Mycroft: Effective and Efficient External Data Augmentation

Anonymous authors
Paper under double-blind review

Abstract

Machine learning (ML) models often require large amounts of data to perform well. When the data available to the model trainer is insufficient to obtain good performance on their desired task, they may need to acquire more data from external sources. Often, useful data is held by private entities who may be unwilling to share their data due to monetary and privacy concerns. This makes it challenging and expensive for model trainers to acquire the data they need to improve their model's performance. To tackle this problem, we propose Mycroft, a data-efficient framework that enables model trainers to compare data sources. Mycroft identifies small but informative data subsets from each data owner by leveraging subset selection methods based on (i) a sparse gradient approximation, (ii) a low-rank kernel matrix approximation, and (iii) a voting-based coverage heuristic. This allows model trainers to identify useful data owners and improve model performance with minimal data exposure. Our experiments on vision datasets show that Mycroft converges rapidly to the performance achievable when all the data is shared. Moreover, Mycroft is robust to label and data noise, and can effectively recover a utility-based ranking of data owners. We believe Mycroft paves the way for principled data markets which can democratize training of high performance ML models.

1 Introduction

Machine learning models' performance relies heavily on their training datasets, but the data available for training is often insufficient, outdated, or unrepresentative of the task (Stacke et al., 2021; Elsahar & Gallé, 2019; Liu et al., 2023). As models expand into new domains and existing ones exhaust public data, data scarcity becomes a key challenge. According to economic and manufacturing experts, one of the primary reasons that AI technology is not being widely adopted in manufacturing is the lack of relevant public data available for production tasks (Alam et al., 2024). While large corporations and governments can afford large-scale data collection or use methods like crowdsourcing (Sigurdsson et al., 2016) and federated learning (Kairouz et al., 2021) to increase their access to diverse datasets, entities without such resources, such as individuals, small businesses and healthcare providers, find it extremely challenging to collect their own data. Furthermore, even if data can be acquired, indiscriminately adding samples can sometimes negatively impact model generalization—a phenomenon known as the data addition dilemma (Shen et al., 2024).

These lower-resourced entities may need to acquire data from private entities (referred to as *data owners*), who are often unwilling to share it publicly, especially when it involves proprietary or sensitive data, such as educational or health records (Spector-Bagdady et al., 2019). Acquiring data from these private data owners introduces a fundamental economic challenge known as Arrow's Information Paradox (Arrow, 1972; Castro Fernandez, 2023): a data buyer cannot accurately assess the value of new data without first accessing it, yet a data owner cannot disclose the data for evaluation without effectively giving away their product for free. Combined with the high overhead of complex data sharing agreements and regulatory compliance (GDPR, 2021), this information asymmetry severely hinders external data augmentation. Therefore, data sharing protocols which help model trainers assess which data owners are most likely to provide useful data—without requiring the owners to expose their full datasets upfront—are crucial. The key question we thus ask in this paper is: *How can a model trainer assess the usefulness of data owners without acquiring their entire dataset?*

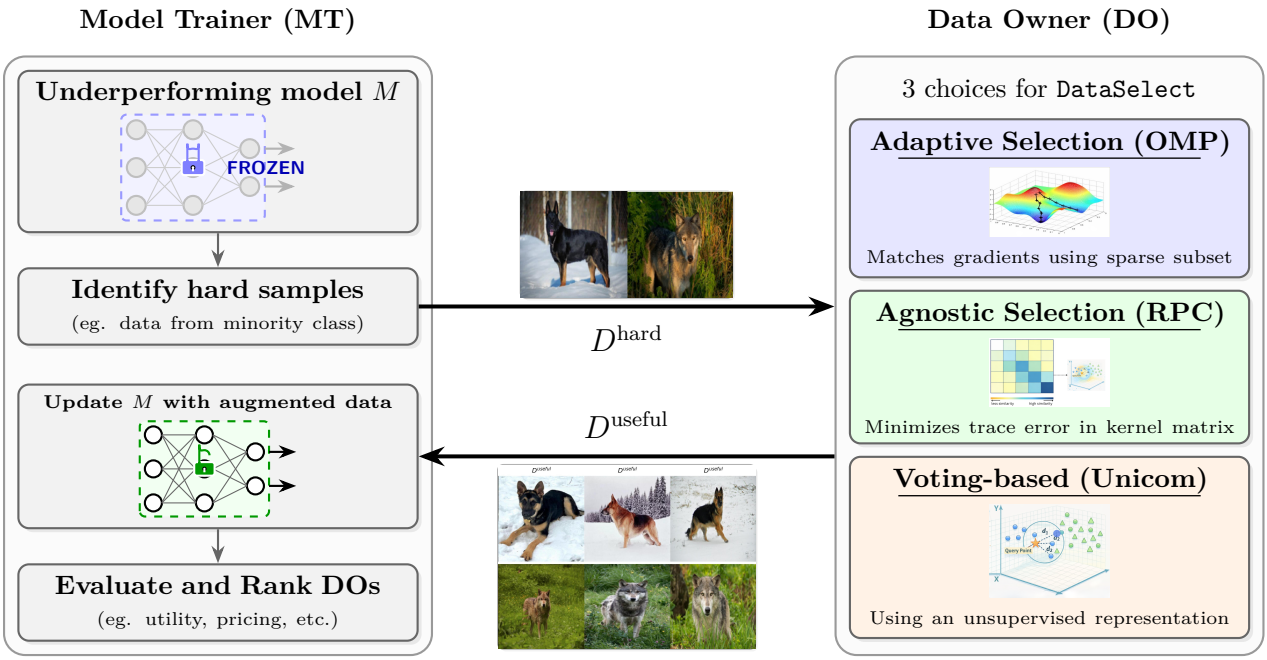


Figure 1: **Overview of the `Mycroft` Protocol.** In this motivating example, the Model Trainer's (`MT`) model relies on a spurious correlation because its training set contains only dogs on grass and wolves on snow. Consequently, it fails on the validation set (D^{hard}), which contains counterfactuals like dogs on snow and wolves on grass. To fix this, the `MT` shares a small query subset of D^{hard} with a Data Owner (`DO`) to execute a `DataSelect` subroutine. Instead of transferring the `DO`'s entire dataset, `Mycroft` identifies and retrieves only the most useful counterfactual samples (D^{useful}). The `MT` then uses this small subset to efficiently evaluate the `DO`'s utility and retrain the model, breaking the spurious correlation with minimal data exposure.

The model trainer needs to efficiently select one or more data owners whose datasets match their requirements. In the case with a profusion of data owners, while a large number can be filtered out just on the basis of metadata (*i.e.* domain, collection methodology, licensing requirements etc.), it is unclear which of the remaining data owners the model trainer should acquire data from. Even in the case of a single data owner with a large pool of data, both the model trainer and data owners will benefit from a data-sparse method to assess the dataset's potential effectiveness. We formalize the problem in §2.

We present our solution `Mycroft` (§3), a data-sharing framework (Figure 1) that helps the model trainer and data owners identify relevant data (from the data owners) for improving the model trainer's performance. The framework follows three main steps: (1) the model trainer sends information about the task on which their model is under-performing, typically through data samples, to data owners; (2) the data owners use an efficient algorithm to identify a small, relevant subset of their data to demonstrate their dataset's usefulness and send it to the model trainer; and (3) the model trainer evaluates this subset to decide whether to acquire additional data from the data owners.

The *key technical challenge* we solve in this paper lies in Step (2), where each data owner must find the most "relevant" training data with respect to the model trainer's transmitted data samples (D^{hard}), which may not follow the same distribution as their training data. We propose three distinct approaches: (1) *Adaptive Selection*: we use subset selection using orthogonal matching pursuit (OMP) over model gradients to match, in Euclidean distance, the average gradient over the transmitted data samples (inspired by GradMatch (Killamsetty et al., 2021a)); (2) *Agnostic Selection*: we adapt the Randomly Pivoted Cholesky (RPC) (Chen et al., 2025) algorithm that selects the columns of a kernel matrix between the data owner's data and D^{hard} to minimise the nuclear norm of a so-called Nyström approximation; and (3) *Voting-based Selection*: both the data owner's data and D^{hard} are embedded in a shared space (we use Unicom (An et al., 2023) for image data), and a subset is greedily selected using Euclidean distance in this space. We evaluate

`Mycroft` across 2 computer vision datasets designed with targeted spurious correlations, 2 model architectures (ResNet-50 and ViT-B/16), and 7 distinct training configurations (varying epochs, schedulers, and weight initializations) for the model trainer (§4.1). We also consider realistic situations such as data owners with varying degrees of utlity and noisy data.

Our key experimental findings are:

1. `Mycroft` approaches `full-information` with a fraction of the budget (§4.2): We evaluate `Mycroft` using all 3 data selection approaches against two baselines: (i) `full-information`, where each data owner shares all their data, which usually serves as the optimal upper bound, and (ii) `random-sampling`. Across multiple classification tasks in the computer vision domain, our experiments show that `Mycroft` significantly outperforms `random-sampling` and quickly approaches the performance of `full-information` under a constrained budget. For instance, on the Dogs & Wolves dataset using a ViT-B/16 model, `Mycroft` improves accuracy on the hard subclass from 14.09% to 49.51% with a budget of just 32 samples per class. This drastically outperforms `random-sampling` (26.40%) and effectively halves the performance gap to the `full-information` upper bound (75.78%).
2. Effective across budgets and training configurations (§4.3): We demonstrate that `Mycroft` consistently outperforms `random-sampling` even at extremely low data retrieval budgets (e.g., $k = 8$) and converges much faster to the `full-information` upper bound. We confirm that `Mycroft`'s performance gains are not dependent on a specific training recipe, showing consistent improvements across various schedules, optimizers, and both fine-tuning and training-from-scratch setups.
3. Recovers Data Owner Rankings in Data Markets (§4.4.1): In scenarios involving multiple data owners of varying qualities and utility profiles, `Mycroft` can successfully recover a true utility-based preference order, a critical task where `random-sampling` frequently fails. Furthermore, on the Spawrious dataset, we simulate a scenario with multiple data owners of varying quality, demonstrating that `Mycroft` not only improves generalization but also successfully recovers a utility-based ranking of the data owners, a critical task where `random-sampling` completely fails.
4. Robust to Data Corruptions (§4.4.2): We show that `Mycroft`'s retrieval mechanisms remain highly resilient when data owners provide severely degraded data or sample from datasets with up to 70% label noise.

Our aim is for `Mycroft` and its associated open-source code[1] to pave the way for more efficient and private data-sharing frameworks, in turn helping democratise the training of performant ML models.

2 Problem Setup and Formulation

In this section, we first present the challenges associated with external data augmentation in practical settings and the desired properties for any proposed method. We then formally define the specific problem we solve and present the required notation.

2.1 Challenges of external data augmentation

The most direct approach to external data augmentation is to acquire all data from all available data providers. However, this approach presents the following challenges:

i) Data owners are unwilling to share all their data: Due to privacy and proprietary concerns, data owners are often unwilling to share all their data. Furthermore, because they typically expect compensation, they face the information asymmetry of Arrow's paradox (Arrow, 1972; Castro Fernandez, 2023)—they must prove the utility of their data to secure any sort of compensation from the model trainer without actually revealing the data itself.
ii) Sharing poorly curated data can negatively impact performance: A large portion of the data owned by data owners may be poorly curated or irrelevant for model training. The inclusion of such data in the training process can degrade model performance (Shen et al., 2024), so it is crucial for model trainers to select only high-quality and relevant subsets from the data owners' contributions.

[1]Code is available at : https://anonymous.4open.science/r/Mycroft/

iii) Data budgets can ease computational concerns: Receiving large volumes of data from multiple data owners makes it difficult and costly for model trainers to update their models frequently.

2.2 Problem Formulation and Notation

We consider a setting where there is a single model trainer (MT) and m data owners (DOs). MT has trained a model M_{MT} on its own dataset D^{MT} and is aiming to improve their performance (or lower their loss) on test data D^{test} via external data augmentation. The performance of MT's model on D^{test} is measured with respect to some predetermined task. In this paper, we assume that the task is supervised learning, so the overall performance is measured as $L(D^{\text{test}}, M_{\text{MT}}) = \frac{1}{|D^{\text{test}}|}\sum_{\mathbf{z}_i \in D^{\text{test}}} \ell(y_i, M_{\text{MT}}(\mathbf{x}_i))$, where $\mathbf{z}_i = (\mathbf{x}_i, y_i)$ is a labeled sample and $\ell(\cdot,\cdot)$ is some appropriate loss function such as the $0-1$ or cross-entropy loss.

We then posit that there exists a subset D^{hard} of D^{test} on which MT is aiming to improve their performance, leading them to use external data augmentation. We explain how D^{hard} is constructed in specific settings in §4.1. Each DO has a dataset D_i which could aid MT in improving their performance on D^{hard}. However, due to the challenges highlighted in §2.1, the DOs do not share all of their data with MT. Rather, they share a small subset D^{useful} of up to size k, which we call the *budget*. If D^{useful} is able to improve the performance of MT's model, then MT and DO could potentially enter into a data-sharing agreement for additional data acquisition. This paper focuses on how each DO can identify D^{useful} and subsequently, how MT can utilize this data to improve performance and if $m > 1$, rank the DOs. The task for each DO is then:

Definition 2.1 (Task for each DO)**.** Find $D_i^{\text{useful}} \subseteq D_i$ such that $|D_i^{\text{useful}}| \leq k$ and $L(D^{\text{hard}}, M'_{\text{MT}}) \leq L(D^{\text{hard}}, M_{\text{MT}})$, where $M_{\text{MT}} = \mathcal{D}(D^{\text{MT}})$ and $M'_{\text{MT}} = \mathcal{D}(D^{\text{MT}} \cup D^{\text{useful}})$.

We summarize our key assumptions below about the data held by various entities:

A.1 There exists a subset $D^{\text{hard}} \subseteq D^{\text{test}}$ with accurate ground-truth labels on which M_{MT} performs poorly, some portion of which MT shares with the DOs.
A.2 Each participating DO has samples from at least one of the classes contained within D^{hard}.

We assume **A.1** because if MT does not share any knowledge of the difficult subset, DOs cannot share meaningful data. Further, if D^{hard} is incorrectly labeled, basic challenges regarding performance evaluation arise. **A.2** just rules out DOs with no relevant data to the task under consideration. Additionally, our algorithm accounts for the following constraints likely to be encountered in practice:

C.1 MT does not share M_{MT} with the DOs.
C.2 No DO shares their full training data with MT, *i.e.* $\forall\, i, k < |D_i|$.

C.1 addresses the reality that the model trainer is unlikely to share their local model for intellectual property and privacy reasons. In spite of the added challenge, we show in §4 that the DOs can effectively share useful data. Additionally, **C.2** arises from the fact that DO will provide just enough data to convince MT of their data utility due to privacy and economic concerns. After utility is established, MT may enter into an agreement with the best data owners. While constraints **C.1** and **C.2** enforce strict limits on model sharing and data budgets to model the most restrictive real-world scenarios, we explicitly test the boundaries of these assumptions in §4.4. Through case studies, we demonstrate that Mycroft remains highly effective even under extreme budget restrictions, varying local training configurations, and adversarial scenarios involving noisy, corrupted, or fully unlabeled datasets from the data owners.

3 Mycroft: Identifying and sharing useful data

In this section, we present Mycroft, our protocol for data sharing between the Model Trainer (MT) and distributed Data Owners (DOs). The key technical component is a data selection subroutine (DataSelect), executed locally by each DO. The goal of DataSelect is *given a budget k, select a subset of local data that maximally improves the MT's performance on a challenging evaluation set D^{hard}.* We propose three approaches to DataSelect that work under different conditions for a DO (Table 1).

In the first approach termed Adaptive Selection (§3.1), we assume that DO has access to a proxy model trained using a differentiable loss function. We then use the gradients of the loss with respect to this model and

D^{hard} to guide data selection using Orthogonal Matching Pursuit (OMP) (Pati et al., 1993). The second approach (§3.2), termed Agnostic Selection, builds a kernel matrix using D^{hard} and D_i, and depending on the kernel function used, relaxes the assumptions regarding access to loss gradients from the Adaptive Selection approach. We adapt the Randomly Pivoted Cholesky (RPC) algorithm (Chen et al., 2025) to efficiently select columns from this kernel matrix that reduce the trace norm under the Nyström approximation, effectively providing a data subset selection method. The final approach (§3.3) is a voting-based method that embeds all the data in a common feature space with a metric and finds the closest samples from D_i to D^{hard} in that feature space.

Protocol Overview: The overall procedure is summarized in Figure 1. In brief, after each `DO` receives the dataset D^{hard} from `MT`, they will use it to identify a subset D^{useful} of size k from their local training data D_i that is relevant for `MT` to predict D^{hard} correctly. The `DO` then sends D^{useful} to `MT`, who will use this data to update their model M_{MT}. The central challenge lies in designing the `DataSelect` subroutine. We now present our three approaches.

3.1 Adaptive Selection using Orthogonal Matching Pursuit

When the `DO`'s loss function $L_{\texttt{DO}}$ is differentiable, we formulate data selection as sparse approximation using loss gradients. This idea has been previously used to sub-select training data to improve the efficiency of model training (Mirzasoleiman et al., 2020; Killamsetty et al., 2021a). We term it *Adaptive Selection*, as it requires access to the model parameters and loss function.

Optimization Formulation: We formulate the problem of finding D^{useful} as obtaining a k-sparse weight vector $\mathbf{w}$ over D_i, with the weight assigned to each sample corresponding to its utility. We find the averaged gradient of the loss L computed on D^{hard}, and the gradient computed on each local sample $z_j \in D_i$. We then seek a k-sparse weight vector $\mathbf{w}$ which is the solution to the following regularized optimization problem:

$$\min_{\|\mathbf{w}\|_0 \leq k} e_\lambda(\mathbf{w}) = \min_{\|\mathbf{w}\|_0 \leq k} \left\| \sum_{\mathbf{z}_j \in D_i} \mathbf{w}_j \nabla_{\boldsymbol{\theta}} \ell(y_j, M(\mathbf{x_j})) - \nabla_{\boldsymbol{\theta}} L(D^{\text{hard}}, M) \right\|_2^2 + \lambda \|\mathbf{w}\|_2^2 . \tag{1}$$

The first term ensures that a weighted sum of the selected samples is close to the gradient of the loss on D^{hard}, while the regularization term prevents the assignment of very large weights to a single instance. The choice of the model M and its corresponding parameters $\boldsymbol{\theta}$ used to compute gradients is a critical choice.

Proxy Model Assumption: Ideally, the `DO` would identify samples that match loss gradients with respect to `MT`'s specific model, M_{MT}. However, due to constraint **C.1**, the `DO` does not have access to M_{MT}. To overcome this, we assume each `DO` utilizes a local "proxy model" (M_i with parameters $\boldsymbol{\theta}_i$) trained on their own dataset. Our hypothesis is that samples which are functionally relevant (*i.e.*, have similar gradient directions) on the proxy model will likely remain relevant for the target model, provided both are trained on a similar task. Empirically, we find this gradient transferability holds (§4.3.3), allowing the `DO` to identify useful data without ever seeing the `MT`'s private weights.

Approximate Solution via OMP: The ℓ_0-"norm" constraint renders Equation (1) NP-Hard. To tackle this issue, we use a greedy algorithm, Orthogonal Matching Pursuit (OMP) (Pati et al., 1993), to find a close approximation due to the sub-modularity of $e_\lambda(\mathbf{w})$ (Elenberg et al., 2016). OMP iteratively selects samples that maximally reduce the residual error in gradient space (detailed in Algorithm 1). The choice of which state of `DO`'s model M_i to use (typically stored as checkpoints) is explored in Figure 5. Further details are in Appendix §B

Other potential gradient-based techniques: Before settling on gradient matching using OMP, we explored various other coreset techniques (Zhou et al., 2022; Coleman et al., 2019; Toneva et al., 2018; Ducoffe & Precioso, 2018; Ren et al., 2018). Because these techniques are designed to approximate the loss of an *entire* dataset during train-from-scratch scenarios, most are not directly applicable to our setting (which

Table 1: **Comparing selection schemes:** k: number of samples D^{useful} sought by the model trainer (MT) from the data owner (DO) M: model size, d: dimension of the feature embeddings, and $|D^{\text{hard}}|$: size of the target query set. ‡Voting-based method requires access to a feature extractor for the domain.

Selection scheme	Model access (for DO)	Label access (for DO)	FLOPS	Memory
Adaptive (OMP)	necessary	necessary	$kM\lvert\mathcal{D}^{\texttt{DO}}\rvert$	$M\lvert\mathcal{D}^{\texttt{DO}}\rvert$
Agnostic (RPC)	optional	unused	$k\lvert D^{\text{hard}}\rvert\lvert\mathcal{D}^{\texttt{DO}}\rvert$	$\lvert D^{\text{hard}}\rvert\lvert\mathcal{D}^{\texttt{DO}}\rvert$
Voting-based (Feature Space)	unused‡	unused	$d\lvert D^{\text{hard}}\rvert\lvert\mathcal{D}^{\texttt{DO}}\rvert$	$d\lvert\mathcal{D}^{\texttt{DO}}\rvert$

requires finding a small subset useful for a *specific task* using a pre-trained model). We opted for OMP because other adaptable techniques either underperform our chosen approach (Mirzasoleiman et al., 2020) or are too computationally intensive to be feasible (Ren et al., 2018).

3.2 Agnostic Selection using Randomly Pivoted Cholesky

In many cases, the data owner may not have access to a model trained using a differentiable loss function. Thus we explore an *Agnostic Selection* method that can operate without access to the loss, model parameters, and even data labels. We construct a joint kernel matrix from D_i and D^{hard}, and then adapt the Randomly Pivoted Cholesky (RPC) method (Chen et al., 2025) to use guidance from D^{hard} obtain a low-rank approximation of the kernel matrix of D_i. The RPC method selects 'pivots' to reduce the approximation error, where each pivot corresponds to a sample selected from D_i (Algorithm 2). This method is flexible with respect to the choice of kernel, and we ablate over the choice in §4.3.4 (see also Table 3). Further details are in Appendix C.

Background on low rank approximation of kernel matrices: Let $K : \Omega \times \Omega \to \mathbb{R}$ be a positive definite kernel function defined on the domain Ω. Let $A, B \subset \Omega$, then $K(A, B) = (K(a, b))_{\substack{a\in A\\ b\in B}} \in \mathbb{R}^{|A|\times|B|}$ denotes the kernel matrix formed from pairwise evaluations of K. For a finite,non-empty subset $S \subset \Omega$, the Nyström approximate kernel $\hat{K}_S$ is defined as

$$\hat{K}_S(x, z) := K(x, S)(K(S, S))^{\dagger}K(S, z) \tag{2}$$

where † is the pseudo-inverse which is the usual inverse if $K(S, S)$ is invertible. For $S = \emptyset$ we define $\hat{K}_S(x, z) = 0$ for all x, z. Note that $\hat{K}_S$ is a positive semi-definite kernel parameterized by S. Furthermore, $K(x, z) = \hat{K}_S(x, z)$ if $x \in S$ or if $z \in S$. For $S \subset \Omega$, $\hat{K}_S(A, A)$ is a low-rank approximation of $K(A, A)$. One can also show that $K(A, A) \succeq \hat{K}_S(A, A)$ for any finite subsets S and A, regardless of whether $S \subseteq A$.

Background on RPC: The algorithm by Chen et al. (2025) was designed to approximately solve

$$\min_{S\subseteq A} J_0(S) = \operatorname{tr}\left(K(A, A) - \hat{K}_S(A, A)\right), \qquad \text{subject to } |S| \le k. \tag{3}$$

While the optimization problem above is combinatorial since it involves searching over $\binom{|A|}{k}$ subsets of A, Chen et al. (2025) proposed a replusive sampling heuristic, which they called RPC, to construct S iteratively as $S_{t+1} = S_t \cup \{s_t\}$ where $s_t \in A$ is drawn with relative probabilities proportional to the marginal reduction in the objective function

$$\mathbb{P}(s_t = s) \propto \Delta_t(s) := J_0(S_t) - J_0(S_t \cup s) \qquad \forall\, s \in A,$$

which can be calculated iteratively as,

$$\Delta_t(s) = \begin{cases} \frac{\left\|K(A,s)-\hat{K}_{S_t}(A,s)\right\|^2}{K(s,s)-\hat{K}_{S_t}(s,s)} & s \notin S_t \\ 0 & s \in S_t \end{cases}. \tag{4}$$

Note that $K(s, s) = \hat{K}_{S_t}(s, s)$ only when $s \in S_t$, whereby the probabilities above are well defined. The condition for $s \in S_t$ makes the above sampling scheme repulsive.

RPC-based data selection in `Mycroft`: We set up the problem of selecting D^{useful} to be that of approximating $K(D^{\text{hard}}, D^{\text{hard}})$ with $\hat{K}_{D^{\text{useful}}}(D^{\text{hard}}, D^{\text{hard}})$. Note that $D^{\text{useful}} \subset D_i$. A minor difference from the RPC framework is that $D^{\text{useful}} \not\subset D^{\text{hard}}$. However, the quantity $J(S)$ can be defined similar to J_0 as

$$D^{\text{useful}} = \arg\min_{S \subset D_i} J(S) = \operatorname{tr}\left(K(X,X) - \hat{K}_S(X,X)\right), \quad \text{subject to } |S| \leq k, \tag{5}$$

$$\text{where } X = D^{\text{hard}} \text{ or } X = D^{\text{hard}} \cup \mathcal{D}_i \tag{6}$$

Diversity in sampling: A way to introduce diversity in the search for useful samples, is to choose D^{useful} based on the choice of X as in Equation (6) which allows two choices for X used to define $J(S)$. The choice $X = D^{\text{hard}}$ only considers the error incurred on the hard samples which we call the *local* diversity setting, whereas $X = D^{\text{hard}} \cup D_i$ considers error incurred on a broader data distribution which we call the *global* diversity setting. We also ablate over this choice in our experiments, see Table 16 for details. Due to the combinatorial nature of Equation (5), we approximate the selection iteratively as $D^{\text{useful}}_{t+1} \leftarrow D^{\text{useful}}_t \cup \{s_t\}$. Chen et al. (2025) noted in their work that the choice $\mathbb{P}(s_t = s) \propto \Delta_t^{\beta}(s)$ unifies 3 sampling schemes, (i) greedy sampling ($\beta = \infty$), (ii) uniform sampling ($\beta = 0$), and (iii) repulsive (or refractory) sampling ($\beta = 1$). We ablate over $\beta \in \{0, 1, \infty\}$ in our experiments.

Label Independence & Kernel Flexibility: Since the above approach only requires access to a kernel function, it is useful especially in scenarios when `DO` does not have access to any labels. It also allows flexibility to the `DO` in choosing the kernel function K which can take into account additional information provided by `MT`. For example, if the `MT` shares the model architecture but no additional information, the `DO` can use the neural tangent kernel (Jacot et al., 2018) or the conjugate kernel (Cho & Saul, 2009) for that architecture as a proxy. If in addition, certain publicly available checkpoints are available for these architectures, the `DO` may use the so-called After-kernel from (Long, 2021) detailed in §4.3.4.

3.3 Voting-based Selection in Feature Space

The previous two methods work best in settings where `DO` has access to a proxy model or labels. However, the `DO` might lack access to a model architecture or target labels. To address this, we propose a lightweight greedy alternative that operates given access to some external feature extractor that is preferably of high-quality.

Feature-space distances: Given a pre-trained feature extractor $\phi(\cdot)$, we map every sample to a dense representation and compute a pairwise distance matrix $\Psi \in \mathbb{R}^{|D^{\text{hard}}| \times |D_i|}$ where $\Psi_{ab} = \|\phi(a) - \phi(b)\|_2$. A naive greedy approach would simply select the single nearest neighbour in D_i for every target sample in D^{hard}. However, this direct one-to-one mapping has two key issues. It can i) retrieve redundant images and ii) overfit to the specific quirks of individual samples.

Frequency-based Voting scheme: To address this issue, we construct D^{useful} using a simple voting mechanism. First, for every target sample in D^{hard}, we retrieve a large candidate pool of its K nearest neighbors from D_i. These candidate pools are then grouped based on the ground-truth class of the target samples. Within the pool constructed for each class, we then compute the selection frequency of each retrieved candidate from D_i. Finally, we select the top-$\frac{k}{|C|}$ candidates that have the highest selection frequency in each class. D^{useful} is formed by taking the union of these selected samples across all classes (algorithmic details and intuition are provided in Appendix §D and Algorithm 3).

For image datasets, we instantiate $\phi(\cdot)$ using Unicom (An et al., 2023). We select this model because its feature space is explicitly optimized for image retrieval, ensuring that the L_2 distances reflect semantic similarity rather than just pixel-level overlap.

4 Experiments

To test `Mycroft`'s effectiveness, we conducted experiments with two datasets where the presence of spurious correlations in training data causes poor performance. The controlled nature of the spurious correlations in

these datasets allows us to precisely evaluate the improvement `Mycroft` provides over standard baselines. Additional setup details are in Appendix F and further experimental results in Appendix G.

4.1 Experimental Setup

Datasets: We use two setups to simulate challenging settings where external data augmentation is required to improve performance: i) *Dogs & Wolves*: This is a custom 10-class dataset where 8 classes are drawn from Imagenet (Deng et al., 2009), and the remaining two classes (the eponymous 'Dogs' and 'Wolves') contain curated images with spurious correlations. The `MT`'s model performs poorly on these due to background shortcuts (*e.g.*, all dogs in the training set being on grass, but the test set having dogs on snow). We use a primary provider (`DO1`) with balanced data for the main results, and five additional providers (`DO2-DO6`) to evaluate ranking (detailed at §F.1.1).; ii) *Spawrious* (Lynch et al., 2023): A multi-domain dataset of dog breeds designed to test whether models rely on background shortcuts. In our setup, the `MT`'s data is completely spuriously correlated (*e.g.*, all dogs of a particular species are on the same background). To evaluate ranking, we simulate four `DO`s with different data compositions (exact sample counts are detailed in §F.1.2): `DO-Good` (large dataset with diverse backgrounds for each breed), `DO-Medium` (a moderately useful dataset containing a mix of both spurious and target samples), `DO-Bad` (harmful as it perfectly matches the `MT`'s background bias), and `DO-Easy` (a "cheat" dataset containing only the exact data needed to overcome the spurious correlation). We use `DO-Good` as the primary provider for our main results.

Model training: For our primary results, we initialize both ResNet-50 (He et al., 2016) and ViT-B/16 (Dosovitskiy et al., 2021) with ImageNet-1k weights and fine-tune them for 40 epochs using the AdamW (Loshchilov & Hutter, 2019) optimizer. Our main training recipe for `MT` after receiving D^{useful} is to *augment and retrain from scratch*. Full hyperparameter configurations and alternative training recipes are in §F.2.

Baseline methods: We consider two baseline methods: i) `random-sampling`: Samples from D_i are picked uniformly at random up till the budget k to construct D^{useful}, and ii) `full-information` where all of D_i is shared, and used to fine-tune or retrain `MT`'s model. We note that the latter generally represents an empirical upper bound on performance, although there are settings where using all of the data can degrade performance (Shen et al., 2024).

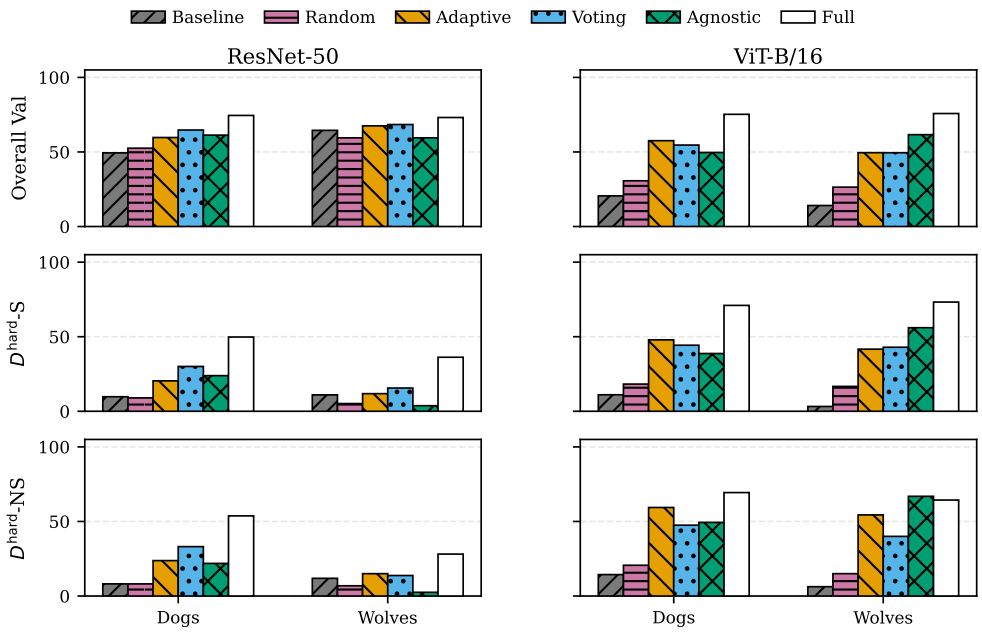
Evaluation Protocol for D^{hard}: To prevent data leakage, a portion of D^{hard} is never shared with the `DO`, which allows us to check generalisation after training with retrieved samples. In particular, we split D^{hard} into two disjoint subsets: i) Query set ($D^{\text{hard}} - S$): A random subset of D^{hard} shared with the `DO` to guide the retrieval (Referred to as "Shared"). In our experiments, we restrict the size of this shared query set to match the retrieval budget k (*i.e.*, $|D^{\text{hard}} - S| = |D^{\text{useful}}| = k$). ii) Held-Out set ($D^{\text{hard}} - NS$): The remaining misclassified samples, which are *never seen* by the `DO`. (Referred to as "Not Shared"). Good performance on $D^{\text{hard}} - NS$ indicates `Mycroft` provides generalisation. We report three evaluation metrics: i) overall validation accuracy on classes exhibiting spurious correlations, ii) accuracy on the shared subset $D^{\text{hard}} - S$, and iii) accuracy on the held-out subset $D^{\text{hard}} - NS$.

Design Choices for Agnostic Selection: There are two key design choices for the Agnostic Selection algorithm. The first is the choice of kernel used to measure similarity between data points (Table 3). The second is the pivoting strategy used. Our primary experiments use the NNGP and Laplace* kernels, which compute dot products and Manhattan-norm distances respectively of the `DO`'s model's activations. These serve as a cheap, label-independent baseline requiring no Jacobian computation. We further evaluate all kernels summarized in Table 3 — including the NTK, Gaussian, standard Laplace kernel on images and Gradient-Loss variants where we take dot product of gradient of loss function — to assess sensitivity to kernel choice across the guided selection pipeline. For pivoting strategies, we vary β in Algorithm 2.

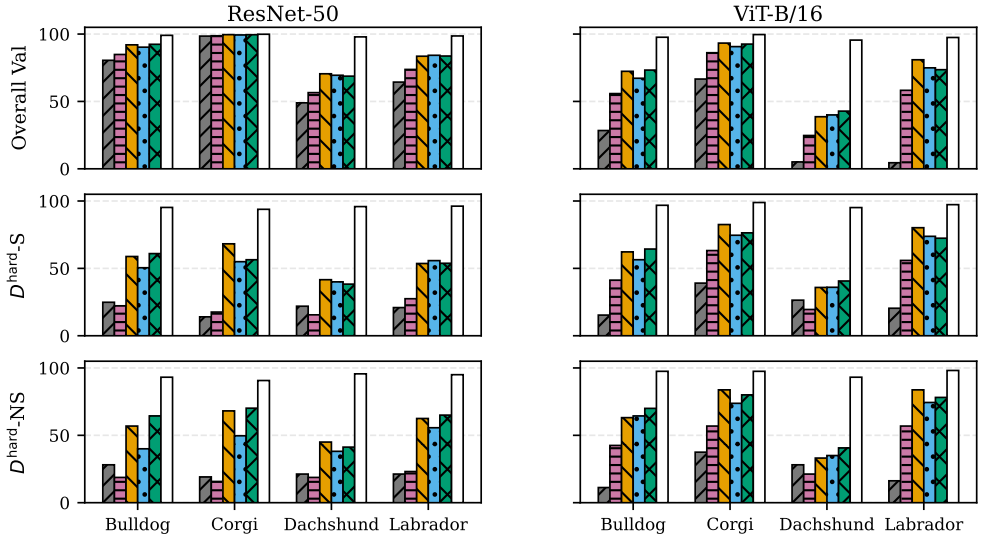
4.2 Main Results

4.2.1 Dataset 1: Addressing Spurious Correlations on Dogs & Wolves

First, we look at how well `Mycroft` can address issues in models that rely partially on spurious correlations. Fig. 2a shows the class-wise validation accuracy for both ResNet-50 and ViT-B/16. We note that for the 8



(a) Dogs & Wolves ($k = 32$).



(b) Spawrious ($k = 32$).

Figure 2: **Main Results.** (a) On Dogs & Wolves, `Mycroft` retrieves counterfactual samples, improving accuracy on both Shared (Query) and Not Shared (Held-out) sets. (b) On Spawrious, `Mycroft` consistently outperforms `random-sampling` across all classes.

other 'benign' classes, the model achieves a validation accuracy above 95%. However, the baseline models struggle on the classes with spurious correlation. Specifically, the overall baseline validation accuracy for ViT-B/16 is only 20.53% on the Dogs class and 14.09% on the Wolves class. ResNet-50 also achieves only 49.38% on Dogs and 64.49% on Wolves. These low accuracies indicate that the models are indeed heavily relying on the background shortcut for classification.

`random-sampling` provides minimal improvements in this scenario. Since the primary provider(`DO-1`) has a dataset of 4,060 samples containing a mix of natural and spurious images (see §F.1.1), random selection

predominantly retrieves "easy" samples (*e.g.*, dogs on grass) that reinforce the model's learned shortcuts. Conversely, `Mycroft`, by design, finds samples similar to D^{hard}. For ViT-B/16, Adaptive Selection pushes the validation accuracy of the class Wolves from 14.09% up to 49.51%, while Agnostic Selection pushes to 62% using a small budget of just 32 retrieved samples per class. Further, we note that each `DataSelect` method improves performance over the baseline, and handily outperforms `random-sampling`. We also see this improvement directly on the held-out hard set, which indicates that the model is not simply overfitting to samples similar to the query set.

4.2.2 Dataset 2: Improving Multi-Domain Generalisation on Spawrious

We next check how well `Mycroft` works on a more challenging dataset where *all* the classes have spurious correlations. In this experiment, we use `DO-Good`, a data owner with high-quality, diverse data that can actually help break the spurious correlations. Results where we rank different data owners are in §4.4.1.

As shown in Fig. 2b, `Mycroft` noticeably shrinks the gap to the `full-information` baseline for ResNet-50 across all four breeds. For example, on the Labrador class, Adaptive Selection achieves 86.11% accuracy—drastically outperforming `random-sampling` (75.25%) and approaching the `full-information` upper bound (99.77%). The improvement is even starker on ViT-B/16, where `random-sampling` performs poorly, sometimes even reducing performance. This happens since `random-sampling` selects a noisy mix of redundant images. We again note that the performance improves on the held-out portion $D^{\text{hard}} - NS$ as well.

4.2.3 Runtime analysis

We measured the computational overhead of the `DataSelect` subroutines on a single NVIDIA L40S GPU. A detailed complexity analysis is provided in Table 1. For the Dogs & Wolves dataset on the primary data owner (`DO-1`) with per class budget of 32 samples, the approximate runtime for each subroutine is as follows:Voting-based selection (Unicom) takes approximately 20 seconds, and Adaptive Selection (OMP) takes approximately 160 seconds. Agnostic Selection (RPC) is highly efficient, with an approximate runtime of under 10 seconds when utilizing the Laplace kernel.

4.3 Ablations

4.3.1 Varying budgets

We varied the retrieval budget $k \in \{8, 16, 32, 64\}$ to assess performance scaling. Table 13 and Table 5 compare all three methods (detailed in §G.2). For a low sampling budget of $k = 8$, `Mycroft` consistently outperforms `random-sampling`. For example, for Spawrious on ResNet-50, Agnostic Selection achieves 77.68% accuracy on Labradors versus 62.12% for `random-sampling`. Further, as the budget k increases to 64, our methods approach the `full-information` ceiling much faster than `random-sampling`. On the ViT-B/16 architecture for the Dogs & Wolves dataset, Adaptive Selection (OMP) at $k = 32$ often matches or exceeds the performance of `random-sampling` at $k = 64$, effectively halving the size of the data that the `DO` must share.

4.3.2 Varying training configurations

We evaluated `Mycroft` across seven distinct training configurations involving variations in the number of epochs (from 20 to 120 epochs), optimizer types (AdamW (PyTorch Contributors, 2023a) versus SGD (PyTorch Contributors, 2023c)), learning rate schedulers (`CosineAnnealingLR`, `CosineAnnealingWarmRestarts`, `ReduceLROnPlateau` (PyTorch Contributors, 2023b)), and initialisation strategies (training-from-scratch versus fine-tuning). In all cases, `Mycroft` consistently outperformed `random-sampling`, confirming the results are not dependent on a specific training recipe (detailed in §F.2).

4.3.3 Adaptive Selection: Impact of proxy model used by `DO`

Here, we evaluate the impact of the choice of the proxy model used by a `DO` for Adaptive Selection using OMP. In particular, we test the cases when the `MT` uses a ViT-B/16 and the `DO` uses a ResNet-50 proxy, and *vice versa*. Interestingly, gradient information transfer proves to be excellent: in the specific case where the `MT`

uses a ResNet-50 model and the DO computes gradients using a ViT-B/16 proxy, Adaptive Selection enhances the MT's overall validation accuracy on the Spawrious dataset (DO-Good) from an initial 71.70% to 85.38%. This demonstrates that Mycroft works effectively regardless of architectural mismatch (detailed in §G.4).

4.3.4 Agnostic Selection: Impact of choice of kernel and pivoting

As seen in §4.1, we performed experiments with different type of kernels. Furthermore we also modified the sampling to a local error sampling for more focus on D^{hard} set in pivoting §C.2 while also performing experiments with standard global error sampling. NNGP, NTK and Laplace* with local error sampling performed better than gradient loss kernel, while also being label-independent. All ablations over the kernel choice are in Table 16.

Since our selection objective is targeted — identifying points in $D^{\texttt{DO}}$ most similar to D^{hard} — the gain signal is highly discriminative. Greedy selection, which deterministically picks the maximum-gain candidate at each step, exploits this signal more aggressively than probabilistic sampling in random and consequently achieves better performance. We also demonstrate this gradual change of performance from uniform to random to greedy sampling by the beta parameter (Table 17). These experiments with different kernel choices and pivoting strategies helped us explore the behaviour for different settings.

4.4 Case Studies

We perform two case studies to evaluate Mycroft under realistic conditions that may be encountered in practice. In the first, we simulate a data market with DOs of varying quality to check if Mycroft can recover the ground-truth ranking. In the second, we add noise to the data features and labels in separate experiments to determine reliability of data selection in noisy environments.

4.4.1 Case Study 1: Ranking in Complex Data Markets

We check if Mycroft can accurately rank data owners by their true usefulness without needing to acquire their full datasets. To measure this, we use Kendall's Rank Correlation Coefficient (τ) (Kendall, 1938) to compare our predicted utility ordering of the DOs against their actual full-information accuracy order. Detailed τ calculations, examples, and exact DO data compositions are provided in §G.5.

Dogs & Wolves: We have six different data providers (DO-1 through DO-6) containing varying amounts of useful data. Mycroft successfully ranks these sources very close to their true utility ($\tau \approx 0.73$), correctly picking out DO-1 and DO-4 as the best data providers. random-sampling, however, fails to tell the good data owners apart from the poor ones ($\tau \approx 0.06$).

Spawrious: Here we rank four data owners with very different types of data. Mycroft perfectly recovers the true full-information ranking for both architectures: for ViT-B/16 using Adaptive Selection, and for ResNet-50 using Agnostic Selection, achieving a perfect $\tau = 1.0$ in both cases. Even though DO-Easy gets high scores by perfectly mimicking the distribution of D^{hard}, Mycroft correctly identifies that the large, diverse dataset (DO-Good) is the most useful of the remaining sets. Importantly, it ranks the deceptive DO-Bad last, which allows the MT to safely filter it out and avoid the severe performance degradation seen with random-sampling (which yields only ∼28% accuracy on ViT-B/16). While random-sampling coincidentally achieves a perfect ranking ($\tau = 1.0$) on ResNet-50, it fails to meaningfully separate high-utility sources (accuracy gaps $< 0.1\%$); a detailed discussion explaining this result and the actual performance gains is provided in §G.5.

4.4.2 Case Study 2: Robustness to Data Corruption

We simulated imperfect data conditions using DO1 for the Dogs & Wolves dataset and DO-Good for Spawrious to determine the reliability of data selection in noisy environments. We evaluated two corruption scenarios: i) *Corrupted images*: The data owner's images are heavily degraded using severe image augmentations such as color jitter, Gaussian blur, and random erasing to simulate hardware, focus, and transmission failures. For the Dogs & Wolves dataset, we selectively apply these corruptions only to the 2 classes of interest while leaving the 8 ImageNet classes clean, creating a targeted feature corruption scenario. ii) *Corrupted labels*:

The DO's dataset is injected with 70% noisy labels. This is achieved by randomly selecting 70% of the samples and permuting their ground-truth labels to incorrect classes.

In both scenarios Mycroft maintains robust performance and successfully retrieves useful data points, whereas random-sampling suffers significant degradation. The complete results for these experiments are in §G.6.

5 Related Work

Data augmentation and synthetic data generation: Numerous studies have investigated ways to enhance training data quality by leveraging existing datasets through methods such as image overlay (Inoue, 2018), random erasure (Zhong et al., 2020), and common data augmentation techniques like rotation and cropping (Inoue, 2018; Zhong et al., 2020; Chlap et al., 2021; Shorten & Khoshgoftaar, 2019; Hussain et al., 2017; Feng et al., 2021). These strategies are designed to generate new samples, reduce overfitting, and improve model generalization. However, they are less effective if the test data deviates significantly from or is underrepresented in the training set. Similarly, using generative models to create additional training samples (Tripathi et al., 2019; Such et al., 2020; Jiang et al., 2024) can enhance performance, but this approach often requires substantial data to train the generative model itself. Moreover, synthetic data may fail to capture the full complexity of real-world data distributions, limiting its impact on hard sample performance.

Data augmentation using publicly available data: Researchers have also explored sourcing data from publicly available data lakes (Nargesian et al., 2023; Castelo et al., 2021; Yakout et al., 2012; Esmailoghli et al., 2021; Santos et al., 2021; Zhu et al., 2016; Castro Fernandez et al., 2019; Fernandez, 2018; Galhotra et al., 2023). For example, METAM (Galhotra et al., 2023) profiles various datasets to identify and select the most relevant ones, thereby improving the quality of training data and enhancing performance on downstream tasks. Another approach, Internet Explorer (Li et al., 2023), retrieves task-specific images from the internet to support self-supervised learning. However, these solutions rely on open access to data and are ineffective in scenarios where data access is restricted.

Selecting data without full information: Projektor (Kang et al., 2024) addresses the problem of selecting and weighting useful data owners without full information about the underlying data. Projektor assumes that some of the DO 's data is publicly available as pilot data and can predict the usefulness of the entire dataset based on this subset. This assumption may not be true in our context as the publicly available pilot data is not selected based on any knowledge about the MT 's task, and may be irrelevant for MT.

Deriving coresets for large datasets: There is also extensive research on identifying useful subsets of datasets, such as work on "coresets" (Killamsetty et al., 2021b; Kim & Shin, 2022; Guo et al., 2022), which focus on selecting a representative subset that approximates the cost function of the entire dataset. However, most of these methods are better suited for dataset compression rather than creating task-specific subsets. In our context, approximating the entire cost function is irrelevant to the model trainer's needs. Instead, we prioritize selecting a subset most relevant to the specific task.

6 Discussion

Limitations: There are 3 key limitations of Mycroft. *First*, we assume that MT already has access to a representative set D^{hard} capturing model failure cases. While this is reasonable in practice, if D^{hard} is too small, noisy, or unrepresentative, the retrieval process may fail to identify truly useful data. *Second*, we assume that theDOs share raw data directly, without using a trusted third-party or privacy-preserving mechanisms. *Finally*, all DOs are considered to behave honestly. In practice, however, a malicious DO could provide adversarially modified samples that appear useful without being so.

Future Work: There are 3 promising directions for future work. *First*, it is important to understand how much residual utility a DO's dataset has after the relevant samples are shared. A scaling law for utility from each DO will improve identification of useful DO*s*. *Second*, Mycroft needs to be made robust and private. Balancing robustness against dishonest or even malicious data owners, while ensuring data privacy is an open challenge. *Finally*, it is instructive to explore methods with less supervision such as density estimation to identify D^{hard}, as well as more tunable data selection methods for identifying D^{useful}.

References

Md Ferdous Alam, Austin Lentsch, Nomi Yu, Sylvia Barmack, Suhin Kim, Daron Acemoglu, John Hart, Simon Johnson, and Faez Ahmed. From Automation to Augmentation: Redefining Engineering Design and Manufacturing in the Age of NextGen-AI. *An MIT Exploration of Generative AI*, mar 27 2024. https://mit-genai.pubpub.org/pub/9s6690gd.

Xiang An, Jiankang Deng, Kaicheng Yang, Jaiwei Li, Ziyong Feng, Jia Guo, Jing Yang, and Tongliang Liu. Unicom: Universal and compact representation learning for image retrieval. *arXiv preprint arXiv:2304.05884*, 2023.

K. J. Arrow. *Economic Welfare and the Allocation of Resources for Invention*, pp. 219–236. Macmillan Education UK, London, 1972. ISBN 978-1-349-15486-9. doi: 10.1007/978-1-349-15486-9_13. URL https://doi.org/10.1007/978-1-349-15486-9_13.

Sonia Castelo, Rémi Rampin, Aécio Santos, Aline Bessa, Fernando Chirigati, and Juliana Freire. Auctus: a dataset search engine for data discovery and augmentation. *Proc. VLDB Endow.*, 14(12):2791–2794, jul 2021. ISSN 2150-8097. doi: 10.14778/3476311.3476346. URL https://doi.org/10.14778/3476311.3476346.

Raul Castro Fernandez. Data-sharing markets: Model, protocol, and algorithms to incentivize the formation of data-sharing consortia. *Proc. ACM Manag. Data*, 1(2), June 2023. doi: 10.1145/3589317. URL https://doi.org/10.1145/3589317.

Raul Castro Fernandez, Jisoo Min, Demitri Nava, and Samuel Madden. Lazo: A cardinality-based method for coupled estimation of jaccard similarity and containment. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pp. 1190–1201, 2019. doi: 10.1109/ICDE.2019.00109.

Yifan Chen, Ethan N. Epperly, Joel A. Tropp, and Robert J. Webber. Randomly pivoted cholesky: Practical approximation of a kernel matrix with few entry evaluations. *Communications on Pure and Applied Mathematics*, 78(5):995–1041, 2025. doi: https://doi.org/10.1002/cpa.22234. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.22234.

Phillip Chlap, Hang Min, Nym Vandenberg, Jason Dowling, Lois Holloway, and Annette Haworth. A review of medical image data augmentation techniques for deep learning applications. *Journal of Medical Imaging and Radiation Oncology*, 65(5):545–563, 2021.

Youngmin Cho and Lawrence Saul. Kernel methods for deep learning. *Advances in neural information processing systems*, 22, 2009.

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.

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

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021.

Melanie Ducoffe and Frederic Precioso. Adversarial active learning for deep networks: a margin based approach. *arXiv preprint arXiv:1802.09841*, 2018.

Ethan R Elenberg, Rajiv Khanna, Alexandros G Dimakis, and Sahand Negahban. Restricted strong convexity implies weak submodularity. *arXiv preprint arXiv:1612.00804*, 2016.

Hady Elsahar and Matthias Gallé. To annotate or not? predicting performance drop under domain shift. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2163–2173, 2019.

Mahdi Esmailoghli, Jorge-Arnulfo Quiané-Ruiz, and Ziawasch Abedjan. Cocoa: Correlation coefficient-aware data augmentation. In *International Conference on Extending Database Technology*, 2021. URL https://api.semanticscholar.org/CorpusID:232283631.

Steven Y Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. A survey of data augmentation approaches for nlp. *arXiv preprint arXiv:2105.03075*, 2021.

Raul Castro Fernandez. *Aurum: a story about research taste*, pp. 387–391. Association for Computing Machinery and Morgan & Claypool, 2018. ISBN 9781947487192. URL https://doi.org/10.1145/3226595.3226631.

Sainyam Galhotra, Yue Gong, and Raul Castro Fernandez. Metam: Goal-oriented data discovery, 2023.

GDPR. General data protection regulation, Oct 2021. URL https://gdpr-info.eu/art-5-gdpr/.

Chengcheng Guo, Bo Zhao, and Yanbing Bai. Deepcore: A comprehensive library for coreset selection in deep learning. In *International Conference on Database and Expert Systems Applications*, pp. 181–195. Springer, 2022.

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

Zeshan Hussain, Francisco Gimenez, Darvin Yi, and Daniel Rubin. Differential data augmentation techniques for medical imaging classification tasks. In *AMIA annual symposium proceedings*, volume 2017, pp. 979. American Medical Informatics Association, 2017.

Hiroshi Inoue. Data augmentation by pairing samples for images classification. *arXiv preprint arXiv:1801.02929*, 2018.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Xi Jiang, Shinan Liu, Aaron Gember-Jacobson, Arjun Nitin Bhagoji, Paul Schmitt, Francesco Bronzino, and Nick Feamster. Netdiffusion: Network data augmentation through protocol-constrained traffic generation. In *Abstracts of the 2024 ACM SIGMETRICS/IFIP PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS/PERFORMANCE '24, pp. 85–86, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400706240. doi: 10.1145/3652963.3655071. URL https://doi.org/10.1145/3652963.3655071.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

Feiyang Kang, Hoang Anh Just, Anit Kumar Sahu, and Ruoxi Jia. Performance scaling via optimal transport: Enabling data selection from partially revealed sources. *Advances in Neural Information Processing Systems*, 36, 2024.

M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1-2):81–93, 06 1938. ISSN 0006-3444. doi: 10.1093/biomet/30.1-2.81. URL https://doi.org/10.1093/biomet/30.1-2.81.

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

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

Yeachan Kim and Bonggun Shin. In defense of core-set: A density-aware core-set selection for active learning. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 804–812, 2022.

Alexander Cong Li, Ellis Langham Brown, Alexei A Efros, and Deepak Pathak. Internet explorer: Targeted representation learning on the open web. In *International Conference on Machine Learning*, pp. 19385–19406. PMLR, 2023.

Shinan Liu, Francesco Bronzino, Paul Schmitt, Arjun Nitin Bhagoji, Nick Feamster, Hector Garcia Crespo, Timothy Coyle, and Brian Ward. Leaf: Navigating concept drift in cellular networks. *Proc. ACM Netw.*, 1 (CoNEXT2), September 2023. doi: 10.1145/3609422. URL https://doi.org/10.1145/3609422.

Philip M Long. Properties of the after kernel. *arXiv preprint arXiv:2105.10585*, 2021.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Aengus Lynch, Gbètondji J-S Dovonon, Jean Kaddour, and Ricardo Silva. Spawrious: A benchmark for fine control of spurious correlation biases, 2023.

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

Fatemeh Nargesian, Ken Pu, Bahar Ghadiri-Bashardoost, Erkang Zhu, and Renée J. Miller. Data lake organization. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):237–250, 2023. doi: 10.1109/TKDE.2021.3091101.

Yagyensh Chandra Pati, Ramin Rezaiifar, and Perinkulam Sambamurthy Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of 27th Asilomar conference on signals, systems and computers*, pp. 40–44. IEEE, 1993.

PyTorch Contributors. torch.optim.adamw. https://docs.pytorch.org/docs/2.0/generated/torch.optim.AdamW.html, 2023a.

PyTorch Contributors. torch.optim.lr_scheduler. https://docs.pytorch.org/docs/2.0/optim.html, 2023b.

PyTorch Contributors. torch.optim.sgd. https://docs.pytorch.org/docs/2.0/generated/torch.optim.SGD.html, 2023c.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

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

Aécio Santos, Aline Bessa, Fernando Chirigati, Christopher Musco, and Juliana Freire. Correlation sketches for approximate join-correlation queries. In *Proceedings of the 2021 International Conference on Management of Data*, SIGMOD '21, pp. 1531–1544, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383431. doi: 10.1145/3448016.3458456. URL https://doi.org/10.1145/3448016.3458456.

Judy Hanwen Shen, Inioluwa Deborah Raji, and Irene Y. Chen. The data addition dilemma. In Kaivalya Deshpande, Madalina Fiterau, Shalmali Joshi, Zachary Lipton, Rajesh Ranganath, and Iñigo Urteaga (eds.), *Proceedings of the 9th Machine Learning for Healthcare Conference*, volume 252 of *Proceedings of Machine Learning Research*. PMLR, 16–17 Aug 2024. URL https://proceedings.mlr.press/v252/shen24a.html.

Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pp. 510–526. Springer, 2016.

Kayte Spector-Bagdady, Amanda Fakih, Chris Krenz, Erica E Marsh, and J Scott Roberts. Genetic data partnerships: academic publications with privately owned or generated genetic data. *Genetics in Medicine*, 21(12):2827–2829, 2019.

Karin Stacke, Gabriel Eilertsen, Jonas Unger, and Claes Lundström. Measuring domain shift for deep learning in histopathology. *IEEE Journal of Biomedical and Health Informatics*, 25(2):325–336, 2021. doi: 10.1109/JBHI.2020.3032060.

Felipe Petroski Such, Aditya Rawal, Joel Lehman, Kenneth Stanley, and Jeffrey Clune. Generative teaching networks: Accelerating neural architecture search by learning to generate synthetic training data. In *International Conference on Machine Learning*, pp. 9206–9216. PMLR, 2020.

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.

Shashank Tripathi, Siddhartha Chandra, Amit Agrawal, Ambrish Tyagi, James M Rehg, and Visesh Chari. Learning to generate synthetic data via compositing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 461–470, 2019.

Mohamed Yakout, Kris Ganjam, Kaushik Chakrabarti, and Surajit Chaudhuri. Infogather: entity augmentation and attribute discovery by holistic matching with web tables. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, SIGMOD '12, pp. 97–108, New York, NY, USA, 2012. Association for Computing Machinery. ISBN 9781450312479. doi: 10.1145/2213836.2213848. URL https://doi.org/10.1145/2213836.2213848.

Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 13001–13008, 2020.

Xiao Zhou, Renjie Pi, Weizhong Zhang, Yong Lin, Zonghao Chen, and Tong Zhang. Probabilistic bilevel coreset selection. In *International Conference on Machine Learning*, pp. 27287–27302. PMLR, 2022.

Erkang Zhu, Fatemeh Nargesian, Ken Q. Pu, and Renée J. Miller. Lsh ensemble: internet-scale domain search. *Proc. VLDB Endow.*, 9(12):1185–1196, aug 2016. ISSN 2150-8097. doi: 10.14778/2994509.2994534. URL https://doi.org/10.14778/2994509.2994534.

a

Reproducibility Statement

We have taken several steps to ensure the reproducibility of our work:

1. **Code Availability:** We open-source the code for all our experiments anonymously at https://anonymous.4open.science/r/Mycroft/. This repository contains comprehensive scripts that allow users to run all configurations and training recipes mentioned in the paper out-of-the-box.

2. **Data Management:** We use a mix of public and newly curated datasets detailed in Section 4. The provided code includes automated scripts to download and preprocess these datasets if they are not detected in the target directory. We will also release the curated Dogs & Wolves dataset to the community.

3. **Compute Details:** We have detailed the compute requirements in Appendix F. All experiments reported in this paper were conducted on **NVIDIA L40s GPUs**.

Appendix

In this Appendix, we aim to (i) provide proofs for the optimality of `Mycroft` (ii) provide more details about the datasets, algorithms and models used to obtain the results in the main body of the paper (iii) present additional experiments to further validate the discoveries in the main body of the paper. The Appendix is organized as follows:

1. Summary of symbols and notations used (§A)

2. Further details about the Adaptive Selection method using OMP (§B)

3. Further details about the Agnostic Selection method using RPC (§C)

4. Further details about the Voting-based Selection method using Feature Space Distances (§D)

5. Runtime analysis (§E)

6. Further details about the experiment setup including the datasets and models used (§F)

7. Additional results and ablation studies (§G)

A Symbols and Notations

Table 2: Table of notations used in the paper.

Symbol	Description
`DO`	Data Owner
`MT`	Model Trainer
$D^{\texttt{MT}}$	`MT`'s dataset
$M_{\texttt{MT}}$	`MT`'s model
D_i	i^{th} `DO`'s dataset
D^{useful}	$\subset D_i$ retrieved by `Mycroft` or `random-sampling`
D^{hard}	$\subset D^{\text{test}}$ which is incorrectly classified and is shared with the `DO`s

B Adaptive Selection: Details on the Orthogonal Matching Pursuit approach

Algorithm 1 adapts the Orthogonal Matching Pursuit (OMP) algorithm from Killamsetty et al. (2021a) to sub-select a sample from $D^{\texttt{DO}}$.

Algorithm 1 Orthogonal Matching Pursuit (OMP)

Require: hard samples D^{hard}, Source dataset $D^{\texttt{DO}}$, Loss function L, Model parameters $\boldsymbol{\theta}$, regularization coefficient λ, subset size k, tolerance ϵ

$D^{\text{useful}} \leftarrow \emptyset$

$e_\lambda(\mathbf{w}) = \left\| \sum_{\mathbf{z}_j \in D^{\texttt{DO}}} \mathbf{w}_j \nabla_{\boldsymbol{\theta}} \ell(y_j, M(\mathbf{x_j})) - \nabla_{\boldsymbol{\theta}} L(D^{\text{hard}}, M) \right\|_2^2 + \lambda \left\| \mathbf{w} \right\|_2^2.$

$\mathbf{r} \leftarrow \nabla_{\mathbf{w}} e_\lambda(\mathbf{w})|_{\mathbf{w}=\mathbf{0}}$

while $|D^{\text{useful}}| \leq k$ and $e_\lambda(\mathbf{w}^*) \geq \epsilon$ **do**

  $m \leftarrow \arg\max_j |\mathbf{r}_j|$

  $D^{\text{useful}} \leftarrow D^{\text{useful}} \cup \{m\}$

  $\mathbf{w}^* \leftarrow \arg\min_{\mathbf{w}} e_\lambda(\mathbf{w})$

  $\mathbf{r} \leftarrow \nabla_{\mathbf{w}} e_\lambda(\mathbf{w}^*)$

end while

return D^{useful}, $\mathbf{w}^*$

C Agnostic Selection: Details on the Randomly Pivoted Cholesky scheme

Algorithm 2 Agnostic selection using the Randomly Pivoted Cholesky scheme

Require: Kernel function K, data owners dataset $D^{\texttt{DO}}$, hard samples D^{hard}, budget k

Require: diversity $\in \{\textsf{global}, \textsf{local}\}$, refractoriness $\beta \in \{0(\textsf{uniform}), 1(\textsf{refractory}), \infty(\textsf{greedy})\}$

Initialize support set $D^{\text{useful}} \leftarrow \emptyset$

for $t = 1, 2, \ldots, k$ **do**

  Update definition of $\hat{K}_{D^{\text{useful}}}$. See equation 2.

  for each $x \in D^{\texttt{DO}} \backslash D^{\text{useful}}$ **do**

    Compute marginal gain from adding sample x:

$$\Delta(x) = \frac{\left\| K(D^{\text{hard}}, x_i) - \hat{K}_{D^{\text{useful}}}(D^{\text{hard}}, x) \right\|^2}{K(x,x) - \hat{K}_{D^{\text{useful}}}(x,x)} + \mathbb{1}_{\{\text{diversity}==\text{`global'}\}} \cdot \frac{\left\| K(\mathcal{D}^{\texttt{DO}}, x) - \hat{K}_{D^{\text{useful}}}(\mathcal{D}^{\texttt{DO}}, x) \right\|^2}{K(x,x) - \hat{K}_{D^{\text{useful}}}(x,x)}$$

  end for

  Select x^* with probability $\dfrac{\Delta^\beta(x)}{\sum_{x \in \mathcal{D}^{\texttt{DO}} \backslash D^{\text{useful}}} \Delta^\beta(x)}$

  $D^{\text{useful}} \leftarrow D^{\text{useful}} \cup \{x^*\}$

end for

return D^{useful}

We go over some details of the Randomly Pivoted Cholesky (RPC) algorithm by Chen et al. (2025). We work with general sets $X, Z \subset \Omega$ the domain. In the `Mycroft` setup, one would apply RPC using $X = \mathcal{D}^{\texttt{DO}}$ and $Z = D^{\text{hard}}$ to obtain $S = D^{\text{useful}}$.

Notation: To simplify notation, for any kernel, we introduce $K(A) := K(A, A)$ and $K^{-1}(A) := K(A, A)^{-1}$.

Recall the definition of the Nyström approximation of the kernel function $\hat{K}_S$ from Equation (2).

Let $\phi : \Omega \to \mathbb{H}$ be a feature map for the kernel function K, for some hilbert space $\mathbb{H}$, *i.e.*, $K(x, z) = \langle \phi(x), \phi(z) \rangle_{\mathbb{H}}$. Consider the matrix $\Phi(A) \in \mathbb{H}^{|A|}$, i.e., if $\mathbb{H} = \mathbb{R}^p$ then $\Phi(A) \in \mathbb{R}^{|A| \times p}$. Note that $K(A) = \Phi(A)\Phi(A)^\top$, whereas $\hat{K}_S(A) = \Phi(A)\Phi(S)^\top(\Phi(S)\Phi(S)^\top)^{-1}\Phi(S)\Phi(A)^\top = \Phi(A)\Phi(S)^\dagger\Phi(S)\Phi(A)^\top$, where $\dagger$ indicates a the pseudo-inverse of an operator.

Proposition C.1. *For any finite non-empty sets S, X we have $K(X) \succeq \hat{K}_S(X)$.*

Proof. Let $\phi : \Omega \to \mathbb{H}$ for some Hilbert space $\mathbb{H}$ be any feature map for K, i.e., $K(x, z) = \langle \phi(x), \phi(z) \rangle_{\mathbb{H}}$. Let $\Phi(X) = [\phi(x_1) \mid \phi(x_2) \mid \ldots \mid \phi(x_n)]^\top \in \mathbb{H}^n$ denote the matrix[2] obtained by stacking the features $\phi(x_i)$ row-wise. Then

$$\begin{aligned} K(X) - \hat{K}_S(X) &= \Phi(X)\Phi(X)^\top - \Phi(X)\Phi(S)^\top(\Phi(S)\Phi(S)^\top)^\dagger\Phi(S)\Phi(X)^\top \\ &= \Phi(X)\left(\mathrm{id} - \Phi(S)^\top(\Phi(S)\Phi(S)^\top)^\dagger\Phi(S)\right)\Phi(X)^\top \end{aligned}$$

The above matrix is positive semidefinite if the following operator

$$\mathrm{id} - \Phi(S)^\top(\Phi(S)\Phi(S)^\top)^\dagger\Phi(S) : \mathbb{H} \to \mathbb{H}$$

is positive semidefinite. Note however that this operator is idempotent. It is in-fact the projection operator onto $\mathrm{null}(\Phi(S)^\top) \subset \mathbb{H}$. Idempotent operators are always positive semidefinite. This concludes the proof. □

Proposition C.2. *Let $S \subseteq T$, then $\hat{K}_T(A) \succeq \hat{K}_S(A)$*

Proof. We first show the above property for $T = S \cup \{s\}$ where $s \notin S$. The case of $s \in S$ is trivial. The more general result follows immediately.

We start by noting the following result obtained via the Sherman-Morrison-Woodbury formula

$$K^{-1}(T) = \begin{bmatrix} K(S) & K(S,s) \\ K(s,S) & K(s) \end{bmatrix}^{-1} = \begin{bmatrix} K^{-1}(S) & \\ & 0 \end{bmatrix} + \frac{1}{K(s) - \hat{K}_S(s)} \begin{bmatrix} K^{-1}(S)K(S,s) \\ -1 \end{bmatrix}^{\otimes 2} \tag{7}$$

Using this,

$$\begin{aligned} \hat{K}_T(A) - \hat{K}_S(A) &= K(A,T)K^{-1}(T)K(T,A) - K(A,S)K^{-1}(S)K(S,A) \\ &= \frac{1}{K(s) - \hat{K}_S(s)} \begin{bmatrix} K(s,S) & K(s) \end{bmatrix} \begin{bmatrix} K^{-1}(S)K(S,s) \\ -1 \end{bmatrix} \begin{bmatrix} K(s,S)K^{-1}(S) & -1 \end{bmatrix} \begin{bmatrix} K(S,s) \\ K(s) \end{bmatrix} \\ &= K(s) - \hat{K}_S(s) \geq 0 \end{aligned}$$

where the last inequality follows from Proposition C.1. □

Monotonicity: Due to the above result, solving the optimization problem (Equation (3))using an iterative update rule $S_{t+1} \leftarrow S_t \cup \{s_t\}$ will yield in monotonic reduction of the objective function. We acknowledge that the problem is highly non-convex and hence we can get stuck in local minima.

C.1 Setup with `Mycroft`

Consider the setting where we have a pretrained neural network $f : \mathbb{R}^d \times \mathbb{R}^p \to \mathbb{R}^c$ so that $\mathbb{R}^d \ni x \mapsto f(x, \theta) \in \mathbb{R}^c$ where θ are model parameters. The so-called After Kernel Long (2021) corresponding to this network is defined as

$$K(x, z) = \langle \mathbf{J}_\theta f(x, \theta), \mathbf{J}_\theta f(z, \theta) \rangle_{\mathbb{R}^{c \times p}} \qquad \text{for all } x, z \in \mathbb{R}^d$$

The RPC scheme can work with difference choices of kernels. For multi-output function spaces, the 'correct' framework would have been to use vector-valued kernels. In fact, our setup extends to those naturally, however we have focused only on scalar kernels, for the sake of simplicity. We list the kernels that we have considered in this work in Table 3.

We also consider normalized and homogenized counterparts for the given kernel K

$$K_{\mathsf{norm}}(x, z) = K\left(\tfrac{x}{\|x\|}, \tfrac{z}{\|z\|}\right), \qquad K_{\mathsf{homo}}(x, z) = \|x\| \cdot \|z\| \cdot K_{\mathsf{norm}}(x, z).$$

[2]We abused notation a bit here. For example if $\mathbb{H} \equiv \mathbb{R}^p$ then $\mathbb{H}^n \equiv \mathbb{R}^{n \times p}$

Table 3: **Kernel functions used in our experiments with RPC.** $f_{\texttt{DO}}(\cdot;\theta)$ refers to the DO's model with parameters θ. $f_{\texttt{DO}}^{(-1)}(x;\theta)$ denotes the activations at the penultimate layer for an input x. $\mathbf{u}(x)$ refers to any unsupervised representation, for example those based on CLIP-based model such as Unicom An et al. (2023). $\|\cdot\|$ is the ℓ_2 norm.

Kernel	**Label independent**	**Model independent**	**Definition** $K(x,z)$
Linear	✓	✓	$\langle x, z\rangle$
Gaussian	✓	✓	$\exp\left(-\|x-z\|^2/2\sigma^2\right)$.
Laplacian	✓	✓	$\exp\left(-\|x-z\|/\sigma\right)$
Laplacian*	✓	✗	$\exp\left(-\|f_{\texttt{DO}}(x;\theta)-f_{\texttt{DO}}(z;\theta)\|\right)$
Tangent(NTK)	✓	✗	$\langle \mathbf{J}_\theta f_{\texttt{DO}}(x;\theta), \mathbf{J}_\theta f_{\texttt{DO}}(z;\theta)\rangle$
Conjugate(NNGP)	✓	✗	$\left\langle f_{\texttt{DO}}^{(-1)}(x;\theta), f_{\texttt{DO}}^{(-1)}(z;\theta)\right\rangle$
GradMatch(Gradloss)	✗	✗	$\langle \nabla_\theta \ell(f_{\texttt{DO}}(x;\theta), y_x), \nabla_\theta \ell(f_{\texttt{DO}}(z;\theta), y_z)\rangle$
Unicom	✓	✗	$\langle \mathbf{u}(x), \mathbf{u}(z)\rangle$

When normalizing, to protect from divide-by-zero errors, we use $x \mapsto \frac{x}{\|x\|+\varepsilon}$ for some small ε, typically 10^{-8}.

Model-independent kernels are malleable. If we have feature maps ψ, we can extend the kernel

$$K_\psi(x,z) = K(\psi(x), \psi(z)).$$

The Tangent kernel, Conjugate kernel, GradMatch, and Unicom kernel are all obtained in this manner.

C.2 Subset selection scheme based on local or global error

The RPC scheme updates the selected subset iteratively as $S_{t+1} \leftarrow S_t + \cup\{s_t\}$ where $\mathbb{P}(s_t = s) \propto \Delta_t^\beta(s)$ for all $s \in X$. We consider the following two definitions of Δ_t above.

1. (`diversity`==global) $\Delta_t(s; X\cup Z) := \mathrm{tr}(\hat{K}_{S_t}(X\cup Z) - K(X\cup Z)) - \hat{K}_{S_t\cup\{s\}}(X\cup Z)) - \mathrm{tr}(K(X\cup Z), \forall s \in X$.
2. (`diversity`==local) $\Delta_t(s; Z) := \mathrm{tr}(K(Z) - \hat{K}_{S_t}(Z)) - \mathrm{tr}(K(Z) - \hat{K}_{S_t\cup\{s\}}(Z))$ for all $s \in X$.

Since the trace is a linear operation which decomposes along block diagonals, we have

$$\Delta_t(s; Z) = \mathrm{tr}(\hat{K}_{S_t\cup\{s\}}(Z) - \hat{K}_{S_t}(Z)), \quad \text{and}$$
$$\Delta_t(s; X\cup Z) = \Delta_t(s; Z) + \Delta_t(s; X), \quad \text{for all } s \in X$$

C.3 Efficient iterative selection

Obtaining $\hat{K}_{S_t}$ can cost $O(t^3)$ if we were to calculate $K^{-1}(S_t)$ naively at each step. However, we can use dynamic programing via the block-inverse relation in Equation (7).

$$K^{-1}(S_t \cup \{s\}) = \begin{bmatrix} K^{-1}(S_t) & \\ & \end{bmatrix} + \frac{1}{K(s) - \hat{K}_{S_t}(s)} \begin{bmatrix} K^{-1}(S_t)K(S_t, s) \\ -1 \end{bmatrix}^{\otimes 2}$$

Using this we can obtain

$$\Delta_t(s; Z) = \frac{\left\| K(s, Z) - \hat{K}_{S_t}(s, Z) \right\|^2}{K(s) - \hat{K}_{S_t}(s)}$$

and similarly $\Delta_t(s; X)$. Note that the above quantity is always well defined. $K(s) = \hat{K}_{S_t}(s)$ if and only if $s \in S_t$, in which case the numerator is 0^2. For $K(s) - \hat{K}_{S_t}(s) = \varepsilon \to 0$, the numerator behaves as ε^2.

Table 4: Resources required to select a subset of size k, where $n = |\mathcal{D}^{\texttt{DO}}|$ and $m = |D^{\text{hard}}|$ using RPC

version	sampling rule	flops	memory	remarks
compute efficient	global	$k(m+n)$	$k^2 + kn$	store & update $K^{-1/2}(S_t)K(S_t, \mathcal{D}^{\texttt{DO}})$
	local	km	$k^2 + km$	store & update $K^{-1/2}(S_t)K(S_t, D^{\text{hard}})$
memory efficient	global	$k(m+n)$	k^2	store & update $K^{-1}(S_t)$, recompute $K(\mathcal{D}^{\texttt{DO}}, x)$ per step
	local	km	k^2	store & update $K^{-1}(S_t)$, recompute $K(D^{\text{hard}}, x)$ per step

C.4 Utility Analysis using RPC

Since the RPC based scheme essentially approximates the matrices $K(X \cup Z, X \cup Z)$ in the case of global sampling or $K(Z, Z)$ in the case of local sampling, a natural notion of utility can be captured via the approximation error on a validation set $\tilde{Z}$.

$$\mathcal{U}(S_t) = \text{tr}\left(K(\tilde{Z}, \tilde{Z}) - \hat{K}_{S_t}(\tilde{Z}, \tilde{Z})\right)$$

D Voting-based Selection: Details on Feature Space Selection

When a Data Owner (`DO`) lacks access to a proxy model or labels, we rely on a pre-trained feature extractor $\phi(\cdot)$ (e.g., Unicom) to map images into a dense semantic space. To avoid the redundancy and outlier-overfitting typical of naive one-to-one nearest neighbor mappings, we implement a frequency-based voting mechanism. The intuition is that the most useful source samples will consistently appear in the local neighborhoods of multiple hard target samples. The procedure operates as follows:

1. **Distance & Pooling:** Using pairwise L_2 distances, we retrieve a localized pool of the K nearest neighbors in the `DO`'s dataset (D_i) for every target sample in D^{hard}.
2. **Class-wise Voting:** We group the target samples by their ground-truth class C. Within each group, we tally how often each D_i sample appears across the localized pools. This count acts as the sample's selection frequency.
3. **Subset Construction:** To ensure diversity and class balance, we select the top $k/|C|$ samples with the highest selection frequencies per class.

The union of these selected samples forms the final dataset, D^{useful}, as formally outlined in Algorithm 3.

E Complexity & Runtime of `Mycroft`

`Mycroft` utilizes three distinct optimization techniques: Adaptive Selection (OMP), Agnostic Selection (RPC) and Voting-based Selection. Here, we discuss the computation and memory complexity of these techniques to provide a sense of their efficiency and practicality. Voting-based operates by projecting all data points in the representation space of the Unicom model, which is based on CLIP Radford et al. (2021), and computing distances between those data points. Thus, its compute and memory requirement scale in proportion to the number of data points to be projected as each sample requires a forward pass through the model to acquire its feature representation which needs to be held in stored for computing distances with other data points. Empirically, we find that this procedure takes less than 5 minutes and requires less than 4 GB of GPU memory for each experiment we present in this paper. Adaptive Selection (OMP) requires computing gradients for each data point in D^{hard} and `DO`'s using the `DO`'s model once. In practice, we only use the gradients of the last two layers, which significantly reduces the compute and memory requirements. The gradients are then used to run the OMP algorithm which has a complexity of $\mathcal{O}(NM + Mk + k^3)$ for each of the k iterations where k is $|D^{\text{useful}}|$, M is the dimension of the gradients and N is $|D_i|$. For experiments in

Algorithm 3 Voting-based Selection in Feature Space

Require: Target subset D^{hard}, Source dataset D^{DO}, Feature extractor $\phi(\cdot)$, candidate pool size N, per-class budget k
Ensure: Retrieved subset D^{useful}

```
D^useful ← ∅
Extract dense embeddings: Φ^hard ← φ(D^hard) and Φ^DO ← φ(D^DO)
for each target class c ∈ classes(D^hard) do
    V ← {}                                   ▷ Initialize empty dictionary for candidate frequencies
    D_c^hard ← {x ∈ D^hard | class(x) = c}
    for each sample x ∈ D_c^hard do
        N_x ← Top-N nearest neighbors of φ(x) in Φ^DO using L_2 distance
        for each candidate y ∈ N_x do
            V[y] ← V[y] + 1                  ▷ Increment candidate's frequency count
        end for
    end for
    S_c ← {y ∈ V | y has one of the k highest frequencies}
    D^useful ← D^useful ∪ S_c
end forreturn D^useful
```

this paper, each experiment ran in under 10 minutes and required approximately 6 GigaBytes of memory. For Agnostic Selection (RPC), it optimizes complexity from $\mathcal{O}(N^3)$ to $\mathcal{O}(k^2N)$ in time and $\mathcal{O}(N^2)$ to $\mathcal{O}(kN)$ in memory

F Further setup details

Further details about the datasets and training process used to obtain the results in the main body of the paper are provided in this section.

F.1 Datasets

F.1.1 Dogs & Wolves dataset

Neural networks often rely on spurious correlations when trained on biased datasets. To simulate a controlled interaction between a Model Trainer (`MT`) and a Data Owner (`DO`), we curated a custom image classification dataset of Dogs and Wolves where the background acts as a strong spurious cue.

The `MT`'s data is divided into a training set and a validation set. In the training set, every image belonging to a specific class shares a similar background environment: all dogs appear on grass, and all wolves appear on snow. Because the training data lacks diverse examples (e.g., dogs on snow or wolves on grass), the `MT`'s model takes a shortcut, learning to associate the background environment with the target class rather than the animal itself. To evaluate whether the model has actually learned the correct semantic features, the validation set contains the exact opposite pairings: dogs on snow and wolves on grass. Because the `MT`'s model relies on the background shortcut, it naturally performs poorly on this validation set.

We simulate a `DO` that possesses a more diverse dataset containing animals on both background types. Because the `DO`'s data breaks the background correlation, their proxy model learns robust features rather than shortcuts. An illustration of these data splits is provided in Figure 4.

If the `MT` wants to improve their model's performance on their validation set, they must acquire external data that breaks the spurious correlation. This economic need motivates the `MT` to query the `DO`. It is important to note that while the `MT` is aware of their poor validation performance, they are oblivious to the fact that a background bias is the specific root cause.

For Case Study 1 (§4.4.1), we curated specific dataset splits for the Model Trainer (`MT`) and six different Data Owners (`DO-1` through `DO-6`) to test if `Mycroft` can accurately rank them. The utility of a `DO` depends

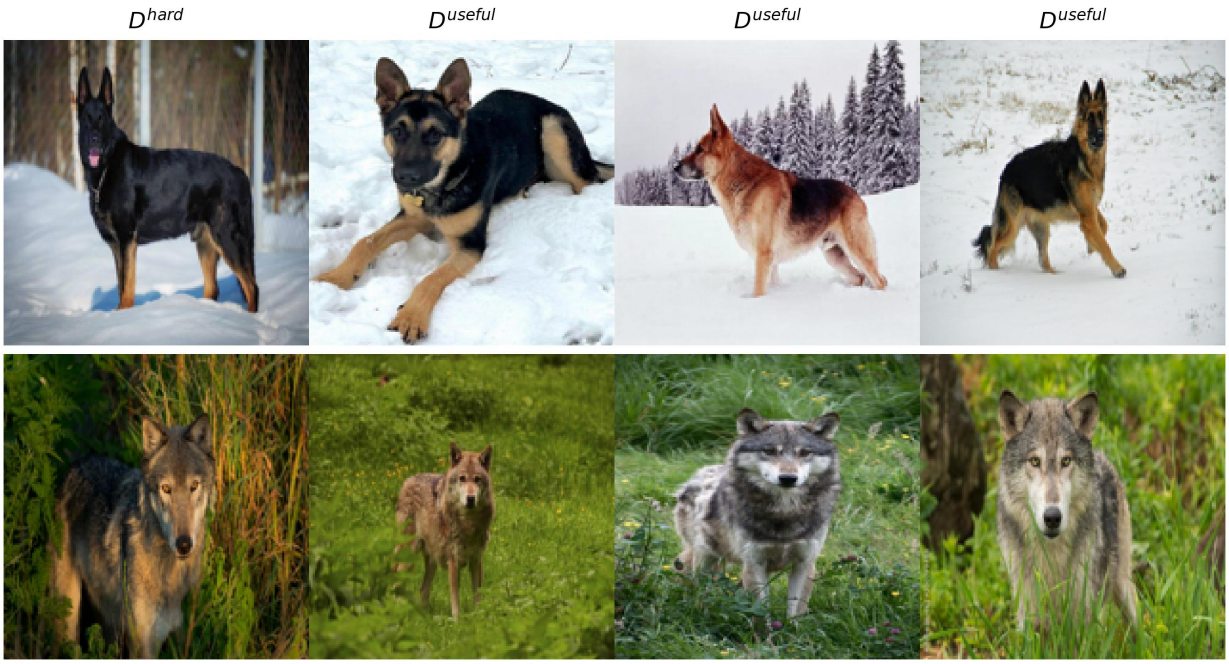


Figure 3: Top-k retrieved D^{useful} samples using Unicom for D^{hard} from the Dogs & Wolves dataset.

Figure 4: Subsets in the Dogs & Wolves dataset. The first column shows the `MT`'s training distribution (dogs on grass, wolves on snow). The second column shows the target validation distribution (dogs on snow, wolves on grass).

heavily on whether it contains images from the hard subclass (D^{hard}), that is, images that break the `MT`'s learned spurious correlations (dogs on snow or wolves on grass).

The detailed composition of the training sets for each entity is as follows. The overall dataset spans 10 total classes: 2 target classes (Dogs and Wolves) and 8 ImageNet classes (e.g., ice bear, king penguin, Persian cat, cougar, gazelle, porcupine, sea lion, mink). For the Dogs and Wolves classes, we refer to the images that follow the background shortcut as "spurious" samples, and the images that break the shortcut as D^{hard} samples(Note: For all `DO`s, any remaining samples needed to reach their total dataset size are drawn from the 8 ImageNet classes):

`MT` Datasets:

- **`MT Train:`** Contains 5,000 total training samples: 4,000 samples from the 8 ImageNet classes, and 1,000 spurious Dogs & Wolves samples. It contains 0 D^{hard} samples, forcing the model to learn the background shortcut.
- **`MT Val:`** Contains 900 validation samples that are 100% D^{hard}. This perfectly isolates the model's blind spot.

`DO` Datasets:

- **`DO-1 (Primary):`** Contains 4,060 samples. This serves as a high-quality, balanced source containing 1,760 spurious samples and 300 D^{hard} samples.
- **`DO-2:`** Contains 5,800 samples, but the distribution is highly skewed. It has 1,760 spurious samples and only 40 D^{hard} samples, making it marginally useful.
- **`DO-3:`** Contains 4,500 samples (500 spurious) and 0 D^{hard} samples.
- **`DO-4:`** Contains 4,998 samples. This dataset is highly useful for the `MT` as it contains 500 spurious samples and 498 D^{hard} samples.
- **`DO-5:`** Contains 4,700 samples (700 spurious) and 0 D^{hard} samples.
- **`DO-6:`** Contains 6,760 samples (1,760 spurious) and 0 D^{hard} samples.

As demonstrated in §4.4.1, `Mycroft` effectively navigates these fine-grained shifts. It highly ranks `DO`s containing the necessary D^{hard} data (like `DO-1` and `DO-4`) and correctly assigns lower utility to datasets that are large but lack the specific images needed to fix the model's blind spot (like `DO-6`).

F.1.2 Spawrious Dataset

To evaluate `Mycroft`'s ability to rank data owners and improve generalization, we use the Spawrious dataset (Lynch et al., 2023). We create a controlled environment where a Model Trainer (`MT`) is forced to learn a spurious correlation, and four Data Owners (`DO`s) possess datasets of varying utility. We refer to the images that follow the background shortcut as "spurious" samples, and the images that break the shortcut as D^{hard} samples.

`MT` Datasets (Sourced from `env 0`):

- **`MT Train:`** Contains 12,672 training samples that are 100% spurious. Each breed appears on a single, unique background (e.g., all Bulldogs are in the jungle), forcing the model to learn the background shortcut.
- **`MT Val:`** Contains 12,672 validation samples that are 100% D^{hard} (class-background pairings are broken). This evaluates if the model has successfully generalized beyond the background shortcut.

`DO` Datasets (Sourced from `env 1` to prevent data leakage):

- **`DO-Good (Primary):`** A massive, diverse dataset containing 60,815 samples. It contains a healthy mix of 10,136 spurious samples and 50,679 D^{hard} samples, making it highly helpful for achieving generalization.
- **`DO-Medium:`** Contains moderately useful data with 30,407 samples. It consists of 2,534 spurious samples and 27,873 D^{hard} samples.
- **`DO-Bad:`** A deceptive dataset containing 10,136 samples that are 100% spurious. This dataset has the exact same spurious correlations as the `MT`'s training set. Acquiring this data actively reinforces the `MT`'s bias.

- **DO-Easy:** A highly specialized dataset containing 10,136 samples that are 100% D^{hard}. This dataset represents a data owner whose entire repository happens to perfectly match the MT's specific target validation distribution, making it the most mathematically dense source of utility for this exact query.

This explicit breakdown of spurious versus D^{hard} data types across the DOs clarifies why Mycroft's ability to rank them in the order of DO-Good > DO-Easy > DO-Medium > DO-Bad (and its ability to actively avoid DO-Bad) is a significant improvement over random-sampling.

F.2 Training details

In this section, we provide a comprehensive overview of the training procedures, dataset constraints, and hyperparameter configurations used to train the neural networks for our computer vision tasks.

Task Difficulty and Baseline Constraints For the image classification tasks evaluated in this work, we utilize two fundamentally distinct datasets to test spurious correlations: the **Dogs & Wolves** dataset (a custom-curated dataset comprising 8 standard ImageNet classes and 2 target classes scraped from the web) and the **Spawrious** dataset (Lynch et al., 2023) (a public benchmark utilizing synthetically generated images across 4 dog breed classes).

Because we initialize our models with powerful ImageNet-1k pretrained weights, we observed that standard baseline models often perform exceptionally well when given access to the full downstream datasets, leaving little room to adequately demonstrate the value of external data augmentation. To simulate a more realistic, data-scarce environment where a Model Trainer (MT) actively needs external augmentation, we artificially constrain the MT's local data. Specifically, we restrict the MT to using only 10% of the available original training data for the target classes. This constraint reliably caps the MT's baseline accuracy at approximately 65%, creating a rigorous starting point to evaluate the efficacy of the data subsets retrieved by Mycroft.

Exploration of Training Recipes To rigorously ensure that the performance gains yielded by Mycroft are robust and not merely an artifact of a specific hyperparameter combination, we conducted extensive ablations across a wide grid of training configurations for both the ResNet-50 and ViT-B/16 architectures. These configurations varied significantly across several axes: training duration (epochs), learning rate schedulers, optimizer choice (AdamW vs. SGD), weight initialization (pretrained vs. random initialization), and feature representation updating (full-network fine-tuning vs. freezing base layers).

The exact specifications for the diverse recipes we evaluated are detailed below:

ResNet-50 Configurations:

- **20-Epoch Fine-tuning:** Initialized with ImageNet-1k pretrained weights. We unfroze all layers (freeze_layers=False) to allow full feature updating. Trained using the AdamW optimizer with a peak learning rate of 8×10^{-6} and a WarmupCosine scheduler.
- **40-Epoch Fine-tuning:** Initialized with ImageNet-1k pretrained weights, with all layers unfrozen (freeze_layers=False). Trained using the AdamW optimizer with a learning rate of 2×10^{-6} and a ReduceLROnPlateau scheduler to adaptively manage the learning rate during convergence.
- **80-Epoch Fine-tuning:** Initialized with ImageNet-1k pretrained weights, with all layers unfrozen (freeze_layers=False). Trained using the AdamW optimizer with a learning rate of 2×10^{-6} and a CosineAnnealingWarmRestarts scheduler to prevent getting stuck in local minima over the longer training duration.
- **120-Epoch Training from Scratch:** To evaluate performance without the bias of prior representation learning, this model was initialized randomly (no pretrained weights) with all layers trainable (freeze_layers=False). It was trained using the standard SGD optimizer with a significantly higher learning rate of 5×10^{-4} and a WarmupCosine scheduler.

ViT-B/16 Configurations:

- **30-Epoch Partial Fine-tuning:** Initialized with ImageNet-1k pretrained weights. Unlike the other configurations, we explicitly froze the base layers (`freeze_layers=True`) to evaluate `Mycroft`'s efficacy when only the classification head and final representations are updated. Trained using the AdamW optimizer with a learning rate of 3×10^{-4} and a `WarmupCosine` scheduler.
- **40-Epoch Full Fine-tuning:** Initialized with ImageNet-1k pretrained weights, with all layers unfrozen (`freeze_layers=False`). Trained using the AdamW optimizer with a learning rate of 3×10^{-4} and a `ReduceLROnPlateau` scheduler.
- **80-Epoch Full Fine-tuning:** Initialized with ImageNet-1k pretrained weights, with all layers unfrozen (`freeze_layers=False`). Trained using the AdamW optimizer with a lower learning rate of 3×10^{-5} and a `WarmupCosine` scheduler to accommodate the extended training horizon.

Standardized Configuration for Main Results Every individual configuration detailed above was executed and averaged over 5 distinct random seeds to ensure strict statistical significance and reproducibility. After thoroughly evaluating these distinct recipes (the results of which are presented in the robustness ablations in Appendix G.3), we standardized on the **40-epoch full fine-tuning setting** for both the ResNet-50 and ViT-B/16 architectures. This 40-epoch setting is used consistently to generate all of the primary results discussed in the main body of the paper (Sections 4 and 4.4).

G Additional results & Ablation studies

Here, we evaluate which training phase of the `DO`'s model provides the most useful gradient information for data selection. For evaluation, we chose the Dogs & Wolves dataset as the `DO` and perform gradient matching for several checkpoints. We compare the percentage of samples in the retrieved subset which belong to the Dogs & Wolves - Natural data subset since those are the only useful samples in the `DO`'s dataset. We plot the results in in Figure 5 and observe that earlier checkpoints indeed provide more useful gradients.

G.1 Budget Ablation

This section provides the detailed class-wise performance for our budget ablation experiments. As discussed in §4.3.1, we varied the per-class retrieval budget $k \in \{8, 16, 32, 64\}$ to evaluate how each method performs under strict data constraints.

The complete results for the Dogs & Wolves and Spawrious datasets are presented in Table 5 and Table 13. Across both datasets and model architectures, the proposed selection methods consistently outperform `random-sampling`, even at a small budget of $k = 8$. As the budget increases, `Mycroft` steadily approaches the performance of the `full-information` baseline. Notably, `Mycroft` with a budget of $k = 32$ frequently achieves higher accuracy than `random-sampling` with a budget of $k = 64$. This demonstrates that the performance improvements come from selecting relevant samples, rather than simply adding more data.

G.2 Checkpoint Ablation

We conduct ablation study on image datasets to see how the selection of checkpoints affects Adaptive Selection (OMP)'s ability to select D^{useful}. We find that earlier checkpoints tend to provide more useful gradient information for the OMP algorithm. Refer to Figure 5 for the results.

s

G.3 Sensitivity to Training Duration and Recipes

In the main paper, we reported results using a fixed training schedule (40 epochs) for consistency. Here, we provide the complete results across all explored training recipes to demonstrate the robustness of `Mycroft`.

Table 5: **Budget Efficiency Analysis (DogsVsWolves).** Validation accuracy (%) across different budget sizes. `Mycroft` methods outperform `random-sampling`, with Adaptive Selection (OMP) at a budget of 32 reaching similar performance to `random-sampling` at a budget of 64.

Model	k	Class	Baseline	Random	Adaptive	Voting	Agnostic
ResNet	8	Dogs	59.33	57.33	60.00	**60.67**	**60.67**
		Wolves	63.78	60.22	**64.44**	61.78	58.67
	16	Dogs	59.33	56.89	62.00	**63.78**	58.67
		Wolves	63.78	59.78	**65.56**	62.89	62.89
	32	Dogs	59.33	59.56	67.78	**71.78**	67.56
		Wolves	63.78	56.89	66.00	**66.67**	64.89
	64	Dogs	59.33	60.89	70.89	**75.11**	74.44
		Wolves	63.78	59.33	**78.67**	76.22	56.44
ViT	8	Dogs	25.11	19.56	30.00	**36.22**	34.89
		Wolves	12.22	**36.67**	29.56	34.67	27.78
	16	Dogs	20.67	26.44	44.89	**48.00**	44.89
		Wolves	18.00	32.44	45.11	**46.00**	34.00
	32	Dogs	26.44	23.56	**54.22**	46.89	44.89
		Wolves	14.67	33.11	52.67	54.67	**55.56**
	64	Dogs	30.22	27.56	**70.22**	64.22	60.00
		Wolves	12.67	33.33	72.44	**76.22**	70.00

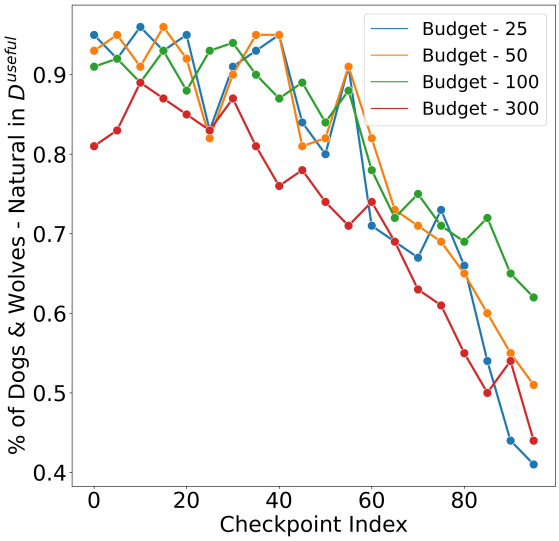


Figure 5: Effect of the checkpoint used for Adaptive Selection (OMP) on D^{useful}. Earlier checkpoints tend to provide more useful gradient information for the OMP algorithm.

For all fine-tuning experiments, we initialize both the **ResNet-50** and **ViT-B/16** backbones with weights pretrained on the **ImageNet-1k** dataset.

For **ResNet-50**, we evaluate four configurations: fine-tuning the ImageNet-initialized model for 20, 40, and 80 epochs, and training from scratch (random initialization) for 120 epochs. For **ViT-B/16**, we evaluate three fine-tuning configurations: 30, 40, and 80 epochs.

Tables 6 and 7 present the class-wise validation accuracy breakdown for the Dogs vs. Wolves dataset. Table 8 presents the validation accuracy for the Spawrious dataset across all Data Owners. We observe that `Mycroft`

consistently outperforms Random sampling and retrieves higher utility data regardless of the training duration or strategy.

Table 6: **Dogs & Wolves Sensitivity (ResNet-50).** Validation accuracy (%) comparison of fine-tuning (20, 40, 80 epochs) vs. training from scratch (120 epochs). `Mycroft` consistently achieves higher validation accuracy on the minority class (Wolves) compared to `random-sampling` across all durations.

	20 Epochs			40 Epochs			80 Epochs			Scratch (120 Ep)		
Method	**Dog**	**Wolf**	**Overall Val.**	**Dog**	**Wolf**	**Overall Val.**	**Dog**	**Wolf**	**Overall Val.**	**Dog**	**Wolf**	**Overall Val.**
Baseline	54.36	59.78	89.71	49.38	64.49	89.53	51.16	62.36	89.41	4.49	7.07	64.48
Random	51.24	60.93	89.41	52.53	59.47	89.25	50.84	59.24	89.21	9.20	9.73	69.26
Adaptive	64.22	64.84	91.38	59.69	67.51	91.13	59.78	65.20	91.06	21.96	20.27	71.75
Full Info	76.36	73.25	93.56	74.49	73.11	93.13	76.45	72.84	93.60	48.00	57.24	78.72

Table 7: **Dogs & Wolves Sensitivity (ViT-B/16).** Validation accuracy (%) comparison across 30, 40, and 80 fine-tuning epochs. `Mycroft` shows significant gains in validation accuracy over `random-sampling`, particularly in the 40 and 80 epoch settings where the model has more time to learn from the retrieved data.

	30 Epochs			40 Epochs			80 Epochs		
Method	**Dog**	**Wolf**	**Overall Val.**	**Dog**	**Wolf**	**Overall Val.**	**Dog**	**Wolf**	**Overall Val.**
Baseline	75.82	93.29	96.70	20.53	14.09	81.21	39.16	45.07	87.35
Random	76.76	93.20	96.78	30.67	26.40	83.99	48.49	43.96	88.66
Adaptive	79.02	93.38	97.02	57.56	49.51	88.54	69.47	75.96	93.98
Full Info	81.07	93.55	97.23	75.24	75.78	92.72	81.15	87.07	96.29

Table 8: **Spawrious Dataset Sensitivity Analysis.** Validation Accuracy on the adversarial set. We compare ResNet-50 (20, 40, 80 ep, and Scratch) and ViT-B/16 (30, 40, 80 ep). `Mycroft` correctly ranks the utility of Data Owners (`Good` > `Easy` > `Medium` > `Bad`) across the majority of configurations. The **Full Info** rows indicate the upper bound performance using the entire dataset for that specific training recipe.

		ResNet-50 Validation Acc (%)				ViT-B/16 Validation Acc (%)		
Target DO	**Method**	**20 ep**	**40 ep**	**80 ep**	**Scratch**	**30 ep**	**40 ep**	**80 ep**
DO-Easy	Baseline	75.24	73.08	75.30	24.93	77.72	26.28	46.11
	Random	83.53	83.73	84.76	83.83	77.58	73.96	77.14
	Adaptive	86.72	86.77	88.06	84.98	78.58	72.54	88.91
	Full Info	99.25	99.16	99.28	99.17	96.20	97.56	98.74
DO-Good	Baseline	75.24	73.08	75.30	24.93	77.72	26.15	45.98
	Random	79.12	78.47	80.01	75.98	77.93	56.23	77.14
	Adaptive	86.86	86.41	88.04	84.10	78.54	71.30	88.56
	Full Info	98.87	98.85	98.95	98.42	96.63	97.56	98.74
DO-Medium	Baseline	75.24	73.08	75.30	24.93	77.72	26.05	45.57
	Random	79.16	78.39	79.83	75.82	79.26	57.60	77.52
	Adaptive	86.57	86.95	87.88	84.21	78.65	65.52	88.66
	Full Info	98.87	98.77	98.95	98.42	96.63	96.77	98.74
DO-Bad	Baseline	75.24	73.08	75.30	24.93	77.72	26.77	46.28
	Random	74.50	72.85	74.43	67.76	28.02	28.02	44.27
	Adaptive	75.98	74.40	76.07	69.49	30.01	30.01	46.12
	Full Info	75.78	74.10	75.87	68.64	78.20	24.76	48.37

G.4 Cross-Model Ablation

This section provides the comprehensive performance breakdowns for the cross-architecture transferability experiments discussed in Section §4.3.3. In these experiments, we simulate a practical market constraint where the Model Trainer (`MT`) and Data Owner (`DO`) utilize mismatched model architectures.

Specifically, the `MT` optimizes a ViT-B/16 architecture, while the `DO` only has access to a ResNet-50 proxy model to compute the gradient matching for Adaptive Selection (OMP). Table 9 demonstrates that despite this architectural mismatch, the gradient information transfers exceptionally well. On the Spawrious dataset, acquiring data from `DO-Good` boosts the ViT-B/16's validation accuracy by over 52% compared to the baseline, confirming that `Mycroft` does not require strict architectural alignment.

Table 9: **Cross-Architecture Transferability.** Overall validation accuracy when the Model Trainer (`MT`) and Data Owner (`DO`) utilize mismatched model architectures. For a ResNet-50 `MT`, the `DO` uses a ViT-B/16 proxy to compute gradients, and vice versa. `Mycroft` robustly improves accuracy across all data sources despite the severe architectural mismatch.

`MT` Model	Source (`DO`)	Baseline (%)	Augmented (%)	Improvement
Dataset: Spawrious				
ResNet-50	DO-Good	71.70	85.38	+13.68%
	DO-Easy	71.70	**85.67**	+13.97%
	DO-Medium	71.70	85.30	+13.60%
	DO-Bad	71.70	73.81	+2.11%
ViT-B/16	DO-Good	26.22	**78.28**	+52.06%
	DO-Easy	26.22	75.80	+49.58%
	DO-Medium	26.22	71.06	+44.84%
	DO-Bad	26.28	27.60	+1.32%
Dataset: Dogs & Wolves				
ResNet-50	DO-1 (Primary)	90.22	91.27	+1.05%
ViT-B/16	DO-1 (Primary)	81.36	**90.51**	+9.15%

G.5 Detailed Rank Correlation Analysis (Kendall's τ)

As discussed in the case study (§4.4.1), Kendall's τ is a statistical measure used to calculate the ordinal association between two measured quantities. In our case, we use it to compare the Data Owner (DO) ranking predicted by our `DataSelect` subroutines against a *ground-truth* ranking established by the `full-information` upper bound.

The formula for Kendall's τ is:

$$\tau = \frac{C - D}{\frac{1}{2}n(n-1)} \tag{8}$$

where C is the number of concordant (agreeing) pairs, D is the number of discordant (disagreeing) pairs, and n is the total number of items being ranked. A τ of 1.0 indicates a perfect ranking match, while 0.0 indicates no correlation.

G.5.1 Dogs & Wolves Dataset Ranking Calculations

For the Dogs & Wolves dataset (Table 10), we rank $n = 6$ data owners, resulting in $\frac{1}{2}(6)(5) = 15$ total possible pairs.

ViT-B/16 Architecture:

- **Ground Truth (`full-information`) Order:** DO-4 > DO-1 > DO-6 > DO-2 > DO-3 > DO-5

- **Adaptive Selection (OMP) Order:** DO-1 > DO-4 > DO-2 > DO-6 > DO-3 > DO-5

Comparing every possible pair between the ground truth and the Adaptive Selection (OMP) predictions, we find that the rankings agree on the relative order of almost all data owners. There are only two discordant pairs where the predicted order flips the ground truth: ranking DO-1 slightly above DO-4, and similarly DO-2 slightly above DO-6. This yields 13 out of $15 = \binom{6}{2}$ concordant pairs ($C = 13$) and 2 discordant pairs ($D = 2$).

$$\tau_{ViT} = \frac{13 - 2}{15} = \frac{11}{15} \approx 0.733 \tag{9}$$

ResNet-50 Architecture:

- **Ground Truth (`full-information`) Order:** DO-4 > DO-1 > DO-6 > DO-3 > DO-5 > DO-2
- **Adaptive Selection (OMP) Order:** DO-4 > DO-1 > DO-6 > DO-2 > DO-3 > DO-5

Similar to the ViT-B/16 architecture, Adaptive Selection (OMP) achieves near-perfect retrieval of the ground truth ranking. It perfectly identifies the top three DOs in exact order (DO-4 > DO-1 > DO-6). The only discordant pairs occur at the bottom of the rankings, where Adaptive Selection (OMP) places DO-2 slightly higher than DO-3 and DO-5. This again yields 13 concordant pairs ($C = 13$) and 2 discordant pairs ($D = 2$).

$$\tau_{ResNet} = \frac{13 - 2}{15} = \frac{11}{15} \approx 0.733 \tag{10}$$

G.5.2 Spawrious Dataset Ranking Calculations

For the Spawrious dataset, we rank $n = 4$ data owners, resulting in $\frac{1}{2}(4)(3) = 6$ total possible pairs.

ViT-B/16 Architecture:

- **Ground Truth (`full-information`) Order:** DO-Easy > DO-Good > DO-Medium > DO-Bad
- **Adaptive Selection (OMP) Order:** DO-Easy > DO-Good > DO-Medium > DO-Bad

Adaptive Selection perfectly recovers the ground truth ranking, yielding 6 concordant pairs ($C = 6$) and 0 discordant pairs ($D = 0$).

$$\tau_{ViT} = \frac{6 - 0}{6} = 1.0 \tag{11}$$

ResNet-50 Architecture:

- **Ground Truth (`full-information`) Order:** DO-Easy > DO-Good > DO-Medium > DO-Bad
- **Agnostic Selection (RPC) Order:** DO-Easy > DO-Good > DO-Medium > DO-Bad

Agnostic Selection also perfectly recovers the ground truth ranking, yielding $C = 6$ and $D = 0$.

$$\tau_{ResNet} = \frac{6 - 0}{6} = 1.0 \tag{12}$$

Why Random Sampling's Perfect Ranking is Misleading: A notable edge case arises on the Spawrious dataset with the ResNet-50 architecture, where `random-sampling` coincidentally achieves a perfect rank correlation ($\tau = 1.0$), while Adaptive Selection (OMP) achieves only $\tau = 0.333$. However, relying strictly on ranking order masks the actual performance gained. `random-sampling` fails to meaningfully separate the sources, scoring `DO-Good` and `DO-Medium` within a negligible 0.08% margin of each other (78.47% vs. 78.39%). In contrast, while Adaptive Selection slightly scrambles the exact order of the top three providers, it extracts

vastly superior downstream utility from all of them, boosting accuracy to ∼86% across the board. This indicates that while gradient-matching may struggle to perfectly sort tightly clustered, high-quality sources, it reliably maximizes actual model performance over random selection. Furthermore, the `MT` can easily resolve this ranking ambiguity by switching to Agnostic Selection (RPC), which correctly recovers the perfect $\tau = 1.0$ ranking for ResNet-50 while maintaining strong absolute accuracy margins.

A complete summary of Kendall's τ rank correlations across all selection methods and datasets is provided in Table 11.

Table 10: **Utility Ranking (DogsVsWolves).** Accuracy across 6 different DOs. `Mycroft` successfully identifies DO-1 and DO-4 as the best sources, while `random-sampling` often gives similar scores to lower-quality datasets. Bold text indicates the highest accuracy per row.

Model	Source	Baseline	Random	Adaptive	Voting	Agnostic	Full
ResNet	DO-1	89.53	89.25	91.13	**91.70**	91.38	93.13
	DO-2	90.22	89.53	90.56	90.64	**91.53**	88.80
	DO-3	90.22	89.49	90.02	89.64	**91.00**	89.31
	DO-4	90.22	90.16	**91.80**	89.62	90.80	95.44
	DO-5	90.22	89.53	89.89	89.80	**90.76**	88.93
	DO-6	90.22	89.64	90.69	**91.11**	90.93	90.98
ViT	DO-1	81.21	83.99	88.54	88.44	**88.71**	92.72
	DO-2	81.64	89.53	86.04	86.80	**89.27**	85.40
	DO-3	80.73	81.69	82.38	81.67	**85.89**	82.22
	DO-4	80.89	85.33	**86.96**	81.56	85.18	95.51
	DO-5	81.09	82.22	82.04	81.76	**84.62**	81.93
	DO-6	81.96	82.62	84.89	85.56	**87.22**	87.13

Table 11: **Detailed Rank Correlation (Kendall's τ).** Comparison of the utility rankings predicted by each selection method against the ground-truth `full-information` ranking across both datasets. A τ of 1.0 indicates a perfect rank match. `Mycroft` methods (Adaptive and Agnostic) consistently yield the highest rank correlations for their respective optimal architectures.

Selection Method	Dogs & Wolves (τ)		Spawrious (τ)	
	ResNet-50	ViT-B/16	ResNet-50	ViT-B/16
Random	0.133	0.467	1.000	0.667
Adaptive (OMP)	**0.733**	**0.733**	0.333	**1.000**
Voting	-0.067	0.067	0.667	0.667
Agnostic (RPC)	-0.200	0.200	**1.000**	0.333

G.5.3 Spawrious Dataset Ranking Calculations

For the Spawrious dataset (Table 12), we rank $n = 4$ data owners, resulting in $\frac{1}{2}(4)(3) = 6$ total possible pairs.

ViT-B/16 Architecture:

- **Ground Truth (`full-information`) Order:** Easy (98.68%) > Good (97.56%) > Medium (96.77%) > Bad (24.76%)
- **Adaptive Selection (OMP) Order:** Easy (72.54%) > Good (71.30%) > Medium (65.52%) > Bad (30.01%)

Comparing the pairs between the two rankings yields 6 concordant pairs ($C = 6$) and 0 discordant pairs ($D = 0$). The calculation is:

$$\tau_{ViT} = \frac{6 - 0}{6} = 1.0 \tag{13}$$

ResNet-50 Architecture:

- **Ground Truth (`full-information`) Order:** Easy (99.16%) > Good (98.85%) > Medium (98.77%) > Bad (74.10%)
- **Agnostic Selection (RPC) Order:** Easy (87.28%) > Good (86.11%) > Medium (85.39%) > Bad (77.64%)

While Adaptive Selection (OMP) slightly mixes the tightly-clustered top three sources ($\tau \approx 0.333$), utilizing Agnostic Selection (RPC) cleanly resolves the ambiguity. Comparing the pairs between the `full-information` ground truth and the RPC predictions yields 6 concordant pairs ($C = 6$) and 0 discordant pairs ($D = 0$). The calculation is:

$$\tau_{ResNet} = \frac{6 - 0}{6} = 1.0 \tag{14}$$

Table 12: **Utility Ranking (Spawrious).** Accuracy on 4 DOs of varying quality. `Mycroft` correctly ranks the data owners by how useful they actually are ($Good > Easy > Medium > Bad$), whereas `random-sampling` fails to separate them cleanly.

Model	Source	Baseline	Random	Adaptive	Voting	Agnostic	Full
ResNet	Good	73.08	78.47	**86.41**	85.82	86.11	98.85
	Easy	73.08	83.73	86.77	86.35	**87.28**	99.16
	Medium	73.08	78.39	86.95	**85.99**	85.39	98.77
	Bad	73.08	72.85	74.40	75.04	**77.64**	74.10
ViT	Good	26.15	56.23	71.30	68.20	**71.96**	97.56
	Easy	26.28	**73.96**	72.54	64.19	72.59	98.68
	Medium	26.05	57.60	65.52	62.82	**72.96**	96.77
	Bad	26.77	28.02	30.01	28.42	**37.87**	24.76

G.6 Robustness to Data Corruption Ablation

This section provides the complete quantitative results for the data corruption case studies discussed in §4.4.2. We evaluate the robustness of the data selection subroutines when the Data Owner provides severely degraded images (Table 14) or highly noisy labels (Table 15).

You may include other additional sections here.

Table 13: **Budget Efficiency Analysis (Spawrious).** Validation accuracy (%) across different budgets. Adaptive Selection (OMP) and Voting based Selection (Unicom) scale better than random sampling at low budgets, successfully grabbing the specific breed-background images needed to break the model's bias.

Model	k	Class	Baseline	Random	Adaptive	Voting	Agnostic
ResNet	8	Bulldog	81.79	81.41	83.84	82.77	**85.76**
		Corgi	98.58	98.55	99.05	**99.12**	99.02
		Dachshund	39.61	44.85	53.09	**56.47**	51.10
		Labrador	66.82	62.12	74.65	77.46	**77.68**
	16	Bulldog	81.79	83.55	**90.81**	88.83	87.82
		Corgi	98.58	98.67	**99.40**	99.34	99.34
		Dachshund	39.61	51.67	**59.79**	58.65	**59.79**
		Labrador	66.82	74.43	79.58	**81.82**	77.90
	32	Bulldog	81.79	88.10	93.21	91.64	**93.59**
		Corgi	98.58	98.45	**99.53**	99.40	99.46
		Dachshund	39.61	56.53	**69.00**	67.01	68.43
		Labrador	66.82	75.25	**86.11**	84.88	82.95
	64	Bulldog	81.79	87.78	**96.12**	93.06	95.08
		Corgi	98.58	98.99	99.68	98.67	**99.75**
		Dachshund	39.61	61.74	**79.42**	71.59	75.19
		Labrador	66.82	78.69	**90.78**	89.17	88.64
ViT	8	Bulldog	23.55	52.37	49.15	54.99	**60.16**
		Corgi	67.46	80.59	84.60	**85.61**	82.17
		Dachshund	8.71	5.81	9.88	29.32	**30.43**
		Labrador	6.38	53.28	48.17	**61.58**	36.49
	16	Bulldog	22.19	61.27	48.26	59.53	**68.97**
		Corgi	66.10	78.28	85.07	88.83	**92.17**
		Dachshund	0.95	7.13	15.09	**31.25**	29.01
		Labrador	15.66	58.24	67.20	**72.10**	70.71
	32	Bulldog	26.29	58.02	**82.39**	65.69	81.22
		Corgi	69.89	84.82	82.73	**90.06**	**90.06**
		Dachshund	6.66	32.39	33.68	37.69	**38.51**
		Labrador	4.92	46.15	**85.07**	78.91	78.06
	64	Bulldog	26.22	71.88	**94.95**	65.06	78.54
		Corgi	66.10	91.70	**97.79**	83.33	97.25
		Dachshund	0.95	31.19	47.82	40.81	**51.48**
		Labrador	15.66	70.64	**92.23**	72.25	84.06

Table 14: **Robustness to Feature Corruption.** Overall validation accuracy when the data owner's images are heavily corrupted. `Mycroft` still manages to retrieve useful data.

Data	Model	Baseline	Random	Adaptive	Voting	Agnostic	Full
DogsVsWolves	ResNet	90.22	89.84	**90.56**	90.44	90.40	91.67
	ViT	81.78	83.56	83.84	**85.42**	83.71	86.24
Spawrious	ResNet	71.70	76.10	77.08	**80.46**	76.62	95.78
	ViT	26.75	25.19	32.84	**37.29**	30.67	49.57

Table 15: **Robustness to Label Corruption.** Validation accuracy when the data owner has 70% noisy labels. Adaptive Selection(OMP) and Agnostic Selection(RPC) handle this well because the conflicting gradients from the bad labels get filtered out naturally.

Data	**Model**	**Baseline**	**Random**	**Adaptive**	**Voting**	**Agnostic**	**Full**
DogsVsWolves	ResNet	90.22	89.71	91.33	**91.78**	90.64	91.93
	ViT	81.44	82.11	**88.62**	87.38	87.47	87.80
Spawrious	ResNet	71.70	72.19	83.89	**85.31**	**85.31**	94.56
	ViT	29.72	39.34	**72.31**	69.33	67.18	68.75

Table 16: **Kernel and Pivoting Ablation.** Validation accuracy (%) for ResNet-50 and ViT-B/16 across different kernels and pivoting strategies. The results demonstrate that local greedy sampling combined with label-independent kernels (like Laplace or NNGP) performs competitively.

	local, greedy Resnet			local, refractory Resnet			global, greedy Resnet			local, greedy ViT		
Method	Dog	Wolf	Overall Val.	Dog	Wolf	Overall Val.	Dog	Wolf	Overall Val.	Dog	Wolf	Overall Val.
Baseline	59.56	63.56	90.2	59.56	63.56	90.2	59.56	63.56	90.2	26.44	14.67	81.96
Random	51.24	60.93	89.41	52.53	59.47	89.25	50.84	59.24	89.21	9.20	9.73	69.26
NNGP	66.67	65.11	91.56	**64.22**	58.44	**90.82**	60.67	**63.78**	90.16	54.21	44.22	87.94
Laplace*	**67.79**	64.67	91.56	63.56	60.22	90.24	63.56	59.56	**90.49**	43.77	**54.45**	88.27
NTK	67.34	**67.56**	**91.62**	62.89	60.00	90.6	**70.00**	58.56	90.29	**66.22**	52.22	**89.60**
Unicom	58.9	59.11	89.96	61.11	56.44	90.11	63.33	59.33	89.76	36.66	36.67	85.38
Laplace-image	62.44	61.11	90.76	62.44	**63.11**	90.47	65.11	58.89	89.91	35.11	45.11	86.11
Full Info	76.36	73.25	93.56	74.49	73.11	93.13	76.45	72.84	93.60	48.00	57.24	78.72

Table 17: **Beta(β) Parameter Ablation.** Validation accuracy (%) across different β values using the Laplace-image kernel with local diversity on ResNet-50. The model exhibits stable performance across various sampling strategies, transitioning from uniform to greedy selection.

Method	**Dog**	**Wolf**	**Overall Val.**
Baseline	59.56	63.56	90.2
Random	51.24	60.93	89.41
$\beta = 1$ (refractory)	**63.78**	55.33	90.31
$\beta = \infty$ (greedy)	60.22	60.89	**90.42**
$\beta = 0$ (uniform)	62.22	58.44	90.42
$\beta = 10$	61.78	**62.89**	90.40
$\beta = 100$	60.67	60.00	90.42
$\beta = 500$	61.33	60.89	90.42
Full Info	76.36	73.25	93.56