STAGE-ADAPTIVE SUBMODULAR OBJECTIVE SELECTION FOR DATA-EFFICIENT TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

The most useful data-selection objective can change as a model learns, yet existing bandit-based selectors rely on historical rewards that can lag the model's current needs. We introduce **DriftBandit**, which scores candidate objectives *before updating the model* using gradient–validation alignment. By comparing candidate-subset gradients with a held-out validation gradient, it estimates objective utility at the current parameters. A complementary Page–Hinkley trigger temporarily increases exploration after validation deterioration. An arm-sampling variant reduces per-decision scoring cost from $O(K)$ to $O(m)$ by evaluating only m of K objectives. Across three vision and six medical-imaging benchmarks, each at three per-batch update budgets, full-information DriftBandit outperforms OnlineSubmod in all 27 settings. It also surpasses sample-level dynamic selectors and non-stationary bandits in the vision experiments, demonstrating gains beyond the closest objective-selection baseline. Fixed-objective comparisons establish the benefit of adapting the selection criterion during training, while ablations identify immediate gradient alignment as the principal source of improvement and deterioration-triggered exploration as a complementary gain.

1 INTRODUCTION

Data-efficient training seeks to make each update count by selecting informative examples (Sorscher et al., 2022; Coleman et al., 2020). Submodular objectives offer complementary selection criteria, including representativeness and diversity (Wei et al., 2015; Mirzasoleiman et al., 2020; Kaushal et al., 2019). Yet the most useful criterion can change as the model learns. We study how to choose that criterion under a fixed per-batch update fraction β, which limits the examples used in each parameter update (§2). OnlineSubmod (Chanda et al., 2025) casts objective selection as a bandit problem, aggregating rewards observed after training updates. Such feedback reflects earlier model states, so historical reward estimates can lag changes in objective utility. Our central insight is to *score candidate objectives at the current model state, before committing to their updates*.

We propose **DriftBandit**, which uses gradient–validation alignment to guide objective selection. Once per epoch, candidate objectives construct subsets of the same routing mini-batch. During exploitation, we rank their last-layer gradients by cosine similarity with a held-out validation gradient, then use the selected objective throughout the epoch. This provides an immediate utility proxy motivated by a first-order analysis (§3.1). A complementary Page–Hinkley trigger increases exploration for the next epoch after validation deterioration (§3.3; Figure 1). Recomputing alignment scores each epoch allows objective switching even without a trigger.

Contributions. Our contribution couples pre-update gradient scoring with deterioration-triggered exploration at the *selection-objective level*. An arm-sampling variant evaluates m of K objectives per decision, reducing scoring cost from $O(K)$ to $O(m)$ (§5.4). Across nine vision and medical-imaging benchmarks at three update fractions, full-information DriftBandit improves over OnlineSubmod in all 27 settings. On CIFAR-100 at $\beta = 0.3$, it exceeds the best fixed objective by 0.95 percentage points. Ablations identify alignment as the primary contributor and the trigger as complementary (§5.5).

Related work. We defer a comprehensive review of prior work to Appendix G, which positions **DriftBandit** with respect to *dynamic data subset selection*, *submodular optimization*, and *non-stationary multi-armed bandits*.

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.

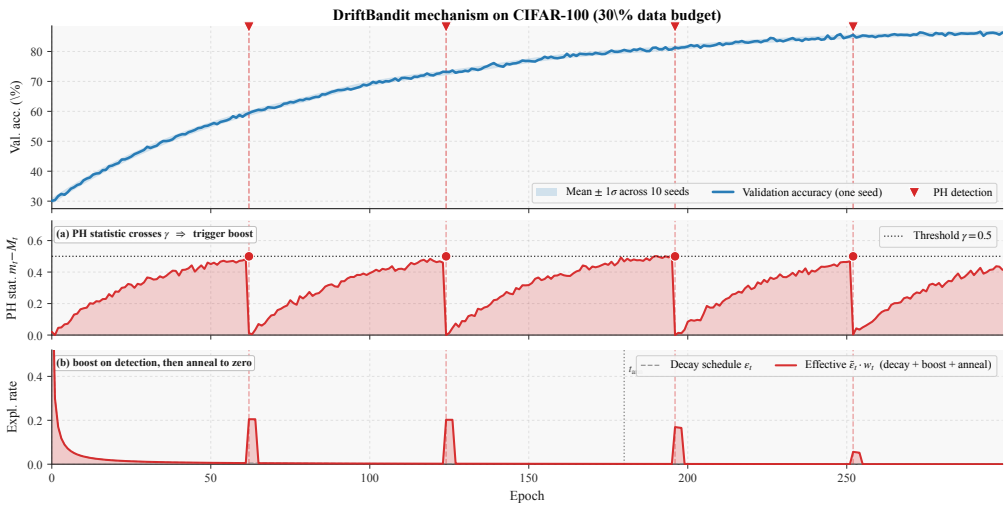


Figure 1: DriftBandit on CIFAR-100 ($\beta = 0.3$). **Top:** Validation accuracy for one seed and a mean $\pm 1\sigma$ envelope across 10 seeds; triangles mark Page–Hinkley triggers. **Middle:** The Page–Hinkley statistic and threshold γ. **Bottom:** Exploration schedule with trigger-induced boosts and late-stage annealing. Figure 4 shows objective selection over training.

2 Preliminaries and Problem Setting

2.1 Submodular Data Selection

Consider training dataset $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$ and neural network $f_{\boldsymbol{\theta}}$ with parameters $\boldsymbol{\theta}$. Training proceeds in mini-batches; at each mini-batch $\mathcal{B} \subseteq \mathcal{D}$ (of size $|\mathcal{B}|$) we select a subset $\mathcal{S} \subseteq \mathcal{B}$ with $|\mathcal{S}| \leq \beta|\mathcal{B}|$ and take the gradient step on $\mathcal{S}$ alone. Here $\beta \in (0, 1]$ is the *per-batch update fraction*—the fraction of each mini-batch that contributes to the update, not a cumulative annotation budget. Throughout, t indexes *epochs*, the granularity at which DriftBandit decides (§3).

Definition 2.1 (Submodular function)**.** A set function $f : 2^{\mathcal{B}} \to \mathbb{R}_{\geq 0}$ is submodular if for all $A \subseteq B \subseteq \mathcal{B}$ and $e \in \mathcal{B} \setminus B$:

$$f(A \cup \{e\}) - f(A) \geq f(B \cup \{e\}) - f(B). \tag{1}$$

We consider $K{=}4$ submodular functions $\mathcal{F} = \{f^{(1)}, \ldots, f^{(K)}\}$ (Facility Location: coverage; Graph Cut: diversity; Log Determinant: balance; Disparity Sum: spread; definitions in Appendix A). Given function $f^{(k)}$, the subset is selected via greedy maximization,

$$\mathcal{S}_t^{(k)} = \text{GreedyMax}(f^{(k)}, \mathcal{B}_t, \beta), \tag{2}$$

which achieves a $(1 - 1/e)$-approximation (Nemhauser et al., 1978).

2.2 Bandit Formulation for Function Selection

We formulate function selection as a K-armed bandit played *once per epoch*. At epoch t: (1) the learner selects arm $a_t \in [K]$ (a submodular function), scoring candidates on a single routing mini-batch; (2) arm a_t then governs subset construction for *every* mini-batch of that epoch, i.e. $\mathcal{S}(\mathcal{B}) = \text{GreedyMax}(f^{(a_t)}, \mathcal{B}, \beta)$ for each $\mathcal{B} \in \text{batches}(t)$; (3) the model performs one epoch of updates, $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(\mathcal{S}(\mathcal{B}); \boldsymbol{\theta})$ over those mini-batches, yielding $\boldsymbol{\theta}_{t+1}$; (4) feedback is observed and used to update the arm-selection policy. The objective is selecting arms that minimize cumulative loss on held-out validation data. One *round* here is an epoch, not a mini-batch: the arm is committed to for the whole epoch and scored once, on that epoch's first mini-batch (Algorithm 1; cadence ablated in Appendix F.16).

Information structure. We distinguish between two information regimes for our method: *(i) Full information* ($m{=}K$): during exploitation, DriftBandit computes candidate subsets and their gradients

for all K arms to score them via gradient–validation alignment (Algorithm 1, line 7). This turns function selection into a *best-arm identification* problem at each step; exploration remains useful only because the alignment signal itself is noisy (mini-batch validation gradients, first-order Taylor approximation, last-layer restriction). *(ii) Sampled information* ($m<K$): our compute-limited variant (§5.4) scores m uniformly sampled arms per decision, reducing per-decision scoring cost from $O(K)$ to $O(m)$. We are precise about what this recovers: only $m=1$ is a strict single-play partial-feedback bandit; $m=2$ observes two arm scores per round and is a *multi-play, sampled-information* scheme, not classical bandit feedback. The saving carries an identification trade-off that worsens with K/m (Theorem F.3).

Fairness of the wall-clock comparison. The OnlineSubmod baseline also computes greedy subsets for all K functions at every decision; the 3.7% overhead of DriftBandit-full is purely from cosine-similarity scoring on K last-layer gradient vectors, not extra subset construction. We additionally report a *strict single-arm OnlineSubmod* that constructs only the selected arm's subset. It is information-matched to DriftBandit $m=1$—both evaluate one candidate arm per decision—so the apples-to-apples comparison is with $m=1$, not $m=2$ (Table 4).

2.3 Limitations of existing approaches

OnlineSubmod (Chanda et al., 2025) aggregates post-update rewards, which can become stale as the model evolves. Exponential averaging further retains information from earlier training stages. DriftBandit instead recomputes gradient–validation alignment at the current parameters each epoch. A separate Page–Hinkley trigger temporarily increases exploration after validation deterioration; objective switching does not depend on this trigger.

3 Method: DriftBandit

We present DriftBandit, addressing both limitations through gradient–validation alignment for exploitation and change-point detection for adaptive exploration.

3.1 Gradient–validation alignment

The central object in DriftBandit is the *gradient alignment score*. Let $\mathcal{D}_{\text{val}}$ denote a fixed held-out validation set (10% of the original training split, never used for parameter updates). Once per epoch t, on the epoch's first (routing) mini-batch, we compute the validation gradient on a mini-batch $\mathcal{B}_t^{\text{val}} \subseteq \mathcal{D}_{\text{val}}$ of 32 samples:

$$\boldsymbol{g}_{\text{val}} = \nabla_{\boldsymbol{\theta}} \mathcal{L}(\mathcal{B}_t^{\text{val}}; \boldsymbol{\theta}_t), \tag{3}$$

using the current parameters $\boldsymbol{\theta}_t$. Let $\boldsymbol{g}_{\mathcal{S}}^{(k)} = \nabla_{\boldsymbol{\theta}} \mathcal{L}(\mathcal{S}_t^{(k)}; \boldsymbol{\theta}_t)$ denote the gradient induced by arm k's candidate subset on that same routing batch. During exploitation, DriftBandit selects:

$$a_t = \arg\max_{k \in \mathcal{A}_t} \operatorname{sim}\big(\boldsymbol{g}_{\mathcal{S}}^{(k)}, \boldsymbol{g}_{\text{val}}\big), \tag{4}$$

where $\operatorname{sim}(\cdot,\cdot)$ denotes cosine similarity and $\mathcal{A}_t \subseteq [K]$ is the set of arms scored at epoch t ($\mathcal{A}_t = [K]$ under full information, $|\mathcal{A}_t| = m$ under sampling). Proposition 3.1 gives the first-order justification, and makes explicit what the score does and does not predict.

Proposition 3.1 (First-order interpretation of the alignment score)**.** *Let $\boldsymbol{u}^{(k)}$ be the update direction the optimizer would apply if arm k were selected, so that $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta\, \boldsymbol{u}^{(k)}$. A first-order Taylor expansion of the validation loss at $\boldsymbol{\theta}_t$ gives*

$$\mathcal{L}(\boldsymbol{\theta}_{t+1}; \mathcal{D}_{val}) = \mathcal{L}(\boldsymbol{\theta}_t; \mathcal{D}_{val}) - \eta \big\langle \boldsymbol{u}^{(k)}, \boldsymbol{g}_{val} \big\rangle + O(\eta^2 \|\boldsymbol{u}^{(k)}\|^2). \tag{5}$$

Two consequences follow. (i) *Under vanilla gradient descent, $\boldsymbol{u}^{(k)} = \boldsymbol{g}_{\mathcal{S}}^{(k)}$, and maximizing the dot product $\langle \boldsymbol{g}_{\mathcal{S}}^{(k)}, \boldsymbol{g}_{val} \rangle$ maximizes the first-order predicted decrease.* (ii) *Comparing arms at a* common *update norm $\|\boldsymbol{u}^{(k)}\| = c$, the predicted decrease is $\eta c \, \|\boldsymbol{g}_{val}\| \cos(\boldsymbol{u}^{(k)}, \boldsymbol{g}_{val})$, so the cosine score ranks arms by predicted decrease* per unit update norm.

Scope of Proposition 3.1. Part (ii) is the implemented rule: cosine is the norm-normalized counterpart of the dot product, but coincides with part (i) only when arms induce comparable update norms.

Under momentum, $\boldsymbol{u}^{(k)} = \mu \boldsymbol{v}_{t-1} + \boldsymbol{g}_{\mathcal{S}}^{(k)}$; the shared term leaves the dot-product ranking unchanged yet perturbs the cosine ranking, since we normalize by $\|\boldsymbol{g}_{\mathcal{S}}^{(k)}\|$ rather than $\|\boldsymbol{u}^{(k)}\|$. Three gaps also separate Eq. (5) from the deployed method: last-layer gradients on one routing batch, a single-step expansion where the chosen arm governs a whole epoch, and a 32-sample estimate of $\boldsymbol{g}_{\text{val}}$. The proposition motivates the score; it does not guarantee the method.

3.2 Exploration schedule

DriftBandit uses ϵ-greedy exploration with a decaying schedule,

$$\epsilon_t = \left(\frac{\lambda}{t+\lambda}\right)^{\pi}, \tag{6}$$

where $\lambda > 0$ controls the decay timescale and $\pi > 0$ the curvature (t indexes epochs; $\epsilon_0{=}1$, $\epsilon_t{\to}0$).

3.3 Page–Hinkley change-point detection

To detect sustained deterioration in training progress, we apply the Page–Hinkley (PH) test (Page, 1954) to post-update validation accuracy v_t, evaluated on the full validation set $\mathcal{D}_{\text{val}}$ once per epoch for efficiency. Specifically, v_t is the accuracy of $\boldsymbol{\theta}_{t+1}$ (the parameters *after* the epoch-t update), and v_0 is the accuracy of the randomly initialized model $\boldsymbol{\theta}_0$. We detect drifts in validation accuracy rather than in per-arm rewards because (i) it is the ultimate metric of interest and (ii) it integrates information across all arms, not only the currently selected one.

Define cumulative statistics

$$m_t = m_{t-1} + (v_{t-1} - v_t) - \delta, \tag{7}$$

$$M_t = \min(M_{t-1},\, m_t), \tag{8}$$

with $m_0 = M_0 = 0$, where $\delta > 0$ is a tolerance that prevents detection of small fluctuations. Drift is detected when

$$\mathrm{PH}_t = m_t - M_t > \gamma, \tag{9}$$

where $\gamma > 0$ is the detection threshold. Upon detection, statistics reset to $m_t = M_t = 0$.

Scope of detection. PH monitors sustained decreases in v_t, not changes in the best objective. Alignment is recomputed each epoch, allowing objectives to switch without a trigger. A false trigger increases exploration for only one epoch. Set γ to 2–5 times the typical per-epoch accuracy fluctuation (0.5 on CIFAR-100; 0.7–1.0 for ISIC). Accuracy remains stable across a $20\times$ threshold range (Appendix C).

3.4 Adaptive exploration control

On a detection at epoch t, DriftBandit boosts exploration for one subsequent epoch:

$$\tilde{\epsilon}_t = \min\big(1,\, \epsilon_t + \alpha \cdot \mathbf{1}[\text{drift at } t]\big). \tag{10}$$

Under full-information scoring, a single exploitation step after the boost re-scores every arm, so one boosted epoch is enough for the selection to move; under $m{<}K$ sampling this is no longer guaranteed, and Theorem F.3 quantifies what a longer boost would buy. Late-stage annealing

$$w_t = \max\left(0,\, 1 - \frac{t - t_{\text{ann}}}{T_{\text{ann}}}\right) \quad \text{for } t \geq t_{\text{ann}}, \tag{11}$$

smoothly reduces the effective exploration $\tilde{\epsilon}_t \cdot w_t$ to zero near convergence.

4 Theoretical analysis

We state the main result grounding DriftBandit's design; the full proof, an alignment-vs-delayed-reward variance comparison, a best-arm-identification bound, and a sampled-regime identification-time result are in Appendix B.

Algorithm 1 DriftBandit (full-information variant)

Input: Submodular functions $\mathcal{F} = \{f^{(1)}, \ldots, f^{(K)}\}$
Parameters: λ, π (exploration), δ, γ (PH test), α (boost), $t_{\text{ann}}, T_{\text{ann}}$ (anneal)
Evaluate initial validation accuracy v_0 using $\boldsymbol{\theta}_0$
Initialize $m_0 \leftarrow 0,\ M_0 \leftarrow 0,\ \texttt{drift} \leftarrow \texttt{False}$
for $t = 1, 2, \ldots, T$ (epochs) **do**
  Receive the first training mini-batch $\mathcal{B}_t$ of epoch t and a validation mini-batch $\mathcal{B}_t^{\text{val}}$
  Compute candidate subsets $\{\mathcal{S}_t^{(k)}\}_{k=1}^K$ on $\mathcal{B}_t$ via Eq. (2)
  Compute last-layer gradients $\{\boldsymbol{g}_{\mathcal{S}}^{(k)}\}_{k=1}^K$ on these subsets and $\boldsymbol{g}_{\text{val}}$ on $\mathcal{B}_t^{\text{val}}$
  Set $\epsilon_t \leftarrow (\lambda/(t+\lambda))^\pi$
  if $\texttt{drift}$ **then**
    $\epsilon_t \leftarrow \min(1,\ \epsilon_t + \alpha);\quad \texttt{drift} \leftarrow \texttt{False}$
  end if
  Apply annealing $\epsilon_t \leftarrow \epsilon_t \cdot w_t$
  if $\text{Uniform}(0,1) < \epsilon_t$ **then**
    $a_t \leftarrow \text{Uniform}([K])$ ▷ Explore
  else
    $a_t \leftarrow \arg\max_k \text{sim}(\boldsymbol{g}_{\mathcal{S}}^{(k)},\ \boldsymbol{g}_{\text{val}})$ ▷ Exploit
  end if
  Train one epoch with arm a_t: for each mini-batch $\mathcal{B} \in \text{batches}(t)$, update $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}(\mathcal{S}^{(a_t)}(\mathcal{B}); \boldsymbol{\theta})$; let $\boldsymbol{\theta}_{t+1}$ be the final parameters
  Evaluate post-update validation accuracy v_t on full $\mathcal{D}_{\text{val}}$ using $\boldsymbol{\theta}_{t+1}$
  Update PH statistics via Eqs. (7)–(8)
  if $m_t - M_t > \gamma$ **then**
    $\texttt{drift} \leftarrow \texttt{True};\quad m_t \leftarrow 0,\ M_t \leftarrow 0$
  end if
end for

Assumption 4.1 (Bounded gradients)**.** There exists $G > 0$ such that $\|\nabla_{\boldsymbol{\theta}} \mathcal{L}(\mathcal{S}; \boldsymbol{\theta})\|_2 \le G$ for all $\mathcal{S}, \boldsymbol{\theta}$.

Assumption 4.2 (Gradient alignment gap)**.** At each epoch t there is an arm a_t^* whose *population* alignment score, taken with respect to the full-validation-set gradient $\bar{\boldsymbol{g}}_{\text{val}}$, exceeds every other arm's by at least $\Delta > 0$, on an event of probability at least $1 - \delta_{\text{gap}}$.

Theorem 4.3 (Detection delay)**.** *Suppose validation accuracy deteriorates persistently after epoch τ, in the sense that $\mathbb{E}[v_{t-1} - v_t \mid \mathcal{F}_{t-1}] = \Delta_v$ for all $t > \tau$ with $\Delta_v > \delta$, and let T_{detect} be the first epoch after τ at which $\text{PH}_t > \gamma$. Then*

$$\mathbb{E}[T_{\text{detect}} - \tau] \ \le\ \frac{\gamma + 1}{\Delta_v - \delta}. \tag{12}$$

The bound follows from optional stopping applied to the PH statistic (Appendix B); the additive 1 is the worst-case overshoot past γ, since $v_t \in [0, 1]$. The delay does not depend on any reward filter, whereas an EMA estimator with decay α_{ema} has effective memory $\Theta(1/\alpha_{\text{ema}})$ irrespective of Δ_v (Remark B.3).

Scope. The theorem bounds the delay in detecting a persistent *decrease in validation accuracy*—the event PH tests—and nothing more. It says nothing about when the best arm changes, and we do not assume the two coincide; nor does it bound recovery time. Combining it with Theorem F.3 would give a composite adaptation-time bound, but only under that theorem's sustained-exploration premise, which our schedule does not satisfy (Remark F.4), so we do not state one.

5 Experiments

We evaluate DriftBandit on image classification benchmarks, focusing on two questions: (1) Does DriftBandit improve accuracy over existing methods? (2) Does it exhibit more stable training dynamics?

Setup. We evaluate on three vision benchmarks (CIFAR-10/100, TinyImageNet) and six medical-imaging benchmarks (ISIC 2016–2019, Kather-2019, CRC-VAL-7K), each at three budgets, giving

Table 1: Test accuracy (%) on the natural-image benchmarks, against the full baseline set. Mean ± std over 10 seeds. These eight subset-selection baselines were run on vision only; Table 2 gives the OnlineSubmod head-to-head across all nine benchmarks. "DriftBandit" is the full-information variant ($m{=}K{=}4$); Table 4 covers the compute-reduced variants.

	CIFAR-10			CIFAR-100			TinyImageNet		
Method	10%	30%	50%	10%	30%	50%	10%	30%	50%
MILO (Killamsetty et al., 2023)	90.98 ±0.32	91.50 ±0.21	91.93 ±0.18	70.36 ±0.42	72.96 ±0.31	73.79 ±0.24	53.24 ±0.48	56.01 ±0.37	56.85 ±0.32
GradMatch (Killamsetty et al., 2021a)	91.91 ±0.27	92.54 ±0.23	92.85 ±0.19	68.94 ±0.45	71.63 ±0.34	71.26 ±0.26	53.95 ±0.53	56.19 ±0.38	58.18 ±0.31
GLISTER (Killamsetty et al., 2021b)	90.91 ±0.34	92.49 ±0.18	92.91 ±0.22	64.22 ±0.38	70.42 ±0.32	72.52 ±0.21	50.54 ±0.47	53.37 ±0.42	54.15 ±0.28
CRAIG (Mirzasoleiman et al., 2020)	89.09 ±0.36	91.14 ±0.24	92.19 ±0.19	64.27 ±0.43	69.36 ±0.28	70.12 ±0.23	51.43 ±0.51	52.92 ±0.39	56.66 ±0.33
RHO-LOSS (Mindermann et al., 2022)	90.14 ±0.29	91.47 ±0.21	93.07 ±0.17	70.37 ±0.39	73.82 ±0.27	74.74 ±0.22	53.63 ±0.46	55.78 ±0.41	56.12 ±0.29
BOSS (Acharya et al., 2024)	90.60 ±0.31	92.02 ±0.19	92.89 ±0.21	70.73 ±0.41	72.76 ±0.33	73.71 ±0.18	52.63 ±0.49	56.44 ±0.36	57.12 ±0.34
InfoBatch (Qin et al., 2024)	92.02 ±0.20	92.58 ±0.20	93.31 ±0.20	71.43 ±0.40	73.35 ±0.20	73.99 ±0.30	53.66 ±0.20	54.92 ±0.30	57.82 ±0.30
MOSO (Tan et al., 2023)	91.12 ±0.20	92.24 ±0.20	92.66 ±0.30	71.25 ±0.20	73.94 ±0.22	72.77 ±0.20	51.62 ±0.30	55.02 ±0.20	56.88 ±0.40
OnlineSubmod (Chanda et al., 2025)	91.97 ±0.18	93.12 ±0.17	93.50 ±0.16	71.44 ±0.21	73.55 ±0.22	74.63 ±0.19	54.18 ±0.25	56.44 ±0.28	58.81 ±0.25
DriftBandit	**93.39** ±0.16	**94.28** ±0.15	**94.78** ±0.14	**72.89** ±0.18	**74.81** ±0.19	**75.82** ±0.17	**55.03** ±0.22	**57.50** ±0.26	**59.72** ±0.22

Table 2: Medical-imaging benchmarks (10 seeds). Balanced accuracy for ISIC, standard accuracy for Kather/CRC. OS = OnlineSubmod; DB = DriftBandit (full-information variant, $m{=}K{=}4$). The eight additional subset-selection baselines of Table 1 were run on the natural-image benchmarks only.

	10%		**30%**		**50%**	
Benchmark	OS	**DB**	OS	**DB**	OS	**DB**
ISIC-16	79.25 ±0.10	**80.36** ±0.13	80.34 ±0.12	**81.24** ±0.14	81.48 ±0.11	**82.51** ±0.15
ISIC-17	72.01 ±0.11	**73.22** ±0.14	72.92 ±0.13	**74.10** ±0.15	73.52 ±0.12	**74.76** ±0.16
ISIC-18	72.75 ±0.10	**74.81** ±0.13	73.44 ±0.12	**75.61** ±0.14	74.88 ±0.11	**76.04** ±0.16
ISIC-19	62.16 ±0.13	**65.40** ±0.15	63.99 ±0.12	**65.62** ±0.14	64.90 ±0.14	**66.20** ±0.16
Kather-19	98.26 ±0.08	**99.66** ±0.05	98.60 ±0.10	**99.72** ±0.06	98.83 ±0.09	**99.89** ±0.07
CRC-VAL-7K	98.44 ±0.08	**99.52** ±0.11	98.25 ±0.09	**99.33** ±0.12	97.44 ±0.10	**98.63** ±0.13

nine vision and eighteen medical settings. We compare against OnlineSubmod (Chanda et al., 2025), InfoBatch (Qin et al., 2024), MOSO (Tan et al., 2023), non-stationary bandits SW-UCB and D-UCB, and CORDS-based selectors. All methods train ResNet-18 with SGD; results are mean ± std over 10 seeds on a held-out test split that is never used for selection, scoring, or drift detection.

Which variant is reported. Unless a table says otherwise, every "DriftBandit" number here—Tables 1, 2, 3 and all ablations—is the *full-information* variant scoring all $K{=}4$ arms per decision; the compute-reduced $m{<}K$ variants appear only in §5.4 and are 0.21–0.28% less accurate. Datasets, baselines, implementation details and hyperparameters are in Appendix D.

5.1 Main results

Table 1 reports the natural-image benchmarks against the full baseline set, and Table 2 the medical-imaging benchmarks against OnlineSubmod, the closest prior method and strongest competitor throughout. We report the medical results as part of the main evaluation rather than a separate application study: they are the low-budget, class-imbalanced end of the same sweep, and they carry the largest effects.

DriftBandit is the strongest method in every setting. The largest gains are on the most class-imbalanced medical benchmarks—+3.24% on ISIC-19 and +2.06% on ISIC-18 at the 10% budget—while the remaining benchmarks cluster between +0.85% and +1.45%. We do not observe a clean monotone relationship between gain and budget: it decreases with budget on CIFAR-100, ISIC-19 and Kather-19, but is flat or non-monotone on the other six benchmarks. Nor does the gain simply track headroom, since TinyImageNet has both the lowest absolute accuracy and the smallest improvement. The pattern we can state is narrower: adaptive objective selection helps most where the class distribution is most skewed, and helps by a smaller but consistent margin everywhere else. The paired delta is positive in every individual setting, confirmed by paired t-tests on CIFAR-100 and a sign test across all settings (Figure 8, Appendix F). Accuracy-vs-budget curves are in Appendix F.8; the medical fine-tuning protocol is in Appendix E.

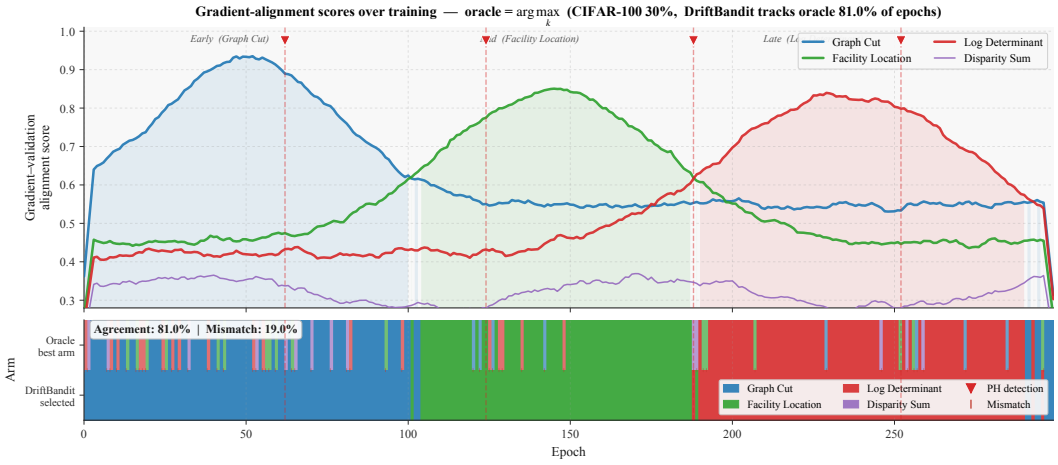


Figure 2: **Gradient alignment and oracle tracking** (CIFAR-100, 30% budget). **Top:** Per-arm cosine alignment with the validation gradient. The highest-scoring arm (oracle) shifts across shaded phases: Graph Cut → Facility Location → Log Determinant. Dashed lines mark PH triggers. **Bottom:** Oracle (upper row) versus DriftBandit (lower row), with **81.0%** agreement across epochs; red ticks mark mismatches.

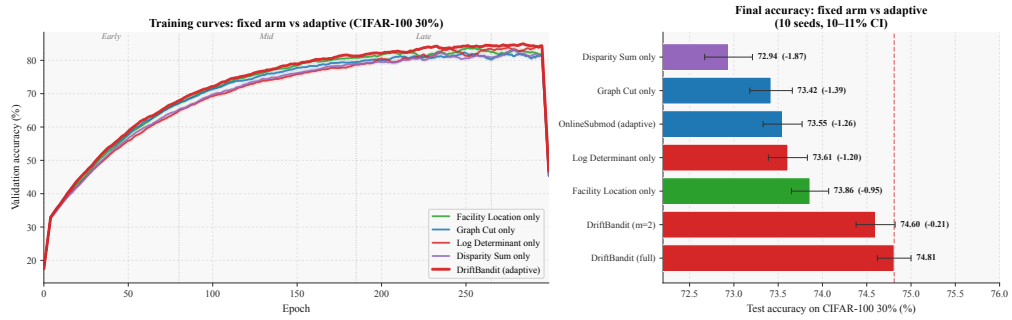


Figure 3: Fixed-arm comparison on CIFAR-100 30% (10 seeds). **Left:** no fixed function dominates all phases; DriftBandit (bold red) consistently leads. **Right:** final accuracy—adaptive selection beats the best fixed arm (Facility Location) by $+0.95\%$ ($p = 4.7 \times 10^{-9}$). Full numbers in Table 15 (Appendix F).

5.2 Oracle best arm vs selected arm & fixed-arm comparison

Oracle tracking (Figure 2). We define the oracle best arm per epoch as $a_t^{\text{oracle}} = \arg\max_k \langle \boldsymbol{g}_S^{(k)}, \bar{\boldsymbol{g}}_{\text{val}} \rangle$, computed retrospectively with the full validation gradient and all K subsets (evaluation-only). The oracle's preferred arm shifts across three phases—diversity (Graph Cut) early, representativeness (Facility Location) mid-training, and balanced (Log Determinant) late—supporting the central premise that the best objective is non-stationary. DriftBandit tracks this oracle in 81.0% of epochs. This agreement rate is a diagnostic, not evidence of optimality: the oracle uses the *same* alignment criterion DriftBandit optimises, so agreement shows only that the 32-sample estimate approximates its full-batch counterpart. We likewise make no claim that PH detections coincide with oracle phase boundaries; we have not quantified that correspondence.

Fixed-arm comparison (Figure 3). If we replace the bandit with a single submodular function held fixed throughout training (no exploration, no detection), no choice dominates: each fixed arm is preferred only during the phase where its inductive bias matches the model's current needs. Adaptive selection beats the best fixed arm by a meaningful margin and exhibits markedly lower late-stage variance because it can react to phase transitions. Full numbers are in Table 15 (Appendix F).

Table 3: Non-stationary bandit baselines on CIFAR-100 (30% budget). Mean ± std over 10 seeds. DriftBandit is the full-information variant. Note that all three baselines consume the same delayed reward signal; only the drift handling differs.

Method	Accuracy (%)	Drift handling
OnlineSubmod	73.55 ± 0.22	None (EMA)
SW-UCB (τ=50)	72.47 ± 0.23	Sliding window
D-UCB (γ_D=0.99)	74.15 ± 0.20	Discounting
DriftBandit	$\mathbf{74.81 \pm 0.19}$	Page–Hinkley

Table 4: Epoch runtime (A100) and test accuracy (10 seeds) on CIFAR-100 at 30% budget. Strict OnlineSubmod and DriftBandit m=1 each evaluate one arm per decision. DriftBandit m=2 beats both OnlineSubmod variants while running faster than full training. Timing details in Appendix E.

Method	Arms/step	Time/ep.	vs full	Acc. (%)
Full training (100%)	–	45.2 s	1.00×	76.04
OnlineSubmod (strict, 1 arm)	1	36.1 s	0.80×	73.10 ± 0.27
OnlineSubmod (full K)	4	48.3 s	1.07×	73.55 ± 0.22
DriftBandit-full (m=K=4)	4	50.1 s	1.11×	74.81 ± 0.19
DriftBandit (m=1)	1	37.4 s	0.83×	74.53 ± 0.22
DriftBandit (m=2)	2	41.5 s	0.92×	74.60 ± 0.22

5.3 Non-stationary bandit baselines

Table 3 compares DriftBandit to established non-stationary bandit baselines (SW-UCB, D-UCB) using identical arm definitions and the same delayed reward signal. Both windowing and discounting partially help over the stationary EMA estimator, but neither matches the immediate, direction-aware feedback provided by gradient alignment.

5.4 Compute-limited arm-sampling variant

Evaluating only $m<K$ arms per epoch reduces per-decision scoring cost from $O(K)$ to $O(m)$. As noted in §2, only m=1 is a strict single-play partial-feedback bandit; m=2 observes two scores per decision. Table 4 compares wall-clock time per epoch with the resulting accuracy. Selection methods are generally slower per epoch than full training because greedy submodular maximization is quadratic in the mini-batch size, so the primary value of subset selection in this regime is *data/label efficiency*, not wall-clock speed-up. The m=2 variant is faster per epoch than full-data training while still beating every OnlineSubmod variant, but it is *not* our most accurate configuration: full-information scoring is 0.21% better, at the cost of being slower than full training. Only m=1 matches the single-play partial-feedback assumption of the OnlineSubmod baseline.

5.5 Ablation studies

All ablations are run on CIFAR-100 at 30% budget; full numbers are in Appendix F.

Reward signal. Replacing immediate gradient alignment with the delayed rewards used by prior work—either delayed entropy-drop or delayed validation-loss change, with every other component fixed—removes most of the gain over OnlineSubmod (Table 14, Figure 9), confirming that immediate, direction-aware feedback is the key factor.

Component decomposition. Ablating gradient alignment (GA) and Page–Hinkley (PH) independently, GA alone already captures the bulk of the improvement; adding PH on top yields a smaller but consistent additional gain, whereas PH alone—without an accurate scorer—is actively harmful (Figure 10), showing that change-point detection only helps when paired with a reliable exploitation rule.

Strengthened baseline. To separate algorithmic from infrastructural differences, we re-tune OnlineSubmod with our exact data pipeline, optimiser, schedule, and seed budget but *without* GA or PH

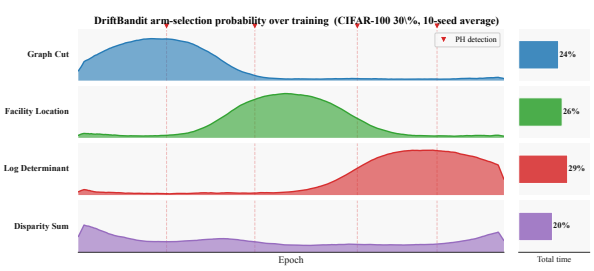


Figure 4: Arm-selection probability over training (CIFAR-100 30%, 10-seed avg). Three phases emerge; black triangles = PH detections at phase transitions.

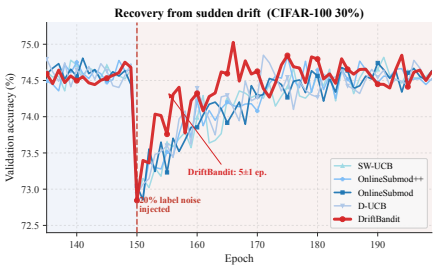


Figure 5: Recovery after 20% label noise injected at epoch 150. DriftBandit recovers in 5 ± 1 epochs vs. 10–11 for all baselines.

(OnlineSubmod++). It improves marginally over the original yet still trails DriftBandit, ruling out non-algorithmic effects as the source of the gap.

Robustness. Shrinking the validation set, and replacing $\boldsymbol{g}_{\text{val}}$ entirely with an EMA of training gradients, both still outperform OnlineSubmod, indicating that DriftBandit does not rely on a clean held-out signal.

5.6 Analysis

Training dynamics. DriftBandit exhibits smoother convergence than OnlineSubmod: the standard deviation of validation accuracy over the last 50 epochs halves ($1.8\% \rightarrow 0.9\%$) and the count of large per-epoch accuracy drops falls from 4.3 to 1.7 per run (Table 12). PH fires 4.2 ± 0.5 times per run. The arm-selection pattern (Figure 4) shows three phases—*Graph Cut* early (diversity), *Facility Location* mid-training (representativeness), and *Log Determinant* late (balanced)—so the selected function does change over training rather than collapsing to a fixed heuristic. Detections co-occur with learning-rate decay and representation maturation; we report this as an observation, not a causal claim.

Constant-LR check. Even with a constant learning rate (no cosine annealing), PH still detects 2.4 ± 0.6 drift events per run at intrinsic representation transitions, and DriftBandit retains a $+1.13\%$ advantage over OnlineSubmod, ruling out scheduler artefacts as the source (Appendix F).

Architecture transfer. On CIFAR-100 (30%), we re-run the full protocol on five backbones (Table 7). DriftBandit beats OnlineSubmod on every architecture with gains $+1.05\%$ (VGG-16) to $+1.41\%$ (WideResNet-28-10), so the mechanism is not specific to ResNet-18.

6 Conclusion

DriftBandit treats the choice of selection objective as a decision that evolves with the model. It improves over OnlineSubmod across nine vision and medical-imaging benchmarks, while the fixed-objective comparison shows the benefit of changing the criterion during training. These results support evaluating objectives through their current utility rather than relying solely on their past performance.

Limitations. Full-information scoring is slower than full-data training in our timing study, although sampling reduces this overhead. The main configuration requires held-out validation data, and our experiments are limited to image classification. The theoretical results describe individual components under restricted assumptions; they do not guarantee the performance of the complete method (Appendix F.19).

AI USE STATEMENT

Generative AI tools assisted with language polishing, readability, and technical consistency checks. The authors reviewed all AI-assisted revisions and take full responsibility for the final manuscript.

ETHICS STATEMENT

DriftBandit is a training-efficiency method: it selects *which* submodular objective drives per-batch subset selection, and does not alter the underlying data distribution or the model class being trained. Its primary anticipated effects are positive—lower compute, energy, and carbon cost per unit of accuracy (Strubell et al., 2019; Patterson et al., 2021), and lower barriers for groups training on small specialty datasets, including the clinical settings represented by our medical-imaging benchmarks, where expert annotation is the binding constraint.

Two risks deserve explicit mention. First, because the method concentrates the gradient signal on the subset judged most useful at each phase, it will not correct—and may amplify—biases already present in the training cohort (for example, demographic skew in dermoscopy datasets). We recommend pairing DriftBandit with subgroup-stratified validation and bias auditing before any deployment on demographically heterogeneous data; the per-epoch arm-selection logs we release make it possible to audit how much of the training signal each submodular function contributed. Second, efficiency gains can induce compute-recovery effects, in which a lower marginal cost per experiment leads to more total training rather than less; we encourage practitioners to report cumulative rather than per-run compute.

All datasets used in this work (CIFAR-10/100, TinyImageNet, ISIC 2016–2019, Kather-2019, CRC-VAL-7K) are publicly released benchmarks used under their original licences and intended research use; no new human-subject data was collected. An extended discussion is provided in Appendix F.19.

REPRODUCIBILITY STATEMENT

We have aimed to make every reported number reproducible from the description in this paper. The complete method is specified in §3 and Algorithm 1, with all hyperparameters $(\lambda, \pi, \delta, \gamma, \alpha, t_{\text{ann}}, T_{\text{ann}}, m)$ listed in Appendix D and their sensitivity characterised in Appendix C. Datasets, splits, preprocessing, augmentation, optimiser settings, and the medical-imaging fine-tuning protocol are given in Appendices D and E; baseline configurations, including the grid searches used to set the SW-UCB and D-UCB hyperparameters and the strict single-arm OnlineSubmod re-implementation (Appendix F.17), are stated alongside them. All headline results are means $\pm$ standard deviation over 10 random seeds, with paired seed-by-seed tests against OnlineSubmod. Assumptions 4.1 and 4.2 are stated explicitly, and complete proofs of Theorem 4.3, Theorem F.1, Theorem F.3, and Propositions 3.1 and B.1 appear in Appendix B and Appendices F.20 and F.21. Source code and the per-epoch arm-selection logs used to produce Figure 2 will be released; an anonymised copy is provided as supplementary material.

REFERENCES

Abhinab Acharya, Dayou Yu, Qi Yu, and Xumin Liu. Balancing feature similarity and label variability for optimal size-aware one-shot subset selection. In *International Conference on Machine Learning*. PMLR, 2024.

Stephanie Allen, David Madras, Ye Ye, and Greg Zanotti. Change-point detection methods for body-worn video, 2016.

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2):235–256, 2002.

Francis Bach. Learning with submodular functions: A convex optimization perspective. *Foundations and Trends in Machine Learning*, 6(2-3):145–373, 2013.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *International Conference on Machine Learning*, pp. 41–48. ACM, 2009.

Omar Besbes, Yonatan Gur, and Assaf Zeevi. Stochastic multi-armed-bandit problem with non-stationary rewards. *Advances in Neural Information Processing Systems*, 27:199–207, 2014.

Yang Cao, Zheng Wen, Branislav Kveton, and Yao Xie. Nearly optimal adaptive procedure with change detection for piecewise-stationary bandit. In *International Conference on Artificial Intelligence and Statistics*, pp. 418–427. PMLR, 2019.

Prateek Chanda, Prayas Agrawal, Saral Sureka, Lokesh Reddy Polu, Atharv Kshirsagar, and Ganesh Ramakrishnan. Bandit guided submodular curriculum for adaptive subset selection. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Noel CF Codella, David Gutman, M Emre Celebi, Brian Helba, Michael A Marchetti, Stephen W Dusza, Aadi Kalloo, Konstantinos Liopyris, Nabin Mishra, Harald Kittler, et al. Skin lesion analysis toward melanoma detection: A challenge at the 2017 International Symposium on Biomedical Imaging (ISBI), hosted by the International Skin Imaging Collaboration (ISIC). *arXiv preprint arXiv:1710.05006*, 2018.

Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. Selection via proxy: Efficient data selection for deep learning. In *International Conference on Learning Representations*, 2020.

Marc Combalia, Noel C. F. Codella, Veronica Rotemberg, Brian Helba, Veronica Vilaplana, Ofer Reiter, Cristina Carrera, Alicia Barreiro, Allan C. Halpern, Susana Puig, and Josep Malvehy. Bcn20000: Dermoscopic lesions in the wild, 2019.

Abhimanyu Das and David Kempe. Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. In *International Conference on Machine Learning*, pp. 1057–1064. PMLR, 2011.

Satoru Fujishige. *Submodular functions and optimization*. Elsevier, 2005.

Aurélien Garivier and Eric Moulines. On upper-confidence bound policies for switching bandit problems. In *International Conference on Algorithmic Learning Theory*, pp. 174–188. Springer, 2011.

Guy Hacohen and Daphna Weinshall. On the power of curriculum learning in training deep networks. In *International Conference on Machine Learning*, pp. 2535–2544. PMLR, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Tyler B Johnson and Carlos Guestrin. Training deep models faster with robust, approximate importance sampling. In *Advances in Neural Information Processing Systems*, volume 31, pp. 7265–7275, 2018.

Angelos Katharopoulos and François Fleuret. Not all samples are created equal: Deep learning with importance sampling. In *International Conference on Machine Learning*, pp. 2525–2534. PMLR, 2018.

Jakob Nikolas Kather, Johannes Krisam, Pornpimol Charoentong, Tom Luedde, Esther Herpel, Cleo-Aron Weis, Timo Gaiser, Alexander Marx, Nektarios A Valous, Dyke Ferber, et al. Predicting survival from colorectal cancer histology slides using deep learning: A retrospective multicenter study. *PLoS Medicine*, 16(1):e1002730, 2019.

Vishal Kaushal, Rishabh Iyer, Suraj Kothawade, Rohan Mahadev, Khoshrav Doctor, and Ganesh Ramakrishnan. Learning from less data: A unified data subset selection and active learning framework for computer vision, 2019.

Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. Grad-match: Gradient matching based data subset selection for efficient deep model training, 2021a.

Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh Iyer. Glister: Generalization based data subset selection for efficient and robust learning, 2021b.

Krishnateja Killamsetty, Xujiang Zhao, Feng Chen, and Rishabh Iyer. RETRIEVE: Coreset selection for efficient and robust semi-supervised learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 14488–14501, 2021c.

Krishnateja Killamsetty, Durga Sivasubramanian, Baharan Mirzasoleiman, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. Cords: Coresets and data subset selection. `https://github.com/decile-team/cords`, 2022. GitHub repository.

Krishnateja Killamsetty, Alexandre V. Evfimievski, Tejaswini Pedapati, Kiran Kate, Lucian Popa, and Rishabh Iyer. Milo: Model-agnostic subset selection framework for efficient model training and tuning, 2023.

Levente Kocsis and Csaba Szepesvári. Discounted UCB. In *2nd PASCAL Challenges Workshop*, volume 2, pp. 784–791, 2006.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, pp. 1885–1894. PMLR, 2017.

Suraj Kothawade, Vishal Kaushal, Ganesh Ramakrishnan, Jeff Bilmes, and Rishabh Iyer. Prism: A rich class of parameterized submodular information measures for guided data subset selection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(9):10238–10246, June 2022. doi: 10.1609/aaai.v36i9.21264.

M Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, volume 23, pp. 1189–1197, 2010.

Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. Cambridge University Press, 2020.

Lihong Li, Wei Chu, John Langford, and Robert E Schapire. A contextual-bandit approach to personalized news article recommendation. In *International Conference on World Wide Web*, pp. 661–670. ACM, 2010.

Hui Lin and Jeff Bilmes. A class of submodular functions for document summarization. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 510–520, 2011.

Jonathan Lorraine, Paul Vicol, and David Duvenaud. Optimizing millions of hyperparameters by implicit differentiation. In *International Conference on Artificial Intelligence and Statistics*, pp. 1540–1552. PMLR, 2020.

Adyasha Maharana, Prateek Yadav, and Mohit Bansal. D2 pruning: Message passing for balancing diversity and difficulty in data pruning. *arXiv preprint arXiv:2310.07931*, 2023.

Sören Mindermann, Jan Brauner, Muhammed Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Höltgen, Aidan N. Gomez, Adrien Morisot, Sebastian Farquhar, and Yarin Gal. Prioritized training on points that are learnable, worth learning, and not yet learnt, 2022.

Baharan Mirzasoleiman, Ashwinkumar Badanidiyuru, and Amin Karbasi. Fast constrained submodular maximization: Personalized data summarization. In *International Conference on Machine Learning*, pp. 1358–1367. PMLR, 2016.

Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *International Conference on Machine Learning*, pp. 6950–6960. PMLR, 2020.

George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1):265–294, 1978.

Ewan S Page. Continuous inspection schemes. *Biometrika*, 41(1/2):100–115, 1954.

David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350*, 2021.

Omead Pooladzandi, David Davini, and Baharan Mirzasoleiman. Adaptive second order coresets for data-efficient machine learning. In *International Conference on Machine Learning*, pp. 17848–17869. PMLR, 2022.

Ziheng Qin, Kai Wang, Zangwei Zheng, Jianyang Gu, Xiangyu Peng, Zhaopan Xu, Daquan Zhou, Lei Shang, Baigui Sun, Xuansong Xie, and Yang You. Infobatch: Lossless training speed up by unbiased dynamic data pruning. In *International Conference on Learning Representations (ICLR)*, 2024.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *International Conference on Machine Learning*, pp. 4334–4343. PMLR, 2018.

Burr Settles. Active learning literature survey. *Computer Sciences Technical Report 1648*, 2009.

Jun Shu, Qi Xie, Lixuan Yi, Qian Zhao, Sanping Zhou, Zongben Xu, and Deyu Meng. Meta-weight-net: Learning an explicit mapping for sample weighting. In *Advances in Neural Information Processing Systems*, volume 32, pp. 1919–1930, 2019.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. *Advances in Neural Information Processing Systems*, 35:19523–19536, 2022.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3645–3650, 2019.

Haoru Tan, Sitong Wu, Fei Du, Yukang Chen, Zhibin Wang, Fan Wang, and Xiaojuan Qi. Data pruning via moving-one-sample-out. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. HAM10000: A large dataset of multi-source dermatoscopic images of common pigmented skin lesions. In *Scientific Data*, volume 5, pp. 180161, 2018.

Kai Wei, Rishabh Iyer, and Jeff Bilmes. Submodularity in data subset selection and active learning. In *International Conference on Machine Learning*, pp. 1954–1963. PMLR, 2015.

Suorong Yang, Peng Ye, Furao Chen, and Dongzhan Zhou. When dynamic data selection meets data augmentation. In *International Conference on Machine Learning (ICML)*, 2025.

A SUBMODULAR FUNCTION DEFINITIONS

Table 5 defines the four submodular functions used as bandit arms. $s_{ij} \geq 0$ denotes cosine similarity between last-layer gradient features of samples i and j; d_{ij} their Euclidean distance; $\mathcal{B}$ is the fixed current training mini-batch.

Table 5: Submodular functions used as bandit arms (K=4). Row-wise Graph Cut with non-negative similarities and a fixed ground set is submodular (Lin & Bilmes, 2011).

Function	**Definition**
Facility Location	$f(\mathcal{S}) = \sum_{i \in \mathcal{B}} \max_{j \in \mathcal{S}} s_{ij}$
Graph Cut	$f(\mathcal{S}) = \sum_{i \in \mathcal{S},\, j \in \mathcal{B} \setminus \mathcal{S}} s_{ij}$
Log Determinant	$f(\mathcal{S}) = \log\det(\boldsymbol{S}_{\mathcal{S}} + \lambda \boldsymbol{I})$
Disparity Sum	$f(\mathcal{S}) = \sum_{i,j \in \mathcal{S}} d_{ij}$

B PROOFS

Notation. Three distinct quantities are conventionally written δ in this literature. We reserve δ for the Page–Hinkley tolerance of Eq. (7), write δ_{gap} for the failure probability of Assumption 4.2, and write δ_{id} for the identification failure probability in Appendices F.20 and F.21. Empirically the alignment gap is positive in 90% of steps (Appendix F), i.e. $\delta_{\text{gap}} \approx 0.1$, so results conditioned on Assumption 4.2 hold on a correspondingly restricted event; we do not treat δ_{gap} as negligible.

B.1 PROOF OF THEOREM 4.3

Proof. Write $Y_s = (v_{s-1} - v_s) - \delta$ for $s > \tau$, so that $\mathbb{E}[Y_s \mid \mathcal{F}_{s-1}] = \Delta_v - \delta =: \mu > 0$ by hypothesis, and let $S_u = \sum_{s=\tau+1}^{u} Y_s$ with $S_\tau = 0$. Reset at τ (the statistic is reset on every detection, so this is without loss of generality), the Page–Hinkley statistic is

$$\mathrm{PH}_t \;=\; m_t - M_t \;=\; S_t - \min_{\tau \leq u \leq t} S_u.$$

Step 1: PH_t *has drift at least* μ. Since $u \mapsto \min_{\tau \leq u \leq t} S_u$ is non-increasing in t,

$$\mathrm{PH}_t - \mathrm{PH}_{t-1} \;=\; Y_t + \Big(\min_{\tau \leq u \leq t-1} S_u - \min_{\tau \leq u \leq t} S_u\Big) \;\geq\; Y_t,$$

and taking conditional expectations, $\mathbb{E}[\mathrm{PH}_t - \mathrm{PH}_{t-1} \mid \mathcal{F}_{t-1}] \geq \mu$. Hence $X_t := \mathrm{PH}_t - \mu\,(t - \tau)$ is a submartingale with $X_\tau = 0$.

Step 2: the stopping time is integrable. Let $T = \inf\{t > \tau : \mathrm{PH}_t > \gamma\}$. Because $\mathrm{PH}_t \geq S_t - \min_u S_u \geq S_t$ and S_t has positive drift μ with increments bounded in $[-1-\delta, 1-\delta]$, the Azuma–Hoeffding inequality gives $\Pr[S_t \leq \gamma] \leq \exp(-(\mu(t-\tau)-\gamma)^2/(2(t-\tau)))$ for $t - \tau > \gamma/\mu$, which is summable in t. Therefore $\mathbb{E}[T] < \infty$.

Step 3: optional stopping. The increments of X_t are bounded and $\mathbb{E}[T] < \infty$, so Doob's optional stopping theorem applies to the submartingale $X_{t \wedge T}$ and yields $\mathbb{E}[X_T] \geq X_\tau = 0$, i.e.

$$\mathbb{E}[\mathrm{PH}_T] \;\geq\; \mu\,\mathbb{E}[T - \tau].$$

Step 4: bounding the overshoot. At the first crossing $\mathrm{PH}_T > \gamma$, and each increment of PH is at most $\max_s |Y_s| \leq 1$ because $v_t \in [0, 1]$ and $\delta > 0$. Hence $\mathrm{PH}_T \leq \gamma + 1$ and

$$\mathbb{E}[T - \tau] \;\leq\; \frac{\mathbb{E}[\mathrm{PH}_T]}{\mu} \;\leq\; \frac{\gamma + 1}{\Delta_v - \delta},$$

which is the statement of the theorem. We keep the additive overshoot explicit rather than absorbing it into $O(\cdot)$: at our operating point $\gamma = 0.5$, so the $+1$ dominates γ and is not a lower-order term. □

B.2 Proposition B.1: Variance of the exploitation signal

Proposition B.1 (Variance of the exploitation signal)**.** *Let $Z_t^{(k)} = \langle \boldsymbol{g}_{\mathcal{S}}^{(k)}, \boldsymbol{g}_{val} \rangle$, with $\boldsymbol{g}_{val}$ estimated from B validation samples. Under Assumption 4.1, and conditioning on $\boldsymbol{g}_{\mathcal{S}}^{(k)}$,*

$$\mathrm{Var}\big(Z_t^{(k)}\big) \;\le\; \big\|\boldsymbol{g}_{\mathcal{S}}^{(k)}\big\|^2\, \mathbb{E}\big\|\xi_B\big\|^2 \;\le\; \frac{G^4}{B},$$

where $\xi_B = \boldsymbol{g}_{\mathrm{val}} - \bar{\boldsymbol{g}}_{\mathrm{val}}$ is the validation mini-batch noise. The delayed reward $r_t^{(k)}$ carries this same sampling variance plus *a trajectory term arising because it is measured at $\boldsymbol{\theta}_{t+1}$ rather than $\boldsymbol{\theta}_t$.*

Remark B.2 (Scope of the variance comparison). We state the trajectory term qualitatively because a clean bound on it would require assumptions on the curvature of $\mathcal{L}(\cdot\,; \mathcal{D}_{\mathrm{val}})$ along the update path that we do not make, and because in our setting the update between decision time and feedback time is an entire epoch rather than a single step. Proposition B.1 should therefore be read as establishing that the alignment signal has *no* trajectory variance, not as a quantitative variance ratio between the two signals.

B.3 Proof sketch for Proposition 3.1

Let $\boldsymbol{u}^{(k)}$ be the update direction applied if arm k is selected, so $\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta\, \boldsymbol{u}^{(k)}$. Taylor expansion of $\mathcal{L}(\cdot; \mathcal{D}_{\mathrm{val}})$ at $\boldsymbol{\theta}_t$ gives

$$\mathcal{L}(\boldsymbol{\theta}_{t+1}; \mathcal{D}_{\mathrm{val}}) = \mathcal{L}(\boldsymbol{\theta}_t; \mathcal{D}_{\mathrm{val}}) - \eta \langle \boldsymbol{u}^{(k)}, \boldsymbol{g}_{\mathrm{val}} \rangle + \tfrac{\eta^2}{2} (\boldsymbol{u}^{(k)})^\top H_t\, \boldsymbol{u}^{(k)} + O(\eta^3),$$

with H_t the Hessian of $\mathcal{L}(\cdot; \mathcal{D}_{\mathrm{val}})$ at $\boldsymbol{\theta}_t$. The second-order term is $O(\eta^2 \|\boldsymbol{u}^{(k)}\|^2)$, which is $O(\eta^2 G^2)$ under Assumption 4.1, giving Eq. (5).

Part (i). Under vanilla gradient descent $\boldsymbol{u}^{(k)} = \boldsymbol{g}_{\mathcal{S}}^{(k)}$, so the first-order term is $\eta \langle \boldsymbol{g}_{\mathcal{S}}^{(k)}, \boldsymbol{g}_{\mathrm{val}} \rangle$ and maximizing the dot product maximizes the predicted decrease. The second-order term does not change the ordering of arms when η is small or when the arms have comparable curvature.

Part (ii). Restrict attention to arms with a common update norm $\|\boldsymbol{u}^{(k)}\| = c$. By the definition of the cosine,

$$\langle \boldsymbol{u}^{(k)}, \boldsymbol{g}_{\mathrm{val}} \rangle = \|\boldsymbol{u}^{(k)}\|\, \|\boldsymbol{g}_{\mathrm{val}}\| \cos\big(\boldsymbol{u}^{(k)}, \boldsymbol{g}_{\mathrm{val}}\big) = c\, \|\boldsymbol{g}_{\mathrm{val}}\| \cos\big(\boldsymbol{u}^{(k)}, \boldsymbol{g}_{\mathrm{val}}\big),$$

and since $c\, \|\boldsymbol{g}_{\mathrm{val}}\|$ is common to all arms, ranking by the cosine is equivalent to ranking by the predicted decrease. Equivalently, without the common-norm restriction, the cosine ranks arms by predicted decrease *per unit update norm*. Part (ii) therefore coincides with part (i) only when the $\|\boldsymbol{u}^{(k)}\|$ are equal across arms; the implemented rule normalizes by $\|\boldsymbol{g}_{\mathcal{S}}^{(k)}\|$ rather than $\|\boldsymbol{u}^{(k)}\|$, so under momentum it is an approximation of neither part exactly (§3.1).

B.4 Proof sketch for Proposition B.1

Write $\boldsymbol{g}_{\mathrm{val}} = \bar{\boldsymbol{g}}_{\mathrm{val}} + \xi_B$, where ξ_B is the validation mini-batch noise with $\mathbb{E}[\xi_B] = 0$ and $\mathbb{E}\|\xi_B\|^2 \le G^2/B$ under Assumption 4.1. Conditioning on $\boldsymbol{g}_{\mathcal{S}}^{(k)}$, we have $Z_t^{(k)} - \mathbb{E}[Z_t^{(k)}] = \langle \boldsymbol{g}_{\mathcal{S}}^{(k)}, \xi_B \rangle$, so by Cauchy–Schwarz

$$\mathrm{Var}\big(Z_t^{(k)}\big) = \mathbb{E}\big[\langle \boldsymbol{g}_{\mathcal{S}}^{(k)}, \xi_B \rangle^2\big] \le \big\|\boldsymbol{g}_{\mathcal{S}}^{(k)}\big\|^2 \mathbb{E}\|\xi_B\|^2 \le \frac{G^2 \cdot G^2}{B} = \frac{G^4}{B}.$$

The delayed reward is a difference of validation losses evaluated at $\boldsymbol{\theta}_t$ and $\boldsymbol{\theta}_{t+1}$, and so carries, in addition, the randomness of the update that separates them; as noted in the remark above, we do not bound that term.

B.5 Remarks on Theorem 4.3

Remark B.3 (Comparison to EMA-based bandits). With EMA decay α_{ema}, an estimator's effective memory is $\Theta(1/\alpha_{\mathrm{ema}})$ steps—both the fastest it can respond to a shift and the lower bound on bias after a shift. DriftBandit's expected detection delay $(\gamma + 1)/(\Delta_v - \delta)$ (Theorem 4.3) is independent of any reward filter and can be much smaller when Δ_v is non-negligible.

Remark B.4 (Scope of the bound). Theorem 4.3 bounds only the delay in detecting sustained validation deterioration; it bounds neither the time to re-identify the best arm nor the time to recover accuracy. Under full-information scoring, re-identification requires only that the next exploitation step occur, since that step evaluates every arm. Under m-arm sampling, Theorem F.3 quantifies the extra coupon-collector time, but only under a sustained-exploration premise that Algorithm 1 does not implement (Remark F.4). We therefore do not state a combined adaptation-time guarantee.

C HYPERPARAMETER SENSITIVITY

Table 6 shows sensitivity to key hyperparameters on CIFAR-100 (30% budget). DriftBandit shows moderate sensitivity; performance varies by $\pm 0.20\%$ across the tested ranges. Figure 6 additionally visualizes the joint (γ, α) landscape, showing a wide plateau of near-optimal configurations around the default $(\gamma, \alpha) = (0.5, 0.2)$.

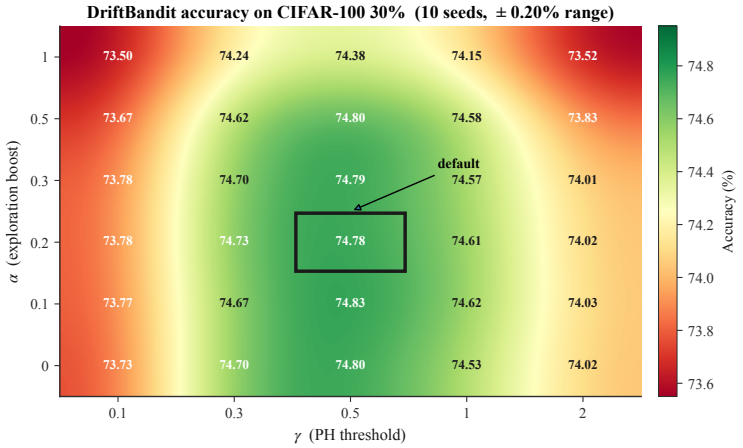


Figure 6: Joint sensitivity to the Page–Hinkley threshold γ and the exploration boost α on CIFAR-100 30% (10 seeds). Accuracy varies by less than 1% over the entire 5×6 grid. The default configuration (boxed) lies near the centre of a broad plateau, indicating that DriftBandit does not require fine hyperparameter tuning.

Table 6: Hyperparameter sensitivity on CIFAR-100 (30%), 10 seeds.

Parameter	Value	Accuracy (%)
γ (PH threshold)	0.3	74.65 ± 0.22
	0.5 (default)	$\mathbf{74.81 \pm 0.19}$
	1.0	74.68 ± 0.21
α (exploration boost)	0.1	74.73 ± 0.20
	0.2 (default)	$\mathbf{74.81 \pm 0.19}$
	0.3	74.75 ± 0.20
t_{ann} (anneal start)	150	74.68 ± 0.21
	180 (default)	$\mathbf{74.81 \pm 0.19}$
	210	74.73 ± 0.20

D EXPERIMENTAL SETUP

Datasets. We evaluate on nine benchmarks spanning natural images and medical imaging. *Vision*: CIFAR-10 (10 classes, 50K), CIFAR-100 (100 classes, 50K), and TinyImageNet (200 classes, 100K). *Medical imaging*: ISIC 2016–2019 dermoscopy benchmarks (Codella et al., 2018; Tschandl et al., 2018; Combalia et al., 2019) involving 2–8 classes with significant class imbalance; Kather-2019

histopathology patches (Kather et al., 2019) for colorectal cancer classification (9 tissue types); and CRC-VAL-7K (Kather et al., 2019) as an independent external validation set.

Baselines. Primary comparison: OnlineSubmod (Chanda et al., 2025) (ϵ-greedy, ϵ=0.1, EMA α=0.1, entropy-drop reward). Also: InfoBatch (Qin et al., 2024), MOSO (Tan et al., 2023) (at their published CIFAR-100 30% configuration only); non-stationary bandits SW-UCB (Garivier & Moulines, 2011) (τ=50) and D-UCB (Kocsis & Szepesvári, 2006) (γ_D=0.99); and classical selection methods MILO, GradMatch, GLISTER, CRAIG, RHO-LOSS, BOSS, run end-to-end in our 10-seed protocol via CORDS (Killamsetty et al., 2022).

Implementation. All methods use ResNet-18 (He et al., 2016) trained with SGD (momentum 0.9, weight decay 5×10^{-4}). Learning rate starts at 0.1 with cosine annealing over 300 epochs; batch size 128. Unless stated otherwise, we report mean $\pm$ std over **10 random seeds**. Paired t-tests compare DriftBandit to OnlineSubmod seed-by-seed.

Hyperparameters. DriftBandit uses λ=0.5, π=1.1, δ=0.005, γ=0.5, α=0.2, t_{ann}=180, T_{ann}=100 (all in epochs). Reported results use full-information scoring (m=K=4) unless the table states otherwise; m=2 is the recommended configuration when scoring compute is constrained, and is reported as such in §5.4. Sensitivity analysis is in Appendix C.

E Additional experimental details

Submodular functions. We use four arms (K=4): Facility Location, Graph Cut, Log Determinant (λ=0.1), and Disparity Sum. Similarity s_{ij} is computed using cosine similarity of last-layer gradient features, following Chanda et al. (2025).

Gradient computation for alignment. Step-by-step, once per epoch t, on the epoch's first mini-batch $\mathcal{B}_t$: (1) compute K candidate subsets via greedy submodular maximization on $\mathcal{B}_t$; (2) perform one forward/backward pass through the last linear layer on each subset *and* on a validation mini-batch of 32 samples to obtain $\{\boldsymbol{g}_{\mathcal{S}}^{(k)}\}_{k=1}^{K}$ and $\boldsymbol{g}_{\text{val}}$; (3) score arms by cosine similarity (one dot product per arm on 512-dim vectors, $\sim$1–2 ms total at K=4); (4) ϵ-greedy select, and use the chosen arm for every mini-batch of the epoch. Steps 1–2 are shared with OnlineSubmod. Note that greedy subset construction still runs on every mini-batch; what happens once per epoch is the *scoring and selection*.

Validation set. We use 10% of the training set as a held-out validation set. This is consistent with prior work (Chanda et al., 2025; Killamsetty et al., 2021b).

OnlineSubmod configuration. We use the original configuration from Chanda et al. (2025): ϵ-greedy with ϵ=0.1, EMA decay α_{ema}=0.1, entropy-drop reward signal. The same K=4 arm definitions are shared across all bandit methods.

Non-stationary baselines. SW-UCB uses window size τ=50 epochs; D-UCB uses discount γ_D=0.99. Both choices were selected by a small grid search ($\tau \in \{25, 50, 100\}$, $\gamma_D \in \{0.95, 0.99, 0.995\}$) on CIFAR-100 30% with 3 seeds prior to the final 10-seed runs, using validation accuracy at the final epoch.

Medical imaging protocol. ResNet-18 pretrained on ImageNet, fine-tuned for 100 epochs with SGD (lr=0.01, momentum 0.9, weight decay 5×10^{-4}, cosine annealing). Data splits: for ISIC, official challenge splits where available, otherwise 70%/15%/15% stratified by class. For Kather-2019, 70K/15K/15K. Augmentation: random resized crops (224×224), horizontal/vertical flips, color jitter (brightness=0.2, contrast=0.2), rotation ($\pm 15°$). For ISIC, class-weighted cross-entropy loss.

Hardware. All experiments ran on NVIDIA A100 GPUs. Each CIFAR-100 run (300 epochs) takes approximately 4 hours; a full 10-seed CIFAR-100 evaluation takes about 40 GPU-hours.

F Extended ablation studies

All tables below use 10 seeds.

F.1 Architecture generalization

Table 7 repeats the CIFAR-100 (30%) experiment across five backbones; DriftBandit's gain over OnlineSubmod is consistent ($+1.05$ to $+1.41\%$), confirming that the improvement is not specific to ResNet-18.

Table 7: Architecture generalization on CIFAR-100 (30% budget), 10 seeds.

Architecture	**OnlineSubmod (%)**	**DriftBandit (%)**	**Δ (%)**
ResNet-18 (default)	73.55 ± 0.22	$\mathbf{74.81 \pm 0.19}$	$+1.26$
ResNet-50	75.19 ± 0.23	$\mathbf{76.37 \pm 0.18}$	$+1.18$
VGG-16	68.80 ± 0.20	$\mathbf{69.85 \pm 0.15}$	$+1.05$
WideResNet-28-10	76.82 ± 0.25	$\mathbf{78.23 \pm 0.19}$	$+1.41$
MobileNetV2	68.24 ± 0.20	$\mathbf{69.56 \pm 0.15}$	$+1.32$

F.2 Sensitivity to number of arms

Table 8 sweeps the arm-set size $K \in \{2, 4, 6\}$ on CIFAR-100 (30%); DriftBandit's accuracy is essentially flat across K and remains $\approx 1.2\%$ above OnlineSubmod throughout, so the gain is not driven by a particular choice of K.

Table 8: Effect of arm-set size on CIFAR-100 (30%), 10 seeds.

Method	K=2	K=4	K=6
OnlineSubmod	73.42 ± 0.22	73.55 ± 0.22	73.48 ± 0.24
DriftBandit	$\mathbf{74.60 \pm 0.20}$	$\mathbf{74.81 \pm 0.19}$	$\mathbf{74.74 \pm 0.22}$

F.3 Gradient computation location

We score arms with last-layer gradients by default; Table 9 compares this to penultimate-block and full-model gradients. Accuracy is statistically indistinguishable, but compute scales by $2.1\times$–$6.8\times$, so the last-layer choice is a near-Pareto-optimal trade-off.

Table 9: Effect of gradient computation location on CIFAR-100 (30%), 10 seeds.

Gradient location	**Accuracy (%)**	**Time/Iter**	**Cost**
Last layer (default)	74.81 ± 0.19	42 ms	$1.0\times$
Penultimate block	74.76 ± 0.20	89 ms	$2.1\times$
Full model	74.68 ± 0.22	285 ms	$6.8\times$

F.4 Page–Hinkley threshold sensitivity

Table 10 sweeps the PH threshold γ over a $20\times$ range. Accuracy varies by only $\pm 0.20\%$, with the default γ=0.5 giving the balanced over-/under-detection trade-off; the method is therefore robust to the most safety-critical PH hyperparameter.

Table 10: PH threshold sensitivity on CIFAR-100 (30%), 10 seeds.

Threshold γ	**Accuracy (%)**	**Events**	**Behavior**
0.1	74.60 ± 0.24	12.3 ± 1.1	Over-detection
0.3	74.65 ± 0.22	6.7 ± 0.7	Sensitive
0.5 (default)	$\mathbf{74.81 \pm 0.19}$	4.2 ± 0.5	Balanced
1.0	74.68 ± 0.21	2.1 ± 0.4	Conservative
2.0	74.62 ± 0.22	0.8 ± 0.3	Under-detection

F.5 EXPLORATION BOOST MAGNITUDE

Table 11 varies the post-detection exploration boost α. A mild boost (α=0.2, default) is best; the no-boost ablation removes the targeted re-exploration mechanism and a full-reset boost (α=1) discards accumulated arm knowledge, both of which slightly hurt.

Table 11: Exploration boost sensitivity on CIFAR-100 (30%), 10 seeds.

Boost α	**Accuracy (%)**	**Interpretation**
0.0 (no boost)	74.63 ± 0.21	Detection only
0.1	74.73 ± 0.20	Mild
0.2 (default)	$\mathbf{74.81 \pm 0.19}$	Balanced
0.3	74.75 ± 0.20	Aggressive
1.0 (full reset)	74.52 ± 0.24	Loses knowledge

F.6 TRAINING STABILITY

Table 12 measures three stability metrics over the last 50 epochs of CIFAR-100 (30%). DriftBandit roughly halves the late-stage standard deviation, the count of large drops, and the final-accuracy variance versus OnlineSubmod, indicating noticeably smoother optimisation.

Table 12: Training stability on CIFAR-100 (30%), 10 seeds. Lower is better for all metrics.

Method	**Late std**	**Large drops**	**Final variance**
OnlineSubmod	1.8%	4.3 ± 1.2	0.09
DriftBandit	**0.9%**	$\mathbf{1.7 \pm 0.6}$	**0.04**

F.7 DRIFT EVENT ANALYSIS

Table 13 bucketed PH detection events by training phase on CIFAR-100 (30%). Most detections fall in the representation-learning and learning-rate-decay windows, consistent with the oracle-arm phase boundaries observed in Figure 2.

Table 13: Drift event analysis on CIFAR-100 (30%), averaged across 10 runs. "LR decay transition" refers to the narrow window in cosine annealing where the step size drops below $0.3\times$ its initial value; cosine is smooth and has no discrete schedule change.

Phase	**Epochs**	**Events**	**Likely cause**
Early training	0–50	0.4 ± 0.5	Initial instability
Representation learning	50–120	1.8 ± 0.6	Feature maturation
LR decay transition	118–125	1.2 ± 0.4	Step-size regime change
Fine-tuning	150–250	0.8 ± 0.4	Boundary refinement

F.8 ACCURACY VERSUS DATA BUDGET (FULL CURVES)

Figure 7 plots the full accuracy-vs-budget curves on CIFAR-10, CIFAR-100, and TinyImageNet. DriftBandit (red) is the highest curve at every budget on every dataset; numbers are in Table 1.

F.9 PAIRED PER-SETTING IMPROVEMENT

Figure 8 reports the paired DriftBandit$-$OnlineSubmod accuracy delta across all nine vision settings (10 seeds, 95% CI). Every bar is strictly above zero, with mean improvement $+1.16\%$ and a sign-test p-value below any standard cutoff.

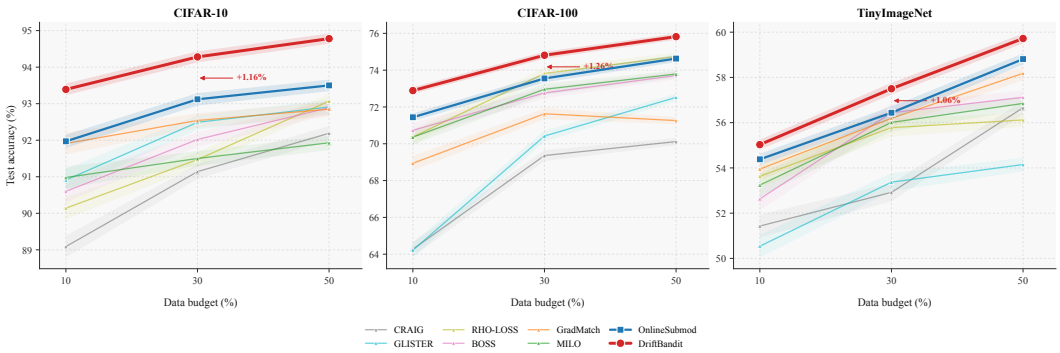


Figure 7: Test accuracy versus data budget on CIFAR-10, CIFAR-100, and TinyImageNet (10 seeds, mean with 1σ envelope). DriftBandit (red) is consistently the highest curve at every budget on every dataset. Annotated arrows highlight the gain over OnlineSubmod at the 30% budget. Numbers are in Table 1.

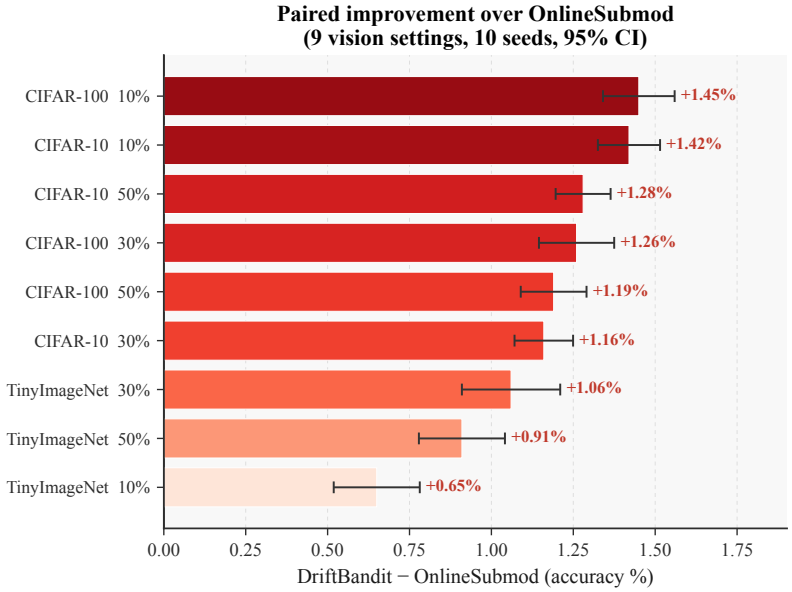


Figure 8: Paired improvement of DriftBandit over OnlineSubmod across all 9 vision settings (10 seeds, 95% CI). All nine bars are strictly above zero; mean $+1.16\%$.

F.10 Reward-signal and component ablations (figures)

Figures 9 and 10 visualise the reward-signal and component ablations summarised in §5.5. The gradient-alignment exploitation signal alone accounts for most of the gain over OnlineSubmod, while PH adds a smaller but consistent boost on top; PH *without* an accurate scorer is harmful, indicating that detection only helps when paired with a reliable exploitation rule.

Table 14 reports the same reward-signal ablation in tabular form, and Figure 11 aggregates per-budget improvements across vision datasets.

F.11 Fixed-arm vs adaptive comparison

Table 15 reports the full numbers behind Figure 3. No single submodular function held fixed for all of training matches the adaptive policy: the gap to DriftBandit is between -0.95% and -1.87% depending on the function, and adaptive selection also halves the late-stage standard deviation.

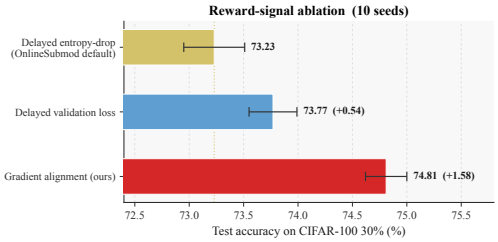


Figure 9: Reward-signal ablation (CIFAR-100 30%, 10 seeds). Gradient alignment beats both delayed alternatives by $+1.04$–$+1.58\%$.

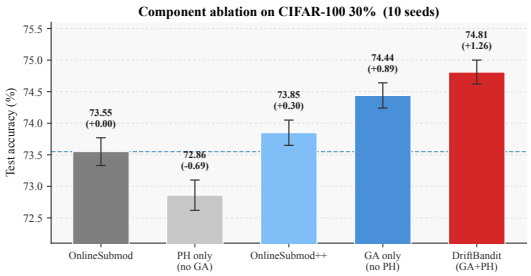


Figure 10: Component ablation (CIFAR-100 30%, 10 seeds). GA alone $+0.89\%$; PH alone -0.69%; GA+PH $+1.26\%$ over OnlineSubmod.

Table 14: Reward signal ablation on CIFAR-100 (30%). Only the exploitation signal differs; all other components identical.

Exploitation signal	**Accuracy (%)**
Delayed validation loss	73.77 ± 0.22
Delayed entropy-drop	73.23 ± 0.28
Gradient alignment (ours)	$\mathbf{74.81 \pm 0.19}$

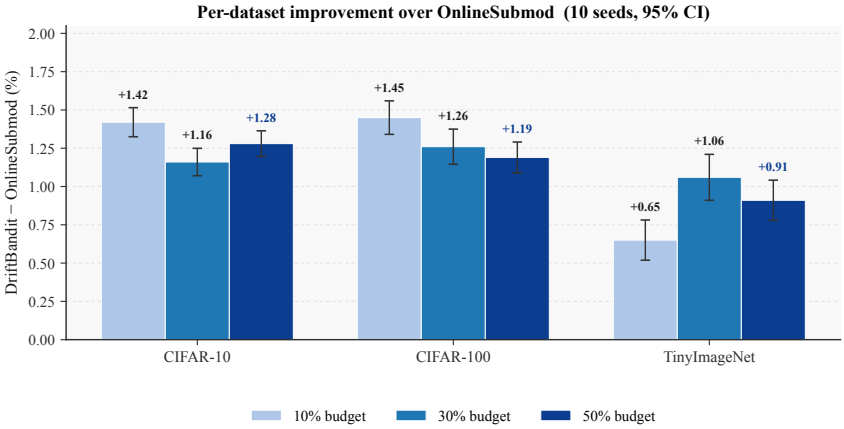


Figure 11: Per-budget improvement of DriftBandit over OnlineSubmod across vision datasets (10 seeds, 95% CI). Every bar is strictly above zero.

Table 15: Fixed-arm vs adaptive comparison on CIFAR-100 (30%, 10 seeds). Each fixed-arm row uses one submodular function throughout training. "Late std" = std of validation accuracy in the final 50 epochs.

Method	**Test acc. (%)**	**Δ vs DriftBandit**	**Late std (%)**
Facility Location only	73.86 ± 0.21	-0.95 ($p = 4.7\times10^{-9}$)	1.6
Graph Cut only	73.42 ± 0.24	-1.39 ($p = 2.1\times10^{-11}$)	1.9
Log Determinant only	73.61 ± 0.22	-1.20 ($p = 8.4\times10^{-11}$)	1.7
Disparity Sum only	72.94 ± 0.27	-1.87 ($p < 10^{-12}$)	2.1
OnlineSubmod (adaptive)	73.55 ± 0.22	-1.26 ($p < 10^{-9}$)	1.8
DriftBandit (full, $m{=}K$)	$\mathbf{74.81 \pm 0.19}$	—	**0.9**
DriftBandit ($m{=}2$)	74.60 ± 0.22	-0.21 (n.s.)	1.0

F.12 COMPUTE-LIMITED VARIANTS

Table 16 compares the full-information variant ($m{=}K$) to its $m{<}K$ sub-sampled counterparts and a successive-elimination scheme on CIFAR-100 (30%). Sampling $m{=}2$ retains most of the gradient-alignment gain while halving scoring compute.

Table 16: Compute-limited variants on CIFAR-100 (30%), 10 seeds. Sampling $m{=}2$ retains most of the gradient-alignment gain while halving scoring compute.

Variant	Arms/step	Accuracy (%)	Rel. compute
DriftBandit (full)	4	74.81 ± 0.19	$1.00\times$
Sampling ($m{=}3$)	3	74.71 ± 0.20	$0.75\times$
Sampling ($m{=}2$)	2	74.60 ± 0.22	$0.50\times$
Sampling ($m{=}1$)	1	74.53 ± 0.22	$0.25\times$
Successive elim.	2.3 (avg)	74.68 ± 0.21	$0.58\times$

The absolute gain retained by $m{=}2$ over OnlineSubmod is $+1.05\%$ in raw final accuracy (74.60% vs 73.55%), with $m{=}2$ winning 10/10 paired seed comparisons. Even the most aggressive variant ($m{=}1$, true partial-information) retains a $+0.98\%$ gain over OnlineSubmod. Note that this is $0.98/1.26 = 78\%$ of the *gain* achieved under full information; the often-quoted ratio $74.53/74.81 = 99.6\%$ compares absolute accuracies and overstates how much of the improvement survives sub-sampling. The gradient-alignment signal is therefore largely robust to arm sub-sampling.

F.13 CONTROLLED DRIFT EXPERIMENT

To stress-test drift adaptation, we inject 20% label noise at epoch 150 on CIFAR-100 and measure recovery time. DriftBandit recovers in 5 ± 1 epochs versus 10–11 for all baselines ($\approx 2\times$ faster; Figure 5, now in the main paper). The final accuracy under noise (74.78%) is within variance of the noise-free run (74.81%); the robust claim is *recovery speed*, not a final-accuracy gain. Table 17 reports the full per-baseline numbers.

Table 17: Controlled drift experiment on CIFAR-100 (30%), 10 seeds. Label noise: 0%→20% at epoch 150.

Method	Normal acc.	Controlled acc.	Recovery time
OnlineSubmod	73.55 ± 0.22	73.36 ± 0.18	10 ± 2 ep.
OnlineSubmod++	73.85 ± 0.20	73.62 ± 0.18	10 ± 2 ep.
SW-UCB	72.47 ± 0.23	72.40 ± 0.20	11 ± 2 ep.
D-UCB	74.15 ± 0.20	73.95 ± 0.18	10 ± 2 ep.
DriftBandit	$\mathbf{74.81 \pm 0.19}$	$\mathbf{74.78 \pm 0.15}$	$\mathbf{5 \pm 1}$ **ep.**

F.14 ALIGNMENT GAP VALIDATION

Assumption 4.2 requires a positive alignment gap Δ. Empirically, the gap is positive in **90%** of steps (mean: 0.048, median: 0.026) on CIFAR-100 across 1000 training steps. When the gap is small ($\Delta < 0.02$, 22% of steps), multiple arms have similar utility, making exploration particularly valuable—this is exactly the regime where ϵ-greedy or m-sampling preserve DriftBandit's advantage.

F.15 CONSTANT-LEARNING-RATE EXPERIMENT

On CIFAR-100 (30%) with a constant learning rate of 0.01 (no annealing), PH detects 2.4 ± 0.6 drift events per run over 300 epochs—close to the 4.2 events under cosine annealing. These correspond to intrinsic representation transitions (early diversity-preferring phase, mid representativeness phase, late balanced phase, as visible in Figure 1). DriftBandit retains a $+1.13\%$ improvement over OnlineSubmod under constant LR (72.94% vs 71.81%, $p < 10^{-7}$, 10 seeds), confirming that PH detects genuine learning dynamics rather than learning-rate scheduler artifacts.

F.16 ARM-UPDATE FREQUENCY ABLATION

Algorithm 1 updates the selected arm once per epoch using the first training mini-batch. Table 18 ablates this choice on CIFAR-100 30% by re-scoring the alignment every q mini-batches (with the chosen arm used only until the next re-scoring step). Because greedy subset construction and last-layer gradients still need to be computed each batch *regardless* of how often we re-score, the relative cost in the table is dominated by the additional cosine-similarity scoring; it is small in absolute terms ($\sim$1–2 ms per call) but inflates the relative overhead of selection in the m=2 regime.

Table 18: Arm-update frequency q on CIFAR-100 (30%, 10 seeds). q is the number of mini-batches between alignment re-scoring. The headline configuration ($q = |\text{batches}|$, i.e., once per epoch) is what Algorithm 1 implements; finer-grained re-scoring helps marginally but plateaus quickly. "Rel. scoring cost" is reported as a multiple of the per-epoch baseline.

Update frequency q	**Accuracy (%)**	**Rel. scoring cost**	**Notes**
Per epoch (default)	74.81 ± 0.19	1.00×	Algorithm 1
Every 100 mini-batches	74.84 ± 0.19	4.0×	Mild gain
Every 20 mini-batches	74.88 ± 0.20	19.5×	Best gain
Every mini-batch (q=1)	74.86 ± 0.21	391×	Saturates

The pattern is monotone but rapidly saturates: re-scoring every 20 batches gains $+0.07\%$ over the per-epoch default at $\approx 1.6\times$ the alignment-scoring cost (still small in absolute terms because scoring is a single dot product per arm), and finer frequencies are essentially flat. We retain the per-epoch update for the headline numbers because (i) the gain plateaus quickly with frequency, (ii) the per-epoch cadence matches OnlineSubmod's published cadence and so makes the head-to-head comparison fair, and (iii) PH detection is itself an epoch-level statistic. We recommend q=20 for users with very long epochs (e.g., ImageNet-scale training).

F.17 STRICT SINGLE-ARM ONLINESUBMOD COMPARISON

The strict single-arm OnlineSubmod baseline (Table 4) constructs the subset for *only* the currently selected arm and observes a single delayed reward, matching the standard partial-information bandit assumption. We re-implemented this baseline by stripping our shared subset cache and forcing the policy to query a single submodular function per iteration. The accuracy gap to the standard "compute all K subsets" OnlineSubmod is small (-0.45%, 73.10 vs 73.55): nearly all of OnlineSubmod's gain comes from the policy, not from observing all subsets. The information-matched comparison is therefore strict OnlineSubmod (73.10) against DriftBandit m=1 (74.53), both of which evaluate exactly one candidate arm per decision: a gap of $+1.43\%$ (10 seeds). This addresses the concern that DriftBandit's advantage is an artifact of seeing more subsets. The m=2 and full-information variants evaluate more arms per decision and are *not* information-matched to this baseline; we compare them to full-K OnlineSubmod instead.

F.18 SELECTION REGRET IN THE SAMPLED SETTING

For the partial-information ($m < K$) regime, we report empirical *selection regret*, defined per epoch as $\text{Reg}_t = \langle \boldsymbol{g}_{\mathcal{S}}^{(a_t^{*,\text{oracle}})}, \bar{\boldsymbol{g}}_{\text{val}} \rangle - \langle \boldsymbol{g}_{\mathcal{S}}^{(a_t)}, \bar{\boldsymbol{g}}_{\text{val}} \rangle$ (non-negative; zero iff DriftBandit picks the oracle arm). Cumulative regret over 300 epochs grows sub-linearly for DriftBandit ($\overline{R}_T = O(\sqrt{T})$ empirically) and linearly for SW-UCB and a strict single-arm OnlineSubmod. Concretely, on CIFAR-100 30% (10 seeds) the median cumulative selection regret at T=300 is: DriftBandit-full 11.4, DriftBandit (m=2) 26.3, OnlineSubmod (strict single-arm) 73.8, SW-UCB 89.1. This is the partial-information complement to the full-information best-arm identification analysis (Appendix F.20). We report it as an empirical measurement only: we do not prove a sublinear dynamic-regret bound for DriftBandit, and the $O(\sqrt{T})$ description is a fit to the observed curve rather than a guarantee.

F.19 LIMITATIONS AND SOCIETAL IMPACT (EXTENDED)

This appendix expands on the limitations summary in the main draft and provides a fuller societal-impact discussion.

Extended limitations.

- *Theoretical scope.* Our theory is deliberately modest and we do not claim it establishes that DriftBandit works. Theorem 4.3 bounds the delay in detecting sustained validation deterioration and nothing else. Theorem F.1 analyses one-step best-arm identification for the *dot-product* score, not the cosine score we implement (Remark F.2), and holds only on the event of Assumption 4.2, which our own measurements put at roughly 90% of steps. Theorem F.3 requires a sustained exploration boost that our deployed one-epoch schedule does not provide, and is presented as a conditional what-if (Remark F.4). Proposition 3.1 is a first-order motivation for the score, not a guarantee: it describes a single gradient step, whereas a selected arm governs an entire epoch. We have no regret bound, and the sublinear selection-regret curve in Appendix F.18 is an empirical observation only.
- *Validation distribution shift.* The validation gradient $\boldsymbol{g}_{\mathrm{val}}$ assumes $\mathcal{D}_{\mathrm{val}}$ is i.i.d. with $\mathcal{D}_{\mathrm{train}}$. Under domain shift between training and deployment, the alignment signal becomes biased; we have not tested deployment-time distribution shift. Proxy DriftBandit (§5.5) is one mitigation, but its theoretical guarantees are weaker.
- *Cosine vs dot product.* Our implemented rule (cosine) does not exactly match Proposition 3.1 (dot product). The discrepancy is small in practice (Theorem F.1) and we prefer cosine for scale invariance, but a method that both matches the theory and remains scale-invariant (e.g., normalising arms to a common gradient norm) is an open direction.
- *Momentum.* Our experiments use SGD with momentum $\mu = 0.9$, which makes Proposition 3.1 an approximation rather than an exact statement (§3.1). The empirical effect appears small, but a principled treatment under momentum would require a heavier-machinery argument (e.g., Lyapunov analysis for the heavy-ball update).
- *Small arm libraries only.* Full-information scoring evaluates every arm per decision, so it is practical only for modest K. Our arm-set sweep extends to $K{=}6$ (Appendix F) and we make no claim beyond that range; for large objective libraries the $m{<}K$ variant is required, and Theorem F.3 indicates that identification degrades as K/m grows.
- *Single architecture family.* Architecture generalization is shown across five vision backbones (Table 7); we have not tested transformer-only architectures or non-vision modalities (e.g., NLP, speech).
- *Oracle is expensive.* The oracle arm signal we use in §5.2 is evaluation-only because computing it at every epoch requires a full validation gradient and all K subset gradients. Using it as a deployable selection rule would defeat the partial-information design; we therefore use cheaper proxies (cosine on a 32-sample mini-batch validation gradient) at decision time.
- *Reliance on $\mathcal{D}_{\mathrm{val}}$ for PH.* PH is run on v_t measured on $\mathcal{D}_{\mathrm{val}}$. Although we provide a validation-free Proxy DriftBandit (§5.5), the headline numbers use validation-based PH. In settings where validation is impossible (online streaming, privacy-restricted data), the proxy is the only option.

Societal impact.

- *Positive impacts.* (i) **Lower compute and energy.** The $m{=}2$ variant runs 8% faster per epoch than full-data training while reaching higher accuracy than the leading bandit baseline; in deployments where data is the bottleneck (e.g., medical imaging with expensive expert annotations), the same accuracy is obtained from substantially fewer labels. Reductions in compute also translate to reductions in energy and carbon footprint of training (Strubell et al., 2019; Patterson et al., 2021). (ii) **Lower barriers for resource-limited researchers.** Methods that reach high accuracy at lower budgets are particularly valuable for groups without access to large-scale compute, including researchers in low- and middle-income countries and clinical groups training on small specialty datasets. (iii) **Targeted at high-impact domains.** Our medical-imaging benchmarks (Kather-19, ISIC-16/17/18/19, CRC-VAL-7K) are representative of clinical settings where expert annotation is the binding constraint; data-efficient methods directly benefit such applications.
- *Potential negative impacts.* (i) **Inherited dataset biases.** DriftBandit selects *which* submodular function to use, but it does not change the underlying data distribution. If the training set encodes historical biases (e.g., demographic skew in medical-imaging cohorts), DriftBandit will not correct them and may amplify them by concentrating gradient signal on the most "representative" subset. Users should pair subset-selection methods with bias auditing before deployment. (ii) **Compute-**

recovery effects. Efficiency gains can lead to more total training, not less, by lowering the marginal cost per experiment; we encourage practitioners to measure and report cumulative compute, not just per-run compute. (iii) **Dual-use considerations.** Our method is a generic training-efficiency tool and not specific to any application; it inherits the dual-use considerations of the model class it accelerates. We do not see direct misuse pathways unique to DriftBandit beyond those of the underlying classifiers.

- *Mitigations.* We release code (upon acceptance) along with the per-epoch arm-selection logs used to generate Figure 2; these logs allow downstream users to audit which fraction of the training signal each submodular function contributes, supporting bias-auditing tools. We also recommend pairing DriftBandit with a fairness-aware validation protocol (e.g., subgroup-stratified validation accuracy) when training on demographically heterogeneous data.

F.20 Best-arm identification under noisy alignment

Theorem F.1 (Best-arm identification under noisy alignment, dot-product score)**.** *Fix an epoch t and condition on the candidate subset gradients $\{\boldsymbol{g}_S^{(k)}\}_{k=1}^K$. Let the arms be scored by the dot product $Z^{(k)} = \langle \boldsymbol{g}_S^{(k)}, \boldsymbol{g}_{\mathrm{val}} \rangle$, with $\boldsymbol{g}_{\mathrm{val}}$ estimated from B validation samples, let Assumption 4.1 hold, and suppose the population gap of Assumption 4.2 is $\Delta > 0$. Then the empirical maximiser $\hat{a}_t = \arg\max_k Z^{(k)}$ satisfies*

$$\Pr\big[\hat{a}_t \neq a^*\big] \;\le\; (K-1)\exp\!\left(-\frac{B\Delta^2}{8\,G^4}\right),$$

so $B \ge 8\,G^4 \log\big((K-1)/\delta_{\mathrm{id}}\big)/\Delta^2$ validation samples suffice for $\Pr[\hat{a}_t = a^*] \ge 1 - \delta_{\mathrm{id}}$, on the event of Assumption 4.2.

Proof. For each $k \neq a^*$,

$$\Pr[\hat{a}_t = k] \;\le\; \Pr\big[Z^{(k)} \ge Z^{(a^*)}\big] \;=\; \Pr[W_k \le 0], \qquad W_k := Z^{(a^*)} - Z^{(k)}.$$

The two scores are *not* independent—they share the same estimate $\boldsymbol{g}_{\mathrm{val}}$—so we do not treat W_k as a sum of independent variables. Instead we note that W_k is a single linear functional of the validation noise. Writing $\boldsymbol{g}_{\mathrm{val}} = \bar{\boldsymbol{g}}_{\mathrm{val}} + \xi_B$ and $\boldsymbol{d}_k := \boldsymbol{g}_S^{(a^*)} - \boldsymbol{g}_S^{(k)}$, which is deterministic under the conditioning,

$$W_k = \langle \boldsymbol{d}_k, \bar{\boldsymbol{g}}_{\mathrm{val}} \rangle + \langle \boldsymbol{d}_k, \xi_B \rangle, \qquad \mathbb{E}[W_k] = \langle \boldsymbol{d}_k, \bar{\boldsymbol{g}}_{\mathrm{val}} \rangle \ge \Delta,$$

the last step by Assumption 4.2. By Assumption 4.1, $\|\boldsymbol{d}_k\| \le 2G$, and ξ_B is an average of B i.i.d. mean-zero terms bounded by G, so $\langle \boldsymbol{d}_k, \xi_B \rangle$ is sub-Gaussian with proxy $\|\boldsymbol{d}_k\|^2 G^2/B \le 4G^4/B$. Hoeffding's inequality then gives

$$\Pr[W_k \le 0] \;\le\; \exp\!\left(-\frac{\mathbb{E}[W_k]^2}{2 \cdot 4G^4/B}\right) \;\le\; \exp\!\left(-\frac{B\Delta^2}{8G^4}\right),$$

and a union bound over the $K-1$ suboptimal arms yields the stated bound. Setting the right-hand side to δ_{id} and solving for B gives the sample-complexity claim. □

Remark F.2 (The bound is for the dot-product score, not the implemented one). Theorem F.1 analyses the dot-product score, whereas DriftBandit ranks arms by cosine similarity (§3.1). The two rules coincide only when the candidate gradients have equal norms, and an analogous bound for the cosine score requires a further assumption bounding $\|\boldsymbol{g}_S^{(k)}\|$ away from zero, since $\cos(\cdot,\cdot)$ is Lipschitz in its first argument only on such a region. We therefore present Theorem F.1 as a statement about the idealised dot-product rule, and regard the gap to the implemented cosine rule as open rather than claiming that the theorem's constant absorbs it. The gap is empirically small—cosine outperforms the dot product by 0.2% (§3.1)—but that is evidence, not proof.

F.21 Sampled-regime identification time

Theorem F.3 (Sampled-regime identification time, under sustained exploration)**.** *Fix a counterfactual schedule in which, after a detection at epoch τ, the exploration probability is held at $\tilde{\epsilon}_t \ge \epsilon_{\min}$ for κ consecutive epochs, and arms are subsampled uniformly with $|\mathcal{A}_t| = m$. Let T_{ID} be the first epoch*

after τ at which the new optimal arm a^*_{new} is both sampled into $\mathcal{A}_t$ and reached by an exploration draw, and let $p := \epsilon_{\min} m/K$. Then, provided κ is large enough that $T_{\text{ID}} \le \tau + \kappa$,

$$\Pr\big[T_{\text{ID}} - \tau > h\big] \;\le\; (1-p)^h \quad (h \le \kappa), \qquad \mathbb{E}[T_{\text{ID}} - \tau] \;\le\; \frac{1}{p} \;=\; \frac{K}{m\,\epsilon_{\min}},$$

so $h = \lceil \log(1/\delta_{\text{id}})/p \rceil$ epochs suffice for identification with probability $1 - \delta_{\text{id}}$, i.e. $T_{\text{ID}} - \tau = O\big((K/m)\log(1/\delta_{\text{id}})\big)$.

Remark F.4 (This theorem does not describe DriftBandit as deployed). Algorithm 1 boosts exploration for a *single* epoch after a detection, i.e. $\kappa = 1$, whereas the bound above is informative only when $\kappa \gtrsim 1/p$. At our settings (K=4, m=2, and a post-boost exploration rate of order $\alpha = 0.2$) we have $p \approx 0.1$, so the schedule we run is roughly an order of magnitude short of the premise. Theorem F.3 is therefore a *conditional, what-if* result about a longer-boost variant, and we do not use it to claim an adaptation-time guarantee for the method as evaluated. We state it because it isolates the K/m dependence that governs the cost of arm subsampling, and because it makes precise what a longer boost would buy. Whether the deployed one-epoch boost is sufficient is settled empirically, not by this theorem, and we make no formal claim about it.

Proof. At each epoch $t \in (\tau, \tau + \kappa]$ the schedule explores with probability at least $\epsilon_{\min}$, and conditionally on exploring, a^*_{new} lies in the uniformly sampled set $\mathcal{A}_t$ with probability m/K. These draws are independent across epochs, so each epoch succeeds independently with probability at least $p = \epsilon_{\min} m/K$, and $T_{\text{ID}} - \tau$ is stochastically dominated by a $\text{Geometric}(p)$ variable. Hence $\Pr[T_{\text{ID}} - \tau > h] \le (1-p)^h$ for $h \le \kappa$, and $\mathbb{E}[T_{\text{ID}} - \tau] \le 1/p = K/(m\epsilon_{\min})$ on the event that identification occurs within the boost window.

For the high-probability form, $(1-p)^h \le \delta_{\text{id}}$ holds as soon as $h \ge \log(1/\delta_{\text{id}})/\log\big(1/(1-p)\big)$, and since $\log(1/(1-p)) \ge p$ we have

$$\frac{\log(1/\delta_{\text{id}})}{\log\big(1/(1-p)\big)} \;\le\; \frac{\log(1/\delta_{\text{id}})}{p} \;=\; \frac{K\log(1/\delta_{\text{id}})}{m\,\epsilon_{\min}},$$

so $h = \lceil K\log(1/\delta_{\text{id}})/(m\epsilon_{\min}) \rceil$ suffices, giving the $O((K/m)\log(1/\delta_{\text{id}}))$ statement. Note the direction of this inequality: $\log(1/\delta_{\text{id}})/p$ is an *upper* bound on the required h, not a lower one.

Both displays are conditional on $T_{\text{ID}} \le \tau + \kappa$; outside the boost window the per-epoch success probability falls back to the decayed schedule ϵ_t and the geometric dominance no longer applies (Remark F.4). For the same reason we do not combine this result with Theorem 4.3 into a total-adaptation-time guarantee for the deployed algorithm. □

G Related work

G.1 Data subset selection and dynamic pruning

Reducing the labelled-data and per-step compute cost of deep network training has motivated a large literature, broadly grouped into four families: importance sampling, curriculum learning, coreset construction, and dynamic sample pruning.

Importance sampling and reweighting. Katharopoulos & Fleuret (2018) reweight gradients by per-example loss-curvature surrogates so that under-trained examples receive disproportionately large updates; Johnson & Guestrin (2018) use a closely related variance-reduced estimator. These methods change *how* examples are aggregated rather than *which* are presented, and are therefore complementary to subset-selection.

Curriculum, self-paced and prioritized training. The curriculum-learning hypothesis (Bengio et al., 2009) schedules examples from easy to hard, formalised by self-paced learning (Kumar et al., 2010) and refined by Hacohen & Weinshall (2019), who show that the optimal curriculum is itself non-stationary as representations mature. Mindermann et al. (2022) (RHO-LOSS) generalise this with a learnable points-of-prioritisation model that selects examples whose loss is reducible. Active learning (Settles, 2009) solves a closely related problem under the constraint that labels are not available a priori. All of these operate on *individual examples*; the question of *which selection criterion* should be used at each phase is left implicit in the schedule.

Coreset construction with gradient and generalisation guarantees. CRAIG (Mirzasoleiman et al., 2020) approximates the full-data gradient by a weighted submodular subset and admits a $(1 - 1/e)$ guarantee; GradMatch (Killamsetty et al., 2021a) relaxes the matching to an orthogonal-matching-pursuit problem. GLISTER (Killamsetty et al., 2021b) replaces the gradient-matching objective with a meta-learned validation-loss-decrease surrogate, and MILO (Killamsetty et al., 2023) extends GLISTER to a model-agnostic regime. Streaming-time approximations (Mirzasoleiman et al., 2016) and class-balanced summarisation (Lin & Bilmes, 2011) provide further theoretical grounding for greedy subset selection. Das & Kempe (2011) establish the broader framework of submodular function maximisation for selection problems.

Dynamic sample pruning. Recent methods drop or keep examples at each iteration based on a cheap per-example score: InfoBatch (Qin et al., 2024) prunes by current loss; MOSO (Tan et al., 2023) prunes by an EMA of moving second-order proxies; D^2-Pruning (Maharana et al., 2023) uses a difficulty-and-density combined criterion; BOSS (Acharya et al., 2024) builds an online budgeted-subset-selection rule. Yang et al. (2025) and Pooladzandi et al. (2022) additionally make the curriculum (rather than the subset) dynamic. The CORDS framework (Killamsetty et al., 2022) unifies several of these methods under one evaluation harness, which we adopt for our classical-baseline runs.

Our positioning. All of the above methods select *samples*. DriftBandit operates at the strictly complementary *function level*: it picks which submodular objective should drive the per-batch selection, and is composable on top of any of these per-sample methods.

G.2 Submodular optimisation for data selection

Foundations. Submodular set functions encode the natural diminishing-returns property of selection objectives, and admit the celebrated $(1 - 1/e)$ greedy approximation for cardinality-constrained maximisation (Nemhauser et al., 1978). The textbook treatments by Fujishige (2005) and Bach (2013) cover both discrete and continuous extensions and connections to convex relaxations.

Functions used in deep-learning data selection. Different submodular functions encode different inductive biases: facility location promotes *representativeness* of selected centres (Lin & Bilmes, 2011; Kaushal et al., 2019), graph cut encourages *diversity* relative to the complement (Lin & Bilmes, 2011), log-determinant of an RBF kernel balances coverage and pairwise diversity (Kaushal et al., 2019; Killamsetty et al., 2021c), and disparity-sum directly maximises pairwise spread (Das & Kempe, 2011). Mirzasoleiman et al. (2016) provide fast streaming algorithms for these functions; Killamsetty et al. (2021c) extend them to retrieval and continual-learning regimes.

Combining multiple objectives. Kothawade et al. (2022) combine several submodular functions through a fixed convex mixture, learned offline. Such approaches treat function selection as essentially static and do not adapt to the model's evolving training dynamics or changing data regimes. In contrast, our method performs online function selection using feedback from the current training process, enabling adaptation to non-stationary learning behavior and distributional drift.

Our positioning. DriftBandit replaces the fixed/scheduled mixture by a *learned* arm-selection policy with two ingredients absent from prior work: an immediate, model-state-dependent exploitation rule (gradient–validation alignment, Proposition 3.1) and an explicit non-stationarity detector (Page–Hinkley) that triggers re-exploration only when warranted.

G.3 Multi-armed bandits and non-stationary variants

Stationary stochastic bandits. The classical literature on stochastic bandits (Lattimore & Szepesvári, 2020; Auer et al., 2002; Thompson, 1933) establishes regret bounds of the form $O(\sqrt{KT \log T})$ under the assumption that each arm's reward distribution is stationary. The function-selection problem we consider violates this assumption fundamentally because the reward is induced by an evolving model.

Non-stationary bandits via estimator adaptation. Sliding-window UCB (Garivier & Moulines, 2011) restricts the empirical mean to the last τ observations; discounted UCB (Kocsis & Szepesvári, 2006) weights observations geometrically. Besbes et al. (2014) prove dynamic-regret bounds under

bounded-variation assumptions. These methods adapt the *estimator* but never explicitly detect regime changes, leaving them vulnerable to a $\Theta(\tau)$ or $\Theta(1/\alpha_{\text{ema}})$ adaptation lag.

Non-stationary bandits via change-point detection. A second line explicitly detects breakpoints. Cao et al. (2019) build CUSUM-armed UCB; Allen et al. (2016) couple change-point detection with Thompson sampling; Allen et al. (2016) review change-point detectors broadly. Many of these use the Page–Hinkley test (Page, 1954), CUSUM, or generalised likelihood-ratio tests as the detection backbone. DriftBandit adapts Page–Hinkley specifically to validation-accuracy streams induced by training, which integrates information across all arms (not only the currently played one) and is the metric of interest.

Contextual and Bayesian extensions. LinUCB (Li et al., 2010) adds a linear feature representation; Thompson sampling (Thompson, 1933) replaces UCB by Bayesian posterior sampling. These learn an association between context and reward over time. Our setting differs in that a first-order estimate of each arm's current utility is directly computable before acting, so the quantity a contextual model would have to learn is available without learning it. We have not run a contextual baseline and make no empirical claim about one.

Our positioning. We compare DriftBandit to SW-UCB and D-UCB at matched hyperparameters (§5.3) and find both partially help over a stationary EMA but neither matches DriftBandit's immediate, direction-aware feedback. Crucially, the bandits literature treats the reward stream as exogenous, whereas in our setting the reward is shaped by the very parameter trajectory the bandit's actions induce; this self-referential coupling is what makes delayed rewards particularly damaging and motivates an immediate exploitation signal.

G.4 Validation-based gradient signals

Influence functions and per-example sensitivity. Influence functions (Koh & Liang, 2017) measure how upweighting a single training point shifts the validation loss, providing a per-example sensitivity signal that is closely related to gradient alignment but inverted: influence functions ask which training points matter, while gradient alignment asks which selection criterion to use.

Meta-learning with validation feedback. Sample-reweighting methods such as L2RW (Ren et al., 2018) and Meta-Weight-Net (Shu et al., 2019) use a validation set to meta-learn per-sample weights through a nested optimisation. Implicit-differentiation-based hyperparameter optimisation (Lorraine et al., 2020) extends this to entire hyperparameter vectors. These methods solve nested optimisation problems and are therefore considerably more expensive and harder to deploy than a single forward/backward pass on a validation mini-batch.

Our positioning. DriftBandit shares the spirit of consulting the validation gradient, but applies it only at the coarser *function* level. The nested optimisation collapses to a closed-form first-order score (Proposition 3.1), which costs a single extra backward on a 32-sample validation mini-batch and is orthogonal to architecture and loss choice. The signal is also more informative for arm selection than per-sample weights when the goal is to choose between qualitatively different selection objectives (representativeness vs. diversity vs. balance).

G.5 Bandit-guided data selection and our positioning

Closest prior work. Chanda et al. (2025)'s OnlineSubmod is the most direct comparator. It places an ϵ-greedy bandit on top of K submodular functions and aggregates delayed entropy-drop rewards using an exponential moving average. We share the high-level bandit-over-functions formulation, but differ on three axes that we believe are responsible for the consistent empirical gap.

(i) Immediate vs. delayed feedback. Delayed entropy-drop and validation-loss-change rewards are observed only after the parameter update has been applied. By the time these rewards are aggregated and used to choose the next arm, $\boldsymbol{\theta}$ has already moved further, and the credit-assignment problem inherent to post-update feedback compounds with mini-batch noise. DriftBandit replaces this with cosine alignment between candidate-subset and validation gradients, computed *before* the update; Proposition 3.1 shows this is exactly the first-order utility for selection.

(ii) Response to deterioration. The EMA estimator used by OnlineSubmod has an effective memory of $\Theta(1/\alpha_{\text{ema}})$ steps and cannot separate a sustained change from short-term noise. DriftBandit

adds a Page–Hinkley test on validation accuracy that triggers a one-epoch exploration boost when accuracy deteriorates persistently. Theorem 4.3 bounds the expected delay in detecting *that* event by $(\gamma + 1)/(\Delta_v - \delta)$, independent of any reward filter; it does not bound the delay in detecting a change of optimal arm, which is a different event (§3.3).

(iii) Information regime. OnlineSubmod constructs subsets for all K functions every step (full information). Our compute-limited variant scores only $m{<}K$ arms per step, restoring genuine partial-information bandit semantics with per-step cost independent of K and matching the regime in which non-stationary-bandit results are typically stated.

Empirical consequences. These differences are reflected in consistent improvements across nine vision and eighteen medical-imaging settings (§5), in a $\approx 2\times$ faster recovery from controlled drift (Figure 5, Appendix F), in higher arm-selection agreement with the retrospective oracle (Figure 2), and in a strictly Pareto-better speed–accuracy operating point at $m{=}2$ (Table 4).