KITE: Kernelized and Information Theoretic Exemplars for In-Context Learning

Vaibhav Singh[1] **Soumya Suvra Ghosal**[2] **Kapu Nirmal Joshua**[3] **Soumyabrata Pal**[4] **Sayak Ray Chowdhury**[3]

[1]IIT Bombay; [2]UMD College Park; [3]IIT Kanpur [4]Adobe Research

arXiv:2509.15676v2 [cs.LG] 2 Jun 2026

Abstract

In-context learning (ICL) has emerged as a powerful paradigm for adapting large language models (LLMs) to new, data-scarce tasks using only a few carefully selected task-specific examples embedded in the prompt. However, given the limited context size of LLMs, a fundamental question arises: Which examples should be selected to maximize performance on a given user query? While nearest-neighbor-based methods have been widely adopted for this purpose, they suffer from well-known drawbacks in high-dimensional embedding spaces, including poor generalization and limited diversity. In this work, we study example selection in ICL from a principled, information-theoretic perspective. We frame the example selection task as a query-specific high-dimensional optimization problem: selecting a subset of exemplars from a larger example bank to minimize prediction error for a given query. To solve this, we first derive a principled surrogate objective that is approximately submodular, enabling the use of a greedy algorithm with an approximation guarantee. We further enhance our method by introducing an optimal design-based regularizer to encourage diversity in the selected examples. Empirically, we demonstrate significant improvements over standard retrieval methods across a suite of classification tasks, highlighting the benefits of structure-aware, diverse example selection for ICL in real-world, label-scarce scenarios.

1 Introduction

With the advent of highly capable large language models (LLMs) (Adiwardana et al., 2020; Wang et al., 2019; Zhang et al., 2021; Wang et al., 2022a), in-context learning (ICL) (Rubin et al., 2022; Liu et al., 2022; Wu et al., 2022) via prompt optimization has emerged as a powerful technique for generating responses to complex user queries in data-scarce settings. In this popular and practical paradigm, we assume access to a small bank of high-quality task-specific examples. For a given user query, a few unique and relevant demonstrations are selected to form additional context for the language model. Since LLMs are pre-trained on large corpora, even a small number of carefully chosen exemplars can often suffice to guide the model toward accurate, task-consistent responses (Luo et al., 2024). Given the limited context window of large language models, a natural and important question arises for ICL: *"Given a specific user query, how can we optimally select and order a subset of task-specific examples from an associated example bank to include in the prompt to maximize performance?"* While several existing methods address this question empirically (Dong et al., 2022; Luo et al., 2024), we take a principled, information-theoretic approach.

Motivated by data-scarce settings and the need for task generalization, we focus on common and practical scenarios where the exemplar retriever is frozen and non-trainable. Among unsupervised retrieval methods, the most widely used is KATE (Liu et al., 2021), which pioneered a nearest neighbor (kNN)-based strategy for selecting in-context examples. Such methods retrieve examples most similar to the query using pretrained embeddings. However, as with classical kNN methods, similarity-based retrieval in high-dimensional embedding spaces inherits its well-known limitations – pairwise distances can become less discriminative and more sensitive to noise (Beyer et al., 1999). As a result, nearest neighbors may not reliably correspond to the most informative examples for prediction. In machine learning, this problem is often addressed by assuming a structured hypothesis class (e.g., linear models), thereby reducing model complexity and improving generalization guarantees with fewer samples. This motivates our central theoretical question in the ICL problem: *"Can we design an example selection algorithm—i.e., a retriever—that, given a user query, operates under a structured modeling assumption*

to improve selection quality?" By incorporating modeling structure into retrieval, we move beyond purely distance-based heuristics and instead select examples that are informative for prediction as well as promote diversity among the selected set.

From a theoretical standpoint, we analyze exemplar selection within a predictive framework in a reproducing kernel Hilbert space (RKHS).[1] Under this formulation, we approximate the model responses by a linear function in a high-dimensional (possibly infinite and implicit) feature representation of inputs. The goal of ICL reduces to selecting a small subset of datapoints that minimizes the prediction error for a test query: *"Given a test point* $\mathbf{z} \in \mathbb{R}^d$ *and an example bank* $(\mathbf{x}_i, y_i)_{i=1}^n \in \mathbb{R}^d \times \mathbb{R}$*, which* r *examples should be selected to train a model (predictor) in an RKHS such that the prediction error at* z *is minimized?"* The resulting selection rule naturally induces a retriever for ICL demonstrations in LLMs. Notably, this departs from standard learning-theoretic goals, which aim to generalize across a distribution of test points. In contrast, we focus on query-specific generalization, i.e., training a model that performs well on a single (and potentially arbitrary) test query using a carefully selected subset of training examples.

To this end, we derive a surrogate loss function over sets that quantifies query-specific prediction error. The negated loss function exhibits two *nice* properties - monotonicity and (approximate) submodularity. This key structural insight allows us to apply a greedy selection algorithm with a provable $1-e^{-\gamma}$ approximation guarantee relative to the optimal subset, where $\gamma \in (0, 1)$ is the approximate submodularity ratio of the negated loss with respect to the example bank. In addition to this strategy that selects examples to minimize prediction error, we introduce a mechanism to enforce diversity among the selected examples. Inspired by maximum information gain theory – a well-established framework in experiment design (Lattimore and Szepesvári, 2020) – we consider the scenario where the goal is to select a subset of examples such that the resulting trained model performs well across all queries in the example bank. We incorporate this design objective as a regularizer into our selection criterion to encourage diversity among selected examples, which improves the model's generalizability and enhances the quality of LLM responses.

Building on this formulation, we design a *computationally efficient and fully unsupervised* example selection algorithm, KITE. We focus on the setting where the *retriever is frozen and training-free*. In many real-world deployments, labeled data is scarce, models are accessed via APIs, and retraining/fine-tuning a retriever is either costly or infeasible. Classic top-r similarity-based retrieval methods are widely used precisely because they are model-agnostic, task-agnostic, and deployable off-the-shelf. Our work falls under this setting, and our goal is to provide a *principled theoretical foundation* for exemplar selection that retains these practical advantages while improving performance.[2]

We show that KITE preserves the simplicity and negligible overhead of top-r retrieval while delivering consistent gains across model families, dataset characteristics, and embedding encoders. On five classification benchmarks with three open-source LLMs (GPT-Neo-2.7B, Qwen-2.5-1.5B, Llama-3.2-3B), KITE attains the highest accuracy in 13 of 15 settings, outperforming kNN-based retrieval (Liu et al., 2021), the DPP-based CEIL retriever (Ye et al., 2023a). These gains extend to four frontier LLMs (GPT-4o-mini, GPT-4.1, Claude-Haiku-4.5, GPT-5.2) across seven datasets, where KITE achieves the best average on every model, and they remain robust across multiple encoders. Beyond classification tasks (which is our theoretical focus), KITE achieves the highest average accuracy across four open-domain QA and reasoning benchmarks on all four frontier models, with gains of $+0.2$ to $+1.8$ points over the best KNN retriever, demonstrating KITE's generalization.

1.1 Related Work

In-Context Learning (ICL) introduced by Brown et al. (2020b) is a powerful paradigm in which LLMs learn new tasks by conditioning on a few input-output examples provided in the prompt, without any parameter updates. The underlying mechanisms of ICL have been extensively studied; some works suggest that ICL allows the model to infer a shared latent concept from demonstrations (Xie et al., 2021). Further research indicates that ICL

[1] We do not assume that LLMs are kernel machines; rather, the model serves as a theoretical framework from which we derive a learning-theoretic objective with guarantees and design an algorithm that transfers effectively to practical ICL settings. Many commonly used kernels (e.g. Gaussian RBF, Matern) are universal and their associated RKHS is dense in the space of continuous functions on compact domains (Micchelli et al., 2006). Hence, RKHS elements are highly expressive and can approximate a wide range of nonlinear functions.

[2] When training is permissible, our framework can naturally incorporate learned embeddings.

is a nuanced process, as models do not always rely strictly on the provided input-output mappings (Min et al., 2022). A broad range of studies has also investigated the factors that influence ICL performance, such as demonstration calibration and corpora effects (Zhao et al., 2021; Shin et al., 2022; Wei et al., 2022; Yoo et al., 2022; Wei et al., 2023).

Example retrieval for few-shot learning. The effectiveness of ICL is highly sensitive to the choice of examples, leading to extensive work on example selection (Brown et al., 2020b; Min et al., 2022). Prior methods can be broadly categorized into training-free and training-based retrieval strategies.

Training-free (unsupervised) retrieval methods select examples at inference time without updating retriever parameters. Classic approaches use semantic similarity (e.g., nearest neighbors in embedding space) to choose top-r exemplars (Liu et al., 2022). Diversity-enhancing subset selection strategies, most notably CEIL (Ye et al., 2023a) – which uses a determinantal point process (DPP) score over a kernel similarity matrix to encourage diverse exemplar selection – also fall into this category. KITE also operates in this regime but differs in that its selection objective is derived from a principled kernelized predictive framework with provable approximation guarantees.

In contrast, training-based methods learn a retriever to select effective demonstrations. These approaches often fine-tune dense retrievers using labels distilled from a language model (Rubin et al., 2022), train neural networks to predict ICL performance (Wu et al., 2024), or employ contrastive objectives to balance relevance and coverage (Li et al., 2023; Nguyen and Wong, 2023; Luo et al., 2023; Wang et al., 2023; Ye et al., 2023b; Ghosal et al., 2025b,a). Such methods improve task adaptation but require labeled data and retriever training, which introduce additional deployment complexity.

2 Problem Formulation

Consider access to a dataset $\mathcal{X}$ of n known input–output pairs $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)$, where inputs are d-dimensional vectors and outputs are real scalars. Given a test query $\mathbf{z} \in \mathbb{R}^d$ and $r \ll n$, the in-context learning (ICL) problem is to select an optimal subset of r examples from $\mathcal{X}$ as context for the LLM's prediction on $\mathbf{z}$, so as to minimize prediction error. The selected subset must contain unique elements, since duplicating examples does not provide additional context in LLMs.

We assume that, conditioned on $\mathbf{z}$, the model response y for any input $\mathbf{x}$ is a noisy evaluation of an unknown function h from a function class $\mathcal{H}$, i.e., $y = h(\mathbf{x}) + \eta$, where η is zero-mean sub-Gaussian noise. Although LLM outputs are token sequences, we treat y as a scalar task-relevant signal extracted from the output — for classification tasks, y is the likelihood of the correct label given $\mathbf{x}$. This scalar regression view is standard in the ICL theory literature (Xie et al., 2021; Garg et al., 2022), and the noise term absorbs stochasticity from sampling, prompt-order effects, and tokenizer-level variability. The function h is defined with respect to the test query $\mathbf{z}$, and the selected subset $\mathcal{S}$ of r examples is used to train the model to predict $\mathbf{z}$.

We focus on ICL tasks with single-valued outputs, such as classification and regression. In these settings, the ICL problem reduces to: *for a given test query $\mathbf{z}$, which subset $\mathcal{S}$ of size at most r from $\mathcal{X}$ should we select so that a predictor $\widehat{h}_{\mathcal{S}} \in \mathcal{H}$ fitted on $\mathcal{S}$ minimizes the expected error at $\mathbf{z}$?* Formally,

$$\underset{\mathcal{S}\subseteq\mathcal{X}\,:\,|\mathcal{S}|\leq r}{\arg\min} \left|h(\mathbf{z}) - \widehat{h}_{\mathcal{S}}(\mathbf{z})\right| . \tag{1}$$

This discrete problem is computationally challenging, with complexity exponential in r and polynomial in n due to the $\binom{n}{r}$ possible subsets. Since this runtime would be incurred per query, and fast inference without sacrificing accuracy is paramount, our goal is an efficient algorithm that finds high-quality solutions to equation 1. The resulting selection rule can be used directly for ICL by providing the chosen examples as demonstrations; the selected subset thus acts as a proxy for training a predictor in $\mathcal{H}$.

To make the problem tractable, we assume $\mathcal{H}$ is a reproducing kernel Hilbert space (RKHS, see Schölkopf and Smola (2002)). An RKHS is fully characterized by a symmetric, positive semi-definite kernel function $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, and vice versa. Three commonly used kernels are linear: $k_{\text{lin}}(\mathbf{x}, \mathbf{x}') = \mathbf{x}^\top \mathbf{x}'$, polynomial with degree m: $k_{\text{Poly}}(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^\top \mathbf{x}' + c)^m$ and Gaussian/RBF with length-scale σ: $k_{\text{Gauss}}(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\sigma^2}\right)$. The kernel trick gives a feature map $\phi : \mathbb{R}^d \to \mathcal{H}$ with $k(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^\top \phi(\mathbf{x}')$, where the inner product is that of $\mathcal{H}$. By the *reproducing property*, $h(x) = \phi(\mathbf{x})^\top h$ for all $h \in \mathcal{H}$ and $x \in \mathbb{R}^d$. Together, these let us perform all computations in the high-dimensional (possibly infinite) Hilbert space $\mathcal{H}$ analogously to Euclidean space, without explicitly computing inner products.

3 Algorithm Design

We develop a principled two-stage approach to solve the in-context learning problem: reformulate equation 1 as a submodular optimization problem and a greedy algorithm to solve it.

3.1 Problem Reformulation

We first invoke the Representer theorem to find a predictor $\widehat{h}_{\mathcal{S}} \in \mathcal{H}$. Consider a fixed, known, positive semi-definite kernel k and the associated (implicit) feature map ϕ. By the reproducing property, the kernel ridge regressor with a regularizer $\beta > 0$ takes the form (Schölkopf and Smola, 2002)

$$\widehat{h}_{\mathcal{S}}(\mathbf{z}) = \phi(\mathbf{z})^\top \mathbf{V}_{\mathcal{S}}^{-1} \sum\nolimits_{i \in \mathcal{S}} \phi(\mathbf{x}_i)\, y_i \ ,$$

where $\mathbf{V}_{\mathcal{S}} = \sum_{i \in \mathcal{S}} \phi(\mathbf{x}_i)\phi(\mathbf{x}_i)^\top + \beta \mathbf{I}$ denotes the β-regularized design covariance matrix and $\mathbf{I}$ denotes the identity matrix (operator) in $\mathcal{H}$. Now, applying Chernoff bound for sub-Gaussian random variables, we can bound, for any $\delta \in (0, 1)$, the prediction error for the test query $\mathbf{z}$ as

$$\left| h(\mathbf{z}) - \widehat{h}_{\mathcal{S}}(\mathbf{z}) \right| \lessapprox \sqrt{\phi(\mathbf{z})^\top \mathbf{V}_{\mathcal{S}}^{-1} \phi(\mathbf{z}) \log(1/\delta)}$$

with probability $\geq 1 - \delta$, where a noise-variance dependent factor is hidden under $\lessapprox$. This is a well-known result on the prediction error in RKHSs (Lattimore and Szepesvári, 2020, Chapter 20). It shows that the upper bound on prediction error is related to the chosen subset $\mathcal{S}$ via the term $\phi(\mathbf{z})^\top \mathbf{V}_{\mathcal{S}}^{-1} \phi(\mathbf{z})$. Hence, we need to minimize this for a fixed $\mathbf{z} \in \mathbb{R}^d$, or, equivalently solve the optimization problem:

$$\underset{\mathcal{S} \subseteq \mathcal{X},\, |\mathcal{S}| \leq r}{\arg\max}\ f_{\mathbf{z}}(\mathcal{S}),\ f_{\mathbf{z}}(\mathcal{S}) = -\phi(\mathbf{z})^\top \mathbf{V}_{\mathcal{S}}^{-1} \phi(\mathbf{z}). \quad (2)$$

Note that we can write the set function $f_{\mathbf{z}}$ in terms of the kernel k, so that it can be implemented solely with kernel computations, avoiding matrix inverses and matrix-vector multiplications in (possibly) infinite-dimensional RKHSs. We start with a few notations - for a set $\mathcal{S} \equiv \{\mathbf{x}_{i_1}, \ldots, \mathbf{x}_{i_{|\mathcal{S}|}}\}$, define a matrix $\Phi_{\mathcal{S}}$ by stacking the feature vectors $\phi(\mathbf{x}_{i_1}), \ldots, \phi(\mathbf{x}_{i_{|\mathcal{S}|}})$ row-wise.

Next, define the kernel vector between an input x and the set $\mathcal{S}$ as $\mathbf{k}_{\mathcal{S}}(x) = \Phi_{\mathcal{S}} \phi(\mathbf{x})$.

The corresponding kernel (Gram) matrix admits $\mathbf{K}_{\mathcal{S}} = \Phi_{\mathcal{S}} \Phi_{\mathcal{S}}^\top$ and the regularized design covariance matrix becomes $\mathbf{V}_{\mathcal{S}} = \Phi_{\mathcal{S}}^\top \Phi_{\mathcal{S}} + \beta \mathbf{I}$. Then, using Woodbury identity together with the kernel trick, we obtain

$$f_{\mathbf{z}}(\mathcal{S}) = -\frac{1}{\beta} \cdot k_{\mathcal{S}}(\mathbf{z}, \mathbf{z}) \ , \text{where} \quad (3)$$

$$k_{\mathcal{S}}(\mathbf{x}, \mathbf{y}) = k(\mathbf{x}, \mathbf{y}) - \mathbf{k}_{\mathcal{S}}(\mathbf{x})^\top (\mathbf{K}_{\mathcal{S}} + \beta \mathbf{I}_{|\mathcal{S}|})^{-1} \mathbf{k}_{\mathcal{S}}(\mathbf{y})$$

denotes the similarity between inputs $\mathbf{x}$ and $\mathbf{y}$ conditioned on the set $\mathcal{S}$ (derivation in appendix).

The set function $f_{\mathbf{z}}$ is monotonically increasing, since for any $\mathcal{S} \subseteq \mathcal{L}$, the design covariance matrices satisfy $\mathbf{V}_{\mathcal{S}} \preceq \mathbf{V}_{\mathcal{L}}$ in the Lowener order. Along with monotonicity, a desirable property for maximizing set functions is submodularity, defined below.

Definition 1 (Submodular function)**.** *A set function $f : 2^{\mathcal{X}} \to \mathbb{R}$ is called* submodular *if for all $\mathcal{S} \subseteq \mathcal{L} \subseteq \mathcal{X}$ and $\mathbf{x} \in \mathcal{X} \setminus \mathcal{L}$, it satisfies*

$$f(\mathcal{S} \cup \{\mathbf{x}\}) - f(\mathcal{S}) \geq f(\mathcal{L} \cup \{\mathbf{x}\}) - f(\mathcal{L}) \ .$$

For monotone and submodular functions, a greedy algorithm admits an $(1 - 1/e)$-factor approximation. However, $f_{\mathbf{z}}$ isn't submodular. Instead, it exhibits approximate submodularity and retains the near-optimality guarantee of greedy algorithms. This approximation is captured by the *submodularity ratio* (Das and Kempe, 2011), defined below.

Definition 2 (Submodularity ratio)**.** *For a ground set $\mathcal{X}$ and a parameter $r \in \mathbb{N}$, the submodularity ratio of a set function $f : 2^{\mathcal{X}} \to \mathbb{R}$ is defined as*

$$\gamma_r(f) = \min_{\mathcal{S}, \mathcal{L} \subseteq \mathcal{X} : |\mathcal{L}| \leq r,\ \mathcal{L} \cap \mathcal{S} = \emptyset} \frac{\sum_{\mathbf{x} \in \mathcal{L}} \left(f(\mathcal{S} \cup \{\mathbf{x}\}) - f(\mathcal{S}) \right)}{f(\mathcal{S} \cup \mathcal{L}) - f(\mathcal{S})}.$$

The submodularity ratio captures how much the value of f can increase by adding any subset $\mathcal{L}$ of size at most r to $\mathcal{S}$, compared to the combined benefits of adding its elements to $\mathcal{S}$. It quantifies the degree to which f satisfies the diminishing returns property, and hence how well greedy algorithms perform in maximizing f under cardinality constraints. If f is submodular, then $\gamma_r = 1$. When $\gamma_r < 1$, the function is approximately submodular.

Lemma 1 (Submodularity ratio of $f_{\mathbf{z}}$)**.** *For any $\mathbf{z} \in \mathbb{R}^d$ and $\beta > 0$, $\gamma_r(f_{\mathbf{z}}) \geq \frac{1}{1 + (r-1)\mu}$, where $\mu = \max_{\mathbf{x}_i, \mathbf{x}_j \notin \mathcal{S}} \frac{|k_{\mathcal{S}}(\mathbf{x}_i, \mathbf{x}_j)|}{\sqrt{\beta + k_{\mathcal{S}}(\mathbf{x}_i, \mathbf{x}_i)} \sqrt{\beta + k_{\mathcal{S}}(\mathbf{x}_j, \mathbf{x}_j)}}$ is the maximum coherence between any pair in the set $\mathcal{X} \setminus \mathcal{S}$.*

3.2 Greedy Algorithm

Selecting relevant examples. For any $\mathcal{S}' \subset \mathcal{S}$ such that $\mathcal{S} \setminus \mathcal{S}' = \{\mathbf{x}\}$, using Sherman-Morrison

Algorithm 1 KERNELIZED INFORMATION THEORETIC EXEMPLARS (KITE)

Require: Dataset $\mathcal{X} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n \in \mathbb{R}^d \times \mathbb{R}$; kernel function k; test query $\mathbf{z}$; subset size r; regularization $\beta > 0$; diversity weight $\lambda > 0$

Initialize available datapoints $\mathbf{mask} \leftarrow \mathbf{1}_n$, selected set $\mathcal{S} \leftarrow \emptyset$, kernel $k_{\mathcal{S}}(\mathbf{z}, \mathbf{x}) \leftarrow k(\mathbf{z}, \mathbf{x})$
for $i = 1$ to r **do**
  $\mathcal{X}_{\text{available}} \leftarrow \{\mathbf{x}_j : \mathbf{mask}_j = 1\}$
  for each $\mathbf{x} \in \mathcal{X}_{\text{available}}$ **do**
    $\text{rel}(\mathbf{x}) \leftarrow \frac{(k_{\mathcal{S}}(\mathbf{z},\mathbf{x}))^2}{\beta + k_{\mathcal{S}}(\mathbf{x},\mathbf{x})}$, where $k_{\mathcal{S}}(\mathbf{z}, \mathbf{x}) = k(\mathbf{z}, \mathbf{x}) - \mathbf{k}_{\mathcal{S}}(\mathbf{z})^\top (\mathbf{K}_{\mathcal{S}} + \beta \mathbf{I}_{|\mathcal{S}|})^{-1} \mathbf{k}_{\mathcal{S}}(\mathbf{x})$
    $\text{div}(\mathbf{x}) \leftarrow \log\left(\beta + k_{\mathcal{S}}(\mathbf{x}, \mathbf{x})\right)$
    $\text{score}(\mathbf{x}) \leftarrow \text{rel}(\mathbf{x}) + \lambda \cdot \text{div}(\mathbf{x})$
  $\mathbf{x}^* \leftarrow \text{argmax}_{\mathbf{x} \in \mathcal{X}_{\text{available}}} \text{score}(\mathbf{x})$
  $\mathcal{S} \leftarrow \mathcal{S} \cup \{\mathbf{x}^*\}$, update $\mathbf{mask}$ to exclude $\mathbf{x}^*$
return $\mathcal{S}$

formula and kernel trick (equation 3), we get

$$f_{\mathbf{z}}(\mathcal{S}) = f_{\mathbf{z}}(\mathcal{S}') + \frac{1}{\beta} \cdot \frac{(k_{\mathcal{S}'}(\mathbf{z}, \mathbf{x}))^2}{\beta + k_{\mathcal{S}'}(\mathbf{x}, \mathbf{x})} .$$

This naturally yields a greedy algorithm. Starting with an empty set $\mathcal{S}_0 = \emptyset$, at each step $i \in \{1, 2, \ldots, r\}$, select the example that provides the maximum marginal gain over already chosen examples $\mathcal{S}_{i-1} = \{\mathbf{x}_1, \ldots, \mathbf{x}_{i-1}\}$, i.e., select

$$\mathbf{x}_i = \underset{\mathbf{x} \in \mathcal{X} \setminus \mathcal{S}_{i-1}}{\text{argmax}} \frac{\left(k_{\mathcal{S}_{i-1}}(\mathbf{z}, \mathbf{x})\right)^2}{\beta + k_{\mathcal{S}_{i-1}}(\mathbf{x}, \mathbf{x})} . \tag{4}$$

Adapting from Das and Kempe (2011), we obtain the following guarantee for the greedy algorithm.

Theorem 1 (Greedy algorithm performance)**.** *For any $\mathbf{z} \in \mathbb{R}^d$, let the greedy algorithm return a set $\hat{\mathcal{S}}$ for the optimization problem in equation 2. Then $f_{\mathbf{z}}(\hat{\mathcal{S}}) \geq \left(1 - e^{-\gamma_r(f_{\mathbf{z}})}\right) \cdot f(\mathcal{S}^*)$, where S^* is an optimal solution of size at most r.*

The approximation guarantee degrades gracefully with the submodularity ratio $\gamma_r(f_{\mathbf{z}})$. When $\gamma_r(f_{\mathbf{z}}) \approx 1$, the greedy algorithm performs nearly as well as the optimal algorithm (up to an $(1 - 1/e)$ factor). Increasing the regularization strength β drives $\gamma_r(f_{\mathbf{z}}) \to 1$. In practice, γ can be estimated, providing useful certificates of near-optimality for greedy selection. Empirical verification on the datasets used in this work (Appendix B, Table 8) shows that the realized submodularity ratio is typically in the 0.7–0.9 range and approaches 1 as r or β grow; the lowest values are observed at small r on HellaSwag (down to $\sim$0.4 at r=5), where greedy selection still performs well empirically (Table 1).

Selecting diverse examples. Equation 4 seeks the example $\mathbf{x}_i$ which is most relevant to the input query $\mathbf{z}$ in the geometry induced by the kernel function $k_{\mathcal{S}_{i-1}}(\cdot, \cdot)$. However, for ICL, both relevance (i.e., choosing examples similar to the input) and diversity (i.e., choosing diverse examples) are essential, and need to be carefully balanced. To select diverse examples, we seek the set $\mathcal{S}$ that fetches the maximum information about the unknown function $h \in \mathcal{H}$ from noisy responses. If we assume h to be sampled from a Gaussian process[3] with covariance function k, then the information gain about h from r noisy responses is given by $g(\mathcal{S}) = \log\det\left(\mathbf{K}_{\mathcal{S}} + \beta \mathbf{I}_r\right)$, where $\mathbf{K}_{\mathcal{S}}$ is the Gram matrix over the set $\mathcal{S}$ (Cover, 1999).

The set function g is known as the D-optimal design (Pukelsheim, 2006) and the most informative (i.e., diverse) set is the one that maximizes g.

Lemma 2 (Submodularity)**.** *For any $\beta > 0$ and $r \in \mathbb{N}$, $g(\mathcal{S}) = \log\det(\mathbf{K}_{\mathcal{S}} + \beta \mathbf{I}_r)$ is monotone, submodular, and admits the greedy selection*

$$\mathbf{x}_i = \underset{\mathbf{x} \in \mathcal{X} \setminus \mathcal{S}_{i-1}}{\text{argmax}} \log\left(\beta + k_{\mathcal{S}_{i-1}}(\mathbf{x}, \mathbf{x})\right) . \tag{5}$$

Combining relevance and diversity. To select relevant as well as diverse examples (set-conditioned), we combine equation 4 and equation 5 to obtain

$$\mathbf{x}_i = \underset{\mathbf{x} \in \mathcal{X} \setminus \mathcal{S}_{i-1}}{\text{argmax}} \frac{\left(k_{\mathcal{S}_{i-1}}(\mathbf{z}, \mathbf{x})\right)^2}{\beta + k_{\mathcal{S}_{i-1}}(\mathbf{x}, \mathbf{x})} + \lambda \log\left(\beta + k_{\mathcal{S}_{i-1}}(\mathbf{x}, \mathbf{x})\right) , \tag{6}$$

where $\lambda \geq 0$ controls the relevance-diversity trade-off. The combined objective remains monotone

[3] A Gaussian process (GP) over $\mathbb{R}^d$, is a collection of random variables $(h(x))_{x \in \mathbb{R}^d}$, such that every finite sub-collection $(h(x_i))_{i=1}^m$ is jointly Gaussian (Williams and Rasmussen, 2006). The GP model only aids in designing the diversity objective, and has nothing to do with the actual function h, which is a fixed, unknown, member of the RKHS $\mathcal{H}$.

and approximately submodular, so the set $\hat{\mathcal{S}}$ obtained from equation 6 achieves a guarantee similar to Theorem 1. We call this algorithm **K**ernelized **I**nformation **T**heoretic **E**xemplars (KITE) for ICL; pseudo-code is in Algorithm 1. In our experiments, the kernel is a tunable hyperparameter of KITE.

4 Experiments

We present a comprehensive empirical analysis of KITE along multiple axes: (i) base accuracy on five classification benchmarks across three open-source LLMs (Table 1); (ii) scaling to 7B/8B variants (Table 2); (iii) generalization to four frontier API-only LLMs (GPT-4o-mini, GPT-4.1, Claude-Haiku-4.5, GPT-5.2) on seven classification datasets (Tables 3, 6); (iv) kernel ablations on both open-source and frontier settings (Tables 4, 17); (v) robustness across multiple encoders (Tables 5, 14); (vi) sensitivity to the exemplar budget r (Tables 11, 21; Fig. 2); (vii) empirical verification of the submodularity property (App. B, Table 8); and (viii) wall-clock runtime analysis showing negligible overhead vs. standard retrieval (App. G).

Implementation details. For experiments with open-source LLM, we use $\beta = 0.02, \lambda = 0.5$, and `bert-base-uncased` (Devlin et al., 2019) embeddings for retrieval. Our main open-source experiments (Table 1) use three LLMs: GPT-Neo-2.7B (Black et al., 2021), Qwen-2.5-1.5B (Hui et al., 2024), and Llama-3.2-3B (Grattafiori et al., 2024), with r=50 in-context examples (truncated to fit the context window) and three kernels: Linear, Polynomial (degree m=3), and Gaussian/RBF (σ=1.0).

For experiments with Frontier-LLM, we use r=5, 10 (matching realistic API usage) and an expanded kernel suite (Linear, RBF, Laplacian, Matérn-3/2, Rational Quadratic) - Tables 6, 12, 13 and 15. We use two distinct encoders - OpenAI's `text-embedding-3-large` and NVIDIA's `NV-Embed-v2` embeddings of 3072 and 4096 dimensions respectively. For each frontier experiment we use 500 inference queries and a candidate bank of 2000 examples (first from val/train splits respectively). In Tables 3 and 6, KITE is reported with λ fixed at 0.5 and the best accuracy across the five kernels; per-kernel and per-λ breakdowns appear in Tables 4, 17, and App. E.5. For reproducibility, we follow a uniform protocol: embeddings are used directly from the pretrained encoder with no centering, whitening, or per-dataset re-scaling; the kernel hyperparameters above (σ, m, polynomial constant c=1, Laplacian/Matérn/RQ length-scales =1) and the model hyperparameters β and λ are *fixed across all datasets and models*. The same fixed values are used for all baselines wherever they admit a kernel parameter, ensuring the comparison is not biased toward KITE.

Datasets. A detailed description of the datasets is given in Section D. For the open-source experiments, to empirically demonstrate the efficacy of KITE, we evaluate on five few-shot classification datasets: SST-2 & 5 (Socher et al., 2013) (single-sentence sentiment), CMSQA (Talmor et al., 2019) (question answering), MRPC (Dolan et al., 2004) (sentence-pair paraphrase), QNLI (Wang et al., 2018) (binary entailment classification), and HellaSwag (Zellers et al., 2019) (common sense NLI). For each task, we use the corresponding validation split for evaluation and employ the deduplicated training split as the candidate exemplar set.

Evaluation metrics. Following Brown et al. (2020a), we formulate all classification tasks as multiple-choice problems. Input prompts concatenate the context with each candidate label. We compute the conditional log-likelihood of the label tokens given the prompt, and select the candidate with the highest log-likelihood as the prediction. Performance is evaluated using accuracy on the validation split. For all methods, we use the retriever's own ranking as the prompt order: kNN uses descending similarity, CEIL uses its greedy DPP-selection order, and KITE uses its greedy acquisition order. This ensures that observed differences are attributable to *which* exemplars are selected rather than *how* they are ordered. For frontier models accessed via API, logits are unavailable; we instead prompt each model to generate the answer directly and report accuracy as exact match of the generated token to the gold label.

Baselines. We compare KITE against Random, BM25 (Robertson et al., 2009), top-r with Dense embeddings (Liu et al., 2021), and the training-free version of CEIL (Ye et al., 2023a), a DPP-based retrieval strategy.

Results. We report evaluation results in Table 1. We find that KITE consistently outperforms all baselines across most datasets and model architectures. On GPT-Neo-2.7B, for example, KITE delivers an accuracy improvement of +4.55% on SST-5 (Socher et al., 2013) and +3.69% on CM-

Method	SST-2			SST-5			CMSQA			MRPC			QNLI			HellaSwag		
	GN	QW	LL	GN	QW	LL	GN	QW	LL	GN	QW	LL	GN	QW	LL	GN	QW	LL
Random	63.36	86.32	89.84	32.31	40.53	42.41	42.25	65.90	67.22	66.38	70.45	71.12	57.56	70.21	68.43	41.75	66.30	65.21
BM25	67.78	90.14	91.63	36.05	47.86	**47.85**	42.91	70.02	70.92	66.96	73.77	70.58	62.12	**71.11**	69.38	41.34	67.85	66.32
Dense	72.94	88.99	92.55	35.96	46.64	44.14	43.98	70.18	72.23	65.83	73.92	71.81	64.42	70.73	70.04	40.44	68.80	67.88
CEIL	71.22	90.25	91.86	36.84	47.88	46.45	37.53	70.05	71.65	68.04	74.25	70.63	63.62	70.53	70.08	40.56	69.32	68.55
KITE	**74.41**	**93.35**	**94.28**	**40.60**	**49.59**	47.59	**46.60**	**71.25**	**72.89**	**68.38**	**75.27**	**71.98**	**65.32**	71.05	**71.68**	**41.68**	**71.02**	**70.12**

Table 1: **Evaluation Results.** We compare classification accuracy (%) of KITE against retrieval baselines on six few-shot benchmarks. Model abbreviations: GN (GPT-Neo 2.7B), QW (Qwen 2.5–1.5B), LL (Llama 3.2–3B).

Method	SST-2				CMSQA			
	QW-1.5B	QW-7B	LL-3B	LL-8B	QW-1.5B	QW-7B	LL-3B	LL-8B
Zero-shot	61.69	65.82	68.73	74.21	57.00	64.53	58.86	61.52
Random	63.36	71.53	89.84	90.12	65.90	68.29	67.22	68.95
Dense	88.99	89.45	92.55	93.35	70.18	72.35	72.23	74.19
CEIL	90.25	91.63	91.86	93.61	70.05	74.82	71.65	73.83
KITE	**93.35**	**94.50**	**94.28**	**94.84**	**71.25**	**75.69**	**72.89**	**75.22**

Table 2: **Scaling results.** Accuracy (%) when scaling from smaller backbones to larger ones. QW denotes Qwen-2.5, and LL denotes Llama-3.2. KITE consistently outperforms zero-shot and retrieval-based baselines across model sizes.

SQA (Talmor et al., 2019) over BM25 (Robertson et al., 2009). This trend is consistently observed across all evaluated models. With Qwen-2.5-1.5B, KITE surpasses the strongest baseline, CEIL (Ye et al., 2023a), on four out of five datasets, achieving notable gains of +1.71% on SST-5 and +1.70% on HellaSwag. On Llama-3B, KITE achieves the highest accuracy on four datasets, including a +2.24% improvement on HellaSwag over CEIL (Ye et al., 2023a). Across all 15 evaluation settings (five datasets and three models), KITE attains the highest accuracy in 13 cases, underscoring its robustness and efficacy as an exemplar retrieval framework.

Computational cost. A natural concern is whether KITE's greedy selection imposes meaningful overhead at r=50. We measure end-to-end wall-clock time per query on SST-5 (8534 examples) using Qwen-2.5-1.5B/7B on a single A6000 GPU. KITE's selection adds only ≤ 0.03 s per query on the 1.5B model and ≤ 0.19 s on the 7B relative to BM25, leaving end-to-end latency effectively unchanged (App. G, Table 24).

Scaling to larger models. To study the effect of model scale, we evaluate KITE on SST-2 (Socher et al., 2013) and CMSQA (Talmor et al., 2019) using both smaller and larger variants of Qwen-2.5 and Llama-3.2. As shown in Table 2, KITE consistently outperforms all baselines across model sizes, and the gains persist as we scale to larger models such as Qwen-2.5-7B and Llama-3.2-8B.

Scaling to Frontier LLMs. To validate that KITE's gains extend beyond open-source models with accessible logits, we evaluate on four frontier LLMs accessed via API: GPT-4o-mini, GPT-4.1, Claude-Haiku-4.5, and GPT-5.2. We use OpenAI's `text-embedding-3-large` for retrieval (as the deployment-realistic choice in an API setting), with r=5, β=0.02, λ=0.5, and evaluate on seven classification datasets: SST-5, MRPC, QNLI, SWAG (Zellers et al., 2018), IMDB, DBPedia, and BoolQ, reporting the best result across five kernels. Table 3 presents the average accuracy. KITE achieves the highest average accuracy on all four models, with gains of $+0.8$ to $+1.3$ points over the best KNN retriever and $+0.7$ to $+1.2$ over CEIL. The improvements are consistent across capability tiers, from the lightweight GPT-4o-mini to the more capable GPT-5.2, showing that KITE's information-theoretic selection provides complementary benefits to model capability.

Model	BM25	KNN	CEIL	KITE
GPT-4o-mini	79.0	80.5	80.4	**81.3**
GPT-4.1	80.7	81.2	81.2	**82.2**
Claude-Haiku-4.5	81.1	82.3	82.8	**83.5**
GPT-5.2	79.7	81.6	81.7	**82.9**

Table 3: Average classification accuracy (%) on seven classification datasets with frontier LLMs (r=5, OpenAI large text embeddings). For each method, we report the best result across five kernels. KITE achieves the highest accuracy on every model.

Ablation study on kernel function. The kernel is a critical hyperparameter in KITE, as it defines the geometry of the feature space. We conduct an ablation comparing three kernels: Linear, Polynomial (degree $m = 3$), and Gaussian RBF ($\sigma = 1.0$). Table 4 shows that no single kernel is universally optimal; the best choice depends on the dataset and its underlying distribution. Table 1 reports the best perfor-

Dataset	Linear	Polynomial	Gaussian RBF
SST-5	47.95	48.38	**49.59**
CMSQA	69.86	71.06	**71.25**
MRPC	**75.27**	71.22	67.15
QNLI	70.46	70.38	**71.05**
HellaSwag	67.18	**71.02**	70.67

Table 4: **Ablation study on the kernel function.** We report accuracy for KITE using different kernels on Qwen-1.5B.

Method	BERT	RoBERTa	E5	DistilBERT	MiniLM
CEIL	47.88	41.41	45.68	48.77	48.13
Dense	46.64	36.33	45.14	49.04	48.68
KITE	**49.59**	**48.31**	**48.04**	**49.29**	**48.72**

Table 5: **Robustness to embedding encoders.** Accuracy (%) on SST-5 with Qwen-2.5-1.5B using different embedding encoders for retrieval. The headers denote `bert-base-uncased`, `roberta-base`, `e5-base-v2`, `distilbert-base-uncased`, and `all-MiniLM-L6-v2`, respectively.

mance across these kernels for each task. For a fully fair kernel-by-kernel comparison against the baselines, we extend this to five kernels (Linear, RBF, Laplacian, Matérn-3/2, Rational Quadratic) on all four frontier LLMs in App. E.4 (Table 17). Two patterns emerge: (i) KNN is largely insensitive to the kernel (since it only enters pairwise similarity), and CEIL shows moderate kernel sensitivity, whereas (ii) KITE's best kernel achieves the highest accuracy among *all* method–kernel combinations on every frontier model, and even KITE's worst-case kernel matches or exceeds the best CEIL configuration. This indicates that KITE's information-theoretic objective is robust to kernel misspecification while still benefiting from a good kernel choice.

Robustness to embedding encoders. A natural question is whether KITE's gains are sensitive to the embedding encoder. To investigate, we evaluate KITE, Dense, and CEIL on SST-5 using Qwen-2.5-1.5B with five embedding models: BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), E5 (Wang et al., 2022b), DistilBERT (Sanh et al., 2019), and MiniLM (Wang et al., 2020). As shown in Table 5, KITE achieves the highest or near-highest accuracy across all encoders. While Dense and CEIL exhibit significant variance (e.g., CEIL drops from 48.77% with DistilBERT to 41.41% with RoBERTa), KITE remains consistently strong, with a much narrower performance range. This shows that our information-theoretic selection criterion is robust to the embedding backbone, underscoring KITE's practical deployability. We further confirm this with the NV-Embed-v2 (Lee et al., 2024) encoder across all four frontier LLMs and seven classification datasets (App. E.3, Table 14): KITE attains the highest average on all models, with gains of +1.0 to +2.1 points over the best KNN retriever.

Per-dataset analysis. Table 6 provides per-dataset results for two representative models. On GPT-4.1, KITE achieves substantial gains on SWAG (+3.2 over both KNN & CEIL), MRPC (+2.4 over KNN), and QNLI (+0.8 over KNN). On Claude-Haiku-4.5, the largest gains appear on SWAG (+3.0 over KNN), QNLI (+1.6), and MRPC (+1.5). On near-saturated benchmarks (IMDB, DBPedia), performance remains tied or within noise, as expected. Across all four models and seven datasets, KITE matches or exceeds the best baseline in the majority of cases, with remaining gaps typically within 0.5% (see Appendix E.1 for full per-model breakdowns).

Extension to QA and reasoning tasks. We further evaluate KITE on four QA and reasoning benchmarks: NQ (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), HotpotQA (Yang et al., 2018), and GSM8K (Cobbe et al., 2021), using four frontier LLMs with r=5. NQ, TriviaQA, and HotpotQA are scored with Soft Exact Match (normalised substring overlap); GSM8K uses chain-of-thought prompting and numeric match on the final answer. As shown in Table 7 (Appendix F), KITE achieves the highest average accuracy on all four models, with gains of +0.2 to +1.8 over KNN.

	GPT-4.1					Claude-Haiku-4.5				
Dataset	KNN	CEIL	KITE	Δ_K	Δ_D	KNN	CEIL	KITE	Δ_K	Δ_D
SST-5	53.6	52.8	**54.2**	+0.6	+1.4	52.8	53.2	**54.0**	+1.2	+0.8
MRPC	68.9	70.1	**71.3**	+2.4	+1.2	71.8	72.8	**73.3**	+1.5	+0.5
QNLI	83.8	83.0	**84.6**	+0.8	+1.6	86.6	87.0	**88.2**	+1.6	+1.2
SWAG	71.8	71.8	**75.0**	+3.2	+3.2	75.8	76.6	**78.8**	+3.0	+2.2
IMDB	**99.4**	99.2	**99.4**	0.0	+0.2	97.6	**97.4**	97.4	+0.2	0.0
DBPedia	**99.0**	**99.4**	99.0	0.0	+ 0.4	99.0	**99.6**	**99.6**	+0.6	0.0
BoolQ	91.6	91.8	**92.2**	+0.6	+0.4	92.8	93.2	**93.0**	+0.2	+0.2
Avg	81.2	81.2	**82.2**	+1.0	+1.0	82.3	82.8	**83.5**	+1.2	+0.7

Table 6: Accuracy (%) on core benchmarks for GPT-4.1 and Claude-Haiku-4.5 (r=5, OpenAI large text embeddings). Best result in **bold**. Δ columns show KITE's gain over each baseline.

Model	BM25	KNN-best	DPP-best	KITE-best	ΔKITE–KNN
GPT-4o-mini	64.1	65.1	65.1	**65.3**	+0.2
GPT-4.1	71.2	71.5	71.7	**72.5**	+1.0
Claude-Haiku-4.5	59.1	60.3	61.0	**62.1**	+1.8
GPT-5.2	68.2	69.8	70.1	**71.0**	+1.2

Table 7: Average accuracy (%) on four QA/reasoning datasets with frontier LLMs (r=5, OpenAI large text embeddings). KITE-best = max over all kernels and λ values.

Effect of exemplar budget (r). Note that redundancy is more likely with larger r (more in-context examples). We repeat the frontier-LLM evaluation at r=10 (Appendix E.2, Tables 11–12). KITE's advantage over KNN *grows* with budget: from +1.0 at r=5 to +1.9 at r=10 on GPT-4.1, and from +1.2

to $+2.1$ on Claude-Haiku-4.5 (App. E.6, Table 21). Per-dataset, the largest $r{=}10$ gains over KNN appear on SWAG ($+3.8$ on GPT-4.1, $+3.0$ on Claude-Haiku-4.5), QNLI ($+2.4$ and $+4.2$), and SST-5 ($+4.4$ and $+3.2$). At the other end of the budget spectrum, App. A (Figure 2) shows that on SST-5 the gap between KITE and BM25/Dense/CEIL is largest in the low-shot regime ($r{=}2$–10), where careful selection matters most. These indicate that KITE is effective across the full budget range.

5 Conclusion

We introduced KITE, a query-specific greedy exemplar selection method for ICL. It selects examples that balance relevance and diversity, yielding consistent improvements over strong retrieval baselines across multiple datasets and models.

Limitations

In Section 4, we demonstrated the effectiveness of our proposed framework across multiple models and benchmarks. However, our theoretical analysis is currently grounded in classification tasks. Extending the theoretical analysis of KITE to open-ended generation tasks such as summarization or dialogue requires defining an appropriate theoretical surrogate objective, which we leave as future work.

A second consideration is computational scaling. KITE's greedy loop is $O(r \cdot n)$ per query, which is comparable to DPP-based methods and incurs negligible overhead at the bank sizes used in our experiments (up to $n \approx 8{,}500$; Appendix G). For substantially larger candidate banks (e.g., $n \gg 10^4$, as in MNLI's $\approx 400{,}000$ examples), KITE is naturally combined with an approximate-nearest-neighbor pre-filter to obtain a top-N candidate pool ($N \ll n$) before the greedy selection step. The approximation guarantee of Theorem 1 then applies with respect to this pruned ground set; characterizing the trade-off between N and the resulting approximation quality is an interesting direction for future work.

References

Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and 1 others. 2020. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*.

Kevin Beyer, Jonathan Goldstein, Raghu Ramakrishnan, and Uri Shaft. 1999. When is "nearest neighbor" meaningful? In *International conference on database theory*, pages 217–235. Springer.

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. Gpt-neo: Large scale autoregressive language modeling with mesh-tensorflow.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, and et al. Amanda Askell. 2020a. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and *et al.* 2020b. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33*, pages 1877–1901.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Thomas M Cover. 1999. *Elements of information theory*. John Wiley & Sons.

Abhimanyu Das and David Kempe. 2011. Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. *arXiv preprint arXiv:1102.3975*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *Preprint*, arXiv:1810.04805.

Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 350–356, Geneva, Switzerland. COLING.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, and 1 others. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*.

Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. 2022. What can transformers learn in-context? a case study of simple function classes. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Soumya Suvra Ghosal, Soumyabrata Pal, Koyel Mukherjee, and Dinesh Manocha. 2025a. Promptrefine: Enhancing few-shot performance on low-resource indic languages with example selection from related example banks. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 351–365.

Soumya Suvra Ghosal, Vaibhav Singh, Akash Ghosh, Soumyabrata Pal, Subhadip Baidya, Sriparna Saha, and Dinesh Manocha. 2025b. Relic: Enhancing reward model generalization for low-resource indic languages with few-shot examples. *arXiv preprint arXiv:2506.16502*.

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

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, Kai Dang, Yang Fan, Yichang Zhang, An Yang, Rui Men, Fei Huang, Bo Zheng, Yibo Miao, Shanghaoran Quan, and 5 others. 2024. Qwen2.5-coder technical report. *Preprint*, arXiv:2409.12186.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

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

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024. Nv-embed: Improved techniques for training llms as generalist embedding models. *Preprint*, arXiv:2405.17428.

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. *arXiv preprint arXiv:2305.04320*.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021. What makes good in-context examples for gpt-3?

Jing 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 the 3rd Workshop on Deep Learning for Low-Resource Natural Language Processing (DeepLo) at ACL*, pages 100–114.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

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*.

Man Luo, Xin Xu, Yue Liu, Panupong Pasupat, and Mehran Kazemi. 2024. In-context learning with retrieved demonstrations for language models: A survey. *arXiv preprint arXiv:2401.11624*.

Charles A Micchelli, Yuesheng Xu, and Haizhang Zhang. 2006. Universal kernels. *Journal of Machine Learning Research*, 7(12).

Sewon Min, Xin Lyu, Ariel Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064.

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

Friedrich Pukelsheim. 2006. *Optimal design of experiments*. SIAM.

Stephen Robertson, Hugo Zaragoza, and 1 others. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Or 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*, pages 2655–2671.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Bernhard Schölkopf and Alexander J Smola. 2002. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press.

Seongjin Shin, Sang-Woo Lee, Hwijeen Ahn, Sungdong Kim, HyoungSeok Kim, Boseop Kim, Kyunghyun Cho, Gichang Lee, Woomyoung Park, Jung-Woo Ha, and 1 others. 2022. On the effect of pretraining corpora on in-context learning by a large-scale language model. *arXiv preprint arXiv:2204.13509*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. 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 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Dingzirui Wang, Longxu Dou, and Wanxiang Che. 2022a. A survey on table-and-text hybridqa: Concepts, methods, challenges and future directions. *arXiv preprint arXiv:2212.13465*.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022b. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in neural information processing systems*, 33:5776–5788.

Xinyi Wang, Wanrong Zhu, Michael Saxon, Mark Steyvers, and William Yang Wang. 2023. Large language models are latent variable models: Explaining and finding good demonstrations for in-context learning. In *Advances in Neural Information Processing Systems 36*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, and 1 others. 2022. Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

Jerry Wei, Jason Wei, Yi Tay, Dustin Tran, Albert Webson, Yifeng Lu, Xinyun Chen, Hanxiao Liu, Da Huang, Denny Zhou, and 1 others. 2023. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846*.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.

Christopher KI Williams and Carl Edward Rasmussen. 2006. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA.

Zhaoxuan Wu, Xiaoqiang Lin, Zhongxiang Dai, Wenyang Hu, Yao Shu, See-Kiong Ng, Patrick Jaillet, and Bryan Kian Hsiang Low. 2024. Prompt optimization with ease? efficient ordering-aware automated selection of exemplars. *Advances in Neural Information Processing Systems*, 37:122706–122740.

Zhiyong Wu, Yaoxiang Wang, Jiacheng Ye, and Lingpeng Kong. 2022. Self-adaptive in-context learning: An information compression perspective for in-context example selection and ordering. *arXiv preprint arXiv:2212.10375*.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2021. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2369–2380.

Jiacheng Ye, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. 2023a. Compositional exemplars for in-context learning. In *Proceedings of the 40th International Conference on Machine Learning*. ICML.

Jiacheng Ye, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. 2023b. Compositional exemplars for in-context learning. In *International Conference on Machine Learning*, pages 39818–39833. PMLR.

Kang Min Yoo, Junyeob Kim, Hyuhng Joon Kim, Hyunsoo Cho, Hwiyeol Jo, Sang-Woo Lee, Sang-goo Lee, and Taeuk Kim. 2022. Ground-truth labels matter: A deeper look into input-label demonstrations. *arXiv preprint arXiv:2205.12685*.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. Swag: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 93–104.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Zijian Zhang, Koustav Rudra, and Avishek Anand. 2021. Explain and predict, and then predict again. In *Proceedings of the 14th ACM international conference on web search and data mining*, pages 418–426.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International conference on machine learning*, pages 12697–12706. PMLR.

A Extended Ablation Study

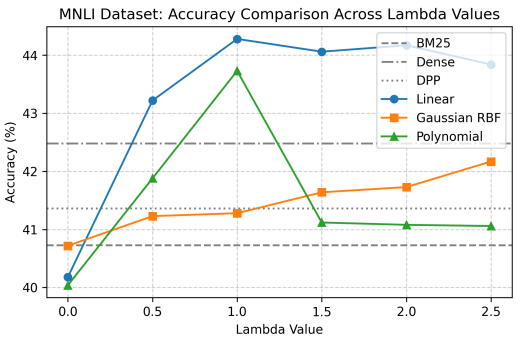


Figure 1: Ablation study on the MNLI dataset illustrating the effect of varying λ values. We report accuracy for KITE using different kernel-based selection strategies on the Qwen-1.5B model. Since MNLI has a large example bank (over 400k examples), a purely relevance-based selection ($\lambda = 0$) underperforms. Incorporating diversity through higher λ values (e.g., $\lambda = 1$) leads to notable improvements in accuracy by promoting a more diverse selection of in-context examples.

Ablation study on λ. In this section, we present an ablation study over λ on the MNLI dataset (Williams et al., 2017). As shown in Figure 1, we evaluate the impact of the trade-off parameter λ, which balances relevance and diversity in our selection process. For the MNLI dataset, which has a very large and varied example bank (over 400k examples), relying solely on relevance-based selection (λ=0) results in suboptimal performance. This is because many examples can be semantically similar, leading to a redundant in-context set. By increasing λ, we introduce diversity into the selection, which proves crucial for this task. The results indicate that a balanced approach, with λ set to approximately 1, achieves the highest accuracy. This demonstrates that for complex, large-scale datasets, a combination of both relevance and diversity is essential for selecting the most effective in-context examples.

Ablation study on number of ICE examples. We also experimented with the number of in-context examples to understand how it affects model performance. In Figure 2, we report the accuracy of the Qwen-1.5B model on the SST-5 dataset as a function of the number of provided examples. As expected, performance for all methods generally improves as more examples are added to the context. However, our proposed KITE algorithm consistently and significantly outperforms all baseline methods, including BM25, Dense retrieval, and DPP. The advantage of KITE is particularly pronounced in low-resource settings (e.g., with 2 or 4 ICE), where it establishes a substantial performance gap. This highlights KITE's superior ability to select high-quality, informative examples, making it highly effective even when the number of in-context examples is limited.

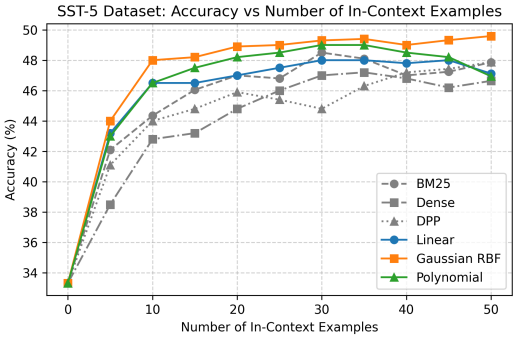


Figure 2: Accuracy comparison of the Qwen-1.5B model on the SST-5 dataset across varying numbers of in-context examples (ICE). The KITE algorithm, using kernel-based selection strategies, significantly outperforms baseline methods (BM25, Dense, DPP) in low-ICE settings.

B Empirical Validation of Submodularity

We empirically validate the submodularity of our set function, $f_z(\mathcal{S})$ by estimating its submodularity ratio (Def. 2) on real-world text embeddings. For this analysis, we used the first 500 examples from each dataset to generate embeddings. We created two distinct types: demonstration embeddings, which concatenate an input with its gold label, and query embeddings, which use the input alone.

To estimate the ratio, we run a Monte Carlo simulation across a grid of hyperparameters for subset size r and regularization β. In each trial, we sample a disjoint triplet $(\mathcal{S}, \mathcal{L}, \mathbf{z})$, where $\mathcal{S}$ is a random set of demonstration embeddings, $\mathcal{L}$ is a diverse set of up to r demonstration embeddings selected via farthest-point sampling, and $\mathbf{z}$ is a query embedding. We then compute the ratio and record the minimum value observed for each configuration ($\gamma_{\min}$). As shown in Table 8, $\gamma_{\min}$ is typically in the 0.7–0.9 range across most datasets and settings, and approaches 1 as r or β grow (right block of Table 8). The lowest values appear on HellaSwag at small r (down to 0.395 at r=5, β=1), where the candidate bank's higher embedding coherence reduces the ratio. Even in that regime, greedy selection remains empirically strong (Table 1); the theoretical guarantee in Theorem 1 degrades gracefully with γ_r, so a γ of 0.4 still yields a ≈ 0.33-factor approximation rather than no guarantee. Overall, the data support

	Vary r (fixed $\beta = 1$)							Vary β (fixed $r = 20$)					
Dataset	5	10	15	20	25	30	35	1	3	5	7	9	11
SST–5	0.748	0.685	0.705	0.689	0.703	0.716	0.696	0.689	0.744	0.719	0.758	0.792	0.827
CMSQA	0.852	0.811	0.816	0.841	0.830	0.797	0.836	0.841	0.804	0.884	0.843	0.907	0.886
MRPC	0.876	0.840	0.793	0.816	0.850	0.887	0.848	0.816	0.765	0.803	0.824	0.824	0.898
QNLI	0.844	0.801	0.900	0.893	0.875	0.837	0.877	0.893	0.829	0.823	0.919	0.912	0.882
HellaSwag	0.395	0.601	0.525	0.596	0.839	0.908	0.806	0.596	0.803	0.767	0.806	0.935	0.852

Table 8: **Analysis on Submodularity ratio $\gamma_{\min}$. Left:** We vary r while fixing $\beta = 1$; **Right:** We vary β while fixing $r = 20$.

the use of a greedy algorithm. This finding justifies our use of a greedy algorithm, as it is expected to yield near-optimal results.

C Theoretical Proofs and Results

In this section, we present the missing proofs.

C.1 Kernel Trick

The following well-known result on linear operators helps us apply the kernel trick to compute the greedy selection rule of KITE, see equation 3.

Lemma 3. *Let* $\mathbf{A}$ *be a linear operator. Then, for any* $\beta > 0$*, the following holds:*

$$(\mathbf{A}^\top\mathbf{A} + \beta\mathbf{I})^{-1}\mathbf{A}^\top = \mathbf{A}^\top(\mathbf{A}\mathbf{A}^\top + \beta\mathbf{I})^{-1},$$
$$\mathbf{I} - \mathbf{A}^\top(\mathbf{A}\mathbf{A}^\top + \beta\mathbf{I})^{-1}\mathbf{A} = \beta(\mathbf{A}^\top\mathbf{A} + \beta\mathbf{I})^{-1}.$$

Using these identities, we get

$$\begin{aligned}
&\phi(\mathbf{z})^\top\mathbf{V}_\mathcal{S}^{-1}\phi(\mathbf{z}) = \phi(\mathbf{z})^\top(\Phi_\mathcal{S}^\top\Phi_\mathcal{S} + \beta\mathbf{I})^{-1}\phi(\mathbf{z}) \\
&= \frac{1}{\beta}\Big(\phi(\mathbf{z})^\top\phi(\mathbf{z}) \\
&- \phi(\mathbf{z})^\top\Phi_\mathcal{S}^\top(\Phi_\mathcal{S}\Phi_\mathcal{S}^\top + \beta\mathbf{I}_{|\mathcal{S}|})^{-1}\Phi_\mathcal{S}\phi(\mathbf{z})\Big) \\
&= \frac{1}{\beta}\Big(k(\mathbf{z},\mathbf{z}) - \mathbf{k}_\mathcal{S}(\mathbf{z})^\top(\mathbf{K}_\mathcal{S} + \beta\mathbf{I}_{|\mathcal{S}|})^{-1}\mathbf{k}_\mathcal{S}(\mathbf{z})\Big) \\
&= \frac{1}{\beta}k_\mathcal{S}(\mathbf{z},\mathbf{z})\ .
\end{aligned}$$

C.2 Proof of Lemma 1

For a set $\mathcal{S} \subseteq \mathcal{X}$ and an element $\mathbf{x}_i \notin \mathcal{S}$, define the marginal gain of adding $\mathbf{x}_i$ to $\mathcal{S}$ in the set function $f_\mathbf{z}(\mathcal{S})$ as (the second step is obtained via an application of the Sherman Morrison formula)

$$\begin{aligned}
\Delta_i &:= f_\mathbf{z}(\mathcal{S}\cup\{\mathbf{x}_i\}) - f_\mathbf{z}(\mathcal{S}) \\
&= \frac{\big(\phi(\mathbf{z})^\top\mathbf{V}_\mathcal{S}^{-1}\phi(\mathbf{x}_i)\big)^2}{1 + \phi(\mathbf{x}_i)^\top\mathbf{V}_\mathcal{S}^{-1}\phi(\mathbf{x}_i)},
\end{aligned}$$

where $f_\mathbf{z}(\mathcal{S}) = -\phi(\mathbf{z})^\top\mathbf{V}_\mathcal{S}^{-1}\phi(\mathbf{z})$ and $\mathbf{V}_\mathcal{S} = \sum_{i\in\mathcal{S}}\phi(\mathbf{x}_i)\phi(\mathbf{x}_i)^\top + \beta\mathbf{I}$. Now consider a set $\mathcal{L} = \{\mathbf{x}_1,\dots,\mathbf{x}_r\} \subseteq \mathcal{X}\setminus\mathcal{S}$. We compare

$$\sum_{\mathbf{x}_i\in\mathcal{L}}\Delta_i, \qquad \text{vs.} \qquad f_\mathbf{z}(\mathcal{S}\cup\mathcal{L}) - f_\mathbf{z}(\mathcal{S}).$$

The ratio of these two quantities defines the *submodularity ratio* γ, which quantifies how closely the function f adheres to submodularity. Let

$$\Phi_\mathcal{L} = \begin{bmatrix}\phi(\mathbf{x}_1)^\top \\ \phi(\mathbf{x}_2)^\top \\ \vdots \\ \phi(\mathbf{x}_r)^\top\end{bmatrix}$$

be the matrix obtained by stacking the feature vectors corresponding to the elements of $\mathcal{L}$. Then the updated covariance matrix becomes

$$\mathbf{V}_{\mathcal{S}\cup\mathcal{L}} = \mathbf{V}_\mathcal{S} + \Phi_\mathcal{L}^\top\Phi_\mathcal{L}.$$

To compute $\mathbf{V}_{\mathcal{S}\cup\mathcal{L}}^{-1}$, we apply the Woodbury matrix identity and obtain

$$\begin{aligned}
\mathbf{V}_{\mathcal{S}\cup\mathcal{L}}^{-1} = \mathbf{V}_\mathcal{S}^{-1} - \\
\mathbf{V}_\mathcal{S}^{-1}\Phi_\mathcal{L}^\top\left(\mathbf{I} + \Phi_\mathcal{L}\mathbf{V}_\mathcal{S}^{-1}\Phi_\mathcal{L}^\top\right)^{-1}\Phi_\mathcal{L}\mathbf{V}_\mathcal{S}^{-1}.
\end{aligned}$$

Substituting this into the objective gives

$$\begin{aligned}
&f_\mathbf{z}(\mathcal{S}\cup\mathcal{L}) = -\phi(\mathbf{z})^\top\mathbf{V}_{\mathcal{S}\cup\mathcal{L}}^{-1}\phi(\mathbf{z}) \\
&= -\phi(\mathbf{z})^\top\mathbf{V}_\mathcal{S}^{-1}\phi(\mathbf{z}) \\
&+ \phi(\mathbf{z})^\top\mathbf{V}_\mathcal{S}^{-1}\Phi_\mathcal{L}^\top\left(\mathbf{I} + \Phi_\mathcal{L}\mathbf{V}_\mathcal{S}^{-1}\Phi_\mathcal{L}^\top\right)^{-1}\Phi_\mathcal{L}\mathbf{V}_\mathcal{S}^{-1}\phi(\mathbf{z}).
\end{aligned}$$

Hence the total gain from adding $\mathcal{L}$ is

$$\begin{aligned}
&f_\mathbf{z}(\mathcal{S}\cup\mathcal{L}) - f_\mathbf{z}(\mathcal{S}) \\
&= \phi(\mathbf{z})^\top\mathbf{V}_\mathcal{S}^{-1}\Phi_\mathcal{L}^\top\left(\mathbf{I} + \Phi_\mathcal{L}\mathbf{V}_\mathcal{S}^{-1}\Phi_\mathcal{L}^\top\right)^{-1}\Phi_\mathcal{L}\mathbf{V}_\mathcal{S}^{-1}\phi(\mathbf{z}).
\end{aligned}$$

Let $\mathbf{w} = \mathbf{V}_\mathcal{S}^{-1}\phi(\mathbf{z})$. Then, we have

$$\begin{aligned}
&f_\mathbf{z}(\mathcal{S}\cup\mathcal{L}) - f_\mathbf{z}(\mathcal{S}) \\
&= \mathbf{w}^\top\Phi_\mathcal{L}^\top\left(\mathbf{I} + \Phi_\mathcal{L}\mathbf{V}_\mathcal{S}^{-1}\Phi_\mathcal{L}^\top\right)^{-1}\Phi_\mathcal{L}\mathbf{w}\ ,
\end{aligned}$$

which can be interpreted as a bilinear form involving the projection of the vector $\mathbf{w}$ onto the subspace spanned by the rows of $\Phi_{\mathcal{L}}$, that is, the set of vectors $\{\phi(\mathbf{x}_j)\}_{j\in\mathcal{L}}$. The matrix $\left(\mathbf{I}+\Phi_{\mathcal{L}}\mathbf{V}_{\mathcal{S}}^{-1}\Phi_{\mathcal{L}}^{\top}\right)^{-1}$ acts as a correction factor that accounts for interactions between the new feature directions $\{\phi(\mathbf{x}_j)\}_{j\in\mathcal{L}}$ and the current inverse covariance $\mathbf{V}_{\mathcal{S}}^{-1}$. Intuitively, the total gain from adding a group of elements depends not only on the alignment of each $\phi(\mathbf{x}_j)$ with $\mathbf{w}$ but also on their mutual (in)dependence in the $\mathbf{V}_{\mathcal{S}}^{-1}$-induced geometry.

Let us define the matrix $\mathbf{G} = \Phi_{\mathcal{L}}\mathbf{V}_{\mathcal{S}}^{-1}\Phi_{\mathcal{L}}^{\top} = \mathbf{D}+\mathbf{N}$, where $\mathbf{D} = \operatorname{diag}\big(\phi(\mathbf{x}_j)^{\top}\mathbf{V}_{\mathcal{S}}^{-1}\phi(\mathbf{x}_j)\big)$ and $\mathbf{N}$ contains the off-diagonal terms $\big(\phi(\mathbf{x}_j)^{\top}\mathbf{V}_{\mathcal{S}}^{-1}\phi(\mathbf{x}_k)\big)_{j\neq k}$. We ideally want these terms to be low, i.e., low mutual coherence among the rows of $\Phi_{\mathcal{L}}$ in the $\mathbf{V}_{\mathcal{S}}^{-1}$-inner product. Now see that

$$\begin{aligned}(\mathbf{D}+\mathbf{N})^{-1} &= \big(\mathbf{D}\left(\mathbf{I}+\mathbf{D}^{-1}\mathbf{N}\right)\big)^{-1}\\ &= \left(\mathbf{I}+\mathbf{D}^{-1}\mathbf{N}\right)^{-1}\mathbf{D}^{-1}.\end{aligned}$$

If $\|\mathbf{D}^{-1}\mathbf{N}\| \ll 1$, then we can expand the inverse using a Neumann series [4]

$$\left(\mathbf{I}+\mathbf{D}^{-1}\mathbf{N}\right)^{-1} = \sum_{k=0}^{\infty}(-1)^k(\mathbf{D}^{-1}\mathbf{N})^k.$$

Then we obtain

$$(\mathbf{D}+\mathbf{N})^{-1} = \left(\sum_{k=0}^{\infty}(-1)^k(\mathbf{D}^{-1}\mathbf{N})^k\right)\mathbf{D}^{-1}.$$

Keeping only the first two terms in the series, we get $(\mathbf{D}+\mathbf{N})^{-1} \approx \mathbf{D}^{-1}-\mathbf{D}^{-1}\mathbf{N}\mathbf{D}^{-1}$. This approximation is accurate when the norm of $\mathbf{N}$ is small relative to $\mathbf{D}$, i.e., $\|\mathbf{D}^{-1}\mathbf{N}\| = \|\mathbf{D}^{-1/2}\mathbf{N}\mathbf{D}^{-1/2}\| \ll 1$ as $\mathbf{N}$ is a symmetric matrix. Therefore, we get

$$\begin{aligned}(\mathbf{I}+\mathbf{G})^{-1} &\approx \operatorname{diag}\left(\frac{1}{1+\phi(\mathbf{x}_j)^{\top}\mathbf{V}_{\mathcal{S}}^{-1}\phi(\mathbf{x}_j)}\right)\\ &+\mathbf{D}^{-1}\mathbf{N}\mathbf{D}^{-1}.\end{aligned}$$

Now, note that

$$\begin{aligned}&\mathbf{w}^{\top}\mathbf{\Phi}_{\mathcal{L}}^{\top}\operatorname{diag}\left(\frac{1}{1+\phi(\mathbf{x}_j)^{\top}\mathbf{V}_{\mathcal{S}}^{-1}\phi(\mathbf{x}_j)}\right)\Phi_{\mathcal{L}}\mathbf{w}\\ &= \sum_{i=1}^{r}\frac{(\phi(\mathbf{z})^{\top}\mathbf{V}_{\mathcal{S}}^{-1}\phi(\mathbf{x}_i))^2}{1+\phi(\mathbf{x}_i)^{\top}\mathbf{V}_{\mathcal{S}}^{-1}\phi(\mathbf{x}_i)} = \sum_{i=1}^{r}\Delta_i\ .\end{aligned}$$

[4] This condition holds when vectors $\phi(\mathbf{x}_i)$ and $\phi(\mathbf{x}_j)$ are nearly orthogonal under the $\mathbf{V}_{\mathcal{S}}^{-1}$-induced geometry, which ensures that the off-diagonal elements of $\mathbf{G}$ are small relative to the diagonal ones.

Similarly, for the diagonal matrix $\mathbf{D} = \operatorname{diag}(d_1, d_2, \ldots, d_n)$ and the off-diagonal matrix $\mathbf{N} = [n_{ij}] \in \mathbb{R}^{n\times n}$, we have

$$\mathbf{w}^{\top}\mathbf{D}^{-1}\mathbf{N}\mathbf{D}^{-1}\mathbf{w} = \sum_{i\neq j}\frac{w_i w_j}{d_i d_j}n_{ij}.$$

This gives us

$$\begin{aligned}&\mathbf{w}^{\top}\Phi_{\mathcal{L}}^{\top}\mathbf{D}^{-1}\mathbf{N}\mathbf{D}^{-1}\Phi_{\mathcal{L}}\mathbf{w}\\ &= \sum_{i\neq j}\Big(\frac{\phi(\mathbf{z})^{\top}\mathbf{V}_{\mathcal{S}}^{-1}\phi(\mathbf{x}_i)}{1+\phi(\mathbf{x}_i)^{\top}\mathbf{V}_{\mathcal{S}}^{-1}\phi(\mathbf{x}_i)}\Big)\\ &\times\Big(\frac{\phi(\mathbf{z})^{\top}\mathbf{V}_{\mathcal{S}}^{-1}\phi(\mathbf{x}_j)}{1+\phi(\mathbf{x}_j)^{\top}\mathbf{V}_{\mathcal{S}}^{-1}\phi(\mathbf{x}_j)}\Big)\times(\phi(\mathbf{x}_i)^{\top}\mathbf{V}_{\mathcal{S}}^{-1}\phi(\mathbf{x}_j))\\ &= \sum_{i\neq j}\sqrt{\Delta_i\Delta_j}\mu_{i,j}\ ,\end{aligned}$$

where μ_{ij} is given by

$$\frac{\phi(\mathbf{x}_i)^{\top}\mathbf{V}_{\mathcal{S}}^{-1}\phi(\mathbf{x}_j)}{\sqrt{1+\phi(\mathbf{x}_i)^{\top}\mathbf{V}_{\mathcal{S}}^{-1}\phi(\mathbf{x}_i)}\sqrt{1+\phi(\mathbf{x}_j)^{\top}\mathbf{V}_{\mathcal{S}}^{-1}\phi(\mathbf{x}_j)}}.$$

Therefore, we obtain

$$f_{\mathbf{z}}(\mathcal{S}\cup\mathcal{L}) - f_{\mathbf{z}}(\mathcal{S}) = \sum_{i=1}^{k}\Delta_i - \sum_{i\neq j}\sqrt{\Delta_i\Delta_j}\mu_{i,j}\ .$$

This implies

$$\gamma_r(f,\mathbf{z},\mathcal{L},\mathcal{S}) = \frac{\sum_{i=1}^{r}\Delta_i}{\sum_{i=1}^{r}\Delta_i - \sum_{i\neq j}\sqrt{\Delta_i\Delta_j}\mu_{ij}}.$$

Let us denote $a_i := \sqrt{\Delta_i}$. Then we have

$$\gamma_r(f,\mathbf{z},\mathcal{L},\mathcal{S}) = \frac{\sum\limits_{i=1}^{r}a_i^2}{\sum\limits_{i=1}^{r}a_i^2 - \sum\limits_{i\neq j}a_i a_j\mu_{i,j}}.$$

Using the bound $|\mu_{i,j}| \leq \mu$, we get

$$\begin{aligned}&|\sum_{i\neq j}a_i a_j\mu_{i,j}| \leq \mu\sum_{i\neq j}|a_i a_j|\\ &\implies -\sum_{i\neq j}a_i a_j\mu_{i,j} \leq \mu\sum_{i\neq j}|a_i a_j|\ .\end{aligned}$$

To bound $\sum_{i\neq j}|a_i a_j|$, note that

$$\begin{aligned}&\left(\sum_{i=1}^{r}|a_i|\right)^2 = \sum_{i=1}^{r}a_i^2 + \sum_{i\neq j}|a_i a_j|\\ &\implies \sum_{i\neq j}|a_i a_j| = \left(\sum_{i=1}^{r}|a_i|\right)^2 - \sum_{i=1}^{r}a_i^2\\ &\implies \sum_{i\neq j}|a_i a_j| \leq (r-1)\sum_{i=1}^{r}a_i^2\ ,\end{aligned}$$

since by the Cauchy-Schwarz inequality $(\sum_{i=1}^{r} |a_i|)^2 \leq r \sum_{i=1}^{r} a_i^2$. Therefore, we can bound the denominator as

$$\sum_{i=1}^{r} a_i^2 - \sum_{i \neq j} a_i a_j \mu_{i,j} \leq (1 + (r-1)\mu) \sum_{i=1}^{r} a_i^2 ,$$

which yields

$$\gamma_r(f, \mathbf{z}, \mathcal{L}, \mathcal{S}) \geq \frac{1}{1 + (r-1)\mu} .$$

Since the bound holds for any $\mathbf{z}$, any $\mathcal{S} \subseteq \mathcal{X}$ and any $\mathcal{L} \subseteq \mathcal{X}$ such that $\mathcal{L} \cap \mathcal{S} = \emptyset$, we can conclude the proof by applying kernel trick to rewrite

$$\mu_{i,j} = \frac{k_{\mathcal{S}}(\mathbf{x}_i, \mathbf{x}_j)}{\sqrt{\beta + k_{\mathcal{S}}(\mathbf{x}_i, \mathbf{x}_i)}\sqrt{\beta + k_{\mathcal{S}}(\mathbf{x}_j, \mathbf{x}_j)}} .$$

C.3 Proof of Theorem 1

From Das and Kempe (2011), we know that for any $\mathbf{z} \in \mathbb{R}^d$, if the greedy algorithm return a set $\mathcal{S}_{\text{greedy}}$ for the optimization problem in equation equation 2, then $\mathcal{S}_{\text{greedy}}$ satisfies

$$f_{\mathbf{z}}(\mathcal{S}_{\text{greedy}}) \geq \left(1 - e^{-\gamma}\right) \cdot f(\mathcal{S}^*),$$

where S^* is an optimal solution of size at most r. Substituting the lower bound on submodularity ratio from Lemma 1, i.e., $\gamma \geq \frac{1}{1+(r-1)\mu}$, we get the desired bound.

C.4 Proof of Lemma 2

The result holds by first noting that $\mathbf{K}_{\mathcal{S}} = \Phi_{\mathcal{S}} \Phi_{\mathcal{S}}^{\top}$, then applying Sylvester's identity to get $\det(\mathbf{K}_{\mathcal{S}} + \beta \mathbf{I}_r) = \det \mathbf{V}_{\mathcal{S}}$, and finally invoking the matrix-determinant lemma, which, for any $\mathbf{x} \notin \mathcal{S}$, yields

$$\begin{aligned} &\log\det(\mathbf{V}_{\mathcal{S}} + \phi(\mathbf{x})\phi(\mathbf{x})^{\top}) - \log\det(\mathbf{V}_{\mathcal{S}}) \\ &= \log(1 + \phi(\mathbf{x})^{\top} \mathbf{V}_{\mathcal{S}}^{-1} \phi(\mathbf{x})) \\ &= \log(1 + k_{\mathcal{S}}(\mathbf{x}, \mathbf{x})/\beta). \end{aligned} \tag{7}$$

Monotonicity follows since this increment $\log(1 + k_{\mathcal{S}}(\mathbf{x}, \mathbf{x})/\beta)$ is non-negative. Submodularity holds because this increment decreases as the set S grows, due to redundancy in the directions already spanned. Furthermore, equation 7 naturally admits the greedy selection rule.

D Dataset Descriptions

Table 25 lists the prompt templates and an example input–output pair for each dataset, along with the train/validation split sizes used in our experiments. The text below provides a short description of each.

SST-2 is a sentiment classification benchmark containing two coarse-grained classes: *positive* and *negative*.

SST-5 is a sentiment classification benchmark containing five fine-grained classes: *very positive*, *positive*, *neutral*, *negative*, and *very negative*.

MRPC is a corpus of sentence pairs automatically extracted from online news sources, with human annotations indicating whether the sentences in each pair are semantically equivalent.

MNLI is a crowdsourced collection of premise–hypothesis sentence pairs annotated for textual entailment. The task is to predict whether the premise *entails*, *contradicts*, or is *neutral* with respect to the hypothesis.

QNLI is a question–sentence dataset derived from QA pairs, where the task is to determine whether the given context sentence contains the answer to the question.

CMSQA is a multiple-choice question-answering dataset requiring diverse forms of commonsense knowledge. Given a question and five candidate answers, the model must select the correct one.

HellaSwag is a large-scale benchmark for grounded commonsense reasoning. Each example pairs a context with four candidate endings: one true video caption (from ActivityNet Captions and the Large Scale Movie Description Challenge) and three adversarially generated distractors designed to fool machines.

SWAG (Zellers et al., 2018) is a multiple-choice sentence-completion benchmark for grounded commonsense inference. Each example consists of a partial sentence followed by four candidate continuations; the model must select the most plausible one. *Note:* the open-source experiments (Table 1) use HellaSwag (Zellers et al., 2019); the frontier LLM experiments (Tables 3–13) use SWAG, a related but distinct benchmark from the same authors.

IMDB is a binary sentiment classification benchmark of movie reviews. Each review is labeled *positive* or *negative*.

DBPedia is a 14-class topic-classification benchmark constructed from Wikipedia article abstracts, with classes spanning company, educational institution, artist, athlete, office holder, mean of transportation, building, natural place, village, animal, plant, album, film, and written work.

BoolQ is a yes/no question-answering benchmark. Each example consists of a passage and a question, and the task is to predict whether the answer is *yes*

or *no* given the passage.

E Extended Results on Frontier LLMs

We provide complete experimental results for the frontier LLM evaluation described in Section 4. All experiments use β=0.02 and report the best result across five kernel choices (Linear, RBF, Laplacian, Matérn-3/2, Rational Quadratic) unless otherwise stated.

E.1 Per-Model Per-Dataset Results (r=5, OpenAI Embeddings)

Tables 9–10 present per-dataset head-to-head comparisons for all four frontier models.

Dataset	KITE	KNN	CEIL	Δ_{KNN}	Δ_{CEIL}
SST-5	49.8	**50.0**	51.4	−0.2	−1.6
MRPC	**72.1**	70.3	69.6	+1.7	+2.5
QNLI	**87.6**	85.6	84.8	+2.0	+2.8
SWAG	**70.2**	69.6	69.4	+0.6	+0.8
IMDB	**99.6**	99.0	98.6	+0.6	+1.0
DBPedia	**99.8**	99.4	99.6	+0.4	+0.2
BoolQ	**90.0**	89.6	89.6	+0.4	+0.4
Avg	**81.3**	80.5	80.4	+0.8	+0.9

Table 9: GPT-4o-mini: Per-dataset accuracy (%) on core benchmarks (r=5, OpenAI large text embeddings of 3072 dimensions).

Dataset	KITE	KNN	CEIL	Δ_{KNN}	Δ_{CEIL}
SST-5	**54.4**	54.0	52.6	+0.4	+1.8
MRPC	**74.5**	72.1	74.0	+2.5	+0.5
QNLI	**86.2**	82.6	81.4	+3.6	+4.8
SWAG	**72.8**	70.6	72.0	+2.2	+0.8
IMDB	99.8	99.8	**100.0**	0.0	−0.2
DBPedia	99.2	99.2	**99.4**	0.0	−0.2
BoolQ	**93.2**	92.6	92.4	+0.6	+0.8
Avg	**82.9**	81.6	81.7	+1.3	+1.2

Table 10: GPT-5.2: Per-dataset accuracy (%) on core benchmarks (r=5, OpenAI large text embeddings).

E.2 Effect of Increasing In-Context Examples (r=10)

To study the interaction between KITE and the number of in-context examples, we repeat the frontier evaluation with r=10 (OpenAI large text embeddings of dim 3072). Table 11 shows that KITE's advantages persist and, in several cases, grow with more examples. On GPT-4.1, the gain over KNN increases from +1.0 (r=5) to +1.9 (r=10); on Claude-Haiku-4.5, from +1.2 to +2.1. This aligns with the intuition that diversity-regularized selection becomes more important as the number of selected examples grows, since redundancy in the selected set is more likely.

Model	BM25	KNN	CEIL	KITE
GPT-4o-mini	80.0	80.6	81.4	**81.4**
GPT-4.1	81.8	81.1	81.9	**83.0**
Claude-Haiku-4.5	83.0	83.0	84.2	**85.1**
GPT-5.2	80.8	82.1	82.6	**83.0**

Table 11: Average accuracy (%) on seven classification datasets with r=10 in-context examples (OpenAI large text embeddings). KITE gains over KNN grow for GPT-4.1 and Claude-Haiku-4.5 compared to r=5 (Table 3).

Table 12 presents per-dataset results for GPT-4.1 and Claude-Haiku-4.5 at r=10. The gains are particularly striking on Claude-Haiku-4.5: KITE improves over KNN by +4.2 on QNLI, +3.2 on SST-5, and +3.0 on SWAG.

Table 13 reports the corresponding per-dataset breakdown for the remaining two frontier models, GPT-4o-mini and GPT-5.2, at r=10. On GPT-5.2 the largest gain over KNN appears on QNLI (+5.2), with additional improvements on SWAG (+1.8) and SST-5 (+0.2). GPT-4o-mini shows more modest gains since several datasets are already near-saturated for this model, but KITE still leads on SWAG (+2.6 over KNN), BoolQ (+1.2), and SST-5 (+1.0).

E.3 Robustness to Embedding Encoder: NV-Embed-v2

Section 4 (Table 5 in the main paper) showed that KITE is robust across five BERT-family encoders. Here we extend this analysis to a fundamentally different and substantially stronger encoder: NV-Embed-v2 (Lee et al., 2024), a state-of-the-art text embedding model. All experiments use r=5.

As shown in Table 14, KITE achieves the highest average on all four frontier models when using NV-Embed-v2 embeddings, with gains of +1.0 to +2.1 points over the best KNN retriever. The gains are broadly consistent with – and in fact stronger than – the OpenAI-embedding results in Table 3, confirming that KITE's selection criterion transfers across embedding architectures.

Tables 15 and 16 provide the complete per-dataset head-to-head breakdown that the averages in Table 14 summarize. Several patterns are worth highlighting. First, KITE's largest gains under NV-Embed-v2 appear on SWAG (+3.0 to +4.4 over KNN across all four models), MRPC (+2.9 to +4.9), and SST-5 (up to +5.4 on GPT-4o-mini) – exactly the datasets where balancing relevance with

	GPT-4.1					Claude-Haiku-4.5				
Dataset	**KNN**	**DPP**	**KITE**	Δ_K	Δ_D	**KNN**	**DPP**	**KITE**	Δ_K	Δ_D
SST-5	51.8	54.6	**56.2**	+4.4	+1.6	54.6	58.6	57.8	+3.2	−0.8
MRPC	69.6	69.6	**71.8**	+2.2	+2.2	74.5	73.3	**76.0**	+1.5	+2.7
QNLI	85.6	87.2	**88.0**	+2.4	+0.8	85.2	87.8	**89.4**	+4.2	+1.6
SWAG	70.4	71.8	**74.2**	+3.8	+2.4	76.6	77.8	**79.6**	+3.0	+1.8
IMDB	**100.0**	99.8	99.6	−0.4	−0.2	99.6	99.6	99.6	0.0	0.0
DBPedia	**99.2**	99.0	**99.2**	0.0	+0.2	99.0	**99.6**	**99.6**	+0.6	0.0
BoolQ	91.4	91.4	**92.2**	+0.8	+0.8	91.4	93.0	**93.6**	+2.2	+0.6
Avg	81.1	81.9	**83.0**	+1.9	+1.1	83.0	84.2	**85.1**	+2.1	+0.9

Table 12: Per-dataset accuracy (%) for GPT-4.1 and Claude-Haiku-4.5 at r=10 (OpenAI large text embeddings of dim 3072).

	GPT-4o-mini					GPT-5.2				
Dataset	**KNN**	**CEIL**	**KITE**	Δ_K	Δ_C	**KNN**	**CEIL**	**KITE**	Δ_K	Δ_C
SST-5	50.4	**53.4**	51.4	+1.0	−2.0	53.8	**54.6**	54.0	+0.2	−0.6
MRPC	**73.0**	71.8	72.8	−0.2	+1.0	74.3	**74.5**	73.5	−0.8	−1.0
QNLI	85.4	**86.8**	86.0	+0.6	−0.8	83.0	84.4	**88.2**	+5.2	+3.8
SWAG	67.6	68.8	**70.2**	+2.6	+1.4	71.8	72.8	**73.6**	+1.8	+0.8
IMDB	99.6	99.8	**100.0**	+0.4	+0.2	**99.8**	**99.8**	**99.8**	0.0	0.0
DBPedia	99.6	**99.8**	99.6	0.0	−0.2	**99.6**	**99.6**	99.2	−0.4	−0.4
BoolQ	88.8	89.6	**90.0**	+1.2	+0.4	92.4	92.2	**92.8**	+0.4	+0.6
Avg	80.6	81.4	**81.4**	+0.8	+0.0	82.1	82.6	**83.0**	+0.9	+0.4

Table 13: Per-dataset accuracy (%) for GPT-4o-mini and GPT-5.2 at r=10 (OpenAI large text embeddings of dim 3072). Best result per dataset in **bold**.

diversity matters most. Second, on near-saturated benchmarks (IMDB, DBPedia) all methods are tied within noise, as expected.

E.4 Kernel Comparison on Frontier LLMs

Table 17 extends the kernel ablation study (Table 4 in the main paper) to frontier LLMs. We report the average accuracy over the seven classification datasets for each method–kernel combination with r=5 and OpenAI large text embeddings.

Several patterns emerge. First, KNN is largely insensitive to the kernel choice, since it only uses the kernel to compute pairwise similarities; all five kernels produce nearly identical results. Second, DPP shows more kernel sensitivity (up to 0.6% spread on GPT-4o-mini), and its best kernel varies across models. Third, KITE's best kernel consistently achieves the highest accuracy among all method–kernel combinations, and its worst-case kernel still matches or exceeds the best DPP configuration. This suggests that KITE's information-theoretic objective is robust to kernel misspecification while still benefiting from a good kernel choice.

Table 18 provides a complementary single-dataset, single-model breakdown on SST-5 with Qwen-3-1.7B, where all three retrievers are evaluated across the same seven kernels (Linear, Poly-3, Poly-4, RBF, Laplacian, Matérn-3/2, Rational Quadratic). The story matches Table 17: KITE's best kernel (50.77) edges out kNN's best (50.68) and CEIL's best (50.59), confirming the fairness conclusion at a finer kernel granularity than the frontier-LLM grid above.

E.5 Kernel and λ Selection Guidance

To guide practitioners, we evaluate all five kernels across $\lambda \in \{0, 0.5, 1\}$ on 13 datasets (9 classification, 4 QA/reasoning) and 4 frontier LLMs, yielding 780 evaluation cells (5 kernels × 3 λ values × 13 tasks × 4 models). For each (task, model, λ) triple, we record which kernel achieves the highest KITE accuracy (*win rate*), and measure the performance range across λ values (λ-*sensitivity*). Table 19 summarizes the results.

Key findings. **(1) Laplacian** achieves the highest win rate on both classification (35.2%) and QA/reasoning tasks (39.6%), and its optimal λ is 0.5—the intermediate setting that balances relevance and diversity. **(2) Matérn-3/2** is the most robust kernel: its accuracy varies by only 0.37 pp across $\lambda \in \{0, 0.5, 1\}$, making it a safe fallback when λ is not tuned. **(3) The Linear kernel is λ-sensitive on QA tasks**: while competitive at λ=0, its per-cell accuracy spread (8.89 pp average) is

Model	BM25	KNN	CEIL	KITE	$\Delta_{\text{KITE–KNN}}$
GPT-4o-mini	79.1	79.6	80.8	**81.7**	+2.1
GPT-4.1	80.2	81.7	81.6	**82.7**	+1.0
Claude-Haiku-4.5	81.1	81.6	82.6	**83.6**	+2.0
GPT-5.2	79.5	81.4	82.5	**82.6**	+1.2

Table 14: Average accuracy (%) on the seven classification datasets using NV-Embed-v2 embeddings (r=5). KITE achieves the highest average on all four frontier models.

	GPT-4o-mini					GPT-4.1				
Dataset	**KNN**	**CEIL**	**KITE**	Δ_{K}	Δ_{C}	**KNN**	**CEIL**	**KITE**	Δ_{K}	Δ_{C}
SST-5	47.6	49.6	**53.0**	+5.4	+3.4	55.2	53.4	**55.4**	+0.2	+2.0
MRPC	69.4	**72.3**	**72.3**	+2.9	0.0	67.4	68.4	**72.1**	+4.7	+3.7
QNLI	85.8	86.6	**87.4**	+1.6	+0.8	**88.2**	87.4	87.2	−1.0	−0.2
SWAG	67.0	68.0	**70.0**	+3.0	+2.0	70.8	71.4	**75.2**	+4.4	+3.8
IMDB	**99.4**	**99.4**	99.2	−0.2	−0.2	99.4	**99.8**	99.4	0.0	−0.4
DBPedia	99.6	**99.8**	99.6	0.0	−0.2	**99.2**	99.0	98.8	−0.4	−0.2
BoolQ	88.6	89.6	**90.2**	+1.6	+0.6	91.4	**91.6**	91.0	−0.4	−0.6
Avg	79.6	80.8	**81.7**	+2.1	+0.9	81.7	81.6	**82.7**	+1.0	+1.1

Table 15: Per-dataset accuracy (%) with NV-Embed-v2 embeddings (r=5) for GPT-4o-mini and GPT-4.1. Best result per dataset in **bold**.

driven by QA settings where λ=1 can hurt substantially; on classification the spread is under 1 pp (Table 20). It is safest to fix λ=0 when using the Linear kernel. **(4) RBF** has the lowest win rate on QA tasks (16.7%) and should generally be avoided. The kernel ranking is consistent between classification and QA (Spearman $\rho = 0.80$), suggesting that these preferences transfer across task types.

Recommendation. Use **Laplacian with** λ=0.5 as the default KITE configuration—it achieves the highest win rate across both task families and benefits from the diversity bonus. If λ cannot be validated (e.g., zero-shot deployment), use **Matérn-3/2 with** λ=0; its flat λ-response ($\leq$0.37 pp spread) makes it essentially insensitive to this hyperparameter. When using the Linear kernel, fix λ=0.

λ sensitivity: classification vs. QA. Table 20 reports the classification counterpart of the QA ablation in Table 22. Two differences stand out. First, λ sensitivity on classification is uniformly small (under 1 pp for every kernel), meaning the choice of λ matters little for focused multi-class tasks at r=5. Second, Linear, RBF, and RQ slightly prefer λ=0 on classification, whereas Laplacian and Matérn-3/2 prefer λ=0.5 on QA — consistent with the win-rate ranking in Table 19. The practical implication: on classification, λ can be fixed to any value in $\{0, 0.5\}$ with negligible effect; on QA, λ=0.5 provides a consistent, if modest, improvement for the stronger kernels.

E.6 Comparison of KITE Gains at r=5 vs. r=10

Table 21 directly compares KITE's improvement over KNN at two exemplar budgets, confirming that the diversity regularizer becomes more valuable as the number of selected examples increases.

Model	Δ at r=5	Δ at r=10
GPT-4o-mini	+0.8	+0.8
GPT-4.1	+1.0	+1.9
Claude-Haiku-4.5	+1.2	+2.1
GPT-5.2	+1.3	+0.9

Table 21: $\Delta_{\text{KITE–KNN}}$ on the seven classification datasets at r=5 vs. r=10 (OpenAI large text embeddings). Gains grow for GPT-4.1 and Claude-Haiku-4.5.

F QA and Reasoning Experiments

We extend the frontier-LLM evaluation to four QA and reasoning benchmarks: Natural Questions (NQ), TriviaQA, HotpotQA, and GSM8K. We retain the same retriever configurations, embedding model, and hyperparameters (r=5, β=0.02) described in Section 4. For NQ, TriviaQA, and HotpotQA, we report Soft Exact Match; for GSM8K, we report numeric match. For each dataset, we take the first 500 queries from the validation split for inference and the first 2000 examples from the train split as example bank.

Cross-Model Summary Table 7 reports the best-of-kernel average accuracy across all four QA tasks. KITE achieves the highest average on every model,

	Claude-Haiku-4.5					GPT-5.2				
Dataset	**KNN**	**CEIL**	**KITE**	Δ_K	Δ_C	**KNN**	**CEIL**	**KITE**	Δ_K	Δ_C
SST-5	49.6	**53.0**	52.6	+3.0	−0.4	51.8	**55.4**	53.6	+1.8	−1.8
MRPC	68.1	69.4	**73.0**	+4.9	+3.6	71.6	**73.5**	**73.5**	+1.9	0.0
QNLI	88.2	87.8	**89.2**	+1.0	+1.4	85.6	86.0	**87.2**	+1.6	+1.2
SWAG	75.6	76.6	**79.4**	+3.8	+2.8	69.8	70.8	**72.0**	+2.2	+1.2
IMDB	**98.4**	**98.4**	98.2	−0.2	−0.2	**100.0**	**100.0**	99.6	−0.4	−0.4
DBPedia	98.6	99.4	**99.6**	+1.0	+0.2	99.4	99.4	**99.6**	+0.2	+0.2
BoolQ	**93.0**	**93.4**	93.0	0.0	−0.4	91.8	92.6	**92.8**	+1.0	+0.2
Avg	81.6	82.6	**83.6**	+2.0	+1.0	81.4	82.5	**82.6**	+1.2	+0.1

Table 16: Per-dataset accuracy (%) with NV-Embed-v2 embeddings (r=5) for Claude-Haiku-4.5 and GPT-5.2. Best result per dataset in **bold**.

Method	**Kernel**	**4o-mini**	**4.1**	**Haiku**	**5.2**
KNN	Linear	78.2	78.8	79.7	78.5
	RBF	78.0	78.6	79.7	78.6
	Laplacian	78.1	78.8	79.7	78.8
	Matérn	78.1	78.6	79.7	78.8
	RQ	78.0	78.6	79.7	79.1
DPP	Linear	77.6	78.3	79.3	78.8
	RBF	77.5	78.0	79.4	78.8
	Laplacian	77.8	78.0	79.2	78.9
	Matérn	77.4	78.1	79.3	78.5
	RQ	77.2	78.4	79.4	78.9
KITE	Linear	78.3	79.2	79.6	**79.8**
	RBF	78.7	79.2	**80.0**	79.4
	Laplacian	78.4	79.4	**80.1**	79.3
	Matérn	**78.8**	**79.6**	79.6	79.6
	RQ	78.4	**79.6**	79.9	79.2

Table 17: Average accuracy (%) over the seven classification datasets by method and kernel (r=5, OpenAI large text embeddings). The best kernel for each method is underlined; the overall best per model is in **bold**.

Kernel	**kNN**	**CEIL**	**KITE**
Linear	**50.68**	50.23	48.50
Poly-3	50.67	50.41	**50.77**
Poly-4	50.68	**50.59**	49.59
RBF	50.66	**50.23**	49.68
Laplacian	50.68	49.59	**50.68**
Matérn-3/2	50.67	49.50	**50.23**
RQ	50.68	50.50	**50.77**
Best	50.68	50.59	**50.77**

Table 18: Accuracy (%) on SST-5 with Qwen-3-1.7B, where each method is evaluated under the same seven kernels. KITE's best kernel attains the highest accuracy; this is the SST-5 single-dataset analog of the frontier-LLM grid in Table 17.

Kernel	**CLS Win%**	**QA Win%**	**λ-range (pp)**	**Rec. λ**
Laplacian	**35.2**	**39.6**	1.38	0.5
Matérn-3/2	34.3	31.2	**0.37**	0.0
Linear	31.9	25.0	8.89†	0.0
RQ	25.9	25.0	0.34	0.0
RBF	22.2	16.7	0.75	0.0

Table 19: Kernel win rates and λ-sensitivity across 13 datasets and 4 frontier LLMs (r=5, OpenAI large embeddings). Win% = fraction of (task, model, λ) triples where this kernel achieves the highest KITE accuracy. λ-range = mean per-cell accuracy spread across $\lambda \in \{0, 0.5, 1\}$ (i.e., best minus worst λ for that task–model pair, averaged over all pairs). † The Linear kernel has a large per-cell λ-range (8.89 pp on average), driven primarily by QA tasks; on classification alone the spread is under 1 pp (Table 20).

with gains of $+0.2$ to $+1.8$ over the best KNN retriever, consistent with the classification results in Table 3.

Per-Task Breakdown Table 23 disaggregates these results by dataset. The largest improvements appear on open-domain factoid tasks: $+5.3$ on TriviaQA (Claude-Haiku-4.5), $+2.6$ on NQ (GPT-5.2), and $+2.2$ on HotpotQA (GPT-5.2). On GSM8K, where all frontier models already exceed 96% accuracy, retrieval method differences are marginal. On HotpotQA, DPP occasionally matches or exceeds KITE (e.g., Claude-Haiku-4.5), suggesting that its global diversity measure may better suit multi-hop settings with distractor paragraphs.

λ Ablation Table 22 reports accuracy averaged over all four QA tasks for each kernel–λ combina-

Kernel	λ=0.0	λ=0.5	λ=1.0
Linear	**81.2**	80.3	80.4
RBF	**81.2**	80.3	80.2
Laplacian	**80.8**	80.7	80.4
Matérn-3/2	80.8	**80.9**	80.4
RQ	**81.1**	80.3	80.4

Table 20: λ ablation for classification: average accuracy (%) over seven classification datasets and four frontier LLMs (r=5, OpenAI large text embeddings). Each entry is the mean accuracy of KITE with the given kernel and λ, averaged over all models and datasets. Best λ per kernel in **bold**. Compare with the QA analogue in Table 22.

Kernel	λ=0.0	λ=0.5	λ=1.0
Linear	**66.40**	66.15	65.84
RBF	64.33	**66.05**	65.94
Laplacian	66.30	**66.51**	65.83
Matérn-3/2	66.19	**66.48**	65.66
RQ	64.45	65.74	**65.99**

Table 22: λ ablation: average accuracy (%) over four QA tasks per kernel and λ value. Each cell is the mean across GPT-4o-mini, GPT-4.1, Claude-Haiku-4.5, and GPT-5.2. Best λ per kernel in **bold**.

tion. For most kernels, λ=0.5 yields the best or near-best accuracy. The linear kernel is an exception where pure relevance (λ=0.0) is slightly preferred. Compared with the classification analogue (Table 20 in App. E.5), QA tasks show a clearer benefit from λ=0.5 for Laplacian and Matérn-3/2, while classification is largely insensitive to λ across all kernels.

KITE achieves the highest average accuracy on all four models, with gains of +0.2 to +1.8 over the best KNN retriever, consistent with the classification results in Table 3 of the main paper.

Discussion. We observe: (1) First, KITE's gains over KNN are most pronounced on open-domain factoid tasks (NQ, TriviaQA), where the diversity-regularized selection helps avoid redundant exemplars that share surface-level similarity but provide limited additional signal. (2) Second, on HotpotQA, a multi-hop reasoning task where the "distractor" split includes irrelevant paragraphs, DPP occasionally matches or exceeds KITE, suggesting that DPP's global diversity measure may be better suited to settings requiring exemplars with complementary evidence chains. (3) Third, on GSM8K, all retrieval methods perform within a narrow range (∼1–2%), as the task is nearly saturated for frontier models; the exemplar choice matters less when the model already achieves >96% accuracy. Overall, the QA results complement the classification findings in the main paper, showing that KITE's information-theoretic framework extends to generative QA settings with consistent, though task-dependent, improvements.

G Computational Cost Analysis

Method	Qwen-2.5-7B (s)	Qwen-2.5-1.5B (s)
BM25	2.42	0.20
Dense	2.45	0.21
CEIL	2.58	0.21
KITE	2.61	0.23

Table 24: End-to-end wall-clock time per query (seconds) on SST-5 with r=50 in-context examples on a single Nvidia RTX A6000 GPU. LLM inference dominates total latency; KITE's greedy selection contributes at most 0.19 s of additional overhead.

A natural concern with greedy submodular selection is its per-query overhead relative to standard top-r retrieval. To quantify this, we measure end-to-end wall-clock time per query on the SST-5 validation set (1,101 queries against an exemplar bank of 8,534 examples) using two Qwen-2.5 variants on a single Nvidia RTX A6000 GPU. All four retrievers select the same number of in-context examples (r=50), so prompt lengths — and therefore LLM inference cost — are matched across methods.

The selection step contributes at most 0.03 s per query on Qwen-2.5-1.5B and 0.19 s on Qwen-2.5-7B relative to BM25 — a small fraction (under 8%) of total per-query latency, which is dominated by LLM inference. KITE's overhead over CEIL — the closest baseline algorithmically, since both perform greedy selection over a kernel matrix — is only 0.02–0.03 s. In practice, KITE's information-theoretic objective therefore imposes no meaningful deployment cost beyond standard kNN/DPP-based retrieval.

H Software and Hardware Used

We run all experiments with Python 3.12.8 and Transformers 4.49.0. For all experimentation, we use one Nvidia RTX A6000 GPU.

Task	Model	BM25	KNN-best	DPP-best	KITE-best	ΔKITE–KNN
NQ	GPT-4o-mini	48.6	50.4	**50.8**	**50.8**	+0.4
	GPT-4.1	54.8	57.2	57.2	**58.2**	+1.0
	Claude-Haiku-4.5	37.4	39.6	41.0	**41.4**	+1.8
	GPT-5.2	52.8	55.6	56.2	**58.2**	+2.6
TriviaQA	GPT-4o-mini	78.6	78.7	77.5	**79.2**	+0.5
	GPT-4.1	87.0	86.3	84.9	**87.8**	+1.5
	Claude-Haiku-4.5	75.8	74.7	74.4	**80.0**	+5.3
	GPT-5.2	**83.8**	83.6	82.2	83.4	−0.2
HotpotQA	GPT-4o-mini	35.2	36.4	**36.8**	36.4	+0.0
	GPT-4.1	45.4	45.4	**46.8**	**46.8**	+1.4
	Claude-Haiku-4.5	25.0	28.6	**29.6**	28.4	−0.2
	GPT-5.2	39.4	42.8	44.8	**45.0**	+2.2
GSM8K	GPT-4o-mini	94.2	95.0	**95.2**	95.0	+0.0
	GPT-4.1	97.4	97.0	**97.8**	97.2	+0.2
	Claude-Haiku-4.5	98.0	98.2	**98.8**	98.4	+0.2
	GPT-5.2	96.6	**97.4**	**97.4**	**97.4**	+0.0

Table 23: Per-task accuracy (%) on QA/reasoning benchmarks (r=5, OpenAI large text embeddings). Best result per row in **bold**. Δ columns show KITE's gain over the corresponding baseline.

Dataset	Prompt	Example	#Train	#Validation
SST-2	`{input} It is {output}`	**Input:** a stirring, funny and finally transporting re-imagining of beauty and the beast. **Output:** positive	67,349	872
SST-5	`{input} It is {output}`	**Input:** this is a stunning film, a one-of-a-kind tour de force. **Output:** very positive	8,534	1,101
MRPC	`{input1} Can we say "{input2}"? {output}`	**Input1:** The company didn't detail the costs of the replacement and repairs. **Input2:** But company officials expect the costs of the replacement work to run into the millions of dollars. **Output:** No	3,668	408
MNLI	`{input1} Can we say "{input2}"? {output}`	**Input1:** yeah i know and i did that all through college and it worked too **Input2:** I did that all through college but it never worked **Output:** No	392,568	19,647
QNLI	`{input1} Can we know "{input2}"? {output}`	**Input1:** As of that day, the new constitution heralding the Second Republic came into force. **Input2:** What came into force after the new constitution was herald? **Output:** Yes	104,707	5,463
CMSQA	`{input}{output}`	**Input:** Sammy wanted to go to where the people were. Where might he go? **Output:** populated areas	9,740	1,221
HellaSwag	`{input}{output}`	**Input:** Members of the procession walk down the street holding small horn brass instruments. A drum line **Output:** passes by walking down the street playing their instruments	52,611	20,006

Table 25: Datasets with corresponding prompts and examples used in the experiments.