SAME BUDGET, DIFFERENT LEARNER: SCHEDULE GAP IN ACTIVE LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Active learning (AL) aims to reduce labeling effort by selecting examples for annotation. AL methods are commonly compared by accuracy at the same annotation budget. However, many small acquisition batches or a few large ones can reach the same budget with different model updates and potentially different accuracies. We study this overlooked schedule dependence by defining the schedule gap, which measures the average accuracy difference between acquisition schedules at shared cumulative budgets. Across ten methods and five natural and medical image classification datasets, the mean schedule gap ranges from 1.42% to 10.25%, and methods with low schedule sensitivity do not necessarily achieve the strongest predictive performance. Motivated by this observation, we introduce ALPS, a scheduler that distributes each acquisition batch across representativeness, coverage, and uncertainty using the estimated learner state. ALPS achieves the highest average predictive performance while maintaining a mean schedule gap of 1.78%, with its largest gains on the medical imaging datasets. Our results show that cumulative budget reflects acquired supervision, but does not fully characterize the learner state or which acquisition strategy is most useful next.[1]

1 INTRODUCTION

Active learning (AL) aims to achieve strong predictive performance with as few annotations as possible by selecting which examples to label (Settles, 2011; Li et al., 2025). AL acquisition methods are therefore commonly compared through learning curves that report accuracy against the cumulative number of acquired labels (Bae et al., 2025; Yehuda et al., 2022; Machnio et al., 2025). Comparing methods at the same cumulative budget makes annotation cost explicit, but does not capture the learning trajectory leading to that budget, *i.e.*, the sequence of acquisitions and model updates.

For example, as shown in Fig. 1(a), a cumulative budget of 400 labels can be reached through forty batches of 10 or four batches of 100. These acquisition schedules differ in how often the model is updated and how often the acquisition method can react to those updates. Consequently, the same AL method can follow different learning trajectories and achieve different accuracies at the same annotation budget depending on the acquisition schedule used to reach it.

Schedule dependence matters when deploying AL on new datasets or under new conditions. An appropriate schedule is rarely known in advance, and the one used in a benchmark may not remain suitable elsewhere. Therefore, schedule sensitivity introduces a deployment risk because a method that performs well under one schedule may use the same budget less effectively under another.

To capture this property, we formalize schedule consistency as invariance of accuracy to the acquisition schedule at shared cumulative budgets. We quantify deviations from this ideal using the *schedule gap*, defined as the mean absolute accuracy difference at cumulative budgets shared by two schedules. Evaluating schedule gap alongside accuracy reveals how consistently a method uses a given annotation budget across different acquisition schedules.

If different acquisition schedules can place the learner in different states at the same cumulative budget, then acquisition decisions based only on budget may become misaligned with the current needs of the learner. Prior work shows that the relative usefulness of acquisition mechanisms changes

[1]The code will be released upon acceptance of the paper.

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.

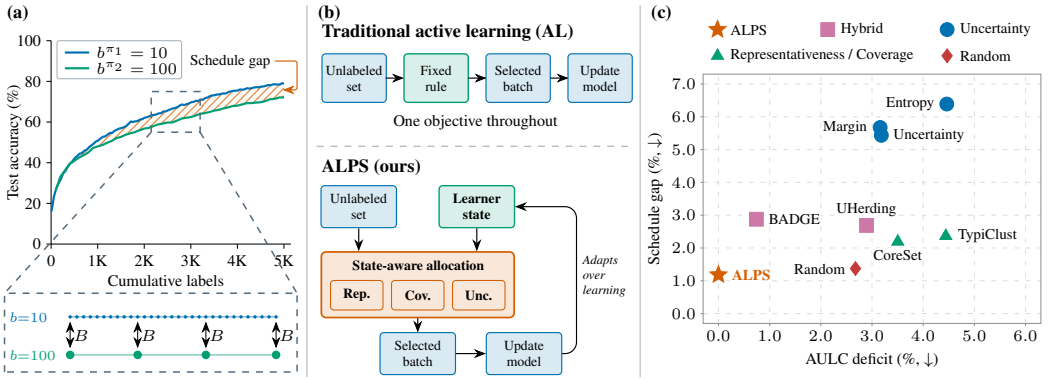

Figure 1: **The same annotation budget can result in different test accuracies.** (a) Different acquisition schedules (b^{π_1}=10 and b^{π_2}=100) can reach the same cumulative budget B with different numbers of model updates. This can lead to varying accuracies, whose differences we refer to as the schedule gap. (b) Traditional AL uses the same acquisition objective throughout learning, while ALPS estimates the learner state and adaptively allocates each batch across representativeness (Rep.), coverage (Cov.), and uncertainty (Unc.) branches. (c) Averaging over two medical datasets (ISIC2019 (Gessert et al., 2020) and BRISC2025 (Fateh et al., 2026)), ALPS lies in the favorable lower-left region, combining low predictive performance deficit with low schedule sensitivity.

throughout the learning trajectory (Machnio et al., 2026; Hacohen & Weinshall, 2023). When supervision is scarce, representative sampling can reduce distribution discrepancy; as labeled coverage expands, targeting under-represented regions can become more useful; and once a meaningful decision boundary has formed, uncertainty sampling can reduce empirical error. Thus, the relevant question becomes *which acquisition mechanism is useful for the model in its current learning state*, rather than which acquisition rule is best for a dataset at a cumulative budget.

To answer this question, we evaluate widely used AL methods under different acquisition schedules to compare how their accuracy depends on the frequency of acquisitions and model updates. We find that schedule sensitivity is often substantial and varies considerably across methods, establishing the acquisition schedule as an evaluation axis that is not captured by conventional budget-based comparisons. We also introduce ALPS (Active Learning Phase Scheduler), illustrated in Fig. 1(b), a modular state-conditioned acquisition scheduler that adapts the composition of each query batch as learning progresses. ALPS estimates the learner state from geometric coverage of the labeled set and the classifier's predictive deficit, and dynamically allocates each batch across representativeness, coverage, and uncertainty mechanisms. This adaptive allocation is designed to reduce schedule sensitivity while maintaining predictive performance. Our results indicate that *annotation budget describes how much supervision has been acquired, but does not fully describe the learner state.*

Contributions. We (i) formalize schedule consistency with respect to the AL acquisition schedule, *i.e.*, how labels are acquired across acquisition rounds, and quantify deviations from it through the schedule gap; (ii) characterize schedule sensitivity across widely used acquisition methods, showing that equal annotation budgets can correspond to different learning trajectories and accuracies; and (iii) introduce ALPS, a state-conditioned scheduler that dynamically allocates acquisition across representativeness, coverage, and uncertainty based on the current learner state.

2 Related Work

AL evaluation commonly compares acquisition methods through their accuracy at different annotation budgets, but the results can depend strongly on the acquisition schedule. Munjal et al. (2022) show that stochastic variation and regularization can substantially affect reported gains over random sampling. Lüth et al. (2023) identify broader evaluation drawbacks and propose a framework for comparing methods across datasets, acquisition settings, and training paradigms. Machnio et al. (2025) complement these studies by modeling learning trajectories through interpretable parameters describing achievable accuracy, coverage efficiency, early performance, and scalability. Our work

complements these evaluations by quantifying accuracy disagreement at cumulative shared budgets, making schedule consistency an explicit consideration alongside predictive performance.

Acquisition methods can be grouped by the mechanisms they use to select examples for labeling. Representativeness-based methods favor examples that expand the labeled support set through typicality-based clustering (Hacohen et al., 2022) or a combination of informativeness and representativeness (Huang et al., 2010). Coverage-based approaches summarize the data distribution via CoreSet selection (Sener & Savarese, 2018), probability (Yehuda et al., 2022), or generalization in representation space (Bae et al., 2024). Uncertainty-based methods rely on model predictions, using Bayesian uncertainty (Gal et al., 2017), ensemble disagreement (Beluch et al., 2018), or joint information gain within a batch (Kirsch et al., 2019). These categories are not mutually exclusive, and batch selection can additionally promote diversity among selected examples to reduce redundancy. For instance, BADGE (Ash et al., 2020) combines uncertainty with diversity in gradient-embedding space, while large-scale batch selection uses diversity to avoid redundant acquisitions (Citovsky et al., 2021). Our proposed scheduler, ALPS, builds on established acquisition mechanisms by controlling how each acquisition batch is divided among representativeness, coverage, and uncertainty.

Learning regimes help explain why the relative effectiveness of these mechanisms changes as supervision increases. Hacohen et al. (2022) show that typical examples are most useful at low budgets, while atypical become more useful as the labeled set grows. More recently, Machnio et al. (2026) connect acquisition effectiveness to data-, transition-, and model-driven regimes, and introduce measurable proxies for changes in the dominant learning bottleneck. Their post hoc analysis explains when different acquisition mechanisms become useful, but it leaves open how to translate these observations into an adaptive scheduler that remains effective across different acquisition schedules. Based on those insights, we condition ALPS's allocation on measurements of the learner state.

Adaptive acquisition changes the selection strategy during learning rather than committing to one acquisition objective. Baram et al. (2004) formulate online selection among active learners as a multi-armed bandit problem, and the method proposed by Hsu & Lin (2015) adapts strategy selection using performance estimates obtained during acquisition. SelectAL (Hacohen & Weinshall, 2023) chooses a strategy according to the problem and its selected budget. Uncertainty Herding (Bae et al., 2025) combines coverage and uncertainty in a single objective and adjusts their balance across low- and high-budget regimes. Unlike budget-dependent strategies, ALPS maintains separate acquisition branches and adaptively allocates each batch across them based on the estimated learner state.

3 Schedule-Consistent Active Learning

In this section, we formalize schedule consistency, an overlooked property of AL evaluation, as invariance of the accuracy to the acquisition schedule at shared cumulative budgets. To quantify this consistency, we introduce the schedule gap and motivate adapting acquisition to the learner state rather than to the annotation budget alone.

3.1 Schedule Consistency and Schedule Gap

Let π denote an acquisition schedule with initial labeled budget $B_0^\pi \geq 0$, consisting of T^π acquisition rounds, with a fixed batch size of b^π labels per round. The set of cumulative budgets reached by π is $\mathcal{B}(\pi) = \{B_0^\pi + tb^\pi \mid t = 1, \ldots, T^\pi\}$. For two schedules π_i and π_j, we define their shared cumulative budgets as $\widetilde{\mathcal{B}}_{ij} = \mathcal{B}(\pi_i) \cap \mathcal{B}(\pi_j)$. Let $\mathrm{Acc}(\mathsf{M}, \pi, B)$ denote the test accuracy of an AL method M under schedule π at cumulative budget B.

Definition 1 (Schedule gap)**.** For any π_i, π_j such that $\widetilde{\mathcal{B}}_{ij} \neq \emptyset$, the *schedule gap* of M is

$$\Delta_\pi(\pi_i, \pi_j) = \frac{1}{|\widetilde{\mathcal{B}}_{ij}|} \sum_{B \in \widetilde{\mathcal{B}}_{ij}} \left| \mathrm{Acc}(\mathsf{M}, \pi_i, B) - \mathrm{Acc}(\mathsf{M}, \pi_j, B) \right|. \tag{1}$$

An AL method M is then schedule consistent if $\Delta_\pi(\pi_i, \pi_j) = 0$. However, a small gap alone does not make a method desirable, since its accuracy may be consistently low. Therefore, our proposed evaluation considers the gap alongside the test accuracy and the area under the learning curve (AULC), computed over a common budget interval. Together, these metrics describe predictive performance and its sensitivity to the acquisition schedule.

3.2 From Schedule Gap to Adaptive Acquisition

As shown in Fig. 1(a), different schedules provide different opportunities to update the model, potentially leaving learners in different states at the same cumulative budget. Since the usefulness of acquisition mechanisms changes over learning (Machnio et al., 2026), an objective that is effective along one trajectory may be less suitable along another. This motivates adapting the composition of each selected batch based on the learner state. An *acquisition scheduler* determines this composition while respecting the externally specified batch size.

Definition 2 (Acquisition scheduler)**.** Let $\mathcal{M} = \{M_1, \ldots, M_K\}$ denote a set of K acquisition mechanisms and let π be an externally specified acquisition schedule. An acquisition scheduler Γ maps the available information $\mathcal{I}^t$ at round t and the requested batch size b^π to allocation weights

$$\Gamma(\mathcal{I}^t, b^\pi) = \mathbf{w}^t \in [0,1]^K, \qquad \sum_{k=1}^{K} w_k^t = 1, \tag{2}$$

where $\mathcal{I}^t$ can include learner-state measurements, and w_k^t is the fraction of the batch allocated to M_k. These weights are converted into integer branch budgets b_k^t, satisfying $\sum_{k=1}^{K} b_k^t = b^\pi$, which are used by the mechanisms to jointly select b^π distinct examples.

A scheduler is *state-conditioned* if its allocation depends on measurements of the current learner state, and *state-agnostic* if the allocation is fixed or depends only on variables such as cumulative budget or batch size. This distinction only concerns allocation across mechanisms, while the mechanisms themselves may still use additional information. Although state conditioning allows different allocations at the same budget when estimated states differ, it does not guarantee a smaller schedule gap because its value depends on whether the measurements guide useful acquisition choices.

4 ALPS: Active Learning Phase Scheduler

We propose ALPS, an active learning phase scheduler which instantiates the acquisition scheduler Γ from Definition 2 by allocating each batch across representativeness, coverage, and uncertainty. It preserves the externally specified batch size b^π and uses established acquisition methods within each branch. The controller adapts their relative shares using measurements of the current learner state, then averages allocations over estimated intermediate states within the batch. State conditioning adapts acquisition to differences in the estimated learner state, while within-batch integration approximates how the allocation may evolve during a large acquisition batch. Fig. 2 provides a detailed view of the ALPS acquisition cycle summarized in Fig. 1(b).

4.1 Estimating the Learner State

At acquisition round t, the learner has a labeled set $\mathcal{L}^t$, an unlabeled set $\mathcal{U}^t$, and a classifier h^t trained on $\mathcal{L}^t$. ALPS estimates the learner state using two normalized deficits that capture geometric coverage of the labeled set and predictive performance of the classifier.

Geometric deficit. Following Machnio et al. (2026), we define feature discrepancy (FD) as the mean nearest-neighbor cosine distance between the features from $\mathcal{L}^t$ and those from a reference subset $\mathcal{R} \subset \mathcal{V}$, where $\mathcal{V}$ is a validation set disjoint from $\mathcal{L}^t, \mathcal{U}^t$, and the test set. Thus, FD measures how well the labeled set covers the reference feature distribution. We normalize FD to obtain the geometric deficit defined as

$$r^t(\mathcal{L}^t) = \mathrm{clip}_{[0,1]}\left(\frac{\mathrm{FD}(\mathcal{L}^t)}{\mathrm{FD}(\mathcal{C})}\right), \tag{3}$$

where $\mathrm{FD}(\mathcal{C}) > 0$ denotes the discrepancy estimated from random subsets. Both $\mathcal{R}$ and $\mathrm{FD}(\mathcal{C})$ remain fixed across all acquisition rounds, and their detailed definitions are provided in App. A.1. Eq. (3) does not require class labels because FD depends only on features extracted from the images.

Predictive deficit. The geometric deficit alone does not establish that h^t is accurate. Therefore, we also measure its error on the fixed validation set:

$$\mathrm{err}^t(h^t) = \frac{1}{|\mathcal{V}|} \sum_{(x,y)\in\mathcal{V}} \mathbb{I}\left[h^t(x) \neq y\right], \tag{4}$$

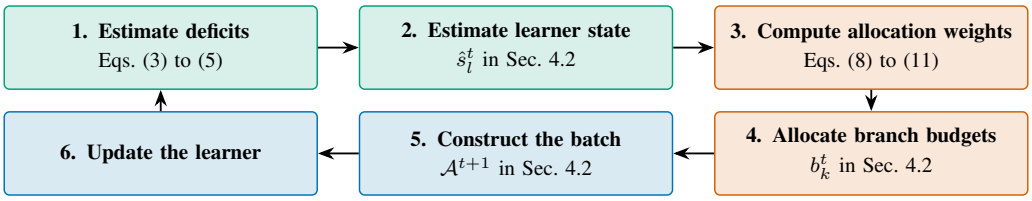

Figure 2: **ALPS acquisition cycle.** The green blocks (1–2) estimate the geometric and predictive deficits and the resulting learner state. The orange blocks (3–4) compute allocation weights and convert them into budgets for the representativeness, coverage, and uncertainty branches. The blue blocks (5–6) merge and deduplicate branch selections to construct a batch of distinct examples, request labels, update the labeled and unlabeled sets, and retrain the classifier.

where $h^t(x)$ is the predicted class. Normalizing by the expected error of uniform prediction gives

$$\rho^t(h^t) = \mathrm{clip}_{[0,1]}\left(\frac{\mathrm{err}^t(h^t)}{1 - 1/C}\right). \tag{5}$$

Joint learner state. We combine the normalized deficits using their geometric mean:

$$s^t(\mathcal{L}^t, h^t) = \sqrt{r^t(\mathcal{L}^t)\rho^t(h^t)} \in [0, 1]. \tag{6}$$

For readability, we omit the explicit arguments and write r^t, ρ^t, and s^t for $r^t(\mathcal{L}^t)$, $\rho^t(h^t)$, and $s^t(\mathcal{L}^t, h^t)$, respectively. A high s^t value requires both deficits to be large, while a low value can result from either one being small. We interpret s^t as a scalar estimate of the learner state, jointly reflecting geometric coverage and predictive performance. It is recomputed after each model update and is not expected to vary monotonically across acquisition rounds.

4.2 State-Conditioned Acquisition Allocation

Motivated by the changing usefulness of acquisition mechanisms throughout learning (Machnio et al., 2026; Hacohen et al., 2022), ALPS instantiates the scheduler in Definition 2 with $K = 3$ mechanisms. Specifically, M_1, M_2, and M_3 correspond to representativeness, coverage, and uncertainty, implemented using TypiClust (Hacohen et al., 2022), CoreSet (Sener & Savarese, 2018), and Margin (Scheffer et al., 2001), respectively.

A single batch allocation computed from the learner state at the beginning of a round treats the allocation as fixed over the entire acquisition batch. However, for large batches geometric coverage may change substantially as more examples are added. To approximate how the state dependent allocation evolves over the batch, ALPS evaluates the learner state at five representative progress points, corresponding to midpoints of five equal portions of the requested batch. These occur at approximately 10%, 30%, 50%, 70%, and 90% of the batch. For $l = 0, \ldots, 4$, let

$$u_l^t = \left\lfloor \frac{b^\pi(l + 1/2)}{5} \right\rceil \tag{7}$$

denote the number of examples that will be temporarily added at midpoint l to the labeled set to estimate the learner state, where $\lfloor \cdot \rceil$ denotes rounding to the nearest integer.

At the beginning of the round, we construct a seeded random ordering of $\mathcal{U}^t$. At midpoint l, we temporarily compute the geometric deficit using $\mathcal{L}^t$ together with the first u_l^t examples from this ordering. These examples remain unlabeled, since feature discrepancy depends only on their extracted features. The predictive deficit ρ^t stays fixed because the classifier is not retrained during this approximation. Let $\hat{s}_l^t$ denote the resulting estimated learner state at midpoint l.

For each midpoint, ALPS assigns the mechanism scores

$$\mathbf{a}_l^t = \left((\hat{s}_l^t)^2,\ \hat{s}_l^t(1 - \hat{s}_l^t),\ (1 - \hat{s}_l^t)^2\right), \tag{8}$$

which are normalized as

$$w_{l,k}^t = \frac{a_{l,k}^t}{\sum_{m=1}^3 a_{l,m}^t}, \qquad k \in \{1, 2, 3\}. \tag{9}$$

The representativeness branch receives the largest weight when $\hat{s}_l^t > 1/2$, whereas the uncertainty branch receives the largest weight when $\hat{s}_l^t < 1/2$. The coverage branch reaches its maximum weight of $1/3$ at $\hat{s}_l^t = 1/2$, where all three mechanisms are weighted equally. Thus, the mapping allows mechanisms to overlap rather than imposing three discrete acquisition phases.

We suppress small weights and sharpen the remaining midpoint allocations using

$$\tilde{w}_{l,k}^t = \frac{\max\left(w_{l,k}^t - \tau, 0\right)^\beta}{\sum_{m=1}^3 \max\left(w_{l,m}^t - \tau, 0\right)^\beta}, \qquad \tau = 0.03, \quad \beta = 1.4, \quad k \in \{1,2,3\}. \tag{10}$$

The resulting midpoint allocations $\tilde{\mathbf{w}}_l^t$ are averaged to obtain the round-level allocation

$$\mathbf{w}^t = \frac{1}{5}\sum_{l=0}^{4} \tilde{\mathbf{w}}_l^t. \tag{11}$$

This yields an allocation conditioned on the estimated evolution of the learner state across the upcoming batch, without relying on predefined transition budgets. The midpoint approximation does not predict branch selections or the classifier response after retraining.

For an ideal L-Lipschitz allocation trajectory, n-point midpoint integration approximates the interval-average allocation with error at most $Lb^\pi/(4n)$, compared with $Lb^\pi/2$ for an allocation computed only at the beginning of the round (App. B). This bound concerns allocation approximation and therefore does not guarantee a smaller schedule gap. In particular, the midpoint states are themselves estimated, and different schedules may induce different learning trajectories.

Finally, largest-remainder rounding converts $b^\pi \mathbf{w}^t$ into integer branch budgets b_k^t satisfying $\sum_{k=1}^3 b_k^t = b^\pi$. The three branches independently select candidates according to their allocated budgets, but branches with $b_k^t = 0$ are not executed. The candidate sets are merged in the fixed order and deduplicated. Any remaining positions are filled using frozen-feature typicality, followed by seeded uniform sampling if necessary, yielding $\mathcal{A}^{t+1} \subseteq \mathcal{U}^t$ with $|\mathcal{A}^{t+1}| = b^\pi$.

At cold start, when no trained classifier is available, we estimate the learner state from the geometric deficit alone, *i.e.*, $s^t = r^t$. We disable the uncertainty branch, redistribute its share proportionally between representativeness and coverage, and use frozen self-supervised features for CoreSet. After labeling $\mathcal{A}^{t+1}$, we update $\mathcal{L}^{t+1} = \mathcal{L}^t \cup \mathcal{A}^{t+1}$ and $\mathcal{U}^{t+1} = \mathcal{U}^t \setminus \mathcal{A}^{t+1}$ and retrain the classifier. The next acquisition round then recomputes the learner state from the updated model and data.

5 Experiments and ALPS Analysis

5.1 Experimental Setup

To isolate the effect of the acquisition trajectory, we keep the dataset, initial labeled set, model architecture, and training protocol fixed across schedules. We use three random seeds per method and schedule. Methods share data splits and evaluation budgets within each schedule, and comparisons across schedules use their shared cumulative budgets. Full details of the experimental setup are provided in App. C.

Datasets and models. We use CIFAR-10 and CIFAR-100 (Krizhevsky, 2009), SVHN (Netzer et al., 2011), ISIC2019 (Gessert et al., 2020), and the classification task of BRISC2025 (Fateh et al., 2026). These datasets cover general-purpose image classification and medical image classification. All experiments use ResNet-18 classifiers (He et al., 2016).

Acquisition methods. We compare ALPS with Random sampling; TypiClust (Hacohen et al., 2022) for representativeness; CoreSet (Sener & Savarese, 2018) and ProbCover (Yehuda et al., 2022) for coverage; Margin (Scheffer et al., 2001), Entropy (Settles, 2009), and Uncertainty (Lewis, 1995) for uncertainty; and the hybrid methods BADGE (Ash et al., 2020) and UHerding (Bae et al., 2025).

Metrics. We measure predictive performance using the normalized area under the learning curve (AULC) over a common budget interval and report the AULC deficit defined as the difference from the method with the highest AULC on a dataset, where lower is better. For analyses emphasizing the low-budget regime, we use normalized $\mathrm{AULC}_{\log}$ computed on a logarithmic budget axis (Guyon et al., 2011). Schedule sensitivity is measured using the schedule gap from Sec. 3.1.

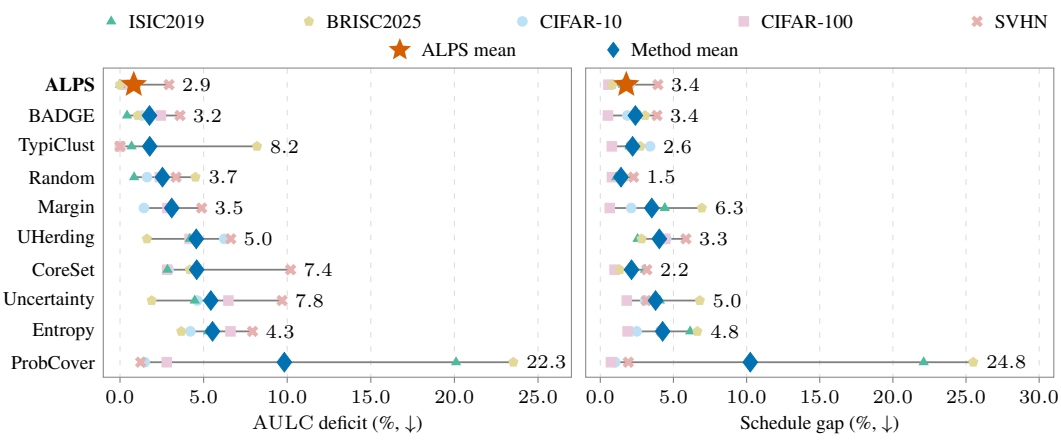

Figure 3: **Predictive performance (AULC deficit, left) and schedule consistency (right) across datasets.** Lower is better in both panels. Colored markers show individual datasets, while diamonds and the star denote cross-dataset means. Horizontal lines span the dataset minimum and maximum; shorter lines indicate less variation, and endpoint labels report the corresponding $\max - \min$ range.

5.2 How Sensitive Are Acquisition Methods to the Acquisition Schedule?

As shown in Fig. 3, predictive performance and schedule sensitivity vary across acquisition methods and datasets. The star and diamond markers denote the mean across datasets, while the horizontal line, summarized with the $\max - \min$ value, captures the cross-dataset variation.

The right panel of Fig. 3 shows that Random sampling is the most schedule consistent method, with a mean schedule gap of 1.42%, while ProbCover is the most schedule sensitive, with a mean schedule gap of 10.25%. However, as discussed in Sec. 3.1, a small gap alone does not make a method desirable. Therefore, schedule gap should be interpreted jointly with the AULC deficit when comparing methods. Notably, Random sampling surpasses six of the evaluated methods when predictive performance and schedule sensitivity are considered jointly.

Among the three methods with the lowest mean AULC deficit, ALPS achieves the smallest deficit (0.82%) while maintaining a small schedule gap (1.78%). This advantage is more pronounced on the two medical datasets shown in Fig. 1(c), where ALPS combines a low predictive performance deficit with low schedule sensitivity. These results indicate that the benefit of state-conditioned allocation is particularly relevant in challenging scenarios where AL methods are not usually benchmarked.

The cross-dataset $\max - \min$ ranges in Fig. 3 further show that the behavior of an acquisition method can vary substantially across datasets. Therefore, performance on one dataset does not necessarily indicate generalization to different data conditions. Despite this variation, ALPS maintains a good combination of predictive performance and schedule sensitivity across the evaluated datasets. This is consistent with state-conditioned allocation retaining useful behavior from its individual branches (TypiClust, CoreSet, and Margin) without inheriting their sensitivity to different datasets and schedules. App. D provides the per-dataset results and additional comparisons between natural and medical datasets.

5.3 How Does the ALPS Allocation Change Across Schedules and Datasets?

Fig. 4 shows the branch allocation weights w_k^t, introduced in Sec. 4.2, throughout the acquisition trajectory for two acquisition schedules with different b^π values for each dataset. Across datasets and schedules, ALPS shifts allocation from representativeness toward uncertainty, with coverage contributing throughout the trajectory.

Within each dataset, the two acquisition schedules follow a broadly similar allocation progression despite providing a different number of model updates. At the same time, the trajectories retain schedule-specific shifts when the estimated learner states differ. Thus, ALPS preserves a common high-level progression without reducing the allocation to a fixed function of annotation budget.

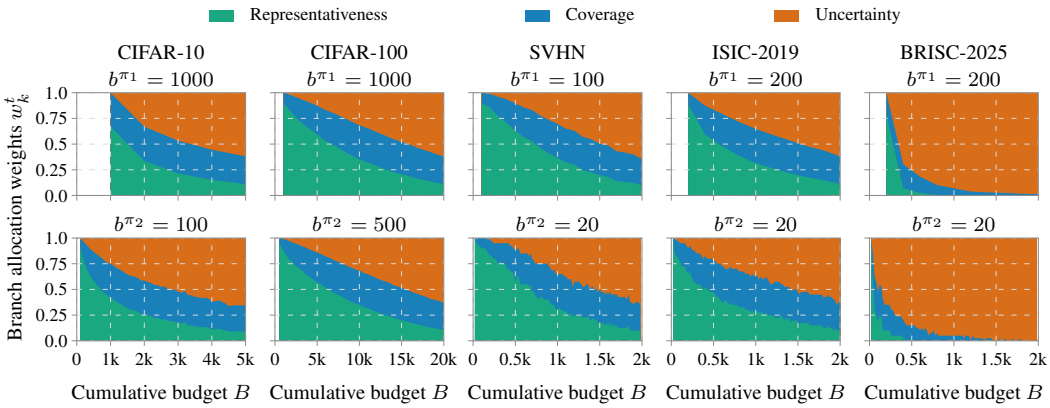

Figure 4: **State-dependent branch allocation across schedules and datasets.** Each panel shows the fraction of each acquisition batch allocated by ALPS to the representativeness, coverage, and uncertainty branches (see Sec. 4.2) as the cumulative annotation budget increases. Columns correspond to datasets, and rows to acquisition schedules.

Across datasets, the transition timing differs substantially. For instance, BRISC2025 shifts rapidly toward uncertainty, while CIFAR-100 retains representativeness and coverage for considerably longer. These differences are consistent with datasets reaching comparable learner states at different rates due to their different characteristics such as the number of classes, image modality and appearance (*e.g.*, grayscale or RGB), and image resolution. Thus, similar allocation transitions do not necessarily occur at the same annotation budgets. This behavior is exactly what state conditioning is intended to capture because ALPS does not impose fixed annotation-budget thresholds for mechanism transitions, but allows the allocation to evolve according to the estimated learner state. This is consistent with the mechanism-driven analysis of Machnio et al. (2026), which showed that acquisition transitions depend on learning progress rather than universal budget thresholds.

5.4 Which ALPS Components Contribute to its Predictive Performance?

Having established how ALPS adapts its allocation across schedules and datasets, we also investigate which components contribute to its predictive performance. Tab. 1 reports the change in $\mathrm{AULC}_{\log}$ relative to full ALPS when modifying the learner state estimation, allocation estimation, or acquisition branches.

Learner state estimation. Using only the geometric deficit r^t or the predictive deficit ρ^t has little effect on CIFAR-10/100 but consistently reduces predictive performance on the two medical datasets, supporting their complementary use. Estimating the learner state using the cumulative budget also degrades performance on the medical datasets, while estimating it using only the round progress produces the largest drop in the table, *i.e.*, -7.49% on BRISC2025. These results support conditioning acquisition on the observed learner rather than on budget or round progress.

Allocation estimation. A fixed equal allocation slightly improves predictive performance on CIFAR-10/100 but reduces $\mathrm{AULC}_{\log}$ by 1.70% and 1.03% on ISIC2019 and BRISC2025, respectively, showing that a fixed branch mixture is not uniformly preferable. Similarly, replacing the proposed mapping (Eq. 8) or the five midpoint estimates (Eq. 7) reduces performance on the medical datasets, supporting both our allocation mapping and within-batch integration.

Acquisition branches. Branch usefulness is strongly dataset dependent. Removing representativeness or coverage slightly improves performance on some CIFAR settings but substantially reduces it on the medical datasets, while removing uncertainty lowers the mean $\mathrm{AULC}_{\log}$ on all four datasets. Because no single branch is uniformly preferable across datasets, ALPS retains complementary mechanisms and adapts their relative allocation as the learner evolves.

Taken together, these ablations support the state-conditioned design of ALPS, as neither fixed allocation progressions nor individual acquisition mechanisms are consistently preferable across datasets.

Table 1: **Ablation of ALPS components.** Each variant changes one component, keeping the remaining configuration, training protocol, and seeds fixed. The first row reports ALPS $\mathrm{AULC}_{\log}$ (%); other rows show variant − ALPS (%). Variant definitions are provided in App. D.4.

Variant	CIFAR-10	CIFAR-100	ISIC2019	BRISC2025
ALPS	$66.91_{\pm 0.15}$	$33.41_{\pm 0.15}$	$59.61_{\pm 0.57}$	$92.92_{\pm 0.26}$
Learner state estimation				
State from r^t only	$-0.10_{\pm 0.98}$	$+0.16_{\pm 1.64}$	$-0.92_{\pm 1.12}$	$-1.14_{\pm 0.63}$
State from ρ^t only	$+0.26_{\pm 1.30}$	$-0.14_{\pm 1.96}$	$-0.86_{\pm 1.96}$	$-1.66_{\pm 0.96}$
State from cumulative budget	$-0.03_{\pm 1.96}$	$+0.01_{\pm 0.73}$	$-1.63_{\pm 1.22}$	$-0.82_{\pm 2.02}$
State from round progress	$-0.73_{\pm 1.32}$	$-0.29_{\pm 0.53}$	$-1.33_{\pm 0.91}$	$-7.49_{\pm 1.24}$
Allocation estimation				
Fixed equal allocation	$+0.23_{\pm 1.21}$	$+0.47_{\pm 0.76}$	$-1.70_{\pm 1.29}$	$-1.03_{\pm 0.64}$
Alternative allocation mapping	$+0.14_{\pm 1.82}$	$+0.12_{\pm 0.68}$	$-1.98_{\pm 1.63}$	$-0.29_{\pm 1.34}$
Single state estimate	$+0.13_{\pm 1.71}$	$-0.11_{\pm 0.13}$	$-1.96_{\pm 0.79}$	$-1.70_{\pm 0.79}$
Acquisition branches				
No representativeness	$+0.14_{\pm 1.69}$	$+0.59_{\pm 0.51}$	$-3.36_{\pm 2.69}$	$-2.09_{\pm 1.85}$
No coverage	$+0.32_{\pm 1.61}$	$-0.32_{\pm 0.55}$	$-1.71_{\pm 1.18}$	$-2.46_{\pm 1.31}$
No uncertainty	$-0.04_{\pm 1.41}$	$-0.10_{\pm 0.46}$	$-1.69_{\pm 0.98}$	$-1.13_{\pm 1.20}$

Table 2: **Acquisition time on CIFAR-100.** Seconds per acquisition batch at early, transition, and late stages using an acquisition batch size of $b^\pi = 1{,}000$, excluding classifier training. Values report mean ± standard deviation over three seeds. The highlighted row corresponds to full ALPS.

Method	Early ($B = 900$)	Transition ($B = 13{,}500$)	Late ($B = 22{,}000$)
TypiClust	$94.7_{\pm 6.0}$	$109.5_{\pm 9.3}$	$100.9_{\pm 10.1}$
CoreSet	$21.4_{\pm 1.9}$	$24.9_{\pm 5.1}$	$25.0_{\pm 5.3}$
Margin	$18.9_{\pm 1.7}$	$20.2_{\pm 4.3}$	$15.9_{\pm 3.4}$
ALPS	$129.4_{\pm 9.5}$	$119.7_{\pm 14.0}$	$89.0_{\pm 10.7}$
ALPS w/o TypiClust	$42.5_{\pm 3.9}$	$48.1_{\pm 9.9}$	$43.6_{\pm 8.8}$
ALPS w/o CoreSet	$114.3_{\pm 8.3}$	$109.7_{\pm 12.7}$	$70.1_{\pm 6.7}$
ALPS w/o Margin	$109.5_{\pm 8.3}$	$104.7_{\pm 11.2}$	$85.7_{\pm 9.2}$

5.5 The Cost of Adaptive Allocation

ALPS uses three acquisition branches and therefore adds computational overhead. Tab. 2 reports acquisition time excluding classifier training. At the early stage, ALPS requires 129.4 seconds per acquisition, compared with 94.7 seconds for TypiClust. Disabling the representativeness branch reduces this to 42.5 seconds, showing that TypiClust accounts for most of the acquisition overhead. As ALPS shifts allocation away from representativeness, its acquisition time decreases to 89.0 seconds at the late stage, compared with 100.9 seconds for TypiClust alone. Thus, adaptive allocation adds computation, but its cost decreases as the learner state shifts allocation toward cheaper branches.

6 Conclusion

Active learning is typically evaluated as if annotation budget fully characterized learning progress, yet the same budget can yield different accuracies under different acquisition schedules. We formalize this dependence through schedule consistency and the schedule gap, and show across ten methods and five natural and medical imaging datasets that schedule sensitivity varies substantially and does not necessarily align with predictive performance. Additionally, we introduce ALPS, a state-conditioned scheduler that allocates each batch across representativeness, coverage, and uncertainty using the estimated learner state. ALPS combines strong predictive performance with low schedule sensitivity, while its allocation adapts across datasets and schedules rather than following fixed budget-dependent transitions. Ablations further show that budget- or round-based progress estimates can substantially degrade performance, supporting learner-state conditioning. Our experimental results reinforce that annotation budget is an incomplete description of learning progress, and that acquisition should instead respond to the current learner state.

AI USE STATEMENT

We used generative AI tools to assist with implementation and code debugging, related-work searches, language editing and rephrasing of manuscript text, and converting existing PNG plots into TikZ figures. Generative AI was not used to originate the research question or proposed method, nor to generate experimental results. It was also used to help organize results produced by our experiments into the tables and figures presented in the paper and appendix. All AI-assisted code, figures, and text were reviewed and verified by the authors. We take responsibility for the complete content of this work.

ETHICS STATEMENT

This work uses publicly available datasets and does not involve human subjects or the collection of new personal data.

REPRODUCIBILITY STATEMENT

We build on an open-source active learning repository and implement ALPS within the same experimental framework. The main paper specifies the method, evaluation protocol, and reported metrics, while the appendix provides implementation details, theoretical results, validation protocol, experimental settings, and additional results. All experiments use fixed data splits and random seeds as described in the paper. The source code will be released upon acceptance.

REFERENCES

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

Wonho Bae, Junhyug Noh, and Danica J. Sutherland. Generalized coverage for more robust low-budget active learning. In *ECCV*, pp. 318–334, 2024.

Wonho Bae, Danica J. Sutherland, and Gabriel L. Oliveira. Uncertainty herding: One active learning method for all label budgets. In *ICLR*, 2025.

Yoram Baram, Ran El-Yaniv, and Kobi Luz. Online choice of active learning algorithms. *J. Mach. Learn. Res.*, 5:255–291, 2004.

William H. Beluch, Tim Genewein, Andreas Nürnberger, and Jan M. Köhler. The power of ensembles for active learning in image classification. In *CVPR*, pp. 9368–9377, 2018.

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *ICCV*, pp. 9620–9629, 2021.

Gui Citovsky, Giulia DeSalvo, Claudio Gentile, Lazaros Karydas, Anand Rajagopalan, Afshin Rostamizadeh, and Sanjiv Kumar. Batch active learning at scale. In *NeurIPS*, pp. 11933–11944, 2021.

Amirreza Fateh, Yasin Rezvani, Sara Moayedi, Sadjad Rezvani, Fatemeh Fateh, Mansoor Fateh, and Vahid Abolghasemi. BRISC: Annotated dataset for brain tumor segmentation and classification. *Scientific Data*, 13:361, 2026.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *ICML*, pp. 1183–1192, 2017.

Nils Gessert, Maximilian Nielsen, Mohsin Shaikh, René Werner, and Alexander Schlaefer. Skin lesion classification using ensembles of multi-resolution EfficientNets with meta data. *MethodsX*, 7:100864, 2020.

Isabelle Guyon, Gavin C. Cawley, Gideon Dror, and Vincent Lemaire. Results of the active learning challenge. In *Active Learning and Experimental Design Workshop in conjunction with AISTATS 2010*, volume 16 of *Proc. Mach. Learn. Res.*, pp. 19–45, 2011.

Guy Hacohen and Daphna Weinshall. How to select which active learning strategy is best suited for your specific problem and budget. In *NeurIPS*, 2023.

Guy Hacohen, Avihu Dekel, and Daphna Weinshall. Active learning on a budget: Opposite strategies suit high and low budgets. In *ICML*, pp. 8175–8195, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016.

Wei-Ning Hsu and Hsuan-Tien Lin. Active learning by learning. In *AAAI*, pp. 2659–2665, 2015.

Sheng-Jun Huang, Rong Jin, and Zhi-Hua Zhou. Active learning by querying informative and representative examples. In *NIPS*, pp. 892–900, 2010.

Andreas Kirsch, Joost van Amersfoort, and Yarin Gal. BatchBALD: Efficient and diverse batch acquisition for deep bayesian active learning. In *NeurIPS*, pp. 7024–7035, 2019.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

David D Lewis. A sequential algorithm for training text classifiers: Corrigendum and additional data. *SIGIR Forum*, 29:13–19, 1995.

Dongyuan Li, Zhen Wang, Yankai Chen, Renhe Jiang, Weiping Ding, and Manabu Okumura. A survey on deep active learning: Recent advances and new frontiers. *IEEE Trans. Neural Networks Learn. Syst.*, 36(4):5879–5899, 2025.

Carsten T. Lüth, Till J. Bungert, Lukas Klein, and Paul F. Jaeger. Navigating the pitfalls of active learning evaluation: A systematic framework for meaningful performance assessment. In *NeurIPS*, 2023.

Julia Machnio, Mads Nielsen, and Mostafa Mehdipour Ghazi. To label or not to label: PALM – A predictive model for evaluating sample efficiency in active learning models. In *ICCV*, pp. 4039–4048, 2025.

Julia Machnio, Mads Nielsen, and Mostafa Mehdipour Ghazi. A mechanism-driven theory of phase transitions in active learning. In *ECCV*, volume 17038 of *LNCS*, pp. 342–358, 2026.

Prateek Munjal, Nasir Hayat, Munawar Hayat, Jamshid Sourati, and Shadab Khan. Towards robust and reproducible active learning using neural networks. In *CVPR*, pp. 223–232, 2022.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo 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.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.*, 115(3):211–252, 2015.

Tobias Scheffer, Christian Decomain, and Stefan Wrobel. Active hidden Markov models for information extraction. In *Int. Conf. Intell. Data Anal.*, LNCS, pp. 309–318, 2001.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *ICLR*, 2018.

Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

Burr Settles. From theories to queries: Active learning in practice. In *Active learning and experimental design workshop in conjunction with AISTATS 2010*, volume 16 of *Proc. of Mach. Learn. Res.*, pp. 1–18, 2011.

Ofer Yehuda, Avihu Dekel, Guy Hacohen, and Daphna Weinshall. Active learning through a covering lens. In *NeurIPS*, 2022.

A ALPS IMPLEMENTATION DETAILS

We provide implementation details for the learner-state estimator, within-batch state approximation, and acquisition batch construction introduced in Sec. 4.

A.1 FEATURE DISCREPANCY AND NORMALIZATION

Let f denote a frozen, ℓ_2-normalized self-supervised feature extractor and let $\mathcal{R} \subset \mathcal{V}$ be the fixed reference subset used in Eq. (3), containing at most 10,000 examples. For any nonempty set of examples $\mathcal{S}$, we compute feature discrepancy as in (Machnio et al., 2026), *i.e.*,

$$\mathrm{FD}(\mathcal{S}) = \frac{1}{|\mathcal{R}|} \sum_{x \in \mathcal{R}} \min_{z \in \mathcal{S}} \left(1 - f(x)^\top f(z)\right). \tag{12}$$

Thus, $\mathrm{FD}(\mathcal{S})$ is the mean cosine distance from each reference example to its nearest example in $\mathcal{S}$. This computation does not require any labels; therefore, $\mathcal{R}$ and $\mathcal{S}$ can be unlabeled.

For a dataset with C classes, we estimate the normalization constant $\mathrm{FD}(\mathcal{C})$ once per run by averaging the feature discrepancy of five seeded random subsets $\mathcal{C} = \{\mathcal{S}_1, \ldots, \mathcal{S}_5\}$ with examples drawn from the training set, *i.e.*, $\mathcal{S}_i \cap \mathcal{R} = \emptyset$:

$$\mathrm{FD}(\mathcal{C}) = \frac{1}{5} \sum_{\mathcal{S}_i \in \mathcal{C}} \mathrm{FD}(\mathcal{S}_i), \qquad |\mathcal{S}_i| = C. \tag{13}$$

This normalization constant is fixed throughout the acquisition trajectory, and we assume $\mathrm{FD}(\mathcal{C}) > 0$.

Monotonicity of feature discrepancy. For a fixed reference set and feature extractor, feature discrepancy cannot increase when the comparison set expands,

$$\mathcal{S} \subseteq \mathcal{S}' \quad \Longrightarrow \quad \mathrm{FD}(\mathcal{S}') \leq \mathrm{FD}(\mathcal{S}). \tag{14}$$

Adding candidates to the nearest-neighbor minimization cannot increase the minimum distance for any reference example. The geometric deficit r^t (Eq. 3) inherits this property for a fixed $\mathrm{FD}(\mathcal{C})$. However, this property applies only to the geometric component; thus, the joint learner state s^t (Eq. 6) may still vary non-monotonically because the predictive deficit ρ^t (Eq. 5) can change after retraining the classifier.

Self-supervised features. For all datasets, we use a ResNet-18 feature extractor f pretrained using SimCLR. The extractor remains frozen throughout acquisition, and its features are ℓ_2-normalized before computing feature discrepancy. The same frozen representation is used by TypiClust and by CoreSet at cold start.

A.2 WITHIN-BATCH STATE APPROXIMATION

At round t, let $\tilde{\mathcal{U}}^t = \left\{o_1^t, \ldots, o_{|\mathcal{U}^t|}^t\right\}$ be the seeded random ordering of the unlabeled pool described in Sec. 4.2. For midpoint l, with temporary prefix size u_l^t (Eq. 7), we define the temporarily augmented labeled set $\mathcal{L}'^t_l = \mathcal{L}^t \cup \{o_1^t, \ldots, o_{u_l^t}^t\}$. The same ordering is used at all five midpoints, so the prefixes are nested. We assume $u_l^t > 0$ for the acquisition schedules considered in this work; therefore, every augmented labeled set $\mathcal{L}'^t_l$ is nonempty, including when $\mathcal{L}^t = \emptyset$. The temporarily added examples remain unlabeled because labels are neither required nor accessed by Eq. (12).

Using the round-level predictive deficit ρ^t, which remains fixed within the batch approximation, the midpoint learner-state estimate is

$$\hat{s}_l^t(\mathcal{L}'^t_l, h^t) = \sqrt{r^t\left(\mathcal{L}'^t_l\right) \rho_t\left(h^t\right)}. \tag{15}$$

These estimates are converted to midpoint allocations using Eqs. (8) to (10), and averaged according to Eq. (11). The midpoint approximation estimates how geometric coverage changes as the batch grows, but it does not predict the examples selected by the acquisition branches or the classifier response after retraining.

A.3 Integer Allocation and Batch Construction

Let $\mathbf{w}^t$ denote the integrated round-level allocation from Eq. (11). The branch budget is defined as

$$z_k^t = b^\pi w_k^t, \qquad k \in \{1, 2, 3\}. \tag{16}$$

Largest-remainder rounding first assigns $\lfloor z_k^t \rfloor$ examples to each branch. The remaining positions are assigned in decreasing order of the fractional remainders $z_k^t - \lfloor z_k^t \rfloor$, producing integer budgets

$$b_k^t \in \mathbb{Z}_{\geq 0}, \qquad \sum_{k=1}^{3} b_k^t = b^\pi. \tag{17}$$

Only branches with $b_k^t > 0$ are executed.

The representativeness, coverage, and uncertainty branches use TypiClust (Hacohen et al., 2022), CoreSet (Sener & Savarese, 2018), and Margin sampling (Scheffer et al., 2001), respectively. TypiClust uses the feature extractor f with self-supervised pretrained weights, which remain frozen throughout acquisition. CoreSet uses the penultimate-layer classifier features when a trained classifier is available and the frozen representation from f at cold start. Margin uses the current classifier predictions.

Cold-start allocation. When no trained classifier is available, the uncertainty branch is disabled as described in Sec. 4.2. Thus, after computing the round-level allocation, the representativeness and coverage weights are redistributed as

$$\begin{aligned} w_1^1 &= \frac{w_1^1}{w_1^1 + w_2^1}, \\ w_2^1 &= \frac{w_2^1}{w_1^1 + w_2^1}, \\ w_3^1 &= 0, \end{aligned} \tag{18}$$

where w_1^1, w_1^2, and w_1^3 correspond to the representativeness, coverage and uncertainty branches, respectively, and w_k^1 corresponds to the first acquisition round with $B_0^\pi = 0$, *i.e.*, cold start.

Candidate combination. Each executed branch returns candidates according to its allocated integer budget b_k^t. ALPS merges the candidate sets in the fixed order of representativeness, coverage, and uncertainty, removes duplicate examples, and truncates the merged set to the requested batch size if necessary. If fewer than b^π distinct examples remain, the missing positions are filled using frozen-feature typicality, followed by seeded uniform sampling from the remaining unlabeled set if needed. The resulting acquisition batch contains b^π distinct examples whenever the unlabeled set contains at least b^π candidates.

B Theoretical Properties of ALPS

We establish continuity of the state-dependent allocation mapping and bound the approximation error of midpoint integration along an ideal allocation trajectory. The results only concern allocation, not the schedule gap in Definition 1.

B.1 Lipschitz Continuity of the Allocation Mapping

Proposition 1 (Lipschitz allocation). *For $0 \leq \tau < 1/3$ and $\beta \geq 1$, the midpoint allocation weights $\tilde{\mathbf{w}}_l^t : [0, 1] \to \mathbb{R}^3$ defined in Eqs.* (8) *to* (10) *are Lipschitz continuous. In particular, there exists a finite constant L_w such that*

$$\left\| \tilde{\mathbf{w}}_l^t(s) - \tilde{\mathbf{w}}_l^t(s') \right\| \leq L_w |s - s'|, \qquad \forall s, s' \in [0, 1]. \tag{19}$$

Proof. The sum of the unnormalized scores is

$$D(s) = s^2 + s(1 - s) + (1 - s)^2 = 1 - s + s^2 \geq \frac{3}{4}. \tag{20}$$

Each normalized weight $w_{l,k}^t(s) = a_{l,k}^t(s)/D(s)$ is therefore continuously differentiable on $[0,1]$ and hence Lipschitz continuous.

For $\beta \geq 1$, the function

$$q(x) = \max(x - \tau, 0)^\beta \tag{21}$$

is Lipschitz continuous on $[0,1]$. Because the normalized weights sum to one, $\max_k w_{l,k}^t(s) \geq 1/3$, giving

$$\sum_{k=1}^{3} q(w_{l,k}^t(s)) \geq (1/3 - \tau)^\beta > 0. \tag{22}$$

Thus, the denominator of the sharpened allocation is bounded away from zero. Each component of $\tilde{\mathbf{w}}_l^t(s)$ is a ratio of Lipschitz functions with a bounded-away-from-zero denominator and is therefore Lipschitz continuous. □

In particular, if $s = \hat{s}_l^t$ and $s' = \hat{s'}_l^t$ are two estimates of the same learner state, Eq. (19) bounds the resulting allocation difference by $L_w|\hat{s}_l^t - \hat{s'}_l^t|$. This statement does not bound state-estimation error itself.

B.2 Midpoint Approximation of an Ideal Trajectory

Let $g : [B, B + b^\pi] \to \mathbb{R}^3$ be an ideal allocation trajectory. We define its interval-average allocation and n-point midpoint approximation as

$$\mathbf{w}^*(B, b^\pi) = \frac{1}{b^\pi} \int_B^{B+b^\pi} g(u)\, du, \tag{23}$$

$$\widehat{\mathbf{w}}_n(B, b^\pi) = \frac{1}{n} \sum_{j=0}^{n-1} g\left(B + \frac{b^\pi(j + 1/2)}{n}\right). \tag{24}$$

Proposition 2 (Midpoint approximation)**.** *If g is L-Lipschitz continuous with respect to the chosen norm, then*

$$\|\widehat{\mathbf{w}}_n(B, b^\pi) - \mathbf{w}^*(B, b^\pi)\| \leq \frac{Lb^\pi}{4n}. \tag{25}$$

By comparison, the beginning-of-round approximation satisfies

$$\|g(B) - \mathbf{w}^*(B, b^\pi)\| \leq \frac{Lb^\pi}{2}. \tag{26}$$

Proof. Partition $[B, B + b^\pi]$ into n subintervals of width $h = b^\pi/n$, with midpoints c_j. By the triangle inequality and the Lipschitz condition,

$$\begin{aligned} &\|\widehat{\mathbf{w}}_n(B, b^\pi) - \mathbf{w}^*(B, b^\pi)\| \\ &\leq \frac{1}{b^\pi} \sum_{j=0}^{n-1} \int_{c_j - h/2}^{c_j + h/2} \|g(c_j) - g(u)\|\, du \\ &\leq \frac{L}{b^\pi} \sum_{j=0}^{n-1} \int_{-h/2}^{h/2} |v|\, dv \\ &= \frac{Lnh^2}{4b^\pi} = \frac{Lb^\pi}{4n}. \end{aligned} \tag{27}$$

For the beginning-of-round estimate,

$$\begin{aligned} \|g(B) - \mathbf{w}^*(B, b^\pi)\| &\leq \frac{1}{b^\pi} \int_B^{B+b^\pi} \|g(B) - g(u)\|\, du \\ &\leq \frac{L}{b^\pi} \int_0^{b^\pi} v\, dv = \frac{Lb^\pi}{2}. \end{aligned} \tag{28}$$

□

For the five-point rule used by ALPS, the ideal midpoint bound is $Lb^\pi/20$, compared with $Lb^\pi/2$ for beginning-of-round allocation. These are worst-case upper bounds, not a guarantee that midpoint integration has lower error for every trajectory.

B.3 Different Partitions of a Common Trajectory

Consider two schedules that partition the same interval $[B, B+H]$ and follow a common ideal allocation trajectory g. Let partition P contain intervals of lengths $b_1, \ldots, b_R$, where $\sum_{i=1}^R b_i = H$. For each interval i, let $\widehat{\mathbf{w}}_{n,i}$ be its n-point midpoint approximation. The budget-weighted aggregate is

$$\widehat{\mathbf{G}}_P = \frac{1}{H} \sum_{i=1}^{R} b_i \widehat{\mathbf{w}}_{n,i}. \tag{29}$$

Corollary 1 (Common-trajectory partitions)**.** *If g is L-Lipschitz continuous, then*

$$\left\| \widehat{\mathbf{G}}_P - \frac{1}{H} \int_B^{B+H} g(u)\, du \right\| \leq \frac{L}{4nH} \sum_{i=1}^{R} b_i^2. \tag{30}$$

For another partition Q of the same interval, with interval lengths $c_1, \ldots, c_S$,

$$\left\| \widehat{\mathbf{G}}_P - \widehat{\mathbf{G}}_Q \right\| \leq \frac{L}{4nH} \left(\sum_{i=1}^{R} b_i^2 + \sum_{j=1}^{S} c_j^2 \right). \tag{31}$$

Proof. Applying Proposition 2 to interval i gives an approximation error of at most $Lb_i/(4n)$. Weighting this error by b_i/H and summing yields

$$\frac{1}{H} \sum_{i=1}^{R} b_i \frac{Lb_i}{4n} = \frac{L}{4nH} \sum_{i=1}^{R} b_i^2. \tag{32}$$

Both partitions approximate the same exact interval average. The bound on their difference follows by the triangle inequality. □

Scope of the analysis. The midpoint and partition results assume exact evaluations along an ideal, common allocation trajectory. In ALPS, intermediate states are approximated using random unlabeled prefixes and the predictive deficit is held fixed within a batch. Moreover, different acquisition schedules may induce different learning trajectories. Therefore, the bounds motivate the integration rule but do not establish a bound on actual cross-schedule allocation differences or guarantee a smaller schedule gap.

C Experimental Protocol and Metrics

C.1 Training and Implementation Details

Classifier training. All classifiers are trained with stochastic gradient descent (SGD) using Nesterov momentum 0.9, weight decay 3×10^{-4}, a cosine learning-rate schedule with an initial learning rate of 0.025, and a batch size of 64. All acquisition methods use the same classifier training and evaluation protocol within each dataset. We use standard image augmentations for all methods, including random cropping and horizontal flipping.

Implementation and hardware. All methods are implemented in PyTorch. The main experiments were run on NVIDIA A100, TITAN RTX, and Quadro RTX 6000 GPUs. The acquisition-time measurements in Tab. 2 were obtained on a single NVIDIA A100 GPU, using the same hardware and software configuration for all compared methods.

Table 3: **Acquisition schedules and evaluation ranges.** The top part reports the paired schedules used in Fig. 1(c) and Fig. 3, while the bottom part reports the single schedules used for the ablations in Tab. 1.

Dataset	Fine schedule (b^{π_1})	Coarse schedule (b^{π_2})	Evaluated range (step)
CIFAR-10	10	100	100–1,000 (100)
CIFAR-100	100	1,000	1,000–5,000 (1,000)
SVHN	20	100	100–1,000 (100)
ISIC2019	20	200	200–1,000 (200)
BRISC2025	20	200	200–1,000 (200)
CIFAR-10	1,000	–	1,000–10,000 (1,000)
CIFAR-100	1,000	–	1,000–20,000 (1,000)
ISIC2019	200	–	200–4,400 (200)
BRISC2025	200	–	200–4,400 (200)

C.2 Validation Set and Information Access

Before starting the acquisition process on a dataset, we hold out 10% of the original training examples using a seeded, non-stratified split. Methods and schedules under a matched seed share exactly the same validation split. Validation examples are excluded from classifier parameter fitting and the unlabeled acquisition set. All methods use the validation set under the common training protocol for checkpoint selection. Consequently, validation supervision already influences the trained classifier used by prediction-based acquisition methods through model selection. ALPS additionally reuses the validation classification error to estimate the predictive deficit ρ^t in Eq. 5 and thereby modulate its branch allocation. Thus, ALPS does not access any additional labeled examples relative to the baselines, but makes additional use of the shared validation supervision as an acquisition signal, although the same validation set is available to all methods. Moreover, validation labels are not counted toward the reported acquisition budget. The test set is used only for evaluation and is disjoint from the training and validation sets.

C.3 Acquisition Schedules and Evaluation Ranges

The results in Fig. 1(c) and Fig. 3 use two acquisition schedules per dataset. Both the AULC deficit and schedule gap are evaluated over cumulative budgets shared by the two schedules. For schedules with smaller acquisition batches, intermediate observations that are not shared by the paired schedule are excluded. The top part of Tab. 3 reports the batch sizes and evaluated budget ranges used for these comparisons.

The ablations in Tab. 1 use a single acquisition schedule per dataset because they report only $\mathrm{AULC}_{\log}$ and do not require cross-schedule comparisons. The bottom part of Tab. 3 reports the schedule and evaluated budget range used for each ablation.

C.4 Predictive Performance and Schedule Gap

Because accuracy at a single annotation budget captures only one point along the learning trajectory, we measure predictive performance using the normalized area under the learning curve. For dataset d, method M, and acquisition schedule π, let $0 < B_1 < \cdots < B_J$ denote the evaluated budgets. The linear-budget AULC is defined as:

$$\mathrm{AULC}(d, \mathsf{M}, \pi) = \frac{1}{B_J - B_1} \sum_{j=1}^{J-1} \frac{\mathrm{Acc}(\mathsf{M}, \pi, B_j) + \mathrm{Acc}(\mathsf{M}, \pi, B_{j+1})}{2} (B_{j+1} - B_j)\,. \tag{33}$$

The complementary $\mathrm{AULC}_{\log}$ uses the same observations but on a logarithmic budget axis:

$$\mathrm{AULC}_{\log}(d, \mathsf{M}, \pi) = \frac{1}{\log(B_J/B_1)} \sum_{j=1}^{J-1} \frac{\mathrm{Acc}(\mathsf{M}, \pi, B_j) + \mathrm{Acc}(\mathsf{M}, \pi, B_{j+1})}{2} \log\left(\frac{B_{j+1}}{B_j}\right). \tag{34}$$

Linear AULC gives equal weight to equal absolute budget intervals, while $\mathrm{AULC}_{\log}$ gives equal weight to equal multiplicative increases in annotation budget. We use linear AULC as the primary

Table 4: **CIFAR-10 results.** Values report mean ± standard deviation over three seeds. The values for b^{π_1}, b^{π_2}, and the ranges used for AULC, $\mathrm{AULC}_{\log}$, and Δ_π are available in Tab. 3.

Method	AULC for b^{π_1}	AULC for b^{π_2}	AULC	$\mathrm{AULC}_{\log}$	Δ_π
ALPS	$40.53_{\pm 1.33}$	$42.34_{\pm 0.29}$	$41.43_{\pm 0.52}$	$37.90_{\pm 0.40}$	$2.02_{\pm 1.10}$
Random	$40.52_{\pm 0.44}$	$41.07_{\pm 0.16}$	$40.79_{\pm 0.29}$	$36.73_{\pm 0.30}$	$1.26_{\pm 0.20}$
TypiClust	$40.65_{\pm 0.81}$	$44.17_{\pm 0.35}$	$42.41_{\pm 0.34}$	$38.64_{\pm 0.60}$	$3.41_{\pm 1.09}$
CoreSet	$39.20_{\pm 1.49}$	$39.80_{\pm 0.86}$	$39.50_{\pm 0.62}$	$35.43_{\pm 0.81}$	$2.14_{\pm 0.34}$
ProbCover	$40.73_{\pm 0.59}$	$41.13_{\pm 0.65}$	$40.93_{\pm 0.60}$	$37.02_{\pm 0.43}$	$1.02_{\pm 0.34}$
Margin	$41.26_{\pm 0.42}$	$40.70_{\pm 1.34}$	$40.98_{\pm 0.88}$	$36.58_{\pm 1.45}$	$2.11_{\pm 0.51}$
Entropy	$38.67_{\pm 0.58}$	$37.73_{\pm 0.83}$	$38.20_{\pm 0.47}$	$33.53_{\pm 0.35}$	$2.49_{\pm 0.31}$
Uncertainty	$38.58_{\pm 2.18}$	$37.01_{\pm 0.40}$	$37.79_{\pm 1.08}$	$32.86_{\pm 1.10}$	$3.06_{\pm 1.21}$
BADGE	$41.59_{\pm 1.13}$	$40.62_{\pm 0.69}$	$41.11_{\pm 0.32}$	$37.14_{\pm 0.38}$	$1.84_{\pm 0.66}$
UHerding	$34.43_{\pm 3.19}$	$37.92_{\pm 1.30}$	$36.18_{\pm 0.99}$	$32.30_{\pm 0.10}$	$4.51_{\pm 3.16}$

predictive performance metric. However, for the component ablations in Tab. 1, we additionally use $\mathrm{AULC}_{\log}$ to place greater relative emphasis on the label-scarce regime, where differences in acquisition behavior are particularly relevant.

Each experiment uses three random seeds. For the main comparisons, $\Pi = \{\pi_1, \pi_2\}$; for the ablations, $\Pi = \{\pi_1\}$. We first average the predictive performance metric across schedules and then across seeds.

$$\overline{\mathrm{AULC}}(d, \mathsf{M}) = \frac{1}{3}\sum_{i=1}^{3}\frac{1}{|\Pi|}\sum_{\pi\in\Pi}\mathrm{AULC}^i(d, \mathsf{M}, \pi), \tag{35}$$

where $\mathrm{AULC}^i(d, \mathsf{M}, \pi)$ corresponds to the AULC for seed i, dataset d, method M, and acquisition schedule π. The same computation is used for $\overline{\mathrm{AULC}}_{\log}(d, \mathsf{M})$. For readability, we omit the explicit arguments and write AULC and $\mathrm{AULC}_{\log}$ for $\overline{\mathrm{AULC}}(d, \mathsf{M})$ and $\overline{\mathrm{AULC}}_{\log}(d, \mathsf{M})$.

Let $\mathcal{H}$ be the set of evaluated methods. Then, the predictive performance deficit is defined as:

$$\text{AULC deficit}(\mathsf{M}) = \max_{\mathsf{M}'\in\mathcal{H}} \mathrm{AULC}(\mathsf{M}') - \mathrm{AULC}(\mathsf{M}). \tag{36}$$

The same computation is used for $\mathrm{AULC}_{\log}$ deficit.

For the schedule gap in Definition 1, we follow an equivalent averaging procedure, *i.e.*, we average across budgets and seeds:

$$\overline{\Delta}_\pi(\pi_1, \pi_2) = \frac{1}{3}\sum_{i=1}^{3}\Delta_\pi^i(\pi_1, \pi_2), \tag{37}$$

where $\Delta_\pi^i(\pi_1, \pi_2)$ corresponds to the schedule gap of seed i. For readability, we omit the explicit arguments and write $\overline{\Delta}_\pi$ for $\overline{\Delta}_\pi(\pi_1, \pi_2)$.

D Detailed Results and Additional Experiments

D.1 Dataset-Level Results

Tabs. 4 to 8 report the AULC for each acquisition schedule, the schedule averaged AULC and $\mathrm{AULC}_{\log}$ over the ranges in Tab. 3, and the schedule gap Δ_π.

To complement these aggregate metrics, Fig. 5 shows the underlying learning curves for ALPS and the strongest baseline on each dataset, selected by schedule averaged AULC. The curves illustrate how schedule dependent differences emerge along the acquisition trajectory, while the schedule gap summarizes these differences across shared cumulative budgets.

D.2 Natural and Medical Image Comparisons

Tab. 9 extends the comparison in Fig. 3 by reporting results averaged over all datasets as well as separately over the natural image and medical image groups. ALPS has the smallest AULC deficit

Table 5: **CIFAR-100 results.** Values report mean ± standard deviation over three seeds. The values for b^{π_1}, b^{π_2}, and the ranges used for AULC, $\text{AULC}_{\log}$, and Δ_π are available in Tab. 3.

Method	AULC for b^{π_1}	AULC for b^{π_2}	AULC	$\text{AULC}_{\log}$	Δ_π
ALPS	$26.26_{\pm 0.42}$	$26.00_{\pm 0.25}$	$26.13_{\pm 0.24}$	$24.12_{\pm 0.20}$	$0.56_{\pm 0.00}$
Random	$24.03_{\pm 0.29}$	$23.88_{\pm 0.73}$	$23.96_{\pm 0.28}$	$21.69_{\pm 0.23}$	$0.79_{\pm 0.41}$
TypiClust	$26.57_{\pm 0.25}$	$26.09_{\pm 0.46}$	$26.33_{\pm 0.34}$	$24.31_{\pm 0.28}$	$0.79_{\pm 0.29}$
CoreSet	$23.52_{\pm 0.78}$	$23.52_{\pm 0.18}$	$23.52_{\pm 0.42}$	$21.21_{\pm 0.34}$	$0.97_{\pm 0.13}$
ProbCover	$23.53_{\pm 0.44}$	$23.55_{\pm 0.40}$	$23.54_{\pm 0.40}$	$21.27_{\pm 0.43}$	$0.73_{\pm 0.16}$
Margin	$23.48_{\pm 0.56}$	$23.51_{\pm 0.63}$	$23.49_{\pm 0.59}$	$21.15_{\pm 0.59}$	$0.65_{\pm 0.27}$
Entropy	$19.38_{\pm 0.62}$	$20.07_{\pm 0.44}$	$19.73_{\pm 0.09}$	$17.28_{\pm 0.19}$	$1.87_{\pm 0.43}$
Uncertainty	$19.88_{\pm 0.40}$	$19.81_{\pm 0.77}$	$19.85_{\pm 0.18}$	$17.39_{\pm 0.02}$	$1.81_{\pm 0.08}$
BADGE	$23.99_{\pm 0.31}$	$23.80_{\pm 0.18}$	$23.90_{\pm 0.19}$	$21.59_{\pm 0.21}$	$0.52_{\pm 0.14}$
UHerding	$20.04_{\pm 0.67}$	$24.30_{\pm 0.56}$	$22.17_{\pm 0.56}$	$20.08_{\pm 0.52}$	$4.44_{\pm 0.46}$

Table 6: **SVHN results.** Values report mean ± standard deviation over three seeds. The values for b^{π_1}, b^{π_2}, and the ranges used for AULC, $\text{AULC}_{\log}$, and Δ_π are available in Tab. 3.

Method	AULC for b^{π_1}	AULC for b^{π_2}	AULC	$\text{AULC}_{\log}$	Δ_π
ALPS	$34.98_{\pm 1.69}$	$38.47_{\pm 0.39}$	$36.73_{\pm 0.98}$	$29.90_{\pm 0.48}$	$3.94_{\pm 1.47}$
Random	$36.04_{\pm 1.24}$	$36.57_{\pm 2.16}$	$36.31_{\pm 1.68}$	$29.16_{\pm 1.17}$	$2.28_{\pm 0.47}$
TypiClust	$39.02_{\pm 0.77}$	$40.30_{\pm 1.20}$	$39.66_{\pm 0.95}$	$31.98_{\pm 0.96}$	$2.11_{\pm 0.11}$
CoreSet	$28.04_{\pm 0.86}$	$30.87_{\pm 2.47}$	$29.45_{\pm 1.66}$	$23.58_{\pm 1.44}$	$3.18_{\pm 1.51}$
ProbCover	$37.99_{\pm 1.48}$	$38.85_{\pm 1.44}$	$38.42_{\pm 1.42}$	$30.82_{\pm 1.51}$	$1.92_{\pm 0.07}$
Margin	$35.19_{\pm 2.17}$	$34.36_{\pm 1.45}$	$34.77_{\pm 0.66}$	$27.69_{\pm 0.82}$	$3.49_{\pm 0.39}$
Entropy	$31.66_{\pm 1.77}$	$31.81_{\pm 1.75}$	$31.73_{\pm 0.01}$	$26.00_{\pm 0.17}$	$4.17_{\pm 0.88}$
Uncertainty	$30.53_{\pm 2.92}$	$29.39_{\pm 1.19}$	$29.96_{\pm 1.89}$	$24.12_{\pm 1.94}$	$3.14_{\pm 0.38}$
BADGE	$34.07_{\pm 2.82}$	$38.06_{\pm 1.98}$	$36.07_{\pm 2.33}$	$28.72_{\pm 1.78}$	$3.88_{\pm 1.30}$
UHerding	$30.20_{\pm 1.37}$	$35.86_{\pm 0.31}$	$33.03_{\pm 0.84}$	$26.74_{\pm 0.70}$	$5.86_{\pm 1.54}$

across all five datasets and on the two medical imaging datasets, ALPS achieves both the smallest AULC (0.00%) and the smallest schedule gap (1.18%), occupying the best position on both evaluation axes. On the other hand, TypiClust has the smallest deficit on the three natural-image datasets. Although Random has the smallest overall schedule gap, its predictive performance lags behind that of several other methods. Additionally, results of ProbCover clearly illustrate the dataset dependent schedule sensitivity because it achieves a low schedule gap on the natural image group but a substantially larger sensitivity on the medical datasets. This is the reason why ProbCover is omitted from Fig. 1(c), because its substantially larger medical dataset values would expand the plot axes making it difficult to differentiate between methods. All these results support a favorable performance-sensitivity trade-off for ALPS across these tasks, rather than dominance over every method on every dataset or metric.

D.3 Additional Evaluation on ImageNet-200

To evaluate ALPS in a larger-scale setting, we conduct an additional experiment on ImageNet-200 (Russakovsky et al., 2015). We exclude these results from Fig. 3 because the classifier setup differs from the main experiments, *i.e.*, instead of a ResNet-18 trained from scratch, we use a frozen ResNet-50 pretrained with MoCov3 (Chen et al., 2021) and train a linear classifier on its features.

ProbCover achieves the lowest schedule gap of 0.54%, but its predictive performance is the worst among the evaluated methods. Therefore, by using the joint analysis of predictive performance and schedule gap, ALPS remains the method with the best trade-off between these two metrics with a mean AULC of 66.14% and a schedule gap of 0.77%. These results demonstrate that the benefits of ALPS are not limited to the smaller datasets in the main evaluation.

D.4 ALPS Ablation Variants

Tab. 1 compares full ALPS with the following ablation variants.

Table 7: **ISIC2019 results.** Values report mean ± standard deviation over three seeds. The values for b^{π_1}, b^{π_2}, and the ranges used for AULC, $\text{AULC}_{\log}$, and Δ_π are available in Tab. 3.

Method	AULC for b^{π_1}	AULC for b^{π_2}	AULC	$\text{AULC}_{\log}$	Δ_π
ALPS	$54.00_{\pm 0.86}$	$54.53_{\pm 1.15}$	$54.27_{\pm 0.52}$	$52.75_{\pm 0.71}$	$1.53_{\pm 0.64}$
Random	$53.11_{\pm 0.84}$	$53.74_{\pm 0.38}$	$53.43_{\pm 0.50}$	$51.82_{\pm 0.74}$	$1.09_{\pm 0.57}$
TypiClust	$53.91_{\pm 1.12}$	$53.23_{\pm 0.70}$	$53.57_{\pm 0.43}$	$52.12_{\pm 0.43}$	$1.99_{\pm 0.38}$
CoreSet	$50.10_{\pm 1.75}$	$52.76_{\pm 1.66}$	$51.43_{\pm 1.57}$	$49.83_{\pm 1.65}$	$3.05_{\pm 0.87}$
ProbCover	$23.21_{\pm 2.22}$	$45.13_{\pm 1.67}$	$34.17_{\pm 1.66}$	$33.49_{\pm 1.84}$	$22.10_{\pm 1.88}$
Margin	$52.95_{\pm 0.34}$	$49.27_{\pm 1.23}$	$51.11_{\pm 0.67}$	$48.45_{\pm 0.57}$	$4.41_{\pm 0.51}$
Entropy	$51.89_{\pm 0.51}$	$46.13_{\pm 0.66}$	$49.01_{\pm 0.33}$	$46.74_{\pm 0.15}$	$6.14_{\pm 0.25}$
Uncertainty	$50.06_{\pm 1.47}$	$49.54_{\pm 0.57}$	$49.80_{\pm 0.55}$	$47.27_{\pm 0.29}$	$4.07_{\pm 1.86}$
BADGE	$52.80_{\pm 0.60}$	$54.89_{\pm 1.88}$	$53.85_{\pm 0.83}$	$52.25_{\pm 0.70}$	$2.69_{\pm 0.81}$
UHerding	$50.08_{\pm 2.95}$	$50.10_{\pm 2.79}$	$50.09_{\pm 2.80}$	$47.66_{\pm 2.99}$	$2.55_{\pm 0.83}$

Table 8: **BRISC2025 results.** Values report mean ± standard deviation over three seeds. The values for b^{π_1}, b^{π_2}, and the ranges used for AULC, $\text{AULC}_{\log}$, and Δ_π are available in Tab. 3.

Method	AULC for b^{π_1}	AULC for b^{π_2}	AULC	$\text{AULC}_{\log}$	Δ_π
ALPS	$90.47_{\pm 0.30}$	$89.93_{\pm 0.29}$	$90.20_{\pm 0.29}$	$88.16_{\pm 0.41}$	$0.83_{\pm 0.43}$
Random	$85.69_{\pm 1.09}$	$85.69_{\pm 0.56}$	$85.69_{\pm 0.32}$	$83.82_{\pm 0.43}$	$1.65_{\pm 0.35}$
TypiClust	$83.07_{\pm 0.97}$	$80.95_{\pm 0.67}$	$82.01_{\pm 0.35}$	$80.71_{\pm 0.34}$	$2.74_{\pm 1.39}$
CoreSet	$85.47_{\pm 1.52}$	$86.58_{\pm 1.04}$	$86.03_{\pm 1.27}$	$84.04_{\pm 1.11}$	$1.33_{\pm 0.31}$
ProbCover	$53.68_{\pm 0.56}$	$79.67_{\pm 1.83}$	$66.67_{\pm 0.74}$	$66.28_{\pm 0.74}$	$25.49_{\pm 2.46}$
Margin	$89.29_{\pm 1.99}$	$84.79_{\pm 3.42}$	$87.04_{\pm 2.64}$	$83.80_{\pm 2.84}$	$6.94_{\pm 2.01}$
Entropy	$89.01_{\pm 0.36}$	$84.06_{\pm 0.71}$	$86.54_{\pm 0.23}$	$83.21_{\pm 0.04}$	$6.63_{\pm 0.89}$
Uncertainty	$90.96_{\pm 0.22}$	$85.65_{\pm 1.25}$	$88.30_{\pm 0.69}$	$85.36_{\pm 0.65}$	$6.80_{\pm 1.13}$
BADGE	$90.07_{\pm 0.83}$	$88.20_{\pm 1.21}$	$89.14_{\pm 0.34}$	$87.05_{\pm 0.46}$	$3.06_{\pm 1.07}$
UHerding	$89.72_{\pm 0.78}$	$87.46_{\pm 0.48}$	$88.59_{\pm 0.46}$	$86.13_{\pm 0.64}$	$2.83_{\pm 0.70}$

State from r^t only. The learner state is estimated using only the geometric deficit, replacing $s^t = \sqrt{r^t \rho^t}$ with $s^t = r^t$. The remaining allocation procedure is unchanged.

State from ρ^t only. The learner state is estimated using only the predictive deficit, replacing $s^t = \sqrt{r^t \rho^t}$ with $s^t = \rho^t$. The remaining allocation procedure is unchanged.

State from cumulative budget. The estimated learner state is replaced by a fixed budget-dependent estimate indexed by labels per class, B/C. This estimate is obtained from the median Random-sampling learner-state trajectory across the design settings and constrained to be non-increasing. The resulting lookup table is fixed before evaluation and is queried at the five within-batch midpoints, so the allocation depends on the cumulative budget but not on the current learner.

State from round progress. The learner state is replaced by $\hat{s}^t = 1 - t/T$, where t is the number of completed acquisition rounds and T is the total number of rounds of the acquisition schedule. The same value is used at all five midpoints, making the allocation depend only on round progress rather than on budget or the observed learner.

Fixed equal allocation. The state-conditioned allocation is replaced by equal weights for representativeness, coverage, and uncertainty, $\mathbf{w}^t = (1/3, 1/3, 1/3)$, while the remaining acquisition pipeline is unchanged.

Alternative allocation mapping. The allocation mapping defined in Eq. (8) is replaced by $\mathbf{a}_l^t = \left((\hat{s}_l^t)^2,\ 2\hat{s}_l^t(1-\hat{s}_l^t),\ (1-\hat{s}_l^t)^2\right)$, which expands the range of the learner state values for which coverage receives the largest score. The remaining acquisition pipeline is unchanged.

Single state estimate. The five midpoint estimates are replaced by the learner state estimated at the beginning of the round, and this single state is used to compute the entire batch allocation. All subsequent normalization, sharpening, rounding, and batch construction steps remain unchanged.

No representativeness. The representativeness branch is disabled throughout acquisition, while the remaining ALPS components are unchanged.

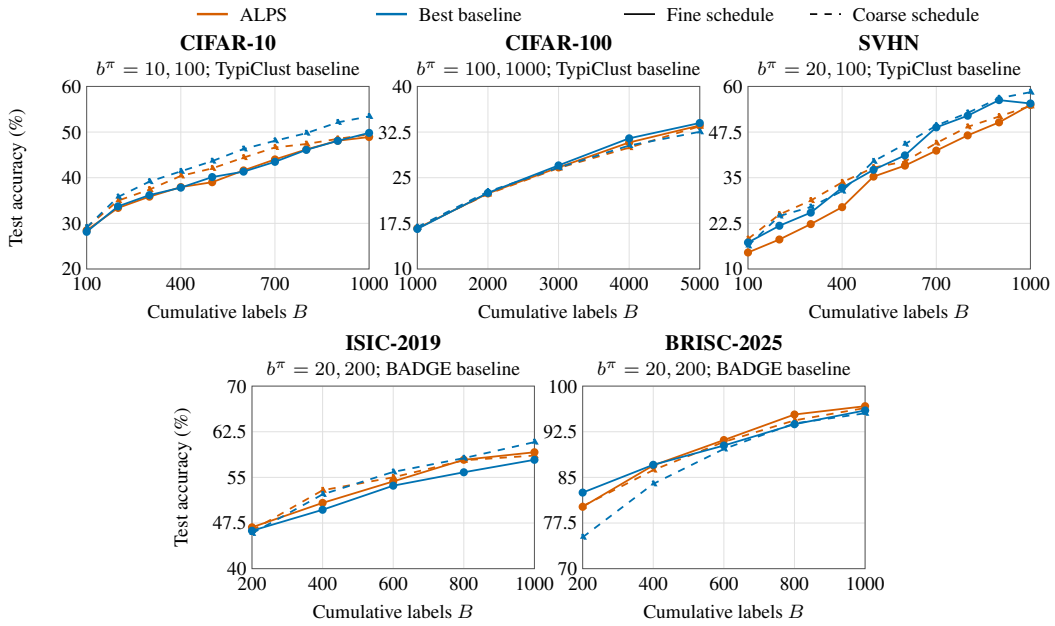

Figure 5: **Learning curves underlying the comparison in Fig. 3.** ALPS and the strongest baseline on each dataset, selected by AULC. Points show mean test accuracy over three seeds at the evaluated budgets; solid and dashed lines correspond to the fine and coarse acquisition schedules, respectively. Panels use different accuracy ranges for readability.

Table 9: **Predictive performance and schedule gap across different dataset groupings.** "All datasets" corresponds to CIFAR-10, CIFAR-100, SVHN, ISIC2019, and BRISC2025; "Natural image datasets" corresponds to CIFAR-10, CIFAR-100, and SVHN; "Medical imaging datasets" corresponds to ISIC2019 and BRISC2025.

	All datasets		Natural image datasets		Medical imaging datasets	
Method	AULC deficit	Δ_π	AULC deficit	Δ_π	AULC deficit	Δ_π
ALPS	0.82	1.78	1.37	2.17	0.00	1.18
Random	2.54	1.42	2.45	1.44	2.68	1.37
TypiClust	1.78	2.21	0.00	2.10	4.44	2.37
CoreSet	4.59	2.14	5.31	2.10	3.50	2.19
ProbCover	9.83	10.25	1.84	1.22	21.81	23.80
Margin	3.09	3.52	3.05	2.08	3.16	5.67
Entropy	5.53	4.26	6.25	2.84	4.46	6.39
Uncertainty	5.43	3.78	6.93	2.67	3.18	5.44
BADGE	1.76	2.40	2.44	2.08	0.74	2.88
UHerding	4.56	4.03	5.67	4.93	2.89	2.69

No coverage. The coverage branch is disabled throughout acquisition, while the remaining ALPS components are unchanged.

No uncertainty. The uncertainty branch is disabled throughout acquisition, while the remaining ALPS components are unchanged.

E LIMITATIONS

ALPS depends on both the learner-state estimate and the acquisition branches used by the scheduler. The scalar learner state combines geometric and predictive deficits, so a low value can result from either deficit being small even when the other remains large. The allocation mapping is also a design choice, and our experiments do not establish that the resulting weights are optimal or that represen-

Table 10: **ImageNet-200 results.** Values report mean ± standard deviation over three seeds. This experiment uses $b^{\pi_1} = 50$, $b^{\pi_2} = 200$, across the range 200–2,000(200).

Method	AULC for b^{π_1}	AULC for b^{π_2}	AULC	$\text{AULC}_{\log}$	Δ_π
ALPS	$66.28_{\pm 0.18}$	$65.99_{\pm 0.02}$	$66.14_{\pm 0.08}$	$62.31_{\pm 0.02}$	$0.77_{\pm 0.00}$
Random	$59.92_{\pm 0.89}$	$58.91_{\pm 1.26}$	$59.42_{\pm 1.07}$	$52.91_{\pm 1.27}$	$1.34_{\pm 0.25}$
TypiClust	$67.60_{\pm 0.13}$	$66.61_{\pm 0.40}$	$67.11_{\pm 0.14}$	$63.48_{\pm 0.12}$	$1.25_{\pm 0.43}$
CoreSet	$57.72_{\pm 0.83}$	$56.97_{\pm 2.16}$	$57.34_{\pm 1.49}$	$50.99_{\pm 1.45}$	$1.42_{\pm 0.30}$
ProbCover	$11.25_{\pm 0.35}$	$10.95_{\pm 1.08}$	$11.10_{\pm 0.71}$	$11.08_{\pm 0.70}$	$0.54_{\pm 0.40}$
Margin	$61.45_{\pm 0.66}$	$60.20_{\pm 0.68}$	$60.82_{\pm 0.01}$	$54.64_{\pm 0.06}$	$1.43_{\pm 0.88}$
Entropy	$55.07_{\pm 0.84}$	$45.55_{\pm 0.61}$	$50.31_{\pm 0.12}$	$42.26_{\pm 0.25}$	$9.10_{\pm 1.41}$
Uncertainty	$55.11_{\pm 0.86}$	$50.42_{\pm 2.34}$	$52.77_{\pm 0.74}$	$45.38_{\pm 0.70}$	$4.54_{\pm 3.25}$
BADGE	$51.18_{\pm 3.49}$	$59.94_{\pm 1.27}$	$55.56_{\pm 2.38}$	$50.10_{\pm 2.46}$	$8.45_{\pm 2.00}$

tativeness, coverage, and uncertainty correspond to distinct causal learning phases. Similarly, the performance of ALPS depends on the quality of its underlying acquisition methods; replacing TypiClust, CoreSet, or Margin with weaker branches may reduce the benefit of adaptive allocation. Our current formulation also assumes that these mechanisms can be represented by separate branches, whereas hybrid acquisition methods may require a different allocation scheme.

The main comparison considers two acquisition schedules per dataset, one classifier architecture, and three random seeds. They show that schedule sensitivity can vary substantially within the evaluated settings, but do not establish invariance to arbitrary acquisition schedules, variable batch sizes, noisy labels, or distribution shifts. The two medical datasets are classification benchmarks and therefore do not establish clinical utility. The theoretical analysis is similarly limited to properties of the allocation mapping and idealized within-batch integration, and does not provide guarantees on classifier accuracy or schedule gap.