In-context Example Selection with Influences

Nguyen Tai and Eric Wong
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104, USA
{taing,exwong}@seas.upenn.edu

Abstract

In-context learning (ICL) emerged as powerful learning paradigm for large language models (LLMs), where examples are given as context to the LLM to improve performance. However, ICL remains sensitive in practice to various design choices, and in particular the specific examples that make up the context. Our work proposes *in-context influences* as an approach to pinpoint example importance in ICL. We show that in-context influences can find minimal data support for changing the prediction on any target example. Our framework is effective in both classification and generation tasks: using fewer than 8-shot on AG News and MBPP can respectively flip 63% and 45% of the respective target sets. Our analysis reveals a positive association between data support and memorization. Beyond analysis of ICL, we also demonstrate applications of in-context influences for comparing models via distinguishing data subpopulations.[1]

1 Introduction

Large language models (LLMs) possess the remarkable ability to perform *in-context learning* (ICL) (Brown et al., 2020), a learning paradigm where the model learns and makes predictions on new, unseen examples from merely a handful of labeled instances provided as input. This unique capability allows LLMs to swiftly adapt to a variety of tasks without necessitating any changes to their underlying weights or architecture.

While we have begun to understand ICL in algorithms and pretraining data (Akyürek et al., 2022; Chen et al., 2024), its performance remains highly variable. In particular, sensitivity has been linked to biases such as the order of the examples (Lu et al., 2022), prompt templates (Lu et al., 2022; Kumar & Talukdar, 2021), and example selection (Liu et al., 2022a). Various mitigation methods have been proposed to address this brittleness in model calibration (Zhao et al., 2021) and template engineering (Liu et al., 2022b).

Given that not all in-context examples are equal, several others have focused on finding the optimal prompts. Liu et al. (2022a) proposes a distance-based selection method, using semantic similarity to rank individual candidate examples. Both Rubin et al. (2022) and Ye et al. (2023) cast ICL selection as a subset selection problem, training retrievers that search for the most relevant and diverse subsets. While these methods have varying effectiveness, there lacks a consensus on which of these signals are most important in ICL.

Motivated by this problem, our paper studies the relationship between influences and ICL to better understand the impact of in-context examples. Influences naturally lend to an offline example selection method that directly links the example presence to ICL outputs.

On 2 LLMs and 6 diverse tasks in text classification and generation, we demonstrate the efficacy of *influence-based example selection* at estimating the effect of train examples. In-context influences outperform several selection baselines at finding the minimal subset required to flip any predictions. By using just 8 examples, we can flip over 65% of the targets in AG

[1]Our code is released at https://github.com/BrachioLab/incontext_influences.

News and Amazon Reviews. We extend our influence framework to perform model comparison. On HANS, we reveal that Llama-13B significantly outperforms Mistral (e.g. 10.5% vs. 86.5%) on subsequence–non-entailment examples when their overall accuracy appears comparable. Analysis shows that *data support* has association with model memorization, while train influences might contain unknown nuances.

Overall, our contributions are as follows:

- We study in-context influence as a framework for selecting in-context examples in few-shot ICL. Our approach outperforms several baselines at measuring data support and robust at predicting model outcomes on unseen subsets.
- We apply influence embedding to compare model behaviors and clearly interpret their differences via data subpopulations.
- Data support shows positive linkage with memorization when memorization is beneficial to the model, and negative when inductive biases are required.

2 In-context Influences

A variety of methods, often regarded as *data attribution*, have been developed to understand how training data affects model performance. To estimate this effect, some methods use gradient information (Koh & Liang, 2017; Koh et al., 2019; Han et al., 2020; Pruthi et al., 2020) while others retrain models on subsets of the training data (Ghorbani & Zou, 2019; Ilyas et al., 2022). These methods all aim to quantify how a training example affects the prediction of a test example after training. Inspired by these frameworks, our goal is to trace how model performs given in-context examples and calculate the corresponding influences.

Our setup follows the retraining-based influence frameworks, which have two main steps. Let S be a training set, x a target, and $f(S, x)$ the model outputs after training on dataset S. Retraining-based influences first collect a "dataset" of M training runs $\mathcal{D}_x = \{(S_i, f(S_i, x)\}_{i=1}^M$ where $S_i \subseteq S$ are random subsets of the original training dataset. The second step is to use this dataset to estimate the influence of each train example $s \in S$, e.g. by learning a linear mapping (Ilyas et al., 2022).

Influences in k-shot prompting. To compute influences for *any* in-context examples, we leverage the following key observation: in ICL, "training" a model on a subset S' reduces to prompting the model on a sequence containing S'. Consequently, constructing the dataset $\mathcal{D}_x$ of training runs for ICL requires no gradient updates and is as costly as computing forward passes through the model. This drastically reduces the cost of calculating retraining-based influences, and can be calculated with only query-access to the model.

Specifically, for the first step, we construct the dataset of training runs $\mathcal{D}_x$ by performing k-shot prompting with subsets $S' \subseteq S$ where $|S'| = k$. For a fixed subset S', the performance of the resulting prompt containing S' is measured by observing model inference on x via a suitable performance metric. We repeat the process of prompting on random subsets $S' \subseteq S$ until each example in S has been seen ideally multiple times, achieving the set $\mathcal{D}_x = \{(S_i, f(S_i, x)\}_{i=1}^M$.

In the second step, we calculate the influence of each in-context example by establishing a linear mapping between examples and outputs (Ilyas et al., 2022). We define *in-context influence*, $\mathcal{I}_x(s)$, as the effect of an example s on model ICL outcomes. Specifically, we fit a linear model g_θ on the dataset $\mathcal{D}_x$ of input-output pairs to predict the margin[2]:

$$g_\theta(S', x) = \theta \cdot \mathbf{1}_{S'}^T + \theta_0 \tag{1}$$

where $S' \subseteq S$ is an example subset, $\mathbf{1}_{S'}$ is an indicator vector with the dimension of the train set S, and θ represents the parameters over S. A value of 1 at position p indicates that the example p is included in the subset S' and a value of 0 means otherwise. Following datamodels, we treat θ as influence estimates, and select in-context examples accordingly.

[2]We use linear regressions with stochastic gradient descent (SGD) and L1 regularization.

Algorithm 1 Influence-based example selection

Input: Language model `LLM`, training set $S = \{s_j\}_{j=1}^{N}$, target example x, f performance metric, number of in-context examples k (hyperparameter), and P number of total subsets (hyperparameter).

Step 1: Subset collections ◁ Compute influences

1: **for** $i = 1$ **to** M **do**
2: Randomly select subset $S_i \subseteq S$, where $|S_i| = k$
3: Compute $f(S_i, x)$ metric for classification/generation
4: Store the pair $\{S_i, f(S_i, x)\}$ for x
5: **end for**

Step 2: Calculate train example influence for x

1: Fit a `LinearSGD` on x to get θ following Equation 1
2: Assign $\mathcal{I}_x(s_j) = \theta_j$, where $|\theta| = N$

Step 3 Select example to flip x ◁ Selection for data support

1: Init S′ = {}
2: Add 1 example per class to S′ (for generation, a single example); observe prediction on $\texttt{LLM}(S', x)$
3: **while** prediction is not flipped **do**
4: Append s_j with next highest/lowest influence $\mathcal{I}_x(s_j)$ to S′
5: Perform $\texttt{LLM}(S', x)$ and observe prediction
6: **end while**

When f measures the margin, a higher score for $\mathcal{I}_x(s)$ corresponds to a higher confidence in model prediction on x when including training point s, analogous to the meaning of influences in the classic, non-prompted setting. As the number of collected subsets grows, estimates of in-context influences become more accurate. A sufficiently large M is one with good *coverage* for each example, meaning that each $s \in S$ gets seen multiple times. In this work, we try to achieve a coverage of 10 for each train example.

Post training, we use the proposed in-context influences to identify highly impactful in-context examples. Specifically, we expect the top influential examples to create the prompt that are likely to help model on target x (with respect to their influence scores). On the converse, we can also use the bottom influential examples to hinder model inference on x. A summary of the pipeline thus far is shown in Step 1 and 2 of Algorithm 1.

2.1 Performance metric

Any downstream performance metric suitable to evaluate a natural language task can be used for $f(S')$. In this work, we find the margin (Ilyas et al., 2022) to be an effective metric for classification, and the aggregated sequence log-probs to be robust for generation tasks.

For classification, the margin is defined as the difference between the log probability of the correct answer and the highest log probability among all incorrect answers, formulated as:

$$\text{MARGIN}(S', x) = \log P(\text{correct}|S') - \max(\log P(\text{incorrect answers}|S')) \tag{2}$$

For generation, we use the sequence score defined by the weighted log-probability of generation texts. A formal definition is provided Appendix Section A.1.

2.2 Cost analysis & Hyperparameters

Training cost. Retraining-based influence frameworks (Ilyas et al., 2022; Ghorbani & Zou, 2019) can require training hundreds of thousands of models. This is necessary to collect a sufficiently large enough dataset $\mathcal{D}_x$ to accurately estimate influences. In contrast, the cost of computing in-context influences is relatively cheap, as we do not need to train an end-to-end model. Instead of training, we simply prompt the LLM using a randomly sampled S' from original training set S. Thus, the complexity of calculating the margin from a sampled subset is proportional to a forward pass through the LLM.

Size of subsets. Our method has one parameter k, which controls the size of the random subsets $S' \subseteq S$ from which $\mathcal{D}_x$ is constructed. For ICL, $k = |S'|$ corresponds to the number of in-context examples given in the prompt. Unlike in the traditional setting, the context window length limit enforces a hard upper limit on the number of examples an LLM can be trained on via prompting. All models used in this work has a window size of 4096 tokens. Table 4 in the Appendix details the specific k chosen for each task.

3 Characterize ICL Brittleness

Building on our influence estimations, we discuss the concept of data support for ICL, which can be efficiently estimated with influences.

3.1 Data support

As a learning paradigm, in-context learning is known to be highly brittle. Various design choices in prompt format, example selection, and ordering can significantly impact model predictions (Lu et al., 2022; Liu et al., 2022a). With a focus on the few-shot examples, we measure brittleness by defining data support $\text{SUPPORT}_{\text{ICL}}(x)$ for each target x:

$$\text{SUPPORT}_{\text{ICL}}(x) = \min_{R \subseteq S} \{|R| : \text{Prediction}(x|R) \neq \text{Prediction}(x|S \setminus R)\} \tag{3}$$

where $R \subseteq S$ represents a subset and $|R|$ is its cardinality. The objective is to identify the smallest R for which model prediction on x changes when prompted with R, compared to when it is prompted with the remainder of the training set $S \setminus R$.

Different from the classical data support, our definition of data support for ICL considers *both* directions of a label flip, meaning we find the minimal subset that can likely flip a correct prediction to incorrect and vice versa. A higher data support implies that a target is resistant to changes in the prompt, which ties in directly with model confidence on x.

In standard end-to-end deep learning, one can reasonably change most model predictions by removing or perturbing all examples belonged to the class label (Ilyas et al., 2022). This is not always possible with LLMs, given that they hold prior knowledge and evidence of reasoning from the pretraining process. In spite of this, our experiments reveal that a major portion of the target set can be flipped on all tasks.

3.2 Estimate data support

Without a selection scheme, an exhaustive search for data support has the upper-bound cost of $\sum_{k=1}^{|S|} \binom{|S|}{k}$. $k = 1$ represents the single-element subsets and $k = |S|$ represents the subset with all elements in the train set S (within the boundary of the context window). This is prohibitively expensive. Thus, we propose several selection baselines to guide this process on a set of diverse language tasks and models.

Datasets. We choose 6 tasks for our study: 4 classification and 2 generation. These tasks cover a wide range of domains, including news classification (AG News), sentiment analysis (Amazon Reviews), textual entailment (HANS), toxicity detection (SBIC), fact recall (TriviaQA), and code generation (MBPP). We sample Train/Target splits to have 1500/600 respectively and maintain a uniform class distribution wherever appropriate.

Models. Our work uses two open-access models: **Llama 2-13B** (Touvron et al., 2023) and **Mistral-7B** (Jiang et al., 2023) .

Inference. For classification, there are multiple ways to perform inference (Holtzman et al., 2021). We follow one popular approach, which ranks all possible continuations to a prompt and chooses the continuation with the highest log-likelihood. We do not perform any token length or answer normalization tricks (Brown et al., 2020). For generation, we always use greedy sampling. Table 12 in the Appendix details the prompt templates used.

ID	Target text	Influence	Embedding
15938	Premise: The doctor and the actor stopped the lawyers. Hypothesis: The actor stopped the lawyers. Answer: entailment	Premise: The professors that thanked the presidents saw the actors. Hypothesis: The professors thanked the presidents. Answer: entailment *Score*: 0.0436 *Opp. Rank*: 614	Premise: The doctor behind the actor stopped the author. Hypothesis: The actor stopped the doctor. Answer: non-entailment *Score*: 0.930 *Opp. Rank*: 442
3476	Q: The works of which dramatic writer feature at least 64 bird species including all seven British crows? A: William Shakespeare	Q: In 2006 Lord Michael Levy earned himself which nickname? A: Lord Cashpoint *Score*: 0.146 *Opp. Rank*: 700	Q: Who composed the 1912 tone poem 'On Hearing the First Cuckoo of Spring'? A: (Frederick) Delius *Score*: 0.815 *Opp. Rank*: 201

Table 1: Top train example for a target on HANS and TriviaQA identified by in-context influence and text embedding baseline. *Opposite Rank* ↓ denotes where an example is ranked in the other selection method. Both show weak agreement. Embedding tends to capture examples with high lexical overlap or thematic similarity with the target, while Influence selects examples with less obvious relationships.

3.2.1 Influence-based selection

We estimate $\text{SUPPORT}_{\text{ICL}}(x)$ by selecting examples with the most positive or negative influence scores according to Algorithm 1. If influence estimates are meaningful, we would expect examples with positive influences to help turn an initially incorrect prediction to correct, and examples with negative influences to have the opposite effect. A summary of this step is detailed in Step 3 of Algorithm 1.

3.2.2 Non-influence baselines

We compare influence-based example selection to the following baselines, which optimize various metrics for example selection on target x:

1. **Random.** We randomly select new in-context examples to add to the prompt.
2. **BM25.** A sparse text representation method (Robertson, 2009) that extends TF-IDF to rank all train examples based on their similarity to target x.
3. **Embedding.** Liu et al. (2022a) finds examples with high semantic similarity to the target substantially improve ICL performance. To represent an example, we use `nomic-embed`[3] (Nussbaum et al., 2024), a state-of-the-art dense text embedding trained for long-context information retrieval. `CosineSim` is used as the distance measure to get the similarity score between x and candidates.[4]
4. **EPR.** Rubin et al. (2022) trains a dense retriever based on BERT-base (Devlin et al., 2019) to retrieve the best train subset with a similar focus on semantic parsing. The learning objective involves a contrastive objective with hard example mining and an LLM as the scoring function.
5. **CEIL.** Compositional Examples for ICL (CEIL) (Ye et al., 2023) builds on top of the EPR retriever, but further finetunes it with a *diversity* objective that leverages the Determinantal Point Process (DPP) to find the best subset. CEIL achieves state-of-the-art example selection across many classificaiton and generation tasks.

[3] `https://huggingface.co/nomic-ai/nomic-embed-text-v1`.

[4] We use Nomic embedding as the default text embedding in the paper, unless stated otherwise.

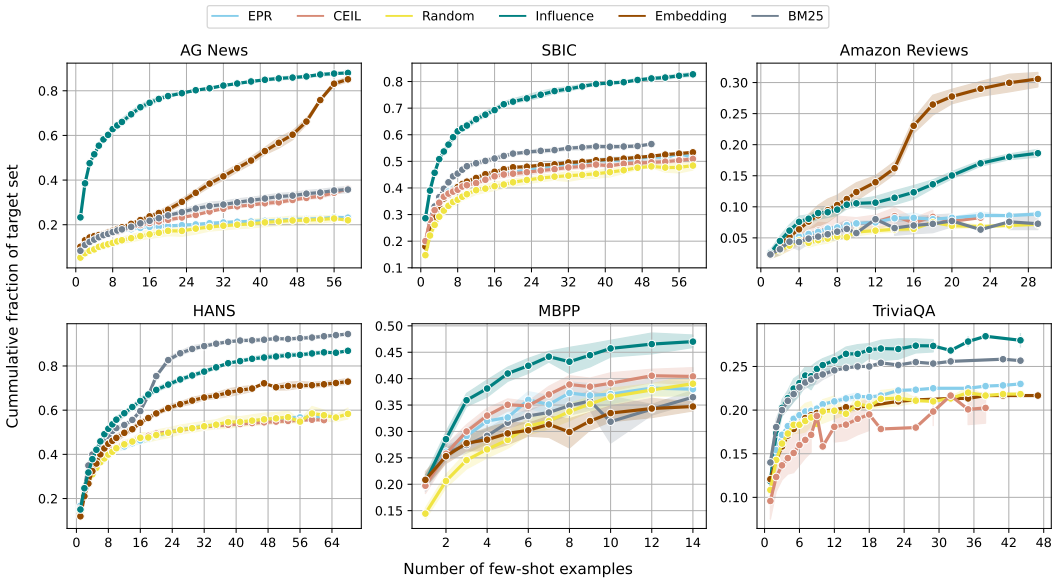

Figure 1: On the target set, find $\text{SUPPORT}_{\text{ICL}}(x)$, the smallest subset of few-shot examples that flips a Mistral-7B prediction. Influence-based example selection can flip over 65% of the target set in AG News using fewer than 10 examples.

Since label flipping defined in $\text{SUPPORT}_{\text{ICL}}(x)$ is bidirectional, it is important that we also select the most negative examples to induce misclassification. We modifiy both EPR and CEIL such that the retrievers considers the most negative candidate subsets, but keep the training process of the retriever the same. For both baselines, Mistral-7B is used as the scoring functions for fair evaluation. We run experiments over 4 seeds with the exception of MBPP, which we run for 20 seeds (Table 4).

3.2.3 Results

We visualize results for data support in Figure 1. Compared to other baselines, influence-based selection performs the best in 4 out of 6 tasks, and second best in HANS and Amazon Reviews. Using in-context influences, we can flip the most number of targets using the fewest number of few-shot examples. On AG News and Amazon Reviews, over 63% of the target set can be flipped with less than 8 examples, while over 40% can be flipped with 3. While text representation-base selections, BM25 and Embedding, can find examples that are semantically close to a target (Liu et al., 2022a), influences identify examples that more closely capture model confidence (Table 1). Both CEIL and EPR are not competitive for this task, likely due to the fact that they optimize for diversity and relevance of an entire subset size k, rather than considering the next most relevant example to.

Interestingly, our results reveal that generation tasks are also brittle, though to a lesser extent. Using 7 examples is sufficient for flipping 45% of MBPP, and using 8 examples is sufficient for flipping up to 25% of TriviaQA. The latter observation is intriguing, as TriviaQA entirely depends on fact recall learned in the pretraining process and we should not expect the examples to significantly alter model confidence.

4 Discussion

In this section, we discuss ICL influences & data support and analyze them across a number of distinct axes.

Predictability of ICL. In Section 2, we have learned a function mapping the presence of training points to outcomes. We follow this up by trying to predict the precise margin on x with unseen and random subsets. We visualize the results of predicting the margin in

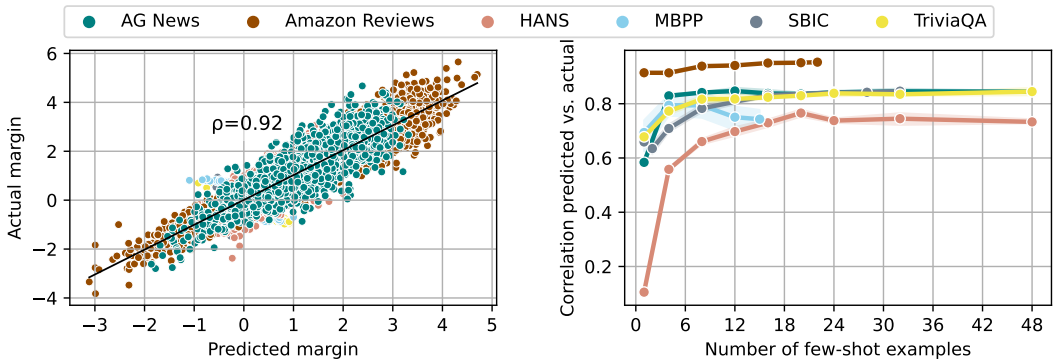

Figure 2: **Left:** Pearson correlation (ρ) between predicted and actual margins using in-context influences on Mistral-7B. There is strong positive correlation in most tasks, signaling high predictability. **Right:** Predictability scales well with increasing number of shot.

	length	**distance**$_{train}$	**ppl**	**ppl/zlib**	**min-K**$_5$	**min-K**$_{30}$	**min-K**$_{50}$
AGN	0.03	0.08	-0.01	-0.00	-0.00	-0.01	-0.01
Amazon	-0.16	-0.05	-0.02	0.06	0.12	-0.00	0.01
HANS	-0.04	-0.06	-0.16	-0.06	-0.04	-0.16	-0.16
SBIC	0.02	-0.05	-0.01	-0.05	-0.04	-0.00	-0.01
MBPP	-0.11	-0.01	0.19	0.18	0.12	0.18	0.19
TriviaQA	0.00	-0.08	0.00	0.01	0.02	-0.01	0.01

Table 2: Correlation (Pearson) between data support and example length, train set embedding distance, perplexity, perplexity/zlib, and different memorization scores from Min-K% (Shi et al., 2023). SUPPORT$_{\text{ICL}}$ has a slight but consistent positive associations with memorization and ppl on MBPP, and slight negative associations with similar metrics in HANS.

Figure 2 for Mistral-7B. Here, our linear mappings achieve good predictability ($\rho = 0.92$) that improves as the number of shot increases up to k, the size of the subsets used to estimate our influences in Equation 1.

Task recognition vs. task learning. With the same analysis on Llama-13B, we observe a curious ICL phenomenon for HANS (Appendix Figure 11), where the ability to predict performance sees a sudden "dip" at around 16 examples. Previous works relate this problem to the dual operating mode of LLMs, called *task recognition* (where the LLM recalls knowledge seen in pretraining) and *task learning* (where the LLM learns from the given demonstrations) (Lin & Lee, 2024; Pan et al., 2023). In our case, given that HANS is a difficult textual entailment tasks (67% acc. on Llama-13B), we hypothesize that the model operates in the second mode. While out of the scope of this work, we believe that a similar influence framework can be used to help study this phenomenon in more details.

Data support and known dimensions. We quantitatively analyze a few known metrics that could be associated with data support. These include example length, mean embedding distance from train set, perplexity, zlib entropy (Carlini et al., 2020), and Min-K% (Shi et al., 2023). The last two are designed to detect whether an example has been memorized from the pretraining data, with Min-K% employing the intuition that unseen example is likely to contain a more outlier words (K) with low probabilities under the LLM than a seen example.

From Table 2, we observe a positive correlations between data support for MBPP (code generation) and our memorization metrics. We speculate that Mistral-7B is likely to have seen duplicate or near-duplicates of these examples in the pretraining data, leading to them being resistant to change when train examples are used. In contrast, data support reveals consistently negative correlations to memorization for HANS, a task consisted of mundane words, but is carefully designed to challenge natural language understanding. We attribute

PC #	Direction	Subpop. representative	Subpop. summary
1	High	P: The artists next to the managers introduced the students. H: The managers introduced the students. Answer: non-entailment	***Output***: non-entailment (10/10) ***Heuristic***: subsequence (9/10) ***Template***: 38 (4/10) ***Mistral-7B/Llama2-13B*** : 10.5%/86.5%
	Low	P: The secretaries advised the professor near the author. H: The secretaries advised the professor. Answer: entailment	***Output***: entailment (10/10) ***Subcase***: lexical overlap around prepositional phrase (5/10) ***Template***: 30 (5/10) ***Mistral-7B/Llama2-13B*** : 99.3%/41.0%
2	High	P: The lawyer was contacted by the banker. H: The lawyer contacted the banker. Answer: non-entailment	***Output***: entail./non-entail. (5/10) ***Subcase***: constituent, embedded under preposition (5/10) ***Template***: 59 (5/10) ***Mistral-7B/Llama2-13B***: 15.1%/52.4%
	Low	P: The lawyer encouraged the athlete, or the artist supported the tourists. H: The artist supported the tourists. Answer: non-entailment	***Output***: non-entailment (10/10) ***Heuristic***: constituent (10/10) ***Subcase***: constituent, disjunction (6/10) ***Template***: 54 (5/10) ***Mistral-7B/Llama2-13B***: 97.1%/99.6%

Figure 3: Summary of target examples (subpopulation size 10) with highest and lowest values in PC #1, PC #2 obtained from decomposing $\Theta^{\text{Mistral-7B-7B}\setminus\text{Llama2-13B}}$ (Figure 4). Using features in HANS, i.e. template/heuristic/subcase, we identify and interpret subpopulations with clear distinctions. For instance, Mistral significantly outperforms Llama2-13B on questions with lexical overlap around prepositions–entailment (99.3% vs. 41.0%)

this to the fact that Mistral has to actually *learn* from its few-shot examples and employs useful inductive biases, which is different from memorization.

Train influences and known dimensions. Similarly, we aggregate the in-context influences by taking their mean over the train set and compare them with the same quantitative metrics. Results from Appendix Table 4 show little to no association between the influences and these metrics. One exception is the target distance on Amazon Reviews, which agrees with results from Figure 1 showing that Embedding is the best baseline for this task.

5 Compare Models via In-context Influences

In addition to finding minimal data support, we also demonstrate a useful application of ICL influences: language model comparison.

From training many g_θ, we can represent the entire target set as a feature embedding $\mathbb{R}^{|Target|\times|Train|}$ for a task. We find such representation to be meaningful, i.e. captures information about the class label, despite being relatively sparse compared to dense text embeddings (more discussion in Appendix Section A.2). In a case study, we focus on a specific application of our computed influence embeddings Θ through `ModelDiff` (Shah et al., 2023).

The main question asked is: How do Mistral-7B and Llama 2-13B behave differently on the task HANS, where they achieve comparable ICL performance (71% vs. 67% accuracy)? One data-centric solution is to find the subpopulations most distinguishable between two models, and interpret these data points. We follow closely the pipeline proposed by Shah et al. (2023): given $\Theta^A, \Theta^B \in \mathbb{R}^{|Target|\times|Train|}$ for models A and B, we first compute the residuals influence embedding $\Theta^{A\setminus B}$, apply PCA on it, and find the target examples with values highly aligned with the residual PC directions.

Figure 4 displays the results of decomposing the residual embeddings for differentiating our models of interest on HANS, a dataset with features that are highly interpretable. We reveal that Llama-13B significantly outperforms Mistral (e.g. 10.5% vs. 86.5%) on subsequence–non-entailment examples. In Appendix Table 8 & Table 10, we apply the same study to SBIC and AG News.

6 Related Works

Example selection. In independent work, Chang & Jia (2023) also studied the use of influences for selecting in-context examples for k-shot prompting They also find influence-based selection to outperform many baseline methods. While both consider influence estimates based on datamodels and data shapley influences, there are some distinct differences. Chang & Jia (2023) integrate the position of an in-context example into the datamodel to directly calculate the influence of position for each example. In contrast, we do not model the effect of example position. In use cases, Chang & Jia (2023) focus on a small number of in-context examples (i.e. $k = 4$) and for ICL stability, while we learn from a large number of examples (i.e. k up to 70) for quantifying brittleness and demonstrating various analyses. Related to our data support measure, Chen et al. (2022) find that good examples are less sensitive to change when perturbed compared to bad examples, which can be viewed as another brittleness quantification.

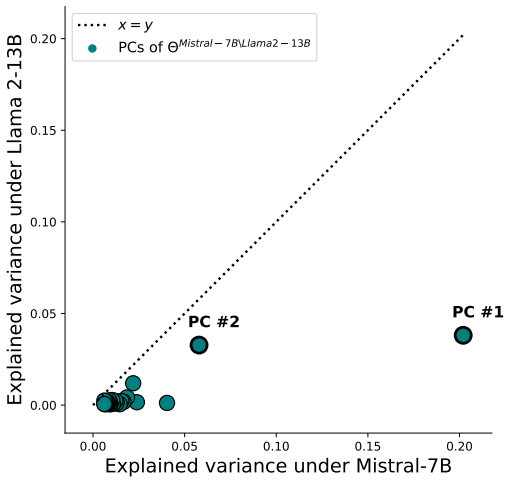

Figure 4: A point $v \in \mathbb{R}^{|Target|}$ represents a principal component (direction) that explains variance in residual Mistral-7B $\setminus$ Llama 2-13B.

In-context learning. ICL comes with high volatility to factors beyond example selection. In the few-shot setting, models have shown a tendency to overly rely on the most frequent labels (majority bias) or labels that appear at late positions in a prompt (recency bias) (Zhao et al., 2021). The latter suggests that the ordering of examples can be optimized for performance gain (Lu et al., 2022). Other findings have discovered that correct input-label mapping has little relevance (Min et al., 2022) and example diversity is more important (Su et al., 2022). Recently, many works have also linked the underlying computations of ICL to various algorithms, include linear regression (Akyürek et al., 2022).

Training data influence. Influence functions (Koh & Liang, 2017) have been used as a way to trace a model's output back to the training data. Influence of a specific training point measures the change in a model's performance when the point is removed from the training set. Data Shapley (Ghorbani & Zou, 2019) and Ilyas et al. (2022) measure similar quantities via retraining the model on subsets of the dataset. Outside of individual attributions, influence functions have also been used to measure group effects, where Koh et al. (2019) found influence estimates of individual data points to be the lower bound of groups.

7 Conclusion

Our work proposes in-context influences as a way to select and analyze examples for ICL. Our example selection scheme outperforms several baselines at estimating data support, the smallest subset that induce a label flip on any target example. We find data support to have associations with different memorization scores, and demonstrate how our influence framework can surface subpopulations for LLM comparison.

One limitation of influence-based frameworks is that they predict ICL performance from fixed train and target sets. However, practitioners often generate original prompts and examples, which may not exist in the training set. One potential research direction is to explore predicting the performance of inputs constructed on the fly, in addition to those in the training set. Moreover, future work could improve influence performance by exploring a relationship closer to how models learn in-context (instead of a linear assumption) and finding more efficient ways to compute them.

8 Ethics Statement

Our work proposes an influence framework for analyzing in-context learning and quantifying its brittleness. We acknowledge that our approach can be used to potentially induce undesirable and harmful behaviors out of LLMs. Thus, it should be applied with caution and keen awareness of risks and biases associated with LLMs and their applications. Additionally, although we solely employ open-access pretrained models, the nature of our work can be prohibitive as it requires a decent amount of computation and costs. This hinders the accessibility of such models and methods to a wider research community.

References

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models, 2022.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom B. Brown, Dawn Xiaodong Song, Úlfar Erlingsson, Alina Oprea, and Colin Raffel. Extracting training data from large language models. In *USENIX Security Symposium*, 2020.

Ting-Yun Chang and Robin Jia. Data curation alone can stabilize in-context learning. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8123–8144, Toronto, Canada, July 2023. Association for Computational Linguistics.

Yanda Chen, Chen Zhao, Zhou Yu, Kathleen McKeown, and He He. On the relation between sensitivity and accuracy in in-context learning, 2022.

Yanda Chen, Chen Zhao, Zhou Yu, Kathleen McKeown, and He He. Parallel structures in pre-training data yield in-context learning. *arXiv preprint arXiv:2402.12530*, 2024.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL-HLT*, 2019.

Amirata Ghorbani and James Y. Zou. Data shapley: Equitable valuation of data for machine learning. In *Proc. of ICML*, Proceedings of Machine Learning Research, 2019.

Xiaochuang Han, Byron C. Wallace, and Yulia Tsvetkov. Explaining black box predictions and unveiling data artifacts through influence functions. In *Proc. of ACL*, 2020.

Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. Surface form competition: Why the highest probability answer isn't always right. In *Proc. of EMNLP*, 2021.

Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. Datamodels: Predicting Predictions from Training Data, 2022.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L'elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *ArXiv*, abs/2310.06825, 2023.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pp. 1885–1894. PMLR, 2017.

Pang Wei Koh, Kai-Siang Ang, Hubert H. K. Teo, and Percy Liang. On the accuracy of influence functions for measuring group effects. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 2019.

Sawan Kumar and Partha Talukdar. Reordering examples helps during priming-based few-shot learning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 2021.

Ziqian Lin and Kangwook Lee. Dual operating modes of in-context learning, 2024.

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

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*, 2022b. ISSN 0360-0300. Just Accepted.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proc. of ACL*, 2022.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?, 2022.

Zach Nussbaum, John X. Morris, Brandon Duderstadt, and Andriy Mulyar. Nomic embed: Training a reproducible long context text embedder, 2024.

Jane Pan, Tianyu Gao, Howard Chen, and Danqi Chen. What in-context learning "learns" in-context: Disentangling task recognition and task learning. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 8298–8319, Toronto, Canada, July 2023. Association for Computational Linguistics.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

S. Robertson. The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.

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

Harshay Shah, Sung Min Park, Andrew Ilyas, and Aleksander Madry. Modeldiff: A framework for comparing learning algorithms. In *International Conference on Machine Learning*, pp. 30646–30688. PMLR, 2023.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. Detecting pretraining data from large language models, 2023.

Hongjin Su, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Tao Yu. Selective annotation makes language models better few-shot learners, 2022.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, 2023.

Jiacheng Ye, Zhiyong Wu, Jiangtao Feng, Tao Yu, and Lingpeng Kong. Compositional exemplars for in-context learning. 2023.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate before use: Improving few-shot performance of language models. In *Proc. of ICML*, Proceedings of Machine Learning Research, 2021.

A Appendix

We include additional results and details that do not fit in the main paper.

A.1 Performance metric for generation tasks

Recall that we require a suitable performance metrics to compute ICL influences (Section 2. Since inference on generation tasks inherently differs from how classfication is done, we need to adopt a different metric. Here, we define SEQ_SCORE(S', x) as the decay-weighted log-probability of the generated sequence. Specifically, given the prompt sequence S' and preceding tokens $t_{0:i-1}$, the sequence log-prob L is defined as:

$$L(S', x) = \frac{\sum_{i=1}^{n} \log P(x_i | S', t_{0:i-1}) \cdot d^{i-1}}{\sum_{i=1}^{n} d^{i-1}} \tag{4}$$

where n is the length of the generated sequence, $\log P(t_i | S', t_{0:i-1})$ denotes the log-probability of generating the i-th token, and d denotes a decay factor ($0 < d < 1$). We use $d = 0.9$ as the decay value, and assign the polarity depending on correctness:

$$\text{SEQ\_SCORE}(S', x) = \begin{cases} L(S', x), & \text{if answer is correct} \\ -L(S', x), & \text{otherwise} \end{cases} \tag{5}$$

A weighted decay accounts for the importance of each token by its position in the sequence, with tokens at the beginning assigned more worth and subsequent tokens downweighted. We employ the intuition that a correct answer is sometimes decided in the first few tokens for LLM generation.

A.2 Influence embedding

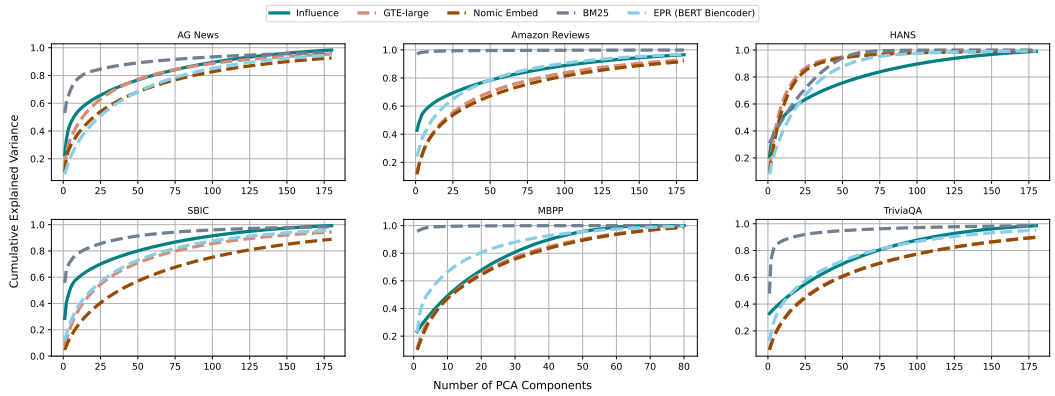

Figure 5: Influence embedding encodes rich information inline with dense text embeddings. Explaining 80% of the variance in the influence embeddings Θ requires more than 50 PCA dimensions for most tasks.

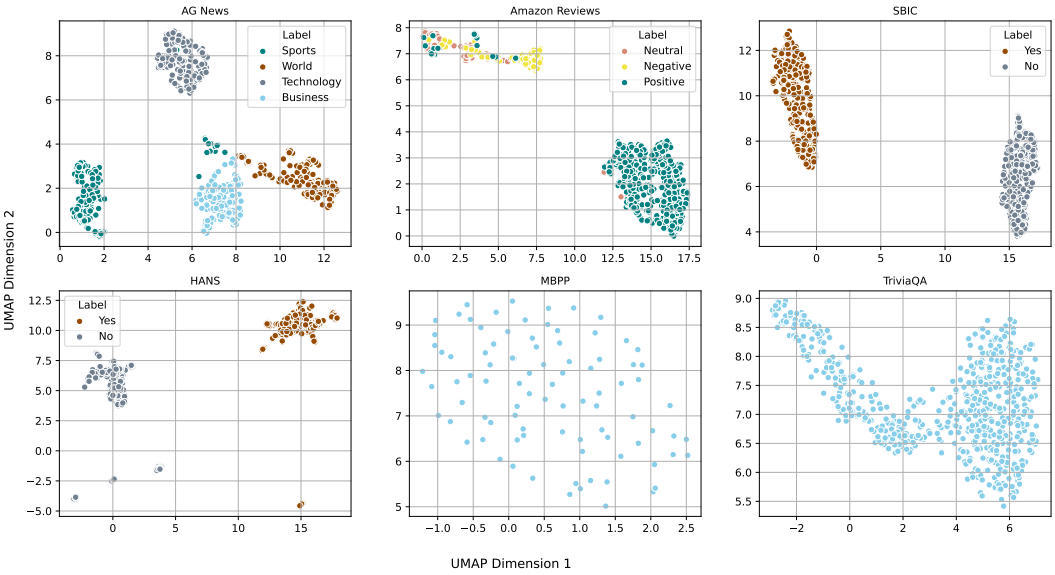

Figure 6: UMAP reduction of influence embeddings $\Theta \in \mathbb{R}^{|Target| \times |Train|}$. On classification tasks, they implicitly encodes class label information.

Our influence embeddings encode dense information comparable to dense text embeddings, including class label. We visualize this in Figure 5 and Figure 6.

A.3 Other details

Our models are loaded in half-precision (`fp16`) and use Flash Attention 2 in consideration for resources.

Hardware. We run all experiments on the NVIDIA A100 and NVIDIA RTX A6000 GPUs with 80GB and 48GB VRAM.

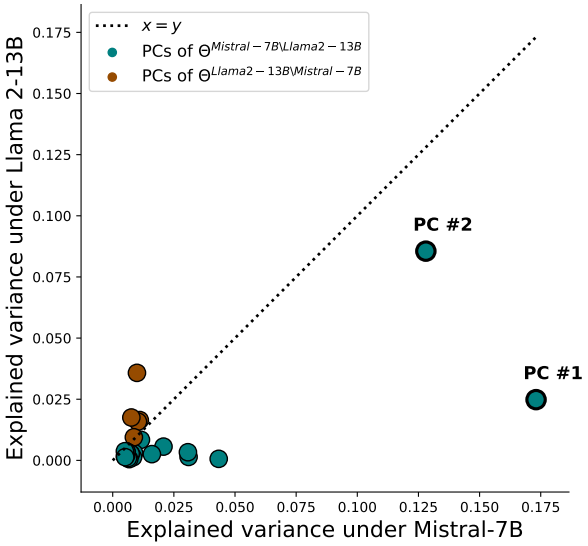

Figure 7: A point $v \in \mathbb{R}^{|Target|}$ represents a principal component (direction) that explains variance in residual Mistral-7B $\setminus$ Llama 2-13B for SBIC.

PC #	Direction	Subpop. representative	Subpop. summary
1	High	Post: My dad has always told me back in his day you had to walk uphill both ways... My generation has to walk through open fire. Answer: offensive	***Output***: offensive (10/10) ***Reason***: personal attack (8/13) ***Target***: conservatives/liberals (4/13) ***Mistral-7B/Llama2-13B*** : 6.3%/7.3%
	Low	Post: Main White House leakers: the sources he taped, I'm told, are Steve Bannon and former White House deputy chief of staff Katie Walsh. Answer: non-offensive	***Output***: non-offensive (10/10) ***Reason***: post does not imply anything offensive (10/10) ***Mistral-7B/Llama2-13B*** : 98.7%/95.4%
2	High	Post: What did Jar Jar Binks say to the retired carpenter? "Meesa-thelioma" Answer: non-offensive	***Output***: non-offensive (10/10) ***Reason***: post does not imply anything offensive (10/10) ***Data support***: 5.9 ***Mistral-7B/Llama2-13B***: 1.6%/34.0%
	Low	Post: What's the worst part about breaking up with a Japanese girl?[REDACTED] Answer: offensive	***Output***: offensive (10/10) ***Reason***: race-based attacks (6/10) ***Data support***: 19.9 ***Mistral-7B/Llama2-13B***: 99.8%/66.0%

Figure 8: [Figure might contain offensive and sensitive material.] Summary of target examples (subpopulation size 10) with highest and lowest values in PC #1, PC #2 obtained from decomposing $\Theta^{\text{Mistral-7B-7B}\setminus\text{Llama2-13B}}$ (Figure 7). Using explanations in SBIC, we identify and interpret subpopulations with clear distinctions, including the label dominance, feature dominance, data support, and performance difference (accuracy) between few-shot Mistral-7B and Llama2-13B.

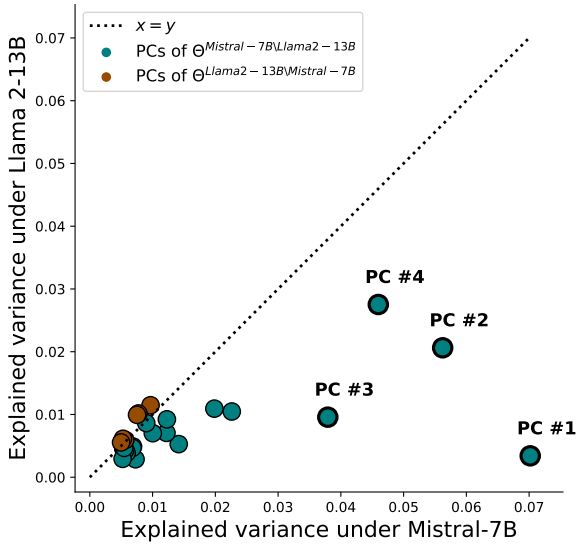

Figure 9: A point $v \in \mathbb{R}^{|Target|}$ represents a principal component (direction) that explains variance in residual Mistral-7B \ Llama 2-13B for AG News.

PC #	Direction	Subpop. representative	Subpop. summary
1	High	Article: FedEx CEO to lead industry security task force FedEx CEO Frederick W. Smith was named today by the Business Roundtable to chair the group's security task force. Answer: Technology	***Output***: Technology (10/10) ***Keywords***: technology, computer, say ***Data support***: 5.75 ***Mistral-7B/Llama2-13B*** : 67.7%/71.2%
	Low	Article: Google 3rd-Qtr Profit More Than Doubles on Web Advertising ... $52 million after sales of Web advertising rose. Answer: Business	***Output***: Business (10/10) ***Keywords***: business, symantiec, plan ***Data support***: 11.8 ***Mistral-7B/Llama2-13B*** : 51.7%/51.8%
4	High	Article: Sprint Sinks $3 Billion into Wireless Network quot;Mobile operators are rolling these things out because they have nothing better to do, quot; said Ken Dulaney, Gartner #39's vice president of mobile computing. Answer: Business	***Output***: Business (5/10) ***Keywords***: business, technology ***Data support***: 4.9 ***Mistral-7B/Llama2-13B***: 11.5%/31.0%
	Low	Article: Loblaw Profit Rises 19 as Lederer Fends Off Wal-Mart (Update1) Loblaw Cos. said third-quarter net income rose 19 percent as the company, Canada #39's largest supermarket chain, cut distribution costs and sold more of its profitable nonfood goods. Answer: Business	***Output***: Business (7/10) ***Keywords***: business, say, game ***Data support***: 15.4 ***Mistral-7B/Llama2-13B***: 89.0%/99.6%

Figure 10: Summary of target examples (subpopulation size 10) with highest and lowest values in PC #1, PC #2 obtained from decomposing $\Theta^{\text{Mistral-7B-7B}\setminus\text{Llama2-13B}}$ (Figure 9).

	length	distance$_{target}$	ppl	ppl/zlib	min-K$_5$	min-K$_{30}$	min-K$_{50}$
AGN	-0.00	0.02	0.00	0.00	0.03	0.01	0.00
Amazon	-0.07	-0.12	0.03	0.05	0.02	0.03	0.04
HANS	-0.01	-0.03	0.01	0.02	0.02	-0.00	0.01
SBIC	-0.06	-0.03	-0.03	0.01	-0.04	-0.03	-0.03
MBPP	0.04	-0.08	-0.01	-0.04	0.02	-0.01	-0.01
TrivQA	-0.03	-0.04	0.04	0.05	-0.03	0.03	0.04

Table 3: Little correlation (Pearson) is found between train example in-context influences and the following dimensions: length, embedding distance from the target set, perplexity, perplexity/zlib, and different memorization scores from Min-K% (Shi et al., 2023).

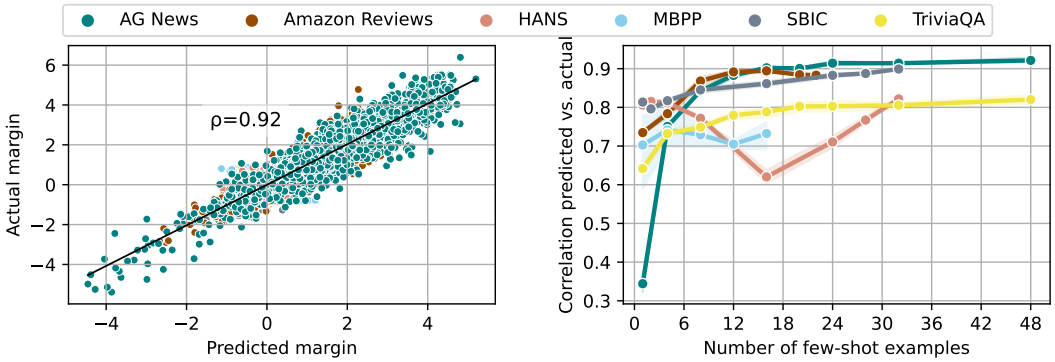

Figure 11: **Left:** Correlation between predicted and actual margins using in-context influences on Llama-13B. **Right:** Predictability scales well with increasing number of shot, with HANS seeing a "dip" at 16 shots.

	Type	Domain	|Train|	|Target|	k
AG News	Classification (multi)	Topic classification	1500	600	50
Amazon Reviews	Classification (multi)	Sentiment analysis	1500	600	22
HANS	Classification (binary)	Textual entailment	1500	600	70
SBIC	Classification (binary)	Toxicity	1500	600	70
TriviaQA	Generation	Fact recall	1500	600	68
MBPP	Generation	Code	374	90	16

Table 4: Tasks used in the paper, with the size of the subset S for collecting training runs on language models with 4096 context window size.

Task	Prompt template
AG News	Article: {`article`} Answer: {}
Amazon Reviews	Review: {`text`} Answer: {}
HANS	Premise: {`premise`} Hypothesis: {`hypothesis`} Given the premise, can we conclude the hypothesis? Yes or No? Answer: {}
SBIC	Post: {`post`} Is the post offensive? Yes or No? Answer: {}
MBPP	Problem: {`question`} Script should pass the following test examples: {`test_list`} Answer: {}
TriviaQA	Q: {`question`} A: {}

Figure 12: Prompt templates for tasks used in this work