GOAT-AL: Explaining Budget Transitions with Irrelevant Variance, for All-Budget Active Learning

Yoel Marcu
School of Computer Science and Engineering
The Hebrew University of Jerusalem
yoel.marcu@mail.huji.ac.il

Daphna Weinshall
School of Computer Science and Engineering
The Hebrew University of Jerusalem
daphna@mail.huji.ac.il

Abstract

Deep active learning faces a basic tension: covering the data works best when labels are scarce, whereas uncertainty sampling works best at larger budgets; the best strategy therefore depends on the labeling budget. Existing methods blend these objectives heuristically, without explaining when or why to switch.

We explain the switch through *irrelevant variance*: the part of within-class feature variance that does not contribute to the linear separation of class means. With few labels, networks can fit the labeled set by increasing this variance before learning semantic features that separate the classes; uncertainty sampling prolongs this stage, delaying representation maturation. As the budget grows, variance shifts toward class separation, making uncertainty sampling effective. We detect this transition without ground-truth labels using *pseudo-CDNV (pCDNV)*, a label-free score inspired by neural collapse and computed from k-means pseudo-labels; its peak marks the onset of the transition.

Building on this finding, we introduce *Geometry-Oriented Adaptive Targeting (*GOAT-AL*)*, which uses pCDNV to adapt a single coverage objective rather than schedule between coverage and uncertainty. The objective uses a composite kernel mixing self-supervised and task-aligned kernels, with the weight shifting toward task-aligned features as the representation matures. On CIFAR-10, CIFAR-100, TinyImageNet, and ImageNet-100, *GOAT-AL* matches or surpasses existing all-budget methods across labeling regimes. Additionally, we show that mixing the same kernels with a weight determined by the labeled fraction of the pool, without using label values or tuning, yields a k-nearest-neighbor classifier that outperforms both the standard softmax head and kNN on frozen self-supervised features.

1 Introduction

In pool-based active learning (AL), the goal is to select a subset of unlabeled data that maximizes model performance under a fixed labeling budget. The optimal strategy depends on the budget: at low budgets, label-agnostic **coverage** of the data manifold is best, whereas at high budgets **uncertainty** near the decision boundary becomes the most effective signal [16]. This *budget regime* dependency has driven "all-budget" methods that interpolate between the two objectives [24, 4], but interpolation introduces a fragile meta-problem: deciding *when* and *how fast* to swap objectives.

We address the question of *why* the transition occurs and *when*, tracing it to **irrelevant variance**: within-class feature variance orthogonal to the class-mean subspace, which therefore does not contribute to the linear separation of the class means. With few labels, an over-expressive network fits the labeled set by inflating this variance, a memorization shortcut, rather than separating classes; by preferentially querying marginal points, uncertainty sampling prolongs this behavior. As labels accumulate, variance shifts toward class separation. This *turnaround* marks the onset of the budget transition: beyond it, querying uncertain examples becomes progressively less harmful.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

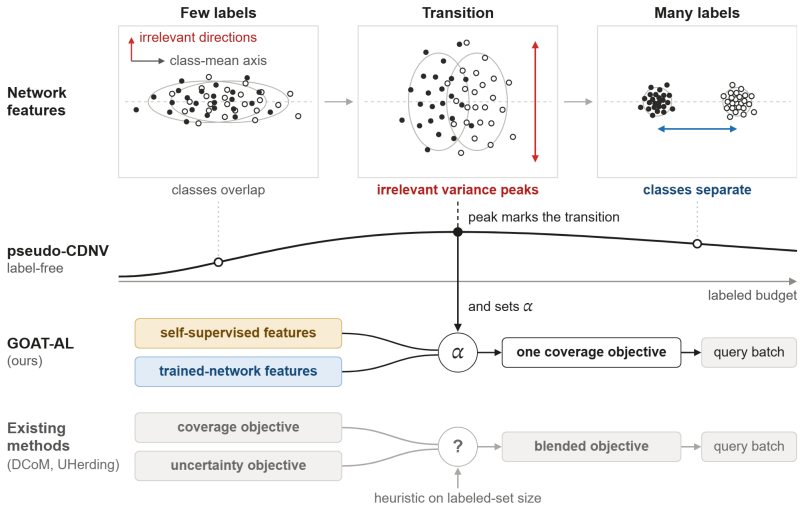


Figure 1: **One coverage objective, an automatically adapting target space.** *Top:* As the labeling budget grows, classes in the network's feature space progress from overlapping to separated; the irrelevant variance orthogonal to the class-mean subspace (red) peaks at this transition. *Middle: pCDNV* tracks this peak without ground-truth labels and uses it to set the blend coefficient α between self-supervised and trained-network features. *Bottom: GOAT-AL* consistently maximizes a single coverage objective over this evolving blend, rather than switching between coverage and uncertainty according to a labeled-set-size heuristic, as existing hybrid methods do.

Because the transition is an event in the learned representation rather than a property of the budget, it can be detected without labels. We introduce *pseudo-CDNV (pCDNV)*, a label-free score inspired by neural collapse [27, 12] computed on k-means pseudo-labels. Its trajectory exhibits a characteristic rise, peak, and decline that tracks the irrelevant-variance lifecycle, and its **peak marks the turnaround**, a principled, label-free trigger for adapting the selection strategy.

Building on this signal, we propose Geometry-Oriented Adaptive Targeting (*GOAT-AL*) (Figure 1), a query strategy that operates across all budgets using a single coverage objective. Rather than switching between objectives, *GOAT-AL* maximizes coverage in a composite feature space that shifts from a frozen self-supervised representation to the network's evolving penultimate-layer representation, with *pCDNV* signaling when this transition should occur. A single coverage objective thus effectively addresses both the low- and high-budget regimes.

Interestingly, we show that a composite kernel with the same components but a classification-specific blend coefficient also provides a strong *classifier*. When used as the similarity measure for k-nearest-neighbor classification over the labeled set, it relies primarily on the self-supervised representation when labels are scarce, and increasingly on the trained representation as more labels become available. This classifier outperforms both the standard softmax head and k-nearest-neighbor classification using the frozen SSL kernel alone.

Our contributions:

1. We provide a mechanistic account of the budget transition in AL, tracing it to a turnaround in the *irrelevant variance* of the trained representation, and validate this account empirically and causally.
2. We introduce *pCDNV*, a label-agnostic, neural-collapse-inspired score whose peak detects this turnaround without access to ground-truth labels.
3. We propose *GOAT-AL*, a purely coverage-based query strategy that uses a single coherent objective across all labeling budgets, matching or outperforming state-of-the-art all-budget methods on four datasets.
4. We show that *GOAT-AL*'s composite kernel also yields a k-nearest-neighbor classifier over the labeled set, with a hyperparameter-free blend, that substantially outperforms both the softmax head and k-nearest-neighbor classification using the frozen SSL kernel alone.

2 Related Work

Low-Budget vs. High-Budget Strategies. Early in the AL process, uncertainty-based methods, such as Entropy Sampling [35], Least Confidence [22], and Margin Sampling [30], often underperform because the model's predictive signal is unreliable and prone to overfitting a small labeled set [16]. In this regime, diversity-based or "representative" strategies that cover the data manifold using self-supervised features are generally superior [16, 37, 3]. However, as the labeled budget grows and the model matures, the dominant signal for performance gain shifts toward model uncertainty, which targets specific samples near the decision boundary to refine classification margins [e.g., 8, 18, 36, 11, 29, 19, 33, 2, 32].

Hybrid and Adaptive Methods. To address this shift, several "all-budget" methods have been proposed. Early attempts used fixed heuristics to combine density and uncertainty [9]. More recent frameworks like SelectAL [15] and TCM [10] treat the transition as a model selection problem, choosing between different AL strategies based on estimated performance. Methods like *DCoM* [24] and *UHerding* [4] instead utilize a continuous interpolation between coverage and uncertainty objectives, using a competence score derived from coverage in a frozen self-supervised space [24], or the product of an uncertainty function and a coverage kernel [4]. Unlike these approaches, which require switching or blending disparate optimization targets, our method (*GOAT-AL*) maintains a consistent coverage objective while adapting the underlying geometry using signals derived from the training dynamics.

Neural Collapse and Training Dynamics. Our work leverages the phenomenon of Neural Collapse (NC) [27], where penultimate-layer features simplify into a symmetric geometric structure at the end of training. While NC was initially studied as an asymptotic property of balanced, fully supervised datasets, recent work has explored its role in generalization [12] and its use as a query signal in active learning [13], though not targeting the all-budget regime and operating through a fundamentally different mechanism. We extend these insights with *pCDNV*, a *label-free* score constructed analogously to the Class-Distance Normalized Variance (*CDNV*), formally defined in Section 3, which is a widely adopted scalar measure of NC introduced by Galanti et al. [12].

Concurrent work. Machnio et al. [23] interpret changes in the relative effectiveness of active-learning strategies through shifts in the dominant components of a PAC-style generalization bound. Our complementary account identifies a geometric mechanism in the learned representation whose turnaround marks when task-aligned features become useful. The pCDNV peak detects this event online, without ground-truth labels, and directly adapts a single deployable coverage objective, avoiding strategy selection at retrospectively estimated breakpoints.

3 Background

3.1 Coverage-Based Active Learning at Low Budgets

Let $\mathcal{U} = \{x_1, \ldots, x_N\}$ denote the unlabeled pool and $\mathcal{L}$ the labeled set. Coverage-based AL selects points to maximize how well $\mathcal{L}$ "covers" $\mathcal{U}$ in some feature space.

***ProbCover*.** Yehuda et al. [37] defined coverage via a fixed radius δ: a point x is covered if $\exists x' \in \mathcal{L}$ with $\|g(x) - g(x')\| \leq \delta$ for a feature extractor g. The method greedily maximizes the number of covered points, and its performance is sensitive to the choice of δ.

***MaxHerding*.** Bae et al. [3] generalized coverage using a continuous kernel k_σ:

$$C_{k_\sigma}(\mathcal{L}) := \mathbb{E}_{\mathbf{x}}\left[\max_{\mathbf{x}' \in \mathcal{L}} k_\sigma(\mathbf{x}, \mathbf{x}'; g)\right] \approx \frac{1}{|\mathcal{U}|} \sum_{\mathbf{x} \in \mathcal{U}} \max_{\mathbf{x}' \in \mathcal{L}} k_\sigma(\mathbf{x}, \mathbf{x}'; g) =: \hat{C}_{k_\sigma}(\mathcal{L}), \tag{1}$$

where $k_\sigma(\mathbf{x}, \mathbf{x}'; g) = \exp(-\|g(\mathbf{x}) - g(\mathbf{x}')\|^2/\sigma^2)$ denotes the RBF kernel on features from a self-supervised extractor g (e.g., SimCLR [5]). *MaxHerding* greedily selects $x^* \in \arg\max_{\tilde{\mathbf{x}} \in \mathcal{U}} \hat{C}_{k_\sigma}(\mathcal{L} \cup \{\tilde{\mathbf{x}}\})$, which provides a $(1 - 1/e)$-approximation guarantee via submodularity.

3.2 All-Budget Active Learning

DCoM. Mishal and Weinshall [24] define a competence score $S_{\mathcal{L}}$, a sigmoid of the fraction of the pool covered by balls around the labeled points in a frozen self-supervised embedding, and select samples using: $R(x) = S_{\mathcal{L}} \cdot (1 - M(x)) + (1 - S_{\mathcal{L}}) \cdot \mathrm{ODR}(x)$, where $M(x)$ is the normalized margin and $\mathrm{ODR}(x)$ is the out-degree rank in a coverage graph. This linearly interpolates between uncertainty (margin) and coverage (out-degree rank) based on the competence score.

UHerding. Bae et al. [4] define uncertainty coverage as $\mathrm{UC}_{k_\sigma}(\mathcal{S}) = \mathbb{E}_{\mathbf{x}}[U(\mathbf{x}; f) \cdot \max_{\mathbf{x}' \in \mathcal{S}} k_\sigma(\mathbf{x}, \mathbf{x}'; g)]$, weighting the coverage kernel by an uncertainty function $U(\mathbf{x}; f)$. By adaptively adjusting the temperature τ (selected each episode from a preset grid by minimizing the calibration error of a model trained on a split of the labeled set) and length-scale σ (the minimum distance between labeled points), *UHerding* smoothly transitions from coverage-dominated to uncertainty-dominated behavior.

3.3 Neural Collapse and *CDNV*

Neural collapse [27] refers to the convergence of learned features to a structured geometric configuration during training. Specifically, within-class features collapse to their class means, and these means converge to a simplex equiangular tight frame (ETF).

The class-distance normalized variance (*CDNV*) [12] quantifies neural collapse:

$$\mathit{CDNV} = \frac{1}{K} \sum_{c=1}^{K} \frac{\mathbb{E}_{x \in \mathcal{X}_c}[\|h(x) - \mu_c\|^2]}{\frac{1}{K} \sum_{c'=1}^{K} \|\mu_c - \mu_{c'}\|^2}, \tag{2}$$

where $h(x)$ denotes the penultimate-layer features, μ_c the mean of class c, K the number of classes, $\mathcal{X}_c$ the set of samples in class c, $N_c = |\mathcal{X}_c|$, and $N = \sum_c N_c$. This is an aggregate K-class form of the pairwise *CDNV* of Galanti et al. [12]: under symmetry assumptions (balanced classes, equal within-class variances, and equiangular class means), it equals $K/(K-1)$ times the pairwise form (Proposition 4, Appendix A.4). Throughout, penultimate features are mean-centered and then ℓ_2-normalized before computing the variance decomposition, *CDNV*, pCDNV, and the adaptive kernel. Lower *CDNV* indicates more collapsed representations. While *CDNV* is typically tracked over training epochs, we track it over labeling budgets in active learning, training a model from scratch at each episode.

4 Method: Geometry-Oriented Adaptive Targeting

4.1 Composite Kernel Coverage

The core idea of *GOAT-AL* is to maximize generalized coverage in a feature space defined by a *composite kernel* that combines two sources of information:

1. **Static kernel** $k^{(s)}$: uses frozen self-supervised features g_s (e.g., SimCLR, DINOv2 [26]) to provide a fixed semantic representation that is not task-specific.
2. **Adaptive kernel** $k^{(a)}$: uses evolving penultimate-layer features g_a, which capture task-specific structure as labels accumulate.

We define the composite kernel as a convex combination:

$$k_{\mathrm{G}}(\mathbf{x}, \mathbf{x}') = \alpha \cdot \frac{k^{(s)}_{\sigma_s}(\mathbf{x}, \mathbf{x}'; g_s)}{\bar{k}^{(s)}} + (1 - \alpha) \cdot \frac{k^{(a)}_{\sigma_a}(\mathbf{x}, \mathbf{x}'; g_a)}{\bar{k}^{(a)}}, \tag{3}$$

where $\bar{k}^{(s)}$ and $\bar{k}^{(a)}$ denote the respective mean kernel values over the pool, used to normalize their scales, and $\alpha \in [0, 1]$ controls their relative contributions. Each component uses the RBF kernel: $k_\sigma(\mathbf{x}, \mathbf{x}'; g) = \exp(-\|g(\mathbf{x}) - g(\mathbf{x}')\|^2 / \sigma^2)$ with data-driven bandwidth σ (described in Appendix K). Section 4.3 describes how the blend coefficient α is automatically determined.

The selection objective remains greedy coverage maximization as in *MaxHerding*, operating on k_{G}:

$$\mathbf{x}^* \in \arg\max_{\tilde{\mathbf{x}} \in \mathcal{U}} \left[\frac{1}{|\mathcal{U}|} \sum_{\mathbf{x} \in \mathcal{U}} \max\Big(\max_{\mathbf{x}' \in \mathcal{L}} k_{\mathrm{G}}(\mathbf{x}, \mathbf{x}'),\ k_{\mathrm{G}}(\mathbf{x}, \tilde{\mathbf{x}}) \Big) - \frac{1}{|\mathcal{U}|} \sum_{\mathbf{x} \in \mathcal{U}} \max_{\mathbf{x}' \in \mathcal{L}} k_{\mathrm{G}}(\mathbf{x}, \mathbf{x}') \right], \tag{4}$$

Because the objective retains the facility-location form with a fixed nonnegative similarity k_{G} as defined in (3), it remains monotone submodular, and the $(1-1/e)$ approximation guarantee of greedy maximization carries over.

Coverage recovers uncertainty-like behavior. Decreasing α does more than interpolate between representations: it can steer *GOAT-AL*'s coverage objective toward uncertainty-like sampling, without ever leaving the coverage framework. When $\alpha = 1$, *GOAT-AL* reduces to greedy coverage on static SSL features, equivalent to *MaxHerding*, but with a data-driven bandwidth σ^* (Appendix K) instead of a fixed $\sigma = 1$. As $\alpha \to 0$, it instead covers the learned representation, which captures discriminative structure at higher budgets, reminiscent of core-set [31].

Why: a Neural Collapse argument. Under Neural Collapse, labeled features concentrate near their class centroids and classifier weights align with those centroids. A bound on the logit margin then shows that points far from every labeled centroid have small margins. Adaptive coverage therefore favors uncertainty-like sampling as this geometry emerges. The bound and its assumptions are given in Appendix L; the empirical distance–margin correlation strengthens from $\rho = -0.53$ at 500 labels to $\rho = -0.93$ at 25,000 on CIFAR-100 (Table 2).

4.2 *pCDNV*: A Label-Free Maturity Signal

The weight α in Eq. (3), derived from *pCDNV*, reflects the reliability of the adaptive features.

Motivation. *CDNV* (Eq. 2) is a natural measure of feature quality: a low value indicates that the penultimate-layer features have acquired a well-separated class structure, making them suitable for guiding coverage. However, *CDNV* requires ground-truth labels, which are not available for the unlabeled pool. We therefore replace them with pseudo-labels. Importantly, the resulting proxy is not an estimate of *CDNV* itself: it follows a qualitatively different, non-monotonic trajectory whose *peak*, rather than its absolute value, provides the useful label-free signal. We formalize this in Section 5.

Construction. We replace ground-truth labels with k-means pseudo-labels. At each active learning episode, after training the model on the current labeled set $\mathcal{L}$, we do the following:

1. Extract penultimate-layer features $h(x)$ for all the data $\mathcal{L} \cup \mathcal{U}$.
2. Run k-means with K clusters (matching the number of classes) to obtain pseudo-labels $\hat{y}(x)$.
3. Compute *CDNV* using pseudo-labels instead of true labels:

$$pCDNV = \frac{1}{K} \sum_{c=1}^{K} \frac{\mathbb{E}_{x:\hat{y}(x)=c}[\|h(x) - \hat{\mu}_c\|^2]}{\frac{1}{K}\sum_{c'=1}^{K} \|\hat{\mu}_c - \hat{\mu}_{c'}\|^2}, \tag{5}$$

where $\hat{\mu}_c$ denotes the centroid of pseudo-cluster c.

Budget-wise trajectory of *pCDNV*. As the labeling budget grows, *pCDNV* follows a characteristic rise-peak-decline trajectory, whereas *true CDNV* starts high and decreases monotonically (Figure 2D; full trajectory in Figure 13). We attribute this to the fact that at low budgets, k-means finds deceptively compact clusters in the overfit, tightly clustered penultimate-layer representation, yielding an artificially low score. As the representation begins to resolve the classes, *pCDNV* rises to a peak; once the true class structure dominates, it declines in step with true *CDNV*. As shown in Section 5, this peak is the signal used by *GOAT-AL* and coincides with the turnaround in the network's irrelevant variance - the point beyond which the trained representation, and consequently uncertainty-based selection, begins to help.

4.3 From *pCDNV* to Kernel Weight

We use the *pCDNV* trajectory to set α, the weight of the static kernel. Once *pCDNV* peaks and begins to decline, weight shifts to the adaptive kernel in proportion to its drop from the peak.

Let P_e denote the current *pCDNV* at episode e, and $P^* = \max_{e' \le e} P_{e'}$ the running peak. We define:

$$\alpha_e = \left(\frac{P_e}{P^*}\right)^{\gamma}, \tag{6}$$

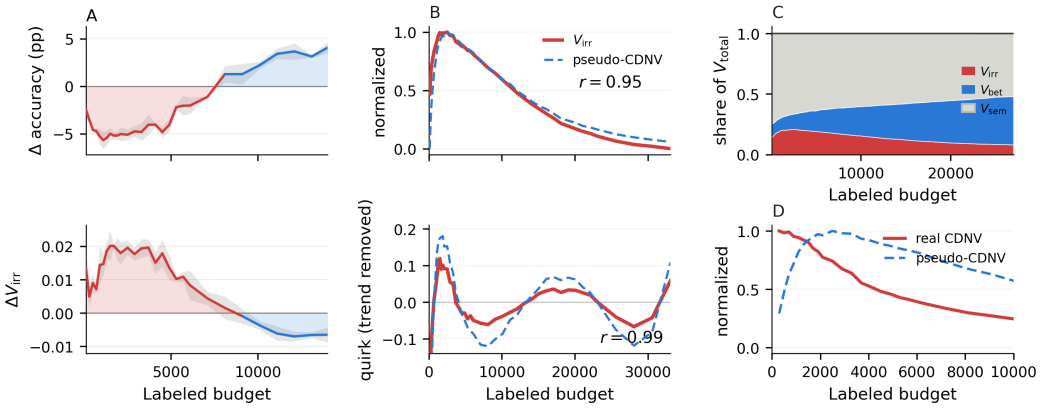


Figure 2: **The irrelevant-variance mechanism on CIFAR-100.** *A:* Margin minus MaxHerding: test-accuracy difference (top, percentage points) and irrelevant-variance difference (bottom) from the same runs. Uncertainty sampling has excess V_{irr} when it underperforms, and the gaps narrow together ($r = -0.97$); the pattern holds across all four datasets (Figure 4). *B:* Under random sampling, *pCDNV* tracks V_{irr} without selection bias. The curves are rescaled to $[0, 1]$ (top, $r = 0.95$); after subtracting a degree-4 polynomial trend from each curve, the residuals remain aligned (bottom, $r = 0.99$; see Appendix E). *C:* V_{sem} (gray) is reallocated to V_{irr} (red) and V_{bet} (blue), with nearly constant total variance. The *pCDNV* peak coincides with the V_{irr} maximum (see Appendix D). *D:* Under random sampling, true *CDNV* (solid) decreases monotonically, whereas *pCDNV* (dashed) rises, peaks, and declines. *GOAT-AL* uses the *pCDNV* peak as a label-free transition signal (see Section 4.2).

where $\gamma > 0$ controls sensitivity. When $P_e = P^*$ (at or before the peak), $\alpha_e = 1$ and *GOAT-AL* relies solely on the static kernel. As P_e falls below P^*, α_e decreases, gradually incorporating the adaptive kernel. The exponent γ (set to 5 in our experiments) sharpens this transition, ensuring rapid adoption of the adaptive kernel once the representation has matured; accuracy is not sensitive to this choice across $\gamma \in \{2, 3, 5, 8, 12\}$, all of which stay clearly above the coverage and uncertainty baselines (Appendix H), so γ is not finely tuned.

4.4 Algorithm Summary

Algorithm 1 Geometry-Oriented Adaptive Targeting (*GOAT-AL*)

Require: Unlabeled pool $\mathcal{U}$, initial labeled set $\mathcal{L}_0$, budget per round B, number of classes K, exponent γ, pre-extracted static features $g_s(x)$ for all $x \in \mathcal{U} \cup \mathcal{L}_0$

Compute static bandwidth σ_s via knee-point analysis of the purity curve on g_s
$P^* \leftarrow 0$ ▷ Running peak of *pCDNV*
for episode $e = 0, 1, 2, \ldots$ **do**
  Train model f_e on $\mathcal{L}_e$
  Extract penultimate features $g_a(x) = h_{f_e}(x)$ for all $x \in \mathcal{U} \cup \mathcal{L}_e$
  Compute pseudo-labels $\hat{y}(x)$ via K-means on g_a
  Compute $P_e \leftarrow \textit{pCDNV}(g_a, \hat{y})$ ▷ Eq. (5)
  $P^* \leftarrow \max(P^*, P_e)$
  $\alpha_e \leftarrow (P_e / P^*)^\gamma$ ▷ Eq. (6)
  Compute adaptive bandwidth σ_a via knee-point analysis of the purity curve on g_a
  Construct composite kernel k_{G} with weight α_e ▷ Eq. (3)
  Select B points greedily maximizing $\hat{C}_{k_{\text{G}}}(\mathcal{L}_e \cup \mathcal{S})$ ▷ Eq. (4)
  $\mathcal{L}_{e+1} \leftarrow \mathcal{L}_e \cup \mathcal{S}, \mathcal{U} \leftarrow \mathcal{U} \setminus \mathcal{S}$
end for

Algorithm 1 summarizes our method. Its key properties are: (i) the objective, coverage maximization, stays fixed; (ii) the feature space adapts automatically via α_e; (iii) no hyperparameter governs the switching point, which emerges from the *pCDNV* peak; and (iv) no auxiliary model is trained to set its parameters, unlike *UHerding*, which trains one at every episode.

Computational cost. We compare the per-episode cost of the selection step; network training follows the same protocol for all methods and is omitted. Let N be the pool size, d the feature dimension, B the batch size, K the number of clusters, and T the number of k-means iterations. *MaxHerding* builds the $N \times N$ RBF kernel on the SSL features, with a fixed bandwidth, at a cost of $O(N^2 d)$, and then selects B points greedily, which takes $O(BN^2)$. *GOAT-AL* performs the same greedy step on the composite kernel (Eq. (4)), so it builds a kernel on each of the two feature spaces, $O(N^2 d)$ each, and combines them in $O(N^2)$. Because *GOAT-AL* sets the bandwidth of each kernel from the data (Appendix K), unlike *MaxHerding*'s fixed bandwidth, it also runs k-means on each feature space, $O(NKdT)$, and finds for every point the nearest point of a different cluster, $O(N^2 d)$ per space. Finally, it computes *pCDNV* from the clustering of the adaptive features, $O(Nd + K^2 d)$. Both methods are therefore $O(N^2(d + B))$: *GOAT-AL*'s overhead is a constant factor (about four distance matrices instead of one) plus the $O(NKdT)$ clustering term, which is of lower order because $KT \ll N$.

4.5 Hyperparameter-Free Kernel Blending for k-NN

GOAT-AL's composite kernel can also serve as the similarity function in a k-nearest-neighbor classifier over the labeled set. Classification requires a different blend coefficient because it relies on a different view of the data: selection aims to maximize coverage and therefore views the entire pool, whereas classification uses only each query's top-k neighbors. We therefore calibrate the classification blend so that the frozen SSL kernel's share of similarity mass within those neighbors equals $1 - |\mathcal{L}|/N_0$, where $N_0 = |\mathcal{L}| + |\mathcal{U}|$ is the fixed total pool size. The share is pooled over query inputs and uses no labels. This label-free rule gradually shifts weight toward the adaptive kernel as the labeled budget grows, while retaining a positive SSL contribution whenever $|\mathcal{L}| < N_0$; a pure adaptive classifier is reached only at the full-pool endpoint. Note that α_e and the classification blend answer different questions: α_e identifies when the representation has matured enough to switch the selection strategy, as indicated by the pCDNV peak, while the classification blend coefficient determines how much each kernel contributes to the vote. Appendix J provides the exact calibration rule.

5 Why the Transition Happens: The Irrelevant-Variance Account

We now formalize the mechanism illustrated in Figure 1 (top): Sections 5.1-5.2 decompose the variance of the penultimate features and identify a class-irrelevant component whose rise-and-decline turnaround marks the budget transition. Section 5.3 shows that the pCDNV peak introduced in Section 4 detects this turnaround without labels, providing a label-free readout of the underlying conserved variance economy.

5.1 The Variance Budget

Let $h(x) \in \mathbb{R}^d$ denote the penultimate features, μ_c the mean of class c, μ_T the total mean, N_c the size of class c, and $V_{\text{total}} = \frac{1}{N}\sum_i \|h(x_i) - \mu_T\|^2$ the total variance. Define the *semantic subspace* $S = \text{span}\{\mu_c - \mu_T : c = 1, \ldots, K\}$, of dimension at most $K - 1$, with orthogonal projectors P_S onto S and $P_\perp$ onto $S^\perp$. Only a linear classifier's projection onto S contributes to separating class means; directions in $S^\perp$ are, by construction, invisible to class-mean separation.

Proposition 1 (Variance budget)**.** *The total penultimate variance decomposes as*

$$V_{\text{total}} \;=\; V_{\text{bet}} \;+\; V_{\text{sem}} \;+\; V_{\text{irr}}, \tag{7}$$

with between-class variance $V_{\text{bet}} = \frac{1}{N}\sum_c N_c \|\mu_c - \mu_T\|^2$*, within-class variance inside* S *-* $V_{\text{sem}} = \frac{1}{N}\sum_i \|P_S(h(x_i) - \mu_{y_i})\|^2$*, and within-class variance orthogonal to* S *-* $V_{\text{irr}} = \frac{1}{N}\sum_i \|P_\perp(h(x_i) - \mu_{y_i})\|^2$.

The identity follows by decomposing V_{total} into between- and within-class components and applying $I = P_S + P_\perp$ to each within-class residual (Appendix A.1). We call V_{irr} the *irrelevant variance*: the within-class variance in $S^\perp$, which does not contribute to linear separation of the class means.

Proposition 2 (Variance identity for normalized features)**.** *For unit-norm features, the total variance satisfies the exact identity*

$$V_{\text{total}} = 1 - \|\mu_T\|^2 \leq 1.$$

Corollary 1 (Conserved variance budget)**.** *If $\|\mu_T\|^2$ is constant across labeling budgets, then $V_{\text{bet}} + V_{\text{sem}} + V_{\text{irr}} = 1 - \|\mu_T\|^2$ is constant. In particular, if $\mu_T = 0$ at every budget, the sum equals 1. Any increase in between-class variance is then matched by an equal decrease in total within-class variance.*

In our runs, V_{total} equals 1.000 to three decimals throughout the labeling trajectory, so the constant-budget corollary applies approximately. The three components therefore trade off: increasing irrelevant variance reduces the variance available for within-subspace scatter and class separation.

5.2 The Irrelevant-Variance Shortcut

A network trained on too few labels is far more expressive than the labeled set demands, and can fit it in two ways: by separating classes in S, or by taking a shortcut that spreads individual samples in $S^{\perp}$, inflating V_{irr}. The overparameterized network prefers the shortcut. Typical points lie in dense, well-separated regions and can be fit without inflating V_{irr}, whereas marginal points near class boundaries are easier to fit by inflating it. Coverage sampling therefore starves the shortcut, while uncertainty sampling feeds it. The effect is consistent across datasets: uncertainty sampling carries excess irrelevant variance for exactly as long as it underperforms in accuracy, and the two gaps close together, with anti-correlation $r = -0.84$ to -0.97 on CIFAR-10, CIFAR-100, TinyImageNet, and ImageNet-100 (Figure 2*A*; per-dataset panels and correlations in Appendix B). This is correlational; to establish causation, a controlled experiment fixes the labeling budget, class balance, and architecture and varies only the marginality of the labeled set, which by itself is enough to make V_{irr} rise and accuracy fall (Appendix C).

Figure 2B shows that the link between *pCDNV* and V_{irr} persists under random sampling, without selection bias: the rescaled trajectories correlate at $r = 0.95$, and the residuals left after subtracting a degree-4 polynomial trend from each at $r = 0.99$. Figure 2C shows the variance reallocation itself. Before the turnaround, V_{sem} decreases while both V_{bet} and V_{irr} grow; afterward, V_{irr} also decreases as class separation continues to grow. The nearly constant total variance ties these changes together, and the *pCDNV* peak coincides with the V_{irr} maximum.

As labels accumulate, the network abandons the shortcut: it becomes harder to memorize every individual sample, and variance is reallocated from $S^{\perp}$ toward class separation. Similar reallocation has been documented across the layers of trained networks [1, 28]; here, we observe it as the labeling budget grows, with each episode trained from scratch, indicating that it is not an artifact of continued training. This turnaround, marked by a peak in V_{irr} followed by its decline and a rise in V_{bet}, marks the onset of the budget transition: beyond it, querying uncertain points becomes progressively less harmful (the accuracy crossover between Margin and MaxHerding occurs somewhat later; Figure 2A).

5.3 pCDNV Detects the Turnaround

For simplicity of the analysis, we work with the *aggregate* CDNV, $\mathit{CDNV} = W/(2V_{\text{bet}}) = \frac{1}{2\phi} - \frac{1}{2}$, where $W = V_{\text{sem}} + V_{\text{irr}}$ is the total within-class variance and $\phi = V_{\text{bet}}/V_{\text{total}}$ is the between-class share. Pooling the variances into two totals is what lets us express the score through the single quantity ϕ; it is a device for the analysis, not a different score to compute, and the pCDNV that drives the method is the one defined in Eq. (5). The aggregate form agrees with the pairwise CDNV of Galanti et al. [12] up to a factor $K/(K-1)$ under the symmetry assumptions of Proposition 4 (proved in Appendix A.4), and yields the exact identity with ϕ that the analysis needs. pCDNV replaces the true classes with k-means clusters, using the between-cluster share $\phi_{\text{km}} = V^*_{\text{bet}}/V_{\text{total}}$, where V^*_{bet} is the k-means between-cluster variance; it requires no labels.

Proposition 3 (pCDNV and the between-cluster share)**.** $\mathit{pCDNV} = \frac{1}{2\phi_{\text{km}}} - \frac{1}{2}$ *is strictly decreasing in ϕ_{km}, so it is maximized exactly when ϕ_{km} is minimized. If $V_{\text{total}} > 0$ is constant across budgets, this maximizer also coincides exactly with the budget at which V^*_{bet} is minimized.*

In the approximately conserved regime observed in our experiments, the pCDNV peak and the maximum of V_{irr} are typically close; we verify this empirically below. In this regime $V_{\text{total}} \approx 1$, and Proposition 3 explains why pCDNV detects a turnaround whereas label-based CDNV does not. Label-based CDNV uses the true between-class variance V_{bet}, which grows monotonically. With approximately constant total variance, CDNV therefore decreases monotonically and exhibits no turnaround.

By contrast, V^*_{bet} is non-monotonic: it reaches a minimum as within-class scatter collapses, then rises as true class structure emerges. At low budgets, when class structure is immature, k-means exploits large within-class semantic scatter to form spurious but well-separated clusters, so V^*_{bet} exceeds V_{bet} (by up to a factor of five on CIFAR-100). Empirically, this minimum and the pCDNV peak coincide with the V_{irr} peak; the pCDNV and V_{irr} trajectories also track small local deviations across all samplers tested (Figure 2B; Appendix E, Figure 8). Thus, pCDNV is a label-free indicator of the maturation event marked by its turnaround, not a measure of irrelevant variance.

Finally, we note that if the initial representation is already sufficiently mature, pCDNV may be monotonic or peak at a very low labeling budget, leaving no later peak to guide the method. We observe this behavior on CIFAR-10 (Figure 7A in Appendix D); in this case, our method treats the origin as the effective peak. This is also the setting in which the automatic evaluation of γ described in Appendix I fails.

6 Empirical Results

6.1 Setup

Datasets. As customary in low-budget active learning, we evaluate our method on CIFAR-10 [20], CIFAR-100 [20], TinyImageNet [21], and ImageNet-100, a 100-class subset of ImageNet [7], following the splits described in [34].

Architecture and training. We trained ResNet-18 [17] from scratch at each active learning episode with SGD, Nesterov momentum 0.9, weight decay 3×10^{-4}, cosine learning rate schedule (base rate 0.025), for 200 epochs with a batch size of 100. Random cropping and horizontal flips were used for data augmentation.

Static features. All coverage-based strategies require a feature space that remains fixed across active learning episodes. For CIFAR-10, CIFAR-100, and ImageNet-100 we use SimCLR [5] representations pretrained on the respective training sets, and for TinyImageNet we use SCAN [34] pretext features. In both cases the representations are ℓ_2-normalized 512-dimensional vectors computed once before the active learning loop begins and kept frozen throughout.

Baselines. We compare against Random selection, *ProbCover* [37] (coverage radius: Appendix N), *MaxHerding* [3], Margin, *DCoM* [24], and *UHerding* [4].

6.2 Main Results

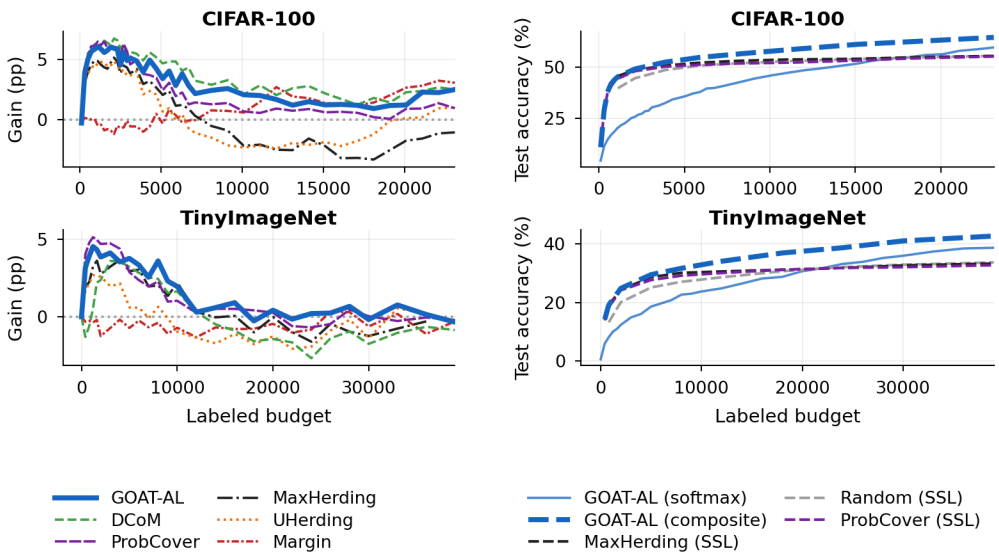
GOAT-AL is competitive with or better than every baseline across the budget range (Figure 3a):

- **Low budgets:** *GOAT-AL* is on par with the strongest coverage baselines, *ProbCover* and *MaxHerding*, confirming that the static kernel alone ($\alpha \approx 1$) is appropriate when learned features are unreliable.
- **High budgets:** while never switching objectives, *GOAT-AL* keeps pace with or exceeds the uncertainty-based and hybrid baselines (Margin, *DCoM*, *UHerding*).

The composite kernel yields better classification than the softmax head or SSL alone. For the same selection method and labeled set, the hyperparameter-free classification blend described in Section 4.5 outperforms both the trained softmax head and k-nearest-neighbor classification using only the frozen SSL kernel. Figure 3b isolates the classifiers, with *GOAT-AL*'s softmax head repeated as a thin line for reference. The margin over the SSL-only k-NN grows as the composite kernel increasingly draws on the trained representation once it matures.

6.3 Ablation Studies

We ablate each component of *GOAT-AL*; the full figures are provided in the appendix. **(i) Kernel components.** Static-only (α=1) saturates at high budgets, and adaptive-only (α=0) fails at low budgets where penultimate features are unreliable; *GOAT-AL*'s *pCDNV*-guided α avoids both failure modes (Figure 10). **(ii) Robustness.** *GOAT-AL*'s *pCDNV* trajectory and accuracy are stable across the clustering algorithm (k-means, spectral, GMM; Appendix G, Figure 11), and the self-supervised



(a) AL methods, softmax head: gain over Random.

(b) k-NN classifiers ($k = 10$).

Figure 3: Results on CIFAR-100 and TinyImageNet (for additional datasets see Appendix M). **(a)** Test-accuracy gain over Random (percentage points) of the AL methods with the softmax head; dotted line: Random. **(b)** k-NN classifiers ($k = 10$): *GOAT-AL*'s composite kernel with the hyperparameter-free blend vs. *MaxHerding*, Random, and *ProbCover* on the frozen SSL kernel.

encoder (SimCLR, BYOL [14], MoCo-v2 [6], Barlow Twins [38]; Appendix F, Figure 9); it is similarly insensitive to the schedule sharpness γ (Section 4.3, Appendix H). **(iii) Automatic γ.** γ can itself be derived from *pCDNV*'s own trajectory rather than tuned, recovering the hand-tuned value on CIFAR-100 at no accuracy cost; the derivation degenerates on CIFAR-10, where *pCDNV*'s rise is too small to anchor a scale, at a small residual cost (Appendix I).

7 Discussion and Limitations

Budget-wise neural collapse. Prior work on Neural Collapse tracks representation geometry over training epochs [27, 12]; we observe an analogous reallocation of variance along the *budget axis*. As labeled data accumulates, irrelevant variance drains into class separation, and *pCDNV* reads this event off without labels. This repositions Neural Collapse as a diagnostic of *data sufficiency*, not only of training convergence: the same geometric maturation that ends training, when evaluated along the budget axis, also marks when the trained representation, and therefore uncertainty-based selection, becomes useful.

Why coverage alone can suffice. Section 4 shows, under Neural Collapse, why covering the trained, discriminative representation tends to target the same near-boundary points that uncertainty sampling selects (cf. [31]): a point far from all labeled centroids has a small classifier-margin bound under idealized Neural Collapse, and nearest-centroid distance is strongly negatively correlated with margin on the unlabeled pool.

Limitations. *GOAT-AL* builds on a frozen self-supervised representation, so its low-budget performance depends on the quality of that representation. When self-supervised features already perform well, as on CIFAR-10, there is little room for the adaptive mechanism to help. More generally, *pCDNV* can locate the transition only when the irrelevant-variance turnaround is discernible within the available budget. *GOAT-AL* also requires the number of classes K and a self-supervised encoder for the target domain. Our evaluation uses a cold-start protocol, retraining a single ResNet-18 from scratch at each episode, rather than fine-tuning. All benchmarks used here are class-balanced, whereas Neural-Collapse-based arguments typically assume balance; behavior under class imbalance is untested. Finally, *pCDNV* requires clustering, though its trajectory is stable across k-means, spectral, and GMM clustering (Figure 11).

Acknowledgments

This work was supported by a grant from the Gatsby Charitable Foundation and AFOSR award FA8655-24-1-7006. The authors declare no competing interests.

References

[1] Alessio Ansuini, Alessandro Laio, Jakob H Macke, and Davide Zoccolan. Intrinsic dimension of data representations in deep neural networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

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

[3] Wonho Bae, Junhyug Noh, and Danica J Sutherland. Generalized coverage for more robust low-budget active learning. In *European conference on computer vision*, pages 318–334. Springer, 2024.

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

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PmLR, 2020.

[6] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[8] Pinar Donmez and Jaime G. Carbonell. From active to proactive learning methods. In *Advances in Machine Learning I*, volume 262 of *Studies in Computational Intelligence*, pages 97–120. Springer Berlin Heidelberg, 2010. doi: 10.1007/978-3-642-05177-7_5.

[9] Pinar Donmez, Jaime G Carbonell, and Paul N Bennett. Dual strategy active learning. In *Machine Learning: ECML 2007*, pages 116–127. Springer, 2007.

[10] Paul Doucet, Benjamin Estermann, Till Aczel, and Roger Wattenhofer. Bridging diversity and uncertainty in active learning with self-supervised pre-training. *arXiv*, 2024. doi: 10.48550/arxiv.2403.03728.

[11] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In *International conference on machine learning*, pages 1183–1192. PMLR, 2017.

[12] Tomer Galanti, András György, and Marcus Hutter. On the role of neural collapse in transfer learning. In *International Conference on Learning Representations*, 2022.

[13] Atharv Goel, Sharat Agarwal, Saket Anand, and Chetan Arora. Reliable active learning from unreliable labels via neural collapse geometry. *arXiv preprint arXiv:2510.09740*, 2025.

[14] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

[15] Guy Hacohen and Daphna Weinshall. How to select which active learning strategy is best suited for your specific problem and budget. *Advances in Neural Information Processing Systems*, 36:13395–13407, 2023.

[16] Guy Hacohen, Avihu Dekel, and Daphna Weinshall. Active learning on a budget: Opposite strategies suit high and low budgets. In *International Conference on Machine Learning*, pages 8175–8195. PMLR, 2022.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[18] Ajay J Joshi, Fatih Porikli, and Nikolaos Papanikolopoulos. Multi-class active learning for image classification. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2372–2379. IEEE, 2009.

[19] Andreas Kirsch, Joost Van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. *Advances in neural information processing systems*, 32:7026–7037, 2019.

[20] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[21] Ya Le and Xuan Yang. Tiny ImageNet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

[22] David D Lewis and William A Gale. A sequential algorithm for training text classifiers. In *SIGIR'94*, pages 3–12. Springer, 1994.

[23] Julia Machnio, Mads Nielsen, and Mostafa Mehdipour Ghazi. A mechanism-driven theory of phase transitions in active learning. In *European Conference on Computer Vision*, pages 342–358. Springer, 2026.

[24] Inbal Mishal and Daphna Weinshall. Dcom: Active learning for all learners. *arXiv preprint arXiv:2407.01804*, 2024.

[25] Liam Mohr. Federated active learning with few labels. Master's thesis, The Hebrew University of Jerusalem, August 2026. URL https://www.cs.huji.ac.il/~daphna/theses/Liam_Mohr_2026.pdf. Chapter 4, ProbCover Delta Selection, pp. 28–34. Supervised by Daphna Weinshall.

[26] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

[27] Vardan Papyan, XY Han, and David L Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. *Proceedings of the National Academy of Sciences*, 117(40):24652–24663, 2020.

[28] Akshay Rangamani, Marius Lindegaard, Tomer Galanti, and Tomaso A Poggio. Feature learning in deep classifiers through intermediate neural collapse. In *International Conference on Machine Learning*, pages 28729–28745. PMLR, 2023.

[29] Hiranmayi Ranganathan, Hemanth Venkateswara, Shayok Chakraborty, and Sethuraman Panchanathan. Deep active learning for image classification. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3934–3938. IEEE, 2017.

[30] Tobias Scheffer, Christian Decomain, and Stefan Wrobel. Active hidden markov models for information extraction. In *International Symposium on Intelligent Data Analysis*, pages 309–318. Springer, 2001.

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

[32] Changjian Shui, Fan Zhou, Christian Gagné, and Boyu Wang. Deep active learning: Unified and principled method for query and training. In *International Conference on Artificial Intelligence and Statistics*, pages 1308–1318. PMLR, 2020.

[33] Samarth Sinha, Sayna Ebrahimi, and Trevor Darrell. Variational adversarial active learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5972–5981, 2019.

[34] Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Scan: Learning to classify images without labels. In *European conference on computer vision*, pages 268–285. Springer, 2020.

[35] Jae Oh Woo. Active learning in bayesian neural networks with balanced entropy learning principle. In *International Conference on Learning Representations*, 2023.

[36] Yi Yang, Zhigang Ma, Feiping Nie, Xiaojun Chang, and Alexander G Hauptmann. Multi-class active learning by uncertainty sampling with diversity maximization. *International Journal of Computer Vision*, 113(2):113–127, 2015.

[37] Ofer Yehuda, Avihu Dekel, Guy Hacohen, and Daphna Weinshall. Active learning through a covering lens. *Advances in Neural Information Processing Systems*, 35:22354–22367, 2022.

[38] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021.

A Proofs for the Irrelevant-Variance Account

Throughout, $h(x) \in \mathbb{R}^d$ are penultimate features, μ_c the mean of class c, μ_T the total mean, $N_c = |\mathcal{X}_c|$, and $S = \mathrm{span}\{\mu_c - \mu_T\}$ with projectors $P_S, P_\perp$. We write $W = V_{\mathrm{sem}} + V_{\mathrm{irr}}$ for the total within-class variance.

A.1 Proof of Proposition 1 (Variance budget)

Total variance splits into between- and within-class parts, $V_{\mathrm{total}} = V_{\mathrm{bet}} + W$ with $W = \frac{1}{N}\sum_i \|h(x_i) - \mu_{y_i}\|^2$. Applying $I = P_S + P_\perp$ to each within-class residual $h(x_i) - \mu_{y_i}$ and using $\|v\|^2 = \|P_S v\|^2 + \|P_\perp v\|^2$ gives $W = V_{\mathrm{sem}} + V_{\mathrm{irr}}$, hence $V_{\mathrm{total}} = V_{\mathrm{bet}} + V_{\mathrm{sem}} + V_{\mathrm{irr}}$. □

A.2 Proof of Proposition 2 and Corollary 1 (Variance identity and conserved budget)

For $\|h(x)\| = 1$, $\|h(x) - \mu_T\|^2 = 1 - 2\,h(x)^\top \mu_T + \|\mu_T\|^2$. Averaging over x and using $\mathbb{E}[h(x)] = \mu_T$ gives $V_{\mathrm{total}} = \mathbb{E}\|h(x) - \mu_T\|^2 = 1 - 2\|\mu_T\|^2 + \|\mu_T\|^2 = 1 - \|\mu_T\|^2 \le 1$. If $\|\mu_T\|^2 = q$ is constant across budgets, Proposition 1 gives $V_{\mathrm{bet}} + V_{\mathrm{sem}} + V_{\mathrm{irr}} = 1 - q$ exactly; if $\mu_T = 0$, the sum is 1. Thus any increase in V_{bet} must come from W. □

A.3 Proof of Proposition 3 (pCDNV and the between-cluster share)

The aggregate CDNV is $W/(2V_{\mathrm{bet}}) = (V_{\mathrm{total}} - V_{\mathrm{bet}})/(2V_{\mathrm{bet}}) = \frac{1}{2\phi} - \frac{1}{2}$ with $\phi = V_{\mathrm{bet}}/V_{\mathrm{total}}$. pCDNV is the same expression with $\phi_{\mathrm{km}} = V^*_{\mathrm{bet}}/V_{\mathrm{total}}$. The map $\phi \mapsto \frac{1}{2\phi} - \frac{1}{2}$ is strictly decreasing on $(0, 1]$, so pCDNV is maximized exactly when ϕ_{km} is minimized. If $V_{\mathrm{total}} > 0$ is constant across budgets, minimizing ϕ_{km} is equivalent to minimizing V^*_{bet}, so the extrema coincide exactly. □

A.4 Agreement of the aggregate and pairwise CDNV

Proposition 4 (Agreement up to $K/(K-1)$)**.** *For balanced classes ($N_c = N/K$) with equal within-class variances σ^2 and unit-norm, equiangular means, the pairwise CDNV* $\mathrm{avg}_{c \neq c'} \frac{\sigma_c^2 + \sigma_{c'}^2}{2\|\mu_c - \mu_{c'}\|^2}$ *of Galanti et al. [12] equals $\frac{K-1}{K}$ times the aggregate CDNV $W/(2V_{\mathrm{bet}})$.*

Proof. Let $d^2 = \|\mu_c - \mu_T\|^2$, equal across c by symmetry, so $V_{\mathrm{bet}} = \frac{1}{K}\sum_c d^2 = d^2$ and $W = \sigma^2$. Since $\sum_c (\mu_c - \mu_T) = 0$, we have $\sum_{c \neq c'} \langle \mu_c - \mu_T, \mu_{c'} - \mu_T \rangle = -Kd^2$; by equiangularity each inner product equals $-d^2/(K-1)$, so $\|\mu_c - \mu_{c'}\|^2 = 2d^2 - 2 \cdot (-d^2/(K-1)) = 2d^2 \frac{K}{K-1}$, constant over pairs. The pairwise CDNV is then $\frac{2\sigma^2}{2 \cdot 2d^2 K/(K-1)} = \frac{\sigma^2}{2d^2} \cdot \frac{K-1}{K}$, while the aggregate CDNV is $W/(2V_{\mathrm{bet}}) = \sigma^2/(2d^2)$. The ratio is $\frac{K-1}{K}$. □

Under the same assumptions, Eq. (2) also equals the aggregate form: its denominator is $\frac{1}{K}\sum_{c'} \|\mu_c - \mu_{c'}\|^2 = \frac{1}{K}(K-1) \cdot 2d^2 \frac{K}{K-1} = 2d^2$ and its numerator is σ^2, so $\mathrm{CDNV} = \sigma^2/(2d^2) = W/(2V_{\mathrm{bet}})$. We adopt the aggregate form because it yields the exact identity with the between-class share ϕ that Proposition 3 relies on.

B The Mechanism Across Datasets

Figure 2 shows CIFAR-100; Figure 4 reports the same construction, uncertainty (Margin) minus coverage (MaxHerding), for accuracy and irrelevant variance, on all four datasets, with the correlation between the two gaps labeled per panel: $r = -0.84$ (CIFAR-10), $r = -0.97$ (CIFAR-100), $r = -0.85$ (TinyImageNet), and $r = -0.90$ (ImageNet-100).

C Causal Experiment: Marginality Drives Irrelevant Variance

The correlations in Figure 2 establish that excess irrelevant variance accompanies the accuracy loss under uncertainty sampling, but not the direction of causation. We isolate it with a controlled dose–response. Holding the labeling budget, class balance, and architecture fixed, we vary only the *marginality* of the labeled set: we construct labeled sets ranging from typical (interior) to marginal (near class boundaries) at matched difficulty, train from scratch on each, and measure V_{irr} and test accuracy. As marginality increases, V_{irr} rises and accuracy falls monotonically, with Pearson correlation $r = +0.98$ on CIFAR-100 (and $r = +0.91$ on CIFAR-10). Because marginality is the only manipulated variable, the labeled set's marginality causally increases irrelevant variance and depresses accuracy, the mechanism the main-text correlations reflect.

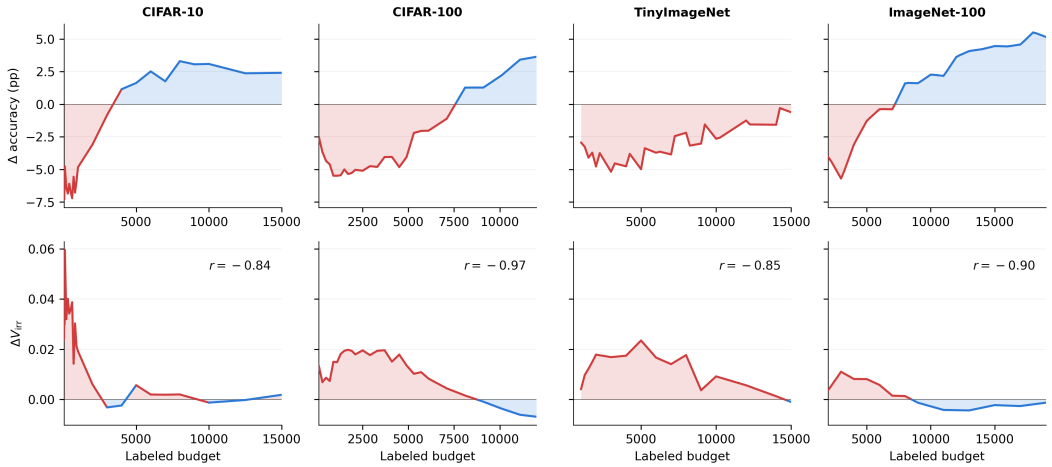


Figure 4: **The accuracy/irrelevant-variance mechanism holds across all four datasets.** Each column is one dataset; each curve is Margin (uncertainty) minus MaxHerding (coverage). *Top:* test-accuracy difference (pp). *Bottom:* irrelevant-variance difference. In every dataset, uncertainty carries excess V_{irr} for as long as it sacrifices accuracy, and the two gaps close together.

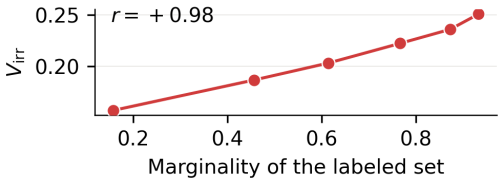


Figure 5: **Marginality causally drives irrelevant variance.** V_{irr} as the labeled set's marginality is swept by construction across six levels, with budget, class balance, and architecture held fixed (CIFAR-100). The monotonic rise ($r = +0.98$) is the dose–response underlying the correlations of Figure 2.

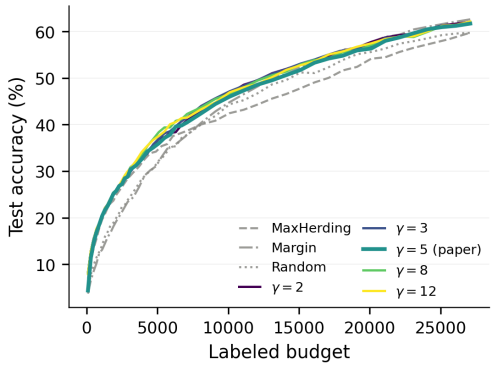


Figure 6: Sensitivity to γ on CIFAR-100, against the MaxHerding, Margin, and Random baselines. The five schedules track closely across the full budget range and stay clearly above all three baselines; $\gamma = 5$ (deployed) is a safe mid-range choice.

D The Conserved Variance Economy

Figure 2C previews this on CIFAR-100 alone; Figure 7 here shows the three components of Proposition 1 along the labeling budget, on all four datasets. Normalization keeps V_{total} essentially constant, so the decomposition is zero-sum. Throughout, V_{sem} is the reservoir: during the over-expressive regime it drains into both V_{bet} and V_{irr} at once, which is how V_{irr} can grow even as V_{bet} also grows; after the turnaround, V_{irr} drains as well, so both it and the continuing contraction of V_{sem} feed V_{bet}. The pCDNV peak coincides with the V_{irr} maximum.

E pCDNV Tracks Irrelevant Variance in Detail

pCDNV follows the same rise–peak–decline as V_{irr} under coverage, uncertainty, and random sampling (Figure 2B previews the random-sampling case; Figure 8 here gives all three). After subtracting a smooth (degree-4 polynomial) trend from both, the residuals still coincide ($r \geq 0.99$), so the agreement is not an artifact of the two

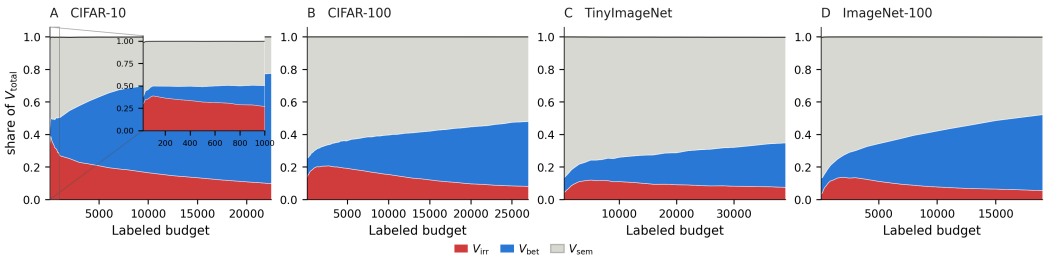


Figure 7: **A conserved variance budget: irrelevant variance swells, then drains into class separation.** Random-selection trajectories on all four datasets (mean over 3 seeds; ImageNet-100 over 2), so the conservation law is shown independent of any query strategy. V_{sem} (gray) is the reservoir throughout: it drains into both V_{irr} (red) and V_{bet} (blue) before the turnaround, then keeps draining into V_{bet} alongside V_{irr} afterward. The three components sum to a near-constant V_{total}. CIFAR-10's turnaround is compressed into the first few hundred labels, consistent with it being the SSL-saturated boundary case discussed in the Limitations.

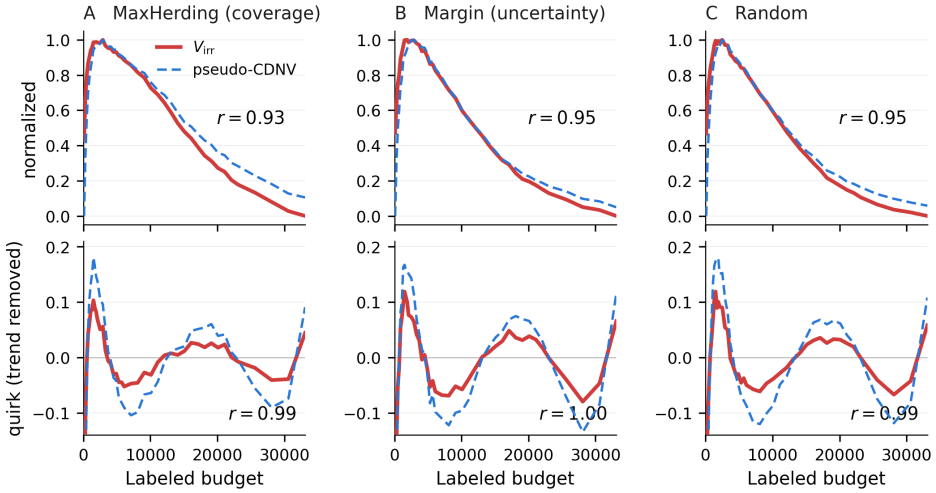


Figure 8: **pCDNV tracks irrelevant variance, whatever the sampler.** CIFAR-100, 3 seeds; each series min–max scaled to $[0, 1]$. *Top:* pCDNV and V_{irr} coincide under coverage, uncertainty, and random sampling. *Bottom:* after subtracting a smooth (degree-4 polynomial) trend from each series, the residuals still agree ($r \geq 0.99$), so the match is not an artifact of the two curves sharing an overall shape.

curves sharing an overall shape: pCDNV reproduces even the small local deviations of the irrelevant-variance trajectory, confirming it as a label-free reading of the same event.

F Robustness to the Self-Supervised Encoder

GOAT-AL's static kernel is built from a frozen self-supervised representation. To test sensitivity to that choice, we re-run the full pipeline with the static features replaced by four different self-supervised encoders (SimCLR, the paper default, plus BYOL [14], MoCo-v2 [6], and Barlow Twins [38]), holding everything else fixed. Accuracy is essentially unchanged on CIFAR-100: averaged over budgets, the four encoders differ by at most ≈ 0.8 pp (Figure 9), and the low-, mid-, and high-budget behavior is preserved. GOAT-AL therefore does not depend on a particular self-supervised representation, only on having suitable self-supervised features.

G Ablation Figures

Figures for the ablations summarized in Section 6.3.

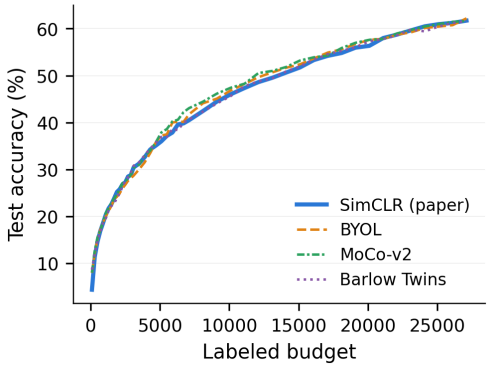


Figure 9: **Robustness to the self-supervised encoder** (CIFAR-100). Replacing the static SSL features (SimCLR, BYOL, MoCo-v2, Barlow Twins) leaves accuracy essentially unchanged across the full budget range.

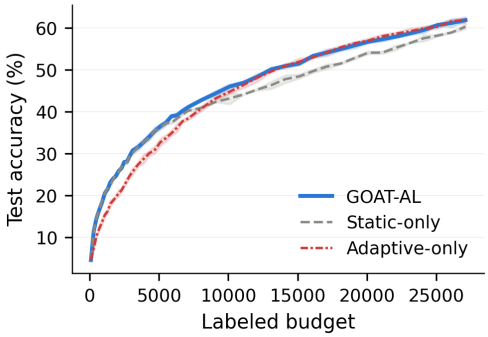


Figure 10: Ablation of kernel components. Static-only (α=1) degrades at high budgets, while adaptive-only (α=0) fails at low budgets due to unreliable features. GOAT-AL, guided by pCDNV, transitions between them and avoids both failure modes across regimes.

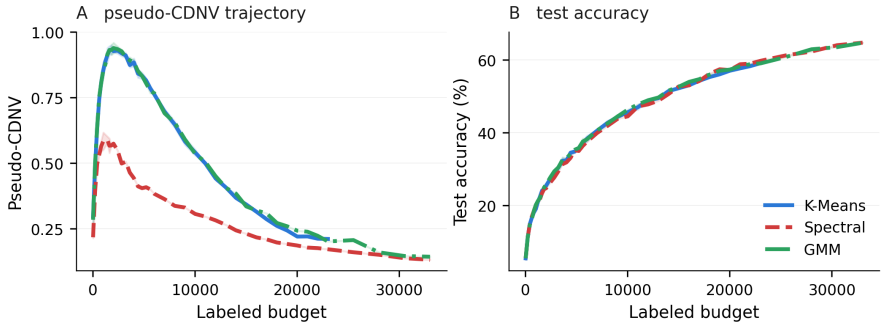


Figure 11: Robustness to the clustering algorithm. pCDNV trajectory (left) and test accuracy (right) on CIFAR-100 for k-means, spectral clustering, and GMM pseudo-labeling, averaged over 3 seeds. All three yield similar trajectories and comparable peak locations, as a result of which selection performance is stable across them.

H Robustness to the Schedule Sharpness γ

The exponent γ in $\alpha_e = (P_e/P^*)^\gamma$ controls only the *rate* of the transition, not its location (the pCDNV peak). Sweeping $\gamma \in \{2, 3, 5, 8, 12\}$ changes mean accuracy over budgets by at most 0.7 pp (Figure 6); the deployed $\gamma = 5$ sits mid-range. The schedules remain tightly grouped in the transition regime, where γ acts.

I Automatic Derivation of the Schedule Sharpness γ

Appendix H shows accuracy is insensitive to γ over a wide range, so it was tuned once (to 5) and fixed. We test whether γ itself can instead be derived from the same label-free signal that already locates the transition. Let P_0 and P^* be pCDNV's episode-0 value and its running peak (both already computed for α_e, Eq. 6), and set $r_0 = P_0/P^*$. For $0 < r_0 < 1$, defining

$$\gamma_{\text{auto}} = \frac{\ln \epsilon}{\ln r_0}, \qquad \epsilon = 0.01,$$

makes α_e reach ϵ exactly when pCDNV has retraced, in log space, the rise it accumulated during the over-expressive regime – a data-driven analogue of the fixed exponent, with no free parameter beyond the "what-counts-as-zero" convention ϵ.

CIFAR-100: recovers the tuned value. Closed-loop against a same-code $\gamma = 5$ control, γ_{auto} converges to the tuned value (Table 1), and the resulting schedule and accuracy are indistinguishable from the control (Figure 12A–B).

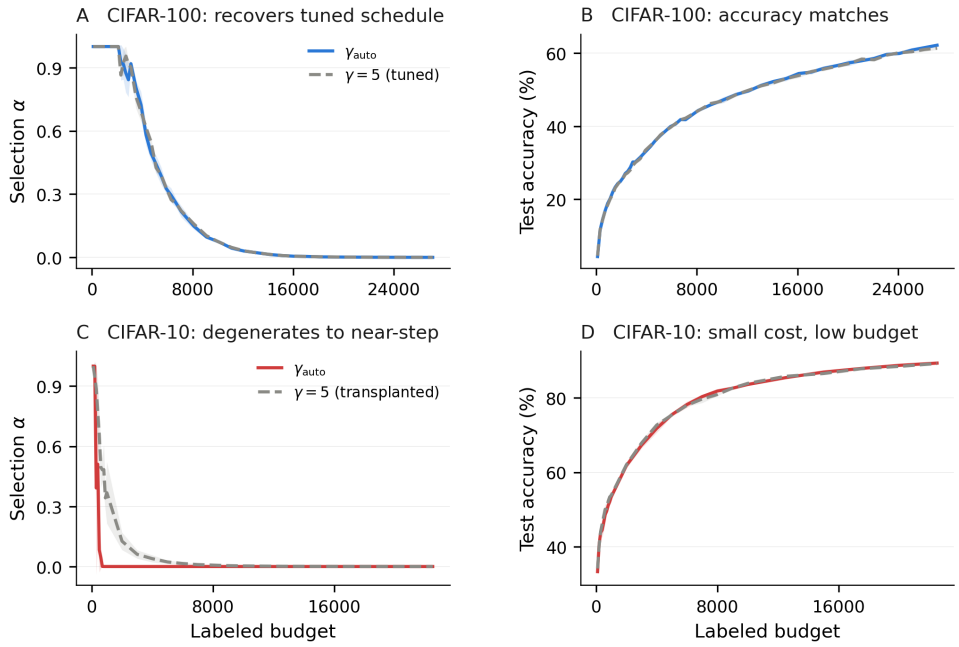


Figure 12: **Automatic derivation of γ: CIFAR-100 recovers the tuned value; CIFAR-10 is a known failure case.** *A, B:* CIFAR-100, closed-loop, mean $\pm$ std over 3 seeds; γ_{auto} (blue) against a same-code $\gamma = 5$ control (gray dashed) is indistinguishable in both schedule (A) and accuracy (B). *C, D:* CIFAR-10, against a $\gamma = 5$ control transplanted from CIFAR-100; γ_{auto} (red) collapses to a near-step schedule (C), producing a small but consistent accuracy cost concentrated at low budgets that closes by the final budget (D).

CIFAR-10: a known failure case. On CIFAR-10, r_0 is not a stable dataset property but a protocol artifact: with 10 classes and already-strong SSL features, pCDNV's rise is small and brief, so wherever episode 0 happens to fall relative to the collapse onset largely determines r_0. Against a $\gamma = 5$ control transplanted from CIFAR-100, γ_{auto} comes out unstable and degenerate across seeds (Table 1), collapsing α to a near-step function rather than a gradual decay (Figure 12C). Restricted to the budgets where the control's α is still active ($\alpha > 0.01$; averaged per seed, then tested across seeds), this costs a small but consistent accuracy gap ($t = -4.30$, $p = 0.050$, $n = 3$) that does not survive to the final budget (Figure 12D). A floor on r_0 makes γ_{auto} deterministic (≈ 20.6) but does not recover the accuracy, so the shortfall is that pCDNV carries too little dynamic range on this dataset for r_0 to be a meaningful scale, not that the derivation is merely noisy.

TinyImageNet and ImageNet-100: parity, in either direction. On the remaining two datasets, against the same transplanted $\gamma = 5$ control, γ_{auto} is stable across seeds (unlike CIFAR-10) and the accuracy delta is small and mixed in sign – effectively parity rather than a consistent cost or gain (Table 1).

Dataset	γ_{auto} (range, 3 seeds)	mean Δacc (auto $-$ tuned)
CIFAR-100	5.06 (4.75–5.38)	$+0.13$ pp
TinyImageNet	2.42 (2.09–3.00)	-0.08 pp
ImageNet-100	3.17 (3.13–3.22)	$+0.39$ pp
CIFAR-10	89 (41–155)	-0.66 pp (active regime)

Table 1: γ_{auto} against a transplanted-or-tuned $\gamma = 5$ control, all four datasets. CIFAR-100 is closed-loop (tuned on that dataset); the other three use $\gamma = 5$ transplanted from CIFAR-100. CIFAR-10 is the only dataset where γ_{auto} is both unstable across seeds and accuracy-costly.

Takeaway. γ_{auto} removes the one remaining tuned constant in GOAT-AL at no measurable cost on three of four datasets, and is a promising but not fully solved direction rather than a deployed default: CIFAR-10 shows the derivation can degenerate when pCDNV's own rise-then-fall has too little range to anchor a scale, so the main results throughout the paper use the fixed, ablated $\gamma = 5$.

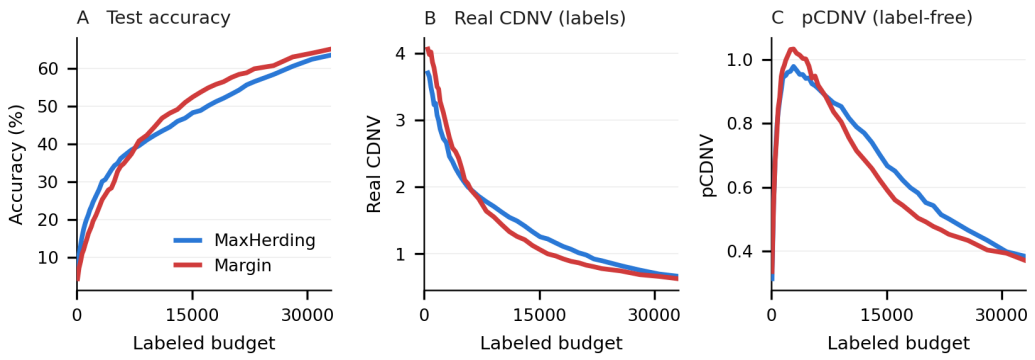


Figure 13: Real CDNV vs. pCDNV (Figure 2D previews this under random sampling). Left: test accuracy. Middle: real CDNV (true labels). Right: pCDNV (k-means pseudo-labels) for MaxHerding (low-budget) and Margin (high-budget) on CIFAR-100. Real CDNV decreases monotonically, while pCDNV follows a rise, peak, decline pattern; beyond the peak pCDNV mirrors real CDNV and is inversely correlated with accuracy. The peak marks the budget transition (Section 5).

J Calibrating the Classification Blend

The composite kernel k_{G} (Eq. 3) is used for classification with its own blend weight, distinct from the selection schedule α_e (Eq. 6). The two cannot share a target: selection's coverage objective (Eq. 4) aggregates k_{G} over the entire pool $\mathcal{U}$, whereas k-nearest-neighbor classification takes into account only the top-k neighbors of each query. Restricting to the top-k neighbors changes the static kernel's share of a query's realized neighborhood nonlinearly as α varies – calibrating against the whole-kernel mixing weight, as selection does, measures the wrong quantity for classification. We calibrate instead directly to the realized vote composition: for a given α, let $\mathcal{N}_k(x;\alpha) \subseteq \mathcal{L}$ denote the query's k nearest neighbors under k_{G}. The static share of their similarity mass is

$$s_x(\alpha) = \frac{\sum_{x' \in \mathcal{N}_k(x;\alpha)} \alpha\, k^{(s)}_{\sigma_s}(x, x'; g_s)/\bar{k}^{(s)}}{\sum_{x' \in \mathcal{N}_k(x;\alpha)} k_{\mathrm{G}}(x, x')}, \tag{8}$$

with $k = 10$. Both the neighbors and their similarity weights depend on α. Let $S_x(\alpha)$ and $T_x(\alpha)$ denote the numerator and denominator of $s_x(\alpha)$, so $s_x = S_x/T_x$. The calibration curve used to choose the classification blend is the *pooled* (mass-weighted) share over a set of query inputs $\mathcal{Q}$ (no labels are used):

$$\mathrm{share}(\alpha) = \frac{\sum_{x \in \mathcal{Q}} S_x(\alpha)}{\sum_{x \in \mathcal{Q}} T_x(\alpha)}. \tag{9}$$

Pooling weights each query by its total similarity mass T_x, so it equals the plain average of the s_x only when all T_x coincide.

The target share is label-free, given directly by the labeled budget $|\mathcal{L}|$ and the total pool size $N_0 = |\mathcal{L}| + |\mathcal{U}|$ (fixed over the run, since $\mathcal{U}$ here denotes the *currently unlabeled* remainder): $s = 1 - |\mathcal{L}|/N_0$. The SSL contribution remains positive for $|\mathcal{L}| < N_0$. Its endpoints follow from this ratio alone: $|\mathcal{L}|/N_0 \to 0$ gives $s \to 1$ (pure static kernel, matching the empirically measured optimum below $|\mathcal{L}| \approx 1000$ on every dataset), and $|\mathcal{L}|/N_0 \to 1$ gives $s \to 0$ (pure adaptive kernel). The classification blend weight is then the α whose *measured* $\mathrm{share}(\alpha)$ equals s, read directly off the logged calibration curve.

This is a different question from the one α_e answers: α_e detects *when* the representation has matured enough to switch (a transition, located by the pCDNV peak), while the classification blend sets *how much* weight a not-yet-fully-trained representation should get in a k-nearest-neighbor vote (a smooth rate). The two blends are calibrated to different targets by construction, and are not expected to coincide.

K Bandwidth Selection

Each kernel component requires a bandwidth parameter σ. We use a data-driven approach based on curvature analysis of the purity-versus-δ curve, following Yehuda et al. [37]. Specifically, we compute purity estimates via k-means pseudo-labels and find the knee point of maximum curvature in the purity curve:

$$\sigma^* = \arg\min_{\sigma} \kappa(\sigma), \quad \text{where} \quad \kappa(\sigma) = \frac{\pi''(\sigma)}{(1 + \pi'(\sigma)^2)^{3/2}} \tag{10}$$

and $\pi(\sigma)$ is the fraction of points whose nearest point from a different pseudo-cluster lies at distance $\geq \sigma$ – equivalently, the fraction of points whose σ-ball contains no point from another pseudo-cluster. For the static kernel, this is computed once and cached. For the adaptive kernel, it is recomputed at each episode using the current penultimate features. We note that MaxHerding [3] claims robustness to σ and fixes $\sigma = 1$ throughout. In our experiments, using the knee-point σ^* consistently outperforms $\sigma = 1$ on the static kernel, suggesting the robustness claim does not hold uniformly across datasets and budgets.

L Centroid Distance and Classifier Margin

Neural Collapse bound

Let $h(x)$ be the unit-norm penultimate feature, $z_c(x) = w_c^\top h(x)$ the class-c logit (with no bias terms), and $z_{(1)}(x) - z_{(2)}(x)$ the classifier margin. Under Neural Collapse [27], the class means are centered and equidistant from the origin ($\sum_c \mu_c^{\mathcal{L}} = 0$, $\|\mu_c^{\mathcal{L}}\| = m$) and, by NC3 (self-duality: classifier weights align with class means), $w_c = \kappa\, \mu_c^{\mathcal{L}}$ for some $\kappa > 0$; since *GOAT-AL* trains on the labeled set $\mathcal{L}$, these are the labeled-set centroids. Then $\sum_c z_c(x) = 0$, so $z_{(2)}(x) \geq -z_{(1)}(x)/(K-1)$, while $z_{(1)}(x) = \frac{\kappa}{2}\big(1 + m^2 - d_{\min}(x)^2\big)$ with $d_{\min}(x) = \min_c \|h(x) - \mu_c^{\mathcal{L}}\|$. Hence

$$z_{(1)}(x) - z_{(2)}(x) \;\leq\; \frac{\kappa K}{2(K-1)}\big(1 + m^2 - d_{\min}(x)^2\big),$$

so points far from all labeled centroids have small margins. The converse need not hold, so nearest-centroid distance is a one-directional proxy for uncertainty – the direction the selection rule needs.

Empirical distance–margin correlation

Table 2 supports the Neural Collapse argument of Section 4. For each labeled budget on CIFAR-100, we take the labeled set from a random-sampling run, train a fresh ResNet-18 on exactly that set, and extract penultimate features and softmax outputs on the unlabeled pool. Class centroids are the means of the labeled points' penultimate features under their true labels. For each unlabeled point we compute its distance to the nearest centroid and its classifier margin (the gap between the two largest softmax probabilities), and report their Spearman correlation ρ, averaged over 3 seeds. The correlation is strongly negative at every budget and grows steadily stronger as the representation collapses.

Labeled budget	ρ(centroid-dist, margin)
500	-0.527
1,400	-0.556
3,000	-0.640
6,000	-0.749
9,500	-0.828
15,000	-0.901
20,000	-0.925
25,000	-0.932

Table 2: Spearman correlation between distance to the nearest labeled class centroid and classifier margin on the CIFAR-100 unlabeled pool, by labeled budget (mean over 3 seeds).

M Main Results on CIFAR-10 and ImageNet-100

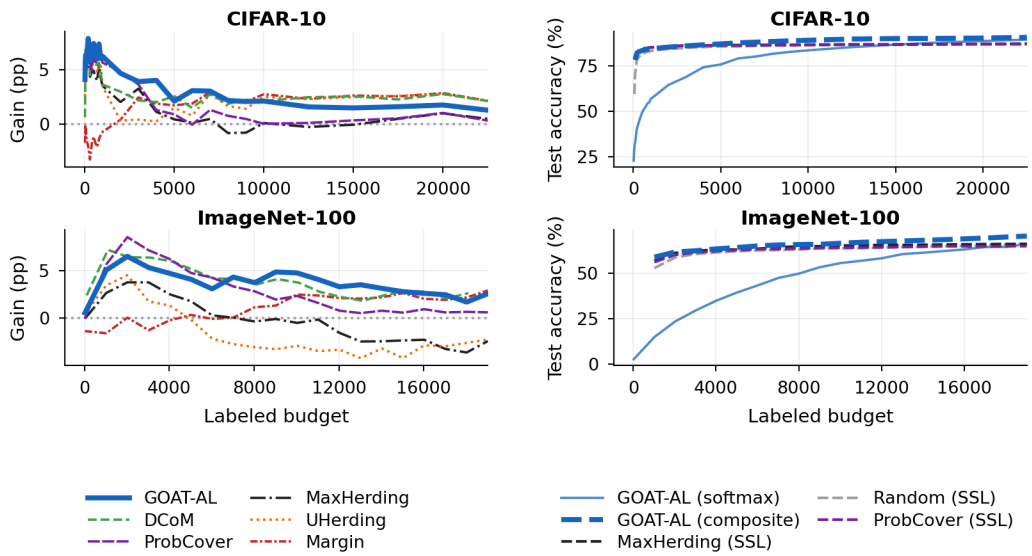
Figure 14 reports the same comparisons as Figure 3 for CIFAR-10 and ImageNet-100.

Table 3 reports the gain over Random on all four datasets, with the standard deviation across seeds.

N ProbCover Baseline: Choice of the Coverage Radius

ProbCover [37] greedily maximizes the coverage of the pool by balls of radius δ around the labeled points, in the frozen self-supervised space. Its behavior depends on δ, so we fix it with the purity-based rules of Mohr [25], computed once per dataset, with no labels, from the same SSL features that the selection uses.

Purity curve. We ℓ_2-normalize the SSL features of the training set and run k-means with K equal to the number of classes to obtain pseudo-labels. The purity at radius δ is the fraction of points whose δ-ball contains



(a) AL methods, softmax head: gain over Random

(b) k-NN classifiers ($k = 10$)

Figure 14: Same as Figure 3, for CIFAR-10 and ImageNet-100.

Table 3: Gain over Random (percentage points; mean $\pm$ sample standard deviation over 3 seeds) of the AL methods with the softmax head, averaged over the labeled-budget range. Each seed is compared with the Random run of the same seed. Bold: highest mean.

	CIFAR-10	CIFAR-100	TinyImageNet	ImageNet-100
GOAT-AL	$\mathbf{+2.48 \pm 0.36}$	$+2.41 \pm 0.23$	$\mathbf{+1.07 \pm 0.05}$	$\mathbf{+3.80 \pm 0.38}$
DCoM	$\mathbf{+2.48 \pm 0.10}$	$\mathbf{+3.01 \pm 0.42}$	-0.21 ± 0.18	$+3.78 \pm 0.31$
ProbCover	$+1.23 \pm 0.24$	$+1.63 \pm 0.21$	$+0.79 \pm 0.36$	$+3.01 \pm 0.70$
MaxHerding	$+0.74 \pm 0.09$	-0.48 ± 0.30	$+0.28 \pm 0.18$	-0.29 ± 0.55
UHerding	$+2.14 \pm 0.14$	$+0.11 \pm 0.44$	-0.39 ± 0.25	-1.59 ± 1.10
Margin	$+2.06 \pm 0.37$	$+0.88 \pm 0.93$	-0.55 ± 0.20	$+1.10 \pm 0.18$

only points of their own pseudo-cluster, $\pi(\delta) = \mathbb{P}\big(\{x : B_\delta(x) \text{ is pure}\}\big)$, equivalently the fraction of points whose nearest point of another cluster is at distance at least δ. $\pi(\delta)$ is monotonically decreasing: coverage grows with δ while purity falls. The original heuristic takes the largest δ with $\pi(\delta) \geq 0.95$. This is the same purity curve that Appendix K uses to set the kernel bandwidth of *GOAT-AL*.

Knee and inflection rules. On a grid $\{\delta_i\}$ we estimate the derivatives of π by central differences for non-uniform spacing, $\pi_i' \approx \frac{\pi_{i+1}-\pi_{i-1}}{\delta_{i+1}-\delta_{i-1}}$ and $\pi_i'' \approx \frac{2}{\delta_{i+1}-\delta_{i-1}}\left(\frac{\pi_{i+1}-\pi_i}{\delta_{i+1}-\delta_i} - \frac{\pi_i-\pi_{i-1}}{\delta_i-\delta_{i-1}}\right)$, and the signed curvature $\kappa_i = \pi_i''/(1+(\pi_i')^2)^{3/2}$. The *knee* is the point of most negative curvature, $\delta_{\text{knee}} = \delta_{i^*}$ with $i^* = \arg\min_i \kappa_i$, where purity starts to fall steeply; it needs no threshold. The *inflection* rule instead takes the point closest to a zero crossing of κ between the knee and the next positive curvature peak, $i^* = \arg\min_{i_{\text{neg}}<i<i_{\text{pos}}} |\kappa_i|$. Mohr [25] reports that the knee identifies an effective radius when a network is trained on the selected images, whereas for classifiers that operate directly on the self-supervised embedding the inflection rule is the better heuristic, although it fails in some settings.

What we use. Accordingly, the softmax-head results (Figure 3a) use the knee radius, and the k-NN classifier on the frozen SSL kernel (Figure 3b, "ProbCover (SSL)") uses the inflection radius. The knee is computed on a 0.05 grid and the inflection on a 0.01 grid:

	CIFAR-10	CIFAR-100	TinyImageNet	ImageNet-100
knee δ	0.60	0.50	0.50	0.45
inflection δ	0.79	0.75*	0.68	0.69

*On CIFAR-100 the inflection on the 0.01 grid (0.49) falls right next to the knee, so we use the value from the 0.05 grid.

Selection and evaluation. ProbCover's selection uses only the frozen features and the labeled and unlabeled sets, never the trained model, and ties in the coverage gain are broken uniformly at random. The labeled set at each budget is therefore a prefix of one greedy run per dataset and seed; every network is then trained from scratch on its labeled set, as for all methods. We report the mean over three seeds.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction accurately state both contributions (pCDNV and GOAT-AL), the experimental scope (CIFAR-10, CIFAR-100, TinyImageNet, and ImageNet-100), and the comparison baselines. See Sections 1 and 6.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Limitations section is provided.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: The paper includes four propositions (variance budget, conserved budget, the pCDNV–between-cluster-share identity, and agreement with the pairwise CDNV of Galanti et al. [12]) in Section 5, each with assumptions stated in its hypothesis and a complete proof either immediately following (main text) or in Appendix A.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: All details necessary to reproduce the main experiments are provided in Section 6: architecture, optimizer, datasets, feature extractors etc.

   Guidelines:

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

5. **Open access to data and code**

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

   Answer: [No]

   Justification: All datasets used (CIFAR-10, CIFAR-100, TinyImageNet, and ImageNet-100, a subset of ImageNet) are publicly available. We do not release the code accompanying this paper. The method, the training protocol, and the hyperparameters are described in Sections 4 and 6 and in the appendices.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Training configuration, data splits, static feature extractors, baseline methods, and budget schedule are all described in Section 6.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: Main results are averaged over 3 independent runs with the shaded region indicating standard deviation across runs.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [No]

Justification: Experiments were run on a shared compute cluster with heterogeneous hardware; we did not track per-run resource usage and cannot report consistent compute specifications.

Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: This work conforms with the NeurIPS Code of Ethics. The research involves no human subjects, sensitive data, or potential for direct misuse.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: This work proposes a foundational method for reducing annotation cost in supervised learning. It has no direct path to negative societal applications beyond those inherent to general-purpose machine learning.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

Answer: [N/A]

Justification: This paper does not release generative models, scraped datasets, or any assets with high risk for misuse.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: All datasets and pretrained feature extractors are properly cited. These are standard academic benchmarks with permissive licenses for research use.

Guidelines:

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

13. **New assets**

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

Answer: [No]

Justification: No new datasets or models are released.

Guidelines:

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

14. **Crowdsourcing and research with human subjects**

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

Answer: [N/A]

Justification: This paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

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

    Answer: [N/A]

    Justification: This paper does not involve human subjects and requires no IRB approval.

    Guidelines:

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

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were not used as a component of the proposed method. Any use for writing assistance does not affect the core methodology or scientific claims.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.