Automatic Combination of Sample Selection Strategies for Few-Shot Learning

Branislav Pecher[♠†], **Ivan Srba**[†], **Maria Bielikova**[†], **Joaquin Vanschoren**[‡]
[♠] Faculty of Information Technology, Brno University of Technology, Brno, Czechia
[†] Kempelen Institute of Intelligent Technologies, Bratislava, Slovakia
[‡] Eindhoven University of Technology, Eindhoven, Netherlands
{branislav.pecher, ivan.srba, maria.bielikova}@kinit.sk, j.vanschoren@tue.nl

Abstract

In few-shot learning, the selection of samples has a significant impact on the performance of the model. While effective sample selection strategies are well-established in supervised settings, research on large language models largely overlooks them, favouring strategies specifically tailored to individual in-context learning settings. In this paper, we propose a new method for Automatic Combination of SamplE Selection Strategies (ACSESS) to leverage the strengths and complementarity of various well-established selection objectives. We investigate and compare the impact of 23 sample selection strategies on the performance of 5 in-context learning models and 3 few-shot learning approaches (meta-learning, few-shot fine-tuning) over 6 text and 8 image datasets. The experimental results show that the combination of strategies through the ACSESS method consistently outperforms all individual selection strategies and performs on par or exceeds the in-context learning specific baselines. Lastly, we demonstrate that sample selection remains effective even on smaller datasets, yielding the greatest benefits when only a few shots are selected, while its advantage diminishes as the number of shots increases.

1 Introduction

Many domains are characterised by a labelled data scarcity due to data collection/annotation costs or privacy considerations, making the training of typical deep learning models unfeasible. Few-shot learning addresses the challenge of adapting models to new tasks when only a handful of labelled samples are available (Song et al., 2023). Two main approaches exist: 1) *in-context learning*, where a pretrained large language model is conditioned on few examples without any parameter updates (Dong et al., 2022; Liu et al., 2023); and 2) *gradient-based few-shot learning*, which includes meta-learning and fine-tuning, where the knowledge from relevant datasets is transferred to a new problem using gradient-based training (Song et al., 2023; Chen et al., 2019; Vanschoren, 2018; Hospedales et al., 2021). More details regarding their differences are included in Appendix J.

The choice of which sample to use is critical in both approaches, as performance can vary drastically depending on sample quality. Existing studies in different settings, such as in LLM alignment (Zhou et al., 2024), have shown that **curating a smaller subset of high-quality samples can lead to better performance than training on the full set of available samples**. This holds for few-shot learning as well, especially for noisy or unbalanced datasets or for in-context learning, which is known to be sensitive to sample choice (Pecher et al., 2024; Agarwal et al., 2021; Köksal et al., 2022).

Many works have already proposed diverse sample selection strategies to identify an optimal set of samples. Such a *sample selection strategy* follows its specific *objective* by considering one or few *properties* of available samples, e.g., to maximise similarity, diversity, informativeness or quality of selected samples (Li and Qiu, 2023; Zhang et al., 2022; Chang and Jia, 2023). With the popularisation of large language models (LLMs), many new selection strategies are proposed for in-context learning. However, they are often tailored only for specific settings, limiting their applicability. At the same time, the well-established and long-standing selection strategies that were shown to perform well in supervised settings, such as active learning, are mostly ignored as part of LLM sample selection (Pecher et al., 2024; Albalak et al., 2024).

In this work, we **demonstrate that well-established strategies can match or exceed the performance of in-context learning specific strategies**. To achieve this, we **automatically and optimally combine the sample selection objectives used by well-established sample selection strategies to identify samples with complemen-**

Findings of the Association for Computational Linguistics: ACL 2026, pages 40385–40416
July 2-7, 2026 ©2026 Association for Computational Linguistics

tary properties. In order to leverage the strengths of different sample selection strategies, we propose **ACSESS**, an effective method for **A**utomatic **C**ombination of **S**ampl**E** **S**election **S**trategies. First, a subset of relevant strategies that can improve the overall success of few-shot learning is identified. Afterwards, the selection objectives from the identified strategies are combined together, using weighting based on their expected contribution, in order to identify the most informative and high-quality samples that can provide the most benefit. Our contributions and findings are as follows[1]:

- We provide the first large-scale study of 23 sample selection strategies across 5 in-context learning models and 3 few-shot learning approaches, including meta-learning and few-shot fine-tuning, across 6 text and 8 image datasets. We show that sample selection can yield consistent gains (up to 5 and 3 percentage points for in-context learning and gradient few-shot learning, respectively), though the effects depend heavily on dataset and approach.

- We propose **ACSESS**, a method that automatically combines and weights sample selection objectives to leverage their strengths and identify samples with complementary properties (i.e., informativeness, representativeness and learnability). Our results show that the combination of well-established strategies identified by the ACSESS method consistently leads to performance on par or better than the existing in-context learning specific baselines.

- Through ablation studies, we find following key insights: 1) sample selection has higher impact when the number of shots is low or when using noisy datasets; 2) at higher number of shots (30-40 on average), the impact of sample selection is negligible as all strategies regress to random selection; 3) after a certain point, the boost in performance from using more shots becomes negligible (30 shots for in-context learning; 50-shot for gradient few-shot learning); 4) sample selection is beneficial even for small dataset sizes (achieving similar performance when selecting from only 25% for in-context learning or 10% of the dataset for gradient few-shot learning).

[1]To support replicability and extension of our results, we openly publish the source code of our experiments at https://github.com/kinit-sk/ACSESS

2 Related Work

A large body of work is dedicated to selecting high-quality samples for in-context learning, where the overall performance was found to be sensitive to this choice, leading to large variability in results (Pecher et al., 2024; Köksal et al., 2022; Zhang et al., 2022). In the supervised setting, the selection is mostly focused on distilling datasets to a lower number of samples (Yu et al., 2023), selecting a core set representative of the full set (Guo et al., 2022), or reducing annotation costs using active learning (Ren et al., 2021). However, the impact of these strategies remains underexplored for other few-shot learning approaches.

Within the majority of the sample selection strategies, the sample selection objective takes into consideration **a single property** of available samples. To this end, various heuristics and unsupervised metrics serve as an estimate of their potential to increase the model's performance. For in-context learning, the most popular approach is selecting samples based on the **similarity** to the test sample, differing only in employed similarity measure and representation (Liu et al., 2022; An et al., 2023; Zemlyanskiy et al., 2022; Gupta et al., 2022; Pasupat et al., 2021; Wang et al., 2022; Nashid et al., 2023; Agrawal et al., 2023; Gao et al., 2021). Besides similarity, the samples are often selected based on their **informativeness** or the **uncertainty**, often using active learning strategies (Köksal et al., 2022; Schröder et al., 2022; Margatina et al., 2021; Park et al., 2022; Mavromatis et al., 2023), bias of the samples (Ma et al., 2023) or adversarial training (Agarwal et al., 2021). Other methods define a notion of **quality** for each sample, either by prompting a LLM to rate the samples (Shin et al., 2021) or observing how the inclusion or removal of the sample affects the performance (Maronikolakis et al., 2023; Nguyen and Wong, 2023). When considering **representativeness**, the samples are selected based on how well they are representative of the full dataset (Guo et al., 2022; Killamsetty et al., 2021; Mirzasoleiman et al., 2020; Paul et al., 2021; Coleman et al., 2020). **Learnability** metric either determines how easy it is to learn the specific sample (Swayamdipta et al., 2020; Zhang and Plank, 2021), or how often the sample is forgotten after being learned (Toneva et al., 2018).

Finally, some approaches **balance multiple sample properties** at the same time. Often, the similarity of samples is combined with their diversity (Wu

et al., 2023; Qin et al., 2023) or the informativeness of the samples is balanced with their representativeness (Su et al., 2022; Levy et al., 2023; Ye et al., 2023; Liu and Wang, 2023). In specific cases, a two step sample search is done, such as finding set of informative samples and then using diversity guided search to improve this set (Li and Qiu, 2023), or using different uncertainty measures to select the top samples (Diao et al., 2024).

Newer strategies **leverage optimisation** to select a single representative set of samples. One possibility is to train a sample retriever, such as a separate scoring model (Rubin et al., 2022; Luo et al., 2023; Li et al., 2023; Wang et al., 2023; Aimen et al., 2023) or define a surrogate scorer for modelling the sample's goodness and use multi-armed bandit to select the best exemplars (Purohit et al., 2024, 2025). Another option is to leverage reinforcement learning (Zhang et al., 2022; Shum et al., 2023; Scarlatos and Lan, 2023). Finally, the Datamodels approach trains a linear model to predict the performance gain of a set of samples to select the subset that would lead to the highest possible performance increase (Ilyas et al., 2022; Chang and Jia, 2023; Jundi and Lapesa, 2022; Vilar et al., 2023).

In this work, we investigate whether the combination of objectives from well-established single-property strategies can rival the selection from the strategies tailored specifically to in-context learning. Similar to the close works of Li and Qiu (2023); Purohit et al. (2024, 2025), we select a single set of samples, but do not focus solely on in-context learning. In essence, we complement work by Agarwal et al. (2021) by exploring and effectively combining sample properties important for few-shot learning performance. Finally, our proposed method is inspired by the Datamodels approach (Ilyas et al., 2022), but performs the selection at the level of strategies instead of samples.

3 ACSESS: Automatic Combination of Selection Strategies

Despite the large number of existing sample selection strategies, most remain constrained to a single objective, typically considering only one sample property. This narrow focus often leads to suboptimal selection, whereas greater performance gains can be achieved by appropriately combining multiple objectives that capture complementary sample properties. For example, the most informative sample that is hard to learn or is often forgotten may not contribute as much for few-shot methods that utilise gradient-based training. On the other hand, for in-context learning, the samples on the decision boundary, which are often hard to learn, may provide more benefit.

Instead of focusing on a single property, we propose to select examples that are characterised by a balance of complementary properties, even when they may not represent the best sample of any given individual property. To accomplish this, we propose **ACSESS**, a method for **A**utomatic **C**ombination of **S**ampl**E** **S**election **S**trategies taking their complementarity into account. The method is composed of three stages: 1) *definition of single-property strategies* that will be considered for the combination; 2) *identification of a subset of well-performing and relevant strategies* that can improve the performance of few-shot learning; and then 3) *combining the identified strategies* to identify the final set of the most beneficial samples.

To combine selection objectives, the ACSESS method assigns a *selection score* for each sample, representing its expected contribution towards a higher performance. It is calculated as a weighted sum of scores provided by individual objectives, formally $score(x) = \sum_{s \in S} w_s * objective_s(x)$, where **S** represents the set of considered strategies, w_s is the weight assigned to the strategy s, x is the input sample to be scored, and $objective_s(x)$is the score assigned to the sample x by strategy s normalised to a $< 0, 1 >$ interval. After assigning a score to each sample x, N samples with the highest score are selected. The steps of the method are summarised in Algorithm 1, with further details, including discussion of the computation complexity and its optimisation, in Appendix A.

Single-Property Strategies We focus on well-established strategies that select based on following properties: 1) *informativeness*, or how informative the samples are for the model; 2) *representativeness*, or how well the subset of samples represents the full dataset; and 3) *learnability*, or how easy it is to obtain or retain the information contained in the samples. In this work, we specifically focus on strategies that work by assigning a score to each sample, sorting based on this score and selecting the samples with the best score. For *informativeness*, this includes selecting samples based in their *Similarity* and *Diversity*, active learning strategies, such as *Entropy*, *Margin*, *Least Confidence* and *Loss* (Park et al., 2022),

and core-set selection including *Contrastive Active Learning (CAL)* (Margatina et al., 2021), *DeepFool* (Ducoffe and Precioso, 2018), *GraNd* (Paul et al., 2021) and *Graph-Cut* (Iyer and Bilmes, 2013; Iyer et al., 2021). For representativeness, we consider *Herding* (Welling, 2009; Chen et al., 2010), *KCenter* (Sener and Savarese, 2018; Agarwal et al., 2020), *CRAIG* (Mirzasoleiman et al., 2020) and *Glister* (Killamsetty et al., 2021). Finally, for learnability, we consider *Forgetting* (Toneva et al., 2018) and *Cartography* (Swayamdipta et al., 2020; Zhang and Plank, 2021), selecting all of each, ambiguous, hard and a combination of easy and ambiguous samples (due to no obvious consensus on which are the best). More details regarding the strategies and their objectives are included in Appendix B.

Identifying the Subset of Relevant Strategies In the second phase, the goal is to identify relevant strategies that, when combined, can select high-quality samples based on their complementary properties. Even though the optimal combination could be achieved by searching through all the combinations of strategies, such a search is infeasible as it requires extensive computation. Instead, we draw inspiration from methods commonly used for feature selection in traditional machine learning – namely the *forward* and *backward* selection – and for the first time apply them to the problem of sample selection. In addition, we adapt *Datamodels*, introduced by Ilyas et al. (2022) for individual sample selection, to work on the level of strategies instead. Such an approach allows us to explore a more diverse set of possible strategy combinations as compared to forward/backward selection. All three of the methods are run independently of each other, each producing a separate set of strategies, i.e., $\mathbf{S_F}$, $\mathbf{S_B}$ and $\mathbf{S_D}$ respectively for forward, backward and datamodels selection. The final set of strategies used by our method is constructed as an intersection between these sets, i.e., $\mathbf{S_{final}} = \mathbf{S_F} \cap \mathbf{S_B} \cap \mathbf{S_D}$. This allows the method to identify the most impactful strategies (i.e., strategies identified by all of the methods), minimise the potential drawbacks of the individual methods (i.e., exploring also diverse combinations), and guarantee efficiency of the method, by identifying the lowest number of strategies possible.

Algorithm 1 Automatic Combination of Sample Selection Objectives

Require: $x \in X$ - a set of available labelled samples
Require: N - number of samples to choose
1: Define $\mathbf{S}$, a set of considered sample selection strategies
2: Apply forward selection, $\mathbf{S_F} = forward(N, \mathbf{S})$ (Alg. 2)
3: Apply backward selection, $\mathbf{S_B} = backward(N, \mathbf{S})$ (Alg. 3)
4: Calculate performance of the few-shot model on the baseline setting, $perf$
5: Apply datamodels selection, $\mathbf{S_D}, weights_D = datamodels(N, \mathbf{S}, perf)$ (Alg. 4)
6: Select final set of strategies $\mathbf{S_{final}} = \mathbf{S_F} \cap \mathbf{S_B} \cap \mathbf{S_D}$
7: $w = weights_D$ if weighted combination else $1/|\mathbf{S_{final}}|$
8: Calculate final set of scores for each sample $score(x) = \sum_{s \in S} w_s * objective_s(x)$
9: Select top N samples based on $score(x)$

Forward Selection iteratively adds strategies that result in the highest performance increase (similarly to forward feature selection). Starting with an empty subset, the method alternates between two steps. First, for each strategy not included in the subset of selected strategies, we determine the increase in performance the strategy would yield if added to the subset. We determine the increase by training the model with N samples with the highest selection score determined by so far selected strategies (with weights uniformly distributed for all strategies, $w_s = 1/|\mathbf{S_{temp}}|$). Second, the strategy that provides the highest positive increase in performance is added to the subset and the first step is repeated. If the addition of any strategy does not lead to a performance increase, the process ends, resulting in the subset of relevant selection strategies ($\mathbf{S_F}$) (see Algorithm 2 in Appendix A for more details). In essence, the forward selection iteratively adds the best-performing strategies to the set while the performance still increases.

Backward Selection iteratively removes strategies until the performance no longer increases (similarly to backward feature selection). The method works the same way as the forward selection, but starts with all selection strategies and removes them from the subset ($\mathbf{S_B}$) based on the performance increase such removal leads to (see Algorithm 3 in Appendix A for more details). In essence, the backward selection iteratively removes strategies that lead to lower performance until the performance no longer increases.

Datamodels Selection draws inspiration from Datamodels (Ilyas et al., 2022; Chang and Jia, 2023). First, a set of M (in our work $M = 150$) random combinations of strategies is created and evaluated (using uniform combination). The random search is constrained to include each strategy at least 5 times to ensure sufficient coverage of each strategy. Afterwards, these combinations and

their performance are used to create a simple regression dataset. The difference in performance to the baseline selection (the classic few-shot learning selection) is used as the target. As features, we use a multi-label presence vector of strategies, i.e., vector containing only zeros and ones, where 1 is assigned to elements corresponding to strategy that was used in the combination, and 0 for all other ones. The dataset is then used to train a regularised linear regression, LASSO (Tibshirani, 1996), which zeros out the weights for as many strategies as possible. The selection strategies with positive weights in the model represent the subset of relevant strategies ($\mathbf{S_D}$) (see Algorithm 4 in Appendix A for more details).

Combining the Identified Strategies After identifying the relevant selection strategies ($\mathbf{S_{final}}$), a weighted combination of objectives is used to perform the selection. The final set consists of N samples with the highest selection score.

In this paper, we evaluate three separate weighting schemes. First, **uniform** combination, assigns the same weight to each of the identified strategies ($w_s = 1/|\mathbf{S_{final}}|$). This represents the simplified version of the ACSESS method that potentially leads to lower performance impact, but also lower computation costs and better transferability, as no further optimisation is required. This weighting scheme is also used for the previous step for the forward, backward and datamodels selection. Second, **weighted** combination, assigns a different weight to each strategy objective, prioritising specific sample properties beneficial for the specific few-shot approach. As the strategy weights for this weighted combination, we use the final trained weights of the *Datamodels* linear model. As this weighting scheme is dataset and model-specific, it often leads to the highest performance impact, at the cost of requiring more extensive computation. For this reason, in this paper, we evaluate the strategy on a more aggregated level in terms of datasets – see Appendix A for details. Finally, the **weighted combination with random selection** extends the *weighted* combination by introducing a random element to it, as the random selection was identified as a strong baseline (Park et al., 2022; Cegin et al., 2025). For this weighting scheme, each sample is assigned an additional uniformly randomised score. To determine the weight of this score, the random selection is included as an additional strategy to the *Datamodels* method.

4 Experiments

Datasets. We focus on representative few-shot learning datasets composed of different tasks with different numbers of classes and with different characteristics, such as class imbalance or amount of noisy samples or labels. Specifically, we use 6 widely-adopted text datasets, composed of tasks with different number of classes and characteristics, specifically: **20 News Group** (Lang, 1995) and **News Category** (Misra, 2022) for news category classification and **ATIS** (Hemphill et al., 1990), **Facebook** (Schuster et al., 2019), **HWU-64** (Liu et al., 2021) and **SNIPS** (Coucke et al., 2018) for intent classification. For each dataset, we select a subset of up to 200 labelled samples per class to simulate the limited data setting if needed (the impact of smaller dataset sizes is analysed in Appendix I). Furthermore, to evaluate the generalisability, we repeat the experiments on image data as well (see Appendix E for details).

Few-shot Learning Approaches. The evaluation is done in a **5-way 5-shot setting** using in-context learning on the text datasets and the gradient few-shot learning on both the text and image datasets. For in-context learning, we use the **Mistral-7B** (Jiang et al., 2023), **Zephyr-7B** (Tunstall et al., 2023), **LLaMA-3.1-8B** (Grattafiori et al., 2024), **Gemma-3-4B** (Team, 2025) and **Qwen-2.5-7B** (Team, 2024) instruction-tuned large language models. For gradient few-shot learning, we use representative and widely-used few-shot learning approaches, specifically 1) meta-learning approaches **Prototypical Networks** (Snell et al., 2017) and **MAML** (Finn et al., 2017), and 2) **Few-Shot Fine-Tuning** (Chen et al., 2019).

Baselines. We compare all selected single-property selection strategies (see also Table 2 in Appendix B) with 2 baselines: 1) **Classic selection**, where a new set of samples is randomly selected for each new task; and 2) **Random selection**, where only a single set of 5 samples per class is randomly selected once and used for every single task, representing a more realistic setting with limited budget. In-context learning is also compared with the recently proposed state-of-the-art in-context learning specific strategies **LENS** (Li and Qiu, 2023), **Active Prompt** (Diao et al., 2024), **EXPLORA** (Purohit et al., 2024) and **CASE** (Purohit et al., 2025).

Each selection strategy is run 10 times with different random seeds to determine their sensitivity to the effects of randomness (e.g., random initialisa-

tion or sample order). In addition, each experiment is repeated 5 times for different data splits (and the choice of the 200 labelled samples) and an average accuracy over these runs is reported (if not specified otherwise). Furthermore, each evaluation run is done on 600 randomly sampled tasks for gradient few-shot learning and 300 tasks for in-context learning (due to high computation costs of LLMs), where each task represents a random subset of 5 classes from the dataset (i.e., to construct the 5-way classification setting, following the standard few-shot learning methodology). As such, repeating the evaluation across multiple tasks and seeds deals with any possibility of overfitting and minimises the possible biases and other confounding factor in the evaluation. Further experiment setup details are in Appendix C.

4.1 Impact of Selection Strategies

In this section, our goal is to answer the following research question: ***RQ1:*** *What is the impact of single-property selection strategies on the few-shot learning?* The objective is to analyse the performance increase of the different selection strategies across various datasets and few-shot learning approaches. The results are shown in Figure 1 and Table 4 in Appendix D, aggregated over all the datasets (the results for image data are included in Appendix E and for individual datasets in supplementary material). We present the results separately for in-context learning and gradient few-shot learning, as we observed a strong dependence of the strategy benefit on the few-shot learning approach used. Additional ablation studies for the selection strategies are included in Appendix F.

The majority of selection strategies fail to outperform the *Classic selection* in case of gradient few-shot learning. In most cases, the strategies lead to performance on par or even lower than the *Classic selection* (for example diverse samples on text datasets). Only specific strategies consistently lead to better performance (e.g., combination of easy to learn and ambiguous samples, or *Forgetting* on text datasets). As such, the **random sample selection represents a strong baseline** for the gradient few-shot learning.

Sample selection is more beneficial for in-context learning. Many of the selection strategies lead to increase in performance for in-context learning, while only few strategies, such as the popular selection based on *similarity* or *diversity*, underperform the *Classic selection*. At the same time, **the increase in performance is higher than with gradient few-shot learning**, although with a larger variance in results – average increase of 1.5 for in-context learning as compared to 0.5 for gradient few-shot learning. In contrast to gradient few-shot learning, the hard to learn samples provide the most benefit for in-context learning. The reason may be that hard to learn samples provide a lot of information beneficial for the classification that cannot be easily leveraged in the gradient few-shot learning due to the information from the samples being hard to learn or retain. However, as there is no explicit "learning" in the in-context learning use of large language models, such samples can be better leveraged. As such, **learnability property is a strong indicator of quality for all few-shot learning approaches**. Selecting samples based on how easy (or hard for in-context learning) it is to learn them or how often they are forgotten consistently leads to the overall highest increase in performance.

4.2 ACSESS Results and Characteristics

In this section, we answer the following research question: ***RQ2:*** *How does the ACSESS method perform in comparison to the in-context learning specific baselines and single-property strategies?* Besides analysing and comparing the benefit with other strategies, we analyse the different weighting schemes of our method.

The ACSESS method consistently leads to performance increases across all few-shot learning approaches. Our proposed method outperforms all the investigated single-property strategies, leading to an average performance increase of up to 2.5 percentage points for in-context learning and 1.8 percentage points for gradient few-shot learning. Similar behaviour can be observed for the in-context specific baselines as well (LENS, Active Prompt, EXPLORA, CASE; see Table 4 and Figure 2 for comparison). Overall, the ACSESS method is able to outperform these baselines as well, achieving on average performance increase of 0.95 percentage points over the worst performing LENS method and 0.16 percentage points over the best performing CASE method. In all cases, the increase is statistically significant using the Wilcoxon signed-rank test. These results confirm that **combining simple and well-established selection strategies leads to performance on par or better than strategies specifically tailored for the task of sample selection for in-context learning.** Although the in-context learning specific strategies

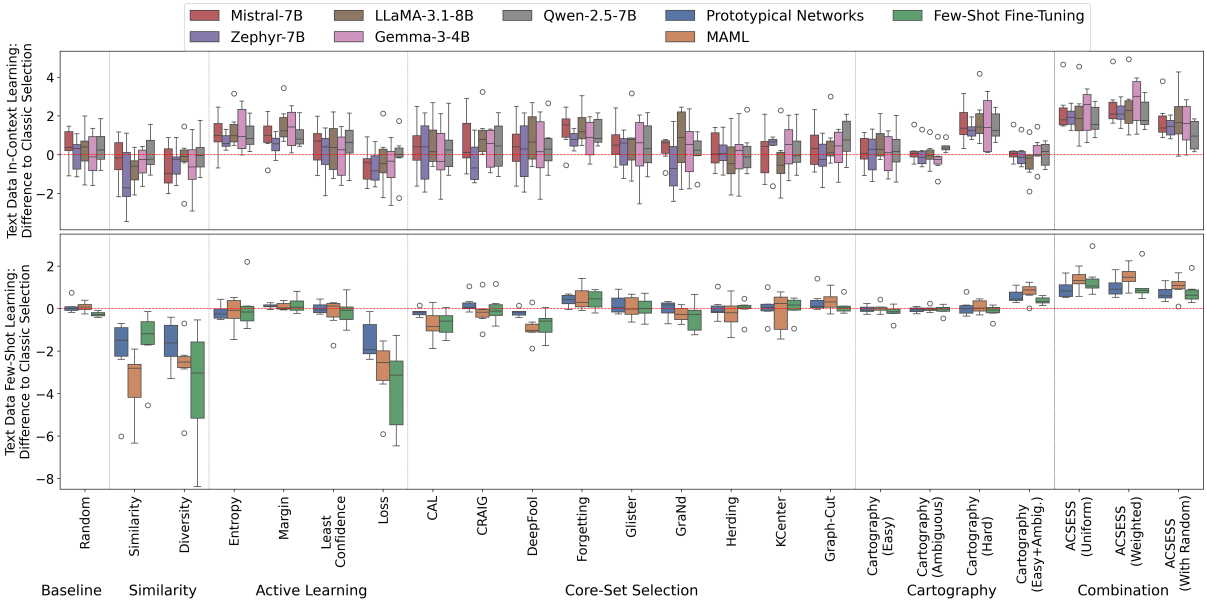

Figure 1: Benefit of the sample selection strategies, calculated as the difference in accuracy between the specific strategy and the classic few-shot selection (represented as the red dashed line), aggregated over the text datasets.

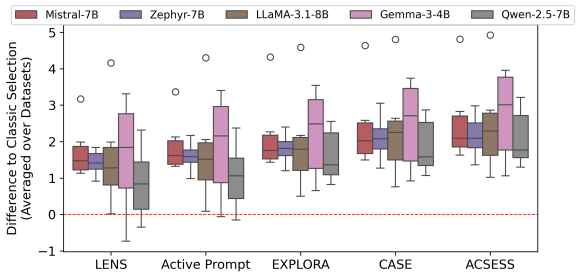

Figure 2: Comparison of the ACSESS method with in-context learning specific baselines.

may show their benefit in specific setting (e.g., similarity based selection for translation), they do not generalise as well to other tasks.

The ACSESS method identifies samples with complementary properties, with higher priority on learnability. For the different types of few-shot learning (in-context vs. gradient learning), our method identifies and combines different set of selection strategies. In case of in-context learning, the combination of hard to learn samples (*Cartography*) that are least often forgotten (*Forgetting*) on the decision boundary (*Margin*) and with highest *Entropy* provide the most benefit. For the gradient few-shot learning approaches on text datasets, the combination of easy and ambiguous samples (*Cartography*) that are least often forgotten (*Forgetting*) on the decision boundary (*Margin*) that provide additional informativeness from the *Graph-Cut* strategy provides the most benefit. Further analysing the identified strategies and their weights, we find the **ACSESS method gives higher importance to samples that are *learnable*, lower importance to *informative* samples** (but with multiple sources for both), **while the strategies that focus on *representativeness* are not included at all**. This further reinforces the finding that learnability is a strong indicator of quality for few-shot learning.

The dataset and approach specific weighting leads to the best performance. The only exception is the Few-Shot Fine-Tuning approach, where the uniform weighting shows the most benefit. At the same time, the weighting strategy that incorporates the random selection often leads to worse performance and higher variance in the results. Even though the uniform combination is computationally less expensive than the weighted combination, its performance increase is only slightly lower (average difference of $0.10-0.25$). In case of in-context learning, the uniform weighting is often on par with the best-performing CASE baslines. As such, the **uniform weighting represents a good trade-off** between the performance increase and the computation costs, making it in general a better choice.

Based on these observations, we can draw practical recommendations. If the higher computation cost is not a problem, or one is interested in the highest possible increase in performance, the full ACSESS method should be run – which includes running the full strategy identification for each dataset and model, and using the weighted com-

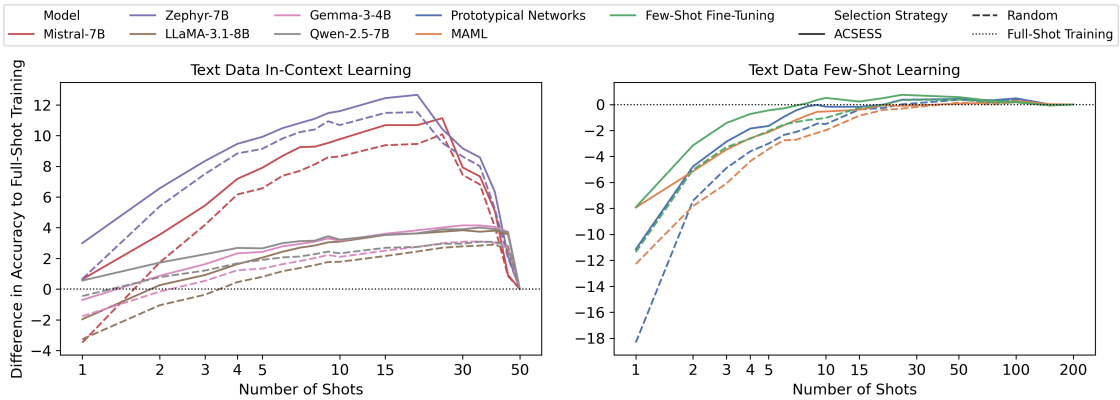

Figure 3: The change in benefit of selection strategies based on the number of shots used, aggregated over datasets. The change is calculated as a difference in accuracy to a setting using all available samples for training. When using a lower number of shots, the benefit of selection strategies over random selection is more significant. When using a larger number of shots, the benefit becomes negligible. In addition, the performance of full-shot training is matched, or in many cases exceeded, using smaller number of shots.

bination of the strategies. The cost can be reduced by aggregating the identification over some of the dimensions, such as across multiple dataset or models, which leads to lower benefit. However, if extensive resource are not available, or we want to determine the potential benefit of sample selection, using a uniform combination of the *Cartography* and *Margin* (and in some cases *Forgetting*) selection strategies provides a good approximation. This configuration does not incur any additional computation costs, while providing consistent improvement on all the datasets and models, which is only slightly lower than when using the model/dataset specific configuration.

4.3 Effects of Variable Number of Shots and Dataset Size

In this section, we answer the following research question: ***RQ3:*** *How do the number of selected shots and the size of the dataset affect the benefit of sample selection?* We perform two ablation studies to investigate how the benefits of sample selection strategies change as we: 1) increase or decrease the number of selected shots; and 2) decrease the number of available labelled samples (i.e., the size of the dataset). For the first ablation study, we apply the *ACSESS* method and *Random* selection to select a different number of shots and compare their benefit with training on all available samples (results aggregated over all datasets are in Figure 3 and for individual datasets in Appendix H). For the second ablation study, we apply the *ACSESS* method and *Random* selection to select 5 samples per class from dataset subsets of different sizes and compare their benefit to the selection from the full dataset (results aggregated over all datasets are in Figure 4 and for individual datasets in Appendix I).

The selection strategies provide more benefit on lower number of shots, while their impact with further increase in number of shots is negligible after a certain point. On the 1-shot setting, the difference in accuracy between the *Random* selection and the *ACSESS* strategy can be as high as 4 or 7 percentage points respectively for the in-context learning and gradient-based few-shot learning. In case of in-context learning, the impact is affected by the model used. For the older models (Mistral and Zephyr) the highest performance increase, as high as $10 - 12$ percentage points, is achieved at 20-25 shots. Afterwards, we observe a significant drop in performance, achieving similar performance to the 1-shot setting when using 50-shots. For newer models (LLaMA, Gemma and Qwen), we observe that increasing number of shots has only a negligible impact after around 5 shots, with the increase in performance around $2 - 3$ percentage points. However, we still observe a drop in performance when selecting 50 or more shots. The main reason for this drop-off may be the limited context size of the large language models, making the model degenerate to random guessing as the cross-over is text length dependent. For the gradient few-shot learning, the highest performance is achieved at lower number of shots (20) and then remains more or less constant (with slight decrease in some cases). In addition, increasing the number of selected shots in case of gradient-based few-shot learning leads to diminished impact, as the *ACSESS*

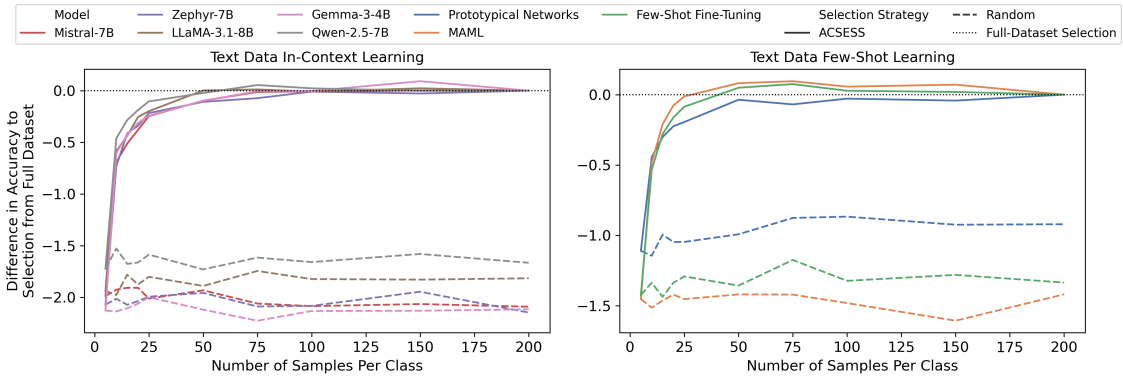

Figure 4: The difference in accuracy of selection strategies when using dataset subsets of different sizes, aggregated over the datasets. The difference is calculated as a comparison of the 5-shot selection using the dataset subset and the full dataset. Using only a fraction of dataset, the sample selection outperforms random baseline and quickly achieves similar performance to full dataset selection.

strategy regresses to random selection when selecting more than 30-40 shots. For in-context learning, we observe this behaviour only after the drop in performance. As such, the focus on **selecting high quality samples is paramount when selecting only a few labelled samples per class**, while the random selection can be safely used with higher number of shots. However, **labelling more samples and using them as additional shots may not always lead to better performance**.

Sample selection is beneficial even when using dataset with low number of samples. Using only 25% (50 samples per class) for in-context learning or 10% of the dataset (20 samples per class) for gradient few-shot learning, sample selection can still identify samples that lead to similar performance as is achieved when using the full dataset. Decreasing the size further, the benefit of sample selection starts to decrease as well. Using only 10 samples per class, the benefit decreases by approximately $20 - 40\%$ (e.g., from performance increase of 2 percentage points to only 1.3), and then quickly deteriorates to the *Random* selection (as the number of samples to choose from is too low).

5 Conclusion

In this paper, we investigate whether the well-established sample selection objectives can rival the strategies tailored specifically to in-context learning. To this end, we propose the *ACSESS* method for the automatic combination of the sample selection objectives to select samples with complementary properties and perform extensive evaluation and comparison of sample selection strategies on in-context learning and other few-shot learning approaches. The experimental results show that although the individual sample selection strategies may not always be beneficial, **their combination using our proposed method consistently leads to better performance, in many cases exceeding the performance from in-context learning specific strategies**. Curiously, we observe that the *learnability* of the samples is more important for few-shot learning than their *informativeness* or *representativeness*, which is not often utilised in prior works. Based on these results, the uniform combination of *Cartography*, *Forgetting* and *Margin* selection strategies is a good alternative transferable across all settings. Analysing the behaviour of the sample selection when changing the number of selected shots and the size of the dataset, we find that the sample selection provides the most benefit with a low number of shots, while with a higher number of shots, all strategies regress to random guessing. In addition, after a certain point, increasing the number of shots used for training no longer leads to improvement in the performance of few-shot learning, but may lead to a decrease in performance in some cases, especially for in-context learning. Overall, the benefit of selecting a subset of high-quality samples lies in improving the performance and reducing its variance for in-context learning, and in the reduction of computation cost, while providing a performance increase on the imbalanced and noisy datasets for the remaining few-shot learning approaches. Finally, even when the datasets contain only a few available samples per class, curating the samples further can often still lead to benefits, regardless of dataset characteristics or approach used.

Limitations

To diminish undesirable randomness effects, we report the results as an average over 5 repeats of data splitting, 10 repeated runs of the selection strategies (resulting in 50 overall evaluation runs), with each run evaluated on 600 (or 300 for in-context learning) tasks. Such setting limits the variance in results, but may also obscure the benefit of selection strategies to a certain extent. On the other hand, although we focus only on a 5-way few-shot learning setting (using only 5 classes per task), we do not expect significant change in result for most of the few-shot learning approaches when using higher number of classes (such as 10-way or 20-way classification). The only exception is the in-context learning, where we observed the dependence on input size and context width of the models. With higher number of classes, the observed performance of large language models would start to deteriorate to the performance that can be achieved with a low number of shots. In this case, using larger model sizes may lead to better utilisation of the samples and higher performance increase from the selection strategies.

Similarly to other research papers focusing on sample selection and few-shot learning via in-context learning, we assume an availability of large-enough annotated dataset from which we select samples using different strategies – we use up to 200 samples per class for each dataset for the main experiments. As such, we also have larger validation or held-out set available for evaluating the selection strategies and hyperparameters for our proposed methods or for the approaches themselves. If we considered the true few-shot learning setting, where we have only unlabelled data available, only the active learning strategies (i.e., Entropy, Margin and Least Confidence) could be used, as they work with unlabelled data only, or a different approach for sample selection should be employed, such as using weak-labels to select the best samples, or assign every single class to all the labels and perform the sample selection that way to select both best performing samples as well as the class for them (as it is done by Chang and Jia (2023)). **We address this (potentially most significant) limitation by running ablation studies to observe the benefit of sample selection** (using our proposed method and random selection) **when using smaller subsets of data** (i.e., selecting samples using only fraction of the dataset) **or using different number of shots per class.**

Before the experiments, we perform basic prompt-engineering for the in-context learning approaches and run hyperparameter optimisation for the gradient few-shot learning models. We follow the recommendations for good prompt engineering, using the meta-tags defined for each model (which lead to better performance of the models), while also not using overly-simple or overly-complicated prompts. The prompts we use are created based on dataset description, the prompts used in related works, our own experience, and the formats recommended for different models (e.g., taking inspiration from Sun et al. (2023)). As we do not perform any extensive or automatic prompt-engineering, these prompts may lead to lowered performance and in some cases also lower impact of the sample selection for the few-shot learning setting. However, running an extensive prompt-engineering would introduce another controllable parameter which could potentially introduce further bias into the results (e.g., different selection strategies showing different impact for different prompt formats).

Ethical Considerations

The experiments in this paper work with publicly available datasets, citing the original authors of each dataset (with the exception of the datasets contained in MetaAlbum, where we cite the creators of the aggregated dataset). All the datasets are used in accordance to their license and terms and conditions of use. We do not work with any personally identifiable information or offensive content and perform no crowdsourcing nor further data annotation. In addition, we are not aware of any potential ethical harms or negative societal impacts of our work, apart from the ones related to the advancement of the field of Machine Learning and Few-Shot learning (including large language models and in-context learning), such as the use of large amount of computation resources. Finally, we follow the license terms for all the models we use – all models and datasets allow their use as part of research. The large language models (Mistral, Zephyr, LLaMA, Gemma and Qwen), which we use for the experiments, may contain biases and may generate potentially offensive or harmful content. However, the authors of the models already reduce this potential bias as much as possible when training the models, while at the same time we

limit the output to few tokens and do not release any output of these models, which should further reduce the potential bias and negative impact.

Impact Statement: CO2 Emissions Related to Experiments The experiments presented in this paper used larger amount of compute resources in order to evaluate all the sample selection strategies for all the different few-shot learning approaches. Overall, all the experiments (including preliminary experiments whose results are not reported in this paper) were conducted using a private infrastructure, which has a carbon efficiency of 0.432 $kgCO_2eq/kWh$. A cumulative of approximately 2500 hours of computation was performed on hardware of type A100 PCIe 40GB (TDP of 250W). Total emissions are estimated to be 270 $kgCO_2eq$ of which 0 percent were directly offset. These estimations were conducted using the Machine-Learning Impact calculator. Whenever possible, we tried to reduce the compute resources used as much as possible. The most compute resources were used by the large language models – Mistral-7B, Zephyr-7B, LLaMA-3.1-8B (Grattafiori et al., 2024), Gemma-3-4B (Team, 2025) and Qwen-2.5-7B, and their further tuning. To reduce the computation costs, we have opted for using their smaller version and in cases when the further tuning is required, we fine-tune only the last classification layer (keeping the rest frozen). In addition, for gradient few-shot learning approaches, we opt for smaller models that do not require such an extensive computation costs, while still providing relevant results. Finally, whenever possible, we reuse the already performed computations for the selection strategies as well as using the compute optimised version of our proposed method (more information included in Appendix A), such as generating the feature representation for the different samples only a single time and re-using it whenever possible (i.e., using pre-trained models without further tuning for these representations).

Acknowledgments

This work was partially supported by the projects funded by the European Union under the EU Horizon 2020: *TAILOR*, GA No. 952215, by the European Union under the Horizon Europe: *vera.ai*, GA No. 101070093, *AI-CODE*, GA No. 101135437; and by the Slovak Research and Development Agency: *Modermed*, GA No. APVV-22-0414.

The presented results were obtained using the computational resources procured in the national project *National competence centre for high performance computing* (project code: 311070AKF2) funded by European Regional Development Fund, EU Structural Funds Informatization of Society, Operational Program Integrated Infrastructure.

References

Mayank Agarwal, Mikhail Yurochkin, and Yuekai Sun. 2021. On sensitivity of meta-learning to support data. *Advances in Neural Information Processing Systems*, 34:20447–20460.

Sharat Agarwal, Himanshu Arora, Saket Anand, and Chetan Arora. 2020. Contextual diversity for active learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, pages 137–153. Springer.

Sweta Agrawal, Chunting Zhou, Mike Lewis, Luke Zettlemoyer, and Marjan Ghazvininejad. 2023. In-context examples selection for machine translation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8857–8873, Toronto, Canada. Association for Computational Linguistics.

Aroof Aimen, Bharat Ladrecha, Sahil Sidheekh, and Narayanan C Krishnan. 2023. Leveraging task variability in meta-learning. *SN Computer Science*, 4(5):539.

Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, Colin Raffel, Shiyu Chang, Tatsunori Hashimoto, and William Yang Wang. 2024. A survey on data selection for language models. *Transactions on Machine Learning Research*. Survey Certification, Featured Certification.

Shengnan An, Bo Zhou, Zeqi Lin, Qiang Fu, Bei Chen, Nanning Zheng, Weizhu Chen, and Jian-Guang Lou. 2023. Skill-based few-shot selection for in-context learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13472–13492, Singapore. ACL.

Jan Cegin, Branislav Pecher, Jakub Simko, Ivan Srba, Maria Bielikova, and Peter Brusilovsky. 2025. Use random selection for now: Investigation of few-shot selection strategies in LLM-based text augmentation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 5533–5550, Suzhou, China. Association for Computational Linguistics.

Ting-Yun Chang and Robin Jia. 2023. Data curation alone can stabilize in-context learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8123–8144, Toronto, Canada. Association for Computational Linguistics.

Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. 2019. A closer look at few-shot classification. In *International Conference on Learning Representations*.

Yutian Chen, Max Welling, and Alex Smola. 2010. Super-samples from kernel herding. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*, UAI'10, page 109–116, Arlington, Virginia, USA. AUAI Press.

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

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, and 1 others. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *arXiv preprint arXiv:1805.10190*.

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

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the ACL: Human Language Technologies*, pages 4171–4186, Minneapolis, Minnesota. ACL.

Shizhe Diao, Pengcheng Wang, Yong Lin, Rui Pan, Xiang Liu, and Tong Zhang. 2024. Active prompting with chain-of-thought for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1330–1350, Bangkok, Thailand. Association for Computational Linguistics.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*.

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

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3816–3830, Online. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

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

Vivek Gupta, Akshat Shrivastava, Adithya Sagar, Armen Aghajanyan, and Denis Savenkov. 2022. RetroNLU: Retrieval augmented task-oriented semantic parsing. In *Proceedings of the 4th Workshop on NLP for Conversational AI*, pages 184–196, Dublin, Ireland. Association for Computational Linguistics.

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

Charles T. Hemphill, John J. Godfrey, and George R. Doddington. 1990. The ATIS spoken language systems pilot corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24-27,1990*.

Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. 2021. Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(9):5149–5169.

Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. 2022. Datamodels: Predicting predictions from training data. In *Proceedings of the 39th International Conference on Machine Learning*.

Rishabh Iyer, Ninad Khargoankar, Jeff Bilmes, and Himanshu Asanani. 2021. Submodular combinatorial information measures with applications in machine learning. In *Proceedings of the 32nd International Conference on Algorithmic Learning Theory*, volume 132 of *Proceedings of Machine Learning Research*, pages 722–754. PMLR.

Rishabh K Iyer and Jeff A Bilmes. 2013. Submodular optimization with submodular cover and submodular knapsack constraints. In *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, and 1 others. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Iman Jundi and Gabriella Lapesa. 2022. How to translate your samples and choose your shots? analyzing translate-train & few-shot cross-lingual transfer. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 129–150, Seattle, United States. Association for Computational Linguistics.

Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh Iyer. 2021. Glister: Generalization based data subset selection for efficient and robust learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8110–8118.

Abdullatif Köksal, Timo Schick, and Hinrich Schütze. 2022. Meal: Stable and active learning for few-shot prompting. *arXiv preprint arXiv:2211.08358*.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Machine Learning Proceedings 1995*, pages 331–339. Morgan Kaufmann, San Francisco (CA).

Itay Levy, Ben Bogin, and Jonathan Berant. 2023. Diverse demonstrations improve in-context compositional generalization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1401–1422, Toronto, Canada. ACL.

Xiaonan Li, Kai Lv, Hang Yan, Tianyang Lin, Wei Zhu, Yuan Ni, Guotong Xie, Xiaoling Wang, and Xipeng Qiu. 2023. Unified demonstration retriever for in-context learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4644–4668, Toronto, Canada. Association for Computational Linguistics.

Xiaonan Li and Xipeng Qiu. 2023. Finding support examples for in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6219–6235, Singapore. ACL.

Hongfu Liu and Ye Wang. 2023. Towards informative few-shot prompt with maximum information gain for in-context learning. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15825–15838.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 55(9).

Xingkun Liu, Arash Eshghi, Pawel Swietojanski, and Verena Rieser. 2021. Benchmarking natural language understanding services for building conversational agents. In *Increasing Naturalness and Flexibility in Spoken Dialogue Interaction: 10th International Workshop on Spoken Dialogue Systems*, pages 165–183. Springer.

Man Luo, Xin Xu, Zhuyun Dai, Panupong Pasupat, Mehran Kazemi, Chitta Baral, Vaiva Imbrasaite, and Vincent Y Zhao. 2023. Dr. icl: Demonstration-retrieved in-context learning. *arXiv preprint arXiv:2305.14128*.

Huan Ma, Changqing Zhang, Yatao Bian, Lemao Liu, Zhirui Zhang, Peilin Zhao, Shu Zhang, Huazhu Fu, Qinghua Hu, and Bingzhe Wu. 2023. Fairness-guided few-shot prompting for large language models. *arXiv preprint arXiv:2303.13217*.

Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. 2021. Active learning by acquiring contrastive examples. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 650–663, Online and Punta Cana, Dominican Republic. ACL.

Antonis Maronikolakis, Paul O'Grady, Hinrich Schütze, and Matti Lyra. 2023. Improving few-shot learning with multilingual transfer and Monte Carlo training set selection. In *Proceedings of the 2023 CLASP Conference on Learning with Small Data (LSD)*, pages 1–10, Gothenburg, Sweden. Association for Computational Linguistics.

Costas Mavromatis, Balasubramaniam Srinivasan, Zhengyuan Shen, Jiani Zhang, Huzefa Rangwala, Christos Faloutsos, and George Karypis. 2023. Which examples to annotate for in-context learning? towards effective and efficient selection. *arXiv preprint arXiv:2310.20046*.

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

Rishabh Misra. 2022. News category dataset. *arXiv preprint arXiv:2209.11429*.

Noor Nashid, Mifta Sintaha, and Ali Mesbah. 2023. Retrieval-based prompt selection for code-related few-shot learning. In *2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE)*, pages 2450–2462.

Tai Nguyen and Eric Wong. 2023. In-context example selection with influences. *arXiv preprint arXiv:2302.11042*.

Dongmin Park, Dimitris Papailiopoulos, and Kangwook Lee. 2022. Active learning is a strong baseline for data subset selection. In *Has it Trained Yet? NeurIPS 2022 Workshop*.

Panupong Pasupat, Yuan Zhang, and Kelvin Guu. 2021. Controllable semantic parsing via retrieval augmentation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7683–7698, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. 2021. Deep learning on a data diet: Finding important examples early in training. In *Advances in Neural Information Processing Systems*, volume 34, pages 20596–20607. Curran Associates, Inc.

Branislav Pecher, Ivan Srba, and Maria Bielikova. 2024. A survey on stability of learning with limited labelled data and its sensitivity to the effects of randomness. *ACM Comput. Surv.*, 57(1).

Kiran Purohit, Venktesh V, Sourangshu Bhattacharya, and Avishek Anand. 2025. Sample efficient demonstration selection for in-context learning. In *Forty-second International Conference on Machine Learning*.

Kiran Purohit, Venktesh V, Raghuram Devalla, Krishna Mohan Yerragorla, Sourangshu Bhattacharya, and Avishek Anand. 2024. EXPLORA: Efficient exemplar subset selection for complex reasoning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5367–5388, Miami, Florida, USA. Association for Computational Linguistics.

Chengwei Qin, Aston Zhang, Anirudh Dagar, and Wenming Ye. 2023. In-context learning with iterative demonstration selection. *arXiv preprint arXiv:2310.09881*.

Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B. Gupta, Xiaojiang Chen, and Xin Wang. 2021. A survey of deep active learning. *ACM Comput. Surv.*, 54(9).

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2655–2671, Seattle, United States. ACL.

Alexander Scarlatos and Andrew Lan. 2023. Reticl: Sequential retrieval of in-context examples with reinforcement learning. *arXiv preprint arXiv:2305.14502*.

Christopher Schröder, Andreas Niekler, and Martin Potthast. 2022. Revisiting uncertainty-based query strategies for active learning with transformers. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2194–2203, Dublin, Ireland. Association for Computational Linguistics.

Sebastian Schuster, Sonal Gupta, Rushin Shah, and Mike Lewis. 2019. Cross-lingual transfer learning for multilingual task oriented dialog. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3795–3805, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Richard Shin, Christopher Lin, Sam Thomson, Charles Chen, Subhro Roy, Emmanouil Antonios Platanios, Adam Pauls, Dan Klein, Jason Eisner, and Benjamin Van Durme. 2021. Constrained language models yield few-shot semantic parsers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7699–7715, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Kashun Shum, Shizhe Diao, and Tong Zhang. 2023. Automatic prompt augmentation and selection with chain-of-thought from labeled data. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12113–12139, Singapore. Association for Computational Linguistics.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30.

Yisheng Song, Ting Wang, Puyu Cai, Subrota K. Mondal, and Jyoti Prakash Sahoo. 2023. A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities. *ACM Comput. Surv.*, 55(13s).

Hongjin Su, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, and 1 others. 2022. Selective annotation makes language models better few-shot learners. In *The Eleventh International Conference on Learning Representations*.

Xiaofei Sun, Linfeng Dong, Xiaoya Li, Zhen Wan, Shuhe Wang, Tianwei Zhang, Jiwei Li, Fei Cheng, Lingjuan Lyu, Fei Wu, and 1 others. 2023. Pushing the limits of chatgpt on nlp tasks. *arXiv preprint arXiv:2306.09719*.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A. Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, Online. Association for Computational Linguistics.

Gemma Team. 2025. Gemma 3.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Robert Tibshirani. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. 2018. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations*.

Lewis Tunstall, Edward Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro von Werra, Clémentine Fourrier, Nathan Habib, and 1 others. 2023. Zephyr: Direct distillation of lm alignment. *arXiv preprint arXiv:2310.16944*.

Ihsan Ullah, Dustin Carrion, Sergio Escalera, Isabelle M Guyon, Mike Huisman, Felix Mohr, Jan N van Rijn, Haozhe Sun, Joaquin Vanschoren, and Phan Anh Vu. 2022. Meta-album: Multi-domain meta-dataset for few-shot image classification. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Joaquin Vanschoren. 2018. Meta-learning: A survey. *arXiv preprint arXiv:1810.03548*.

David Vilar, Markus Freitag, Colin Cherry, Jiaming Luo, Viresh Ratnakar, and George Foster. 2023. Prompting PaLM for translation: Assessing strategies and performance. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15406–15427, Toronto, Canada. Association for Computational Linguistics.

Jingyao Wang, Zeen Song, Xingzhe Su, Lingyu Si, Hongwei Dong, Wenwen Qiang, and Changwen Zheng. 2023. Learning to sample tasks for meta learning. *Preprint*, arXiv:2307.08924.

Shuohang Wang, Yichong Xu, Yuwei Fang, Yang Liu, Siqi Sun, Ruochen Xu, Chenguang Zhu, and Michael Zeng. 2022. Training data is more valuable than you think: A simple and effective method by retrieving from training data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3170–3179, Dublin, Ireland. Association for Computational Linguistics.

Max Welling. 2009. Herding dynamical weights to learn. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 1121–1128, New York, NY, USA. ACM.

Zhiyong Wu, Yaoxiang Wang, Jiacheng Ye, and Lingpeng Kong. 2023. Self-adaptive in-context learning: An information compression perspective for in-context example selection and ordering. *Preprint*, arXiv:2212.10375.

Xi Ye, Srinivasan Iyer, Asli Celikyilmaz, Veselin Stoyanov, Greg Durrett, and Ramakanth Pasunuru. 2023. Complementary explanations for effective in-context learning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4469–4484, Toronto, Canada. Association for Computational Linguistics.

Ruonan Yu, Songhua Liu, and Xinchao Wang. 2023. Dataset distillation: A comprehensive review. *arXiv preprint arXiv:2301.07014*.

Yury Zemlyanskiy, Michiel de Jong, Joshua Ainslie, Panupong Pasupat, Peter Shaw, Linlu Qiu, Sumit Sanghai, and Fei Sha. 2022. Generate-and-retrieve: Use your predictions to improve retrieval for semantic parsing. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4946–4951, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Mike Zhang and Barbara Plank. 2021. Cartography active learning. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 395–406, Punta Cana, Dominican Republic. ACL.

Yiming Zhang, Shi Feng, and Chenhao Tan. 2022. Active example selection for in-context learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9134–9148, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, and 1 others. 2024. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36.

A ACSESS Method Supplementary Details

In this appendix, we provide and discuss supplementary details of our proposed strategy. This includes detailed steps of the forward selection in Algorithm 2, backward selection in Algorithm 3 and datamodels selection in Algorithm 4. In addition, we provide low-level details how the method works, suggestions for using the method in order to consider its generalisability and reducing its model/dataset dependence, the computational costs of the method (in comparison with other baselines) and how we reduce them to provide compute optimised version of our method, and, finally, suggestions for selecting the best configuration of the ACSESS method based on the compute-performance trade-off of these configurations. Some parts of this section are based on the observations from our experiments.

Algorithm 2 Forward Selection

Require: N - number of samples to choose
Require: $\mathbf{S}$ - set of sample selection strategies
$\mathbf{S_F} \leftarrow \{\}$
$best_perf$ = -1
while Performance is increasing **or** $\mathbf{S_F}$ is empty **do**
  for all s in $\mathbf{S}$ **do**
    Create $\mathbf{S_{temp}}$ by combining $\mathbf{S_F}$ and s
    Set $w_s = 1/|\mathbf{S_{temp}}|$
    Calculate for each sample $score_{temp}(x) = \sum_{s \in S_{temp}} w_s * objective_s(x)$
    Calculate $perf_i$ by evaluating the few-shot model on N samples selected using $score_{temp}$
    if $perf_i > best_perf$ **then**
      $best_perf = perf_i$
      $strategy = s$
    end if
  end for
  if Any $strategy$ improves performance **then**
    Add found strategy to $\mathbf{S_F}$
    Remove found strategy from $\mathbf{S}$
  end if
end while
return $\mathbf{S_F}$

Algorithm 3 Backward Selection

Require: N - number of samples to choose
Require: $\mathbf{S}$ - set of sample selection strategies
$\mathbf{S_B} = \mathbf{S}$
Set $w = 1/|\mathbf{S_B}|$
Calculate for each sample $score_{temp}(x) = \sum_i^{|\mathbf{S_B}|} w_s * objective_s(x)$
Calculate $best_perf$ by evaluating the few-shot model on N samples selected using $score_{temp}$
while Performance is increasing **do**
  for all s_i in $\mathbf{S}$ **do**
    Create $\mathbf{S_{temp}}$ by removing s_i from $\mathbf{S_B}$
    Set $w_s = 1/|\mathbf{S_B}|$
    Calculate for each sample $score_{temp}(x) = \sum_{s \in S_B} w_s * objective_s(X)$
    Calculate $perf_i$ by evaluating the few-shot model on N samples selected using $score_{temp}$
    if $perf_i > best_perf$ **then**
      $best_perf = perf_i$
      $strategy = s$
    end if
  end for
  if Any $strategy$ improves performance **then**
    Remove found strategy from $\mathbf{S_B}$
  end if
end while
return $\mathbf{S_B}$

Algorithm 4 Datamodels Selection

Require: N - number of samples to choose
Require: $\mathbf{S}$ - set of sample selection strategies
Require: $perf$ - performance of the few-shot model on the baseline setting
$\mathbf{S_D} \leftarrow \{\}$
Determine number of random combinations to create M
Create a set $\mathbf{C}$ containing M unique, random combinations of strategies from $\mathbf{S}$, ensuring each strategy is contained at least 5 times
for all C_j in $\mathbf{C}$ **do**
  Set $w_s = 1/|\mathbf{C_j}|$
  Calculate for each sample $score_j(x) = \sum_{s \in C_j} w_s * objective_s(x)$
  Calculate $perf_j$ by evaluating the few-shot model on N samples selected using $scores_j$
  Calculate $diff_j = perf_j - perf$
  Add $diff_j$ to the list of differences to baseline $Diff$
  Create a new multi-label presence vector using strategies from C_j and add it to the list $Strategies$
end for
Train a $LASSO$ linear regression using $Strategies$ as features and $Diff$ as targets
Obtain vector of weights $\vec{W}$, each corresponding to one strategy in the multi-label presence vector
Select strategies with positive weights in $\vec{W}$ and add them to $\mathbf{S_D}$
Set W_D using weights of the selected strategies from $\mathbf{S_D}$
return $\mathbf{S_D}$, W_D

Considerations for the computation costs of the ACSESS method Although the proposed ACSESS method may at first seem to require significantly more computation resources than the single-property selection strategies or the in-context learning specific baselines, there are easy to apply considerations and modifications that increase the compute-efficiency and reduce the costs. We define this analysis into two separate parts as they introduce different computation costs: 1) using the method in practice after the optimal set of strategies is already identified; and 2) identifying the optimal set of strategies to combine. As all of the commonly used single-property strategies perform the selection by training the model for a few epochs (or just running an inference in order to obtain embeddings in case of Similarity and Diversity) and then using the training dynamics, these strategies can be run at the same time, reusing the same trained model. As such, when using the proposed ACSESS method in practice, we need to train the model only a single time and can use it across all the combined strategies. Therefore the cost in this case is equivalent to the single-property selection strategies (and lower than the in-context learning specific baselines). Similar optimisation can also be used when identifying the optimal set of strategies to combine, where the training of the model is the main computation cost. Even though the strategies are combined with different ones, they are designed to assign a specific score to each sample, which can then be reused in the combination. In other words, the strategies are needed to be run only once to obtain the scores which are then used as strong approximation for the objectives. The only cost that remains is the cost of evaluating the set of selected strategies by each of the combination. In case of

the gradient-based few-shot learning strategies, this cost is often negligible, as we are using only a small number of samples. However, in the case of in-context learning, the cost scales with the size of the model used and the number of combinations that are explored, as the inference itself is quite expensive. The cost of this evaluation step is also dependent on the number of samples and the model used (i.e., training on 25 samples vs. full dataset; inference on smaller vs. larger models). Due to this, the computation cost for the ACSESS strategy is higher (by the number of required inferences and the cost of single inference) over the single-property selection strategy. At the same time, the cost is lower or on par with in-context learning specific baselines, which use multiple inferences or trainings of the model while selecting the samples (even though such comparison is hard to make and is based only on our empirical observations during the experiments). Finally, the computation cost can be further reduced by foregoing the model and dataset specific tuning (e.g., using the ACSESS method on multiple datasets at the same time), as described in the following section. The specific cost comparison of the ACSESS with the baselines is presented in Table 1. The costs are calculated as the average GPU wall-clock time for each strategy averaged across all datasets.

Table 1: Comparison of costs, calculated as the average GPU wall-clock time across all datasets, for the individual sample selection strategies. The ACSESS strategy performs on par with in-context learning specific strategies, while showing higher cost than single property selection strategies.

STRATEGY	GPU TIME
Similarity/Diversity	0.56
Active Learning	1.02-1.11
Core-Set	0.97-1.24
LENS	6.19
Active Prompt	5.37
EXPLORA	4.98
CASE	2.04
$\text{ACSESS}_{\text{Weighted}}$	2.05
$\text{ACSESS}_{\text{Uniform}}$	1.48

Considerations for using the ACSESS method and reducing its model/dataset dependence Similarly to hyperparameter optimisation, we suggest to identify and combine the different strategies using the validation set to prevent any data leakage (i.e., evaluating the performance of different strategies using the validation set). At its core, the method is designed to be run separately for each model and dataset. Although such setting leads to the most benefit and highest increase in performance, this makes it significantly model/dataset dependent and requires significantly more computation resources. In order to reduce the model/dataset dependence, the different steps of the method be run across multiple models or datasets. This may be best illustrated on the identification of strategies and further weighting using the Datamodels method. Instead of having a separate set of combinations of strategies for each dataset a model, we can concatenate them together to create a single dataset. After training the LASSO method on this concatenated dataset, we obtain the relevant strategies and their weights for all the dataset and models. Similar modification can be applied to the forward and backward selection as well, where we evaluate the added/removed strategy on all the models/datasets, aggregate the score and select the strategies on this aggregated score. However, as we observed a strong dependence on the few-shot approach and the data modality, we suggest to run the ACSESS method separately for each few-shot learning approach and modality. In our experiments we also follow this suggestion and aggregate only over the datasets, as we have noticed only negligible changes when running the ACSESS method separately for each dataset (i.e., in some cases, the set of identified strategies we report in the results included 1 more strategy that performed really well on that specific dataset). Running the ACSESS strategy on even higher aggregation (over all the approaches, models and modalities) is not recommended as our preliminary experiments indicate that it would not lead to significant performance benefit and the method could fail to identify any relevant selection strategy in some cases. Finally, foregoing the specific model/dataset weighted combination and instead using the uniform combination (or the weights provided by the Datamodels approach across multiple datasets) can increase the transferability and generalisability of the ACSESS method further.

Choosing the best configuration for the ACSESS method Based on the results of our experiments, to get the best performance benefit of the ACSESS method, running the full strategy identification, either separately for each model and dataset or aggregated across datasets, and using the weighted combination of these strategies should be used. This

configuration provides the best results across all datasets, approaches and models, with the exception of Few-Shot Fine-Tuning. However, this may require extensive computation resources. If such resources are not available, or if we want to quickly determine the potential benefit of the sample selection, using a uniform combination of the *Cartography* and *Margin* (and in some cases *Forgetting*) selection strategies provides a good approximation. This configuration does not incur any additional computation costs, while providing consistent improvement on all the datasets and models, which is only slightly lower than when using the model/-dataset specific configuration. Although, running the full method should always provide the most benefit, this configuration (uniform combination of the 3 best performing strategies identified in our experiments) is a good alternative that is transferable across all settings.

B Sample Selection Strategies Details

In this appendix, we provide further details about the different single-property sample selection strategies that we evaluate in this paper, as well as for the baselines. In addition, we specify the hyperparameters of these methods and how we use them for selecting the set of samples. The high-level overview of the single-property strategies is included in Table 2.

The idea behind the **uncertainty-based** and **active learning** selection strategies is that the most informative samples with higher impact are the ones that the model is least certain about (Coleman et al., 2020; Park et al., 2022). These strategies select samples iteratively over multiple steps. In our experiments, we select 1 sample for every class in individual step. In addition, the first sample for every class is chosen randomly, as such setting was observed to lead to better active learning results (Park et al., 2022). When using these methods for combination, the samples that are selected after running all the required steps (e.g., 4 steps when choosing 5-shots) use the score assigned to them at their respective step when they were chosen, while the remaining samples are assigned the average score over all the selection steps. The **Entropy** strategy measures the entropy for each of the samples over the predicted class probabilities (which is used as the sample score) and selects the sample with highest entropy. The **Margin (Breaking Ties)** strategy assigns to each sample the difference between the probability of the two most probable classes. The samples with lowest such difference are selected as those are the ones the model is most uncertain about (and which can be considered on the decision boundary of the model). The **Least Confidence** strategy simply selects the samples where the most likely class has the lowest probability assigned to it. Finally, the **Loss** strategy selects samples with the highest loss.

The idea behind the core-set selection strategies is to select subset of samples that are representative of the whole dataset (i.e., the subset has similar properties to the full set). The included strategies select samples based on different criteria, such as error/loss, gradient or submodularity. These strategies select the full set of samples at the same time according to the assigned scores. As such, when using these strategies for combination, the full sample scores are used from this single iteration.

The **Decision Boundary** based methods select samples the are close to the decision boundary, as these represent the samples that are hard to separate and should provide the most information to the models. The **Contrastive Active Learning (CAL)** (Margatina et al., 2021) selects the samples whose predictive likelihood diverges the most from their neighbourhood as determined by the KL divergence. The **DeepFool** strategy (Ducoffe and Precioso, 2018) approximates the samples on boundary by perturbing the samples until the predicted class changes and selects the samples that require the smallest perturbation.

The idea behind **Error or Loss** based methods is that the samples that contribute more loss or error to the training of model have higher importance and should contribute more to the training when selected. The **GraNd** strategy (Paul et al., 2021) calculates the average contribution of each sample to the decline of the training loss at early stage of training across several independent training runs (calculating gradient norm expectation of the sample). The **Forgetting** strategy (Toneva et al., 2018) simply calculates how often the specific sample is incorrectly classified after being classified correctly in the previous epoch. The number of such forgetting events is used as the sample score and samples that are forgotten the most often (or least often) are selected. In our evaluation of single-property strategies, we consider only the setting of most often forgotten samples (as it has shown the best performance in the preliminary experiments). However, when using the strategy for combination,

Table 2: Common single-property selection strategies used in the ACSESS method, with the description of their objective, and categorised based on the sample property they consider and the group of selection strategies it belongs to.

	Group	Strategy	Description	Reference
Informativeness	Similarity	Similarity	Most similar samples.	—
	Similarity	Diversity	Most diverse/dissimilar samples.	—
	Active Learn.	Entropy	Highest entropy over class probabilities.	(Park et al., 2022)
	Active Learn.	Margin	Lowest difference in class probability between the 2 most probable classes.	(Park et al., 2022)
	Active Learn.	Least Confidence	Class with lowest assigned probability.	(Park et al., 2022)
	Active Learn.	Loss	Highest loss.	(Park et al., 2022)
	Core-Set	Contrastive Active Learning (CAL)	Predictive likelihood that diverges most from the neighbourhood as determined by KL divergence.	(Margatina et al., 2021)
	Core-Set	DeepFool	Smallest amount of perturbation to change class.	(Ducoffe and Precioso, 2018)
	Core-Set	GraNd	Highest contribution to decline of training loss.	(Paul et al., 2021)
	Core-Set	Graph-Cut	Submodularity function that measures diversity and informativeness.	(Iyer and Bilmes, 2013; Iyer et al., 2021)
Representativeness	Core-Set	Herding	Minimising distance of subset centre to full dataset.	(Welling, 2009; Chen et al., 2010)
	Core-Set	KCenter	Minimising distance of every sample in subset to full dataset.	(Sener and Savarese, 2018; Agarwal et al., 2020)
	Core-Set	CRAIG	Gradients representative of full dataset.	(Mirzasoleiman et al., 2020)
	Core-Set	Glister	Bi-level optimisation maximising log-likelihood.	(Killamsetty et al., 2021)
Learnability	Core-Set	Forgetting	How often the samples are forgotten.	(Toneva et al., 2018)
	Cartography	Cartography	How easy it is to learn the samples. Due to no obvious consensus, all of easy, ambiguous, hard and combination of easy and ambiguous samples are considered.	(Swayamdipta et al., 2020; Zhang and Plank, 2021)

we also consider the setting of selecting the least forgotten ones.

The **Submodularity** based methods use submodular functions (Iyer and Bilmes, 2013; Iyer et al., 2021), which naturally measure the informativeness and diversity of a subset of samples. We evaluate the selection based on **Graph-Cut**, as the method performed the best from all the different submodular functions in the preliminary experiments.

The **Geometry** based methods assume that samples that are close in the feature space have similar properties. As such, these methods try to remove samples that are close to each other, as they provide redundant information. Removing enough of such samples then leads to a small subset of sample that cover the properties of the whole dataset. The **Herding** strategy (Welling, 2009; Chen et al., 2010) iteratively and greedily adds one sample to the subset which minimises the distance between the centre of the subset and the centre of the whole dataset. On the other hand, the **KCenter** strategy (Sener and Savarese, 2018; Agarwal et al., 2020) selects sample that minimises the largest distance between each sample already in the subset and its closest sample not yet included in subset.

The **Gradient Matching** based methods try to find a subset of samples whose gradient can imitate the whole dataset. The **CRAIG** strategy (Mirzasoleiman et al., 2020) converts the problem into maximisation of a monotone submodular function which is then solved greedily.

Finally, the **Bilevel Optimisation** based methods convert the problem into a bilevel optimisation, with different objectives in the outer and inner level. The **Glister** strategy (Killamsetty et al., 2021) uses a validation set on the outer optimisation, where the subset selection is used as objective, and log-likelihood in the inner optimisation, which handles model parameter optimisation.

In addition, we also evaluate and use sample selection based on the **Cartography** (Swayamdipta et al., 2020; Zhang and Plank, 2021), which measures how easy or hard it is to learn different samples. This ease of learning is determined by observing the training dynamics of all the samples across few epochs, looking at the average confidence/probability of the correct class and the variance of this confidence. The samples with high confidence and low variance are considered to be

the *easy* to learn samples. At the same time the samples with small confidence and small to medium variance are considered the *hard* to learn ones. The remaining samples are considered to be *ambiguous* (medium confidence or samples with high variance). We explore four different settings in our experiments and choose the samples accordingly: 1) **Easy** samples, where we sort the samples based on confidence and choose the top-K samples with highest confidence for each class; 2) **Hard** samples, where we sort the samples based on confidence and choose the bottom-K samples for each class (i.e., the lowest confidence samples); 3) **Ambiguous**, where we first calculate average confidence and standard deviation of all the samples, select the samples whose confidence is close around the average confidence (in the interval defined by 1 standard deviation around the average confidence), and then randomly sample from them; and 4) **Easy + Ambiguous**, where we choose half of the samples from the *easy* set and the other half from *ambiguous* set.

All the approaches mentioned so far perform the selection using a model that is trained for some epochs. The number of epochs the training is done for significantly affects the outcome of different strategies. For this reason, we search for an optimal number of epochs by evaluating the selection strategies and the samples they select on a single run. The optimal number of epochs determined by this search is 10% of the full epochs used for the few-shot fine-tuning models. This number represents a good trade-off between the computational cost of the methods and the quality of samples. Running the training for full number of epochs may result in better selection of samples, but we consider such setting to not be representative. In addition, we use the same number of epochs for each strategy to keep the comparison as representative as possible. In addition, we also search for an optimal set of hyperparameters for the selection strategies (such as the learning rate, the optimiser used or batch size). The optimal set of hyperparameters that are used throughout the experiments is the use of the Adam optimiser with Cosine Annealing learning rate scheduler, with learning rate of 0.01 for image data and 0.0001 for text data and a batch size of 64.

To keep the results as comparable as possible, each selection strategy uses the same base model as the one used in the main experiments, i.e., ResNet-18 (He et al., 2016) pretrained on Imagenet dataset (Deng et al., 2009) for image data and pretrained BERT-base (Devlin et al., 2019) for text data in case of gradient few-shot learning, and the same large language models (Mistral (Jiang et al., 2023) and Zephyr (Tunstall et al., 2023)), but with only their last classification layer being learned, for in-context learning. In case the selection strategy requires sample features, we use the output of last fully-connected layer (the input of classification layer) as the feature vector similar to Guo et al. (2022).

We also evaluate the sample selection based on **Similarity** and **Diversity** as it represents popular approach for in-context learning. To select the samples, we use the cosine similarity between the feature representation (obtained from the penultimate layer) of the samples. For in-context learning, we instead the most similar (for Similarity) or the most dissimilar (for Diversity) samples for each test sample as is common for this approach (Sun et al., 2023). To select the samples for gradient few-shot learning, we first randomly select one sample for each class. Afterwards, we expand the set of selected samples by iteratively adding the most similar (for Similarity) or most dissimilar (for Diversity) sample to the average feature representation of the set of selected samples for each class (i.e., if we already have 5 selected samples in the specific class, we calculate the similarity to the representation of each of these 5 samples and take an average over these similarities as the final score).

Finally, for the in-context learning specific baselines, when these baselines need a base model for optimising the selection (e.g., evaluating the selected samples as part of the method or running different steps), we use all the same models as during all of the experiments (Mistral, Zephyr, LLaMA, Gemma and Qwen). In other words, we select a set of samples for each of the models as done for the gradient-based few-shot learning as well. This also includes running a hyperparameter search informed by the recommendations of the authors to identify the optimal setup for the search. In addition, whenever necessary, we modify the source code to allow for running the method.

For the core-set selection strategies, we use the modified implementation of the methods released by Guo et al. (2022) and its extension provided by Park et al. (2022) for active learning methods. For the Cartography strategy, we use the modified implementation released by the original authors Swayamdipta et al. (2020).

C Experiments Setup: Further Details

The experiments and evaluation of different sample selection strategies are done on the within-domain few-shot learning, where the classes in training, validation and testing datasets all come from the same domain (e.g., from the same dataset). Each evaluation run is done on 600 randomly selected tasks for the gradient few-shot learning approaches and 300 tasks for the in-context learning approaches (due to high computation costs of LLMs). Each task is a 5-way 5-shot (or K-shot for experiments where we change the number of shots) classification task evaluated on 16 test samples per class, following the standard few-shot learning methodology and the evaluation from Ullah et al. (2022). The performance of a single evaluation run is reported as the mean accuracy across these tasks and the different splits of data into labelled/unlabelled and train-validation-test sets (we use 5 splits of data, where in each we split the available data into train-validation-test sets and select 200 samples per class for training, as mentioned in the main part of the paper). These scores are then used to report the findings in the main experiments, either as mean and standard deviation over multiple repeated runs of the specific sample selection strategy or over the results from different models and datasets to produce the aggregated figures. As such, we do not explicitly report the standard deviation caused by the different data splits or different sampled tasks. However, the observed deviation from these two factors is quite small (on average up to 0.1 in terms of accuracy), as it is calculated over large number of repeats. This is also the reason why we can observe zero standard deviation on some of the sample selection strategies.

For the in-context learning approaches, we use the instruction-tuned Mistral-7B (Jiang et al., 2023) (instruct v0.1), Zephyr-7B (Tunstall et al., 2023) (alpha version), LLaMA-3.1-8B (Grattafiori et al., 2024), Gemma-3-4B (Team, 2025) and Qwen-2.5-7B (Team, 2024) large language models in full precision (although 4-bit precision has shown minimal impact on evaluation). We perform no further training for these models (except in case of sample selection, which is covered in Appendix B). Instead we run a basic prompt engineering to find the best performing template for each model. In the end, we opted for the single template that performed the best on average for each model and each dataset, which is illustrated in Table 3, along with the verbaliser (i.e., the possible outputs of the model that are mapped to the classes in the dataset). Finally, the in-context learning models are set to provide deterministic outputs (controlling the temperature and disabling sampling) and set to generate maximum of 10 tokens. In case where multiple words that can be mapped to a dataset class are generated, the output is treated as incorrect as we consider the model to be just hallucinating and not working as intended (this behaviour was mainly observed when using the models in zero-shot or 50-shot setting, where the model either does not understand the task or the context is not sufficient for it to follow the instructions and often either repeats all the classes or generates further sentences).

For the gradient few-shot learning approaches we use a pretrained BERT-base (Devlin et al., 2019) model for the text data and the ResNet-18 (He et al., 2016) for image data, pretrained on Imagenet dataset (Deng et al., 2009), as base models. All the models are further trained on each dataset using the different few-shot learning approaches, namely the Prototypical Networks (Snell et al., 2017), Model-Agnostic Meta-Learning (MAML) (Finn et al., 2017) and Few-Shot Fine-Tuning (Chen et al., 2019). In case of Few-Shot Fine-Tuning, the further training is done on the concatenation of all the meta-training classes and their corresponding samples, while at the meta-test time, only the last layer is replaced and fine-tuned on the specific task with 5 classes. The Prototypical Networks and MAML follow the typical few-shot training setting, training on large number of 5-way few-shot tasks. The meta-learning approaches on image data are trained on 15 000 tasks and the Few-Shot Fine-Tuning approach is trained on 15 000 randomly sampled batches of size 16. On text data, the number of sampled tasks and batches is increased to 25 000. In addition, the MAML approach uses 5 training iteration and 10 testing iterations on image data and 10 training and 15 testing on text data, while Prototypical Networks use only 1 in both cases. Finally, all approaches use the Adam optimiser with 0.001 (meta) learning rate and 0.01 base learning rate. The full training process is repeated 3 times and the best performing model, selected using a separate validation dataset, is selected for use throughout the reported experiments. For the LASSO model we use the alpha value of 0.001.

All the hyperparameters for all models were determined using a hyperparameter search on each data modality and model using a single validation

Table 3: Prompt formats and verbalisers used for different datasets in the paper. The News classification datasets include **20 News Group** and **News Category** datasets and Intent classification includes **ATIS**, **Facebook**, **HWU-64** and **SNIPS**. The *[Option 1-5]* are replaced with the names of the classes as defined by the verbaliser. The *[Input]* is replaced by the sentence of the samples and the *[Output]* is replaced with the name of class as defined by the verbaliser. The *[Input]* and *[Output]* are repeated for each in-context sample, while the final *[Output]* is used to determine the predicted class. The same format is used for all in-context learning models as it was the best performing one from the ones tried during basic prompt engineering and following the recommendations from (Sun et al., 2023) and our own experience.

Datasets	**Prompt Format**
News classification	Determine category of the sentence using following options: 1) *[Option 1]* 2) *[Option 2]* 3) *[Option 3]* 4) *[Option 4]* 5) *[Option 5]*. *[Input]* *[Output]*
Intent classification	Determine intent of the sentence using following options: 1) *[Option 1]* 2) *[Option 2]* 3) *[Option 3]* 4) *[Option 4]* 5) *[Option 5]*. *[Input]* *[Output]*
Dataset	**Verbaliser**
20 News Group	{IBM, Middle East Politics, Windows XP, Motorcycles, Medicine, For Sale, Religion, MS Windows, Baseball, Auto, Hockey, Mac, Graphics, Christianity, Guns, Electronics, Space, Crypto, Atheism, Politics}
News Category	{Politics, World News, Parenting, Money, Wellness, Business, Weddings, Entertainment, Impact, Black Voices, Queer Voices, Crime, Divorce, Food and Drink, Worldpost, Parents, Travel, The Worldpost, Healthy Living, Taste, Media, Culture and Arts, Style, Weird News, Style and Beauty, Comedy, Home and Living, Sports, Environment, Education, Good News, Fifty, Women, Science, Arts and Culture, US News, Green, Tech, Religion, Latino Voices, College, Arts}
ATIS	{Flight, Abbreviation, City, Airfare, Ground Service, Ground Fare, Airline, Flight Number, Aircraft, Distance, Capacity, Flight Time, Quantity, Airport}
Facebook	{Road Condition, Departure, Duration, Event, Arrival, Directions, Distance, Traffic, Update, Combine, Route, Location}
HWU-64	{Set Alarm, Definition, Play Music, Set Calendar, Play Radio, Confirm, Quirky, Send Email, Currency, Like, Social Query, News, Neutral, Stop, Calendar Query, Alarm Query, Recommend Movie, Play Audiobook, Weather Query, Praise, Social Post, Play Podcast, Negate, Recipe, Affirm, Remove Calendar, Recommend Location, Remove List, Explain, Email Query, Maths, Stock, Transport Query, Order Takeaway, Datetime Query, Repeat, Factoid, Taxi, Add List, Add Contact, Recommend Event, Mute Volume, List Query, Ticket, Convert Datetime, Joke, Remove Alarm, Traffic, Volume Up, Takeaway Query, Play Game, Contact Query, Music Query, Volume Other, Volume Down, Music Setting, Greet, Dislike}
SNIPS	{Playlist, Weather, Event, Musing, Creative Work, Rate Book, Book Restaurant}

split. For the experiments, we use a modified implementation provided by the MetaAlbum (Ullah et al., 2022), extended to further approaches, models and datasets. In addition, we also use the hyperparameters used in the paper as a starting point for the hyperparameter search.

D Additional Results: Table Results from Text Datasets

In this appendix, we provide the results from the main experiments in table form – in Table 4.

E Additional Experiments: Performance on Image Data

In this appendix, we present the setup and results of the main experiments done on image data. The datasets are chosen similarly to text datasets, where we focus on datasets with different characteristics, such as class imbalance and noisy datasets. Specifically, we use 8 datasets from the **Meta-Album** (Ullah et al., 2022), specifically **LR_AM.DOG** and **LR_AM.AWA** for animal classification, **VCL.APL** for plane classification, **HUM_ACT.ACT_410** for classification of human actions, **MNF.TEX_DTD** for classification of texture, **MCR.PRT** and **MCR.PNU** for classification of microscope images, and **PLT.FLW** for flower classification.

The results of the main experiments, where we investigate the impact of different single-property sample selection strategies on few-shot learning, as well as provide the comparison with the proposed ACSESS method, are presented in an aggregated form in Table 5 and Figure 5. Similarly to experiments on text data, the results for each individual

Table 4: Benefit of the different selection strategies calculated as the difference in accuracy to the *classic* few-shot selection strategy, aggregated over the text datasets. The subscript represents the standard deviation of the difference over the aggregated datasets.

STRATEGY	TEXT DATA - ICL					TEXT DATA - FSL		
	MISTRAL	ZEPHYR	LLAMA	GEMMA	QWEN	PROTONET	MAML	FT
Classic	$+0.00_{0.00}$	$+0.00_{0.00}$	$+0.00_{0.00}$	$+0.00_{0.00}$	$+0.00_{0.00}$	$+0.00_{0.00}$	$+0.00_{0.00}$	$+0.00_{0.00}$
Random	$+0.46_{0.86}$	$+0.01_{0.82}$	$+0.25_{1.10}$	$-0.04_{1.07}$	$+0.38_{0.89}$	$+0.10_{0.30}$	$+0.07_{0.21}$	$-0.26_{0.11}$
Similarity	$-0.24_{1.11}$	$-1.22_{1.61}$	$-0.77_{0.81}$	$-0.41_{0.76}$	$+0.08_{0.92}$	$-2.15_{1.82}$	$-3.51_{1.50}$	$-1.56_{1.45}$
Diversity	$-0.66_{1.08}$	$-0.42_{0.80}$	$-0.23_{1.19}$	$-0.63_{1.35}$	$+0.03_{0.95}$	$-1.65_{1.02}$	$-2.77_{1.55}$	$-3.65_{2.69}$
Entropy	$+1.02_{0.98}$	$+0.70_{0.38}$	$+1.32_{0.91}$	$+1.25_{1.12}$	$+1.02_{0.69}$	$-0.17_{0.33}$	$-0.20_{0.68}$	$+0.07_{1.03}$
Margin	$+0.93_{0.94}$	$+0.51_{0.50}$	$+1.59_{0.94}$	$+1.34_{0.97}$	$+1.02_{0.61}$	$+0.12_{0.10}$	$+0.11_{0.17}$	$+0.18_{0.35}$
Least Confidence	$+0.49_{1.03}$	$+0.04_{1.17}$	$+0.38_{1.27}$	$+0.00_{1.15}$	$+0.57_{1.10}$	$+0.02_{0.25}$	$-0.25_{0.72}$	$-0.14_{0.59}$
Loss	$-0.56_{0.88}$	$-0.65_{0.83}$	$-0.26_{1.34}$	$-0.59_{1.15}$	$-0.03_{1.17}$	$-1.49_{0.89}$	$-3.00_{1.46}$	$-3.75_{1.92}$
CAL	$+0.39_{1.27}$	$+0.18_{1.56}$	$+0.58_{1.17}$	$-0.04_{1.48}$	$+0.35_{1.23}$	$-0.18_{0.18}$	$-0.76_{0.69}$	$-0.69_{0.54}$
Craig	$+0.66_{1.36}$	$-0.38_{1.03}$	$+1.03_{1.10}$	$+0.33_{0.98}$	$+0.47_{1.22}$	$+0.22_{0.40}$	$-0.15_{0.69}$	$-0.00_{0.62}$
DeepFool	$+0.40_{1.29}$	$+0.18_{1.52}$	$+0.88_{1.25}$	$+0.25_{1.59}$	$+0.44_{1.18}$	$-0.18_{0.18}$	$-0.90_{0.65}$	$-0.76_{0.59}$
Forgetting	$+1.27_{0.96}$	$+0.77_{0.41}$	$+1.46_{0.88}$	$+0.81_{0.95}$	$+1.17_{0.70}$	$+0.39_{0.26}$	$+0.49_{0.54}$	$+0.41_{0.42}$
Glister	$+0.64_{0.98}$	$+0.21_{0.92}$	$+0.60_{1.41}$	$+0.29_{1.60}$	$+0.48_{1.19}$	$+0.19_{0.43}$	$+0.06_{0.49}$	$+0.03_{0.47}$
Grand	$+0.29_{0.62}$	$-0.54_{1.41}$	$+0.70_{1.60}$	$+0.29_{1.43}$	$+0.14_{0.87}$	$-0.01_{0.38}$	$-0.27_{0.34}$	$-0.40_{0.67}$
Herding	$+0.24_{0.91}$	$+0.11_{0.78}$	$-0.26_{1.26}$	$-0.00_{1.32}$	$+0.15_{1.11}$	$+0.03_{0.51}$	$-0.24_{0.69}$	$-0.04_{0.45}$
KCenter	$+0.13_{1.24}$	$+0.31_{0.88}$	$-0.30_{1.39}$	$+0.41_{1.18}$	$+0.23_{1.02}$	$+0.04_{0.58}$	$-0.15_{0.91}$	$+0.03_{0.49}$
Graph-Cut	$+0.42_{1.11}$	$-0.17_{0.92}$	$+0.59_{1.15}$	$+0.25_{1.00}$	$+0.83_{0.99}$	$+0.34_{0.50}$	$+0.35_{0.44}$	$+0.10_{0.33}$
$\text{Cartography}_{\text{Easy}}$	$+0.16_{0.76}$	$+0.06_{0.98}$	$+0.60_{0.85}$	$+0.04_{0.96}$	$+0.23_{1.08}$	$-0.04_{0.17}$	$+0.02_{0.21}$	$-0.19_{0.31}$
$\text{Cartography}_{\text{Ambiguous}}$	$+0.21_{0.64}$	$+0.02_{0.64}$	$+0.04_{0.61}$	$-0.26_{0.68}$	$+0.40_{0.26}$	$-0.06_{0.12}$	$-0.01_{0.13}$	$-0.06_{0.21}$
$\text{Cartography}_{\text{Hard}}$	$+1.60_{0.93}$	$+1.26_{0.39}$	$+1.82_{1.18}$	$+1.53_{1.38}$	$+1.47_{0.70}$	$+0.05_{0.37}$	$+0.08_{0.29}$	$-0.14_{0.28}$
$\text{Cartography}_{\text{Easy+Ambig.}}$	$+0.21_{0.64}$	$+0.02_{0.64}$	$-0.33_{0.93}$	$+0.15_{0.78}$	$+0.02_{0.59}$	$+0.59_{0.30}$	$+0.78_{0.39}$	$+0.37_{0.16}$
LENS	$+1.73_{0.70}$	$+1.42_{0.32}$	$+1.57_{1.31}$	$+1.61_{1.41}$	$+0.87_{0.91}$	N/A	N/A	N/A
Active Prompt	$+1.89_{0.71}$	$+1.59_{0.36}$	$+1.72_{1.32}$	$+1.89_{1.28}$	$+1.05_{0.85}$	N/A	N/A	N/A
EXPLORA	$+2.17_{1.00}$	$+1.81_{0.38}$	$+1.98_{1.29}$	$+2.23_{1.10}$	$+1.60_{0.68}$	N/A	N/A	N/A
CASE	$+2.40_{1.07}$	$+2.10_{0.56}$	$+2.33_{1.28}$	$+2.47_{1.11}$	$+1.87_{0.71}$	N/A	N/A	N/A
$\text{ACSESS}_{\text{Uniform}}$	$+2.30_{1.11}$	$+1.93_{0.48}$	$+2.08_{1.30}$	$+2.21_{1.08}$	$+1.78_{0.70}$	$+0.92_{0.41}$	$+1.34_{0.45}$	$\mathbf{+1.36_{0.75}}$
$\text{ACSESS}_{\text{Weighted}}$	$\mathbf{+2.55_{1.08}}$	$\mathbf{+2.15_{0.53}}$	$\mathbf{+2.48_{1.25}}$	$\mathbf{+2.73_{1.13}}$	$\mathbf{+2.09_{0.74}}$	$\mathbf{+1.02_{0.44}}$	$\mathbf{+1.49_{0.47}}$	$+1.07_{0.69}$
$\text{ACSESS}_{\text{With Random}}$	$+1.78_{0.98}$	$+1.42_{0.45}$	$+1.85_{1.38}$	$+1.55_{1.05}$	$+0.99_{0.73}$	$+0.72_{0.33}$	$+1.03_{0.48}$	$+0.80_{0.54}$

dataset and model are presented in supplementary material.

Although the results on the image modality are slightly different, the main findings hold true. Many selection strategies perform worse than the *Classic selection* (such as hard to learn samples), while only specific ones lead to consistently better performance (e.g., easy to learn samples or those that are forgotten the least).

A specific new finding that we can draw is that **the overall benefit of sample selection is higher for noisy datasets and limited for high-quality datasets.** Sample selection leads to a considerable performance increase over the *Classic selection*, on average around 2 percentage points, for the datasets with many noisy samples (as determined by the *Cartography*) or for datasets with large class imbalance (e.g., difference between lowest and median number of samples per class in few hundred samples), such as HUM_ACT.ACT_410, MCR.PRT or MCR.PNU. At the same time, performance increase is quite limited (on average around 0.1 percentage points) for the datasets with few noisy samples or no extensive class imbalance, such as the VCL.APL, PLT.FLW or NewsCategory.

When it comes to the proposed ACSESS strategy, we still observe the priority on learnability properties. In the case of the image data, the combination of easy to learn samples (*Cartography*) that are located on the decision boundary (*Margin*) provides the most benefit. This represents the combination of the lowest number of strategies we have observed so far, which indicates that the sample selection strategies are well-designed for image data, or that the image datasets contain high-quality samples that are easier to identify.

Finally, when it comes to the impact of increasing the number of shots, or using datasets of different sizes, we still observe similar findings on image data (see Figure 6). When changing the number of shots, the highest performance is achieved around $10 - 30$ shots and then remains the same across all of further increases. Similarly, when selecting from datasets of smaller sizes, we can observe the same performance when using as few as 50 samples (as compared to using the full dataset). At the same time, we observe that the more sophisticated selection strategies outperform random selection most

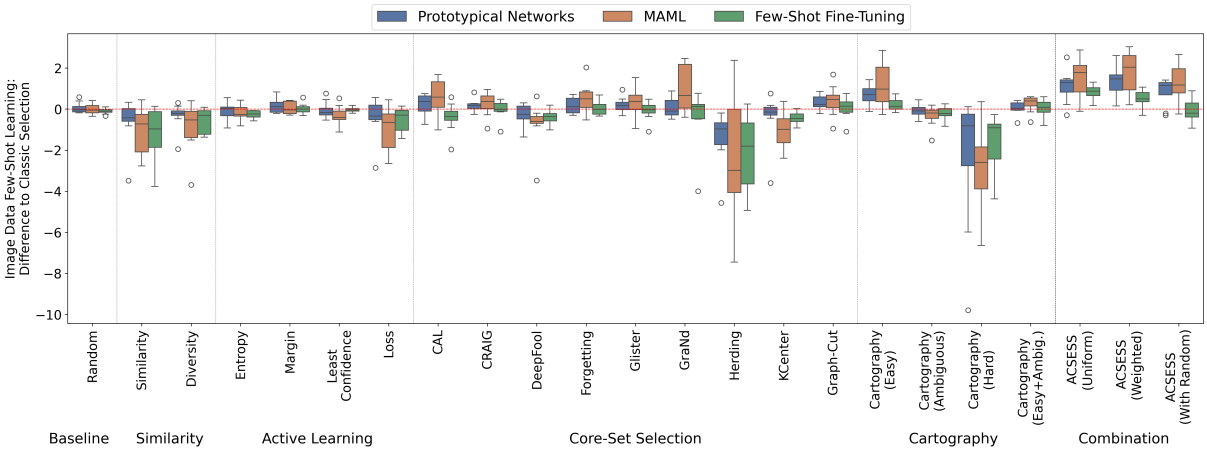

Figure 5: Benefit of the different sample selection strategies, calculated as the difference in accuracy between the specific strategy and the classic few-shot selection, aggregated over the image datasets (box plots show the distribution of results across the datasets). The performance of the classic selection is represented as the red dashed line (zero value). The consistently beneficial selection strategies depend on the approach (meta-learning vs. fine-tuning). Our proposed method ACSESS consistently leads to improved performance.

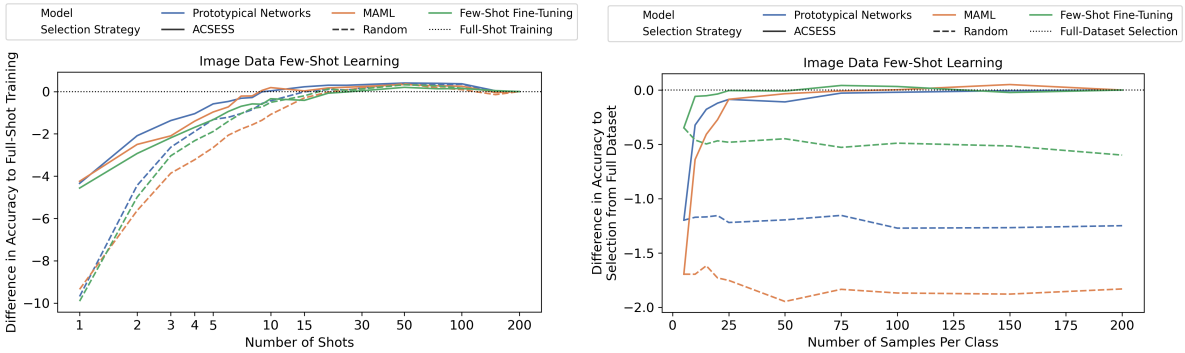

Figure 6: The change in benefit of selection strategies based on the number of shots (left) and the difference in accuracy of selection strategies when using dataset subsets of different sizes (right) aggregated over the image datasets.

of the time, while degrading to random selection on a high number of shots.

F Additional Experiments: Dataset and Model Dependence of Sample Selection Strategies and their Sensitivity to Random Components

In this appendix, we provide additional insights from ablation studies. We are mainly focus on: 1) how the impact of selection strategies is affected by the used models and datasets, i.e., look at the model and dataset dependence of different selection strategies and configurations of our proposed method (based on dataset specific results); and 2) how sensitive the strategies are to their random components, i.e., look at how the set of selected samples changes when the strategies are run multiple times and how this affects the overall performance (based on the variance from repeated strategy runs in Figure 7).

Specific strategies show large sensitivity to the effects of random seed. Changing the random seed for specific strategies often leads to different set of selected samples and different performance impact. The core set selection strategies are less affected by this randomness. Specific core set strategies (such as *Glister* or *Graph-Cut*) consistently select the same set of samples and show no variance in results. Other strategies core set strategies (e.g., *Forgetting*, *DeepFool* or *Herding*) show only small variance. On the other hand, strategies with random initial condition, such as the similarity selection or some of the active learning strategies, are more sensitive to the effects of the random seed. **The ACSESS method shows lower sensitivity to**

Table 5: Benefit of the different selection strategies calculated as the difference in accuracy to the *classic* few-shot selection strategy, aggregated over the image datasets. The subscript represents the standard deviation of the difference over the aggregated datasets.

STRATEGY	IMAGE DATA - FSL		
	PROTONET	MAML	FT
Random	$+0.06_{0.26}$	$-0.00_{0.27}$	$-0.10_{0.13}$
Similarity	$-0.64_{1.14}$	$-1.05_{1.15}$	$-1.21_{1.24}$
Diversity	$-0.35_{0.65}$	$-0.93_{1.22}$	$-0.57_{0.58}$
Entropy	$-0.12_{0.43}$	$-0.17_{0.37}$	$-0.26_{0.18}$
Margin	$+0.15_{0.34}$	$+0.05_{0.29}$	$+0.02_{0.26}$
Least Confidence	$-0.05_{0.41}$	$-0.31_{0.46}$	$-0.03_{0.12}$
Loss	$-0.44_{1.01}$	$-1.02_{1.07}$	$-0.53_{0.57}$
CAL	$+0.19_{0.54}$	$+0.54_{0.88}$	$-0.43_{0.72}$
CRAIG	$+0.18_{0.29}$	$+0.26_{0.60}$	$-0.04_{0.45}$
DeepFool	$-0.29_{0.51}$	$-0.77_{1.11}$	$-0.41_{0.36}$
Forgetting	$+0.16_{0.37}$	$+0.53_{0.72}$	$+0.03_{0.34}$
Glister	$+0.20_{0.37}$	$+0.33_{0.73}$	$-0.10_{0.45}$
GraNd	$+0.09_{0.49}$	$+0.95_{1.11}$	$-0.42_{1.41}$
Herding	$-1.43_{1.31}$	$-2.36_{2.94}$	$-2.14_{1.78}$
KCenter	$-0.45_{1.24}$	$-1.03_{0.87}$	$-0.42_{0.31}$
Graph-Cut	$+0.32_{0.36}$	$+0.40_{0.75}$	$+0.05_{0.53}$
$\text{Cartography}_{\text{Easy}}$	$+0.65_{0.50}$	$+1.17_{1.11}$	$+0.21_{0.28}$
$\text{Cartography}_{\text{Ambiguous}}$	$-0.08_{0.36}$	$-0.36_{0.50}$	$-0.21_{0.32}$
$\text{Cartography}_{\text{Hard}}$	$-2.42_{3.35}$	$-3.03_{2.26}$	$-1.71_{1.47}$
$\text{Cartography}_{\text{Easy+Ambig.}}$	$+0.05_{0.33}$	$+0.24_{0.40}$	$+0.04_{0.40}$
$\text{ACSESS}_{\text{Uniform}}$	$+1.12_{0.80}$	$+1.59_{0.97}$	$\mathbf{+0.81_{0.35}}$
$\text{ACSESS}_{\text{Weighted}}$	$\mathbf{+1.31_{0.75}}$	$\mathbf{+1.83_{0.97}}$	$+0.49_{0.41}$
$\text{ACSESS}_{\text{With Random}}$	$+0.85_{0.65}$	$+1.32_{0.91}$	$-0.06_{0.56}$

repeated runs (with the exception of weighting with random element), **but still may lead to different sets of samples based on the random seed** (as the standard deviation is higher than in core set strategies, i.e., up to 0.5 for the ACSESS method as opposed to 0 or 0.1 for majority of the core set strategies).

Many selection strategies show strong dataset dependence. In many cases, the strategies that perform well on one dataset (such as *DeepFool* on the HUM_ACT.ACT_410 dataset) may perform quite poorly on other datasets (such as MCR.PRT or MCR.PNU). Due to this dependence, such strategies cannot be used on any dataset without first evaluating them how well they perform on it. **Only few strategies perform consistently on the majority of the datasets**, such as *Herding* showing poor performance on all image datasets, or *Cartography* showing good performance across all datasets and models (while the most beneficial configuration of this strategy is dependent on data modality). Finally, we **do not observe a significant dataset dependence of the ACSESS method**, as it identifies the well-performing strategies and leverages their strengths for the specific dataset. This observation holds through for the weighting schemes as well (i.e., both *uniform* and *weighted* combinations show consistent performance across datasets).

Many selection strategies show strong model dependence and sensitivity to selected samples. The strategies performing well for one model (*GraNd* on MCR.PNU for MAML, or *KCenter* on 20 News for Zephyr) may result in a decrease of performance for other models (*GraNd* for Few-Shot Fine-Tuning, or *KCenter* for Mistral on the same setting). We do not observe such strong dependence for our proposed method ACSESS, with the exception for Few-Shot Fine-Tuning, where the uniform combination is always better than the weighted combination. Overall, having a different set of selected samples, either as a result from using different strategy or from repeated runs of the same strategy, **affects the Few-Shot Fine-Tuning less**, often leading to smaller benefit. At the same time, the **MAML** method and the **in-context learning** models (Mistral and Zephyr) are **more affected by the selected samples**, leading to more varied results and often higher performance increase. Such sensitivity to sample selection was observed in previous works as well (Pecher et al., 2024; Köksal et al., 2022; Zhang et al., 2022; Li and Qiu, 2023). Curiously, Prototypical Networks gain quite large

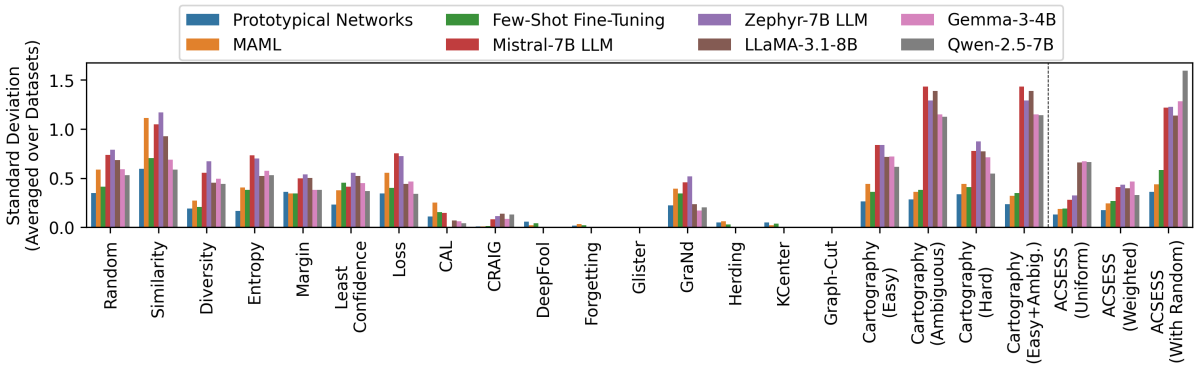

Figure 7: Standard deviation introduced by multiple runs of the different selection strategies. The results for Prototypical Networks, MAML and Few-Shot Fine-Tuning are aggregated over both the image and text datasets, while the results for Mistral and Zephyr are only from text datasets. The ACSESS method shows lower sensitivity to repeated runs compared to the majority of the strategies included in the combination, but still higher sensitivity than specific core set strategies that always lead to the same set of samples.

performance increase from sample selection, but show low sensitivity to the selected samples and variance from repeated runs of sample selection. As such, **sample selection is more important for MAML and In-Context Learning and less important for Few-Shot Fine-Tuning.**

G Additional Experiments: Comparison of Sample Selection Impact Between 5-Shot and 10-Shot Setting

To better explore the effect of increasing number of selected shots on the sample selection strategies, we repeat the evaluation of all the selection strategies in the 10-shot setting (using the same setup and methodology) and compare the results to 5-shot setting. For both 5-shot and 10-shot setting, we provide a distribution of the results as a difference to the *Classic* selection and illustrate the comparison of these distributions in Figure 8.

Overall, when moving from 5-shot to 10-shots, we observe an increase in the overall performance of the models for all the selection strategies (with the exception of some bad performing ones) as well as the *Classic* selection. At the same time, the increase in the performance of the *Classic* selection is higher, leading to lower difference and lower impact of the different evaluated selection strategies. This finding holds for all strategies across all models and few-shot learning approaches.

In addition, we also compare the results using Wilcoxon signed-rank test, between the results of 5-shot and 10-shot setting. The resulting p-value ($1.93e-07$) indicates that the different to Classic selection is more significant in case of 5-shot setting.

All in all, these experiments further reinforce the findings from comparison of ACSESS and *Random* selection, where we observe that the the sample selection progressively leads to lower impact as we increase the number of shots we select.

H Additional Experiments: Impact of Increasing Number of Shots for Individual Datasets

In this appendix, we provide results from investigating the impact of increasing the number of shots on the benefit of sample selection separately for the individual dataset. For each dataset, we increase the number of selected shots, starting from 1-shot up to the full dataset. The datasets HUM_ACT.ACT_410 and MCR.PRT contain only up to 50 samples per class when all samples are used and the MNF.TEX_DTD only up to 100. All the remaining datasets either contain at maximum 200 samples, or are subsampled to 200 samples per class if necessary (this is especially done for the text datasets, which are often significantly larger). For in-context learning, we increase the number of shots only up to 50 samples per class, as increasing the number further resulted in significantly lower performance (mainly due to limited context size of the large language models). The results for the image datasets are included in Figure 11, for the text datasets using the gradient few-shot learning in Figure 10 and for text datasets using the in-context learning in Figure 9.

For the majority of the datasets, we observe that the highest performance of the sample selection

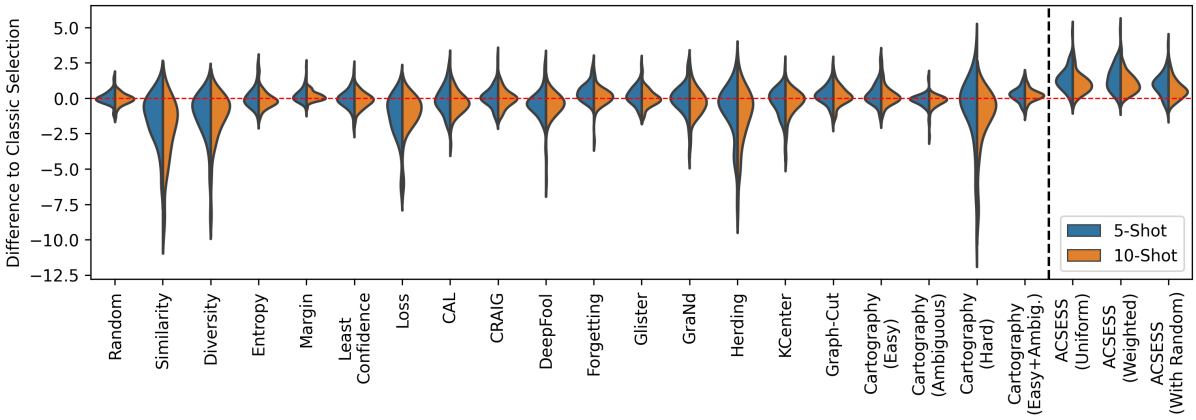

Figure 8: Distribution of the selection strategies benefit (calculated as difference to the classic few-shot selection) for the 5-shot and 10-shot setting. The benefit of the different selection strategies is more significant at lower number of shots.

using the ACSESS method is achieved on lower number of shots and then either remains the same until the full dataset, or gradually decreases as we use more samples. Only for specific datasets (LR_AM.DOG and LR_AM.AWA datasets for all approaches; PLT.FLW and SNIPS for MAML; and MNF.TEX_DTD for Prototypical Networks and MAML) we observe continual increase of performance when using sample selection as we increase the number of shots, with the highest performance being achieved only at the full dataset (or slightly beforehand). In addition, on specific datasets, not all models behave the same, for example, on the MNF.TEX_DTD dataset, only the Few-Shot Fine-Tuning approach achieves highest performance on lower number of shots (which then monotonically decrease as we increase the number of used samples), while the MAML and Prototypical Networks meta-learning approaches achieve the highest performance only when using almost the full dataset.

On the other hand, using the Random selection, we often observe monotonic increase in performance as the number of the selected shots increases, with the highest performance being achieved on the full dataset. This behaviour can be observed across all datasets and few-shot learning approaches (with exception of in-context learning). As such, this further reinforces our findings and main motivation that the data-centric approach can be beneficial for the few-shot learning. Curating a subset of the most informative and high-quality samples can often lead to performance on par or better than the one when using all the samples for training.

Finally, we also observe that the benefit of sample selection is more significant at lower number of shots. The difference in performance between the ACSESS method and the Random selection on the lower number of shots is significantly higher and slowly decreases as we use more shots for training. The exception is MAML on the PLT.FLW and SNIPS datasets, or Few-Shot Fine-Tuning on PLT.FLW and MCR.PNU datasets, where the performance of the sample selection is similar to the one using random selection even when using small number of shots. After a certain point, the sample selection using the ACSESS method degrades to Random selection. As such, the focus on selecting high quality samples is paramount when selecting only few labelled samples per class, while random selection can be safely used when using higher number of shots.

I Additional Experiments: Selection From Datasets of Different Sizes

In this section, we analyse how the benefit of sample selection strategies changes as we decrease the size of the dataset available. For this ablation study, we apply the *ACSESS* method and *Random* selection to select 5 samples per class from dataset subsets of different sizes and compare their benefit to the selection from the full dataset. Each dataset is subsampled into multiple sizes, starting with 5 samples per class up to the full dataset. Each subsampling is repeated 10 times and an average result over these repeats are reported. The full-dataset setting represents the selection from 200 samples per class for each datasets (for datasets with a lower number of samples per class, such as HUM_ACT.ACT_410, MCR.PRT or MNF.TEX_DTD, we observe a constant line

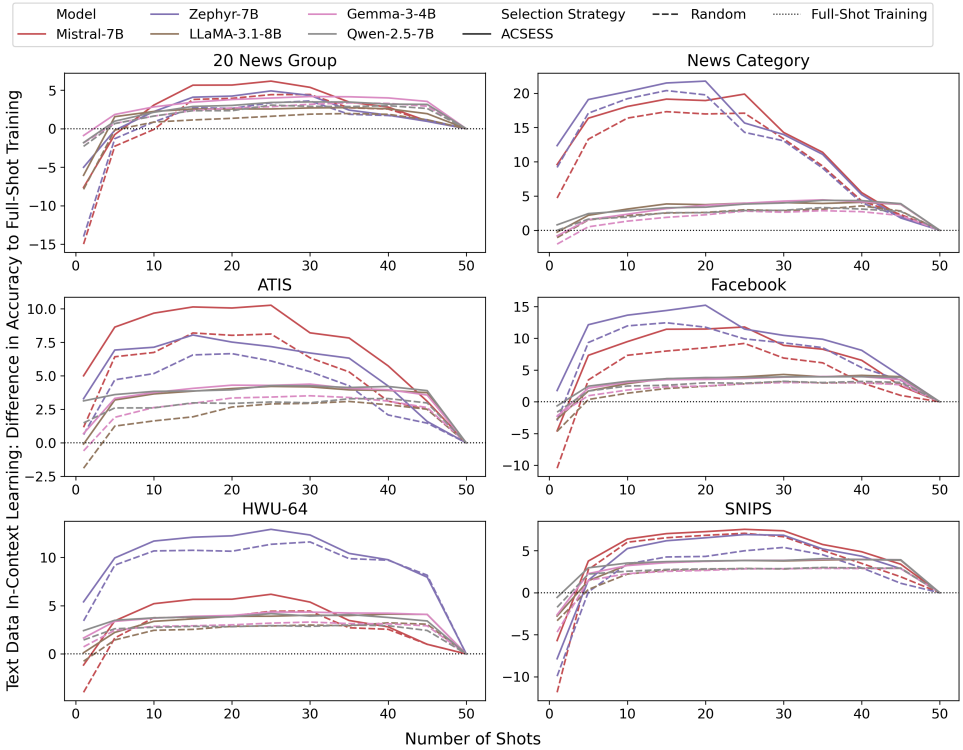

Figure 9: The different benefit of the selection strategies based on the number of selected shots for the text datasets using in-context learning. The reported benefit is calculated as a difference in performance to the one when using the full dataset, for both the ACSESS method and the Random selection.

after the full dataset was achieved). In all cases we perform selection of 5 shots per class using the ACSESS method and Random selection. Results aggregated over all datasets are in Figure 4 and for individual datasets for the image datasets are included in Figure 14, for the text datasets using the gradient few-shot learning in Figure 13 and for text datasets using the in-context learning in Figure 12.

Sample selection is beneficial even when using dataset with low number of samples. Using only 25% (50 samples per class) for in-context learning or 10% of the dataset (20 samples per class) for gradient few-shot learning, sample selection can still identify samples that lead to similar performance as is achieved when using the full dataset. Decreasing the size further, the benefit of sample selection starts to decrease as well. Using only 10 samples per class, the benefit decreases by approximately $20 - 40\%$ (e.g., from performance increase of 2 percentage points to only 1.3), and then quickly deteriorates to the *Random* selection (as the number of samples to choose from is too low).

For majority of the combinations of datasets and approaches, we can observe that sample selection can achieve similar performance using only a fraction of the dataset. For the gradient few-shot learning, this performance is achieved using only 10% of the dataset (20 sample per class), while for in-context learning, this performance is achieved using 25% of the dataset (50 samples per class). However, we observe that this strongly depends on the dataset and the model/approach used. For example on the ATIS dataset and in-context learning, using only 10 samples per class is enough to achieve the same performance as when selecting from the full dataset, while on the SNIPS dataset the number of samples per class increase up to 100. Similarly, on the ATIS dataset with gradient few-shot learning, the MAML a Few-Shot Fine-Tuning approaches achieve the similar performance using only 15-20 samples per class, while Prototypical Networks require the full dataset. Curiously, in specific cases, we can observe that running sample selection on subset of the dataset can lead to better performance than when performing selection from the full dataset (such as on the News Category dataset and gradient few-shot learning).

At the same time, we observe no change in impact of random selection from datasets of different sizes. In this case, the performance remains more or less constant (and often significantly below selection using the ACSESS method) throughout all

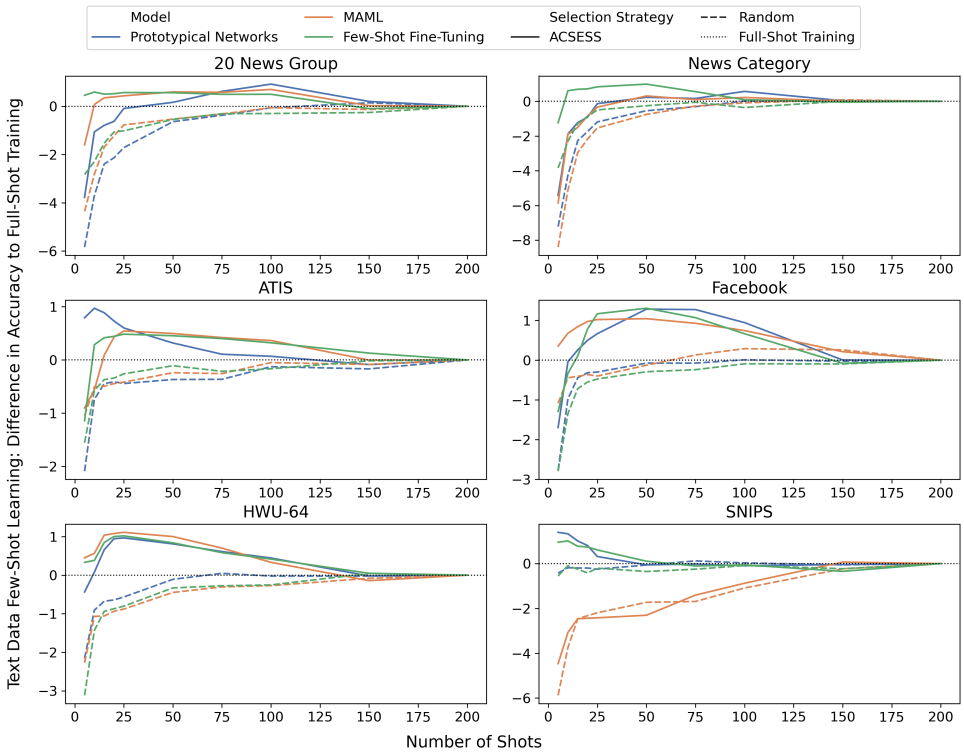

Figure 10: The different benefit of the selection strategies based on the number of selected shots for the text datasets using gradient few-shot learning. The reported benefit is calculated as a difference in performance to the one when using the full dataset, for both the ACSESS method and the Random selection.

the sizes – only difference in performance stems from the random subsampling. Finally, using only 10 samples per class, we can observer that sample selection using the ACSESS method can lead to quite big performance improvement over the random selection. Therefore we can conclude that the sample selection is beneficial even when using a small fraction of the dataset size as the one used in the main experiments, although it will lead to a lowered benefit of the sample selection and its impact on the overall performance (which is still higher than when using random selection).

J Few-Shot Learning Problem Formulation

In this section, we introduce the definition of the few-shot learning problem and the difference between gradient-based few-shot learning and few-shot learning via in-context learning. Note that in-context learning can be applied only to text datasets as it utilises pretrained large language models.

For the standard few-shot learning, we assume a larger collection of instances $D = (x_i, y_i)$, with larger number of classes. From this dataset, a separate task $D_T \subset D$ is created by randomly sampling few classes N. The task is characterised by a training set (also called support set, S) and test set (also called query set, Q). To create these sets, few samples (k) are sampled for each class. The few-shot learning algorithms then use the few samples from the support set S_T to learn to solve the task D_T quickly and are evaluated on this ability using the query set Q_T. This represents the N-way k-shot classification. The few-shot learning approaches mostly consist of two phases, training phase and testing phase. Often, these two phases work with separate sets of D_T, i.e., the classes appearing as part of tasks in the testing phase do not appear during training. As each task works only with subset of classes, the evaluation is often done on many such tasks and the average performance from these tasks is reported.

The gradient-based few-shot learning approaches follow episodic training similar to typical supervised training to gather knowledge and experience and learn to quickly adapt when encountering new task. Using the support set, the model is adapted to the task using gradient descent based approaches and then evaluated on the query set. Based on the evaluation, some kind of knowledge on how to better adapt is saved in part of the

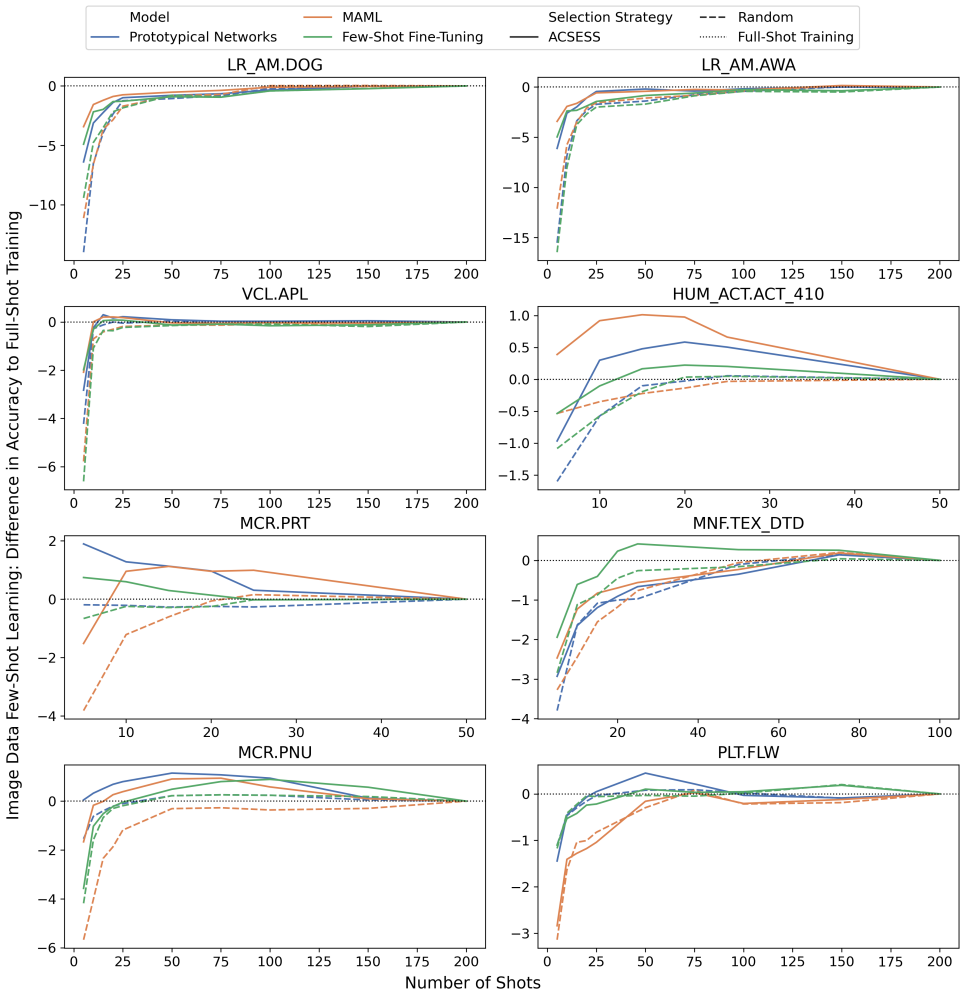

Figure 11: The different benefit of the selection strategies based on the number of selected shots for the image datasets using gradient few-shot learning. The reported benefit is calculated as a difference in performance to the one when using the full dataset, for both the ACSESS method and the Random selection.

model, while the rest of the model is reset. This is done for a large number of episodes (often called meta-training), with the model accumulating the knowledge. After enough episodes are run, the model is evaluated (in a meta-testing phase) using the exact same setup.

On the other hand, no training phase is present in few-shot learning via in-context learning. Instead, a pretrained large language model is used without any additional parameter updates. For each new task, a task-specific text prompt is formed by concatenating task-specific instructions, the samples from the support set along with their labels, and a specific test sample. After being given this prompt, the large language model generates a prediction in the form of a word (or multiple words). These generated words are then mapped to the task labels, and a class is assigned. The process is repeated for each sample in the query set. For example, in the case of binary classification, the task-specific instruction may be "Determine sentiment of following sentence", and the concatenated samples from the support set may be formatted as "sentence, sentiment: positive/negative", with the word "positive" mapping to the class representing positive sentiment and "negative" to the class representing negative sentiment. As such, the only supervision for the large language model is from the instruction and the concatenation of support samples. Therefore, the model is limited by its maximum input length. At the same time, this approach was found to be sensitive to the selected samples in the support set and their order (Zhang et al., 2022; Pecher et al., 2024; Chang and Jia, 2023).

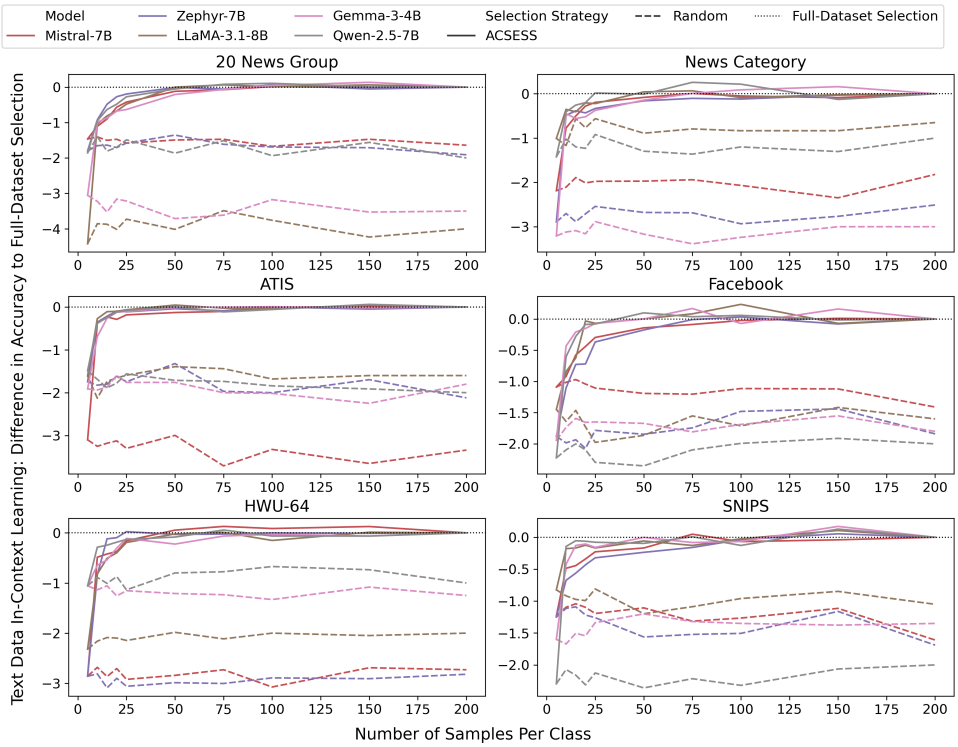

Figure 12: The difference in performance of sample selection strategies on dataset subsets of different sizes for the text datasets using in-context learning. The reported difference is in comparison to using sample selection from the full dataset.

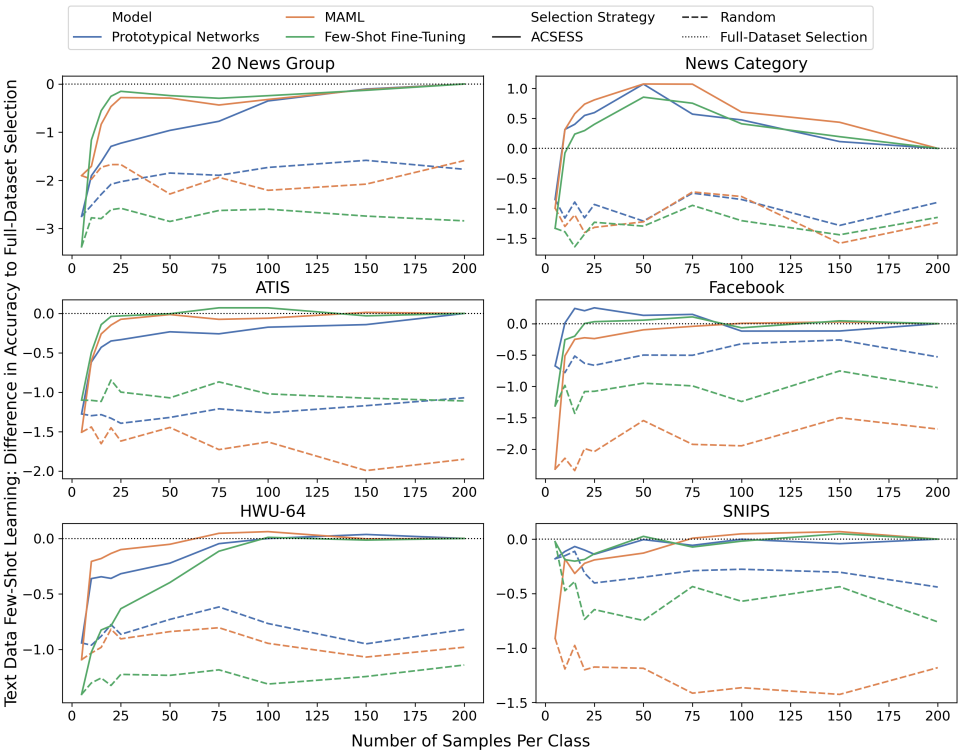

Figure 13: The difference in performance of sample selection strategies on dataset subsets of different sizes for the text datasets using gradient few-shot learning. The reported difference is in comparison to using sample selection from the full dataset.

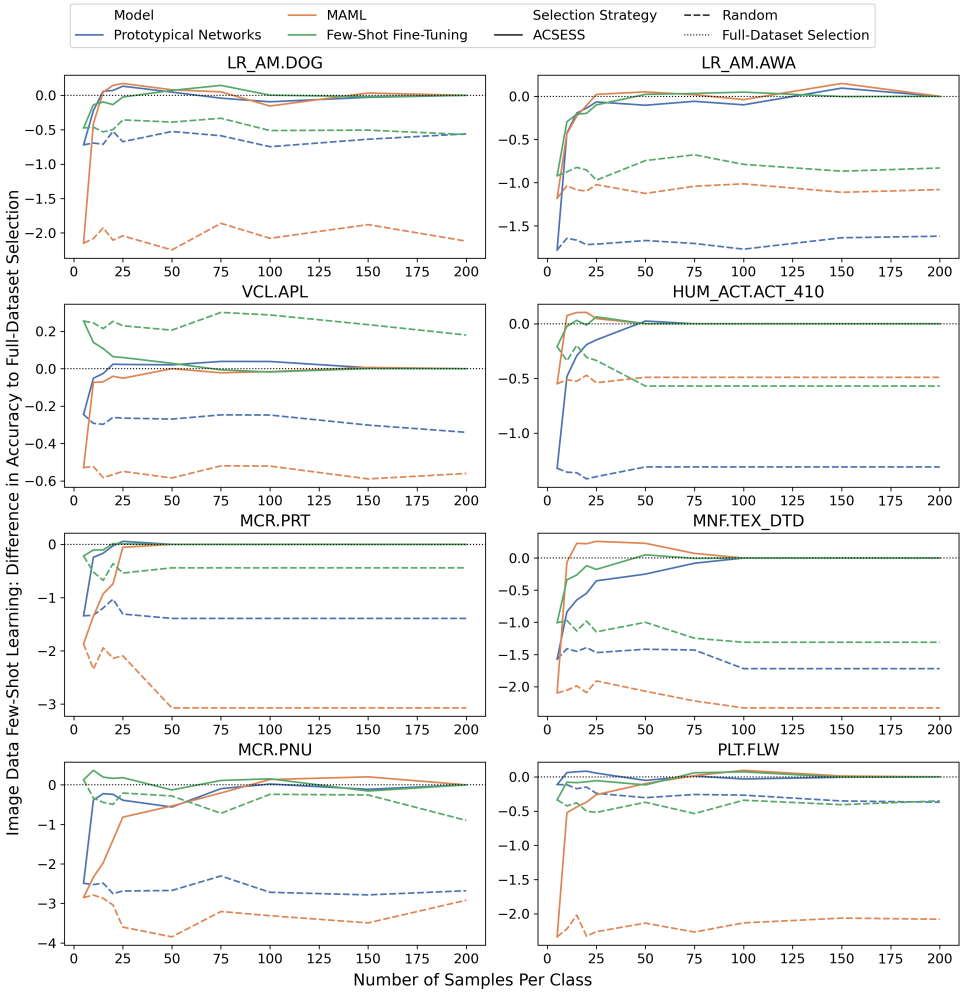

Figure 14: The difference in performance of sample selection strategies on dataset subsets of different sizes for the image datasets using gradient few-shot learning. The reported difference is in comparison to using sample selection from the full dataset.