WASSAL: WASSERSTEIN DISTANCE-BASED SOFT SUBSETTING FOR ACTIVE LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce Wasserstein Distance-Based Soft Subsetting for Active Learning (WASSAL), an approach that enhances active learning for image classification by explicitly using distributional information when selecting samples for expert labeling. Traditional AL methods often overlook this perspective and rely mainly on model uncertainty or geometric diversity. The key to our methodology is a Weighted Wasserstein adaptation formulation, which learns class-specific weights for the unlabeled samples so that the weighted samples are distributionally close to the labeled datapoints of a given class. WASSAL employs these learned weights in two ways: (i) samples with the least weights across classes are treated as most confusing from a distributional perspective and form the query set for expert labeling; (ii) samples with the highest weights are treated as typical/landmark examples of their respective classes and can be soft-labeled without expert consultation. We propose a novel weighted loss term involving these soft-labeled examples in the next training round. Hence WASSAL not only leverages expert knowledge on the few queried examples, but also utilizes information implicitly available in non-queried examples. Our formulation extends the Wasserstein core-set view of Mahmood et al. (2022) by making the matching class-conditional. Across five benchmarks, WASSAL and WASSALWITHSOFT lead every budget on Caltech-101 and STL-10, remain within a few points of the best baselines on CIFAR-10 and PneumoniaMNIST, and hold their own under the high-variance SVHN pool, showing that distributional soft subsetting is a practical addition to the active learning toolbox.

1 INTRODUCTION

Active Learning (AL) frameworks (Ren et al., 2021) are pivotal in optimizing the labeling process for large datasets, especially for image classification. These frameworks are designed to intelligently select the most informative samples for expert annotation. This task becomes increasingly crucial as deep learning models seek to achieve high performance with limited labeled data. The core of AL lies in its ability to evaluate each sample's potential to enhance a model's performance. This is typically achieved by assessing the uncertainty or margin of individual samples relative to a partially trained model, focusing on those exhibiting the most ambiguity or proximity to decision boundaries. Such strategic selection allows for iterative model training on an expanding labeled dataset, promising an efficient path to model refinement.

Despite their effectiveness, current state-of-the-art AL methods (Yang et al., 2015; Killamsetty et al., 2021b; Kothawade et al., 2022; Agrawal et al., 2021; Sener & Savarese, 2018; Killamsetty et al., 2021a; Ash et al., 2020) primarily focus on identifying and labeling the "worst-performing" samples, neglecting insights that could be gleaned from distributional aspects of the data. Distributional and geometric information has recently been used for selection itself, most prominently by Wasserstein-based core-set selection (Mahmood et al., 2022) and coverage-maximizing methods (Yehuda et al., 2022; Bae et al., 2024; 2025), which are especially effective at low budgets (Hacohen et al., 2022). A comparison of the unlabeled set with the given labeled set under a distributional perspective may provide global information regarding the "typical" as well as "confusing" examples. Such information can be more reliable than queries chosen only from the perspective of the model being learned. Hence, to optimize the labeling process of large datasets, considering distributional information may be critical. The "typical/easy" or "difficult" examples from a distributional perspective (Wang et al., 2021) can be very informative.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

In light of this observation, we put the well-known Wasserstein distance (Rüschendorf, 1985) to a new use in sample selection. Our proposed methodology, Wasserstein Distance-Based Soft Subsetting for Active Learning (WASSAL), leverages distributional information to make informed decisions about which samples are to be expert labeled. The distance itself is long established; what is new is how we use it. More specifically, the key idea is to learn class-specific weights which, when used to re-weight the examples in the unlabeled set, induce a distribution that is closest to the distribution of the respective class's examples in the labeled set. This is posed as a convex optimization problem using the 2-Wasserstein distance as the metric between distributions and solved using projected gradient descent. We expect high weights to correspond to landmark examples in each class, whereas examples with low weights across classes are expected to correspond to those examples whose label identity is unclear given the labeled set. Hence we consider the latter (low-weight) examples to be ideal candidates for expert labeling. Observe that the proposed strategy for query selection is based on class-conditional distributional information, going beyond the current model's perspective alone.

Another innovation in our proposal is to include the valuable information present in the identified potential landmark examples. This is done by including a novel weighted loss term in the training of the machine learning model in the subsequent round. The weighted loss includes the learned weights in order to enforce soft matching of predicted labels with the corresponding class label. The higher the weight, the more strongly the predicted labels are matched with the corresponding label and vice versa. Thus, WASSAL distinguishes itself by not only focusing on the challenging or uncertain samples but also by incorporating those that are typical or representative of their respective classes. This dual approach ensures a more balanced and informative active learning process, potentially reducing the number of rounds/expert queries needed to achieve high model performance. We denote the full method (query selection plus soft loss, with $\beta = 0.3$) by WASSALWITHSOFT, and treat query-only WASSAL as an ablation where those runs are available.

Our contributions through the development of WASSAL include:

- A sample selection strategy (Section 3.2.1) that gives the established Wasserstein distance a new, class-conditional role in active learning. The matching is learned separately for each class, so the selection signal reflects distributional structure that the perspective of the model being learned alone cannot reveal.
- An innovative weighted soft-loss mechanism (Sections 3.2 and 3.3) that helps utilize the information implicitly present in the potential landmark examples identified in the weight learning step. This incorporation of the weighted loss term critically helps our AL method to utilize information from labels not sent to expert labeling.
- Empirical validation (Section 4) of our methodology on five benchmarks—CIFAR-10, SVHN, PneumoniaMNIST, STL-10, and Caltech-101—comparing WASSALWITHSOFT against the classical strategies (BADGE, GLISTER, GradMatch-TSS, CoreSet, uncertainty/margin/least-confidence sampling, and random), the distributionally motivated Wasserstein-IP selection of Mahmood et al. (2022), and recent low-budget state of the art including MaxHerding (Bae et al., 2024) and UHerding (Bae et al., 2025), reporting mean final accuracy over three seeds where available.

By integrating distributional insights with active learning, WASSAL aims to make active learning more representative and effective in leveraging unlabeled data for model improvement.

2 RELATED WORK

Active Learning (AL) frameworks have long aimed to improve labeling efficiency for large datasets through various strategies. Uncertainty Sampling (Wang et al., 2021) prioritizes ambiguous samples where the model's predictions are least confident. CoreSet (Sener & Savarese, 2018) focuses on selecting representative samples that span the dataset's feature space. In the case of Gradient Matching (Killamsetty et al., 2021a), a subset is selected by matching the gradient updates of a subset with those of the entire dataset to maintain effective learning trajectories. Agrawal et al. (2021) explore least-confidence sampling by focusing on samples with the lowest confidence score, thus aiming to address samples near the decision boundary. In contrast, margin-based sampling (Balcan et al., 2007) chooses samples with the smallest margin between the top two predicted classes, targeting the most uncertain data points. GLISTER (Killamsetty et al., 2021b) employs combinatorial optimization to

select subsets that maximize a surrogate objective function for better generalization. BADGE (Ash et al., 2020) samples in gradient embedding space via k-means++ initialization to balance uncertainty and diversity within each batch.

While these methods focus on uncertainty or representativeness from the perspective of individual samples, WASSAL integrates distributional insights into selection. By leveraging the Wasserstein distance, it identifies uncertain samples while also weighing how representative each sample is of its class. Its weighted loss then ensures that the information in soft-labeled samples is used, reducing the dependency on expert labeling for every informative point.

Low-budget active learning. A recent line of work shows that in the low-budget regime, where the labeled set is only a few times larger than the number of classes, uncertainty estimates of an under-trained model are unreliable and can even underperform random selection (Hacohen et al., 2022). This has motivated representative selection rules: TypiClust (Hacohen et al., 2022) queries cluster centroids of typical examples, ProbCover (Yehuda et al., 2022) greedily maximizes graph coverage of the feature space, and DCoM (Mishal & Weinshall, 2024) combines clustering-based diversity with model confidence. Closest to our setting, Mahmood et al. (2022) select a core-set that minimizes the discrete Wasserstein distance between the labeled set and the unlabeled pool, posed as an integer program and solved approximately with a k-medoids heuristic. MaxHerding (Bae et al., 2024) generalizes ProbCover's coverage objective via kernel herding, and UHerding (Bae et al., 2025) extends it with an uncertainty-weighted coverage that adapts across label budgets. Complementing these rules, Hacohen & Weinshall (2023) learn which strategy best suits a given budget. All of these methods use distributional or geometric information only to *select* samples: every selected example is sent to the expert, and non-selected examples contribute nothing to training. In contrast, WASSAL uses class-conditional Wasserstein matching in the opposite direction—assigning weights to unlabeled samples instead of re-weighting the labeled set—and exploits both ends of the weight spectrum: low-weight samples are queried, while high-weight landmark samples are soft-labeled through a weighted loss. This makes the closest prior work, Wasserstein-IP (Mahmood et al., 2022), a natural baseline; we compare against it and the herding family in Section 4.

Optimal transport in machine learning. Beyond selection, optimal transport has been used for domain adaptation (Cuturi, 2013), generative modeling, and dataset-level distances, with Sinkhorn iterations making the computations scalable. In active learning specifically, transport formulations have mostly appeared as Wasserstein core-set objectives (Mahmood et al., 2022) or distribution matching between queried batches and the pool. WASSAL differs in combining a transport-based selection signal with a transport-derived *training* signal: the same class-conditional matching problem that ranks confusing samples also yields landmark weights that parameterize a novel weighted loss, so that the unlabeled pool contributes to learning beyond the expert-labeled queries.

3 THE WASSAL FRAMEWORK

In Figure 1, we provide an overview of our WASSAL framework. Central to the methodology is the weighted Wasserstein adaptation strategy, which learns the class-wise weights for each unlabeled sample such that the resultant distribution matches the class-conditional distribution of labeled examples belonging to that class. While examples with low weights across classes are passed on for expert labeling, the high weights are employed in a weighted-loss term as a soft subset for the subsequent round. We set up some preliminaries in Section 3.1 before explaining our approach in detail starting with Section 3.2.

3.1 PRELIMINARIES

Let $\mathcal{X}$ be a set endowed with a distance metric c. The 2-Wasserstein distance between two probability measures μ, ν is defined as follows:

$$W_2(\mu,\nu)^2 = \inf_{\gamma\in\Pi(\mu,\nu)} \int_{\mathcal{X}^2} c^2(x,y)\,\mathrm{d}\gamma(x,y), \tag{1}$$

where $\Pi(\mu,\nu)$ is the set of all joint probability measures with marginals μ, ν. If the given probability measures are discrete, i.e., $\mu = \sum_{i=1}^{m} a_i\delta_{x_i}$ and $\nu = \sum_{i=1}^{n} b_i\delta_{y_i}$, where δ is the Dirac delta function,

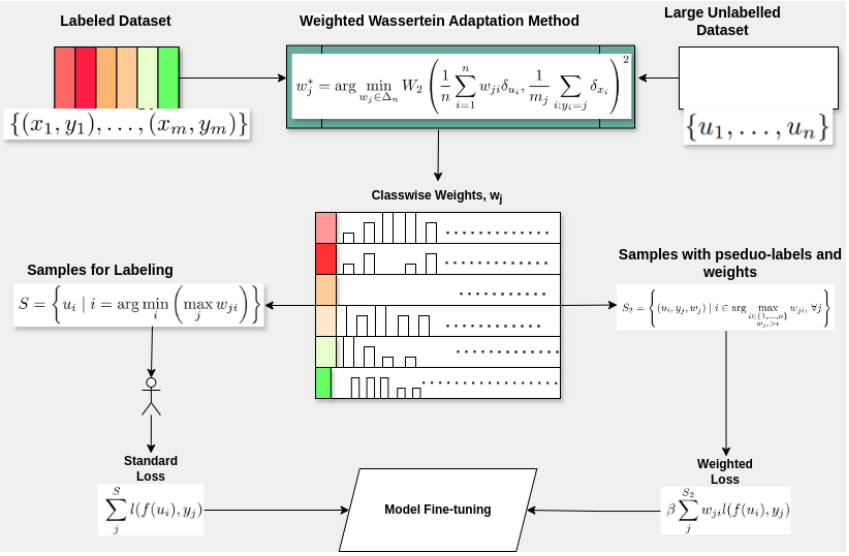


Figure 1: Overview of the Wasserstein Distance-Based Soft Subsetting for Active Learning (WASSAL) framework. The diagram illustrates the key steps involved, ranging from adaptation of the labeled dataset using the weighted Wasserstein adaptation method, and the selection of samples with low weights for labeling, to the final model fine-tuning incorporating both standard and weighted loss.

then the Wasserstein problem reduces to:

$$\min_{\gamma \in \mathbb{R}_+^{m \times n}} \quad \langle \gamma, C \rangle_F$$
$$\text{s.t.} \quad \gamma \mathbf{1} = a, \ \gamma^\top \mathbf{1} = b, \tag{2}$$

where C is the $m \times n$ matrix with the i, j entry as $c^2(x_i, y_j)$. The challenge in solving this linear program is its computational complexity, scaling as $O(n^3 \log n)$ (if $m = n$). To mitigate this, Cuturi (Cuturi, 2013) introduced an entropic regularization:

$$\min_{\gamma \in \mathbb{R}_+^{m \times n}} \quad \langle \gamma, C \rangle_F - \varepsilon \mathcal{H}(\gamma)$$
$$\text{s.t.} \quad \gamma \mathbf{1} = a, \ \gamma^\top \mathbf{1} = b, \tag{3}$$

where $\mathcal{H}(\gamma)$ represents the entropy of γ. The Sinkhorn–Knopp algorithm tackles this regularized problem efficiently. It approaches the solution as an ε-strongly convex function by iteratively updating scaling variables, starting from an arbitrary positive vector. It is memory efficient and can be computed in mini-batches, as analyzed by Fatras et al. (2019), so it scales to large datasets.

3.2 WEIGHTED WASSERSTEIN ADAPTATION

At the core of our methodology is a Wasserstein adaptation formulation, which we describe in this section. Let $\mathcal{L} = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ be the (small) labeled set and $\mathcal{U} = \{u_1, \ldots, u_n\}$ be the (large) unlabeled set available in the context of active learning at a particular round.

At an intuitive level, the idea behind Wasserstein adaptation is as follows. We wish the label information provided in the labeled set to be "adapted" to the unlabeled set, leading to label information among the unlabeled examples. Suppose we magically knew the subset of unlabeled examples belonging to a particular class; then these examples would be distributionally close to those in the labeled set. On the other hand, an arbitrary subset of unlabeled examples is expected to have no distributional resemblance to those in the labeled set. Indeed this is true as long as the class-conditional distributions are not arbitrarily close to each other, which is reasonable to assume in a classification scenario. Hence finding a subset of unlabeled examples that is distributionally closest to the labeled examples of a particular class is one way of finding the subset of unlabeled examples belonging to

that class. We implement this intuitive idea by finding soft-subsets via weights instead of hard subsets (i.e., binary weights). This has a two-fold advantage: (i) real-valued (soft) weights are more informative than binary weights and can be employed in the AL methodology; (ii) optimization is easier with soft weights.

Our novel strategy hence aims to find class-specific weights for the unlabeled examples, such that the weighted samples are distributionally close to the examples from a particular class in the labeled set. If w_{ji} denotes the weight for the i-th unlabeled example for belonging to the j-th class, then the corresponding re-weighted distribution of the unlabeled set is given by $\sum_{i=1}^{n} w_{ji}\delta_{u_i}$, where $w_j \in \Delta_n$ and $\Delta_n \equiv \{\alpha \in \mathbb{R}^n \mid \alpha \geq 0, \alpha^\top \mathbf{1} = 1\}$ is the n-dimensional simplex. We wish to find these weights such that this re-weighted distribution is as close as possible to the j-th class-conditional as per the labeled set, i.e., $\frac{1}{m_j}\sum_{i:y_i=j}\delta_{x_i}$, where $m_j = \left|\{i \mid y_i = j\}\right|$. We wish to match these two distributions using the 2-Wasserstein distance. Hence the overall formulation for finding these weights is:

$$w_j^* = \arg\min_{w_j \in \Delta_n} W_2\left(\sum_{i=1}^{n} w_{ji}\delta_{u_i}, \frac{1}{m_j}\sum_{i:y_i=j}\delta_{x_i}\right)^2. \tag{4}$$

If w_{ji}^* is large, it means that u_i is a typical (landmark) example from the j-th class. If w_{ji}^* is low for all j, then it is perhaps the most borderline (confusing) example.

We minimize equation 4 using a projected gradient descent solver. The gradient is numerically obtained through automatic differentiation, which in turn uses a standard Sinkhorn implementation for computing the Wasserstein distance. After taking the gradient step, the weights are projected onto the simplex (Duchi et al., 2008). The algorithm can be used to solve equation 4 for every class individually.

Relation to Wasserstein-IP selection. Our formulation directly extends the Wasserstein core-set view of Mahmood et al. (2022). There, one selects a subset of the unlabeled pool that minimizes the Wasserstein distance to the full pool distribution—a pool-level, unsupervised matching posed as an integer program over binary selection variables and approximated with a k-medoids heuristic. Here, matching is class-conditional, with each class's labeled distribution defining its own target; the weights are continuous points on the simplex instead of binary indicators; and the optimal weights are reused beyond selection: low weights identify the confusing examples to query, while high weights define landmark examples that enter the training loss as soft labels. Selection and training thus share a single transport problem.

The learned weights can be used in two different ways in the context of active learning. Firstly, they can be used to select samples for expert labeling (Section 3.2.1). Secondly, we propose to use the label information adapted from the labeled set via a novel weighted loss term in the subsequent learning round (Section 3.3).

3.2.1 Selection of samples for expert labeling

At the selection stage we examine the simplex of each class and prioritize the instances with the smallest weights. Low weights mark samples that resemble a class's labeled data yet carry little distributional confidence for it, so querying them targets the points that most need expert attention. This strategy can be expressed as

$$S_1 = \text{top few } i \text{ in ascending order of } \max_j w_{ji}. \tag{5}$$

3.3 Incorporation of weighted loss term

Diverging from conventional Active Learning methodologies, in Wassal we introduce a novel approach to harness the potential of the highly weighted samples. This strategy involves a selective process that identifies samples displaying the highest values within their class-specific weights, as per the weighted Wasserstein adaptation. Since these are potential landmark examples for the respective classes, we propose including them in the labeled set, while preserving the weights to reflect the fact that they are not expert labeled:

$$S_2 = \cup_j \left\{(u_i, y_j, w_{ji}) \mid i \in \text{ top few in descending order of } w_{ji}\right\}. \tag{6}$$

Algorithm 1 WASSAL round: sample selection and weighted-soft training

Require: Unlabeled set $\mathcal{U}$, labeled set $\mathcal{L}$, learning rate α, soft-loss weight β, budget b

1: Initialize $w_{i,j}^{(0)} = 1/n$ for every class c_i and every $u_j \in \mathcal{U}$
2: Find the optimal w^* by solving equation 4 with projected gradient descent
3: Select the **query** set S_1 with equation 5
4: Select the **soft** subset S_2 with equation 6
5: Obtain labels for S_1 from the oracle O
6: **for** each training iteration **do**
7: Minimize the sum of equation 8 and equation 7, namely $\sum_{i \in S_1} l(f(u_i), y_i') + \sum_{i=1}^{m} l(f(x_i), y_i) + \beta \sum_{(u_i, y_j, w_{ji}) \in S_2} w_{ji}\, l(f(u_i), y_j)$
8: **end for**

The weights themselves play a crucial role in determining the influence of each sample on the learning process. Samples with higher weights, indicative of greater confidence or importance assigned by the adaptation step, contribute more significantly to the training process. This is formalized via a novel weighted loss term:

$$\beta \sum_{(u_i, y_j, w_{ji}) \in S_2} w_{ji}\, l(f(u_i), y_j), \tag{7}$$

where f is the classifier being learned and l is the relevant loss function for learning the classifier. We denote by β the hyperparameter used to control the overall contribution of weighted loss equation 7 to the total loss. The labeled set for the subsequent round includes the expert-labeled pairs (u_i, y_i') for $i \in S_1$, where y_i' are the expert-provided labels. Hence the standard loss is represented as

$$\sum_{i \in S_1} l\left(f(u_i), y_i'\right) + \sum_{i=1}^{m} l\left(f(x_i), y_i\right). \tag{8}$$

This weighted training approach is a strategic enhancement over traditional AL practices. By including these well-matched samples in the training phase, WASSAL leverages their inherent information to refine and strengthen the model's learning trajectory. The overall WASSAL methodology is summarized in Algorithm 1. We call the full pipeline WASSALWITHSOFT; query-only selection without equation 7 is denoted WASSAL.

4 EXPERIMENTS

4.1 SETUP

We conducted experiments on five class-imbalanced image classification benchmarks to evaluate our proposed methodology: CIFAR-10 (Krizhevsky & Hinton, 2009), SVHN (Netzer et al., 2011), PneumoniaMNIST (Yang et al., 2023), STL-10 (Coates et al., 2011), and Caltech-101 (Fei-Fei et al., 2004). We implemented the following common experimental setup across datasets. A small, randomly selected subset of labeled data was used to initialize the model (Initial Training Set). The remaining data constituted the unlabeled pool from which samples were selected for annotation during the active learning process (Unlabeled Pool). Model performance was evaluated using accuracy as the primary metric. Table 1 summarizes the splits. We limited the unlabeled pool to roughly 4k–4.7k points so that the Wasserstein distance can be computed over the full pool in one batch. Our experiments used NVIDIA GPUs and ResNet-18 as the primary architecture (STL-10 uses a ResNet-18-STL variant). Results are mean final test accuracy at the last AL round, averaged across three seeds (exp2–4; seeds 48/86/28) in most cells. On STL-10 the classical strategies at budgets 100–200 average two seeds, since the final repeat there fell outside our compute window. We report final accuracy, since averaging accuracy gain is sensitive to whether a run reuses a cached round-0 checkpoint or rebuilds it. The soft-loss hyperparameter is $\beta = 0.3$, used both in WASSALWITHSOFT and, as an ablation, on top of each classical query rule. Baselines include BADGE (Ash et al., 2020), GLISTER (Killamsetty et al., 2021b), GradMatch-TSS (Killamsetty et al., 2021a), CoreSet (Sener & Savarese, 2018), margin, least-confidence, and random sampling. Beyond these classical strategies, we compare against three recent distributionally motivated selection methods: Wasserstein-IP (Mahmood et al., 2022), the closest prior work to ours, which minimizes the discrete Wasserstein

Table 1: Experimental setup for datasets: initial labeled set, unlabeled pool, and number of AL rounds.

Dataset	Labeled / Unlabeled	Rounds
CIFAR-10 (Krizhevsky & Hinton, 2009)	200 (20/cls) / 4000 (400/cls)	10
SVHN (Netzer et al., 2011)	200 (20/cls) / 4000 (400/cls)	10
STL-10 (Coates et al., 2011)	200 (20/cls) / 4700 (470/cls)	10
PneumoniaMNIST (Yang et al., 2023)	100 (50/cls) / 4000 (1000:3000)	10
Caltech-101 (Fei-Fei et al., 2004)	816 (8/cls) / 4590 (45/cls)	8

Table 2: Caltech-101 mean final accuracy (%). Best of the listed methods in bold.

Strategy	25	50	100	175	200
WassalWithSoft	**20.69**	**22.66**	**24.85**	**25.94**	**27.40**
Wassal	15.59	19.09	18.87	20.84	22.22
BADGE	15.96	20.18	18.91	23.75	24.37
GLISTER	14.28	17.85	21.60	23.86	24.15
GradMatch-TSS	15.70	19.16	21.38	24.01	25.17
CoreSet	15.45	18.07	18.36	20.84	22.00
LeastConf	16.39	16.90	18.07	19.53	22.15
Margin	15.05	17.81	21.35	22.95	22.95
Random	16.10	20.26	20.26	23.75	24.15
DCoM	19.23	21.31	22.15	22.91	24.15
MaxHerding	16.10	20.07	22.70	23.53	23.39
UHerding	15.52	20.15	20.40	22.00	22.19
Wasserstein-IP	15.59	17.30	19.89	23.28	24.44

distance between the labeled and unlabeled sets via integer programming; MaxHerding (Bae et al., 2024), which greedily maximizes a kernelized coverage objective; and UHerding (Bae et al., 2025), which augments that coverage with uncertainty and adapts across label budgets. We also report DCoM (Mishal & Weinshall, 2024) as a recent low-budget representative. For a controlled comparison, all methods use the same task-model features, budgets, and training protocol, and differ only in the selection rule. Budgets: CIFAR-10/STL-10 $\{25, 50, 100, 125, 150, 175, 200\}$; SVHN/Caltech-101 $\{25, 50, 100, 175, 200\}$; PneumoniaMNIST $\{20, \ldots, 100\}$ in steps of 10. Where available, we also report query-only Wassal as an ablation alongside WassalWithSoft.

4.2 Results

Our experiments ask whether a class-conditional transport signal can complement the model's own uncertainty across label budgets, and where that signal earns its keep. Tables 2 and 3 give the per-budget results for the two datasets where the Wassal family is strongest, and Figure 2 plots the curves; a cross-dataset summary and the remaining per-budget tables are in Appendix A.

Caltech-101. Caltech-101 is the most demanding split in the study (Table 2): 102 classes with only eight labeled images per class at the start. This is precisely the regime in which class-conditional matching has the most to offer, and WassalWithSoft leads at every budget, from 20.69% at budget 25 to 27.40% at 200%, ahead of the next best method at each budget. The soft loss is what carries the method here: the landmark pseudo-labels drawn from the unlabeled pool compensate for the very thin per-class labeled measures, and the weighted loss turns a weak selection signal into a reliable one. That a class-conditional transport objective holds up on the split with the largest label space is the clearest single result in the paper.

STL-10. STL-10 is the second dataset where the Wassal family sets the pace (Table 3). Wassal leads at every budget from 50 upward (23.88% to 27.66%), edging the strongest classical baselines at each budget, and WassalWithSoft takes the smallest budget (23.80%). The two ingredients cover the range between them: the soft loss is at its best while the labeled seed is tiny, and the class-conditional query rule carries the method once labels accumulate. The coverage-based herding

Table 3: STL-10 mean final accuracy (%). Best of the listed methods in bold.

Strategy	25	50	100	125	150	175	200
WassalWithSoft	**23.80**	23.75	23.97	25.20	24.82	25.24	25.88
Wassal	21.34	**23.88**	**25.20**	**26.02**	**26.27**	**26.66**	**27.66**
BADGE	20.42	22.49	24.49	24.85	25.29	26.41	27.36
GLISTER	21.39	22.71	24.89	25.16	25.42	26.46	27.62
GradMatch-TSS	21.24	22.62	24.86	25.51	25.59	26.26	27.44
CoreSet	21.22	22.43	24.14	24.70	24.81	25.21	26.15
LeastConf	18.95	21.17	24.26	25.03	25.36	25.95	27.08
Margin	20.10	21.34	24.27	24.55	24.54	26.25	26.77
Random	20.10	22.25	23.79	24.88	25.21	25.11	26.14
DCoM	21.43	22.30	24.75	24.55	24.93	25.75	26.80
MaxHerding	20.70	22.54	24.18	24.78	24.85	26.13	26.64
UHerding	20.47	22.65	24.53	25.14	25.21	26.04	27.09
Wasserstein-IP	21.10	22.38	24.78	25.07	25.47	26.50	26.81

methods remain competitive in the middle of the range. The classical rows at budgets 100–200 average two seeds here; every other row in the table uses three.

CIFAR-10. CIFAR-10 is a tightly packed field where the ranking shifts with the budget (Table 5). Both Wassal variants stay within a few points of the leader at every budget: WassalWithSoft takes budget 50 (35.97%) and Wassal takes budget 150 (41.67%), while the remaining budgets are shared among strong single baselines—UHerding at 25 (32.79%), Random at 100 (39.38%), MaxHerding at 125 (40.23%), LeastConf at 175 (42.51%), and BADGE at 200 (42.99%). No strategy runs away with the dataset, and the margins among the top methods sit within seed variation, so we read the ranking as mixed, with no clear hierarchy. Wasserstein-IP stays behind the leaders at every budget, which is consistent with its pool-level, class-agnostic matching.

SVHN. SVHN is the most variable pool in our study (Table 6): accuracies move by several points across seeds and several curves are non-monotone. With this much spread the ranking is genuinely mixed, and no method is uniformly best. Wassal is strongest at the smallest budget (28.85%), GradMatch-TSS at 50 (38.22%), Margin at 100 (48.86%), and LeastConf at 175 and 200 (64.86% and 67.99%). The official SVHN train and test splits are drawn from different distributions (Xiao et al., 2023), so a pool matched to the labeled training slice is an imperfect proxy for the test set; distributional matching has the most to offer while labels are scarce. The spread here is a reminder that single-seed comparisons on this pool are unreliable.

PneumoniaMNIST. PneumoniaMNIST is a close contest inside a narrow 81–87% band (Table 7), where the differences at the top are small. GradMatch-TSS is best at budgets 30 and 40 (86.70%/87.29%), UHerding at 20 and 80 (86.86%/86.70%), CoreSet at 50 (85.79%), BADGE at 60 and 100 (86.38%/86.22%), Wassal at 70 (85.79%), and MaxHerding at 90 (86.06%). Wassal closes on the leaders as the labeled seed grows: it is within one to two points of the best method from budget 50 onward and takes the top spot at budget 70. On this skewed binary pool (1000:3000) the top methods sit within about a point of each other at most budgets, and Wasserstein-IP generally sits at the lower end of the band.

4.3 Discussion

Across the five benchmarks no single method dominates, and the field is mixed in a way we find informative. The closest prior work, Wasserstein-IP, does not lead a budget: its pool-level matching is class-agnostic and cannot tell which unlabeled examples belong to which class. Wassal keeps the transport view and makes it class-conditional, lets the weights be continuous points on the simplex, and reuses them as a training signal. The comparison supports each of these moves: class-conditional matching is what carries the method on Caltech-101, the query rule is what carries it on STL-10, and the combination is competitive everywhere else, including on the high-variance SVHN pool.

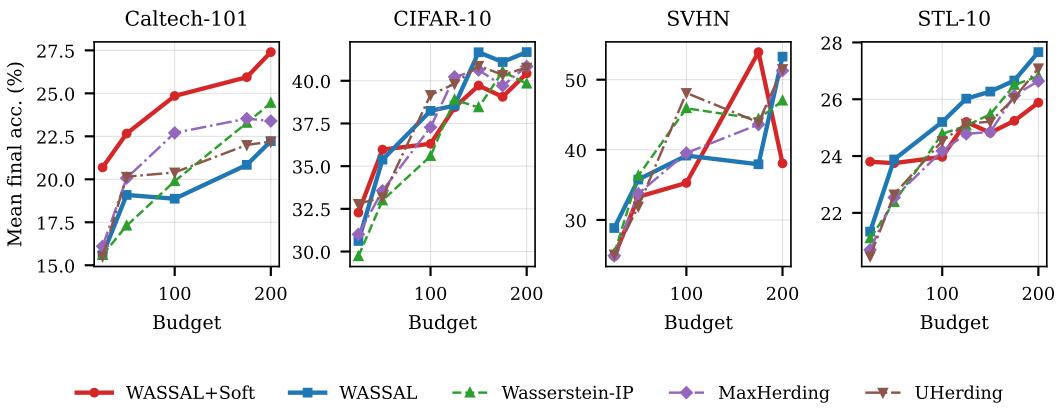


Figure 2: Mean final accuracy across annotation budgets for the WASSAL family and the optimal-transport and herding baselines on Caltech-101, CIFAR-10, SVHN, and STL-10 (three seeds where available). Full per-method results are in Appendix A.

The pattern has a natural reading: class-conditional matching helps most where the per-class labeled measures are thin, as on Caltech-101 and STL-10, and it becomes one useful contribution among several once the labeled seed grows or the pool turns noisy and shifted, as the CIFAR-10, PneumoniaMNIST, and large-budget SVHN results show. This is the behaviour we would want from a complementary signal, and it is why the query rule and the soft loss sit naturally in one pipeline.

We also examine the soft loss as a portable component, applying it on top of the other query rules (Appendix B). Its budget-averaged effect is positive for every rule on Caltech-101 and mixed on CIFAR-10 and PneumoniaMNIST, while on STL-10 and SVHN it helps some rules and hurts others. The pattern matches the rest of our results: soft-labeling landmark examples pays off when class-conditional structure is well supported by the labeled seed, and is close to neutral when the pool is small, noisy, or shifted. Taken together, the soft subsetting idea holds up as a general modification of transport-based selection, one that improves the transport baseline it is built on and can be attached to other query rules when the data support it.

5 Conclusion

In this paper, we introduced WASSAL, a class-conditional extension of Wasserstein core-set selection that turns a single optimal-transport problem into both a query rule and a training signal. Our method addresses key limitations in traditional active learning frameworks by focusing not only on the most uncertain samples but also on representative examples within the dataset. This dual approach—querying distributionally confusing samples while soft-labeling landmark examples through a weighted loss—ensures a balanced and comprehensive use of the unlabeled pool.

Our empirical evaluations on CIFAR-10, PneumoniaMNIST, STL-10, SVHN, and Caltech-101 paint a detailed picture. The WASSAL family leads every budget on STL-10 and Caltech-101, stays within a few points of the best baselines on CIFAR-10 and PneumoniaMNIST, and remains competitive on the high-variance SVHN pool, where classical uncertainty sampling is very strong at the largest budgets. The two ingredients complement each other: the class-conditional query rule is strongest on STL-10 as labels accumulate, while the soft loss carries the method on 102-class Caltech-101, where the per-class labeled measures are thin.

WASSAL reduces the reliance on expert labels for every informative point and offers a cost-effective pathway to high accuracy when distributional structure is available. Future research will refine the weighted loss mechanism, extend the method to more real-world applications, and study how the soft loss interacts with pool size and pool shift. An interesting direction is using soft-subsetting standalone within semi-supervised learning, or attaching it to other AL query rules so that the idea reaches a broader family of methods.

Reproducibility statement

All experiments use publicly available benchmarks (CIFAR-10, SVHN, PneumoniaMNIST, STL-10, and Caltech-101) with the splits summarized in Table 1. Every reported number is the mean final test accuracy over three fixed seeds (48/86/28); on STL-10 the classical strategies at budgets 100–200 use two seeds, since the final repeat fell outside our compute window. Per-round per-class selection counts are logged for each run so that any reported number can be audited and re-plotted. The full training and selection pipeline, including the projected-gradient solver for the weighted Wasserstein formulation equation 4 and the soft-loss configuration ($\beta = 0.3$), will be released as supplementary material together with configuration files and result logs.

Ethics statement

This work studies label-efficient training on standard public image classification benchmarks and involves no human-subject research. The PneumoniaMNIST experiments use a de-identified public medical imaging dataset released for research use. Better active learning can reduce the annotation cost of machine learning systems; we do not anticipate specific harms arising from this study beyond those common to improved image classifiers. On the noisier pools we report the spread across seeds alongside the means, so that any gain can be weighed against run-to-run variation.

AI use statement

Generative AI tools were used in this work for two purposes: software engineering assistance (implementing and debugging the experimental baselines and analysis code) and assistance in drafting and editing the text of this manuscript. The research ideas, the design of the method, and the analysis of the experimental results are the authors' own. All AI-assisted code was verified by running the full experimental pipeline, and every AI-assisted passage was reviewed and edited by the authors, who take full responsibility for the final content, including the text, claims, and artifacts.

References

Ankit Agrawal, Sarsij Tripathi, and Manu Vardhan. Active learning approach using a modified least confidence sampling strategy for named entity recognition. *Progress in Artificial Intelligence*, 10 (2):113–128, 2021.

Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. In *International Conference on Learning Representations*, 2020.

Wonho Bae, Junhyug Noh, and Danica J. Sutherland. Generalized coverage for more robust low-budget active learning. In *European Conference on Computer Vision (ECCV)*, 2024.

Wonho Bae, Gabriel L. Oliveira, and Danica J. Sutherland. Uncertainty herding: One active learning method for all label budgets. In *International Conference on Learning Representations (ICLR)*, 2025.

Maria-Florina Balcan, Andrei Broder, and Tong Zhang. Margin based active learning. In *International Conference on Computational Learning Theory*, pp. 35–50. Springer, 2007.

Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 215–223. PMLR, 2011.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems*, volume 26, 2013.

John Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the l1-ball for learning in high dimensions. In *Proceedings of the 25th International Conference on Machine Learning*, pp. 272–279, 2008.

Kilian Fatras, Younes Zine, Rémi Flamary, Rémi Gribonval, and Nicolas Courty. Learning with minibatch Wasserstein: Asymptotic and gradient properties. *arXiv preprint arXiv:1910.04091*, 2019.

Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, 2004.

Guy Hacohen and Daphna Weinshall. How to select which active learning strategy is best suited for your specific problem and budget. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Guy Hacohen, Shai Dekel, and Daphna Weinshall. Active learning on a budget: Opposite strategies suit high and low budgets. In *International Conference on Machine Learning (ICML)*, 2022.

Krishnateja Killamsetty, Sivasubramanian Durga, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. GRAD-MATCH: Gradient matching based data subset selection for efficient deep model training. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 5464–5474. PMLR, 2021a.

Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh Iyer. GLISTER: Generalization based data subset selection for efficient and robust learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 8110–8118, 2021b.

Suraj Kothawade, Atharv Savarkar, Venkat Iyer, Lakshman Tamil, Ganesh Ramakrishnan, and Rishabh Iyer. CLINICAL: Targeted active learning for imbalanced medical image classification. In *Workshop on Medical Image Learning with Limited and Noisy Data*, pp. 119–129. Springer, 2022.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, Toronto, ON, Canada, 2009.

Rafid Mahmood, Sanja Fidler, and Marc T. Law. Low budget active learning via Wasserstein distance: An integer programming approach. In *International Conference on Learning Representations (ICLR)*, 2022.

Inbal Mishal and Daphna Weinshall. DCoM: Active learning for all learners. *arXiv preprint arXiv:2407.01804*, 2024.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Baolin Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B. Gupta, Xiaojiang Chen, and Xin Wang. A survey of deep active learning. *ACM Computing Surveys*, 54(9):1–40, 2021.

Ludger Rüschendorf. The Wasserstein distance and approximation theorems. *Probability Theory and Related Fields*, 70(1):117–129, 1985.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018.

Zengmao Wang, Bo Du, Weiping Tu, Lefei Zhang, and Dacheng Tao. Incorporating distribution matching into uncertainty for multiple kernel active learning. *IEEE Transactions on Knowledge and Data Engineering*, 33(1):128–142, 2021.

Tim Z. Xiao, Johannes Zenn, and Robert Bamler. The SVHN dataset is deceptive for probabilistic generative models due to a distribution mismatch. *arXiv preprint arXiv:2312.02168*, 2023.

Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. MedMNIST v2: A large-scale lightweight benchmark for 2D and 3D biomedical image classification. *Scientific Data*, 10(1):41, 2023.

Yi Yang, Zhigang Ma, Feiping Nie, Xiaojun Chang, and Alexander G. Hauptmann. Multi-class active learning by uncertainty sampling with diversity maximization. *International Journal of Computer Vision*, 113:113–127, 2015.

Ofer Yehuda, Avihu Dekel, Guy Hacohen, and Daphna Weinshall. Active learning through a covering lens. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

A Full per-budget results

Table 4 gives each method averaged over the budgets of each dataset, and Tables 5 to 7 list the per-budget means for CIFAR-10, SVHN, and PneumoniaMNIST; the Caltech-101 and STL-10 tables appear in Section 4.

Table 4: Mean final accuracy (%) of each method, averaged over the budgets of each dataset. Best of the listed methods in bold.

Strategy	CIFAR-10	SVHN	PneumoniaMNIST	STL-10	Caltech-101
WassalWithSoft	37.46	37.08	82.23	24.67	**24.31**
Wassal	38.18	39.00	84.40	**25.29**	19.32
BADGE	38.04	43.08	84.68	24.47	20.63
GLISTER	37.77	40.91	84.31	24.81	20.35
GradMatch-TSS	36.92	43.32	84.85	24.79	21.08
CoreSet	37.12	38.59	83.89	24.09	18.94
LeastConf	**38.58**	**46.42**	84.42	23.97	18.61
Margin	37.45	41.00	83.51	23.97	20.02
Random	37.84	39.32	82.88	23.93	20.90
DCoM	35.92	35.40	83.46	24.36	21.95
MaxHerding	37.61	38.62	83.33	24.26	21.16
UHerding	38.14	40.11	**85.05**	24.45	20.05
Wasserstein-IP	36.57	39.77	82.16	24.59	20.10

Table 5: CIFAR-10 mean final accuracy (%). Best of the listed methods in bold.

Strategy	25	50	100	125	150	175	200
WassalWithSoft	32.28	**35.97**	36.31	38.44	39.72	39.06	40.43
Wassal	30.62	35.38	38.24	38.55	**41.67**	41.10	41.68
BADGE	30.18	33.12	37.72	39.70	40.22	42.38	**42.99**
GLISTER	31.38	32.98	38.91	39.19	40.34	40.36	41.26
GradMatch-TSS	30.01	34.61	36.33	37.80	38.73	40.01	40.96
CoreSet	30.26	35.88	36.17	38.70	39.66	40.76	38.43
LeastConf	31.87	33.90	38.47	40.10	41.01	**42.51**	42.18
Margin	29.20	31.94	38.35	38.72	40.58	42.45	40.88
Random	30.43	33.99	**39.38**	38.75	40.32	42.24	39.75
DCoM	28.94	31.72	36.69	37.23	38.81	37.50	40.52
MaxHerding	31.01	33.56	37.28	**40.23**	40.63	39.71	40.84
UHerding	**32.79**	33.18	39.17	39.82	40.88	40.35	40.77
Wasserstein-IP	29.72	32.99	35.59	38.89	38.44	40.54	39.83

Table 6: SVHN mean final accuracy (%). Best of the listed methods in bold.

Strategy	25	50	100	175	200
WassalWithSoft	24.81	33.29	35.28	53.94	38.07
Wassal	**28.85**	35.76	39.18	37.94	53.29
BADGE	27.13	34.75	43.57	49.46	60.48
GLISTER	26.50	37.42	47.05	45.28	48.31
GradMatch-TSS	25.86	**38.22**	46.27	53.80	52.44
CoreSet	26.15	32.75	40.66	48.34	45.07
LeastConf	24.86	28.58	45.79	**64.86**	**67.99**
Margin	25.17	36.79	**48.86**	40.01	54.19
Random	23.72	34.70	42.51	48.61	47.08
DCoM	24.39	30.73	41.60	39.72	40.56
MaxHerding	24.90	33.75	39.55	43.61	51.29
UHerding	25.08	31.92	48.11	43.92	51.53
Wasserstein-IP	25.23	36.30	45.92	44.39	47.01

Table 7: PneumoniaMNIST mean final accuracy (%). Best of the listed methods in bold.

Strategy	20	30	40	50	60	70	80	90	100
WassalWithSoft	82.64	83.87	82.53	82.64	81.84	82.69	81.09	80.13	82.64
Wassal	83.28	83.44	82.64	84.67	84.83	**85.79**	85.52	84.78	84.67
BADGE	84.29	84.13	83.39	83.97	**86.38**	84.62	84.56	84.56	**86.22**
GLISTER	85.31	85.15	85.52	84.83	84.99	83.65	82.64	82.37	84.35
GradMatch-TSS	85.36	**86.70**	**87.29**	85.36	83.97	83.81	83.01	84.08	84.08
CoreSet	83.12	83.76	84.19	**85.79**	84.03	84.88	83.60	82.80	82.85
LeastConf	83.44	83.17	84.56	84.51	84.72	84.78	82.96	85.84	85.84
Margin	82.85	84.29	84.67	84.46	78.21	83.71	84.67	83.44	85.26
Random	85.63	84.03	80.82	83.01	83.39	83.44	83.87	80.45	81.30
DCoM	85.84	82.21	83.44	83.17	83.65	84.72	82.32	83.01	82.75
MaxHerding	83.65	81.84	84.51	82.69	82.85	83.28	84.62	**86.06**	80.50
UHerding	**86.86**	85.26	85.36	84.56	82.21	84.88	**86.70**	85.47	84.13
Wasserstein-IP	81.94	82.96	79.81	83.92	81.94	80.66	81.94	82.26	84.03

B Soft-loss transfer ablation

The weighted soft loss of equation 7 is not tied to the Wassal query rule, so we can attach it to any selection strategy and ask whether it helps on its own. For each classical rule we re-ran its query selection with the soft loss added, keeping everything else fixed, and report the accuracy change averaged over the budgets of each dataset. A positive entry means the soft loss improved that rule on average.

Table 8: Budget-averaged accuracy change (points) from adding the weighted soft loss on top of each classical query rule. Positive is better.

Rule	CIFAR-10	SVHN	Pneumo	STL-10	Caltech-101
BADGE	-0.43	+0.27	+0.57	-0.04	+0.46
GLISTER	+0.23	-5.49	-0.15	-3.59	+3.29
GradMatch-TSS	+0.57	-4.24	-0.35	-0.78	+1.94
CoreSet	+0.44	-2.31	+0.05	-5.42	**+4.91**
LeastConf	-0.32	**+0.50**	+0.57	-0.46	+4.44
Margin	**+0.83**	-2.95	**+1.12**	**+0.37**	+3.67