A Implementation Details

A.1 Deep Active Learning Decomposition

For any uncertainty sampling algorithm, picking the top-B most uncertain examples can be easily decomposed into an iterative procedure that picks the next most uncertain example. Next, for diversity based deep active learning algorithms, one usually rely on a greedy iterative procedure to collect a batch, e.g. K-means++ for BADGE [Ash et al., 2019] and greedy K-centers for Coreset [Sener and Savarese, 2017]. Lastly, deep active learning algorithms such as Cluster-Margin [Citovsky et al., 2021] and GALAXY [Zhang et al., 2022] have already proposed their algorithms as iterative procedures that select unlabeled examples sequentially.

A.2 Implementation of Modified Submodular

Instead of requiring access to a balanced holdout set [Kothawade et al., 2021], we construct the balanced set using training examples. We use the Submodular Mutual Information function FLQMI as suggested by Table 1 of Kothawade et al. [2021]. The proposed greedy submodular optimization is itself an iterative procedure that selects one example at a time. While SIMILAR usually performs well, our modification that discards the holdout set is unfortunately ineffective in our experiments. This is primarily due to the lack of the holdout examples, which may often happen in practical scenarios.

A.3 Stanford Car Multi-label Dataset

We transform the original labels into 10 binary classes of

1. If the brand is “Audi".
2. If the brand is “BMW".
3. If the brand is “Chevrolet".
4. If the brand is “Dodge".
5. If the brand is “Ford".
6. If the car type is “Convertible".
7. If the car type is “Coupe".
8. If the car type is “SUV".
9. If the car type is “Van".
10. If the car is made in or before 2009.

A.4 Negative Weighting for Common Classes

For multi-label classifications, for some classes, there could be more positive associations (label of 1s) than negative associations (label of 0s). Therefore, in those classes, the rarer labels are negative. In class diverse reward $\langle v^t_{div}, y\rangle$ in Section 4.1, we implement an additional weighting of $\mathbb{1}^t_{rare} * v_{div^t}$, where $*$ denotes an elementwise multiplication. Here, each element $\mathbb{1}^t_{rare,i} \in \{1, -1\}$ takes value -1 when $\text{COUNT}^t(i)$ is larger than half the size of labeled set. This negative weighting can been seen as upsampling negative class associations when positive associations are the majority.

A.5 Model Training

All of our experiments are conducted using the ResNet-18 architecture [He et al., 2016] pretrained on ImageNet. We use the Adam optimizer [Kingma and Ba, 2014] with learning rate of 1e-4 and weight decay of 5e-5.

A.6 Baseline Algorithms

In the original GALAXY work by Zhang et al. [2022], their algorithm construct K one-vs-rest linear graphs, one for each class. GALAXY requires finding the shortest shortest path among all

K graphs, an operation whose computation scales linearly in K. When K is large, this becomes computationally prohibitive to run. Therefore, we instead include K separate GALAXY algorithms, each only bisecting on one of the one-vs-rest graphs. This is equivalent with running K GALAXY algorithms, one for each binary classification task between class $i \in [K]$ and the rest. As a baseline, we interleave these algorithms uniformly at random.

For Uncertainty sampling in multi-label settings, we simply have K individual uncertainty sampling algorithms, where the i-th algorithm samples the most uncertain example based only on the binary classification task of class i.

B Proof of Theorem 5.2

Our proof follows a similar procedure from regret analysis for Thompson Sampling of the stochastic multi-armed bandit problem [Lattimore and Szepesvári, 2020]. Let $\alpha^t := \{\alpha^{t,j}\}_{j=1}^B$ and $y^t := \{y^{t,j}\}_{j=1}^B$ denote the actions and observations from the i-th round. We define the history up to t as $H_t = \{\alpha^1, y^1, \alpha^2, y^2, ..., \alpha^{t-1}, y^{t-1}\}$. Moreover, for each $i \in [M]$, we define $H_{t,i} = \{y^{t',j} \in H_t : \alpha^{t',j} = i\}$ as the history of all observations made by choosing the i-th arm (algorithm).

Now we analyze reward estimates at each round t. When given history H_t and arm $i \in [M]$, each observation $y \in H_{t,i}$ is an unbiased estimate of θ^i as $y \sim \mathbb{P}_{\theta^i}$. Therefore, for any fixed v^t, $\langle v^t, y\rangle$ is an unbiased estimate of the expected reward $\langle v^t, \theta^i\rangle$, which we denote by $\mu^{t,i}$.

For each arm i, we can then obtain empirical reward estimate $\bar{\mu}^{t,i}$ of the true expected reward $\mu^{t,i}$ by $\bar{\mu}^{t,i} := \frac{1}{1 \vee |H_{t,i}|} \sum_{y \in H_{t,i}} \langle v^t, y\rangle$ where $\bar{\mu}^{t,i} = 0$ if $|H_{t,i}| = 0$. Since expected rewards and reward estimates are bounded by $[-1, 1]$, by standard sub-Gaussian tail bounds, we can then construct confidence interval,

$$\mathbb{P}\left(\forall i \in [M], t \in [T], |\bar{\mu}^{t,i} - \mu^{t,i}| \le d^{t,i}\right) \ge 1 - \frac{1}{T}$$

where $d^{t,i} := \sqrt{\frac{8\log(MT^2)}{1 \vee |H_{t,i}|}}$. Additionally, we define upper confidence bound as $U^{t,i} = \text{clip}_{[-1,1]}\left(\bar{\mu}^{t,i} + d^{t,i}\right)$.

At each iteration t, we have the posterior distribution $\mathbb{P}(\Theta = \cdot | H_t)$ of the ground truth $\Theta = \{\theta^i\}_{i=1}^M$. $\widehat{\Theta} = \{\widehat{\theta}^i\}_{i=1}^M$ is sampled from this posterior. Consider $i_\star^t = \arg\max_{i \in M} \langle v^t, \theta^i\rangle$ and $\alpha^{t,j} = \arg\max_{i \in M} \langle v^t, \widehat{\theta}^i\rangle$. The distribution of $i_\star^t$ is determined by the posterior $\mathbb{P}(\Theta = \cdot | H_t)$. The distribution of $\alpha^{t,j}$ is determined by the distribution of $\widehat{\Theta}$, which is also $\mathbb{P}(\Theta = \cdot | H_t)$. Therefore, $i_\star^t$ and $\alpha^{t,j}$ are identically distributed. Furthermore, since the upper confidence bounds are deterministic functions of i when given H_t, we then have $\mathbb{E}[U^{t,\alpha^{t,j}} | H_t] = \mathbb{E}[U^{t,i_\star^t} | H_t]$.

As a result, we upper bound the Bayesian regret by

$$\begin{aligned} BR(\texttt{TAILOR}) &= \mathbb{E}\left[\sum_{t=1}^{T}\sum_{j=1}^{B} \mu^{t,i_\star^t} - \mu^{t,\alpha^{t,j}}\right] \\ &= \mathbb{E}\left[\sum_{t=1}^{T}\sum_{j=1}^{B} (\mu^{t,i_\star^t} - U^{t,i_\star^t}) + (U^{t,\alpha^{t,j}} - \mu^{t,\alpha^{t,j}})\right]. \end{aligned}$$

Now, note that since $\bar{\mu}^{t,i} \in [-1, 1]$ we have $\text{clip}_{[-1,1]}\left(\bar{\mu}^{t,i} + d^{t,i}\right) = \text{clip}_{[-\infty,1]}\left(\bar{\mu}^{t,i} + d^{t,i}\right)$, where only the upper clip takes effect. Based on the sub-Gaussian confidence intervals $\mathbb{P}\left(\forall i \in [M], t \in [T], |\bar{\mu}^{t,i} - \mu^{t,i}| \le d^{t,i}\right) \ge 1 - \frac{1}{T}$, we can derive the following two confidence bounds:

$$\begin{aligned}\mathbb{P}(\forall i \in [M], t \in [T], \mu^{t,i} > U^{t,i}) &= \mathbb{P}(\forall i \in [M], t \in [T], \mu^{t,i} > \text{clip}_{[-1,1]}(\bar{\mu}^{t,i} + d^{t,i})) \\ &= \mathbb{P}(\forall i \in [M], t \in [T], \mu^{t,i} > \bar{\mu}^{t,i} + d^{t,i}) \text{ , since } \mu^{t,i} \leq 1 \\ &= \mathbb{P}(\forall i \in [M], t \in [T], \mu^{t,i} - \bar{\mu}^{t,i} > d^{t,i}) \leq \frac{1}{2T} \\ \mathbb{P}(\forall i \in [M], t \in [T], U^{t,i} - \mu^{t,i} > 2d^{t,i}) &= \mathbb{P}(\forall i \in [M], t \in [T], \text{clip}_{[-1,1]}(\bar{\mu}^{t,i} + d^{t,i}) - \mu^{t,i} > 2d^{t,i}) \\ &\leq \mathbb{P}(\forall i \in [M], t \in [T], \bar{\mu}^{t,i} + d^{t,i} - \mu^{t,i} > 2d^{t,i}) \\ &= \mathbb{P}(\forall i \in [M], t \in [T], \bar{\mu}^{t,i} - \mu^{t,i} > d^{t,i}) \leq \frac{1}{2T}.\end{aligned}$$

Now with the decomposition,

$$\begin{aligned}BR(\texttt{TAILOR}) &= \mathbb{E}\left[\sum_{t=1}^{T}\sum_{j=1}^{B} \mu^{t,i_\star^t} - \mu^{t,\alpha^{t,j}}\right] \\ =&\mathbb{E}\left[\sum_{t=1}^{T}\sum_{j=1}^{B} \mu^{t,i_\star^t} - U^{t,i_\star^t}\right] + \mathbb{E}\left[\sum_{t=1}^{T}\sum_{j=1}^{B} U^{t,\alpha^{t,j}} - \mu^{t,\alpha^{t,j}}\right]\end{aligned}$$

we can bound the two expectations individually.

First, to bound $\mathbb{E}\left[\sum_{t=1}^{T}\sum_{j=1}^{B} \mu^{t,i_\star^t} - U^{t,i_\star^t}\right]$, we note that $\mu^{t,i_\star^t} - U^{t,i_\star^t}$ is negative with high probability. Also, the maximum value this can take is bounded by 2 as $\mu^{t,i}, U^{t,i} \in [-1, 1]$. Therefore, we have

$$\mathbb{E}\left[\sum_{t=1}^{T}\sum_{j=1}^{B} \mu^{t,i_\star^t} - U^{t,i_\star^t}\right] \leq \left(\sum_{t=1}^{T}\sum_{j=1}^{B} 0 \cdot \mathbb{P}(\mu^{t,i_\star^t} <= U^{t,i_\star^t}) + 2 \cdot \mathbb{P}(\mu^{t,i_\star^t} > U^{t,i_\star^t})\right) \leq 2TB \cdot \frac{1}{2T} = B.$$

Next, to bound $\mathbb{E}\left[\sum_{t=1}^{T}\sum_{j=1}^{B} U^{t,\alpha^{t,j}} - \mu^{t,\alpha^{t,j}}\right]$ we decompose it similar to the above:

$$\begin{aligned}\mathbb{E}\left[\sum_{t=1}^{T}\sum_{j=1}^{B} U^{t,\alpha^{t,j}} - \mu^{t,\alpha^{t,j}}\right] &\leq \left(\sum_{t=1}^{T}\sum_{j=1}^{B} 2\mathbb{P}(U^{t,\alpha^{t,j}} - \mu^{t,\alpha^{t,j}} > 2d^{t,i})\right) + \left(\sum_{t=1}^{T}\sum_{j=1}^{B} 2d^{t,i}\right) \\ &\leq B + \left(\sum_{t=1}^{T}\sum_{j=1}^{B}\sqrt{\frac{32\log(MT^2)}{1 \vee |H_{t,\alpha^{t,j}}|}}\right)\end{aligned}$$

where recall that $|H_{t,i}|$ is the number of samples collected using algorithm i in rounds $\leq t$.

To bound the summation, we utilize the fact that $\frac{1}{1 \vee |H_{t,i}|} \leq \frac{B}{k}$ for each $k \in [|H_{t,i}|, |H_{t+1,i}|]$, since $|H_{t+1,i}| - |H_{t,i}| \leq B$. As a result, we get

$$\begin{aligned}&\sum_{t=1}^{T}\sum_{j=1}^{B}\sqrt{\frac{32\log(MT^2)}{1 \vee |H_{t,\alpha^{t,j}}|}} \\ &\leq \sum_{t=1}^{T}\sum_{i=1}^{M}\sum_{k=1}^{|H_{T,i}|}\sqrt{\frac{32\log(MT^2) \cdot B}{k}} \\ &\leq O(\sqrt{B(\log T + \log M)}\sum_{i=1}^{M}\sqrt{|H_{T,i}|}) \\ &\leq O(\sqrt{B(\log T + \log M)}) \cdot O(\sqrt{BMT}) = O(B\sqrt{MT(\log T + \log M)})\end{aligned}$$

where last two inequalities follow from simple algebra and the fact that $\sum_{i=1}^{M} |H_{T,i}| = TB$.

Finally, to combine all of the bounds above, we get $BR(\texttt{TAILOR}) \leq B + B + O(B\sqrt{MT(\log T + \log M)}) = O(B\sqrt{MT(\log T + \log M)})$.

C Time Complexity

Let N_{train} denote the total neural network training. The time complexity of collecting each batch for each active learning algorithm $\mathcal{A}_i$ can be separated into P_i and Q_i, which are the computation complexity for preprocessing and selection of each example respectively. As examples of preprocessing, BADGE [Ash et al., 2019] computes gradient embeddings, SIMILAR [Kothawade et al., 2021] further also compute similarity kernels, GALAXY [Zhang et al., 2022] constructs linear graphs, etc. The selection complexities are the complexities of each iteration of K-means++ in BADGE, greedy submodular optimization in SIMILAR, and shortest shortest path computation in GALAXY. Therefore, for any individual algorithm $\mathcal{A}_\rangle$, the computation complexity is then $O(N_{train} + TP_i + TBQ_i)$ where T is the total number of rounds and B is the batch size. When running `TAILOR` , as we do not know which algorithms are selected, we provide a worst case upper bound of $O(N_{train} + T \cdot (\sum_{i=1}^{M} P_i) + TB \cdot \max_{i \in [M]} Q_i)$, where the preprocessing is done for every candidate algorithm. In practice, some of the preprocessing operations such as gradient embedding computation could be shared among multiple algorithms, thus only need to be computed once. While the computation of rewards and Thompson sampling updates incur some extra complexity, they are usually dominated in practice by the complexity of neural network training and running each candidate algorithm.

D Study of Candidate Algorithms

We compare the performance when we use the following two sets of candidate algorithms:

1. **Active learning algorithms only:** Uncertainty sampling, GALAXY and EMAL for multi-label classification; Uncertainty sampling, GALAXY and BADGE for multi-class classification.
2. **Active learning and search algorithms:** Uncertainty sampling, GALAXY, MLP, EMAL and Weak Sup for multi-label classification; Uncertainty sampling, GALAXY, MLP, BADGE and Modified Submodular for multi-class classification.

Note Modified Submodular is classified as an active search algorithms since we are using a balanced set of training examples as the conditioning set. We are effectively searching for examples similar to the ones that are annotated in these classes.

As shown in Figures 5 and 6, regardless of the meta algorithm, the performance is better when using active learning algorithms as candidates only. Nonetheless, even with active search algorithms as candidates, `TAILOR` still outperforms other meta active learning algorithms.

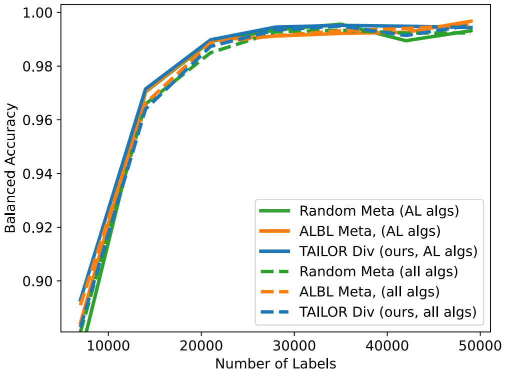


Figure 5: SVHN, Balanced Accuracy

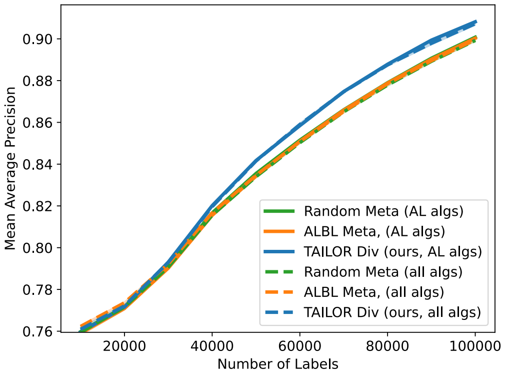
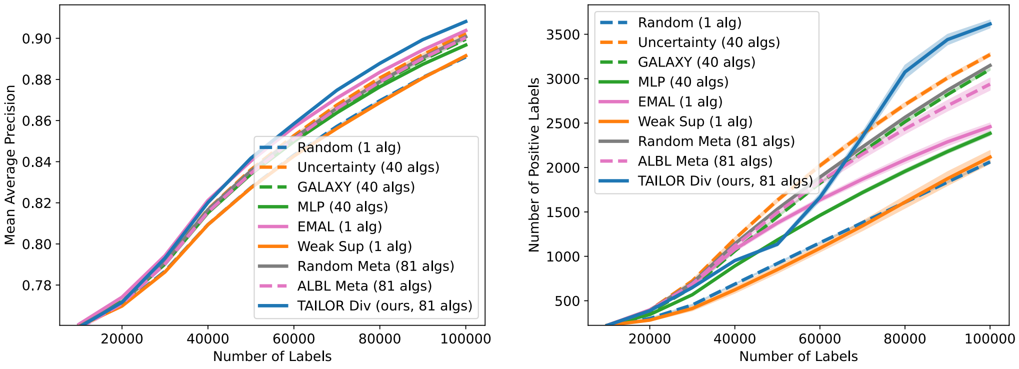


Figure 6: CelebA, mAP

E Full Results

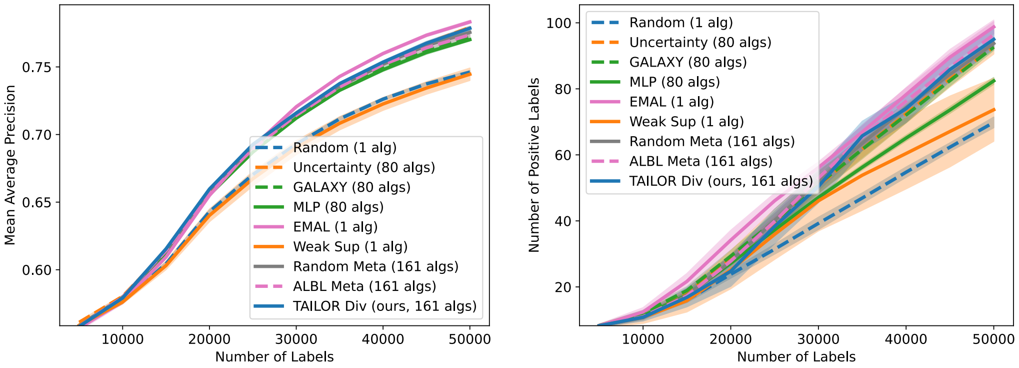
All of the results below are averaged from four individual trials except for Imagenet, which is the result of a single trial.

E.1 Multi-label Classification



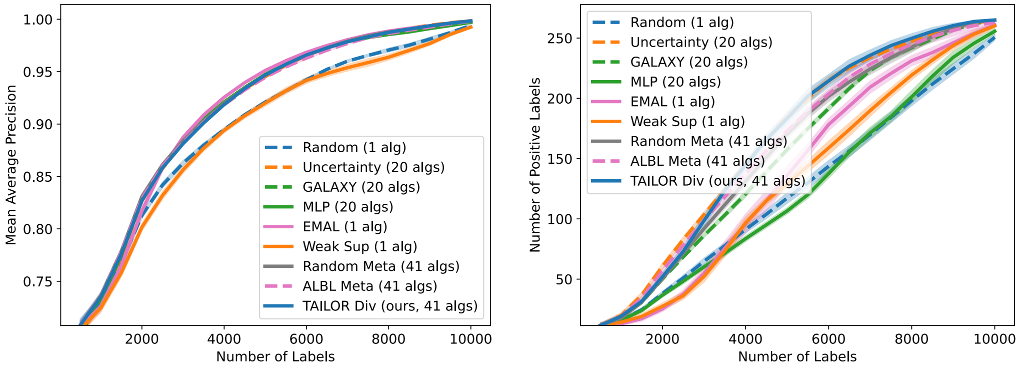
(a) Mean Average Precision (b) Number of Labels in Rarest Class

Figure 7: CelebA



(a) Mean Average Precision (b) Number of Labels in Rarest Class

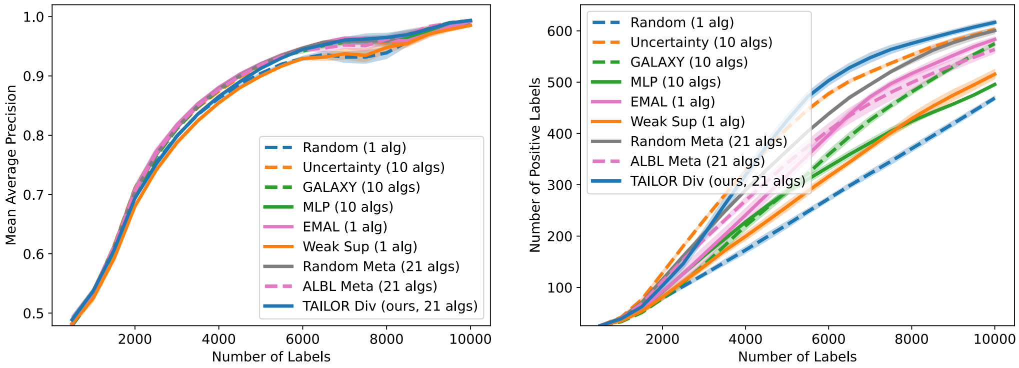
Figure 8: COCO



(a) Mean Average Precision

(b) Number of Labels in Rarest Class

Figure 9: VOC

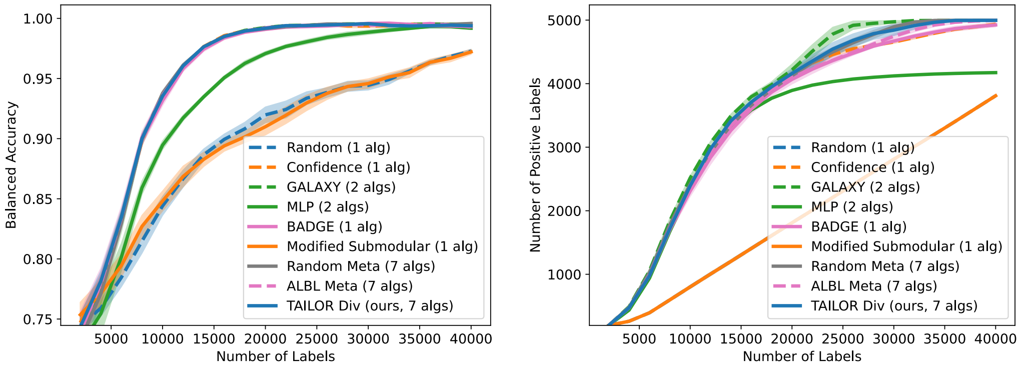


(a) Mean Average Precision

(b) Number of Labels in Rarest Class

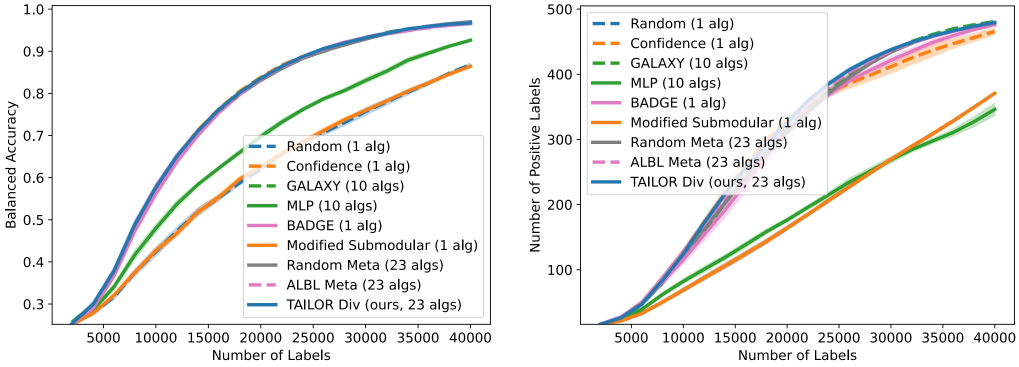
Figure 10: Stanford Car

E.2 Multi-class Classification



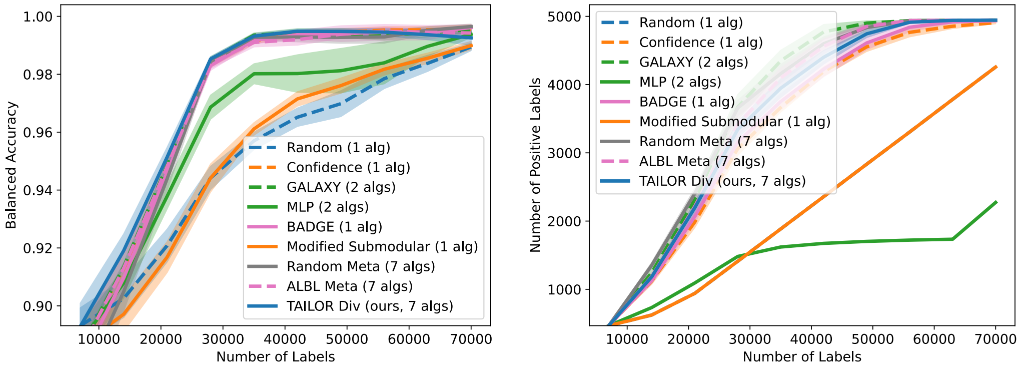
(a) Balanced Accuracy (b) Number of Labels in Rarest Class

Figure 11: CIFAR-10, 2 classes



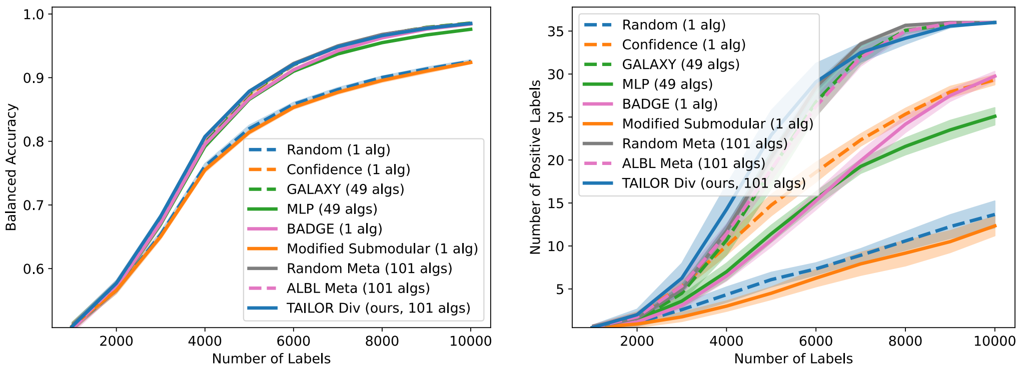
(a) Balanced Accuracy (b) Number of Labels in Rarest Class

Figure 12: CIFAR-100, 10 classes



(a) Balanced Accuracy (b) Number of Labels in Rarest Class

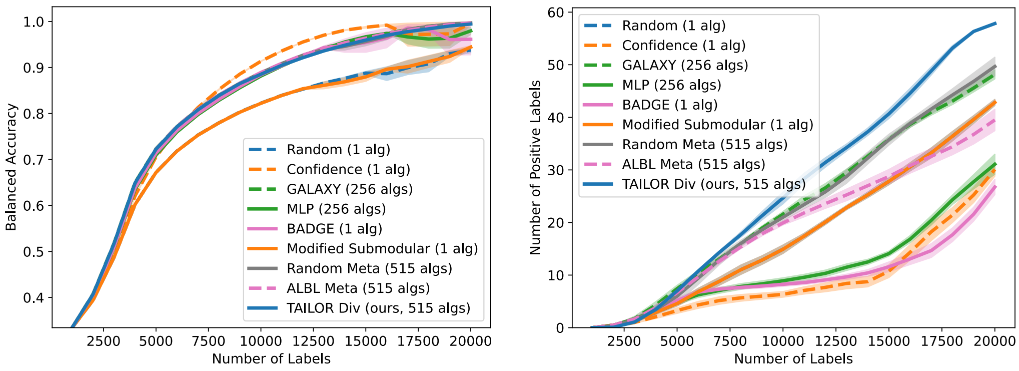
Figure 13: SVHN, 2 classes



(a) Balanced Accuracy

(b) Number of Labels in Rarest Class

Figure 14: Kuzushiji-49



(a) Balanced Accuracy

(b) Number of Labels in Rarest Class

Figure 15: Caltech256

E.3 Multi-label Search

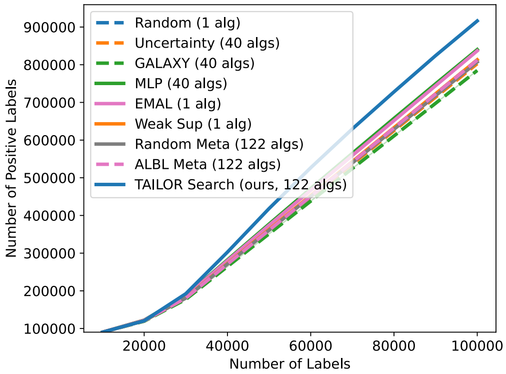


Figure 16: CelebA, Total Number of Positive Labels

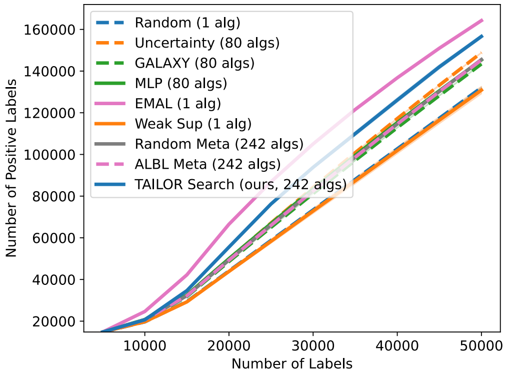


Figure 17: COCO, Total Number of Positive Labels

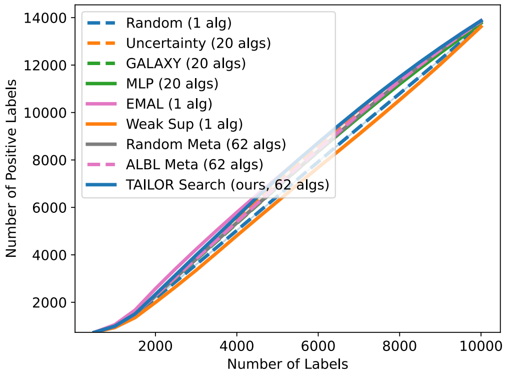


Figure 18: VOC, Total Number of Positive Labels

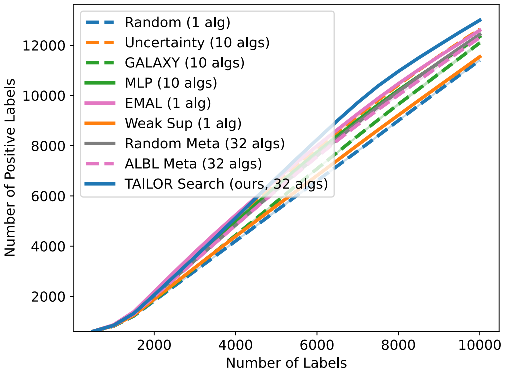


Figure 19: Stanford Car, Total Number of Positive Labels

F What Algorithms Does `TAILOR` Choose?

In the following two figures, we can see `TAILOR` chooses a non-uniform set of algorithms to focus on for each dataset. On CelebA, `TAILOR` out-perform the best baseline, EMAL sampling, by a significant margin. As we can see, `TAILOR` rely on selecting a *combination* of other candidate algorithms instead of only selecting EMAL.

On the other hand, for the Stanford car dataset, we see `TAILOR` 's selection mostly align with the baselines that perform well especially in the later phase.

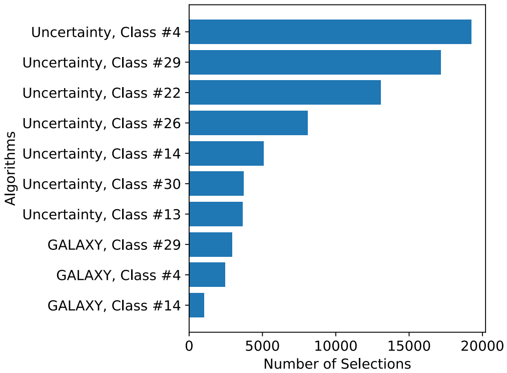


Figure 20: `TAILOR` Top-10 Most Selected Candidate Algorithms on CelebA Dataset

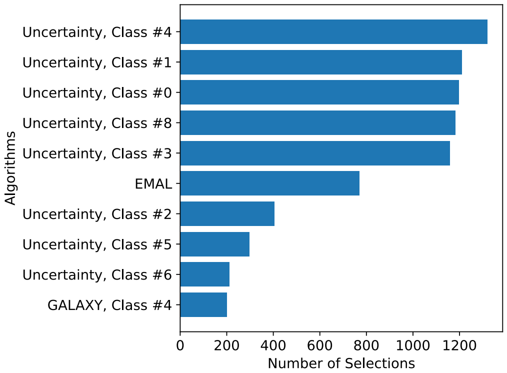


Figure 21: `TAILOR` Top-10 Most Selected Candidate Algorithms on Stanford Car Dataset

In the following figures, we plot the number of times the most frequent candidate algorithm is chosen. As can be shown, `TAILOR` chooses candidate algorithm much more aggressively than other meta algorithms in eight out of the ten settings.

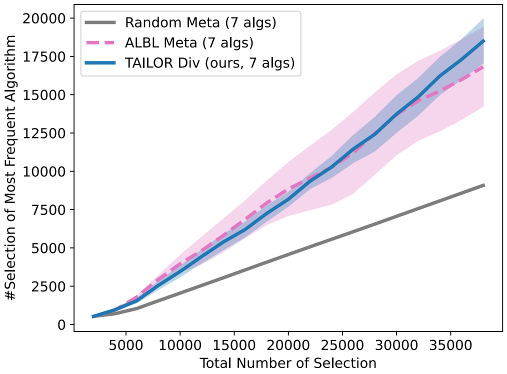


Figure 22: CIFAR-10, 2 Classes, Number of Pulls of The Most Frequent Selection

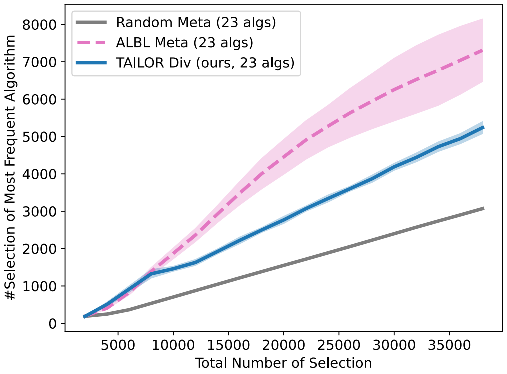


Figure 23: CIFAR-100, 10 Classes, Number of Pulls of The Most Frequent Selection

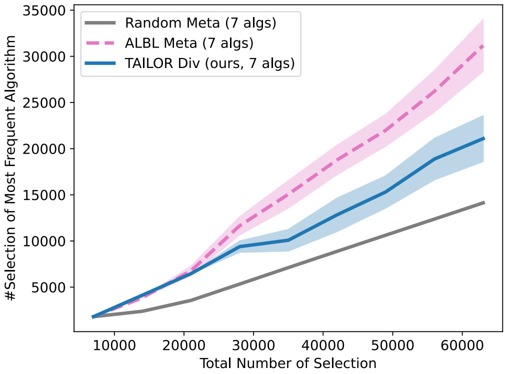


Figure 24: SVHN, 2 Classes, Number of Pulls of The Most Frequent Selection

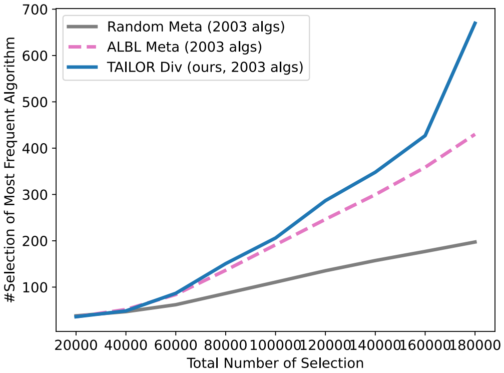


Figure 25: ImageNet-1k, Number of Pulls of The Most Frequent Selection

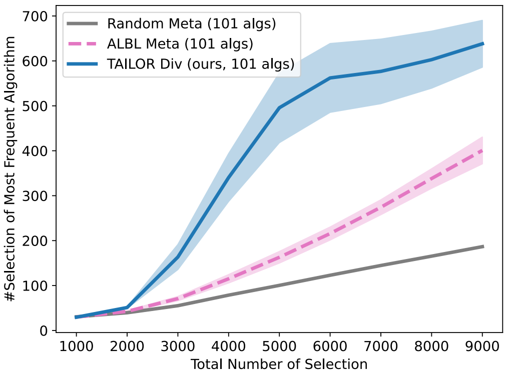


Figure 26: Kuzushiji-49, Number of Pulls of The Most Frequent Selection

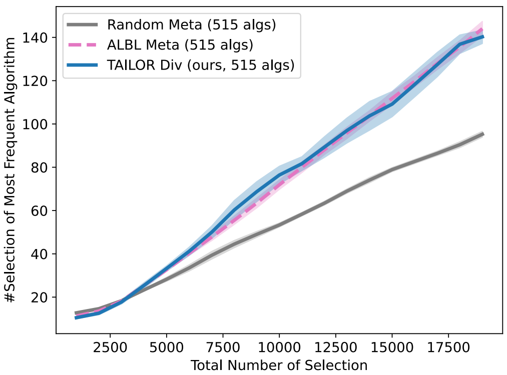


Figure 27: Caltech256, Number of Pulls of The Most Frequent Selection

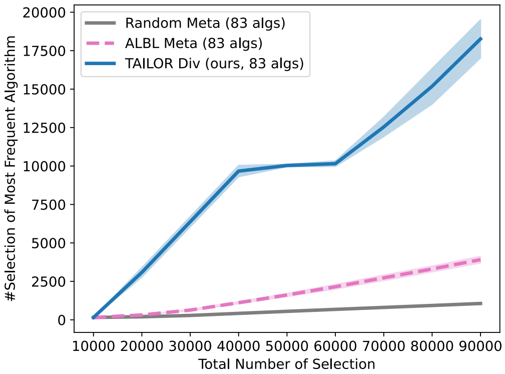


Figure 28: CelebA, Number of Pulls of The Most Frequent Selection

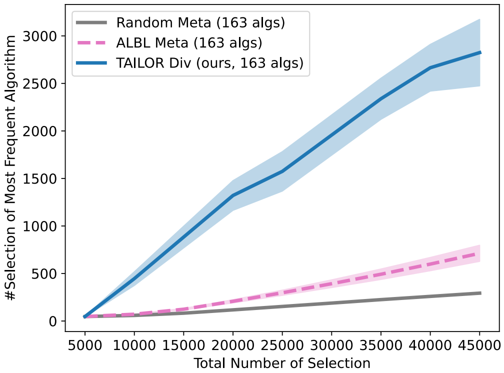


Figure 29: COCO, Number of Pulls of The Most Frequent Selection

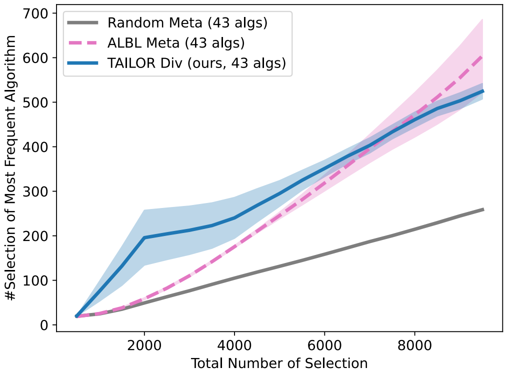


Figure 30: VOC, Number of Pulls of The Most Frequent Selection

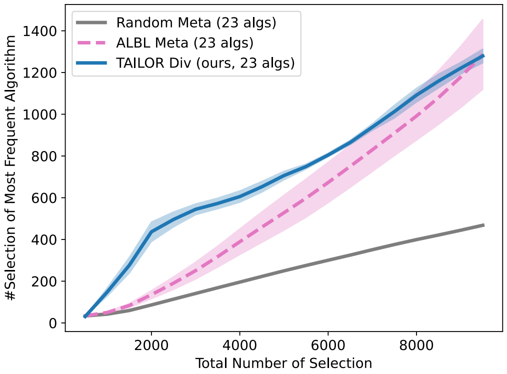


Figure 31: Stanford Car, Number of Pulls of The Most Frequent Selection