} \leftarrow$randperm(size(S, 2)) ▷ Replaced by something more clever elsewhere
  $\mathbf{r}(\mathbf{evaluated}) \leftarrow -$Inf
  return $j_+ \leftarrow$ argmax($\mathbf{r}$)

Algorithm 3 SimpleRankMedian

procedure INITIALIZATIONSCHEME(S)
  $\mathbf{r} \leftarrow$median(S, 2) ▷ Column-wise median
  return $j_+ \leftarrow$ argmax($\mathbf{r}$)
procedure SELECTNEXT(S,$\mathbf{t}$)
  $\mathbf{evaluated} \leftarrow$notNaN($\mathbf{t}$)
  $\mathbf{r} \leftarrow$median(S, 2) ▷ Column-wise median
  $\mathbf{r}(\mathbf{evaluated}) \leftarrow -$Inf
  return $j_+ \leftarrow$ argmax($\mathbf{r}$)

Algorithm 4 ActiveMetaLearningCofiRank

procedure INITIALIZATIONSCHEME(S)
  $\mathbf{r} \leftarrow$median(S, 2) ▷ Column-wise median
  return $j_+ \leftarrow$ argmax($\mathbf{r}$)
procedure SELECTNEXT(S,$\mathbf{t}$)
  $\mathbf{evaluated} \leftarrow$notNaN($\mathbf{t}$)
  $\mathbf{r} \leftarrow$ CofiRank(S,$\mathbf{t}$) ▷ Collaborative filtering on [S; **t**] returning last line
  $\mathbf{r}(\mathbf{evaluated}) \leftarrow -$Inf
  return $j_+ \leftarrow$ argmax($\mathbf{r}$)

Algorithm 5 MedianLandmarks1CofiRank

procedure INITIALIZATIONSCHEME(S)
  $\mathbf{r} \leftarrow$median(S, 2) ▷ Column-wise median
  return $j_+ \leftarrow$ argmax($\mathbf{r}$)
procedure SELECTNEXT(S,$\mathbf{t}$)
  $\mathbf{evaluated} \leftarrow$notNaN($\mathbf{t}$)
  if $length(\mathbf{evaluated}) < num\_landmarks$ **then**
    $\mathbf{r} \leftarrow$median(S, 2) ▷ Column-wise median
  else if $length(\mathbf{evaluated}) == num\_landmarks$ **then**
    static $\mathbf{r} \leftarrow$ CofiRank(S,$\mathbf{t}$) ▷ Keep the CofiRank predictions thereafter
  $\mathbf{r}(\mathbf{evaluated}) \leftarrow -$Inf
  return $j_+ \leftarrow$ argmax($\mathbf{r}$)

Table 1: **Statistics of benchmark datasets used**. *#Datasets=number of datasets, #Algo=number of algorithms, Rank=rank of the performance matrix.*

	Artificial	Statlog	OpenML	AutoML
#Dataset	50	21	76	30
#Algo	20	24	292	17
Rank	20	21	76	17
Metric	None	Error rate	Accuracy	BAC or R^2
Preprocessing	None	Take square root	None	Scores for aborted algo. set to 0
Source	Generated by authors	Statlog Dataset in UCI database	Alors (9) website	AutoML1 (2015-2016)

Table 2: **Results of meta-learning methods for all 4 meta-datasets**. *Performances of meta-learning algorithms are measured as the area under the meta learning curve (AUMLC) normalized by the area of the best achievable curve. Active_Meta _Learning w. CofiRank (our proposed method) performs always best, although not significantly considering the 1-sigma error bars of the leave-one-dataset-out procedure.*

	Artificial	Statlog	OpenML	AutoML
Active_Meta _Learning w. CofiRank	$\mathbf{0.91}$ $(\pm\mathbf{0.03})$	$\mathbf{0.802}$ $(\pm\mathbf{0.117})$	$\mathbf{0.96}$ $(\pm\mathbf{0.04})$	$\mathbf{0.84}$ $(\pm\mathbf{0.11})$
Random	0.81 (±0.05)	0.77 (±0.05)	0.95 (±0.03)	0.79 (±0.07)
SimpleRank w. median	0.7 (±0.2)	0.798 (±0.102)	0.95 (±0.04)	0.82 (±0.12)
Median _LandMarks w. 1-CofiRank	0.88 (±0.04)	0.795 (±0.099)	0.92 (±0.08)	0.83 (±0.11)

4 Benchmark data

To benchmark our proposed method, we gathered datasets from various sources (Table 1). Each dataset consists of a matrix S of performances of algorithms (or models) on tasks (or datasets). Datasets are in lines and algorithms in columns. The performances were evaluated with a single training/test split or by cross-validation. The tasks were classification or regression tasks and the metrics quasi-homogeneous for each S matrix (*e.g.* Balanced Accuracy a.k.a. BAC for classification and R^2 for regression). We excluded data sources for which metrics were heterogeneous (a harder problem that we are leaving for further studies). Although ACTIVMETAL lends itself to using sparse matrices S (with a large fraction of missing values), these benchmarks include only full matrices S.

The artificial dataset was constructed from a matrix factorization to create a simple benchmark we understand well, allowing to easily vary the problem difficulty. Matrix S is simply obtained as a product of three matrices $U\Sigma V$, U and V being orthogonal matrices and Σ a diagonal matrix of "singular values", whose spectrum was chosen to be exponentially decreasing, with $\Sigma_{ii} = exp(-\beta i)$, $\beta = 100$ in our experiments. The other benchmarks were gathered from the Internet or the literature and represent the performances of real algorithms on real datasets. We brought back all metrics to scores that are "the larger the better". In one instance (StatLog), we took the square root of the performances to equalize the distribution of scores (avoid a very long distribution tail). For AutoML, many algorithms were aborted due to execution time constraints. We set the corresponding performance to 0. To facilitate score comparisons between benchmark datasets, all S matrices were globally standardized (*i.e.* we subtracted the global mean and divided by the global standard deviation). This scaling does not affect the results.

We conducted various exploratory data analyses on the benchmark data matrices, including two-way hierarchical clustering, to visualize whether there were enough similarities between lines and columns to perform meta-learning. See our supplemental material referenced at the end of this paper.

5 Results

In this section, we analyze the experimental results of Table 2 and Figure 1. The graphs represent meta-learning curves, that is the performance of the best algorithm found so far as a function of the number of algorithms tried.[2] The ground truth of algorithm performance is provided by the values of the benchmark matrices (see Section 4).

We remind the reader that in a meta-learning problem, each sample is a dataset. To evaluate meta-learning we use the equivalent of a leave-one-out estimator, *i.e.* leave-one-dataset-out. Hence, we use as meta-learning training data all datasets but one, then create the learning curve for the left-out dataset. Thus, given a benchmark data matrix, we generate meta-learning curves using as matrix S a sub-matrix with one line left out (held out), which serves as target vector $\mathbf{t}$ for the dataset tested. Subsequently, we average all meta-learning curves, step-by-step. Thus the result shown in Figure 1 are the averaged learning curves obtained with the leave-one-dataset-out scheme, *i.e.* averaged over all datasets, for a given benchmark dataset.

To evaluate the significance of the efficiency of our proposed method, we ran 1000 times the Random search algorithm, in which algorithms are ran in a random sequence. We drew the curves of median performance (blue curves) and showed as blue shadings various quantiles. The fact that the red curves, corresponding to the proposed algorithm Active Meta Learning w. CofiRank is generally above the blue curve comforts us that the method is actually effective. It is not always significantly better than the median of Random search. However, this is a very hard benchmark to beat. Indeed, the median of Random search is not a practical method, it is the average behavior of random search over many runs. Thus, performing at least as good as the median of Random search is actually pretty good.

We also compared our method with two other baselines. (1) The SimpleRank w. median (green curves) uses the median performance of algorithms on all but the left-out dataset. Thus it does not perform any *active* meta-learning. (2) The Median Landmark w. 1 CofiRank (pink curves) makes only one call of CofiRank to reduce computational expense, based on the performance of only 3 Landmark algorithms, here simply picked based on median ranking.

The first benchmark using artificial data (Figure 1(a)) a relative position of curves that we intuitively expected: $SimpleRank w.median$ (in green) does not perform well and Active Meta Learning w. CofiRank (in red) is way up in the upper quantiles of the random distribution, close to the ideal curve that goes straight up at the first time step (selects right away the best algorithm). Median Landmark w. 1 CofiRank (in pink) quickly catches up with the red curve: this is promising and shows that few calls to CofiRank might be needed, should this become a computational bottleneck.

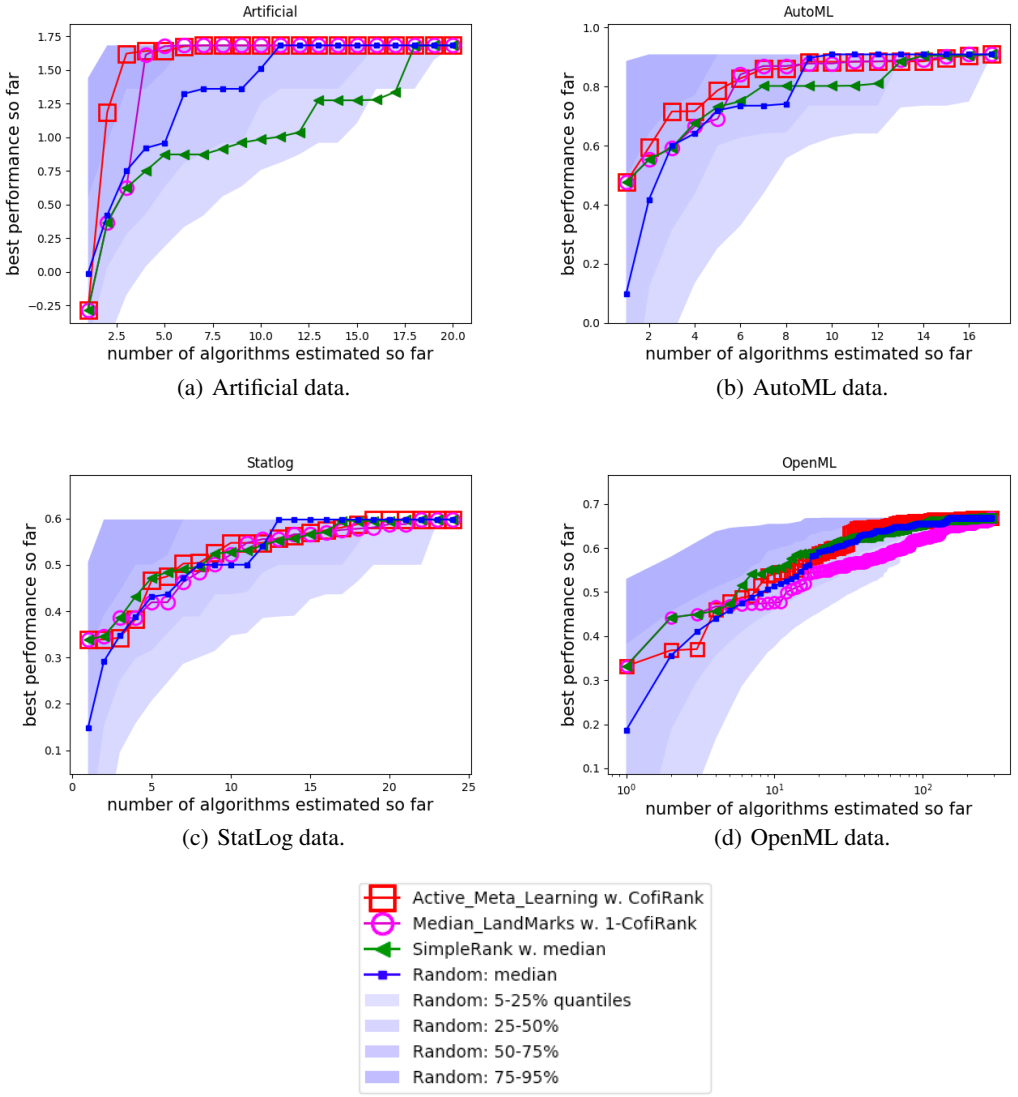
However, the analysis of the results on real data reveals a variety of regimes. The first benchmark using the datasets of the AutoML challenge (Figure 1(b)) gives results rather similar to artificial data in which Active Meta Learning w. CofiRank still dominates, though SimpleRank w. median performs surprisingly well. More surprisingly, Active Meta Learning w. CofiRank does not beat SimpleRank w. median on the StatLog benchmark and beats it with difficulty (after more than 10% of the algorithms have been trained/tested) on the OpenML benchmark. Also, the cheap algorithm calling CofiRank just once (Median Landmark w. 1 CofiRank, performing no active learning) which looked promising on other benchmark datasets, performs poorly on the OpenML dataset. This is unfortunate since this is the largest dataset, on which running active-learning is most computationally costly. We provide a discussion of computational considerations in Section 6.

Table 2 sums up the results in terms of area under the meta-learning curves (AUMLC). Active Meta Learning w. CofiRank consistently outperforms other methods, although not significantly according to the error bars.

6 Discussion and conclusion

We have presented an approach to algorithm recommendation (or model selection) based on meta-learning, capitalizing on previous runs of algorithms on a variety of datasets to rank candidate algorithms and rapidly find which one will perform best on a new dataset. The originality of the

[2] In the future, when we have meta-learning datasets for which the computational run time of algorithms is recorded, we shall tackle the harder and more interesting problem of meta-learning performance as a function of "total" computational time rather than number of algorithms tried.



(a) Artificial data.

(b) AutoML data.

(c) StatLog data.

(d) OpenML data.

Figure 1: **Meta-learning curves.** *We show results of 4 methods on 4 meta-learning datasets, using the leave-one-dataset-out estimator. The learning curves represent performance of the best model trained/tested do far, as a function of the number of models tried. The curves have been averaged over all datasets held-out. The method Active Meta Learning w. CofiRank (red curve) generally dominates other methods. It always performs at least as well as the median of random model selection (blue curve), a hard-to-beat benchmark. The more computationally economical Median Landmark w. 1 CofiRank consisting in training/testing only 3 models (Landmarks) to rank methods using only 1 call to CofiRank (pink curve) generally performs well, except on OpenML data for which it would be most interesting to use it, since this is the largest meta learning datasets. Thus active learning cannot easily be replaced by the use of Landmarks, lest more work is put into Landmark selection. The method SimpleRank w. median that ranks algorithm with their median performance (green curve) is surprisingly a strong contender to Active Meta Learning w. CofiRank for the StatLog and OpenML datasets, which are cases in which algorithms perform similarly on all datasets.*

paper lies in its active learning approach based on a collaborative-filtering algorithm: CofiRank. Collaborative filtering is a technique to fill in missing data in a collaborative matrix of scores, which in our case represents performances of algorithms on datasets. Starting from the evaluation of a single algorithm on a new dataset of interest, the CofiRank method ranks all remaining algorithms by completing the missing scores in the collaborative matrix for that new dataset. The next most promising algorithm is then evaluated and the corresponding score added to the collaborative matrix. The process is iterated until all missing scores are filled in, by trying all algorithms, or until the allotted time is exhausted.

We demonstrated that Active Meta Learning w. CofiRank performs well on a variety of benchmark datasets. Active Meta Learning w. CofiRank does not always beat the naive SimpleRank w. median baseline method, but it consistently outperforms the "hard-to-beat" median of Random ranking, while SimpleRank w. median does not.

We also investigated whether the (meta-) active learning aspect is essential or can be replaced by running CofiRank a single time after filling in a few scores for Landmark algorithms. This technique (called Median Landmark w. 1 CofiRank) seemed promising on the smallest benchmark datasets, but gives significantly worse results that Active Meta Learning w. CofiRank on the largest benchmark dataset on which it would help most (computationally). One avenue of future research would be to put more effort in the selection of better Landmarks.

Further work also includes accounting for the computational expense of model search in a more refined way. In this work, we neglected the cost of performing meta-learning compared to training and testing the algorithms. This is justified by the fact that their run time is a function of the volume of training data, which is considerably smaller for the collaborative matrix (of dimension usually $\simeq 100$ datasets times $\simeq 100$ algorithms) compared to modern-times "big data" datasets (tens of thousands of samples times thousands of features). However, as we acquire larger meta learning datasets, this cost may become significant. Secondly, we assumed that all algorithms had a comparable computational time (to be able to use meta-learning datasets for which this information was not recorded). In the future, we would like to take into account the duration of each algorithm to better trade-off accuracy and computation. It is also worth noting that ACTIVMETAL does not optimize the exploration/exploitation trade-off. It is more geared toward exploitation than exploration since the next best algorithm is chosen at every step. Further work may include incorporating monitoring the exploration/exploitation trade-off. In particular, as said, so far we have not taken into account the computational cost of running algorithms. When we have a total time budget to respect, exploring first using faster algorithms then selecting slower (but better) algorithms may be a strategy that ActivMetal could adopt (thus privileging first exploration, then exploitation).

At last, the experiments performed in this paper assumed that, except to the new dataset being tested, there were no other missing values in the collaborative matrix. One of the advantages of collaborative filtering techniques is that they can handle matrices sparsely populated. This deserves further investigation.

7 Supplemental material

From here is your material, reorganize

7.1 Visualization of all meta datasets

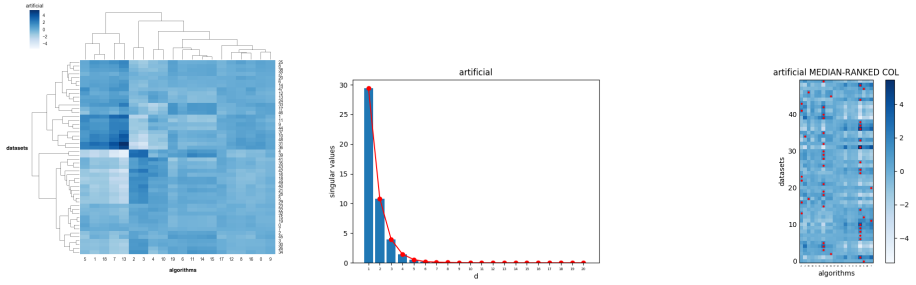


Figure 2: $S_{artificial}$ after global normalization. LEFT: Hierarchical clustering. MIDDLE: Singular values . RIGHT: Columns arranged based on their medians (from highest to lowest), the maximum values for each dataset are marked with a red dot.

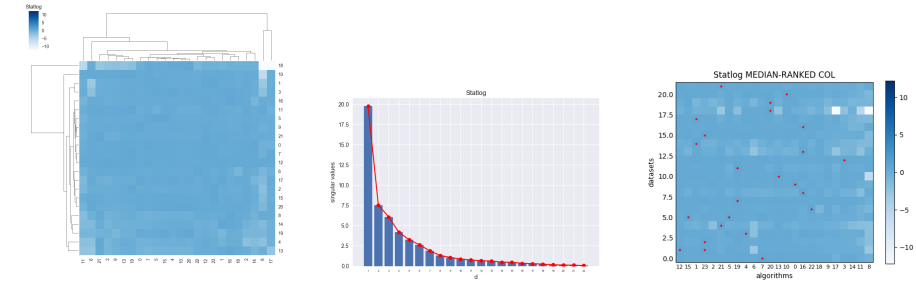


Figure 3: $-S_{statlog}$ after taking natural logarithm and global normalization. LEFT: Hierarchical clustering. MIDDLE: singular values . RIGHT: Columns arranged based on their medians (from highest to lowest), the maximum values for each dataset are marked with a red dot.

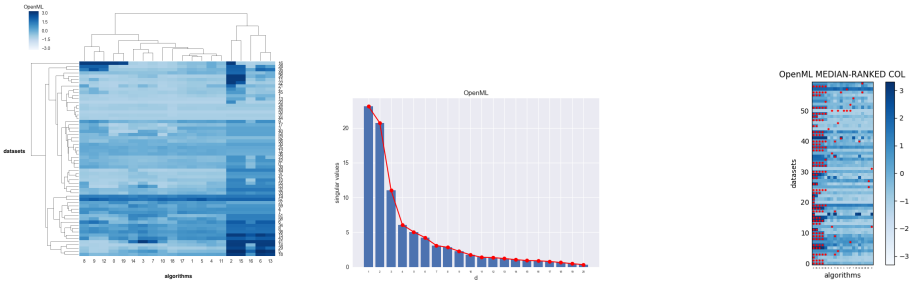


Figure 4: S_{OpenML} after global normalization. LEFT: Hierarchical clustering. MIDDLE: singular values . RIGHT: Columns arranged based on their medians (from highest to lowest), the maximum values for each dataset are marked with a red dot.

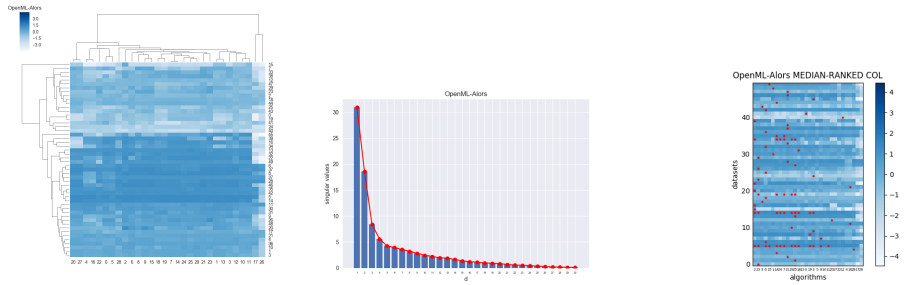


*Figure 5: $S_{OpenML-Alors}$ (sub-matrix of (50 datasets * 30 algos)) after global normalization. LEFT: Hierarchical clustering. MIDDLE: singular values . RIGHT: Columns arranged based on their medians (from highest to lowest), the maximum values for each dataset are marked with a red dot.*

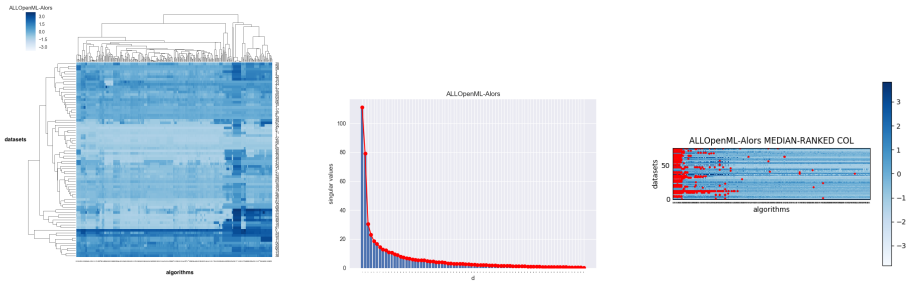


*Figure 6: $S_{ALLOpenML-Alors}$ (the whole matrix of (76 datasets * 292 algos)) after global normalization. LEFT: Hierarchical clustering. MIDDLE: singular values . RIGHT: Columns arranged based on their medians (from highest to lowest), the maximum values for each dataset are marked with a red dot.*

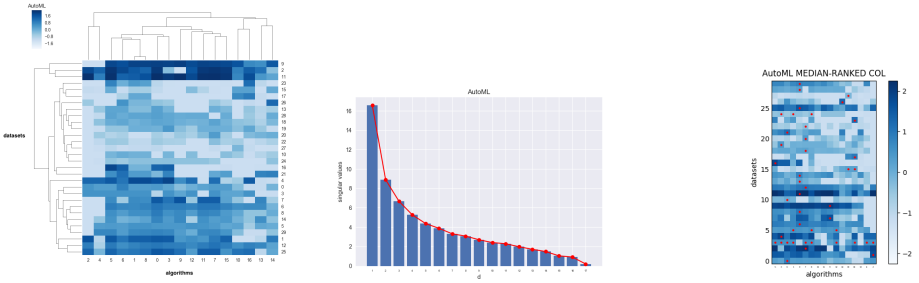


Figure 7: S_{AutoML} after global normalization. LEFT: Hierarchical clustering. MIDDLE: singular values . RIGHT: Columns arranged based on their medians (from highest to lowest), the maximum values for each dataset are marked with a red dot.

7.2 Comparison of algorithms for some single dataset

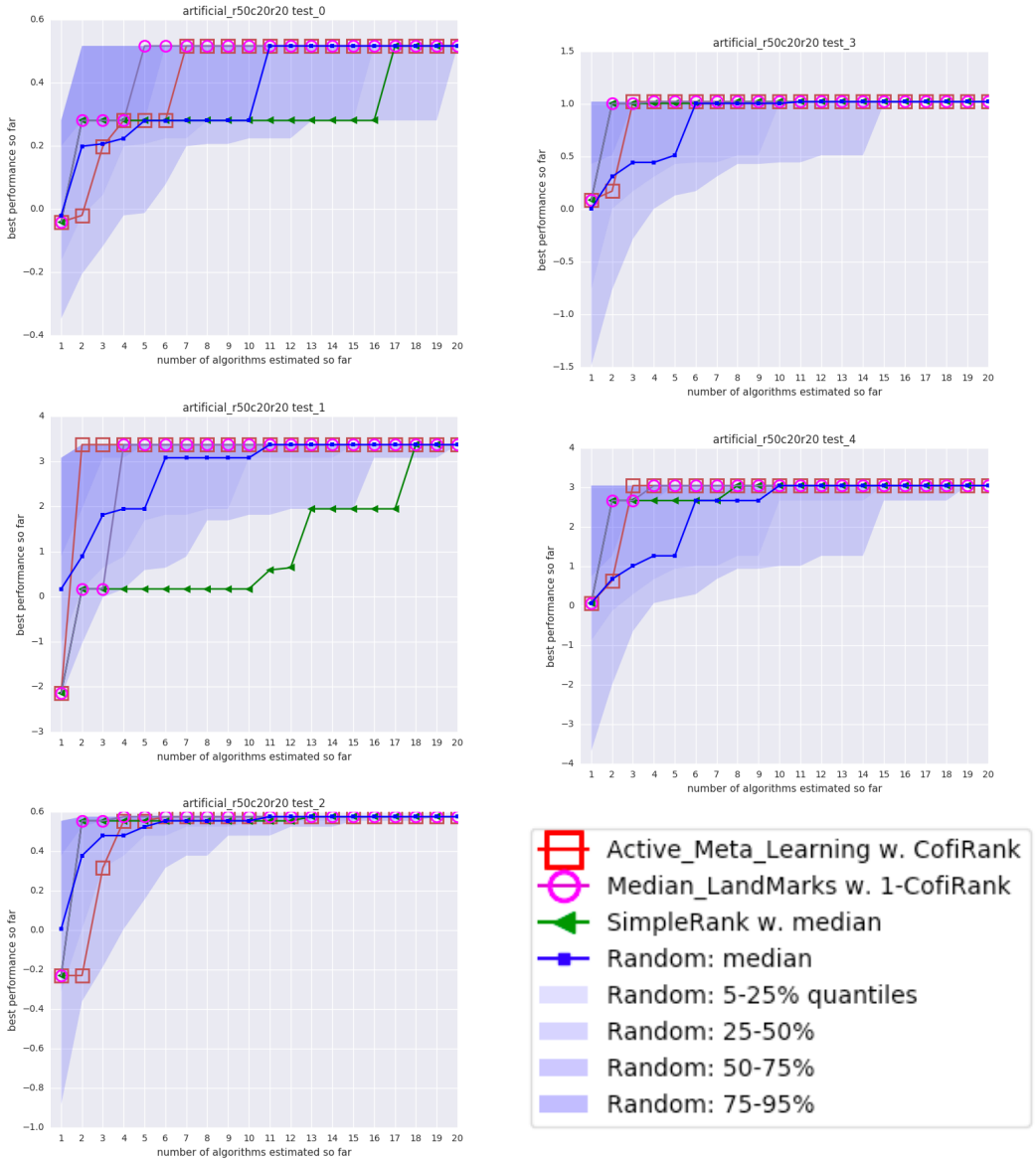


Figure 8: ARTIFICIAL DATA After global normalization: the comparison of meta learning algorithms for 5 single test datasets are showed. The random curves are median over 1000 runs, the shading area are 5%, 25%, 75% and 95% quantiles.

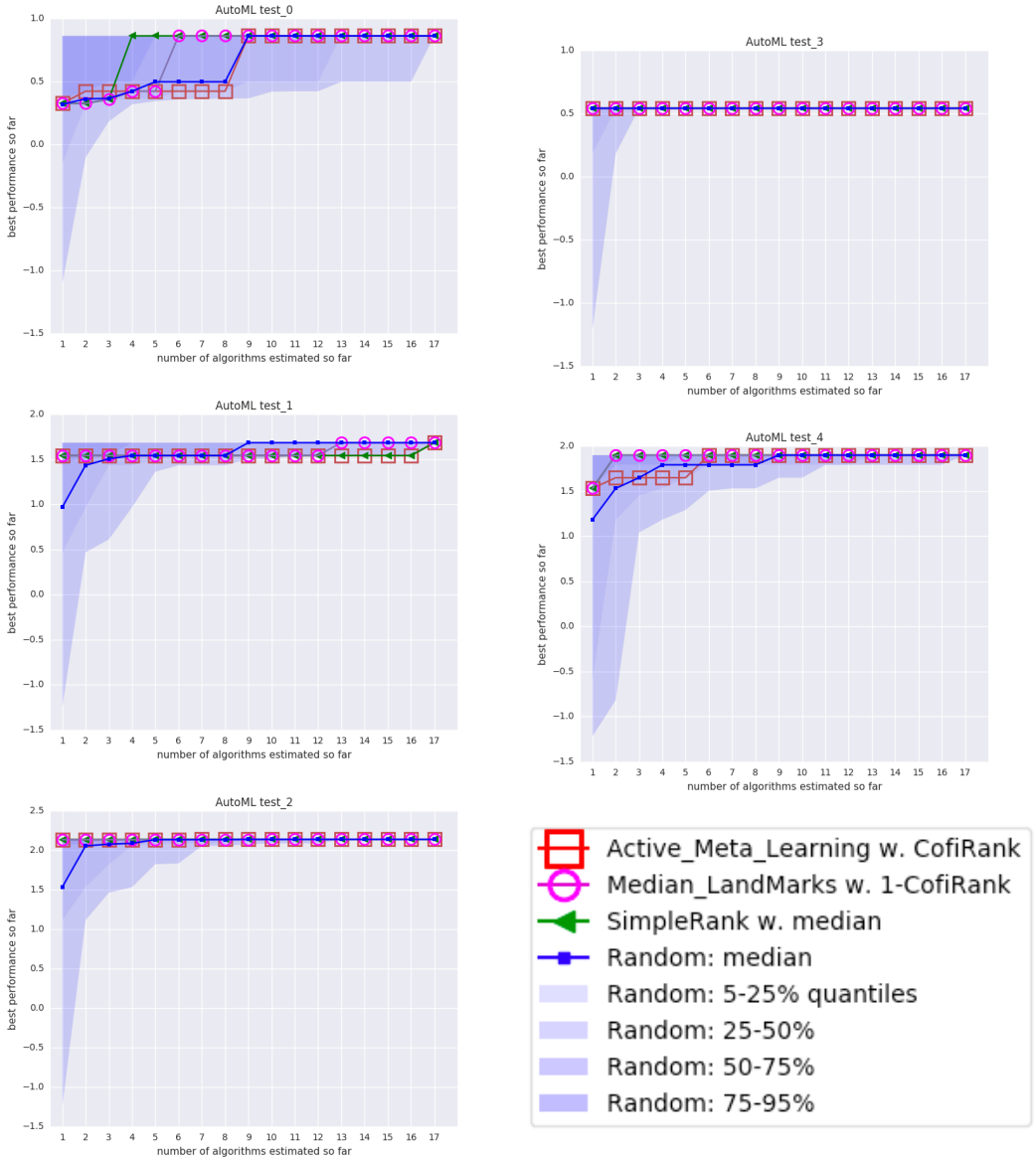


Figure 9: AUTOML DATA After global normalization: the comparison of meta learning algorithms for 5 single test datasets are showed. The random curves are median over 1000 runs, the shading area are 5%, 25%, 75% and 95% quantiles.

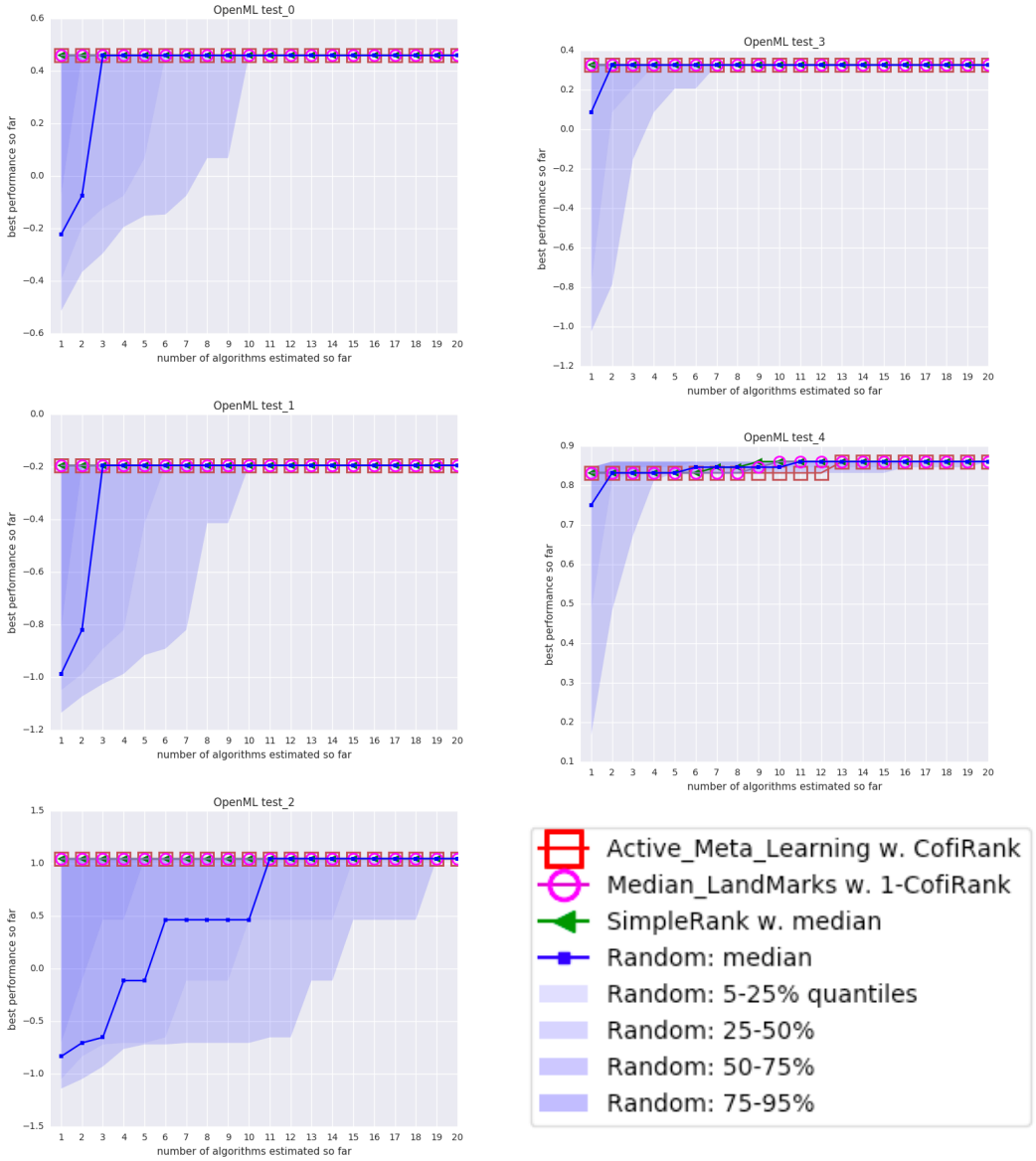


Figure 10: OPENML DATA After global normalization: the comparison of meta learning algorithms for 5 single test datasets are showed. The random curves are median over 1000 runs, the shading area are 5%, 25%, 75% and 95% quantiles.

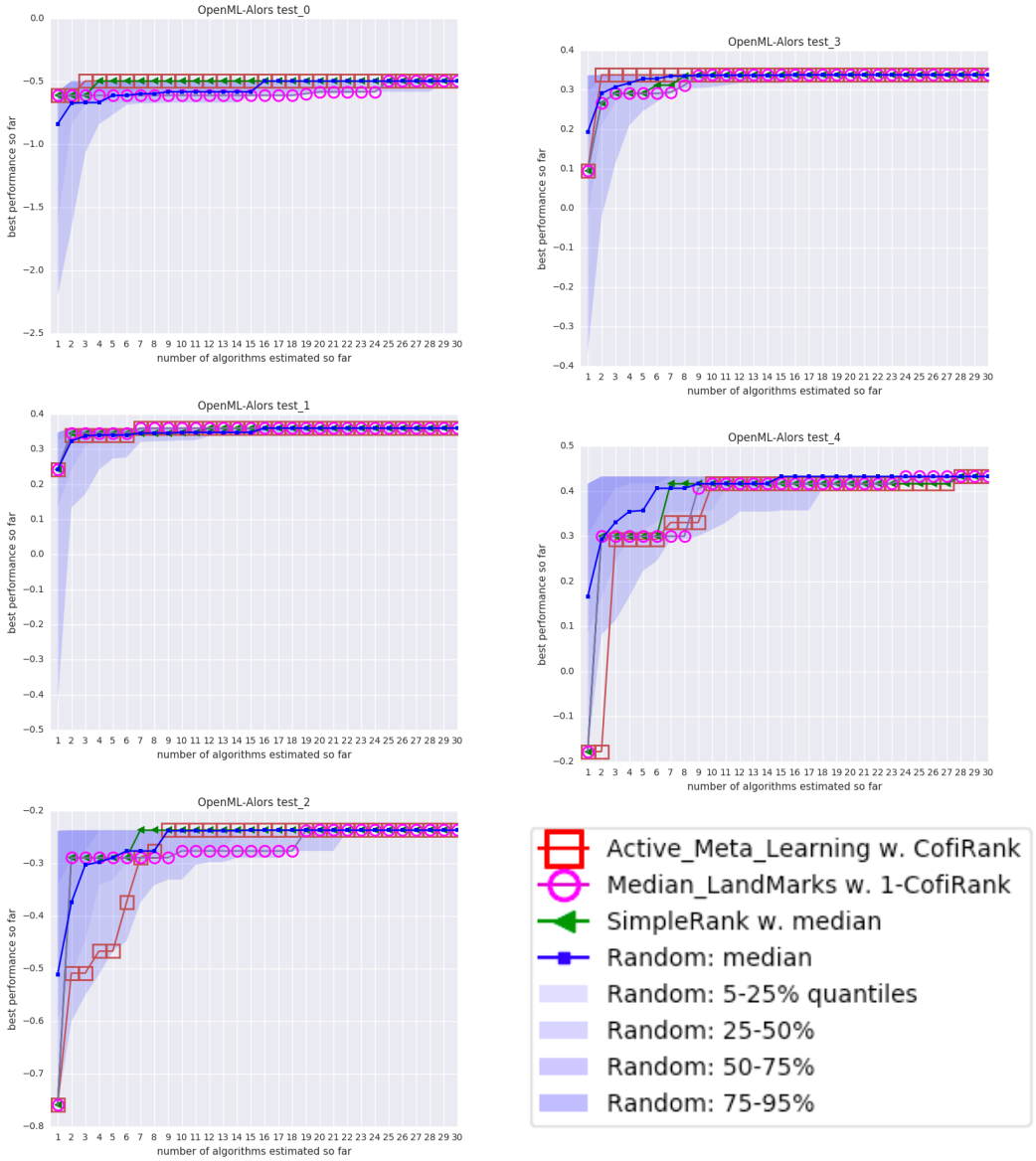


Figure 11: OPENML-ALORS DATA after global normalization: the comparison of meta learning algorithms for 5 single test datasets are showed. The random curves are median over 1000 runs, the shading area are 5%, 25%, 75% and 95% quantiles.

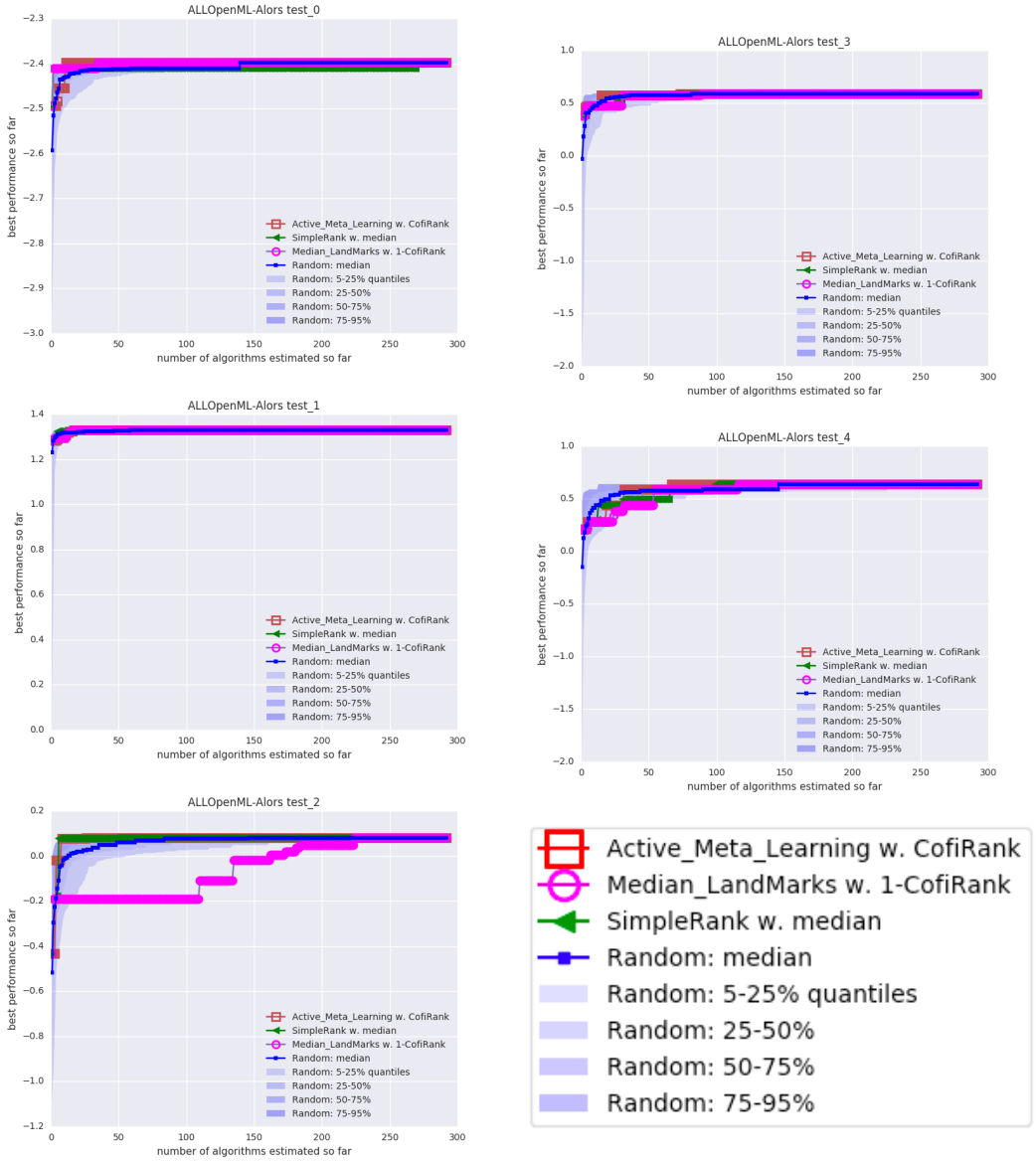


Figure 12: ALLOPENML-ALORS DATA After global normalization: the comparison of meta learning algorithms for 5 single test datasets are showed. The random curves are median over 1000 runs, the shading area are 5%, 25%, 75% and 95% quantiles.

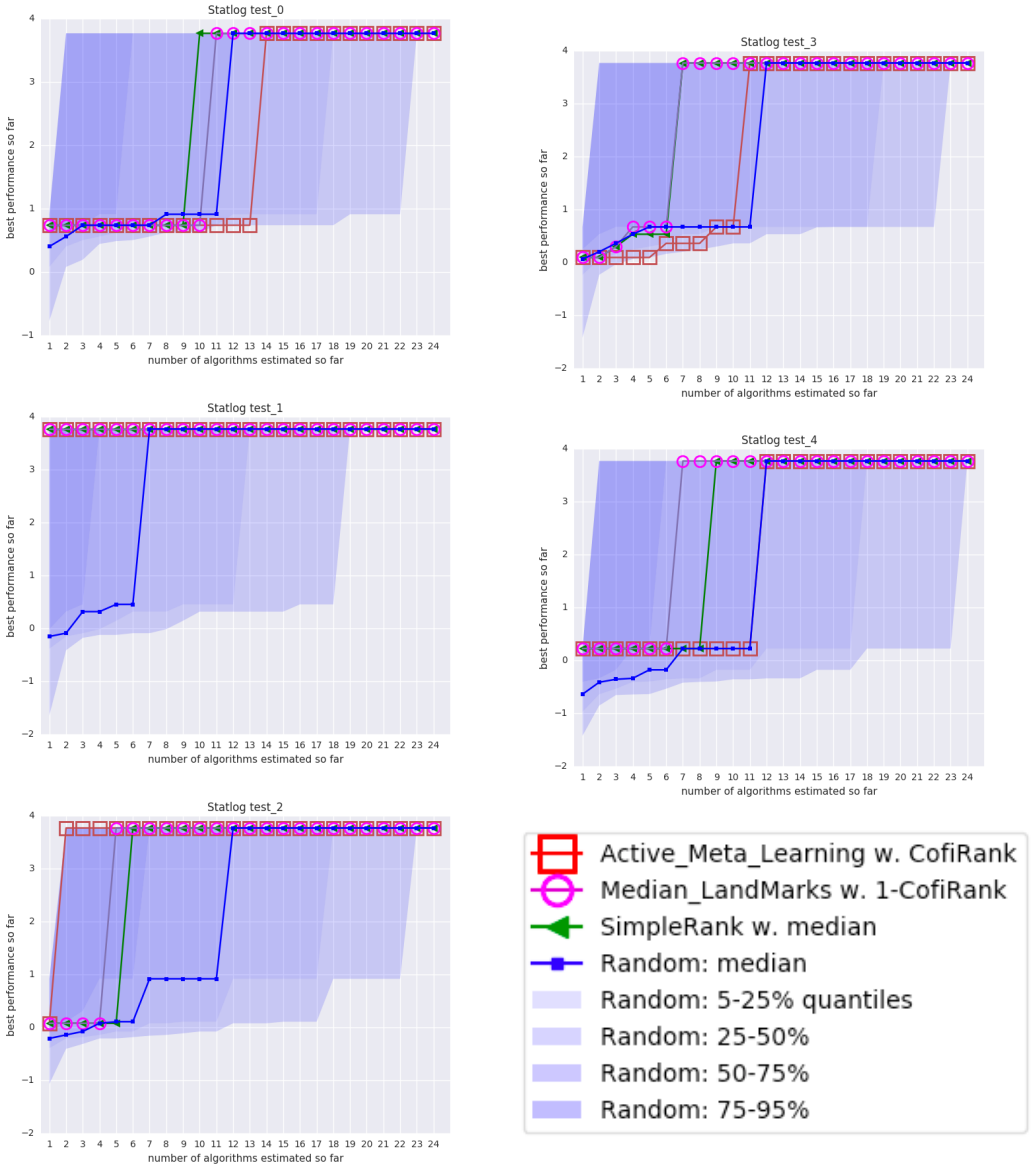
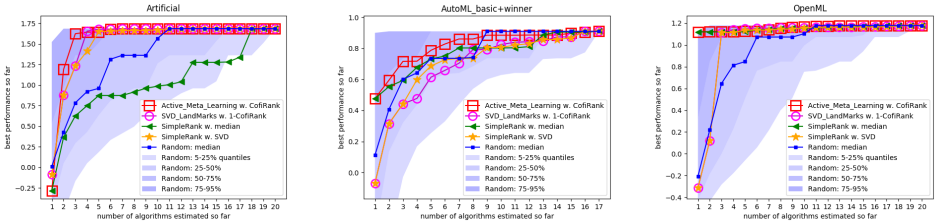


Figure 13: Statlog data After global normalization: the comparison of meta learning algorithms for 4 single test datasets are showed. The random curves are median over 1000 runs, the shading area are 5%, 25%, 75% and 95% quantiles.

7.3 Comparison with SVD-based algorithms



Supplemental material, data and code

For full reproducibility of our results, datasets and code are available on Github. To run it, CofiRank must be installed. We recommend using the Docker (1) image we built for this purpose. Please refer to the Github repository for all instructions. Our repository also includes a Jupyter-notebook with additional graphs referred to in the text.

References

[1] Docker. https://www.docker.com/

[2] Bennett, J., Lanning, S., Netflix: The Netflix prize. KDD Cup and Workshop in conjunction with ACM SIGKDD p. 201–206 (2007)

[3] Bobadilla, J., Ortega, F., Hernando, A., Gutiérrez, A.: Recommender systems survey. Knowledge-Based Systems **46**, 109–132 (2013)

[4] Eggensperger, K., Feurer, M., Hutter, F., Bergstra, J., Snoek, J., Hoos, H., Leyton-Brown, K.: Towards an empirical foundation for assessing bayesian optimization of hyperparameters. In: NIPS workshop on Bayesian Optimization in Theory and Practice (2013)

[5] Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M., Hutter, F.: Efficient and robust automated machine learning. In: Proceedings of the Neural Information Processing Systems, pp. 2962–2970 (2015), https://github.com/automl/auto-sklearn

[6] Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M., Hutter, F.: Methods for improving bayesian optimization for automl. In: Proceedings of the International Conference on Machine Learning 2015, Workshop on Automatic Machine Learning (2015)

[7] Feurer, M., Springenberg, J., Hutter, F.: Initializing bayesian hyperparameter optimization via meta-learning. In: Proceedings of the AAAI Conference on Artificial Intelligence. pp. 1128–1135 (2015)

[8] Gunawardana, A., Meek, C.: Tied boltzmann machines for cold start recommendations. In: Proceedings of the 2008 ACM conference on Recommender systems. pp. 19–26. ACM (2008)

[9] Mısır, M., Sebag, M.: Alors: An algorithm recommender system. Artificial Intelligence **244**, 291–314 (2017)

[10] Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE Transactions on knowledge and data engineering **22**(10), 1345–1359 (2010)

[11] Rice, J.: The algorithm selection problem. Advances in computers **15**, 65–118 (1976)

[12] van Rijn, J., Bischl, B., Torgo, L., Gao, B., Umaashankar, V., Fischer, S., Winter, P., Wiswedel, B., Berthold, M., Vanschoren, J.: OpenML: A collaborative science platform. In: Blockeel, H., Kersting, K., Nijssen, S., Železný, F. (eds.) Proceedings of the Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD Part III, LNCS, vol. 8190, pp. 645–649. Springer (2013)

[13] Schein, A.I., Popescul, A., Ungar, L.H., Pennock, D.M.: Methods and metrics for cold-start recommendations. In: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 253–260. ACM (2002)

[14] Srebro, N., Rennie, J., Jaakkola, T.: Maximum-margin matrix factorization. Advances in neural information processing systems **17**(5), 1329–1336 (2005)

[15] Stern, D., Herbrich, R., Graepel, T., Samulowitz, H., Pulina, L., Tacchella, A.: Collaborative expert portfolio management. In: AAAI. pp. 179–184 (2010)

[16] Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. Advances in artificial intelligence **2009**, 4 (2009)

[17] Sun-Hosoya, L., Guyon, I., Sebag, M.: Lessons learned from the automl challenge. In: Conférence sur l'Apprentissage Automatique 2018. Rouen, France (June 2018)

[18] Weimer, M., Karatzoglou, A., Le, Q., Smola, A.: CofiRank-maximum margin matrix factorization for collaborative ranking. In: Proceedings of the 21st Annual Conference on Neural Information Processing Systems (NIPS). pp. 222–230 (2007)

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the paper's main contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations are discussed in the section "Discussion and conclusion".

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: This paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: All information needed to reproduce the experimental results is given in form of pseudo-code of provided Algorithms and their code accessible to public.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open Access to Data and Code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The github repo for reproducibility is given, in which you can find the code and dataset.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: These details are given in the section of "Results".

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: These details are given in the section of "Results" and the figures in this section.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: We only discussed the related computational expenses.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We reviewed the code of ethics and our paper conforms with it.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: Our paper doesn't discuss potential positive societal impacts and negative societal impacts of the work performed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for Existing Assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: creators or original owners of assets (e.g., code, data, models), used in the paper are cited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: This is given in the github repo, whose link is given in the paper.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.