LABEL-FREE CORESET SELECTION WITH FOUNDATION MODELS FOR EFFICIENT ANNOTATION IN COMPUTATIONAL PATHOLOGY

Anonymous authors
Paper under double-blind review

ABSTRACT

Computational pathology has the potential to improve clinical outcomes through a demonstrated increase in diagnostic and prognostic accuracy. However, the development and validation of deep learning algorithms still require annotated data, a costly procedure involving expert pathologists who already face critical workforce shortages. Existing coreset selection methods to optimize annotation efforts currently all rely on hyperparameters tuned on natural-image benchmarks that do not transfer to histopathology and are cumbersome to use in clinical practice. In this study, we present GCcore, a novel label-free coreset selection method that embeds every image of a dataset with any pathology foundation model and greedily selects the samples that collectively maximize the global coverage of the embedding space. The proposed method provides a lower-bound guarantee on the global coverage of the returned coreset for any coreset size, while being completely hyperparameter-free and deterministic. We demonstrate GCcore's superior performance over 14 baselines including state-of-the-art methods across 10 tasks and datasets spanning whole slide image classification, tile classification, and tissue segmentation, where it ranks first on six and within the top three on nine, while also demonstrating how existing methods can shift by up to five rank positions depending on their hyperparameter settings.

1 INTRODUCTION

Histopathology plays a central role in clinical practice, underpinning a wide range of diagnostic, , prognostic, and treatment selection decisions, particularly in oncology. At the same time, recent advances in artificial intelligence (AI) have demonstrated considerable potential to enhance and extend histopathological analysis. Digitization of whole-slide images (WSIs) and the increasing use of computational pathology has enabled faster and more standardized (semi)automated tissue evaluation (Verghese et al., 2023). Deep learning (DL)-based methods, in particular, have achieved performance comparable to that of expert pathologists across a number of tasks, while also broadening the scope to tasks that were previously too time-consuming, such as detailed WSI tissue segmentation. More recently, large-scale pathology foundation models (FMs) pretrained on millions of image tiles have demonstrated to be robust feature extractors (Campanella et al., 2025; Bareja et al., 2025), enabling a wide range of downstream tasks while requiring smaller amounts of annotated data (Roth et al., 2024; Nicke et al., 2025).

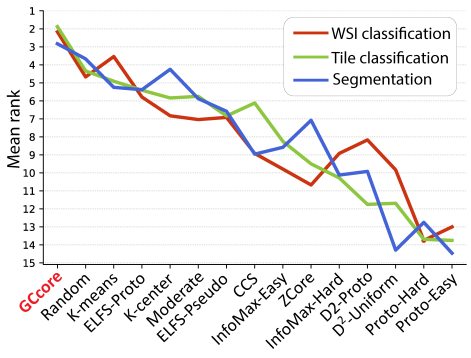

Figure 1: Mean rank of GCcore and competing label-free selection methods across 10 datasets spanning three common computational pathology tasks, in the low-retention regime (tens to hundreds of samples in the coreset that require annotation). Lower rank indicates better downstream performance.

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

Nevertheless, the development of DL models in computational pathology still relies heavily on expert-annotated data, whether for fully supervised training or for fine-tuning in the context of FMs. Obtaining such high-quality annotations is time-consuming and cognitively demanding (Montezuma et al., 2023), motivating a fundamental question: given an unlabeled histopathology dataset of N samples and a budget of $k \ll N$ annotations, which k WSIs or tiles, i.e., which coreset, should be selected for expert annotation to maximize downstream performance? Selecting a coreset without access to prior label information remains challenging, and, to date, no systematic and reproducible strategy has been established and validated for histopathology.

In computer vision, coreset selection methods have been proposed, but with the primary goal of reducing computational cost during training rather than optimizing annotation efforts (Coleman et al., 2019; Zheng et al., 2025; Griffin et al., 2026). Consequently, most methods were formulated with access to labels and only subsequently adapted to the label-free regime. Existing approaches can be broadly grouped into three categories. *Coverage-based* methods select samples according to their distribution in the embedding space, aiming to minimize redundancy among selected samples. *Importance-based* methods assign each sample a scalar importance score, derived from training dynamics or embedding geometry, and select samples based on this score alone. *Hybrid* methods combine these two principles by jointly accounting for the individual importance of each sample and the redundancy among selected samples.

However, we show that existing coreset selection methods cannot be reliably transferred to histopathology datasets for two main reasons. First, existing label-free coreset selection methods usually rely on hyperparameters that were selected based on downstream performance on a limited set of natural-image benchmarks (a full list is given in Appendix C), and neither the values nor the ranges of robustness established in this context are necessarily transferable. For instance, InfoMax (Tan et al., 2025) was reported to be robust to its pairwise weight α over the range $[0.3, 4]$ on CC12M. On PANDA at a 2% retention rate, however, varying α within this range yields quadratic weighted Cohen's κ (QWK) values between 0.58 and 0.70. Similarly, ELFS (Zheng et al., 2025) sets the number of pseudo-classes $\tilde{C}$ to the number of classes in the dataset. On TCGA-BRCA (binary classification) at a 5% retention rate, varying $\tilde{C}$ from 2 to 20 yields test balanced accuracies between 0.71 and 0.85, with the prescribed value $\tilde{C} = 2$ performing worst. Moreover, this rule does not transfer to the four-class segmentation task of BEETLE, where individual tiles do not carry a single class label; varying $\tilde{C}$ from 4 to 25 at a 0.02% retention rate yields test weighted Dice values between 0.70 and 0.84. These examples show that (1) a range reported as robust on natural images can produce substantial variation on histopathology data; (2) the recommended value can be among the worst choices; and (3) the default may be undefined, as for segmentation. Moreover, it is also unclear whether hyperparameters optimized for histopathology generalize across different cancer types and tasks, or whether they need to be determined separately for each new application, which may itself require additional labelled data in contradiction with the main goal of coreset selection. This motivates a genuinely hyperparameter-free coreset selection method for computational pathology. Second, existing methods generally provide no *a priori* guarantee on the quality of the selected subset with respect to a specified objective function. Such a guarantee is of particular value in histopathology, where every selected sample is converted into a dense annotation at a cost measured in pathologist hours, and where this cost cannot be recovered once a selection has proven poor.

To address these two challenges, we propose GCcore, a novel label-free coreset selection method that iteratively adds the sample from the remaining pool that maximizes the global coverage (GC) of the embedding space (Algorithm 1). It accounts for both coverage and importance without a weighting term, unlike existing hybrid methods: a sample is selected only when it is both representative, i.e., surrounded by many samples, and novel, i.e., far from the samples already selected. We compare GCcore with 14 baselines on 10 pathology datasets spanning WSI classification, tile classification and segmentation (Section 4). GCcore achieves the best mean rank on all three task families (2.17, 1.88 and 2.75, respectively; Fig. 1), ranking first on six datasets and within the top three on nine, while remaining computationally efficient: it selects a coreset of 100 samples from a pool of 100,000 in 89 s. We further prove GCcore attains a lower bound guarantee $1 - 1/e$ on the GC objective on any dataset and at every coreset size (Section 3).

Our contributions are: (1) **A new selection principle.** We formulate label-free coreset selection as the maximization of a scale-invariant GC objective, defined as the area under the coverage–radius curve, and show that its greedy maximization balances representativeness and diversity without any

weighting term. The resulting method, GCcore, is hyperparameter-free and deterministic, applies to the embeddings of any FM, and requires no labels. (2) **An *a priori* guarantee at every budget.** We prove that GC is monotone and submodular, so that the greedy solution attains at least $1 - 1/e$ of the maximum GC achievable at every coreset size. Its practical value is that pathologists can annotate samples in the order chosen and stop whenever the budget is exhausted, with a certified bound on the coverage of the annotated subset, whereas existing label-free methods generally offer no such assurance. (3) **Comprehensive evaluation and superior performance.** We benchmark 15 label-free coreset selection methods on 10 histopathology datasets and three tasks, including segmentation, the most annotation-intensive task in histopathology, on which none of the existing methods had previously been evaluated. GCcore achieves the best overall mean rank (2.2), whereas no baseline is consistently competitive; moreover, the rank of a baseline shifts by 0.9 to 2.1 positions on average, and by up to 4.5 positions on a single dataset, depending solely on its own hyperparameters within recommended intervals (Table 1).

2 Related work

We discuss below the existing coreset selection methods that are label-free by design or that can be adapted into a label-free variant (see Appendix C for a full list), including coverage-based, importance-based, and hybrid methods.

Coverage-based selection. Coverage-based methods select samples according to their arrangement in the embedding space and differ in the covering criterion they optimize. Clustering-based selection partitions the embedding space into groups and selects representatives from each (MacQueen, 1967; Ester et al., 1996; Arthur et al., 2007); how samples are allocated across clusters (uniformly or proportionally to cluster size) and which representatives are chosen within each cluster (the sample nearest the centroid, or samples stratified by distance from it) are hyperparameters with no principled label-free setting. K-center greedy (Sener & Savarese, 2017) minimizes the radius needed to cover the single least-covered point; the selection is therefore dictated by the extremes of the distribution and systematically favors outliers over representative samples.

Importance-based selection. Importance-based methods assign a scalar score to each sample and select samples based on this score alone, thereby treating set-level information as a sum of independent contributions. The earliest methods derive scores from training dynamics (Toneva et al., 2018; Pleiss et al., 2020; Paul et al., 2021). All require labels and training on the full dataset, and are thus inapplicable to annotation planning. Prototype (Sorscher et al., 2022) clusters features with K-means and scores each sample by its distance to its cluster centroid, keeping atypical samples when the budget is abundant and prototypical samples when it is scarce. Moderate coreset (Xia et al., 2022) instead keeps samples whose distance to their class center is closest to the class median. Without labels, both require the number of clusters, and Prototype additionally requires the budget threshold at which the pruning mode switches. CCS (Zheng et al., 2022) prunes a budget-dependent fraction of the hardest samples by AUM and samples uniformly from equal-width bins of the remaining score range; the pruning rate and the number of bins must be set for each budget. Finally, ELFS (Zheng et al., 2025) recovers training-dynamics scores through pseudo-labels obtained by deep clustering and prunes both the hardest and the easiest samples. Its selection inherits the errors of the clustering step, depends on the proxy model's training settings, and requires the number of pseudo-classes and the pruning fraction.

Hybrid methods. Hybrid methods account for both the importance of individual samples and the redundancy among selected samples, but predominantly through heuristic procedures that require weighting the two. D^2-Pruning (Maharana et al., 2024) builds a k-nearest-neighbor graph over difficulty-weighted samples, propagates neighborhood difficulty into each node's score by forward message passing, and then repeatedly selects the highest-scoring node while discounting its neighbors by reverse message passing. The number of neighbors and the forward and reverse message-passing weights, which balance difficulty against redundancy, must be fixed in advance. InfoMax (Tan et al., 2025) defines an explicit set-level objective, the sum of per-sample importance scores minus an α-weighted sum of pairwise similarities among selected samples. In its unsupervised mode, the per-sample term is the Prototype score (Sorscher et al., 2022), so InfoMax inherits the hyperparameters of that score in addition to α. ZCore (Griffin et al., 2026) draws random

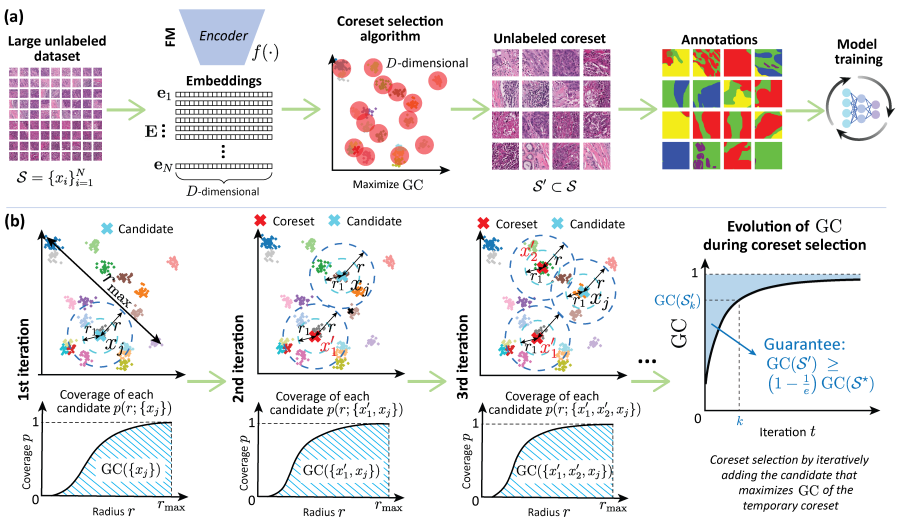

Figure 2: (a) Overview of the label-free coreset selection methodology. Embeddings of the full training set are first extracted using any histopathology FM and represented in feature space, after which the proposed algorithm identifies a subset of samples that collectively maximize coverage, defining the coreset. Annotations for these selected samples are then obtained from pathologists and used to train any task-specific model. (b) Schematic illustration of the proposed coreset selection algorithm. Each iteration adds the single sample whose marginal gain in GC, evaluated jointly with the samples already selected, is largest, so that GC increases monotonically as the coreset grows.

probes in random low-dimensional subspaces, rewards the sample nearest each probe while penalizing its neighbors, and retains the top-k scorers; the procedure is stochastic and trades runtime against performance through the number of sampling iterations, which must be set together with the dimensionality of the subspaces.

In summary, every method above exposes at least one hyperparameter whose value was calibrated on natural-image benchmarks and cannot be re-tuned on a new cohort without the labels that coreset selection is meant to spare. We treat the random seed as a hyperparameter as well, since for stochastic methods, such as K-center, K-means and ZCore, it determines the selected subset. Moreover, none of these methods offers an *a priori* guarantee for an objective that reflects coverage of the data distribution as a whole. A third limitation concerns evaluation: none of these methods have been assessed on segmentation, the most annotation-intensive task in histopathology, and the one for which the number of clusters required by clustering-based selection, Prototype and ELFS is hardest to set, since it does not correspond to any observable notion of class.

3 METHODS

As illustrated in Fig. 2(a), the overall pipeline consists of three steps. First, the entire unlabeled dataset $\mathcal{S} = \{x_1, x_2, \ldots, x_N\}$ is encoded by a pre-trained FM $f(\cdot)$ into embeddings $\mathbf{E} = \{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_N\}$, where $\mathbf{e}_i = f(x_i) \in \mathbb{R}^D$, N is the number of samples, and D is the feature dimension. Second, our proposed method, the global coverage-based coreset selection algorithm (GCcore), identifies a representative subset $\mathcal{S}' \subset \mathcal{S}$ of size $|\mathcal{S}'| = k \ll N$ that covers the data distribution in the embedding space through iterative global coverage maximization; the selection mechanism is illustrated in Fig. 2(b) and detailed below. Finally, the selected coreset is annotated by pathologists to provide ground-truth labels, which serve as the basis for supervised training. Table 3 in Appendix B summarizes the notation used throughout.

Coverage measure. A natural way to quantify how well a subset $\mathcal{T} \subseteq \mathcal{S}$ covers the dataset is the fraction of samples that lie within a radius r of their nearest subset member. For each sample x_i, let

$$d_i(\mathcal{T}) = \min_{x_j \in \mathcal{T}} \|\mathbf{e}_i - \mathbf{e}_j\|_1, \qquad \forall\, x_i \in \mathcal{S}, \tag{1}$$

denote its nearest-neighbor distance to $\mathcal{T}$, and collect these into $\mathbf{d}(\mathcal{T}) = \big(d_1(\mathcal{T}), \ldots, d_N(\mathcal{T})\big)^\top \in \mathbb{R}^N$, where $\|\cdot\|_1$ is the ℓ_1 (Manhattan) norm, $\|\mathbf{e}_i - \mathbf{e}_j\|_1 = \sum_{m=1}^{D} |e_{im} - e_{jm}|$, which we adopt throughout unless stated otherwise. The coverage at radius r is

$$p(r; \mathcal{T}) \;=\; \frac{1}{N} \big| \{\, x_i \in \mathcal{S} : d_i(\mathcal{T}) \le r \,\} \big|, \tag{2}$$

where $|\cdot|$ denotes the cardinality of a set. The scalar $p(r; \mathcal{T})$ is highly sensitive to the choice of r, which is arbitrary and varies with the scale of the embeddings across datasets. To obtain a scale-robust measure, we integrate $p(r; \mathcal{T})$ over all radii up to the maximum pairwise distance $r_{\max} = \max_{i<j} \|\mathbf{e}_i - \mathbf{e}_j\|_1$, a constant determined by $\mathcal{S}$ alone and strictly positive whenever $\mathcal{S}$ contains at least two distinct embeddings, and normalize, defining the global coverage as the area under the coverage–radius curve:

$$\mathrm{GC}(\mathcal{T}) \;=\; \frac{1}{r_{\max}} \int_0^{r_{\max}} p(r; \mathcal{T})\, dr. \tag{3}$$

Global objective. Coreset selection is then formulated as the maximization of coverage subject to a cardinality budget,

$$\mathcal{S}^\star \;=\; \operatorname*{arg\,max}_{\mathcal{T} \subseteq \mathcal{S},\, |\mathcal{T}| = k} \; \mathrm{GC}(\mathcal{T}). \tag{4}$$

In contrast to per-sample importance-scoring methods, which assign each example a fixed importance score independently of which other examples are selected, Eq. 4 specifies a single global objective defined over the selected set as a whole, enabling the approximation guarantee established in Appendix D.

Efficient coverage evaluation. Directly estimating GC requires sampling r_{steps} radii to evaluate the coverage $p(r; \mathcal{T})$ and integrating the resulting curve, incurring an $O(N r_{\text{steps}})$ cost per candidate evaluation. We avoid the radius sweep entirely via the tail-integral identity for the nonnegative quantities $d_i(\mathcal{T})$, which satisfy $d_i(\mathcal{T}) \le r_{\max}$ for every $x_i \in \mathcal{S}$:

$$\bar{d}(\mathcal{T}) \;=\; \frac{1}{N} \sum_{i=1}^{N} d_i(\mathcal{T}) \;=\; \frac{1}{N} \sum_{i=1}^{N} \int_0^{r_{\max}} \mathbf{1}\{d_i(\mathcal{T}) > r\}\, dr. \tag{5}$$

Substituting the definition of $p(r; \mathcal{T})$ from Eq. 2 gives

$$\bar{d}(\mathcal{T}) \;=\; \int_0^{r_{\max}} \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\{d_i(\mathcal{T}) > r\}\, dr \;=\; \int_0^{r_{\max}} \big(1 - p(r; \mathcal{T})\big)\, dr, \tag{6}$$

and comparison with Eq. 3 yields

$$\mathrm{GC}(\mathcal{T}) \;=\; 1 - \frac{\bar{d}(\mathcal{T})}{r_{\max}}. \tag{7}$$

The global objective of maximizing GC is therefore equivalent to minimizing the mean nearest-coreset distance $\bar{d}$. Beyond removing the $O(r_{\text{steps}})$ factor and its discretization error, we note that, when a sample x_j is added to the coreset, the nearest-coreset distances need only be updated by an elementwise minimum,

$$d_i(\mathcal{T} \cup \{x_j\}) \;=\; \min\big(d_i(\mathcal{T}),\, \|\mathbf{e}_i - \mathbf{e}_j\|_1\big), \qquad \forall\, x_i \in \mathcal{S}. \tag{8}$$

Maintaining $\mathbf{d}$ thus lets each candidate be scored by a single $O(ND)$ update, rather than the $O(NkD)$ of recomputing all nearest distances from scratch.

Greedy construction. As summarized in Algorithm 1, we optimize Eq. 4 greedily. Starting from $\mathcal{S}' = \varnothing$, each iteration scores every candidate x_j in the remaining pool by its marginal gain,

$$g_j(\mathcal{S}') \;=\; \bar{d}(\mathcal{S}') - \bar{d}\big(\mathcal{S}' \cup \{x_j\}\big) \;=\; \frac{1}{N} \sum_{i=1}^{N} \big(d_i(\mathcal{S}') - \|\mathbf{e}_i - \mathbf{e}_j\|_1\big)_+, \tag{9}$$

where $(\cdot)_+ = \max\{0, \cdot\}$ and the second equality follows from Eq. 8, and permanently adds the maximizer. Because appending a sample can only decrease each d_i, every g_j is non-negative (monotonicity) and non-increasing across iterations (submodularity), the diminishing-returns property proved

in Lemma 1. A gain computed at an earlier iteration is therefore always an upper bound on the current one, which the lazy greedy variant of (Minoux, 2005) exploits: the gains are kept in a max-heap $\mathcal{Q}$, and each iteration re-evaluates only the top element, re-inserting it if its exact gain falls below the next key (Algorithm 1, lines 10–19). A recomputed top gain that still dominates the heap is provably the exact maximizer, so most candidate evaluations are skipped while the selection remains identical to naive greedy.

Global optimality guarantee. In Appendix D, we prove that the objective in Eq. 4 is monotone submodular, so that the greedy algorithm (Algorithm 1) attains at least $(1 - 1/e)$ of the maximum coverage achievable under the same budget (Nemhauser et al., 1978), a dataset-independent, *a priori* lower bound that holds at every coreset size.

4 Experiments

We assessed the performance of GCcore on 10 datasets (dataset details in Appendix E) covering three common yet distinct computational pathology tasks: WSI classification, tile classification and tissue segmentation. Collectively, these datasets span prostate, colorectal, breast, lung and gastric tissue, as well as a pan-cancer cohort, with several datasets collected across multiple medical centers.

4.1 Experimental setup

For each dataset, we first extracted embeddings for all images using an FM. Coresets of varying sizes were then selected from the embeddings of the full training set, and used to train a model comprising a frozen FM as the encoder followed by a trainable task-specific head. Performance was evaluated on the held-out test set. Unless otherwise specified, test performance is measured using QWK for ordinal multiclass classification, balanced accuracy for binary classification, and weighted Dice for segmentation, following the community standard (Ding et al., 2025). For each dataset, we defined four coreset sizes in the low-retention regime (0.01%-20% depending on the dataset, as detailed in Appendix E.1), corresponding to the selection of only a few tens to a few hundred samples for annotation, a quantity that could realistically be annotated by a pathologist in clinical practice.

Baselines We compared our deterministic GCcore against 14 label-free baselines. For each baseline we sweep a single characteristic hyperparameter over five values, sampled within the range reported as robust on natural-image benchmarks or around the reported default value, holding the remainder at their default settings, and report the mean test performance across the five configurations. We sweep rather than fix a value because the defaults and robust ranges of these hyperparameters, established on natural images, generally do not transfer reliably to histopathology (see Appendix F.1), so any single value could favor or penalize a baseline. Averaging over five values instead estimates the performance a practitioner can expect on a new cohort without labels for tuning, and the standard deviation (SD) quantifies the sensitivity of the baseline to its own hyperparameter.

(1) **Random** selection, across random seeds. (2) **K-center** (Sener & Savarese, 2017), across random seeds. (3) **K-means** (MacQueen, 1967), across the number of clusters M. The budget is allocated to clusters in proportion to their size, and within each cluster we perform radial stratified sampling at evenly spaced distances from the centroid, so that the subset spans each cluster from its core to its periphery. (4) **Proto** (Sorscher et al., 2022), across the number of clusters M. We define the *Proto score* of a sample as the cosine distance to its assigned cluster centroid, so larger values indicate less prototypical samples. We retain either the hardest (least prototypical) samples, corresponding to the data-abundant regime, or the easiest (most prototypical) ones, corresponding to the data-scarce regime, denoted **Proto-Hard** and **Proto-Easy**. (5) **Moderate** (Xia et al., 2022), across the number of clusters M, retaining the samples whose Proto score lies closest to the per-cluster median. (6) **CCS** (Zheng et al., 2022), across the hard-cutoff rate β. Using the Proto score as difficulty, CCS prunes the hardest β fraction, stratifies the remainder into equal-width difficulty bins, and allocates the budget to the least-populated bins first. (7) **D^2** (Maharana et al., 2024), across the number of nearest neighbors k_{nn} in the graph, with a uniform initial difficulty score (**D^2-Uniform**), and with the Proto score (**D^2-Proto**). (8) **ELFS** (Zheng et al., 2025), performing double-end pruning with either the AUM (Pleiss et al., 2020) of a pseudo-label linear probe (**ELFS-Pseudo**), swept across the number of pseudo-classes $\tilde{C}$, or the Proto score (**ELFS-Proto**), swept across the number of

Table 1: Mean rank of the 15 coreset selection methods on each dataset; the subscript denotes the SD of that rank across five hyperparameter values. GCcore is deterministic and reports no SD. **Mean**: average across the 10 datasets.

Selection method	WSI classification			Tile classification				Segmentation			**Mean**
	PANDA	IMP-CRS	TCGA-BRCA	WSSS4 LUAD	SICAP	CRC-100k	Deep HP	BEE TLE	panCK	CAME-LYON16	
GCcore	3.0	**1.0**	**2.5**	**1.0**	2.5	3.0	**1.0**	**1.3**	5.8	**1.3**	**2.2**
Random	$6.5_{1.8}$	$4.8_{1.2}$	$2.8_{1.2}$	$5.5_{1.9}$	$3.3_{1.9}$	$5.0_{1.1}$	$3.5_{0.9}$	$2.3_{1.6}$	$4.0_{1.6}$	$5.5_{1.1}$	$4.3_{1.4}$
K-means	$\mathbf{2.0}_{0.8}$	$3.0_{1.4}$	$5.8_{2.7}$	$6.6_{1.6}$	$6.0_{3.1}$	$\mathbf{1.3}_{0.9}$	$5.8_{0.7}$	$3.5_{0.8}$	$5.0_{2.0}$	$7.0_{1.8}$	$4.6_{1.6}$
ELFS-Proto	$5.8_{1.3}$	$4.8_{1.4}$	$6.8_{3.1}$	$8.8_{1.4}$	$\mathbf{2.0}_{3.0}$	$7.3_{1.0}$	$3.8_{0.5}$	$4.3_{1.2}$	$8.5_{1.7}$	$4.8_{0.8}$	$5.7_{1.5}$
K-center	$3.0_{1.5}$	$6.3_{0.8}$	$11.3_{4.5}$	$6.5_{3.0}$	$4.8_{1.7}$	$4.0_{1.5}$	$8.3_{0.4}$	$6.5_{0.4}$	$\mathbf{3.3}_{0.6}$	$2.8_{0.7}$	$5.7_{1.5}$
Moderate	$7.8_{2.8}$	$9.0_{1.5}$	$4.5_{2.4}$	$9.1_{2.6}$	$5.8_{1.4}$	$4.3_{1.9}$	$4.0_{1.0}$	$3.8_{1.2}$	$8.3_{2.8}$	$7.3_{2.7}$	$6.4_{2.0}$
ELFS-Pseudo	$4.5_{1.7}$	$6.0_{2.4}$	$10.3_{3.4}$	$4.0_{1.4}$	$8.3_{2.6}$	$8.0_{1.5}$	$7.0_{0.8}$	$6.8_{0.9}$	$9.0_{2.2}$	$4.0_{1.1}$	$6.8_{1.8}$
CCS	$7.3_{0.9}$	$10.8_{2.7}$	$8.8_{2.6}$	$6.5_{1.2}$	$6.0_{1.7}$	$8.5_{2.3}$	$3.5_{0.2}$	$8.5_{3.3}$	$6.3_{2.1}$	$11.3_{3.8}$	$7.7_{2.1}$
InfoMax-Easy	$8.8_{3.3}$	$10.3_{0.3}$	$10.3_{1.9}$	$7.3_{3.8}$	$10.8_{0.7}$	$6.5_{3.3}$	$8.5_{1.7}$	$8.5_{1.5}$	$6.8_{3.8}$	$6.8_{0.3}$	$8.4_{2.0}$
ZCore	$11.5_{1.3}$	$8.8_{0.8}$	$11.8_{1.0}$	$9.5_{2.5}$	$10.3_{0.9}$	$7.3_{2.0}$	$11.0_{0.9}$	$10.3_{0.4}$	$5.5_{1.9}$	$6.0_{2.8}$	$9.2_{1.4}$
InfoMax-Hard	$11.0_{1.4}$	$5.5_{1.5}$	$10.3_{1.7}$	$5.0_{1.2}$	$12.0_{1.7}$	$11.0_{2.0}$	$13.0_{0.4}$	$10.5_{0.5}$	$9.3_{1.4}$	$9.8_{2.3}$	$9.7_{1.4}$
D^2-Proto	$9.3_{2.7}$	$9.0_{2.7}$	$6.3_{1.1}$	$8.8_{2.2}$	$12.0_{2.1}$	$12.8_{0.4}$	$13.5_{1.1}$	$13.5_{0.3}$	$6.5_{2.4}$	$11.8_{3.9}$	$10.3_{1.9}$
D^2-Uniform	$11.0_{0.9}$	$14.3_{0.1}$	$4.3_{3.5}$	$12.8_{4.5}$	$7.5_{1.7}$	$14.8_{0.4}$	$11.8_{1.0}$	$14.3_{0.4}$	$14.0_{1.9}$	$14.9_{0.2}$	$11.9_{1.5}$
Proto-Hard	$14.8_{1.6}$	$12.0_{1.4}$	$14.5_{2.5}$	$13.8_{2.0}$	$14.5_{3.2}$	$12.3_{0.4}$	$14.3_{0.5}$	$12.0_{0.5}$	$13.0_{1.8}$	$13.0_{1.5}$	$13.4_{1.5}$
Proto-Easy	$14.0_{2.1}$	$14.8_{0.2}$	$10.3_{2.0}$	$15.0_{0.3}$	$14.5_{1.1}$	$14.3_{0.2}$	$11.3_{1.3}$	$14.3_{0.1}$	$15.0_{0.1}$	$14.1_{1.1}$	$13.7_{0.9}$

clusters M. (9) **InfoMax** (Tan et al., 2025), across the pairwise weight α, with the Proto score as the per-sample informativeness term: higher scores are more informative in hard mode (**InfoMax-Hard**), lower in easy mode (**InfoMax-Easy**). (10) **ZCore** (Griffin et al., 2026), across random seeds.

WSI classification For WSI classification, we conducted experiments on PANDA (Bulten et al., 2022), IMP-CRS (Oliveira et al., 2021; Neto et al., 2022; 2024), and TCGA-BRCA datasets. We used the PRISM slide encoder (Shaikovski et al., 2024) to extract the slide embeddings of WSIs. The classification head is a single linear layer that maps a slide embedding $\mathbf{e} \in \mathbb{R}^D$ to C logits, where D is the native dimensionality of the slide encoder and C is the number of classes defined by the dataset label. Since the classes are mutually exclusive, the logits are normalized by a softmax and the head is trained with class-weighted cross-entropy loss.

Tile classification The tile classification task encompasses experiments on four datasets, namely WSSS4LUAD (Han et al., 2022), SICAP (Silva-Rodríguez et al., 2020), CRC-100k (Kather et al., 2018), and DeepHP (Goncalves et al., 2022; Gonçalves e Gonçalves et al., 2023). We used the H-optimus-1 tile encoder (Scalbert et al., 2026) to extract the embeddings of tiles. For SICAP, CRC-100k and DeepHP, the classification head is identical to that of WSI classification, except that D is the dimensionality of the tile encoder. In WSSS4LUAD the tile labels are not mutually exclusive, since one tile may contain several tissue types; we therefore replace the softmax with $C = 3$ independent sigmoids applied to the logits of the same linear layer and train with binary cross-entropy loss. Performance on WSSS4LUAD is measured using macro AUROC, averaged over the three classes. Performance on CRC-100k is measured using balanced accuracy, since its nine tissue classes are nominal and QWK presumes an ordinal label scale.

Segmentation We experimented on BEETLE (Lems et al., 2025), panCK (Komura et al., 2023), and CAMELYON16 (Ehteshami Bejnordi et al., 2017) for segmentation. For coreset selection we use the global [CLS] token of H-optimus-1 (Scalbert et al., 2026) as the representation of each tile, as in the tile-classification experiments. For training and evaluation, however, the segmentation head instead uses the patch tokens, reshaped into a two-dimensional grid that retains the spatial correspondence between each token and the region it covers within the input tile. The segmentation head comprises four hierarchical upsampling stages producing pixel-wise class predictions at the working resolution, as detailed in Appendix F.3. It is trained with class-weighted cross-entropy, with unannotated pixels, the ignore class in BEETLE and background in CAMELYON16, excluded from both the loss and the evaluation metrics.

Table 2: Test performance of the 15 selection methods at the lowest retention rate (0.01%–5%, depending on the dataset; see Appendix E.1) resulting in only a few tens of samples selected for annotation; the subscript is the SD across five hyperparameter values. All values are multiplied by 100. GCcore is deterministic and reports no SD. D^2-Uniform on DeepHP is marked because its selected subset contains samples from only one of the two classes.

Selection method	WSI classification			Tile classification				Segmentation		
	PANDA	IMP-CRS	TCGA-BRCA	WSSS4 LUAD	SICAP	CRC-100k	Deep HP	BEE TLE	panCK	CAME-LYON16
GCcore	73.3	**91.7**	85.9	**94.8**	66.8	**86.6**	**80.0**	**84.4**	**76.2**	**97.8**
Random	$70.8_{5.8}$	$87.4_{1.9}$	$\mathbf{89.8}_{3.0}$	$93.5_{1.2}$	$62.9_{8.7}$	$74.6_{9.0}$	$75.0_{3.4}$	$81.6_{2.7}$	$75.9_{0.9}$	$94.9_{1.6}$
K-means	$73.7_{3.2}$	$86.2_{2.9}$	$88.0_{2.9}$	$92.2_{1.3}$	$68.6_{2.8}$	$85.4_{5.9}$	$71.7_{4.4}$	$84.2_{1.5}$	$75.8_{0.7}$	$91.2_{4.4}$
ELFS-Proto	$\mathbf{74.2}_{3.1}$	$85.8_{2.3}$	$86.4_{2.8}$	$91.8_{1.3}$	$\mathbf{69.7}_{5.3}$	$74.7_{6.0}$	$72.7_{4.2}$	$80.1_{3.6}$	$75.3_{0.7}$	$95.5_{2.5}$
K-center	$71.5_{4.3}$	$88.4_{1.1}$	$79.9_{8.5}$	$92.7_{2.1}$	$63.4_{8.5}$	$77.0_{9.0}$	$65.5_{2.6}$	$76.5_{2.1}$	$74.4_{0.8}$	$96.8_{0.6}$
Moderate	$68.1_{3.3}$	$85.8_{4.1}$	$85.5_{2.8}$	$92.9_{2.1}$	$62.2_{4.1}$	$80.2_{7.1}$	$66.6_{7.3}$	$81.6_{2.7}$	$74.6_{2.5}$	$92.2_{3.2}$
ELFS-Pseudo	$68.1_{9.2}$	$82.6_{7.7}$	$84.3_{4.5}$	$92.9_{1.6}$	$48.2_{10.4}$	$81.4_{4.9}$	$69.4_{4.5}$	$79.5_{5.6}$	$74.9_{0.8}$	$96.5_{2.4}$
CCS	$71.7_{2.9}$	$85.6_{2.5}$	$80.9_{5.8}$	$93.4_{0.6}$	$67.2_{7.1}$	$73.6_{8.9}$	$74.4_{2.3}$	$76.1_{10.1}$	$75.9_{0.6}$	$90.8_{6.3}$
InfoMax-Easy	$62.4_{9.3}$	$76.7_{10.3}$	$84.3_{0.4}$	$94.0_{3.3}$	$51.3_{2.4}$	$75.1_{16.2}$	$70.8_{4.6}$	$79.0_{5.9}$	$69.8_{8.7}$	$96.3_{0.4}$
ZCore	$60.1_{6.4}$	$58.9_{8.3}$	$76.4_{10.2}$	$88.7_{4.1}$	$36.6_{3.5}$	$64.9_{12.8}$	$60.3_{11.3}$	$67.6_{4.0}$	$68.7_{4.1}$	$93.5_{5.8}$
InfoMax-Hard	$47.3_{24.6}$	$83.7_{5.3}$	$81.1_{5.8}$	$93.4_{3.1}$	$37.2_{20.4}$	$44.2_{19.7}$	$52.6_{4.8}$	$71.1_{3.4}$	$70.9_{1.8}$	$91.8_{5.3}$
D^2-Proto	$51.5_{11.9}$	$51.0_{31.7}$	$82.6_{4.5}$	$86.5_{5.9}$	$35.8_{8.2}$	$15.3_{3.6}$	$62.5_{9.5}$	$36.8_{15.1}$	$73.9_{2.8}$	$86.3_{4.8}$
D^2-Uniform	$53.2_{10.8}$	$-8.4_{12.7}$	$87.2_{4.5}$	$89.7_{2.8}$	$53.0_{16.3}$	$8.5_{1.6}$	–	$44.8_{2.5}$	$63.1_{5.0}$	$84.1_{0.0}$
Proto-Hard	$37.7_{14.7}$	$52.1_{31.3}$	$74.5_{9.7}$	$75.2_{12.3}$	$19.0_{15.9}$	$17.1_{3.7}$	$51.3_{1.5}$	$48.3_{4.3}$	$63.7_{3.7}$	$84.5_{0.8}$
Proto-Easy	$40.3_{18.4}$	$-12.9_{21.6}$	$81.2_{5.2}$	$57.6_{26.4}$	$34.8_{10.9}$	$11.9_{2.4}$	$61.9_{2.6}$	$46.3_{0.0}$	$53.4_{8.6}$	$84.1_{0.0}$

4.2 Main results

Tables 1 and 2 show that no single baseline is consistently competitive. GCcore achieves the best mean rank over the 10 datasets, 2.2 out of 15, ahead of random selection (4.3), K-means (4.6), and ELFS-Proto and K-center (both 5.7). It ranks first on six of the 10 datasets and within the top three on nine. Every baseline that is competitive on average is weak somewhere: K-means is best on CRC-100k (1.3) but reaches only 7.0 on CAMELYON16, K-center is best on panCK (3.3) but 11.3 on TCGA-BRCA, and ELFS-Proto is best on SICAP (2.0) but 8.8 on WSSS4LUAD. Table 2 shows the same asymmetry in test performance at the lowest retention rate of each dataset. GCcore achieves the best test performance on seven of the 10 datasets and trails the best method by at most 3.9 points on the remaining three, whereas a competitive baseline that outperforms GCcore on one dataset can be behind GCcore by up to 11.9 points on another (Random, $+3.9$ on TCGA-BRCA against -11.9 on CRC-100k; ELFS-Proto, $+2.9$ on SICAP against -11.9 on CRC-100k; K-means, $+2.2$ on TCGA-BRCA against -8.3 on DeepHP).

The subscripts in both tables quantify the sensitivity of a method to its own configuration. For each dataset the subscript is the SD, across five hyperparameter values, of the rank (Table 1) or of the test performance (Table 2) the method obtains. Because GCcore selects its coreset as a deterministic function of the embeddings and exposes no such hyperparameter, this SD is zero on every dataset, whereas the rank SD of a baseline averages 0.9 to 2.1 positions across datasets and reaches 4.5 positions on a single dataset. This sensitivity is not predicted by how well a baseline ranks: across the 14 baselines, the Spearman correlation between mean rank and mean rank SD is weak and not significant ($\rho = -0.21$, $p = 0.48$). A baseline that ranks well on average is therefore not necessarily one whose rank can be trusted from a single configuration. Figure 4 resolves the test-performance subscripts of Table 2 per retention rate, with GCcore drawn as a single level over each group and each box spanning one SD of a baseline. It further illustrates that a baseline mean above the GCcore level does not imply a reliable advantage. At 0.5% retention on PANDA, K-means with number of clusters $M = 6, 10, 15, 20, 25$ yields 72.3, 71.7, 73.8, 79.2 and 71.3, a mean of 73.7 against 73.3 for GCcore, yet three of the five K-means configurations fall below GCcore and the mean is carried by a single setting. Choosing the value of M that realizes the advantage requires held-out labels, which is what a clinical low-label setting precludes. See Appendix G.1 for further details.

Computational efficiency We measured the selection runtime of GCcore and 14 baselines on a single machine (AMD Ryzen Threadripper PRO 3995WX, 64 cores / 128 threads, 256 GB RAM)

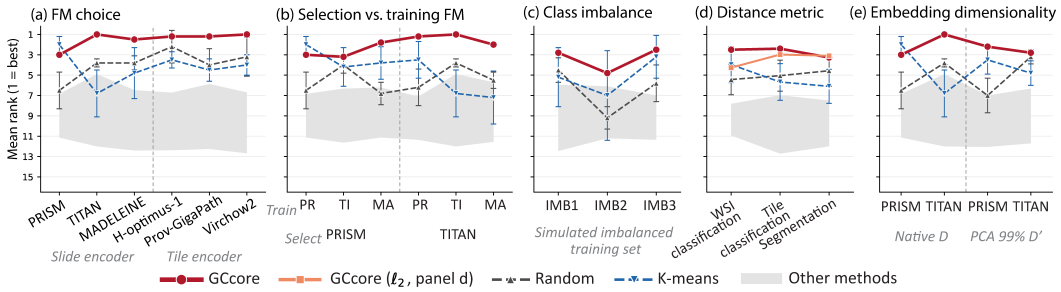

Figure 3: Ablation studies over the experimental design and the data: FM choice, FM choice for selection and for model training, class imbalance in the data, distance metric used in GCcore, and dimensionality of the FM embeddings. Experiments in (a) are conducted on PANDA and BEETLE, those in (b), (c) and (e) on PANDA, and those in (d) on all 10 datasets.

using CRC-100k, averaged over five runs. Under low retention rates (20–200 tiles retained from a 100,000-tile training set), GCcore averages 86.4 s and is faster than 12 of the 14 baselines, behind only Random (0.0002 s) and K-center (0.49 s). Selecting 100 tiles, for instance, takes GCcore 88.9 s, against 0.0002 s for random and 0.53 s for K-center. In return, GCcore attains a mean balanced accuracy of 0.905 across the four budgets, against 0.868 for random and 0.875 for K-center.

4.3 Ablation studies

We investigate the robustness of GCcore to the experimental design and to the data: the choice of FM for extracting the embeddings, a discrepancy between the FM used for coreset selection and the FM used for downstream model training, class imbalance in the dataset, the distance used to instantiate the objective, and the embedding dimensionality. Figure 3 summarizes the five studies and Appendix G.2–G.6 give the experimental details and the analysis for each. (a) **FM choice (Table 7).** GCcore is the best-ranked of the 15 selection methods under five of the six feature extractors and is never worse than rank 3.0, giving a mean rank of 1.5 against 3.9 for the strongest baseline; the advantage is therefore a property of the selection method rather than of any particular encoder. (b) **Selection vs. training FM (Table 8).** GCcore attains the best mean rank of all methods (2.7); whenever a stronger FM (PRISM or TITAN) provides the selection embedding, it is among the two best selection methods irrespective of which FM is used for training, so a coreset need not be recomputed for a different downstream encoder. (c) **Class imbalance (Table 9).** Across three simulated imbalanced training sets (imbalance ratio 55.7–231.3) GCcore has the best mean rank (3.3) and is the only strategy placed among the two best on all three; on IMB2 it recovers 90.7% of the full-data test performance from 19 WSIs, which Random does not match until 751 WSIs. (d) **Distance metric (Table 10).** GCcore with the ℓ_1 and with the ℓ_2 distance both rank ahead of every baseline on average (2.7 and 3.4 among 16, against 5.0 for the strongest baseline), so the advantage follows from the global coverage maximization mechanism rather than from the distance used to instantiate it. Note that the guarantee of Eq. 11 holds for any distance metric. (e) **Embedding dimensionality (Table 11).** After principal component analysis (PCA) retaining 99% of the explained variance, which removes 73–85% of the dimensions, GCcore remains the best-ranked method (2.5), suggesting that GCcore can be applied directly to compressed slide representations without re-tuning.

5 Conclusion

To optimize the annotation and training process of DL models in computational pathology, we proposed a novel hyperparameter-free coreset selection algorithm, GCcore, that exploits embeddings from any histopathology FM to identify representative samples within heterogeneous and unlabeled histological datasets. Overall, at clinically realistic retention rates, our approach outperformed existing methods and showed a more consistent performance across different cancer types and tasks. GCcore therefore provides a reliable and reproducible strategy for optimal data selection, identifying informative subsets that support downstream training while substantially reducing the required labeling effort.

AI USE STATEMENT

In this work, we used generative AI tools to implement methods and assist with translation. We have not used generative AI tools to help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, design or provide feedback on research methodology or experiments, support qualitative and thematic data analysis, and interpret results; generating synthetic datasets, proposing or refining hypotheses, and cleaning and reformatting datasets are not applicable to this work. Additionally, we used generative AI tools to create or edit software code, summarize or analyze existing literature, brainstorm, source/search for information, edit the research paper to improve readability, and identify relevant literature. We have reviewed all AI-assisted work: the authors verified the correctness of LLM-generated code, checked that readability edits and translated passages preserved the original meaning, and screened LLM-suggested references for relevance by carefully reading them. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

All datasets used in this work are publicly available datasets obtained from established public sources. We do not collect any new data involving human subjects.

REPRODUCIBILITY STATEMENT

We provide detailed experimental settings in the appendix, along with the pseudo-code for the proposed method in Algorithm 1. The code will be released.

REFERENCES

David Arthur, Sergei Vassilvitskii, et al. k-means++: The advantages of careful seeding. In *Soda*, volume 7, pp. 1027–1035, 2007.

Rohan Bareja, Francisco Carrillo-Perez, Yuanning Zheng, Marija Pizurica, Tarak Nath Nandi, Jeanne Shen, Ravi Madduri, and Olivier Gevaert. Evaluating vision and pathology foundation models for computational pathology: A comprehensive benchmark study. *medRxiv*, pp. 2025–05, 2025.

Wouter Bulten, Kimmo Kartasalo, Po-Hsuan Cameron Chen, Peter Ström, Hans Pinckaers, Kunal Nagpal, Yuannan Cai, David F Steiner, Hester Van Boven, Robert Vink, et al. Artificial intelligence for diagnosis and gleason grading of prostate cancer: the panda challenge. *Nature medicine*, 28(1):154–163, 2022.

Gabriele Campanella, Shengjia Chen, Manbir Singh, Ruchika Verma, Silke Muehlstedt, Jennifer Zeng, Aryeh Stock, Matt Croken, Brandon Veremis, Abdulkadir Elmas, et al. A clinical benchmark of public self-supervised pathology foundation models. *Nature Communications*, 16(1): 3640, 2025.

Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. Selection via proxy: Efficient data selection for deep learning. *arXiv preprint arXiv:1906.11829*, 2019.

Tong Ding, Sophia J Wagner, Andrew H Song, Richard J Chen, Ming Y Lu, Andrew Zhang, Anurag J Vaidya, Guillaume Jaume, Muhammad Shaban, Ahrong Kim, et al. A multimodal whole-slide foundation model for pathology. *Nature medicine*, 31(11):3749–3761, 2025.

Babak Ehteshami Bejnordi, Mitko Veta, Paul Johannes van Diest, Bram Van Ginneken, Nico Karssemeijer, Geert Litjens, Jeroen AWM Van Der Laak, Camelyon16 Consortium, Meyke Hermsen, Quirine F Manson, et al. Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. *Jama*, 318(22):2199–2210, 2017.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pp. 226–231, 1996.

Wanderson Goncalves E Goncalves, Marcelo Henrique Paula dos Santos, Leonardo Miranda Brito, Helber Gonzales Almeida Palheta, Fábio Manoel França Lobato, Samia Demachki, Andrea Ribeiro-dos Santos, and Gilderlanio Santana de Araújo. Deephp: A new gastric mucosa histopathology dataset for helicobacter pylori infection diagnosis. *International Journal of Molecular Sciences*, 23(23):14581, 2022.

Wanderson Gonçalves e Gonçalves, Marcelo Henrique Paula Dos Santos, Leonardo Miranda Brito, Helber Gonzales Almeida Palheta, Fábio Manoel França Lobato, Samia Demachki, Ândrea Ribeiro-Dos-Santos, and Gilderlanio Santana de Araújo. Deephp: A gastric mucosa histopathology image collection for helicobacter pylori infection diagnosis, 2023.

Brent A Griffin, Jacob Marks, and Jason J Corso. Zero-shot coreset selection via iterative subspace sampling. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2114–2124, 2026.

Chu Han, Xipeng Pan, Lixu Yan, Huan Lin, Bingbing Li, Su Yao, Shanshan Lv, Zhenwei Shi, Jinhai Mai, Jiatai Lin, et al. Wsss4luad: Grand challenge on weakly-supervised tissue semantic segmentation for lung adenocarcinoma. *arXiv preprint arXiv:2204.06455*, 2022.

Guillaume Jaume, Anurag Vaidya, Andrew Zhang, Andrew H. Song, Richard J. Chen, Sharifa Sahai, Dandan Mo, Emilio Madrigal, Long Phi Le, and Faisal Mahmood. Multistain pretraining for slide representation learning in pathology. In *European Conference on Computer Vision*, pp. 19–37. Springer, 2024.

Jakob Nikolas Kather, Niels Halama, and Alexander Marx. 100,000 histological images of human colorectal cancer and healthy tissue. *Zenodo*, 2018.

Daisuke Komura, Takumi Onoyama, Koki Shinbo, Hiroto Odaka, Minako Hayakawa, Mieko Ochi, Ranny Rahaningrum Herdiantoputri, Haruya Endo, Hiroto Katoh, Tohru Ikeda, et al. Restaining-based annotation for cancer histology segmentation to overcome annotation-related limitations among pathologists. *Patterns*, 4(2), 2023.

Carlijn Lems, Leslie Tessier, John-Melle Bokhorst, Mart van Rijthoven, Witali Aswolinskiy, Matteo Pozzi, Natalie Klubickova, Suzanne Dintzis, Michela Campora, Maschenka Balkenhol, et al. A multicentric dataset for training and benchmarking breast cancer segmentation in h&e slides. *arXiv preprint arXiv:2510.02037*, 2025.

J MacQueen. Multivariate observations. In *Proceedings of the 5th Berkeley symposium on mathematical statistics and probability*, volume 1, pp. 281–297. University of California press Oakland, CA, USA, 1967.

Adyasha Maharana, Prateek Yadav, and Mohit Bansal. D2 pruning: Message passing for balancing diversity & difficulty in data pruning. In *International Conference on Learning Representations*, volume 2024, pp. 55740–55759, 2024.

Michel Minoux. Accelerated greedy algorithms for maximizing submodular set functions. In *Optimization Techniques: Proceedings of the 8th IFIP Conference on Optimization Techniques Würzburg, September 5–9, 1977*, pp. 234–243. Springer, 2005.

Diana Montezuma, Sara P Oliveira, Pedro C Neto, Domingos Oliveira, Ana Monteiro, Jaime S Cardoso, and Isabel Macedo-Pinto. Annotating for artificial intelligence applications in digital pathology: A practical guide for pathologists and researchers. *Modern Pathology*, 36(4):100086, 2023.

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

Pedro C Neto, Sara P Oliveira, Diana Montezuma, João Fraga, Ana Monteiro, Liliana Ribeiro, Sofia Gonçalves, Isabel M Pinto, and Jaime S Cardoso. imil4path: A semi-supervised interpretable approach for colorectal whole-slide images. *Cancers*, 14(10):2489, 2022.

Pedro C Neto, Diana Montezuma, Sara P Oliveira, Domingos Oliveira, João Fraga, Ana Monteiro, João Monteiro, Liliana Ribeiro, Sofia Gonçalves, Stefan Reinhard, et al. An interpretable machine learning system for colorectal cancer diagnosis from pathology slides. *NPJ precision oncology*, 8 (1):56, 2024.

Till Nicke, Jan Raphael Schäfer, Henning Höfener, Friedrich Feuerhake, Dorit Merhof, Fabian Kießling, and Johannes Lotz. Tissue concepts: Supervised foundation models in computational pathology. *Computers in Biology and Medicine*, 186:109621, 2025.

Sara P Oliveira, Pedro C Neto, Joao Fraga, Diana Montezuma, Ana Monteiro, João Monteiro, Liliana Ribeiro, Sofia Gonçalves, Isabel M Pinto, and Jaime S Cardoso. Cad systems for colorectal cancer from wsi are still not ready for clinical acceptance. *Scientific Reports*, 11(1):14358, 2021.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, Adam Lerer, et al. Automatic differentiation in pytorch. 2017.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. *Advances in neural information processing systems*, 34:20596–20607, 2021.

Geoff Pleiss, Tianyi Zhang, Ethan Elenberg, and Kilian Q Weinberger. Identifying mislabeled data using the area under the margin ranking. *Advances in Neural Information Processing Systems*, 33:17044–17056, 2020.

Benedikt Roth, Valentin Koch, Sophia J Wagner, Julia A Schnabel, Carsten Marr, and Tingying Peng. Low-resource finetuning of foundation models beats state-of-the-art in histopathology. In *2024 IEEE International Symposium on Biomedical Imaging (ISBI)*, pp. 1–5. IEEE, 2024.

Marin Scalbert, Charlie Saillard, Thomas Peeters, Liam Gonzalez, Dasha Valter, Felipe Llinares-López, Zelda E. Mariet, and Rodolphe Jenatton. H-optimus-1: A foundation model for computational histopathology. In *Proceedings of the American Association for Cancer Research Annual Meeting 2026; Part 2 (Late-Breaking, Clinical Trial, and Invited Abstracts)*, volume 86, pp. LB174. AACR, 2026. doi: 10.1158/1538-7445.AM2026-LB174.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017.

George Shaikovski, Adam Casson, Kristen Severson, Eric Zimmermann, Yi Kan Wang, Jeremy D Kunz, Juan A Retamero, Gerard Oakley, David Klimstra, Christopher Kanan, et al. Prism: A multi-modal generative foundation model for slide-level histopathology. *arXiv preprint arXiv:2405.10254*, 2024.

Julio Silva-Rodríguez, Adrián Colomer, María A Sales, Rafael Molina, and Valery Naranjo. Going deeper through the gleason scoring scale: An automatic end-to-end system for histology prostate grading and cribriform pattern detection. *Computer methods and programs in biomedicine*, 195: 105637, 2020.

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

Haoru Tan, Sitong Wu, Wei Huang, Shizhen Zhao, and Xiaojuan Qi. Data pruning by information maximization. *arXiv preprint arXiv:2506.01701*, 2025.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. *arXiv preprint arXiv:1812.05159*, 2018.

Gregory Verghese, Jochen K Lennerz, Danny Ruta, Wen Ng, Selvam Thavaraj, Kalliopi P Siziopikou, Threnesan Naidoo, Swapnil Rane, Roberto Salgado, Sarah E Pinder, et al. Computational pathology in cancer diagnosis, prognosis, and prediction–present day and prospects. *The Journal of pathology*, 260(5):551–563, 2023.

Xiaobo Xia, Jiale Liu, Jun Yu, Xu Shen, Bo Han, and Tongliang Liu. Moderate coreset: A universal method of data selection for real-world data-efficient deep learning. In *The Eleventh International Conference on Learning Representations*, 2022.

Hanwen Xu, Naoto Usuyama, Jaspreet Bagga, Sheng Zhang, Rajesh Rao, Tristan Naumann, Cliff Wong, Zelalem Gero, Javier González, Yu Gu, et al. A whole-slide foundation model for digital pathology from real-world data. *Nature*, 630(8015):181–188, 2024.

Haizhong Zheng, Rui Liu, Fan Lai, and Atul Prakash. Coverage-centric coreset selection for high pruning rates. *arXiv preprint arXiv:2210.15809*, 2022.

Haizhong Zheng, Elisa Tsai, Yifu Lu, Jiachen Sun, Brian Bartoldson, Bhavya Kailkhura, and Atul Prakash. Elfs: Label-free coreset selection with proxy training dynamics. In *International Conference on Learning Representations*, volume 2025, pp. 63391–63410, 2025.

Eric Zimmermann, Eugene Vorontsov, Julian Viret, Adam Casson, Michal Zelechowski, George Shaikovski, Neil Tenenholtz, James Hall, David Klimstra, Razik Yousfi, et al. Virchow2: Scaling self-supervised mixed magnification models in pathology. *arXiv preprint arXiv:2408.00738*, 2024.

A PSEUDO-CODE

Algorithm 1 Coreset Selection by GC Maximization via Mean Nearest-Neighbor Distance

Inputs: $\mathcal{S} = \{x_i\}_{i=1}^N$: unlabeled dataset; $f(\cdot)$: foundation model; k: target coreset size.
Outputs: Coreset $\mathcal{S}' \subset \mathcal{S}$ with $|\mathcal{S}'| = k$, and its coverage $\mathrm{GC}(\mathcal{S}')$.

For each $x_i \in \mathcal{S}$, $\mathbf{e}_i \leftarrow f(x_i)$; $\mathbf{E} \leftarrow \{\mathbf{e}_i\}_{i=1}^N \subset \mathbb{R}^D$ ▷ Extract D-dimensional embeddings
$r_{\max} \leftarrow \max_{i<j} \|\mathbf{e}_i - \mathbf{e}_j\|_1$ ▷ Max pairwise distance; ensures $\mathrm{sim} \geq 0$ (Appendix D)
$\mathbf{d} \leftarrow (r_{\max}, \ldots, r_{\max}) \in \mathbb{R}^N$; $\bar{d} \leftarrow r_{\max}$ ▷ Invariant: $\mathbf{d} = \mathbf{d}(\mathcal{S}')$, $\bar{d} = \bar{d}(\mathcal{S}')$; $d_i(\varnothing) = r_{\max}$
$(\mathcal{S}', \mathcal{R}) \leftarrow (\varnothing, \mathcal{S})$
for each $x_j \in \mathcal{R}$ **do**
    $g_j \leftarrow \frac{1}{N} \sum_{i=1}^N \big(d_i - \|\mathbf{e}_i - \mathbf{e}_j\|_1\big)_+$ ▷ Initial reduction in $\bar{d}$ from adding x_j
end for
$\mathcal{Q} \leftarrow$ max-heap over $\{(g_j, x_j)\}_{x_j \in \mathcal{R}}$ ▷ Lazy-greedy priority queue
while $|\mathcal{S}'| < k$ **do**
    repeat
        $(g_j, x_j) \leftarrow \mathcal{Q}.\textsc{PopMax}()$
        $g_j \leftarrow \frac{1}{N} \sum_{i=1}^N \big(d_i - \|\mathbf{e}_i - \mathbf{e}_j\|_1\big)_+$ ▷ Recompute exact gain at current $\mathbf{d}$
        valid $\leftarrow \big(\mathcal{Q} = \varnothing\big) \vee \big(g_j \geq \mathcal{Q}.\textsc{PeekKey}()\big)$
        if $\neg$ valid **then**
            $\mathcal{Q}.\textsc{Push}(g_j, x_j)$
        end if
    until valid
    $d_i \leftarrow \min\big(d_i, \|\mathbf{e}_i - \mathbf{e}_j\|_1\big), \ \forall x_i \in \mathcal{S}$ ▷ Eq. 8; incremental coverage update
    $\bar{d} \leftarrow \bar{d} - g_j$; $(\mathcal{S}', \mathcal{R}) \leftarrow (\mathcal{S}' \cup \{x_j\}, \mathcal{R} \setminus \{x_j\})$
end while
return $\mathcal{S}'$, $\mathrm{GC}(\mathcal{S}') = 1 - \bar{d}/r_{\max}$ ▷ Eq. 7

B NOTATIONAL CONVENTION

Table 3: Comprehensive overview of the notational convention.

Notation	Description
$\mathcal{S}$	The unlabeled training set, with $\lvert\mathcal{S}\rvert = N$.
$x_i \in \mathcal{S}$	A training sample.
N	The number of samples in $\mathcal{S}$.
$f(\cdot)$	The pre-trained FM used as a frozen feature extractor.
$\mathbf{E} = \{\mathbf{e}_i\}_{i=1}^N$	The set of embeddings of the full training set.
$\mathbf{e}_i = f(x_i) \in \mathbb{R}^D$	The embedding of sample x_i.
D	The embedding dimension.
$\mathcal{S}' \subset \mathcal{S}$	The coreset selected by GCcore, with $\lvert\mathcal{S}'\rvert = k$.
k	The annotation budget and the target coreset size.
$\mathcal{T} \subseteq \mathcal{S}$	An arbitrary subset of $\mathcal{S}$; the argument of the coverage objective.
r	A coverage radius.
$d_i(\mathcal{T})$	The distance from sample x_i to its nearest representative in $\mathcal{T}$, see Eq. (1); by convention $d_i(\varnothing) = r_{\max}$.
$\mathbf{d}(\mathcal{T}) \in \mathbb{R}^N$	The vector of nearest-coreset distances, $\big(d_1(\mathcal{T}), \ldots, d_N(\mathcal{T})\big)^\top$.
e_{im}	The m-th component of $\mathbf{e}_i$.
$p(r; \mathcal{T})$	The coverage at radius r: the fraction of samples within r of $\mathcal{T}$, see Eq. (2).
$r_{\max}$	The maximum pairwise distance over $\mathcal{S}$, $r_{\max} = \max_{i<j} \lVert\mathbf{e}_i - \mathbf{e}_j\rVert_1$.
$\mathrm{GC}(\mathcal{T})$	The global coverage of $\mathcal{T}$: the area under the coverage–radius curve, see Eq. (3). $\mathrm{GC} \in [0, 1]$, with $\mathrm{GC}(\varnothing) = 0$ and $\mathrm{GC}(\mathcal{S}) = 1$.
$\mathcal{S}^\star \subset \mathcal{S}$	The optimal coreset, $\mathcal{S}^\star = \arg\max_{\lvert\mathcal{T}\rvert=k} \mathrm{GC}(\mathcal{T})$, see Eq. (4).
r_{steps}	The number of sampled radii required by a direct numerical integration of $p(r; \mathcal{T})$.
$\bar{d}(\mathcal{T})$	The mean nearest-coreset distance, $\bar{d}(\mathcal{T}) = \frac{1}{N}\sum_{i=1}^N d_i(\mathcal{T})$.
$g_j(\mathcal{S}')$	The marginal gain of candidate x_j: the reduction in $\bar{d}$ from adding x_j to $\mathcal{S}'$, see Eq. (9).
$\mathcal{R}$	The pool of candidates not yet selected at the current iteration.
$\mathcal{Q}$	The max-heap of candidate gains used by the lazy greedy solver.
$\mathrm{sim}(x_i, x_j)$	The nonnegative similarity $r_{\max} - \lVert\mathbf{e}_i - \mathbf{e}_j\rVert_1 \geq 0$.
$c_i(\mathcal{T})$	The per-sample coverage $\max_{x_j \in \mathcal{T}} \mathrm{sim}(x_i, x_j)$, with $c_i(\varnothing) = 0$.
$F(\mathcal{T})$	The total coverage $\sum_{i=1}^N c_i(\mathcal{T}) = N r_{\max} \mathrm{GC}(\mathcal{T})$.
t	An intermediate cardinality, $t \leq k$; the first t samples selected.
C	The number of classes defined by the dataset label.
D'	The reduced embedding dimension after principal component analysis retaining 99% of the variance.
$g \times g$	The square patch-token grid produced by the final transformer block.
R	The working resolution used for the segmentation task, i.e. the side length of the square grid at which predicted logits and ground-truth masks are aligned.
c	A class index, $c \in \{1, \ldots, C\}$.
n_c^+	The number of positive samples of class c within the selected coreset $\mathcal{S}'$.
n_c^-	The number of negative samples of class c within the selected coreset $\mathcal{S}'$.
λ_c	The weight applied to the positive term of class c in the multi-label binary cross-entropy loss, $\lambda_c = \min(n_c^-/n_c^+, \lambda_{\max})$.
$\lambda_{\max}$	The cap on the positive-class weight, $\lambda_{\max} = 10$.
$\lVert\cdot\rVert_1$	The ℓ_1 (Manhattan) norm, used throughout unless stated otherwise.
$\lvert\cdot\rvert$	The cardinality of a set.
$\mathbf{1}\{\cdot\}$	The indicator function.
$(\cdot)_+$	The positive part, $(a)_+ = \max\{0, a\}$.

C HYPERPARAMETERS IN EXISTING LABEL-FREE SELECTION METHODS

Table 4: Hyperparameters of coreset selection methods that are label-free by design or that can be adapted into a label-free variant. For every method that operates on embeddings, we regard the feature-extraction step as label-free, since applying a pathology FM to obtain embeddings requires no annotations. Hyperparameter symbols are local to each method and follow the original publications, except where a symbol would collide with the notation of this paper (Table 3).

Method	**Label-free by design**	**Label-free adaptation**	**Hyperparameters of the label-free method**	**Reported values and how they were obtained**
Random	Yes	—	s: random seed.	—
K-center (Sener & Savarese, 2017)	Yes	—	s: random seed.	—
K-means (MacQueen, 1967)	Yes	—	s: random seed; M: number of clusters.	—
Prototype (Sorscher et al., 2022)	Yes	—	M: number of clusters; mode $\in \{$*abundant*, *scarce*$\}$: pruning direction. Hard (least prototypical) samples are preferred when the retained set is large (*abundant*), and easy (prototypical) samples when it is small (*scarce*).	$M = 1{,}000$ (ImageNet). The crossover point between the two modes is derived by solving saddle-point equations that depend on the student–teacher overlap and on the margin distribution along the teacher, both defined with respect to a labeled decision boundary. It is therefore not identifiable without labels, so we treat the pruning direction as a hyperparameter.
Moderate (Xia et al., 2022)	No	Class centers are replaced by K-means centroids.	M: number of clusters.	—
CCS (Zheng et al., 2022)	No	The label-dependent difficulty score is replaced by the prototype score.	M: number of clusters used to compute the prototype score; β: hard-cutoff rate, i.e. the fraction of the highest-difficulty samples discarded before stratification; B: number of strata; s: random seed for the uniform draws within each stratum.	$B = 50$ for all datasets and pruning rates (CIFAR-10); β obtained by grid search with a step size of 0.1, separately for every dataset and every pruning rate.
D^2-Pruning (Maharana et al., 2024)	No	*Variant 1*: all nodes are initialized with a uniform difficulty score.	k_{nn}: number of nearest neighbors in the graph; γ_f: forward message weight; γ_r: reverse message weight.	γ_f fixed to 1.0; $k_{\text{nn}} \in \{1, 5, 10, 15\}$ and $\gamma_r \in \{0, 0.1, \ldots, 1.0\}$ were swept on CIFAR-10 and CIFAR-100, and the resulting trends used to select three configurations for ImageNet-1K.

Continued on the next page

Table 4 (continued)

Method	Label-free by design	Label-free adaptation	Hyperparameters of the label-free method	Reported values and how they were obtained
		Variant 2: nodes are initialized with the prototype score.	M: number of clusters used to compute the prototype score; k_{nn}: number of nearest neighbors in the graph; γ_f: forward message weight; γ_r: reverse message weight.	For CIFAR-10 at a coreset ratio of 0.3: $k_{\text{nn}} = 10$, $\gamma_r = 0.9$.
ELFS (Zheng et al., 2025)	Yes	*Variant 1*: identical to the original method; the difficulty score is the AUM score obtained from pseudo-labels.	$\tilde{C}$: number of pseudo-classes used in deep clustering; training configuration of the model trained on pseudo-labels to compute AUM scores (epochs, batch size, learning rate, optimizer, weight decay); β: hard-cutoff rate, i.e. the fraction of the highest-difficulty samples discarded before stratification.	$\tilde{C}$ is set to the number of classes C in the dataset; 200 epochs, batch size 512, AdamW, learning rate 10^{-4}, weight decay 10^{-4} (CIFAR-10, CIFAR-100, STL-10, ImageNet); β obtained by grid search with a step size of 0.1 on a pseudo-label validation split, separately for every dataset and every pruning rate.
		Variant 2: the AUM score is replaced by the prototype score.	M: number of clusters used to compute the prototype score; β: hard-cutoff rate.	—
InfoMax (Tan et al., 2025)	Yes	*Variant 1*: the prototype score in *abundant* mode (hard samples carry high intra-sample information).	α: pairwise weight balancing the first- and second-order terms in Eq. (2) of the InfoMax paper; k_{nn}: sparsity rate, i.e. the neighborhood size of each sample in the similarity matrix; N_{part}: size of each partition subset; n_{iter}: number of iterations.	$N_{\text{part}} \geq 10^6$ samples (CC3M and CC12M); $k_{\text{nn}} = 5$, $\alpha = 0.3$, $n_{\text{iter}} = 20$ (CC12M).
		Variant 2: the prototype score in *scarce* mode (easy samples carry high intra-sample information).		
ZCore (Griffin et al., 2026)	Yes	—	s: random seed (selection of the sampled subspaces and initialization of the per-sample importance score); D_{sub}: dimensionality of the sampled subspaces; α: number of nearest neighbors penalized as redundant; β: exponent of the distance penalty; n_{iter}: number of sampling iterations.	$D_{\text{sub}} = 2$, $\alpha = 1{,}000$, $\beta = 4$, $n_{\text{iter}} \approx 10^6$ (CIFAR-100).

D PROOF

In this section, we show that the objective in Eq. 4 is monotone submodular, so that the greedy algorithm attains a $(1 - 1/e)$ approximation of the optimal coreset (Nemhauser et al., 1978). This provides a dataset-independent, a priori lower bound on the coverage of the selected coreset. To the best of our knowledge, the baselines generally do not provide a proof of such a lower-bound guarantee.

D.1 GC IS MONOTONE AND SUBMODULAR

For each ordered pair (x_i, x_j), define the nonnegative similarity $\mathrm{sim}(x_i, x_j) = r_{\max} - \|\mathbf{e}_i - \mathbf{e}_j\|_1 \geq 0$. For $\mathcal{T} \subseteq \mathcal{S}$, let the per-sample coverage be $c_i(\mathcal{T}) = \max_{x_j \in \mathcal{T}} \mathrm{sim}(x_i, x_j)$ with $c_i(\varnothing) = 0$, consistent with $d_i(\varnothing) = r_{\max}$, and the total coverage $F(\mathcal{T}) = \sum_{i=1}^{N} c_i(\mathcal{T})$. Since $c_i(\mathcal{T}) = r_{\max} - d_i(\mathcal{T})$, we have $F(\mathcal{T}) = N r_{\max}\big(1 - \bar{d}(\mathcal{T})/r_{\max}\big)$, and by Eq. 7,

$$\mathrm{GC}(\mathcal{T}) \;=\; \frac{F(\mathcal{T})}{N\, r_{\max}}. \tag{10}$$

As $N r_{\max}$ is a positive constant independent of $\mathcal{T}$, GC and F share the same monotonicity, submodularity, and maximizers.

Lemma 1. *F, and hence* GC*, is normalized, monotone, and submodular.*

Proof. $F(\varnothing) = 0$ gives normalization. For $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{S}$, each c_i is a maximum over a larger set, so $c_i(\mathcal{A}) \leq c_i(\mathcal{B})$ and $F(\mathcal{A}) \leq F(\mathcal{B})$, giving monotonicity. For submodularity, the per-sample marginal gain of any $x \in \mathcal{S} \setminus \mathcal{B}$ is $c_i(\mathcal{T} \cup \{x\}) - c_i(\mathcal{T}) = \big(\mathrm{sim}(x_i, x) - c_i(\mathcal{T})\big)_+$. Since $c_i(\mathcal{A}) \leq c_i(\mathcal{B})$ and $a \mapsto (a)_+$ is nondecreasing, $\big(\mathrm{sim}(x_i, x) - c_i(\mathcal{A})\big)_+ \geq \big(\mathrm{sim}(x_i, x) - c_i(\mathcal{B})\big)_+$; summing over i yields $F(\mathcal{A} \cup \{x\}) - F(\mathcal{A}) \geq F(\mathcal{B} \cup \{x\}) - F(\mathcal{B})$. All three properties transfer to GC under scaling by the positive constant $1/(N r_{\max})$. □

D.2 APPROXIMATION GUARANTEE

By (Nemhauser et al., 1978), for a normalized monotone submodular set function h, the greedy selection $\mathcal{T}_h$ of size k satisfies $h(\mathcal{T}_h) \geq (1 - 1/e) \max_{|\mathcal{T}| \leq k} h(\mathcal{T})$. Algorithm 1 is exactly this procedure applied to GC, so Lemma 1 gives

$$\mathrm{GC}(\mathcal{S}') \;\geq\; \big(1 - \tfrac{1}{e}\big)\, \mathrm{GC}(\mathcal{S}^\star) \;\approx\; 0.632\, \mathrm{GC}(\mathcal{S}^\star), \tag{11}$$

where $\mathcal{S}'$ is the coreset returned by Algorithm 1 and $\mathcal{S}^\star$ is the optimal coreset of Eq. 4. The selected coreset thus attains at least $\approx 63.2\%$ of the optimal coverage, independently of the dataset or embedding geometry. As the guarantee holds at every cardinality, each intermediate coreset ($t \leq k$) is likewise $(1 - 1/e)$-optimal, so a single run certifies the selection at every budget. Pathologists can therefore annotate the selected samples in the order they were chosen and stop whenever the budget is exhausted, with the guarantee holding at every stopping point.

E DATASETS

E.1 SUMMARY

Table 5: Overview of the datasets used for evaluation. Sample counts are given as W (whole-slide images), T (tiles) or P (patients).

Task	Dataset	Tissue	Label	Dataset size	Train set	Test set
WSI classification	PANDA	Prostate	ISUP score	10,616 W	8,492 W	2,124 W
	IMP-CRS	Colon	non-neoplastic, low-grade, high-grade	5,333 W	4,267 W	1,066 W
	TCGA-BRCA	Breast	IDC, ILC	949 P	759 P	190 P
Tile classification	WSSS4LUAD	Lung	tumor epithelium, tumor-associated stroma, normal tissue	10,091 T	8,054 T	2,037 T
	SICAP	Prostate	NC, GG3, GG4, GG5	12,081 T	9,959 T	2,122 T
	CRC-100k	Colon	ADI, BACK, DEB, LYM, MUC, MUS, NORM, STR, TUM	107,180 T	100,000 T	7,180 T
	DeepHP	Stomach	HPP, HPN	394,926 T	315,941 T	78,985 T
Segmentation	BEETLE	Breast	invasive epithelium, non-invasive epithelium, necrosis, other	400,941 T	340,200 T	60,741 T
	panCK	10 tissue types	epithelium, other	26,509 T	21,912 T	4,597 T
	CAMELYON16	Breast	metastatic tissue, normal tissue, background	479,772 T	365,112 T	114,660 T

Table 5 summarizes the 10 datasets used in our experiments. Coresets are selected from the full training set of each dataset, which ranges in size from 759 patients to 365,112 tiles. Because a fixed retention rate translates into very different annotation workloads across this range, we define a separate set of four low retention rates per dataset, chosen so that every coreset comprises tens to several hundred samples, an amount that a pathologist could realistically annotate in clinical practice.

For WSI classification, we retain $\{0.5, 1, 2, 5\}\%$ of PANDA, $\{1, 2, 5, 10\}\%$ of IMP-CRS and $\{5, 10, 15, 20\}\%$ of TCGA-BRCA. For tile classification, we retain $\{0.5, 1, 2, 5\}\%$ of WSSS4LUAD, $\{0.5, 1, 2, 5\}\%$ of SICAP, $\{0.02, 0.05, 0.1, 0.2\}\%$ of CRC-100k and $\{0.01, 0.05, 0.1, 0.2\}\%$ of DeepHP. For segmentation, we retain $\{0.02, 0.05, 0.1, 0.2\}\%$ of BEETLE, $\{0.2, 0.5, 1, 2\}\%$ of panCK and $\{0.02, 0.05, 0.1, 0.2\}\%$ of CAMELYON16.

Training and test sets are split at the patient level when patient identifiers are available and at the slide level otherwise. When an official split is provided, we adopt the released split.

E.2 DESCRIPTIONS

PANDA (Bulten et al., 2022) consists of 10,616 H&E formalin-fixed paraffin-embedded (FFPE) diagnostic histopathology WSIs of core-needle biopsies of prostate cancer from the Radboud University Medical Center and the Karolinska Institute, digitized at 20× magnification. Each slide is assigned an International Society of Urological Pathology (ISUP) score defining the prostate cancer grade (six-class grading task). We partitioned PANDA into label-stratified training and test sets at a ratio of 80:20, yielding 8,492 WSIs (grade 0, 2,313; grade 1, 2,133; grade 2, 1,074; grade 3, 994; grade 4, 999; grade 5, 979) in the full training set and 2,124 WSIs (grade 0, 579; grade 1, 533; grade 2, 269; grade 3, 248; grade 4, 250; grade 5, 245) in the test set.

IMP-CRS (Oliveira et al., 2021; Neto et al., 2022; 2024) contains 5,333 colorectal biopsy and polypectomy slides, retrieved from the data archive of IMP Diagnostics laboratory, Portugal, digi-

tized at 40× magnification. Each case is assigned to one of three classes: non-neoplastic; low-grade lesions, comprising conventional adenomas with low-grade dysplasia; and high-grade lesions, comprising conventional adenomas with high-grade dysplasia, intramucosal carcinomas and invasive adenocarcinomas. We performed a label-stratified 80:20 split of the dataset into training and test subsets, yielding 4,267 WSIs (678:2,278:1,311) in the full training set and 1,066 WSIs (169:569:328) in the test set, following the non-neoplastic:low-grade:high-grade class order.

TCGA-BRCA comprises H&E FFPE diagnostic histopathology WSIs of breast invasive carcinoma from The Cancer Genome Atlas (TCGA), primarily scanned at 20× magnification. Slides are assigned to one of two classes: invasive ductal carcinoma (IDC) and invasive lobular carcinoma (ILC). After excluding patients with missing slides or other cancer types, we retained WSIs from 949 patients (IDC, 758; ILC, 191), of whom 892 were represented by a single slide and 57 by multiple. We partitioned the cohort at the patient level into label-stratified training and test subsets at a ratio of 80:20, yielding 759 patients (IDC, 606; ILC, 153) in the full training set and 190 patients (IDC, 152; ILC, 38) in the test set. For each patient, we used the mean of all available slide embeddings as the patient-level embedding.

WSSS4LUAD (Han et al., 2022) consists of lung adenocarcinoma (LUAD) image tiles from Guangdong Provincial People's Hospital (GDPH) and TCGA, all scanned at 40× magnification. Each tile carries a three-element binary label vector indicating the presence of tumor epithelium, tumor-associated stroma and normal tissue. As tile-level labels are not released for the official validation and test sets, we constructed our dataset from the original training set alone, comprising 10,091 tiles from 63 WSIs, with tile dimensions ranging from 150×150 to 300×300 pixels. We partitioned the data at the WSI level and stratified by center of origin; the split was chosen to yield an approximate 80:20 ratio of tiles, giving 8,054 tiles from 53 WSIs (GDPH, 41; TCGA, 12) in the full training set and 2,037 tiles from 10 WSIs (GDPH, 8; TCGA, 2) in the test set.

SICAP (Silva-Rodríguez et al., 2020) comprises 512×512 pixel tiles extracted from 155 WSIs of prostate core-needle biopsies, digitized at 10× magnification. Each tile is annotated with one of four Gleason grades, non-cancerous (NC), Gleason grade 3 (GG3), Gleason grade 4 (GG4), and Gleason grade 5 (GG5), yielding a four-class grading task. We adopt the official split, which partitions the dataset into 9,959 tiles from 124 WSIs for training and 2,122 tiles from 31 WSIs for testing.

CRC-100k (Kather et al., 2018) consists of 224×224 pixel tiles extracted from H&E-stained histological images of human colorectal cancer (CRC) and normal tissue, digitized at 20× magnification. Each tile is assigned to one of nine classes: adipose tissue (ADI), background (BACK), debris (DEB), lymphocytes (LYM), mucus (MUC), smooth muscle (MUS), normal colon mucosa (NORM), cancer-associated stroma (STR) and colorectal adenocarcinoma epithelium (TUM). We adopted the official splits, comprising 100,000 tiles for training and 7,180 tiles for testing.

DeepHP (Goncalves et al., 2022; Gonçalves e Gonçalves et al., 2023) comprises 394,926 tiles of 224×224 pixels, extracted from gastric biopsies digitized at 20× magnification, of which 111,005 are labeled Helicobacter pylori-positive (HPP) and 283,921 Helicobacter pylori-negative (HPN). We partitioned the tiles into label-stratified training and test subsets at a ratio of 80:20, yielding 315,941 tiles (HPP, 88,804; HPN, 227,137) in the full training set and 78,985 tiles (HPP, 22,201; HPN, 56,784) in the test set.

BEETLE (Lems et al., 2025) is a multicenter, multiscanner dataset of breast cancer biopsies and surgical resections, spanning both major histological subtypes of invasive breast cancer, all molecular subtypes and all histological grades, provided at 20× magnification. From the official development set, we selected 396 WSIs from patients with pixel-level annotations of four classes: invasive epithelium, non-invasive epithelium, necrosis and other. We partitioned these WSIs at the patient level and stratified by center of origin. From each WSI we then extracted non-overlapping 224×224 pixel tiles, retaining only those in which more than 50% of pixels were annotated. This yielded 340,200 tiles from 317 WSIs (Netherlands Cancer Institute (NKI), 90; Radboud University Medical Center (RUMC), 169; Santa Chiara Hospital (SCH), 44; TCGA, 14) in the full training set and 60,741 tiles from 79 WSIs (NKI, 23; RUMC, 42; SCH, 11; TCGA, 3) in the test set.

panCK (Komura et al., 2023) consists of 26,509 tiles of 984×984 pixels at $40\times$ magnification, collected from FFPE tissue of 407 patients and covering 10 tissue types and 16 tumor types. Each tile is provided with pixel-level annotations of two classes, epithelium and other, derived from immunofluorescence restaining. We retained the official training split and merged the official validation and test sets into a single test set, giving 21,912 tiles in the full training set and 4,597 tiles in the test set.

CAMELYON16 (Ehteshami Bejnordi et al., 2017) contains H&E-stained WSIs of sentinel lymph node sections from breast cancer patients at Radboud University Medical Center and University Medical Center Utrecht, digitized at $40\times$ magnification. We constructed our dataset from the 158 WSIs containing metastasis, each of which carries pixel-level annotations of three classes: background, normal tissue and metastatic tissue. We partitioned these at the slide level at a ratio of 80:20, then extracted non-overlapping tiles of 512×512 pixels at $20\times$ magnification, retaining only those with more than 50% non-background area. This yielded 365,112 tiles from 126 WSIs in the full training set and 114,660 tiles from 32 WSIs in the test set.

F Implementation details

F.1 Hyperparameter settings of baselines

Why natural-image hyperparameters are not used directly. All baselines except Random and K-center expose hyperparameters whose default values or robust ranges were determined on natural-image benchmarks. As illustrated in Section 1, neither transfers reliably to histopathology. First, a range reported as robust on natural images can produce substantial variation on histopathology data: InfoMax (Tan et al., 2025) was reported to be robust to $\alpha \in [0.3, 4]$ on CC12M, yet on PANDA at a 2% retention rate, $\alpha = 0.3, 1, 2, 3, 4$ yields QWK values of 0.60, 0.58, 0.70, 0.66 and 0.69, respectively. Second, the prescribed default can be among the worst choices: ELFS (Zheng et al., 2025) sets the number of pseudo-classes $\tilde{C}$ to the number of classes C, but on TCGA-BRCA ($C = 2$) at a 5% retention rate, $\tilde{C} = 2, 4, 10, 15, 20$ yields balanced accuracies of 0.71, 0.85, 0.84, 0.83 and 0.83, with the default performing worst. Third, the default may be undefined: for segmentation on BEETLE, tiles do not carry a single class label, and $\tilde{C} = 4, 10, 15, 20, 25$ at a 0.02% retention rate yields test weighted Dice values of 0.81, 0.84, 0.70, 0.84 and 0.78, with no monotonic trend.

Sweep protocol. Tuning these hyperparameters for each cohort would require additional labels, which conflicts with the purpose of coreset selection; we therefore assume that no labels are available for tuning. For each baseline, rather than selecting the best setting, we sweep a characteristic hyperparameter over five values and report the mean test performance across the five resulting coresets, together with its SD. The five values are sampled within the range reported to be robust on natural-image benchmarks, where such a range is given, and otherwise around the reported default value, including the default itself. All other hyperparameters are kept at their default settings. For Random, K-center, and ZCore, we sweep the random seed. The mean thus indicates the expected performance of a baseline on a new cohort without labels for tuning, while the SD of the test performance across the five settings indicates the sensitivity to the swept hyperparameter. GCcore is deterministic and hyperparameter-free and therefore has no SD.

Per-method settings. Random, K-center (Sener & Savarese, 2017) and ZCore (Griffin et al., 2026) are swept over the random seeds $\{17, 83, 146, 271, 394\}$. K-means, Proto-Hard, Proto-Easy, Moderate and ELFS-Proto are swept over the number of clusters M, and ELFS-Pseudo over the number of pseudo-classes $\tilde{C}$. Following the original works, the default is the number of classes C of the dataset, so the five values are dataset-specific. The values for each dataset are listed in Table 6. CCS (Zheng et al., 2022) is swept over the hard-cutoff rate β. The original work selects β by grid search for each dataset and pruning rate, which requires labels; the reported optimal values lie between 0 and 0.5 and increase with the pruning rate, ranging from 0.2 to 0.5 at pruning rates of 90% and above. Since a majority of our experiments operate at pruning rates above 90%, we use $\beta \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$. D^2-Uniform and D^2-Proto (Maharana et al., 2024) are swept over the number of nearest neighbors $k_{\text{nn}} \in \{1, 5, 10, 15, 20\}$. InfoMax-Hard and InfoMax-Easy (Tan et al., 2025) are swept over $\alpha \in \{0.3, 1, 2, 3, 4\}$, sampled within the range reported as robust.

F.2 Software and hardware specifications

We implemented our method in PyTorch (Paszke et al., 2017); experiments were conducted across two servers, one with two NVIDIA RTX A6000 GPUs (48 GB each) and one with three NVIDIA RTX PRO 6000 Blackwell GPUs (96 GB each). FM embeddings are extracted once and cached, so every selection run and every downstream head reuses the same features.

For all model training, only the head parameters are optimized, while the encoder remains frozen throughout. All heads are trained using AdamW with a learning rate of 10^{-3} and weight decay of 10^{-4}, with batch sizes of 256 for classification heads and 32 for segmentation heads. When the subset size k is smaller than the batch size, minibatches are sampled from the subset with replacement, ensuring that each gradient step uses a full batch. Following (Zheng et al., 2022; Maharana et al., 2024; Tan et al., 2025), the number of epochs is set per run so that the number of iterations on the selected subset matches that of the corresponding full-set run. Identical settings are used across all servers, selection strategies and retention rates. To mitigate the influence of the training procedure on test performance, the task-specific head is trained five times on each coreset, whether selected

Table 6: Swept hyperparameter and its five values for each baseline. (a) Hyperparameters swept for each baseline. (b) Dataset-specific values of the number of clusters M and the number of pseudo-classes $\tilde{C}$.

(a) Swept hyperparameters

Baseline	**Swept hyperparameter**	**Values**
Random, K-center, ZCore	Random seed s	$\{17, 83, 146, 271, 394\}$
K-means, Proto-Hard, Proto-Easy, Moderate, ELFS-Proto	Number of clusters M	Dataset-specific, see (b)
ELFS-Pseudo	Number of pseudo-classes $\tilde{C}$	Dataset-specific, see (b)
CCS	Hard-cutoff rate β	$\{0.1, 0.2, 0.3, 0.4, 0.5\}$
D^2-Uniform, D^2-Proto	Number of nearest neighbors k_{nn}	$\{1, 5, 10, 15, 20\}$
InfoMax-Hard, InfoMax-Easy	Pairwise weight α	$\{0.3, 1, 2, 3, 4\}$

(b) Dataset-specific values of M and $\tilde{C}$

Task	**Dataset**	$\boldsymbol{C}$	**Values of M and $\tilde{C}$**
WSI classification	PANDA	6	$\{6, 10, 15, 20, 25\}$
	IMP-CRS	3	$\{3, 4, 10, 15, 20\}$
	TCGA-BRCA	2	$\{2, 4, 10, 15, 20\}$
Tile classification	WSSS4LUAD	3	$\{3, 4, 10, 15, 20\}$
	SICAP	4	$\{4, 10, 15, 20, 25\}$
	CRC-100k	9	$\{9, 10, 15, 20, 25\}$
	DeepHP	2	$\{2, 4, 10, 15, 20\}$
Segmentation	BEETLE	4	$\{4, 10, 15, 20, 25\}$
	panCK	2	$\{2, 4, 10, 15, 20\}$
	CAMELYON16	3	$\{3, 4, 10, 15, 20\}$

by GCcore or by a single configuration (hyperparameter value) of a baseline, using five different training seeds that govern head initialization and minibatch ordering. The mean test performance across these five runs is reported.

F.3 Segmentation head architecture

The segmentation head operates on the spatial token grid of the frozen encoder and is the only trainable component of the model. To obtain this grid, embeddings are extracted from the final transformer block of the frozen tile encoder, and the prefix tokens ([CLS] and any register tokens) are discarded to retain only the patch tokens carrying spatial information. The retained tokens are then reshaped from a flattened sequence into a two-dimensional grid, preserving spatial correspondence with the original image locations, so that each token encodes features from a distinct non-overlapping region of the input. Throughout, let D denote the encoder embedding dimensionality, $g \times g$ the token grid, and R the working resolution at which the loss and the evaluation metrics are computed; the input to the head is therefore a feature map of size $D \times g \times g$.

The head begins with a 1×1 convolutional projection that preserves the channel width at D, followed by batch normalization and a GELU activation. Four successive transposed-convolutional blocks (kernel size 4, stride 2, padding 1) then upsample the spatial resolution by a factor of two at each stage while halving the channel dimensionality, each block again followed by batch normalization and GELU. A final 1×1 convolution projects the resulting feature maps onto the C segmentation classes, yielding pixel-wise logits at $16\times$ the token-grid resolution. For H-optimus-1, a ViT-g/14 at an input resolution of 224×224, the 257 output tokens of the final block reduce to 256 patch tokens once the [CLS] token is discarded, giving $g = 16$ and $D = 1536$; the decoder therefore takes the

feature map from 16×16 to 32×32, 64×64, 128×128 and finally 256×256, with the channel width reducing from 1536 to 768, 384, 192 and 96.

Because both the token grid size g and the working resolution R may vary across encoders and datasets, the logits are bilinearly resampled from $16g \times 16g$ to $R \times R$; this is the identity when $16g = R$ and a mild resampling otherwise. We set $R = 224$ for BEETLE, panCK and CAME-LYON16, so the 256×256 logit map is slightly reduced in each case. Masks are resized to $R \times R$ by nearest-neighbor interpolation, and patches are resized to the encoder's native input resolution without cropping, so that image and label remain in spatial correspondence.

G FURTHER RESULTS

G.1 GCCORE IS DETERMINISTIC

Fig. 4 shows that the performance of the 14 baseline methods is sensitive to their respective characteristic hyperparameters. Varying each hyperparameter across five values leads to a SD of up to 0.507 in test performance (Proto-Easy on IMP-CRS at 10% retention). Across the 40 dataset–retention-rate combinations (10 datasets $\times$ four low retention rates), Proto-Easy is the most sensitive method, with a mean range of 0.231 (SD = 0.099), while the least sensitive, Random, still has a mean range of 0.051 (SD = 0.021).

GCcore exceeds the mean of every baseline in 26 of the 40 combinations. Among the remaining 14 combinations, for 73% of the 55 baseline configurations whose mean performance exceeds that of GCcore, the advantage rests on only one or two settings. For example, at 0.5% retention on PANDA, K-means achieves QWK values of 0.723, 0.717, 0.738, 0.792, and 0.713 for 6, 10, 15, 20, and 25 clusters, respectively. The resulting mean of 0.737 is higher than the GCcore value of 0.733; however, K-means exceeds GCcore at only two of the five settings. Similarly, at 0.5% retention on SICAP, CCS achieves 0.580, 0.713, 0.626, 0.762, and 0.677 across its five settings. Although its mean of 0.672 is 0.004 higher than the GCcore value of 0.668, the range across settings is 0.182, approximately 48 times this difference.

Importantly, identifying the most favorable hyperparameter setting requires evaluating the candidate coresets on labeled data, the need for which coreset selection is intended to reduce. Consequently, the advantage of a baseline at its favorable settings cannot be reliably obtained in a practical, label-efficient scenario. In contrast, GCcore requires no hyperparameter tuning and achieves higher overall performance, making it directly applicable in practice.

G.2 GCCORE IS ROBUST TO THE CHOICE OF FEATURE EXTRACTOR

Figure 5 and Figure 6 disentangle two factors that are usually confounded in coreset evaluations: how well a selection method performs, and how much of that performance is inherited from the feature extractor on which it is built. The full-dataset panels establish the ceiling of the three slide encoders (PRISM (Shaikovski et al., 2024) 0.856 $>$ TITAN (Ding et al., 2025) 0.833 $>$ MADELEINE (Jaume et al., 2024) 0.748) and the three tile encoders (H-optimus-1 (Scalbert et al., 2026) 0.923 $\approx$ Virchow2 (Zimmermann et al., 2024) 0.922 $>$ Prov-GigaPath (Xu et al., 2024) 0.909).

GCcore is both the strongest method and the one least dependent on the slide encoder (Figure 5). Over the 12 combinations of encoder and retention rate (3 encoders $\times$ 4 low retention rates, 0.5–5%), GCcore ranks in the top four of the 15 selection strategies in all 12 and is the single best method in 8 of them (TITAN at every rate, MADELEINE at 1%, 2% and 5%, and PRISM at 2%). No baseline is comparably stable: the next most consistent method, ELFS-Pseudo, falls to 7th, and every other baseline falls to 8th or worse in at least one combination. We quantify encoder sensitivity relative to each encoder's own ceiling: for a given method and retention rate we compute how far the method falls below the full-dataset score of each encoder, take the range of these three gaps, and average it over the four low retention rates. Smaller values indicate a method whose loss relative to the achievable ceiling is more stable across encoders. GCcore attains the smallest value of all methods (11.7 points), ahead of ELFS-Proto (13.3) and far ahead of D^2-Uniform (32.6); the full ordering is GCcore (11.7) $<$ ELFS-Proto (13.3) $<$ Random (13.9) $<$ Moderate (14.2) $<$ CCS (15.4) $<$ InfoMax-Easy (16.3) $<$ K-means (17.3) $<$ D^2-Proto (18.4) $<$ ELFS-Pseudo (18.8) $<$ InfoMax-Hard (24.5) $<$ K-center (25.6) $<$ ZCore (26.4) $<$ Proto-Hard (28.4) $<$ Proto-Easy (30.2) $<$ D^2-Uniform (32.6).

GCcore remains robust across tile encoders (Figure 6). Over the 12 combinations of tile encoder and retention rate (3 encoders $\times$ 4 low retention rates, 0.02–0.2%), GCcore is the best of the 15 selection strategies in 10 and is never worse than 2nd, giving it both the best worst-case rank (2nd) and the best mean rank (1.17) of any method. It also has the lowest encoder sensitivity (1.5 points), ahead of CCS (1.8) and ELFS-Proto (2.0).

Taken together, GCcore attains the top mean rank across six different feature extractors (Table 7) and its advantage is a property of the selection method itself rather than of any particular encoder.

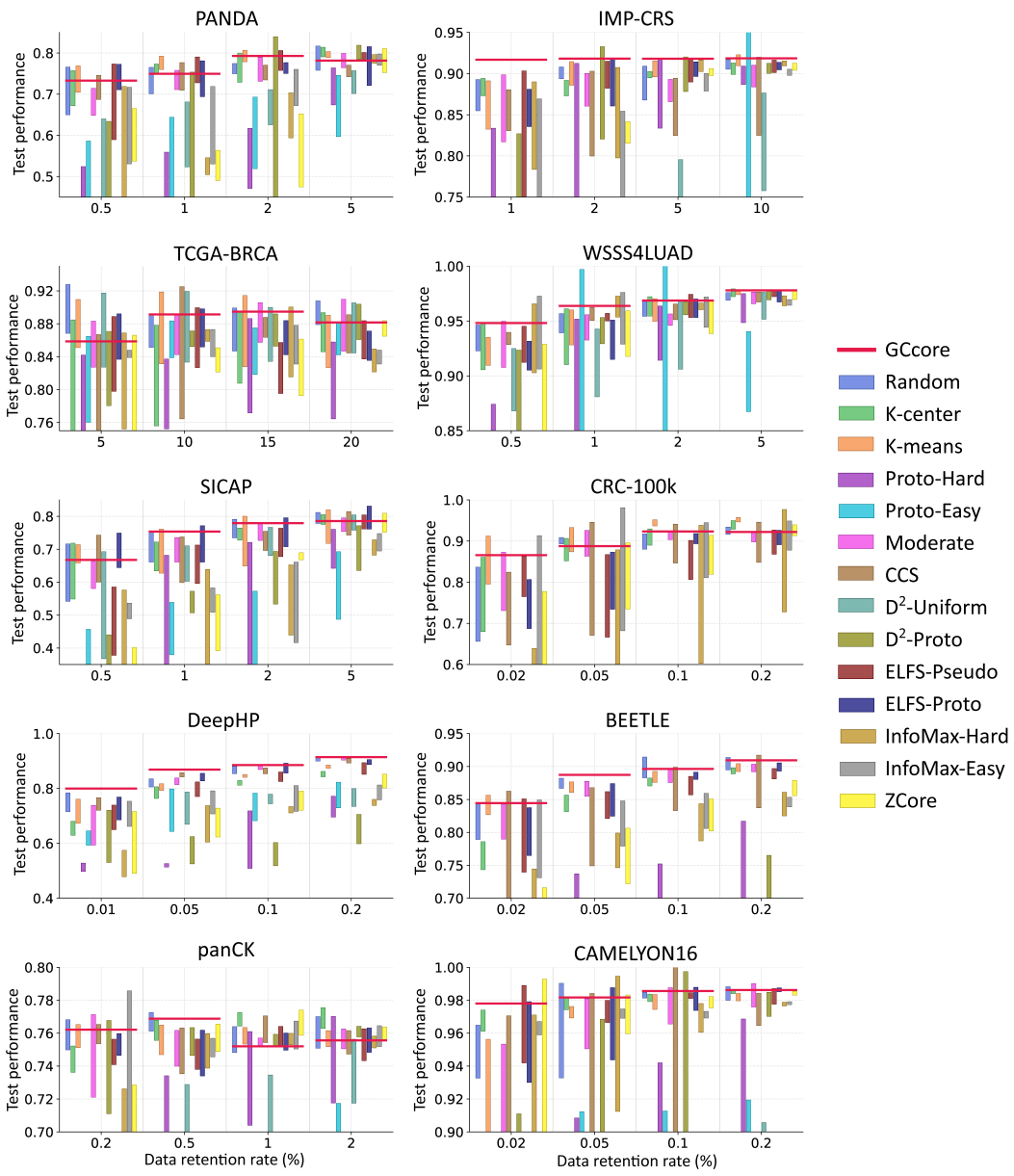

Figure 4: Baselines are sensitive to their hyperparameters; GCcore is deterministic and hyperparameter-free. Test performance at the four low data retention rates for 10 histopathology datasets spanning WSI classification (PANDA, IMP-CRS, TCGA-BRCA), tile classification (WSSS4LUAD, SICAP, CRC-100k, DeepHP) and segmentation (BEETLE, panCK, CAMELYON16). Each shaded box spans the mean ± one SD of a baseline across its five hyperparameter values; the box therefore reports the spread induced by the selection procedure itself. GCcore (red) is deterministic and yields a single test performance per retention rate, shown as a horizontal level over the corresponding group.

G.3 GCCORE TRANSFERS ACROSS SELECTION AND TRAINING ENCODERS

As explained in Section 4, the embeddings extracted from a single FM were used for both coreset selection and downstream model training. In this section we investigate how far GCcore transfers across the FM used for coreset selection and the FM used for downstream training, using the PANDA dataset.

Table 7: Mean rank of the 15 selection methods under each FM as the feature extractor; the subscript is the SD of that rank across five hyperparameter values. GCcore is deterministic and reports no SD. **Mean**: average across the six feature extractors.

Selection method	Slide encoder (PANDA)			Tile encoder (BEETLE)			**Mean**
	PRISM	TITAN	MADELEINE	H-optimus-1	Prov-GigaPath	Virchow2	
GCcore	3.0	**1.0**	**1.5**	**1.3**	**1.3**	**1.0**	**1.5**
Random	$6.5_{1.8}$	$3.8_{0.4}$	$3.8_{0.7}$	$2.3_{1.6}$	$4.0_{1.6}$	$3.3_{1.9}$	$3.9_{1.3}$
Moderate	$7.8_{2.8}$	$4.0_{1.8}$	$4.0_{1.5}$	$3.8_{1.2}$	$2.5_{1.1}$	$3.0_{0.7}$	$4.2_{1.5}$
K-means	$\mathbf{2.0}_{0.8}$	$6.8_{2.3}$	$4.8_{2.5}$	$3.5_{0.8}$	$4.5_{1.1}$	$4.0_{1.0}$	$4.3_{1.4}$
ELFS-Proto	$5.8_{1.3}$	$3.8_{0.7}$	$2.0_{1.2}$	$4.3_{1.2}$	$6.0_{1.6}$	$4.8_{1.2}$	$4.4_{1.2}$
ELFS-Pseudo	$4.5_{1.7}$	$5.3_{1.2}$	$6.8_{3.1}$	$6.8_{0.9}$	$5.5_{1.6}$	$5.3_{1.3}$	$5.7_{1.6}$
K-center	$3.0_{1.5}$	$10.5_{0.8}$	$9.3_{0.3}$	$6.5_{0.4}$	$4.5_{1.5}$	$7.3_{1.1}$	$6.8_{0.9}$
CCS	$7.3_{0.9}$	$3.8_{3.0}$	$5.5_{2.4}$	$8.5_{3.3}$	$9.5_{2.3}$	$8.0_{1.7}$	$7.1_{2.3}$
InfoMax-Easy	$8.8_{3.3}$	$8.5_{1.8}$	$9.0_{1.8}$	$8.5_{1.5}$	$9.0_{0.8}$	$9.0_{1.3}$	$8.8_{1.8}$
ZCore	$11.5_{1.3}$	$9.8_{0.8}$	$12.3_{0.7}$	$10.3_{0.4}$	$8.8_{1.1}$	$10.0_{0.8}$	$10.4_{0.8}$
InfoMax-Hard	$11.0_{1.4}$	$11.5_{1.1}$	$11.5_{0.8}$	$10.5_{0.5}$	$10.5_{0.6}$	$10.5_{1.4}$	$10.9_{1.0}$
D^2-Proto	$9.3_{2.7}$	$10.3_{2.5}$	$8.5_{2.2}$	$13.5_{0.3}$	$13.5_{0.1}$	$13.4_{0.3}$	$11.4_{1.4}$
Proto-Hard	$14.8_{1.6}$	$13.5_{1.0}$	$13.0_{1.0}$	$12.0_{0.5}$	$12.0_{0.8}$	$12.5_{1.0}$	$13.0_{1.0}$
D^2-Uniform	$11.0_{0.9}$	$13.5_{2.9}$	$13.8_{1.3}$	$14.3_{0.4}$	$14.0_{0.0}$	$14.6_{0.6}$	$13.5_{1.0}$
Proto-Easy	$14.0_{2.1}$	$14.3_{1.6}$	$14.5_{0.7}$	$14.3_{0.1}$	$14.5_{0.1}$	$13.5_{0.8}$	$14.2_{0.9}$

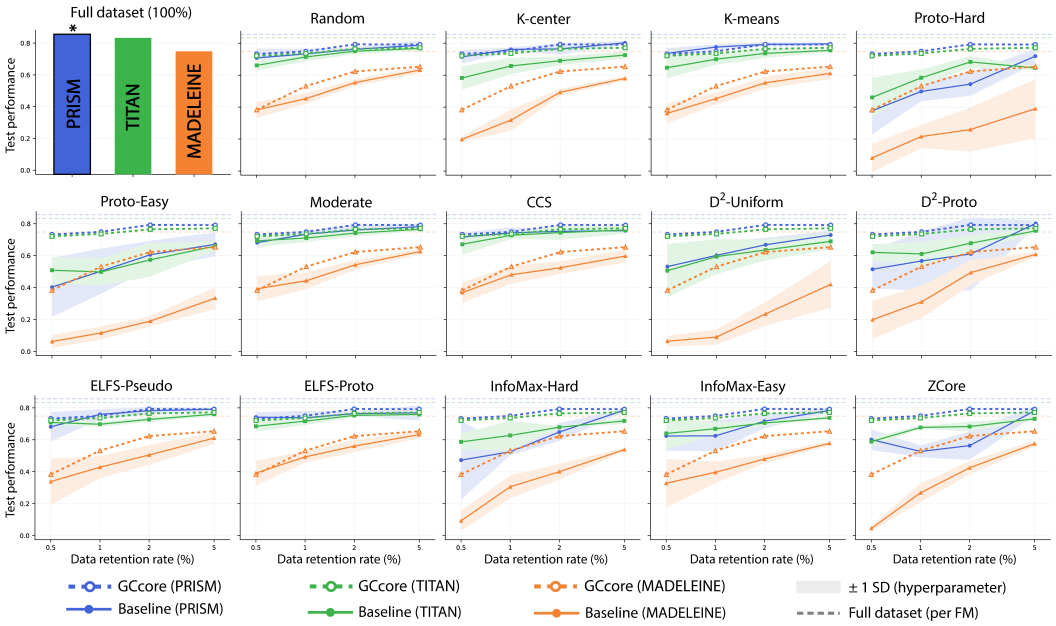

Figure 5: Test performance on different data retention rates (0.5–5%, log scale) for three slide encoders (FMs): PRISM (blue), TITAN (green), and MADELEINE (orange) on PANDA dataset. The leftmost panel reports full-dataset (100%) test performance per FM, with the strongest model starred. The remaining 14 panels each compare GCcore (dotted) against one baseline selection method (solid).

Table 8 shows that GCcore attains the best mean rank of all methods (2.69). The transfer is strongest when a stronger FM is used for selection. Over the six pairs in which PRISM or TITAN provides the selection embedding, GCcore attains a mean rank of 2.04 and is never worse than 3.25, irrespective of which of the three FMs is used for training; in the two pairs where it is not first, it is second or joint second. When MADELEINE provides the selection embedding, however, GCcore remains best in the matched setting (1.50) but falls to 4.75 (5th) and 5.75 (7th) when the coreset is transferred to PRISM and TITAN, giving a mean rank of 4.00 over those three pairs. MADELEINE also yields

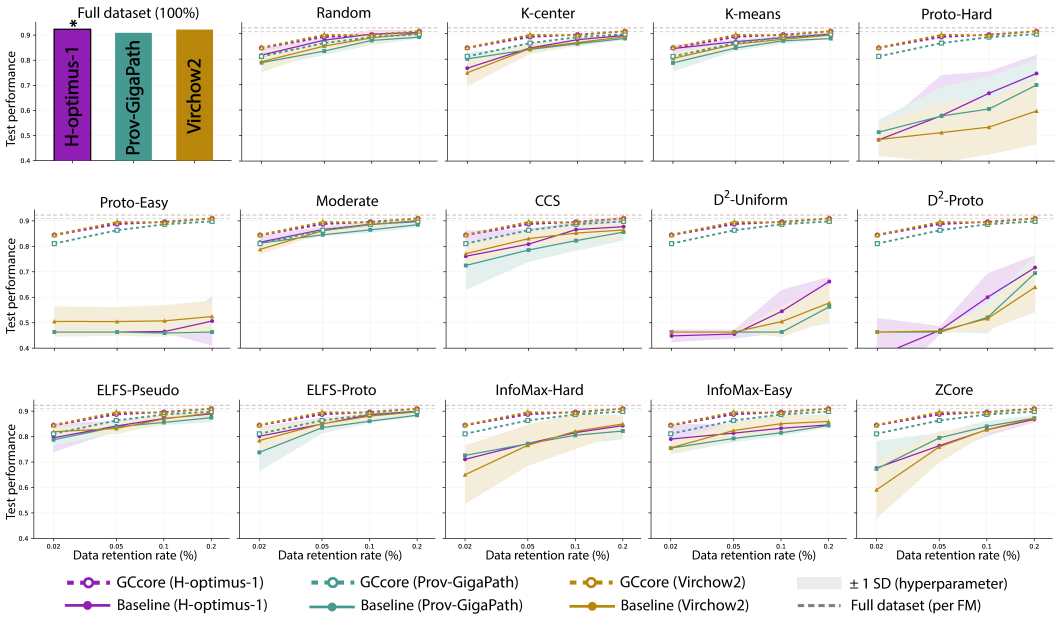

Figure 6: Test performance on different data retention rates (0.02–0.2%, log scale) for three tile encoders (FMs): H-optimus-1 (purple), Prov-GigaPath (teal), and Virchow2 (ochre) on BEETLE dataset. Panels and line styles as in Figure 5.

Table 8: Mean rank of the 15 selection methods for every pair of selection and training FMs on PANDA; the subscript is the SD of that rank across five hyperparameter values. GCcore is deterministic and reports no SD. Column groups give the FM whose embedding was used for coreset selection; the three sub-columns within a group give the frozen FM used for model training and testing. **Mean**: average across the nine selection–training FM pairs.

Selection method	Selection: PRISM			Selection: TITAN			Selection: MADELEINE			**Mean**
	PRISM	TITAN	MADELEINE	PRISM	TITAN	MADELEINE	PRISM	TITAN	MADELEINE	
GCcore	3.0	**3.3**	**1.8**	**1.3**	**1.0**	2.0	4.8	5.8	**1.5**	**2.7**
ELFS-Pseudo	$4.5_{1.7}$	$3.8_{1.6}$	$3.3_{1.2}$	$4.0_{1.2}$	$5.3_{1.2}$	$\mathbf{1.5}_{0.7}$	$3.5_{1.2}$	$4.3_{1.3}$	$6.8_{3.1}$	$4.1_{1.5}$
K-means	$\mathbf{2.0}_{0.8}$	$4.3_{1.9}$	$3.8_{1.6}$	$3.5_{1.8}$	$6.8_{2.3}$	$7.3_{2.6}$	$3.8_{0.5}$	$4.8_{1.0}$	$4.8_{2.5}$	$4.5_{1.7}$
Moderate	$7.8_{2.8}$	$3.8_{0.4}$	$4.5_{1.4}$	$6.8_{2.5}$	$4.0_{1.8}$	$3.3_{2.2}$	$6.3_{2.5}$	$\mathbf{4.0}_{0.5}$	$4.0_{1.5}$	$4.9_{1.7}$
Random	$6.5_{1.8}$	$4.0_{0.8}$	$6.8_{1.1}$	$6.3_{1.8}$	$3.8_{0.4}$	$5.5_{0.8}$	$4.0_{1.8}$	$\mathbf{4.0}_{0.9}$	$3.8_{0.7}$	$4.9_{1.1}$
ELFS-Proto	$5.8_{1.3}$	$6.5_{2.6}$	$7.3_{1.8}$	$7.3_{0.7}$	$3.8_{0.7}$	$6.0_{1.4}$	$\mathbf{3.3}_{2.3}$	$\mathbf{4.0}_{2.5}$	$2.0_{1.2}$	$5.1_{1.6}$
CCS	$7.3_{0.9}$	$9.0_{0.9}$	$10.3_{1.5}$	$8.3_{1.1}$	$3.8_{3.0}$	$4.8_{1.9}$	$5.0_{2.7}$	$5.0_{2.2}$	$5.5_{2.4}$	$6.5_{1.8}$
InfoMax-Easy	$8.8_{3.3}$	$8.8_{1.4}$	$3.8_{2.6}$	$5.5_{0.6}$	$8.5_{1.8}$	$9.0_{2.4}$	$8.3_{2.0}$	$7.5_{1.8}$	$9.0_{1.8}$	$7.7_{2.0}$
K-center	$3.0_{1.5}$	$5.8_{2.9}$	$10.3_{1.3}$	$9.8_{1.1}$	$10.5_{0.8}$	$8.8_{1.5}$	$11.0_{2.5}$	$9.3_{0.6}$	$9.3_{0.3}$	$8.6_{1.4}$
D²-Proto	$9.3_{2.7}$	$12.0_{0.5}$	$9.8_{1.9}$	$10.9_{3.5}$	$10.3_{2.5}$	$9.3_{2.0}$	$8.8_{2.2}$	$8.0_{2.7}$	$8.5_{2.2}$	$9.6_{2.3}$
InfoMax-Hard	$11.0_{1.4}$	$11.5_{1.1}$	$6.8_{2.5}$	$7.8_{2.2}$	$11.5_{1.1}$	$11.0_{0.8}$	$12.3_{0.8}$	$10.3_{0.5}$	$11.5_{0.8}$	$10.4_{1.2}$
ZCore	$11.5_{1.3}$	$8.3_{1.9}$	$10.5_{1.3}$	$8.9_{1.0}$	$9.8_{0.8}$	$9.8_{1.2}$	$13.0_{0.5}$	$12.8_{0.7}$	$12.3_{0.7}$	$10.7_{1.0}$
Proto-Hard	$14.8_{1.6}$	$10.8_{1.4}$	$13.0_{1.8}$	$13.3_{0.6}$	$13.5_{1.0}$	$13.3_{0.2}$	$11.0_{3.8}$	$12.3_{4.3}$	$13.0_{1.0}$	$12.8_{1.7}$
D²-Uniform	$11.0_{0.9}$	$13.5_{1.5}$	$13.5_{0.6}$	$13.0_{2.7}$	$13.5_{2.9}$	$13.8_{1.6}$	$13.5_{0.5}$	$14.0_{2.1}$	$13.8_{1.3}$	$13.3_{1.6}$
Proto-Easy	$14.0_{2.1}$	$15.0_{0.3}$	$15.0_{0.1}$	$13.8_{1.8}$	$14.3_{1.6}$	$15.0_{0.2}$	$11.8_{0.9}$	$14.3_{3.2}$	$14.5_{0.7}$	$14.2_{1.2}$

the lowest full-dataset ceiling of the three FMs (QWK 0.748, against 0.856 for PRISM and 0.833 for TITAN).

These results indicate that, on PANDA, a coreset selected by GCcore with a stronger encoder (PRISM or TITAN) remains among the top-ranked choices for training with any of the three encoders, so it need not be recomputed for a different downstream feature extractor. Selecting with

the weakest available encoder, on the other hand, is not advisable when the coreset is intended for transfer.

G.4 GCcore is robust to class imbalance

Handling class imbalance during training. As described in Section 4, the classification and segmentation heads were trained with a class-weighted cross-entropy loss on all datasets except WSSS4LUAD, in order to mitigate the class imbalance of the training set. Because WSSS4LUAD is a multi-label problem, a weighted categorical cross-entropy loss is not appropriate as the three tissue classes are not mutually exclusive and a single tile may contain several tissue types. We therefore weight the positive term of class c by

$$\lambda_c = \min\left(\frac{n_c^-}{n_c^+}, \lambda_{\max}\right), \qquad \lambda_{\max} = 10, \tag{12}$$

where n_c^+ and n_c^- are the positive and negative counts of class c within the selected coreset, since that is the distribution to which the head is exposed. The cap $\lambda_{\max}$ prevents a class that becomes very rare in a small coreset from dominating the gradient.

Simulating strongly imbalanced training sets. To further investigate the robustness of GCcore to strongly class-imbalanced training data, we simulated imbalanced training sets from the relatively balanced PANDA training set (grade 0, 2,313; grade 1, 2,133; grade 2, 1,074; grade 3, 994; grade 4, 999; grade 5, 979 WSIs; 8,492 in total). In each simulation, five of the six ISUP grades were drawn at random and down-sampled to 1%, 20%, 40%, 60% and 80% of their WSIs, respectively, while the remaining grade was retained in full; within each down-sampled grade, the retained WSIs were selected uniformly at random. Repeating this procedure three times yielded three imbalanced training sets (IMBs). In IMB1, grades 5, 3, 1, 4 and 2 retained 1%, 20%, 40%, 60% and 80% of their WSIs, respectively, and grade 0 was retained in full, giving 4,833 WSIs in total (grade 0, 2,313; grade 1, 853; grade 2, 859; grade 3, 199; grade 4, 599; grade 5, 10); the imbalance ratio between the largest and the smallest grade thereby increased from 2.4 to 231.3. In IMB2, grades 0, 5, 3, 1 and 2 retained 1%, 20%, 40%, 60% and 80% of their WSIs, respectively, and grade 4 was retained in full, giving 3,755 WSIs in total (grade 0, 23; grade 1, 1,280; grade 2, 859; grade 3, 398; grade 4, 999; grade 5, 196; imbalance ratio 55.7). In IMB3, grades 4, 0, 2, 1 and 3 retained 1%, 20%, 40%, 60% and 80% of their WSIs, respectively, and grade 5 was retained in full, giving 3,957 WSIs in total (grade 0, 463; grade 1, 1,280; grade 2, 430; grade 3, 795; grade 4, 10; grade 5, 979; imbalance ratio 128.0).

Each IMB was then treated as the full dataset from which coresets were selected, at the four low retention rates $\{0.5, 1, 2, 5\}\%$ and using each of the 15 selection strategies. Performance is always reported on the complete test set (2,124 WSIs).

GCcore ranks best overall across three imbalance regimes. Table 9 shows that GCcore attains the best mean rank over the three IMBs (3.3 of 15) and is the only strategy placed among the two best on all three sets. Notably, GCcore reaches a QWK of 0.657 from only 19 WSIs (0.5% retention) on IMB2, recovering 90.7% of the QWK obtained with the entire imbalanced training set (0.724), against 0.549 for the best competing strategy and 0.477 for Random. No competing strategy reaches this level at retention rates up to 2% (75 WSIs), and Random requires 20% retention (751 WSIs) to reach a comparable level; that is, a coreset of 19 slides selected by GCcore performs on par with a randomly selected coreset roughly 40 times larger.

The advantage of GCcore does not necessarily arise from class re-balancing. A natural explanation for robustness under imbalance is that a selection strategy implicitly re-balances the label distribution. We tested this by quantifying the class balance of each coreset as the normalized Shannon entropy over the six ISUP grades (evenness; 1 = uniform) and relating it to test performance (Fig. 7). Balance and performance are positively associated across strategies on IMB1 ($R^2 = 0.57$, $p < 0.001$) and on IMB3 ($R^2 = 0.42$, $p < 0.001$), and on both GCcore selects the most balanced coreset of all 15 strategies at 0.5 and 1% retention while also ranking first in test performance. On IMB2, however, the association weakens markedly ($R^2 = 0.10$, $p = 0.037$), with evenness explaining only 10% of the variation in performance. GCcore is only mid-ranked in balance (evenness rank 6–11 of 15 at retention $\leq 5\%$) while ranking first in test performance at 0.5 and 1% retention. The most balanced coreset in fact performs below the median at both rates: Proto-Hard attains the

Table 9: Mean rank of the 15 selection methods on each class-imbalanced training set; the subscript is the SD of that rank across five hyperparameter values. **Mean**: average across the three imbalanced training sets.

Selection method	Class-imbalanced training sets			**Mean**
	IMB1	IMB2	IMB3	
GCcore	2.8	4.8	**2.5**	**3.3**
K-means	5.3 2.9	7.0 4.4	3.3 2.1	5.2 3.1
ELFS-Proto	**2.3** 1.5	6.8 2.0	7.0 2.0	5.3 1.8
K-center	4.8 1.3	5.0 4.9	6.3 1.9	5.3 2.7
ELFS-Pseudo	6.3 2.4	**4.0** 2.6	7.8 3.0	6.0 2.6
InfoMax-Easy	8.8 2.4	6.5 3.1	3.0 1.9	6.1 2.4
Random	4.5 1.2	9.3 1.1	5.8 1.8	6.5 1.4
Moderate	5.3 2.9	6.3 1.8	8.5 2.8	6.7 2.5
CCS	7.3 2.7	9.0 3.2	7.0 1.0	7.8 2.3
D^2-Proto	8.3 2.3	5.8 1.9	11.3 1.9	8.4 2.0
InfoMax-Hard	11.8 1.1	7.0 2.3	9.0 1.2	9.3 1.5
ZCore	13.5 1.3	11.0 1.1	10.5 0.9	11.7 1.1
D^2-Uniform	12.3 1.4	12.8 1.7	11.8 1.4	12.3 1.5
Proto-Easy	13.3 0.3	13.3 2.7	12.8 1.1	13.1 1.4
Proto-Hard	14.0 0.8	11.8 2.5	13.8 1.7	13.2 1.6

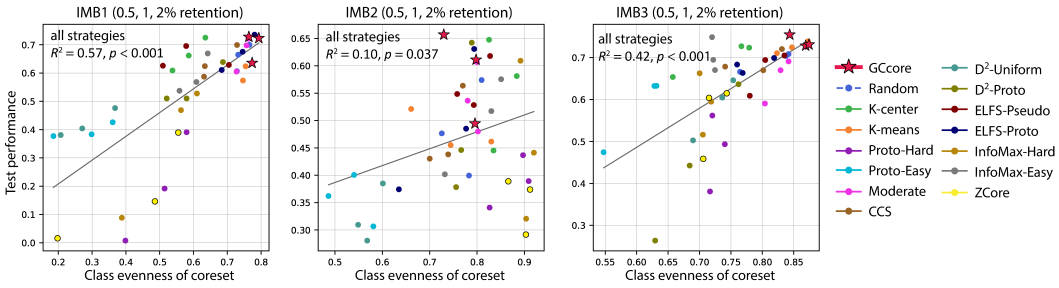

Figure 7: A class-balanced selection is neither necessary nor sufficient for better performance.

highest evenness at 0.5% retention (0.909, versus 0.730 for GCcore) yet ranks only 10 of 15 in test performance (0.389), and InfoMax-Hard attains the highest evenness at 1% retention (0.920, versus 0.798 for GCcore) yet ranks 9 of 15 (0.441). A more even label distribution is therefore neither necessary nor sufficient for performance under class imbalance.

G.5 GCcore is robust to the choice of distance metric

GCcore is formulated for a general distance in Eq. 1. The derivation in Appendix D uses only two properties: that the pairwise dissimilarity is symmetric and nonnegative, and that $r_{\max}$ is the maximum pairwise value of that same dissimilarity over $\mathcal{S}$. Together, these give $\mathrm{sim}(x_i, x_j) \geq 0$ and $d_i(\mathcal{T}) \in [0, r_{\max}]$, which are all that Eq. 7, the identity $c_i(\mathcal{T}) = r_{\max} - d_i(\mathcal{T})$, and Lemma 1 require. Therefore, the $(1 - 1/e)$ guarantee of Eq. 11 holds for any distance metric. The selection procedure is otherwise unchanged: only the pairwise distances and $r_{\max}$ are recomputed.

We use the ℓ_1 distance throughout the paper. To verify that the method is not tied to this choice, we repeat the selection with the Euclidean distance

$$\|\mathbf{e}_i - \mathbf{e}_j\|_2 = \Big(\sum_{m=1}^{D} (e_{im} - e_{jm})^2 \Big)^{1/2}, \tag{13}$$

i.e., $d_i(\mathcal{T}) = \min_{x_j \in \mathcal{T}} \|\mathbf{e}_i - \mathbf{e}_j\|_2$ in Eq. 1, with $r_{\max}$ recomputed as the maximum pairwise ℓ_2 distance over $\mathcal{S}$.

Table 10 shows that both GCcore (ℓ_1) and GCcore (ℓ_2) rank ahead of every baseline on average, with mean ranks of 2.7 and 3.4 respectively, against 5.0 for the strongest baseline (Random). Per

Table 10: Mean rank of the 16 selection methods on each dataset, with GCcore's two distance variants treated as separate methods; the subscript is the SD of that rank across five hyperparameter values. GCcore is deterministic and reports no SD. **Mean**: average across the 10 datasets.

Selection method	WSI classification			Tile classification				Segmentation			**Mean**
	PANDA	IMP-CRS	TCGA-BRCA	WSSS4 LUAD	SICAP	CRC-100k	Deep HP	BEE TLE	panCK	CAME-LYON16	
GCcore (ℓ_1)	3.1	**1.5**	**2.9**	**1.5**	3.0	4.0	**1.1**	**1.8**	6.5	**1.5**	**2.7**
GCcore (ℓ_2)	4.6	1.8	6.4	3.3	4.0	2.8	1.9	**1.8**	5.5	2.0	3.4
Random	$7.5_{2.1}$	$5.8_{1.2}$	$3.0_{1.2}$	$6.3_{2.0}$	$3.8_{2.1}$	$5.5_{1.1}$	$4.5_{0.9}$	$3.0_{1.7}$	$4.3_{1.8}$	$6.5_{1.1}$	$5.0_{1.5}$
K-means	$\mathbf{2.0}_{1.0}$	$3.8_{1.5}$	$6.3_{2.9}$	$7.6_{1.9}$	$6.8_{3.5}$	$\mathbf{1.5}_{1.1}$	$6.8_{0.7}$	$4.5_{0.9}$	$5.8_{2.3}$	$8.0_{1.8}$	$5.3_{1.7}$
ELFS-Proto	$6.5_{1.5}$	$5.8_{1.5}$	$7.3_{3.3}$	$9.5_{1.5}$	$\mathbf{2.3}_{3.4}$	$8.3_{1.1}$	$4.8_{0.5}$	$5.3_{1.2}$	$9.3_{1.8}$	$5.5_{0.7}$	$6.4_{1.6}$
K-center	$3.5_{1.8}$	$7.3_{0.8}$	$12.0_{4.7}$	$7.3_{3.3}$	$5.3_{1.9}$	$4.8_{1.7}$	$9.3_{0.4}$	$7.5_{0.4}$	$\mathbf{3.8}_{0.8}$	$3.8_{0.8}$	$6.4_{1.7}$
Moderate	$8.8_{3.1}$	$10.0_{1.5}$	$5.0_{2.6}$	$10.1_{2.7}$	$6.5_{1.6}$	$5.3_{2.1}$	$5.0_{1.1}$	$4.8_{1.3}$	$9.0_{2.9}$	$8.3_{2.8}$	$7.3_{2.2}$
ELFS-Pseudo	$4.8_{2.0}$	$7.0_{2.4}$	$11.3_{3.6}$	$4.8_{1.5}$	$9.3_{2.9}$	$9.0_{1.6}$	$8.0_{0.8}$	$7.8_{0.9}$	$9.8_{2.4}$	$5.0_{1.2}$	$7.7_{1.9}$
CCS	$8.3_{1.1}$	$11.8_{2.7}$	$9.5_{2.8}$	$7.5_{1.3}$	$6.8_{1.8}$	$9.5_{2.4}$	$4.5_{0.2}$	$9.5_{3.4}$	$7.0_{2.3}$	$12.3_{3.9}$	$8.7_{2.2}$
InfoMax-Easy	$9.8_{3.8}$	$11.3_{0.3}$	$11.0_{2.0}$	$8.3_{4.1}$	$11.8_{0.7}$	$7.5_{3.6}$	$9.5_{1.7}$	$9.5_{1.5}$	$7.3_{4.0}$	$7.8_{0.3}$	$9.4_{2.2}$
ZCore	$12.5_{1.4}$	$9.8_{0.8}$	$12.5_{1.1}$	$10.5_{2.7}$	$11.3_{1.1}$	$8.0_{2.2}$	$12.0_{0.9}$	$11.3_{0.4}$	$6.0_{2.1}$	$7.0_{3.0}$	$10.1_{1.6}$
InfoMax-Hard	$12.0_{1.5}$	$6.5_{1.5}$	$11.0_{1.8}$	$5.5_{1.4}$	$13.0_{1.7}$	$12.0_{2.0}$	$14.0_{0.4}$	$11.5_{0.5}$	$10.0_{1.5}$	$10.8_{2.3}$	$10.6_{1.5}$
D^2-Proto	$10.0_{3.0}$	$10.0_{2.7}$	$6.8_{1.2}$	$9.5_{2.3}$	$13.0_{2.2}$	$13.8_{0.4}$	$14.5_{1.1}$	$14.5_{0.3}$	$7.0_{2.7}$	$12.8_{3.9}$	$11.2_{2.0}$
D^2-Uniform	$12.0_{0.9}$	$15.3_{0.1}$	$4.5_{3.8}$	$13.8_{4.7}$	$8.5_{1.8}$	$15.8_{0.4}$	$12.8_{1.3}$	$15.3_{0.4}$	$15.0_{2.0}$	$15.9_{0.2}$	$12.9_{1.6}$
Proto-Hard	$15.8_{1.6}$	$13.0_{1.4}$	$15.5_{2.7}$	$14.8_{2.1}$	$15.5_{3.4}$	$13.3_{0.4}$	$15.3_{0.5}$	$13.0_{0.5}$	$14.0_{2.0}$	$14.0_{1.5}$	$14.4_{1.6}$
Proto-Easy	$15.0_{2.1}$	$15.8_{0.2}$	$11.3_{2.2}$	$16.0_{0.3}$	$15.5_{1.1}$	$15.3_{0.2}$	$12.3_{1.3}$	$15.3_{0.1}$	$16.0_{0.1}$	$15.1_{1.1}$	$14.7_{0.9}$

Table 11: Mean rank of the 15 selection methods on PANDA under the native slide-embedding dimensionality D and the PCA-reduced dimensionality D' (99% explained variance; PRISM $1280 \to 189$, TITAN $768 \to 207$). The subscript is the SD of that rank across five hyperparameter values. GCcore is deterministic and reports no SD. **Mean**: average across the two slide encoders.

Selection method	Full embedding ($\mathbf{e} \in \mathbb{R}^D$)			Reduced embedding ($\mathbf{e} \in \mathbb{R}^{D'}$)		
	PRISM	TITAN	**Mean**	PRISM	TITAN	**Mean**
GCcore	3.0	**1.0**	**2.0**	**2.3**	**2.8**	**2.5**
K-means	$\mathbf{2.0}_{0.8}$	$6.8_{2.3}$	$4.4_{1.5}$	$3.5_{1.4}$	$4.8_{1.2}$	$4.1_{1.3}$
ELFS-Proto	$5.8_{1.3}$	$3.8_{0.7}$	$4.8_{1.0}$	$5.8_{0.9}$	$3.3_{1.3}$	$4.5_{1.1}$
InfoMax-Easy	$8.8_{3.3}$	$8.5_{1.8}$	$8.6_{2.6}$	$2.5_{2.4}$	$6.5_{2.5}$	$4.5_{2.4}$
ELFS-Pseudo	$4.5_{1.7}$	$5.3_{1.2}$	$4.9_{1.4}$	$3.0_{1.2}$	$7.0_{1.7}$	$5.0_{1.5}$
Random	$6.5_{1.8}$	$3.8_{0.4}$	$5.1_{1.1}$	$7.0_{1.7}$	$3.3_{0.7}$	$5.1_{1.2}$
CCS	$7.3_{0.9}$	$3.8_{3.0}$	$5.5_{2.0}$	$7.3_{0.7}$	$5.0_{1.4}$	$6.1_{1.0}$
InfoMax-Hard	$11.0_{1.4}$	$11.5_{1.1}$	$11.3_{1.3}$	$9.0_{1.3}$	$5.8_{0.9}$	$7.4_{1.1}$
Moderate	$7.8_{2.8}$	$4.0_{1.8}$	$5.9_{2.3}$	$9.0_{2.9}$	$8.3_{1.8}$	$8.6_{2.4}$
K-center	$3.0_{1.5}$	$10.5_{0.8}$	$6.8_{1.1}$	$7.8_{2.0}$	$10.0_{1.9}$	$8.9_{2.0}$
D^2-Proto	$9.3_{2.7}$	$10.3_{2.5}$	$9.8_{2.6}$	$10.5_{1.0}$	$10.8_{2.2}$	$10.6_{1.6}$
D^2-Uniform	$11.0_{0.9}$	$13.5_{2.9}$	$12.3_{1.9}$	$13.0_{1.8}$	$11.3_{1.9}$	$12.1_{1.9}$
ZCore	$11.5_{1.3}$	$9.8_{0.8}$	$10.6_{1.0}$	$11.8_{0.5}$	$14.0_{0.2}$	$12.9_{0.4}$
Proto-Hard	$14.8_{1.6}$	$13.5_{1.0}$	$14.1_{1.3}$	$12.8_{0.6}$	$13.3_{2.2}$	$13.0_{1.4}$
Proto-Easy	$14.0_{2.1}$	$14.3_{1.6}$	$14.1_{1.8}$	$15.0_{0.9}$	$14.3_{0.6}$	$14.6_{0.7}$

dataset, the two differ by a median of 1.0 rank positions, take the top two places on five datasets, and both fall within the top five on eight. GCcore's advantage over the baselines is therefore a property of its global coverage maximization mechanism, not of the distance used to instantiate it.

G.6 GCcore is robust to the dimensionality of the FM embedding

As described in Section 4.1, the WSI classification head maps a slide embedding $\mathbf{e} \in \mathbb{R}^D$ to the class logits, where D is the native output dimensionality of the slide encoder ($D = 1280$ for PRISM, $D = 768$ for TITAN). To test the robustness of GCcore to the geometry of the embedding space, we applied PCA to the slide embeddings of PANDA and retained the leading components accounting for 99% of the variance, giving $\mathbf{e} \in \mathbb{R}^{D'}$ with $D' = 189$ for PRISM (-85%) and $D' = 207$

for TITAN (-73%). We then repeated the comparison of all 15 selection methods in $\mathbb{R}^{D'}$ at low retention rates. Note that random selection does not use the embeddings, so its coresets are identical in both regimes.

Table 11 shows that GCcore achieves the best mean rank in both regimes, 2.0 on the native embeddings and 2.5 on the reduced embeddings, and is the best-ranked method on the reduced embeddings of both FMs. More broadly, the ranking of the 15 selection strategies is largely preserved under dimensionality reduction: Spearman's rank correlation between the full and reduced embeddings is $\rho = 0.80$ for PRISM, $\rho = 0.82$ for TITAN and $\rho = 0.89$ for the mean over both encoders (all $p < 0.001$). In practice this means GCcore can be applied directly to compressed slide representations without modification, which is relevant given the large native dimensionalities across pathology FMs.