

Learning vs Retrieval: The Role of In-Context Examples in Regression with Large Language Models

Aliakbar Nafar¹, Kristen Brent Venable^{2,3}, Parisa Kordjamshidi¹

¹Michigan State University

²Florida Institute for Human and Machine Cognition

³University of West Florida

{nafarali, kordjams}@msu.edu, bvenable@ihmc.org

Abstract

Generative Large Language Models (LLMs) are capable of being in-context learners. However, the underlying mechanism of in-context learning (ICL) is still a major research question, and experimental research results about how models exploit ICL are not always consistent. In this work, we propose a framework for evaluating in-context learning mechanisms, which we claim are a combination of retrieving internal knowledge and learning from in-context examples by focusing on regression tasks. First, we show that LLMs can solve real-world regression problems and then design experiments to measure the extent to which the LLM retrieves its internal knowledge versus learning from in-context examples. We argue that this process lies on a spectrum between these two extremes. We provide an in-depth analysis of the degrees to which these mechanisms are triggered depending on various factors, such as prior knowledge about the tasks and the type and richness of the information provided by the in-context examples. We employ three LLMs and utilize multiple datasets to corroborate the robustness of our findings. Our results shed light on how to engineer prompts to leverage meta-learning from in-context examples and foster knowledge retrieval depending on the problem being addressed.

1 Introduction

The emergence of transformers (Vaswani et al., 2017) has revolutionized natural language processing, leading to the development of LLMs such as GPTs (Brown et al., 2020). In addition to their impressive zero-shot performance, these models demonstrated the capability of in-context learning (ICL), by which they learn a task from examples provided in the context of the prompt (Brown et al., 2020). In tasks where both zero-shot and ICL settings are applicable, ICL consistently outperforms the zero-shot setting (Brown et al., 2020; Liu et al.,

2022). However, the inner mechanism of ICL in LLMs remains an open area of research.

Broadly speaking, current research identifies two main approaches to explain the ICL mechanism (Dong et al., 2024; Li et al., 2024): 1) Meta-learning, 2) Knowledge retrieval. With meta-learning (Schmidhuber, 1987), the LLM **learns** a new **unseen** pattern from the in-context examples, allowing it to generalize to new inputs that follow the same pattern. In contrast, with knowledge retrieval, the LLM uses the in-context examples as cues to access and apply information already stored within its pre-trained parameters without inferring new patterns from the provided examples. For instance, given examples of input/output number pairs in the context, meta-learning allows the model to learn their relationship and apply it to new inputs. However, if the prompt informs the model that the pairs are the earth’s population given a year, the LLM utilizes its knowledge retrieval and infers that the function is exponential and plateaus at around 8-10 billion after the year 2000.

Contrary to current research, we propose a different approach, arguing that ICL is not merely learning or retrieving knowledge, but rather, its behavior lies on a *spectrum* between the two, which can be adjusted depending on various factors. Before exploring our proposed approach, we elaborate on the hypotheses mentioned above.¹

The first hypothesis suggests that transformers are effective meta-learners, and LLMs can generate accurate predictions based solely on the given input-output pairs. (Bai et al., 2023) theoretically proves and practically tests the capability of an encoder-based transformer to implement generalized linear models in-context. (Garg et al., 2022) uses a decoder-based model, a GPT-2 architecture modified for regression, that uses the input features inside the embeddings (instead of using to-

¹ See Appendix A for a detailed discussion on related work.




Configuration	In-context Examples			Query
a) 	Fuel Economy: 16 Mileage: 0 Passenger Car: 1 Price: 95595	Fuel Economy: 16 Mileage: 34028 Passenger Car: 0 Price: 21700	... Fuel Economy: 13 Mileage: 0 Passenger Car: 0 Price: 35200	Fuel Economy: 28 Mileage: 37405 Passenger Car: 1 Price: ?
b) 	Feature 1 _{ny} : 16 Feature 2 _{ny} : 0 Feature 3 _{ar} : 1 Output _{ny} : 95595	Feature 1 _{ny} : 16 Feature 2 _{ny} : 34028 Feature 3 _{ar} : 0 Output _{ny} : 21700	... Feature 1 _{ny} : 13 Feature 2 _{ny} : 0 Feature 3 _{ar} : 0 Output _{ny} : 35200	Feature 1 _{ny} : 28 Feature 2 _{ny} : 37405 Feature 3 _{ar} : 1 Output _{ny} : ?
c) 	Fuel Economy: 16 Mileage: 0 Passenger Car: 1 Price: $\mathcal{N}(\mu, \alpha)$	Fuel Economy: 16 Mileage: 34028 Passenger Car: 0 Price: $\mathcal{N}(\mu, \alpha)$... Fuel Economy: 13 Mileage: 0 Passenger Car: 0 Price: $\mathcal{N}(\mu, \alpha)$	Fuel Economy: 28 Mileage: 37405 Passenger Car: 1 Price: ?

Figure 1: The three main prompt configurations: In configuration a) the actual names of the features and the output are known, and the LLM is asked to guess the “price of a used Toyota or Maserati in 2019”. Configuration b) is similar to a) except that the feature names are anonymized. Here, the LLM is asked to estimate the “Output”. In Configuration c), we replace the real prices of in-context examples with randomly (Gaussian) generated numbers.

kens) and outputs a number. Their model’s performance surpasses a 2-layer Multi-Layer Perceptron (MLP) and a decision tree by meta-learning. Further, (Vacareanu et al., 2024) directly uses LLMs to test regression capabilities using a limited number of features (less than 3) and concludes that LLMs are capable regressors based on in-context examples. However, according to our findings, their claim of avoiding data contamination is not well-supported. We note that none of these research works use realistic datasets or consider the combination of meta-learning with knowledge retrieval.

The second approach emphasizes knowledge retrieval while downplaying the learning aspect. For example, (Min et al., 2022) examines 12 LLMs across various classification tasks, concluding that altering output labels, which are needed for learning, has no impact on performance. They argue that the crucial elements of ICL are limited to defining the label space, input distribution, and overall task format. Meanwhile, (Kossen et al., 2024) argues that LLMs utilize the labels but in an arbitrary manner when experimenting with the same classification tasks. (Li et al., 2024) further dismisses the significance of output labels and instead proposes two knowledge retrieval approaches. These include the retrieval of pre-existing solutions from training data (Min et al., 2022; Wang et al., 2023; Wies et al., 2023), and a novel *solution composition* approach, inspired by (Hahn and Goyal, 2023), which proposes that learned solutions can be combined. While this work offers valuable insights into ICL mechanisms, its findings are limited to a narrow selection of datasets and models. Here, the

exclusive use of LLaMA 2 (Touvron et al., 2023), which is known to struggle with long token contexts (Machlab and Battle, 2024; Zuhashaik et al., 2023) results in some different findings compared to our work. For instance, they report that remapping the inputs of in-context input-output pairs to alternative text reduces the accuracy to that of a random model (even with up to 40 in-context examples). On the other hand, under similar conditions, our experiments indicate that in newer models, learning occurs with optimal performance. We refrain from making specific claims about task selection or composition in our work. Instead, we categorize these processes under the broader umbrella of *knowledge retrieval*². Similarly, to investigate ICL, (Pan et al., 2023) uses simple classification tasks and older models (e.g., GPT-3 Ada), and as a result, its findings are partially different from ours.

We propose a different hypothesis that resolves the contradictions in the research community and support our claims with extensive empirical testing. We argue that ICL is not merely learning or retrieving knowledge but uses a combination of the two, which lies on a spectrum determined by factors we can manipulate. In this regard, we propose an evaluation framework and conduct a comparative study of different LLMs and datasets, focusing specifically on regression problems as our testing ground. In our evaluation framework, we query the LLM to estimate an output number based on a set of (feature, value) pairs given a set of ICL examples, as shown in Figure 1(a).

²See Appendix B for a more detailed analysis of the discrepancies between our findings and those of (Li et al., 2024).

We opted for regression tasks for the following reasons: This choice aligns our work with the majority of related ICL meta-learning research, thus facilitating direct comparisons and building upon existing findings. These include works that alter the output module of LLMs, such as (Garg et al., 2022) and those that use the token generation output of LLMs as is, such as (Vacareanu et al., 2024). Moreover, (Bhattamishra et al., 2024) demonstrates that the ICL capability of LLMs is independent of their input/output modules and is present in the intermediary layers of the LLMs. Finally, while LLMs are capable of regression (even when inputs are given to them in a textual format (Vacareanu et al., 2024)), this task still has a complex output space (e.g., continuous or unbounded), which is challenging for LLMs (Fang et al., 2024).

We show that LLMs can perform regression on realistic datasets and measure the extent to which the LLM retrieves its internal knowledge versus learning from in-context examples. We provide an in-depth analysis of the degrees to which these mechanisms are triggered depending on the factors we use in our framework: the number of (feature, value) pairs, the number of in-context examples, and the prompting strategies.

In summary, our contributions are as follows: **1)** We demonstrate that LLMs can effectively learn from regression examples of **realistic** datasets in-context, extending previous work on synthetic data to more practical scenarios. **2)** We propose a new hypothesis about the ICL mechanism that combines both learning and knowledge retrieval during the LLM inference, reconciling the research community’s findings. **3)** We introduce an evaluation framework that allows for systematic comparison of ICL mechanisms across different LLMs, datasets, and prompt configurations³. **4)** We provide a comprehensive analysis of how LLMs balance internal knowledge retrieval and learning from in-context examples and propose prompt engineering techniques to control them.

2 Problem Setting

Our study focuses on a regression task where we use LLMs to predict numerical outputs based on the given inputs. We assume a regression dataset, $D = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, comprising input-output pairs, is given. Each x_i comprises a set of pairs (f_{ij}, v_{ij}) where f_{ij} represents the fea-

ture name and v_{ij} is the numerical value of f_{ij} . The target variable y_i is the numerical value of the output. For example, if we consider a dataset providing the price of used cars given their fuel economy and mileage, we could have an input with (feature, value) pairs, (Fuel Economy, 16) and (Mileage, 0), and an output, 95595, indicating the price.

To resemble a regression problem setting, we present the LLM with a subset of m input-output examples $\{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$ taken from the in-context split of D for learning. Then, we query the LLM with the feature pairs of x taken from the test split of D to obtain the answer y as the output.⁴ Figure 1(a) depicts our problem setting in the context of a regression task that involves predicting the “price of a used Toyota or Maserati in 2019”, which is the target variable name, given three (feature, value) pairs with a set of provided in-context examples. When m equals 0, we obtain the zero-shot setting without prior examples.

2.1 Prompt Configurations

We employ three main prompt configurations for prompting the models, as shown in Figure 1. These configurations vary in different ways, such as hiding the real name of features (Figure 1(b)) or the actual ground truth (Figure 1(c)). We also prompt the LLM with simple numerical generation as a baseline in our experiments. This results in a total of four prompt configurations. We also consider other prompt configurations which either served as ablations or didn’t provide any significant insights. In each prompt configuration, the context of the prompt comprises a task instruction, in-context examples (sample regression task input-outputs), and a query. The task instruction asks the LLM to estimate the objective of the dataset (such as the price of a used car) based on the given features by **just providing a number and no explanation as the output**. Each of these parts can be changed or removed according to the prompt configuration.

Named Features (Configuration a): This is our most straightforward prompt configuration. It reveals the actual names of the features and the required target variable (depending on the dataset) in the prompt. For instance, in the case of a dataset about the price of used cars, the LLM is instructed to estimate a “Used Car Price” based on “City fuel Economy”, “Mileage” and “Passenger Car Classifi-

³The code and analytical tools are available at [🔗](#).

⁴We note that since we do not change the architecture of the LLMs, all numbers must be represented to and read by the LLM in textual form.

cation”. Following this instruction, sample input-output examples are given to the model, and then the query is asked as shown in Figure 1(a).

Anonymized Features (Configuration b): In this setup, which is shown in Figure 1(b), we present the LLM with input-output pairs where the actual name of the features and the target variable are changed to “Feature #” and “Output”, respectively. The instruction asks the LLM to estimate the “Output” based on the given “Features”. Here, the LLM can not use its domain knowledge, and **can only** use the provided number of the features.

Randomized Ground Truth (Configuration c): In our final main prompt configuration shown in Figure 1(c), we maintain the named features but replace the ground truth values with randomly generated numbers (independent of the feature values). These random numbers are generated using a Gaussian distribution based on the dataset statistics. This setup mainly serves as a control configuration to test to what degree the LLMs are truly learning from the provided ground truth in the examples.

Direct Question Answering (Direct QA): To establish an LLM baseline, we ask LLMs to estimate the target variable based on the given named features without any in-context examples ($m = 0$). We also define the scope of the output by providing the mean and the standard deviation of the questioned dataset in the instruction to the LLM. An example of this added information is: “Estimate the insurance cost of this person given the information. An issuance cost is typically around 13270.42 with a standard deviation of 12110.01”.

2.2 Models and Metrics

We evaluate several LLMs, including LLaMA 3 70B (AI@Meta, 2024), GPT3.5 (Brown et al., 2020), and GPT4 (OpenAI, 2023). We initially considered smaller LLMs such as Mistral 7B (Jiang et al., 2023), but found their performance for regression tasks to be insufficient and consequently excluded them. To provide a comprehensive comparison, we also employed classical machine learning techniques alongside these LLMs. Specifically, we utilized a straightforward Ridge regression (Hoerl and Kennard, 1970) and the more advanced RandomForest model (Breiman, 2001). Further details about these models, such as their hyper-parameters, are included in Appendix D.

In our analysis, for each dataset, we test the LLMs with a mix of **factors** (we call these factors to distinguish them from the features f_{ij} de-

finied in our problem setting). The first factor is the *prompt configurations*, which can be Named Feature, Anonymized Features, Randomized Ground Truth, or Direct QA. The second and third factors are the *number of in-context examples* and the *number of (feature, value) pairs*, which we will refer to as the number of features. The number of in-context examples is 0, 10, 30, and 100, and the number of features is 1, 2, and 3 in our experiments. These feature numbers indicate using the first feature (F1), the first two features (F2), and all three features (F3) (features are ordered and sorted in decreasing order of importance). Not all these factors can be used together as the 0 in-context examples factor can only be used with the Direct QA prompt configuration, and other prompt configurations can not be used with the 0 in-context examples factor. To assess the performance of the LLMs and the machine learning models on regression tasks, we use Mean Squared Error (MSE) as our main evaluation metric. We also calculate and report the coefficient of determination R^2 and Mean Absolute Error (MAE) in Appendix E.

3 Experiments

This section presents our experimental findings, highlighting the interplay between various factors that influence the performance of LLMs across regression tasks and datasets.

3.1 Datasets

To assess the LLMs’ performance across diverse domains and complexity levels, we select three regression datasets and three of their most important features, where feature importance was calculated with RandomForest (Louppe et al., 2013): 1) Admission Chance: the *likelihood of admission to graduate programs for Indian students* (Acharya et al., 2019) with three highly correlated features. Since this dataset is about Indian students, there is a lower chance of being observed in the LLM’s training data, reflecting the general imbalance that skews towards USA-centric data (Zhu et al., 2023). 2) Insurance Cost: focuses on predicting a similar distribution of the *annual individual medical costs billed by health insurance companies in the USA*, drawing from demographic statistics compiled by the U.S. Census Bureau (Lantz, 2013). For this dataset, the first feature is the most important one by far. 3) Used Car Prices: the estimation of the *price of a used Toyota or Maserati car in 2019*.

Here, the first two features are the most impactful ones.⁵ These datasets are pre-processed to select their most important features. We divide each dataset into two splits: an in-context subset comprising 100 instances and a test subset with 300 instances. See Appendix C for details.

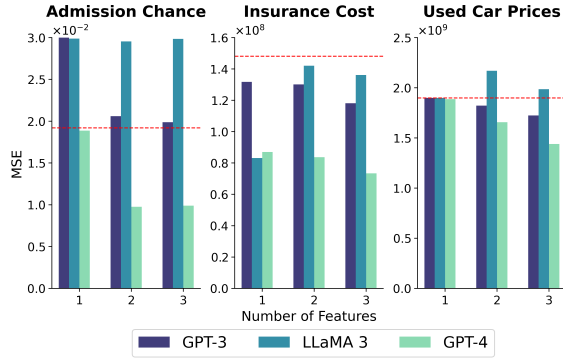


Figure 2: Baseline results (Direct QA configuration) across datasets and number of features. The dashed red line shows the performance of the Mean model.

3.2 Knowledge Retrieval Assessment

To establish a baseline for our subsequent analyses, we first evaluate the LLMs’ performance using only their knowledge retrieval, without any in-context examples. This assessment is required to understand how the in-context examples in other prompt configurations modulate the LLMs’ performance and ICL mechanisms. The performance of LLMs with *Direct QA* prompt configuration is shown in Figure 2. The red dashed line shows the performance of Mean model, which outputs the mean of the dataset independent of the input features. As can be observed, the **additional features generally improve the performance** across tasks (with inconsistencies when using LLaMA 3). However, the LLMs’ utilization of features does not directly correlate with feature importance, as explained in Section 3.1. As for the impact of datasets, the Admission Chance dataset consistently shows the poorest results, with most outcomes at or above the Mean model’s MSE. As mentioned earlier, this dataset is the least exposed to LLMs, which likely explains these findings.

We further explored the Direct QA approach by asking the LLM to explain its reasoning based on the given features before providing the final estimate. This variation yielded predictions compara-

ble to the Direct QA results without offering additional insights and consequently was excluded from our main prompt configurations. See Appendix F for example answers and related diagrams.

3.3 Learning/Knowledge Retrieval Interplay

We have defined four main prompt configurations to vary the degree to which the model uses its knowledge retrieval or learning from the in-context examples. By comparing the performance of these prompt configurations, we can understand how LLMs utilize knowledge retrieval and learning from the outputs. Figure 3 compares these prompt configurations’ effects on LLMs. The *Randomized Ground Truth* prompt configuration, shown with the lime color, consistently yields the worst results. Note that the names of the features are revealed here. As a result, this prompt configuration creates a scenario where patterns in the data may contradict the model’s internal knowledge. For instance, in the Insurance dataset, some in-context examples show that people who smoke less, require lower insurance costs, while others indicate higher costs for the same group. The negative impact of this prompt configuration on performance becomes more significant as the number of in-context examples with random outputs increases. This is particularly evident when using 100 in-context examples. These results suggest that the LLMs are using the output variables to learn from the examples, and increasing the number of **in-context examples shifts the spectrum from knowledge retrieval to learning**. Our findings, which indicate that LLMs can be pushed to prioritize learning from the in-context examples over knowledge retrieval, challenge the uselessness of the outputs claimed by (Min et al., 2022; Li et al., 2024).

Comparing the *Named Features* and *Anonymized Features* prompt configurations shows the power of combining the two paradigms of learning from in-context examples and knowledge retrieval. Anonymized Features prompt configuration, shown with the green color in Figure 3, allows only usage of the numeric part of the features for learning. This prompt yields better results than the Direct QA and the Mean model. Named Features prompt configuration, which adds the actual name of the features to the examples, is shown in purple. By encouraging the use of knowledge resulting from the added names of the features, it consistently outperforms Anonymized Features prompt configuration across variations of the factors, that is the number

⁵All datasets used in this study are publicly available. For references and details of our pre-processing code, visit [\[link\]](#).

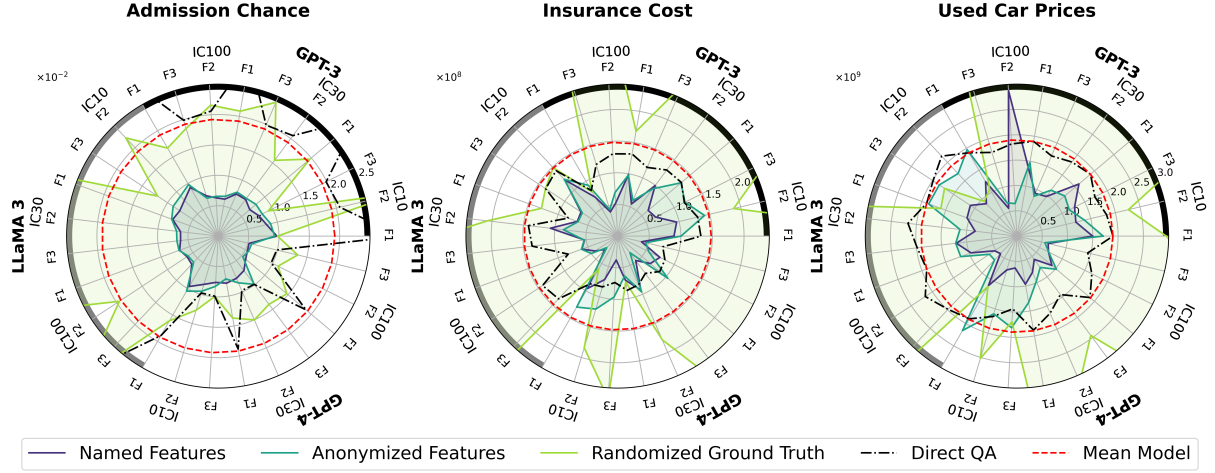


Figure 3: Comprehensive comparison of prompt configurations’ effects on our models across various factors, shown in a hierarchy. The top level for each dataset distinguishes between GPT-3, LLaMA 3, and GPT-4 results using black, grey, and white arcs, respectively. The notation IC_i indicates the number of in-context examples, while F1, F2, and F3 represent the use of the first feature, the first two features, and all three features, respectively. The MSE scale of each dataset is shown at top left corner.

of in-context examples and the number of features. These results show that LLMs can exploit **both in-context examples’ outputs for learning and clues like feature names for knowledge retrieval**. Note that when we replaced the feature names with random, unrelated names with the same ranges as the original features (e.g., replacing Smoker Status with Married since both are binary), the results remained the same as those of the Anonymized Features prompt configuration.

3.4 Knowledge Retrieval Compensates for In-context Examples

As previously noted, the *Named Feature* prompt configuration, which combines knowledge and learning, generally outperforms the *Anonymized Feature* prompt configuration, which relies only on learning. Regarding these prompt configurations, analyzing our defined factors, such as the number of in-context examples, reveals more insights about ICL. Conventionally, in the realm of LLMs, more in-context examples tend to improve the task outcome. However, as demonstrated in Figure 3, performance deteriorates with more in-context examples when outputs are random, even if the model possesses knowledge about the subject. Figure 4, which compares the performance of different numbers of in-context examples, reveals that the Named Feature prompt configuration performs better when fewer in-context examples are used (solid vs dashed lines). A significant performance gap between the two prompt configurations

is observed at 10 in-context examples, shown with the solid and dashed purple lines. However, at 100 in-context examples, shown with the solid and dashed lime lines, the performance levels converge. This underscores the potential for **reducing the number of required in-context examples** by providing task-specific information, thereby shifting the spectrum from learning to knowledge retrieval.

Moreover, Figure 5 shows the advantage of the Named Features prompt configuration over both the Anonymized Features prompt configuration and also the traditional machine learning models by comparing them across various numbers of in-context examples. With fewer in-context examples (30 and especially 10), Named Features prompt configuration models generally outperform all other models. Their results indicate that the type of information in the LLMs’ prompt can encourage exploiting their internal knowledge. Moreover, the LLMs’ capability to **retrieve knowledge makes them more data-efficient, i.e., few-shot learners, than even classical machine learning models**, RandomForest and Ridge. However, this advantage applies primarily in a low-data regime and does not account for issues such as the higher training and execution costs of LLMs, which we discuss further in Section 4. All the models eventually converge at 100 examples when sufficient data for regression is provided.

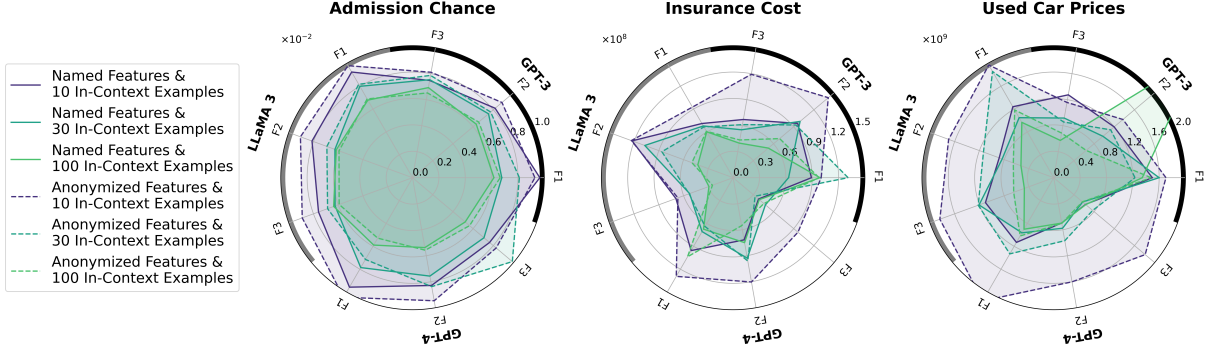


Figure 4: Comparison of the number of in-context examples using Named Features (straight lines) and Anonymized Features (dashed lines) prompt configurations. F1, F2 and F3 indicate using 1st (F1), then 1st and 2nd (F2), and all three features (F3). The MSE scale of each dataset is shown at the top left corner.

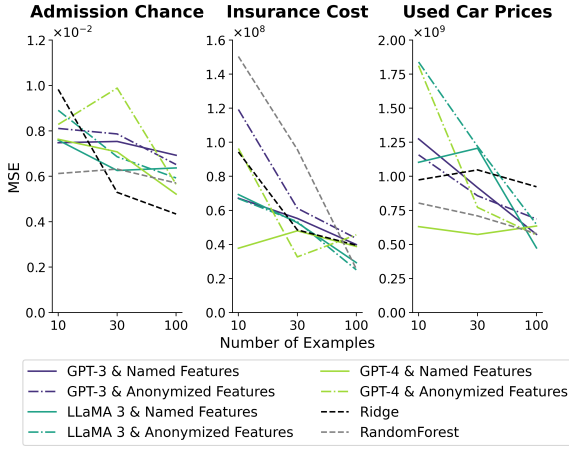


Figure 5: Performance of Named Features and Anonymized Features prompt configurations, Ridge, and RandomForest for 3 features based on the number of in-context examples.

3.5 More Features Help Knowledge Retrieval

Unlike the number of in-context examples, the number of features stands out as a unique factor that can improve **both learning and knowledge retrieval** in LLMs. As additional features are incorporated into the context, LLMs can better learn the relationships between features and outputs and also retrieve more relevant knowledge. This clearly helps in the case of Named Features prompt configuration.

However, the less obvious insights in this section emerge from the *Anonymized Features* prompt configuration, which relies only on learning and consistently improves with the addition of features. As shown in Figure 6 for 100 in-context examples (the dotted lines), each additional feature either improves the results or maintains near-optimal performance. A similar, but less robust, trend is seen in the 30 in-context examples’ diagram in Ap-

pendix F. While adding more features improves results, this improvement is more significant in language models compared to machine learning models such as Ridge and RandomForest in Figure 6. Given that in this prompt configuration 1) feature names are anonymous, 2) only raw numerical data is available for output estimation, and 3) the improvement in the **performance of LLMs does not correlate with the importance of the features**, we conjecture that the results could be influenced by **data contamination from the observed numbers**. This assertion is supported by the fact that this phenomenon is absent in the Admission Chance dataset, the least likely seen dataset by LLMs.

In the Named Feature prompt configuration for 100 in-context examples depicted in Figure 6, the results show an overall improvement, yet the trend is not as consistently downward or smooth as observed in the Anonymized Features prompt configuration when features are added. This fluctuation suggests that the knowledge retrieval aspect becomes more prominent and occasionally supersedes the learning component as the number of features increases. Based on these observations that happen at a high number of 100 in-context examples, we hypothesize that adding features **primarily enhances the knowledge retrieval aspect rather than contributing to the learning**.

3.6 Quantitative Analysis

Following (Yoo et al., 2022), we introduce a quantitative metric to assess the impact of knowledge on model performance. We denote the rate of improvement of a prediction after the addition of knowledge (named features) as *Knowledge Effect Ratio* (KER), which can be calculated as such:

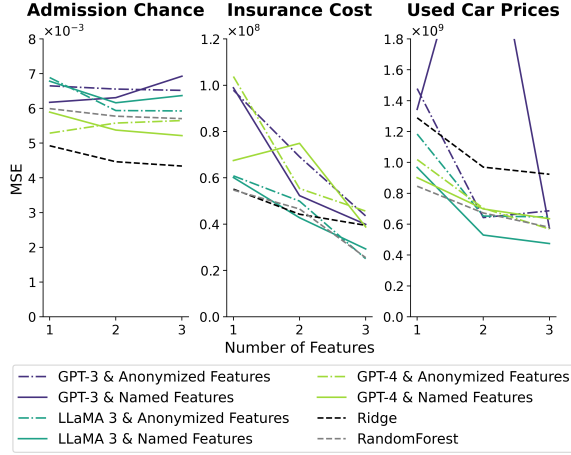


Figure 6: Performance of Anonymized Features prompt configuration, Ridge, and RandomForest for 100 in-context examples based on the number of features.

$$\text{KER} = \frac{|Y_{AF} - Y_{GT}| - |Y_{NF} - Y_{GT}|}{|Y_{AF} - Y_{GT}|} \times 100 \quad (1)$$

where Y_{AF} shows a prediction from the Anonymized Features Configuration, Y_{NF} shows a prediction from the Named Features Configuration and Y_{GT} is the ground truth value. This formula measures the rate of change in error (Mean Absolute Error) due to added knowledge in learning. We compute the KER for each dataset and combination of factors and then select its median improvement rate to mitigate the influence of outliers. As expected, this rate remains near zero across almost all factor combinations for the Admission Chance dataset, given that it does not benefit from knowledge in our tests. For the other two datasets, Figure 7 shows the average improvements across feature factors for each dataset and the number of in-context examples. Notably, the most significant gains emerge when the number of in-context examples is small, aligning with our earlier observations.

4 Discussion

Our evaluation framework and extensive experiments provide insights into the ICL mechanisms of LLMs. In this section, we discuss the implications of our findings, address limitations, and suggest directions for future research.

4.1 Controlling ICL Mechanisms

Understanding and manipulating ICL mechanisms in LLMs proves invaluable for practical applications and effective prompt engineering. Our exper-

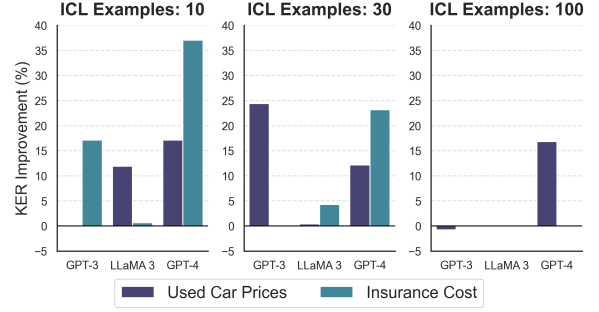


Figure 7: KER improvement across different models and datasets with varying numbers of in-context examples. The improvements become more notable as the number of in-context examples decreases.

iments show that the LLMs use both learning from the in-context input-output examples and prior information with varying extents. We found that increasing the number of in-context examples encourages learning while adding more features **mainly boosts knowledge retrieval**. However, these benefits only materialize when there is room for improvement in the respective mechanism. For example, using more than 100 in-context examples did not help LLMs, and in the Admission Chance dataset, where the features were highly correlated, the addition of features did not help the performance either. This insight explains why (Min et al., 2022)’s tasks appeared not to utilize output labels for learning; In their tasks, the dominant knowledge retrieval aspect likely eclipsed any learning benefits. Our findings also challenge the view of LLMs as merely meta-learners. While previous research demonstrates decoder models’ capacity for meta-learning, the complex interplay of different training objectives (unknown in some proprietary models) and other factors introduce a significant knowledge retrieval component. Consequently, earlier findings can only be directly extrapolated to LLMs when considering all these dynamics.

4.2 Practical Applications

Our findings emphasize the importance of striking an optimal balance between the number of meaningful features and the number of in-context examples to optimize LLMs performance. By strategically reducing the quantity of in-context examples while increasing the number of named features, it is possible to achieve resource efficiency and other potential benefits, such as mitigating data biases, without compromising performance. Conversely, when tackling tasks unfamiliar to the model, prun-

ing less important features can free up space in the LLM’s token context, allowing it to accommodate more in-context examples. This shifts the ICL mechanism towards the learning side of the spectrum and improves the model’s performance.

4.3 Traditional ML Models vs LLMs

Our findings show that LLMs can be more data-efficient in low-data regimes, leveraging their pre-trained knowledge to achieve strong performance with only a few in-context examples. This advantage holds against traditional ML models even with the addition of more than 3 features, given that in classical ML models, with sparse data, more features lead to overfitting- a problem we encountered on our datasets as well. In addition, in our experiments, LLMs excelled at identifying the most important features of the datasets, performing on par with the classical feature selection method we used. On the negative side, LLMs require substantial computational resources to train and deploy. Meanwhile, traditional models such as Ridge regression or RandomForest can be trained rapidly on standard hardware and may eventually surpass LLMs if enough data is provided.

4.4 Data Contamination

In our experiments, we showed that LLMs can perform regression on realistic datasets, even when feature names are anonymized. However, as detailed in the experiments section and illustrated in Figure 6, the improvement does not correlate with the actual importance of input features in the dataset. This discrepancy strongly suggests that data contamination occurs even at the level of numerical values. These findings challenge the claims made by (Vacareanu et al., 2024), which uses a prompt configuration similar to Anonymized Features to avoid data contamination. Their study utilizes either well-known Friedman formulas or formulas that can be closely approximated (using two decimal places) by common mathematical expressions⁶.

4.5 Order of Features & In-Context Examples

In our experiments, we explored two factors: rearranging the order of features in the Direct QA prompt configuration and sorting the in-context examples in the Anonymized Features prompt configuration. When permuting the order of our three features (resulting in six possible permutations),

we found only a slight change in performance. After sorting the in-context examples by their label values (with three features), we tested the impact on performance using sets of 10 and 100 in-context examples. While both scenarios resulted in a drop in performance, the drop was notably steeper when using 100 examples. Among the models we evaluated, GPT-4 was the least affected, maintaining robust performance with 10 sorted examples but showing a marked decrease with 100 examples.

We hypothesize that sorting examples by label values may cause LLMs to prioritize the sorted label pattern rather than the relationships between features and labels—an effect possibly affected by LLMs’ inherent ability to detect and continue sorting patterns (Besta et al., 2024). We noticed that when the examples were sorted in ascending order, the mean predicted value went up, and when sorted in descending order, it went down. This suggests that the sorting pattern guides the model’s predictions, and it is best to keep the order of in-context examples randomized.

5 Conclusion

In this study, we have explored the ICL capabilities of LLMs through an empirical study on regression tasks. Our results demonstrate that LLMs utilize a blend of retrieved internal knowledge and learned information from in-context examples. Our findings extend the evaluations of prior hypotheses on ICL. For example, we evaluate the usage of outputs in in-context learning examples and show how to manipulate their effectiveness. This work not only advances our understanding of LLMs’ in-context learning phenomenon but also offers practical insights for optimizing their application through careful prompt engineering.

Acknowledgments

This project is partially supported by the Office of Naval Research (ONR) grant N00014-23-1-2417. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of Office of Naval Research. We thank the anonymous reviewers for their valuable feedback and suggestions, which have helped improve this work.

⁶E.g. $y = 10x + \sin(5\pi x) + \cos(6\pi x) \approx 10x$.

Limitations

Scope of Study

Our framework focuses on regression tasks in line with most previous meta-learning research. We also utilize three diverse regression datasets. More regression (or classification) datasets with different characteristics, such as different relationships between their features, can be used for future investigation, which could elaborate more on the findings of our work with the previous work that dismissed the importance of outputs for learning.

Interpretability

We interact with LLMs as a black box and the underlying neural mechanisms remain opaque in our experiments. Integrating this approach with interpretability techniques could yield a deeper understanding of ICL.

Experimental Constraints

The token limit of some LLMs prevented us from testing beyond 100 in-context examples and three features in certain combinations. Although we found that results with 200 in-context examples closely mirrored those with 100, including a 4th feature could have further reinforced our findings. When tested, the results of the 4th feature aligned with our conclusions. However, we ultimately restricted our analysis to three features to maintain a comprehensive combination across all models.

Data Contamination Challenge

Addressing the challenge of data contamination remains a complex issue (Sainz et al., 2023; Balloccu et al., 2024). As mentioned earlier, it is hard to distinguish the degree of Data Contamination from approximate Knowledge Retrieval. Based on our findings, data contamination can happen even with the sequence of numerical values without any linguistic clues. To mitigate this issue, we suggest using distributions that are neither widely recognized nor easily approximated by common mathematical expressions. Furthermore, for datasets, it is advisable to utilize information sourced from regions beyond the United States or outside the primary training domain of the language model. This approach helps to minimize the potential influence of knowledge retrieval in generating the results.

References

- Mohan S Acharya, Asfia Armaan, and Aneeta S Antony. 2019. [A comparison of regression models for prediction of graduate admissions](#). In *2019 International Conference on Computational Intelligence in Data Science (ICCIDS)*, pages 1–5.
- AI@Meta. 2024. Llama 3 model card. https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md. Accessed: 2024-07-01.
- Anthropic. 2024. Claude 3. <https://www.anthropic.com>. Large language model, accessed June 2024.
- Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. 2023. [Transformers as statisticians: Provable in-context learning with in-context algorithm selection](#). In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Simone Balloccu, Patrícia Schmidová, Mateusz Lango, and Ondrej Dusek. 2024. [Leak, cheat, repeat: Data contamination and evaluation malpractices in closed-source LLMs](#). In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 67–93, St. Julian’s, Malta. Association for Computational Linguistics.
- Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michał Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefer. 2024. [Graph of thoughts: solving elaborate problems with large language models](#). In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence, AAAI’24/IAAI’24/EAAI’24*. AAAI Press.
- Satwik Bhattamishra, Arkil Patel, Phil Blunsom, and Varun Kanade. 2024. [Understanding in-context learning in transformers and LLMs by learning to learn discrete functions](#). In *The Twelfth International Conference on Learning Representations*.
- Leo Breiman. 2001. [Random forests](#). *Machine Learning*, 45(1):5–32.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

- Xiang Cheng, Yuxin Chen, and Suvrit Sra. 2024. [Transformers implement functional gradient descent to learn non-linear functions in context](#). In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 8002–8037. PMLR.
- Julian Coda-Forno, Marcel Binz, Zeynep Akata, Matt Botvinick, Jane Wang, and Eric Schulz. 2023. [Meta-in-context learning in large language models](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 65189–65201. Curran Associates, Inc.
- Gilad Deutch, Nadav Magar, Tomer Natan, and Guy Dar. 2024. [In-context learning and gradient descent revisited](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1017–1028, Mexico City, Mexico. Association for Computational Linguistics.
- Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. 2024. [A survey on in-context learning](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1107–1128, Miami, Florida, USA. Association for Computational Linguistics.
- Xi Fang, Weijie Xu, Fiona Anting Tan, Jiani Zhang, Ziqing Hu, Yanjun (Jane) Qi, Scott Nickleach, Diego Socolinsky, "SHS" Srinivasan Sengamedu, and Christos Faloutsos. 2024. [Large language models \(llms\) on tabular data: Prediction, generation, and understanding - a survey](#). *Transactions on Machine Learning Research*.
- Shivam Garg, Dimitris Tsipras, Percy S 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*, volume 35, pages 30583–30598. Curran Associates, Inc.
- Michael Hahn and Navin Goyal. 2023. [A theory of emergent in-context learning as implicit structure induction](#). *Preprint*, arXiv:2303.07971.
- Arthur E Hoerl and Robert W Kennard. 1970. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. 2023. [Mistral 7b](#). *Preprint*, arXiv:2310.06825.
- Jannik Kossen, Yarin Gal, and Tom Rainforth. 2024. [In-context learning learns label relationships but is not conventional learning](#). In *The Twelfth International Conference on Learning Representations*.
- Brett Lantz. 2013. *Machine Learning with R*. Packt Publishing Ltd.
- Jiaoda Li, Yifan Hou, Mrinmaya Sachan, and Ryan Cotterell. 2024. [What do language models learn in context? the structured task hypothesis](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12365–12379, Bangkok, Thailand. Association for Computational Linguistics.
- Yingcong Li, Kartik Sreenivasan, Angeliki Gianou, Dimitris Papailiopoulos, and Samet Oymak. 2023. [Dissecting chain-of-thought: Compositionality through in-context filtering and learning](#). In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. [What makes good in-context examples for GPT-3?](#) In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.
- Gilles Louppe, Louis Wehenkel, Antonio Sutera, and Pierre Geurts. 2013. Understanding variable importances in forests of randomized trees. *Advances in neural information processing systems*, 26.
- Daniel Machlab and Rick Battle. 2024. [Llm in-context recall is prompt dependent](#). *Preprint*, arXiv:2404.08865.
- Sewon Min, Xinxu Lyu, Ari 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, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Ananay Mital. 2023. [US Used Cars Dataset](#).
- OpenAI. 2023. [Gpt-4 technical report](#). *Preprint*, arXiv:2303.08774.
- Jane Pan, Tianyu Gao, Howard Chen, and Danqi Chen. 2023. [What in-context learning “learns” in-context: Disentangling task recognition and task learning](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8298–8319, Toronto, Canada. Association for Computational Linguistics.
- Karl Pearson. 1895. Vii. note on regression and inheritance in the case of two parents. *proceedings of the royal society of London*, 58(347-352):240–242.

- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.
- Yasaman Razeghi, Robert L Logan IV, Matt Gardner, and Sameer Singh. 2022. [Impact of pretraining term frequencies on few-shot numerical reasoning](#). In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 840–854, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Oscar Sainz, Jon Campos, Iker García-Ferrero, Julen Etxaniz, Oier Lopez de Lacalle, and Eneko Agirre. 2023. [NLP evaluation in trouble: On the need to measure LLM data contamination for each benchmark](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10776–10787, Singapore. Association for Computational Linguistics.
- Jurgen Schmidhuber. 1987. Evolutionary principles in self-referential learning. *On learning how to learn: The meta-meta-... hook.* Diploma thesis, Institut f. Informatik, Tech. Univ. Munich, 1(2):48.
- Lingfeng Shen, Aayush Mishra, and Daniel Khashabi. 2024. [Position: Do pretrained transformers learn in-context by gradient descent?](#) In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 44712–44740. PMLR.
- Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, Soroosh Mariooryad, Yifan Ding, Xinyang Geng, Fred Alcober, Roy Frostig, Mark Omernick, Lexi Walker, Cosmin Paduraru, Christina Sorokin, Andrea Tacchetti, Colin Gaffney, Samira Daruki, Olcan Serincinoglu, and Others. 2024. [Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context](#). Preprint, arXiv:2403.05530.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajwal Bhargava, Shrutu Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem 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, Rishi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. [Llama 2: Open foundation and fine-tuned chat models](#). Preprint, arXiv:2307.09288.
- Robert Vacareanu, Vlad Andrei Negru, Vasile Suciu, and Mihai Surdeanu. 2024. [From words to numbers: Your large language model is secretly a capable regressor when given in-context examples](#). In *First Conference on Language Modeling*.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- Johannes von Oswald, Eyvind Niklasson, Maximilian Schlegel, Seijin Kobayashi, Nicolas Zucchet, Nino Scherrer, Nolan Miller, Mark Sandler, Blaise Agüera y Arcas, Max Vladymyrov, Razvan Pascanu, and João Sacramento. 2023. [Uncovering mesa-optimization algorithms in transformers](#). Preprint, arXiv:2309.05858.
- 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 *Thirty-seventh Conference on Neural Information Processing Systems*.
- Noam Wies, Yoav Levine, and Amnon Shashua. 2023. [The learnability of in-context learning](#). In *Thirty-seventh Conference on Neural Information Processing Systems*.
- 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](#). In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2422–2437, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Wenhao Zhu, Yunzhe Lv, Qingxiu Dong, Fei Yuan, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2023. [Extrapolating large language models to non-english by aligning languages](#). Preprint, arXiv:2308.04948.
- Zuhashaik et al. 2023. Llama-2-70b model: Challenges with long token sequences. <https://github.com/meta-LLaMA/LLaMA/issues/887>. GitHub Issue, accessed June 2024.

A Related Work

As mentioned in the introduction, the research in this field can be categorized into meta-learning (Schmidhuber, 1987) and knowledge retrieval. So far, most works are theoretical, and most can be categorized into one of these two groups. However, both of these aspects are changing as this field rapidly expands; New empirical research is coming along, and new ideas are emerging, such as ours, that can not easily fit into one of these groups. For example, (Li et al., 2023) proposes a structure that improves a basic transformer’s in-context learning capabilities by breaking the input into multiple steps. The transformer still uses the outputs but also the knowledge that is given to it **in its input**. Another example would be (Kossen et al., 2024), which empirically tests LLMs and claims that they can learn from the output labels but in an arbitrary manner, a challenge our paper aims to address. Another interesting example that we cannot simply put into a group is (Coda-Forno et al., 2023), which performs meta-in-context learning by showing similar entire tasks and their solutions in the context of the model. Despite these emerging works, it’s worth noting that most papers can still be classified into our established groups. When evaluating the literature, it’s crucial to distinguish between theoretical claims and those based on empirical or synthetic data. Some theoretical papers suggest or imply that their findings might extend to LLMs, but such extrapolations should be approached with caution, as highlighted by (Deutch et al., 2024; Shen et al., 2024).

In the meta-learning research, (Bai et al., 2023) theoretically prove that transformers can be trained to implement linear models within a specific error threshold. Then, they practically test the capability of an encoder-based transformer to implement generalized linear models in context and show that they are also robust against noisy data. In order to test the ICL capability of transformers, (Garg et al., 2022) use a GPT-2 architecture which is closer to an LLM than a simple transformer. However, they modify this GPT-2 architecture for regression using the input features inside the embeddings instead of using tokens and output a number instead of generating tokens. Their model’s performance surpasses linear models as well as KNN models, XGBoost, and a 2-layer neural network. Different from these, (von Oswald et al., 2023) hypothesize that the strong performance of Trans-

formers comes from an architectural bias towards mesa-optimization which they test on simple sequencing tasks, and (Cheng et al., 2024) suggest Transformers can implement gradient descent in function space, enabling them to learn linear and non-linear models.

In the realm of empirical research, our focus is exclusively on studies examining ICL, distinct from related works involving numerical concepts such as numerical reasoning (Razeghi et al., 2022). (Vacareanu et al., 2024) uses a lot of different LLMs such as Gemini-pro (Team et al., 2024), Claude 3 (Anthropic, 2024), GPT-4 (OpenAI, 2023) to test their ICL regression capabilities. They use various settings with up to three features, but not all features are relevant to the output as limited as a result. This work conducts rigorous testing with many models, and the only issue it has is regarding its claim of avoiding data contamination, which is not well-supported. They either use well-known Friedman formulas or formulas that can be closely approximated (using two decimal places) by common mathematical expressions such as $y = 10x + \sin(5\pi x) + \cos(6\pi x)$ simplified as $y = 10x$. Also, the data is based on formulas, which are not realistic regression datasets.

The first paper that argues output labels do not matter was (Min et al., 2022). It tested various LLMs across multiple tasks and concluded that replacing the outputs with random labels does not affect the results, and the only important elements of ICL are limited to defining the label space, input distribution, and overall task format. Following this work, (Pan et al., 2023) uses the same sentiment analysis, toxicity detection, natural language inference/paraphrase detection, and topic/stance classification tasks but argues that learning and retrieval occur under different circumstances. In contrast to our research, their approach employs less complex tasks (same tasks used in (Min et al., 2022)), which they acknowledge could influence the observed patterns. Additionally, they utilize more basic models, including earlier and smaller iterations of GPT-3 such as ada. As a result, their findings yield distinct trends and visual representations compared to our work. Inspired by (Hahn and Goyal, 2023), (Li et al., 2024) dismisses the significance of output labels and proposes that learned solutions can be combined for inference. The main limitation of this work is the exclusive use of LLaMA 2 (Touvron et al., 2023). Our experimental results differ from (Min et al., 2022; Pan et al., 2023; Li et al.,

2024), which requires a unified platform for direct comparison, which is practically infeasible. We argue this conflict is mainly due to the task and model selection. We will explain further on (Li et al., 2024) in Appendix B.

B Learning from Outputs

In this section, we examine the paper (Li et al., 2024) and why its findings differ from ours. As mentioned earlier, the main problem with this work lies in its exclusive use of LLaMA 2 (Touvron et al., 2023), a model known to struggle with long token contexts (Machlab and Battle, 2024; Zuhashaik et al., 2023) which negatively impacts the ICL. While the choice of LLaMA 2 as a powerful open-source LLM at the time is understandable, LLaMA 3 (AI@Meta, 2024), which addressed many of LLaMA 2’s issues and offered significantly improved capabilities, was already available at the time of publication. Some of their experiments require direct access to the LLM architecture, which necessitates an open-source LLM. However, many of their experiences can be done with the GPT family or other powerful models. Based on the LLM choice alone, this work is severely limited. Further limiting the study, the authors selected three simple review sentiment/news-type datasets that even basic machine learning models can excel at.

Regarding specific experiments, experiment 4.1 demonstrates that mapping each label to another fixed label yields results worse than a random baseline, even with 40 in-context examples. This finding contradicts our work and other related studies, which have shown that models can learn from outputs. We suspect this discrepancy may be attributed to LLaMA 2’s well-documented issue of forgetting tasks as the number of tokens increases. Confusion ensues because the outputs likely appear nonsensical to LLaMA 2, and it has forgotten its instructions. It would have been beneficial for the authors to specify whether LLaMA 2 was outputting incorrect labels or entirely irrelevant answers.

Experiments 4.2 and 5.1 also raise questions, as they rely on token embeddings, which, without further evidence, appear unrelated to ICL. For example, (Li et al., 2023) suggests that the level-by-level inference in transformers is necessary for ICL. The combination of embeddings and other factors, such as training the linear model for 80 epochs to match LLaMA 2’s 80 layers, seems arbitrary and lacks clear justification in these experiments.

In conclusion, while the paper (Li et al., 2024) offers interesting insights specifically regarding its categorization of the three hypotheses, its methodological choices and reliance on LLaMA 2 significantly limit its relevance to the broader discussions of in-context learning in language models.

C Datasets

To assess the LLMs’ performance across diverse domains and complexity levels, we select three regression datasets. These datasets are pre-processed to select their most important features. All numerical values in the datasets are rounded to two decimal points. We divide each dataset into two splits: an in-context subset comprising 100 instances and a test subset with 300 instances.

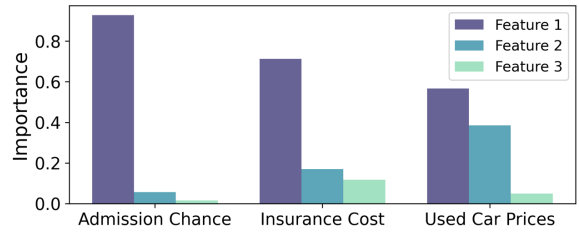


Figure 8: The relative importance of features across our datasets calculated with RandomForest (also reflecting improvements in our Ridge as later shown in Figure 6.

C.1 Admission Chance

This dataset estimates the *likelihood of admission to graduate programs for Indian students* (Acharya et al., 2019). Since this dataset is about Indian students, it may be less seen in our model’s training data, reflecting the general imbalance that skews towards USA-centric data (Zhu et al., 2023). The dataset’s selected features are CGPA (Cumulative Grade Point Average), GRE Score, and TOEFL Score. Among our three datasets, this is the only dataset with high inter-correlation among each feature pair (Pearson correlation (Pearson, 1895) greater than 0.80). As a result, the feature importances, which are calculated with RandomForest (Louppe et al., 2013) and shown in Figure 8, assign minimal importance to the second and third features. The target variable has a mean of 0.72 and a standard deviation of 0.14.

C.2 Insurance Cost

This dataset focuses on predicting a similar distribution of the *annual individual medical costs billed*

by health insurance companies in the USA, drawing from demographic statistics compiled by the U.S. Census Bureau (Lantz, 2013). The selected features are Smoker Status, BMI, and Age with their importances shown in Figure 8. Smoking Status emerges as the most critical feature, followed by BMI, and then Age. The average and standard deviations of costs are 13, 270.42 and 12, 110.01, respectively.

C.3 Used Car Prices

We use selected subset of the used car prices dataset (Mital, 2023) that involves predicting the price of a used Toyota or Maserati car in 2019. The features include City Fuel Economy, Mileage with similar high importance, and Passenger Car Classification with minimal importance, as shown in Figure 8. The target variable has a mean of 42, 279.49 and a standard deviation of 50, 014.51.

D Hyper-Parameters

As mentioned earlier, we evaluate several LLMs, which are LLaMA 3 70B (AI@Meta, 2024), GPT3.5 (Brown et al., 2020), and GPT4 (OpenAI, 2023). We also used Ridge regression (Hoerl and Kennard, 1970) and RandomForest (Breiman, 2001). The GPT-3 and GPT-4 versions we use are "gpt-4-0125-preview", "gpt-3.5-turbo-0125". In these models, the temperature is set to 0.1, and make tokens is set to 10. Each time the models do not generate a number (i.e., generate a disclaimer that says "the data is insufficient"), the seed value, which is initialized to 100, is increased by one. This precaution was not necessary for GPT-4, but GPT-3.5 sometimes refused to answer the questions. For LLaMA 3, top p value is set to 0.99, max tokens to 6, and the temperature to 0.1. Similar to GPT models, we prompt LLaMA 3 again if it fails to generate a number. For Ridge, the default Python class of scikit (Pedregosa et al., 2011) has the alpha value (the normalizing variable) of 1. For RandomForest, however, we wanted to make it more generalized. Consequently, we used 10, 000 estimators and a max depth of 2 for it.

E MAE and R^2

In this section, we depict the main diagrams in the paper with MAE and $1 - R^2$ metrics instead of MSE. MAE is less sensitive to outliers compared to MSE and is expressed in the same units as the original data. R^2 , also known as the coefficient of

determination, ranges from 0 to 1 and represents the proportion of variance in the dependent variable that is predictable from the independent variable(s). An R^2 value of 1 indicates that the model explains all the variability of the response data around its mean, while 0 means the model explains none of the variability. It can also be negative, indicating that the model fits the data worse than the Mean model. R^2 differs from MSE and MAE because the higher, the better. As a result, to make R^2 diagrams comparable, instead of R^2 , we show the results for $1 - R^2$ (1 is included to show the negative results. This also moves the baseline results of R^2 to 1 for the Mean model.

For Figures 2,3,4,5 and 6 of the main paper, the MAE results are shown in Figures 9,11,13,15 and 17, respectively. For Figures 2,3,4,5, and 6 of the main paper, the $1 - R^2$ results are shown in Figures 10,12,14,16 and 18, respectively.

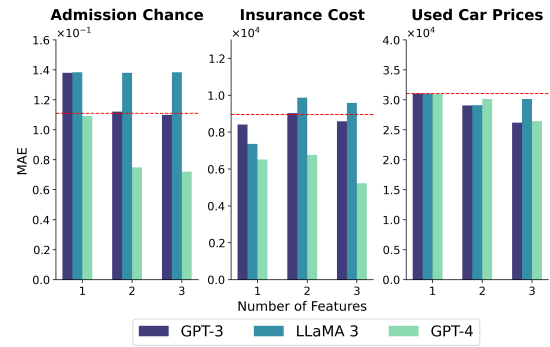


Figure 9: This figure shows Figure 2 of the main paper with MAE instead of MSE. The figure shows the baseline results of LLMs' (Direct QA configuration) across datasets and number of features. The dashed red line shows the performance of the Mean model.

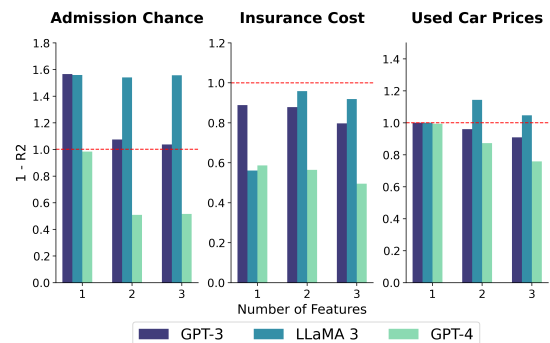


Figure 10: This figure shows Figure 2 of the main paper with $1 - R^2$ instead of MSE. The figure shows the baseline results of LLMs' (Direct QA configuration) across datasets and number of features. The dashed red line shows the performance of the Mean model.

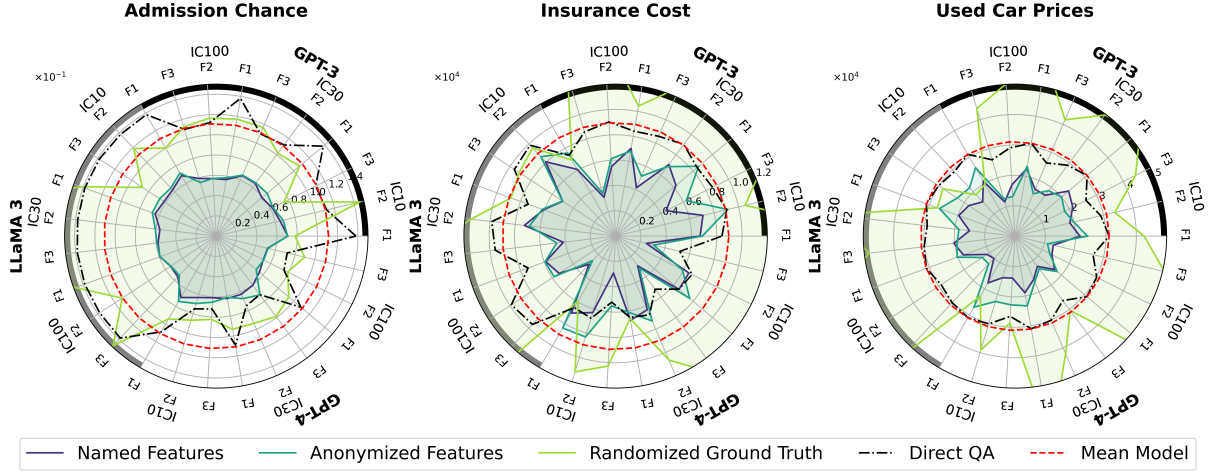


Figure 11: This figure shows Figure 3 of the main paper with MAE instead of MSE. The figure shows the comprehensive comparison of prompt configurations’ effects on our models across various factors, shown in a hierarchy. The top level for each dataset distinguishes between GPT-3, LLaMA 3, and GPT-4 results using black, grey, and white arcs, respectively. The notation IC_i indicates the number of in-context examples, while F_1 , F_2 , and F_3 represent the use of the first feature, the first two features, and all three features, respectively.

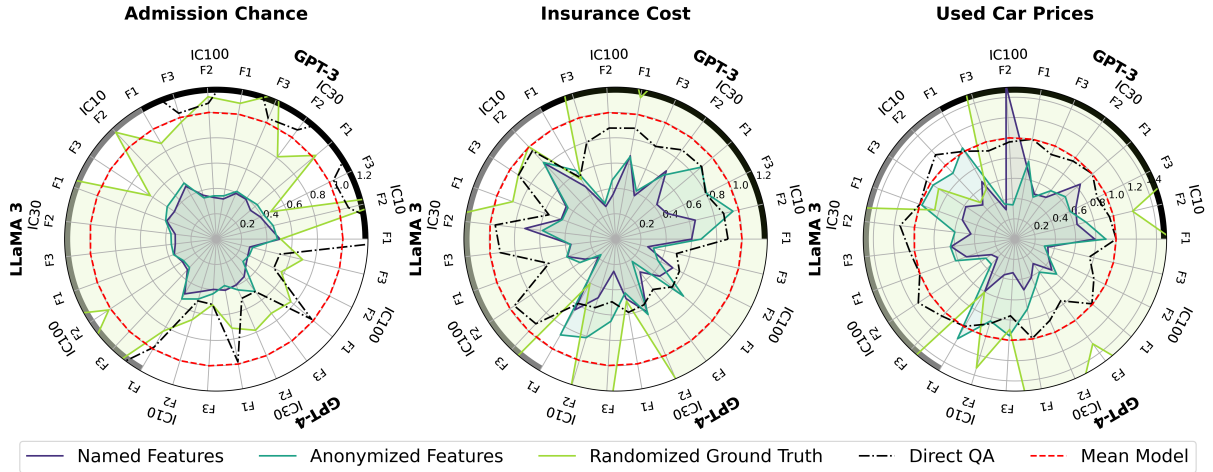


Figure 12: This figure shows Figure 3 of the main paper with $1-R^2$ instead of MSE. The figure shows the comprehensive comparison of prompt configurations’ effects on our models across various factors, shown in a hierarchy. The top level for each dataset distinguishes between GPT-3, LLaMA 3, and GPT-4 results using black, grey, and white arcs, respectively. The notation IC_i indicates the number of in-context examples, while F_1 , F_2 , and F_3 represent the use of the first feature, the first two features, and all three features, respectively.

F Reasoning Configuration and Additional Diagrams

In this section, we include some diagrams that could have provided more insight and were removed from the main text for space. We also show examples of the reasoning configurations and diagrams related to that.

F.1 Additional Diagrams of the Main Paper

Here, we included the variants of Figures 5 and 6 of the main paper that we omitted for space. Figure 5 with 1 and 2 features is shown in Figures 19 and

20. Figure 6, with 10 and 30 in-context examples, is shown in Figures 21 and 22.

F.2 Reasoning Examples

In this section, we show examples of reasoning for estimating an output for each dataset and LLM with 3 features. We will show the results of this setting in the next section.

F.2.1 GPT-3 Reasoning over the Admission Chance Dataset:

Based on the given information, the applicant has a high Cumulative GPA of 8.48, a decent GRE score

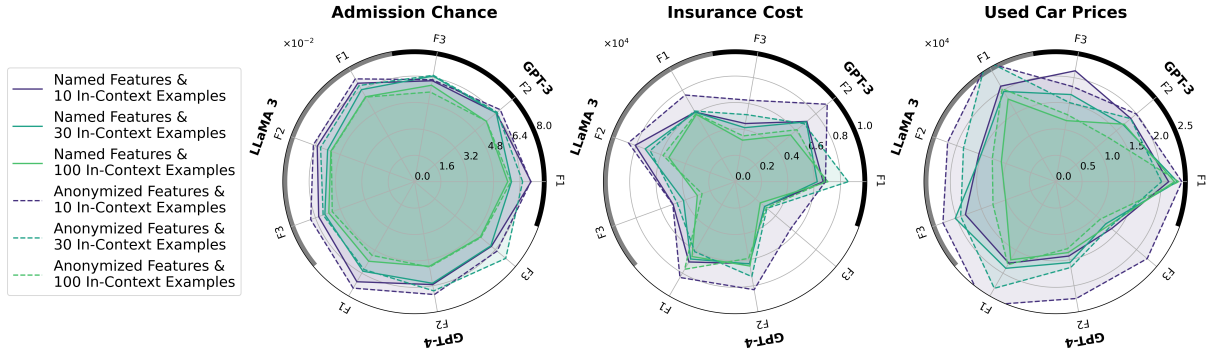


Figure 13: Figure 4 of the main paper with MAE instead of MSE. The figure shows the comparison of the number of in-context examples using Named Features (straight lines) and Anonymized Features (dashed lines) prompt configurations. F1, F2 and F3 represents using 1st (F1), then 1st and 2nd (F2), and all three feature (F3).

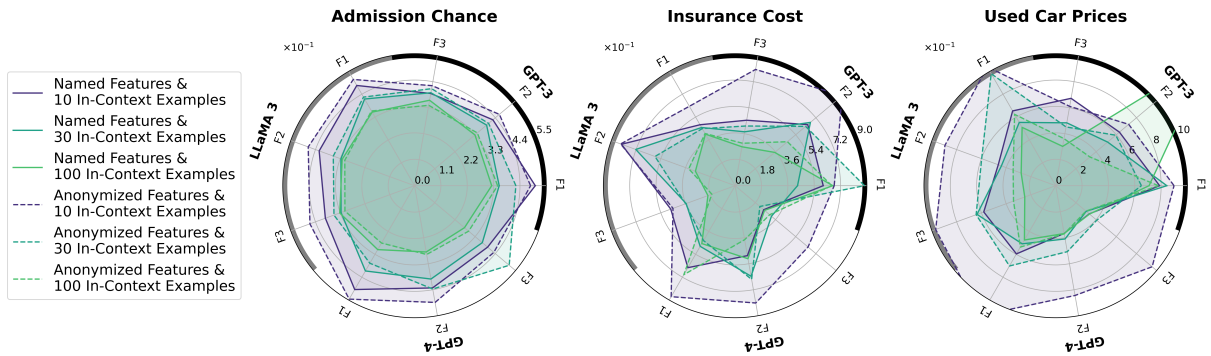


Figure 14: Figure 4 of the main paper with $1-R^2$ instead of MSE. The figure shows the comparison of the number of in-context examples using Named Features (straight lines) and Anonymized Features (dashed lines) prompt configurations. F1, F2 and F3 represents using 1st (F1), then 1st and 2nd (F2), and all three feature (F3).

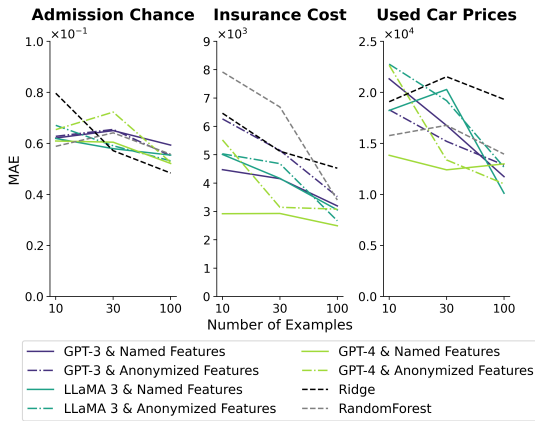


Figure 15: This figure shows Figure 5 of the main paper with MAE instead of MSE. The figure shows the performance of Named Features and Anonymized Features prompt configurations, Ridge, and RandomForest for 3 features based on the number of in-context examples.

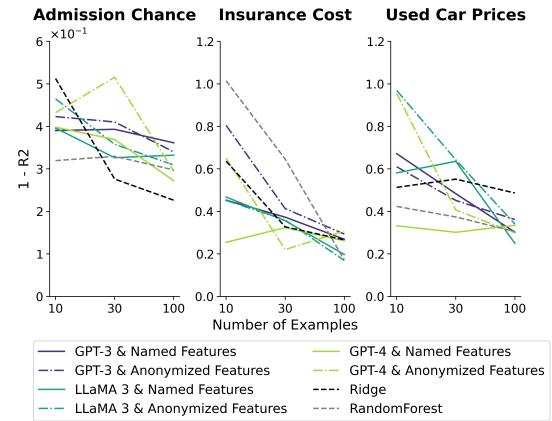


Figure 16: This figure shows Figure 5 of the main paper with $1-R^2$ instead of MSE. The figure shows the performance of Named Features and Anonymized Features prompt configurations, Ridge, and RandomForest for 3 features based on the number of in-context examples.

of 311.0, and a good TOEFL score of 104.0. These scores are above average and indicate a strong academic background.

Considering that the average Chance of Admis-

sion is 0.74 with a standard deviation of 0.14, and the applicant's strong academic profile, I would estimate that the Chance of Admission for this applicant is higher than the average.

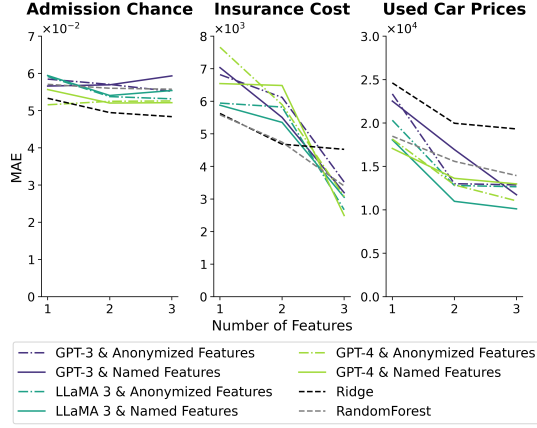


Figure 17: This figure shows Figure 6 of the main paper with MAE instead of MSE. The figure shows the performance of Anonymized Features prompt configuration, Ridge, and RandomForest for 100 in-context examples based on the number of features.

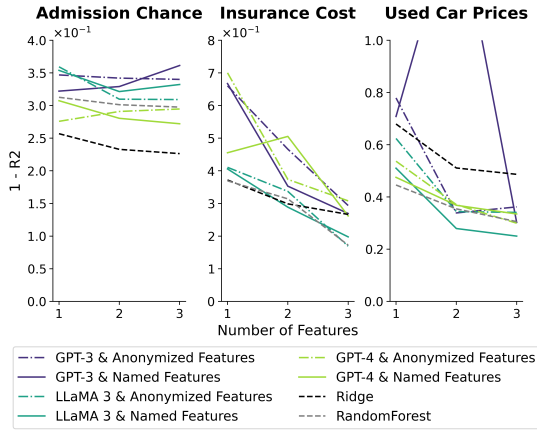


Figure 18: This figure shows Figure 6 of the main paper with $1-R^2$ instead of MSE. The figure shows the performance of Anonymized Features prompt configuration, Ridge, and RandomForest for 100 in-context examples based on the number of features.

My final estimation is 0.80.

F.2.2 LLaMA Reasoning over the Admission Chance Dataset:

Based on the provided information, I will estimate the Chance of Admission.

The given scores are all above average, indicating a strong applicant. The cumulative GPA of 8.48 is excellent, suggesting a high level of academic achievement. The GRE score of 311.0 is also above average, indicating strong verbal and quantitative skills. The TOEFL score of 104.0 is well above the average required for most graduate programs, demonstrating a high level of English proficiency.

Considering these strong scores, I would expect

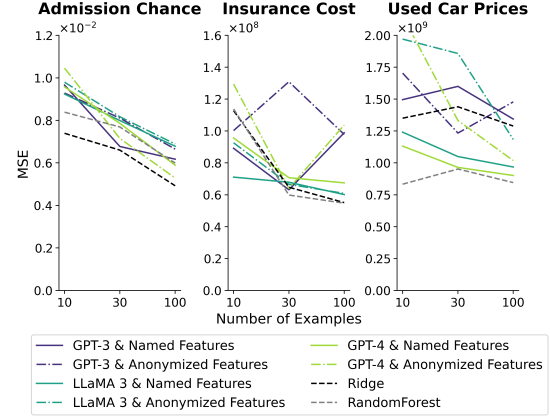


Figure 19: This figure shows Figure 5 of the main paper with 1 feature instead of 3. The figure shows the performance of Named Features and Anonymized Features prompt configurations, Ridge, and RandomForest based on the number of in-context examples.

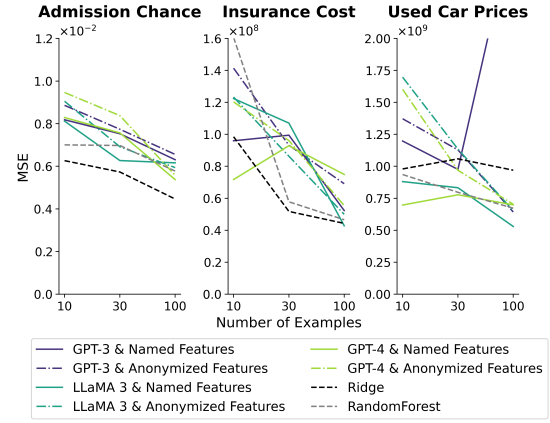


Figure 20: This figure shows Figure 5 of the main paper with 2 features instead of 3. The figure shows the performance of Named Features and Anonymized Features prompt configurations, Ridge, and RandomForest based on the number of in-context examples.

the Chance of Admission to be higher than the average of 0.74. Given the standard deviation of 0.14, I would estimate the Chance of Admission to be around one standard deviation above the mean.

My final estimation is 0.88.

F.2.3 GPT-4 Reasoning over the Admission Chance Dataset:

To estimate the Chance of Admission based on the provided information, we need to consider how the given data points (Cumulative GPA, GRE Score, TOEFL Score) might relate to the average Chance of Admission and its standard deviation. However, without specific information on how these factors correlate with admission chances, we'll have to make some general assumptions based on typical

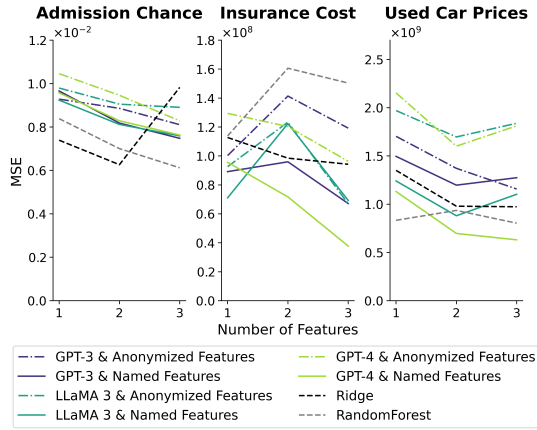


Figure 21: This figure shows Figure 6 of the main paper with 10 in-context examples instead of 100. The figure shows the performance of Anonymized Features prompt configuration, Ridge, and RandomForest based on the number of features.

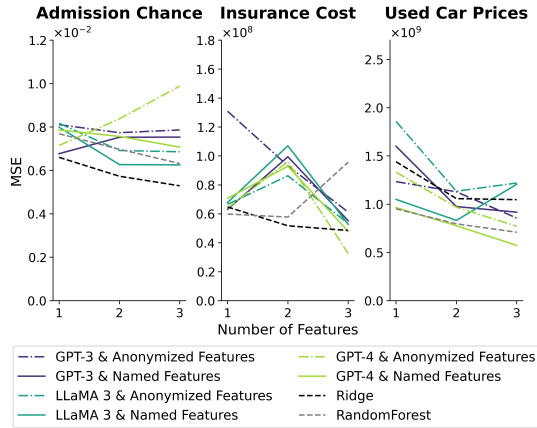


Figure 22: This figure shows Figure 6 of the main paper with 30 in-context examples instead of 100. The figure shows the performance of Anonymized Features prompt configuration, Ridge, and RandomForest based on the number of features.

admissions criteria.

1. **Cumulative GPA**: An 8.48 GPA, assuming it's on a 10-point scale, is quite high. This would generally be seen as a strong positive in the admissions process, suggesting a higher than average chance of admission.

2. **GRE Score**: A score of 311 is somewhat above average (the average GRE scores for many programs tend to hover around the low to mid-300s). While not outstanding, it's competitive enough not to detract significantly from an application, especially with a strong GPA.

3. **TOEFL Score**: A TOEFL score of 104 is good, indicating a strong proficiency in English. For most universities, this score is above the min-

imum requirement and should be sufficient to not negatively impact the chance of admission.

Given the average Chance of Admission is 0.74 with a standard deviation of 0.14, and considering the strong GPA, adequate GRE, and TOEFL scores, it's reasonable to estimate the Chance of Admission to be above average but not exceptionally so due to the GRE score not being far above average.

Taking into account the positive impact of a high GPA and good TOEFL score, balanced against the GRE score, a reasonable estimate might place the Chance of Admission slightly above the average but not by a large margin. Therefore, considering the given data and the average Chance of Admission, my final estimation is 0.78."

F.2.4 GPT-3 Reasoning over the Insurance Cost Dataset:

Based on the given information of being a smoker, having a BMI of 24.7, and being 43 years old, we can expect higher annual medical costs compared to non-smokers, individuals with lower BMIs, and younger individuals. Smokers tend to have higher medical costs due to the increased risk of various health conditions such as lung cancer, heart disease, and respiratory issues. Additionally, as individuals age, they are more likely to develop chronic health conditions that require ongoing medical care, which can also contribute to higher medical costs. A BMI of 24.7 is considered within the normal range, which may have a slightly lower impact on medical costs compared to individuals with higher BMIs.

Taking into account these factors, along with the average annual individual medical costs billed by health insurance in the USA of \$13,270.42 and a standard deviation of \$12,110.01, I estimate that the annual individual medical costs for a 43-year-old smoker with a BMI of 24.7 in the USA would be higher than the average due to the increased health risks associated with smoking and age. Therefore, my final estimation is \$20,000." "Based on the provided information of being a non-smoker with a BMI of 25.8 and an age of 46, we can estimate that the individual is relatively healthy. Typically, younger individuals with a healthy BMI and who do not smoke tend to have lower medical costs compared to older individuals or those with unhealthy habits.

Considering the average annual individual medical costs billed by health insurance in the USA of \$13,270.42 with a standard deviation of

\$12,110.01, we can assume that this individual's medical costs might be lower than the average due to their healthy profile. However, since medical costs can vary significantly based on individual health conditions and unforeseen circumstances, it is challenging to provide an accurate estimate without more specific information.

Taking into account the individual's healthy profile and the average costs in the USA, my final estimation is \$10,000."

F.2.5 LLaMA Reasoning over the Insurance Cost Dataset:

Based on the provided information, we can make an educated estimation of the annual individual medical costs billed by health insurance in the USA.

We are given the average annual individual medical costs billed by health insurance in the USA, which is \$13,270.42, along with the standard deviation of \$12,110.01. This suggests that the distribution of medical costs is likely to be skewed to the right, with a long tail of high-cost individuals.

The additional information provided about the individual, such as being a smoker (smoker: 1.0), having a BMI of 24.7, and being 43 years old, can also influence the estimation. Smokers tend to have higher medical costs due to the increased risk of smoking-related illnesses. A BMI of 24.7 is considered overweight, which can also lead to higher medical costs. The age of 43 is relatively middle-aged, which may not have a significant impact on medical costs compared to older or younger individuals.

Considering these factors, I would adjust the average annual individual medical costs upwards to account for the increased risk associated with smoking and being overweight. However, I would not adjust it too drastically, as the individual's age is not extremely high.

My final estimation is \$16,500."

F.2.6 GPT-4 Reasoning over the Insurance Cost Dataset:

To estimate the annual individual medical costs billed by health insurance in the USA for a 43-year-old smoker with a BMI of 24.7, we need to consider how the given factors (smoking status, BMI, and age) typically influence medical costs, based on the provided average and standard deviation.

1. ****Smoking Status****: Being a smoker generally increases medical costs significantly due to the higher risk of diseases associated with smok-

ing (e.g., lung cancer, heart disease). 2. ****BMI****: A BMI of 24.7 is within the normal weight range (18.5-24.9), which might not significantly increase medical costs compared to those with higher BMI values. 3. ****Age****: At 43 years old, an individual might start to experience more health issues than a younger person, but not as many as someone in a more advanced age group. This could mean slightly higher than average costs due to age but not the highest possible.

Given the average annual medical costs of \$13,270.42 and considering the significant impact of smoking, the effect of age, and the relatively neutral effect of a normal BMI, the costs for this individual would likely be higher than the average but not excessively so due to the mitigating factor of a healthy BMI.

Considering the standard deviation of \$12,110.01, which indicates a wide range of costs, and the fact that smoking is a major cost driver, it's reasonable to estimate that the costs for this individual would be above average but not at the extreme end of the spectrum.

Given these considerations, a reasonable estimate for the annual individual medical costs for this person, factoring in the significant increase due to smoking but moderated by age and a healthy BMI, would be somewhat above the average but not by the full extent of one standard deviation.

My final estimation is \$20,000."

F.2.7 GPT-3 Reasoning over the Used Car Prices Dataset:

To estimate the price of a used car that can be either a Toyota or Maserati in 2019, we can consider the following factors:

1. **City Fuel Economy**: The given city fuel economy of 14.0 suggests that the car may not be very fuel-efficient, which could potentially lower its price.

2. **Mileage**: The mileage of 0.0 indicates that the car is essentially brand new or has very low mileage, which generally tends to increase the price of a used car.

3. **Is Toyota**: The fact that the car is not a Toyota implies that it could be a Maserati, which is a luxury brand known for higher price tags compared to Toyota vehicles.

Considering these factors, we can infer that the car is likely a Maserati due to its non-Toyota status, which would generally have a higher price tag. Given that the average price of a used car that can

be either a Toyota or Maserati in 2019 is \$50,014.51 with a standard deviation of \$42,279.49, and considering the low mileage and potential luxury status of the car, my final estimation for the price of this used car would be higher than the average, so I estimate it to be \$60,000."

F.2.8 LLaMA Reasoning over the Used Car Prices Dataset:

Based on the given information, we can see that the average price of a used car that can be either a Toyota or Maserati in 2019 is 50014.51 with a standard deviation of 42279.49. Since the mileage is 0.0, it's likely that the car is relatively new or has been well-maintained, which could increase its value. The city fuel economy is 14.0, which is a moderate value, but it's not a key factor in determining the price of a used car.

The most important factor is the brand, and since Is Toyota is 0.0, it means the car is a Maserati, which is a luxury brand known for its high-performance and premium vehicles. Maserati cars tend to hold their value well, and their prices are generally higher than those of Toyota cars.

Considering these factors, I would estimate the price of this used car to be higher than the average price. My final estimation is 65000."

F.2.9 GPT-4 Reasoning over the Used Car Prices Dataset:

Given the information, we have a car with a city fuel economy of 14.0, mileage of 0.0, and it is not a Toyota (implying it could be a Maserati since the options are either Toyota or Maserati). Maseratis are luxury vehicles and typically have a higher price point compared to Toyotas. The average price of a used car from the given data is \$50,014.51 with a standard deviation of \$42,279.49.

Considering the car in question is a Maserati (as indicated by ""Is Toyota: 0.0""), and it has 0.0 mileage, which suggests it's either new or nearly new, we can expect the price to be higher than the average of the mixed pool of Toyota and Maserati used cars. The low city fuel economy is characteristic of high-performance or luxury vehicles, which aligns with it being a Maserati.

Given these factors, and knowing that Maseratis are premium vehicles, the price would likely be above the average. However, without knowing the exact model or year (beyond it being a used car as of 2019), we can't pinpoint the exact price but can infer it would be higher due to the brand and the

newness indicated by zero mileage.

Considering the standard deviation and the average, and the fact that this is a luxury vehicle likely on the higher end of the price spectrum, a reasonable estimation, taking into account the luxury brand premium and the new condition, would be one standard deviation above the mean. Thus, my final estimation is \$92,294."

F.3 Reasoning Configuration Diagrams

The diagrams that compare the Direct QA prompt configuration to the reasoning configuration are shown in Figures 23, 24 and 25 for metrics MSE, MAE and $1 - R^2$, respectively. As mentioned in the text of the main paper, there is no rhyme or reason for the behavior of the LLMs in this regard.

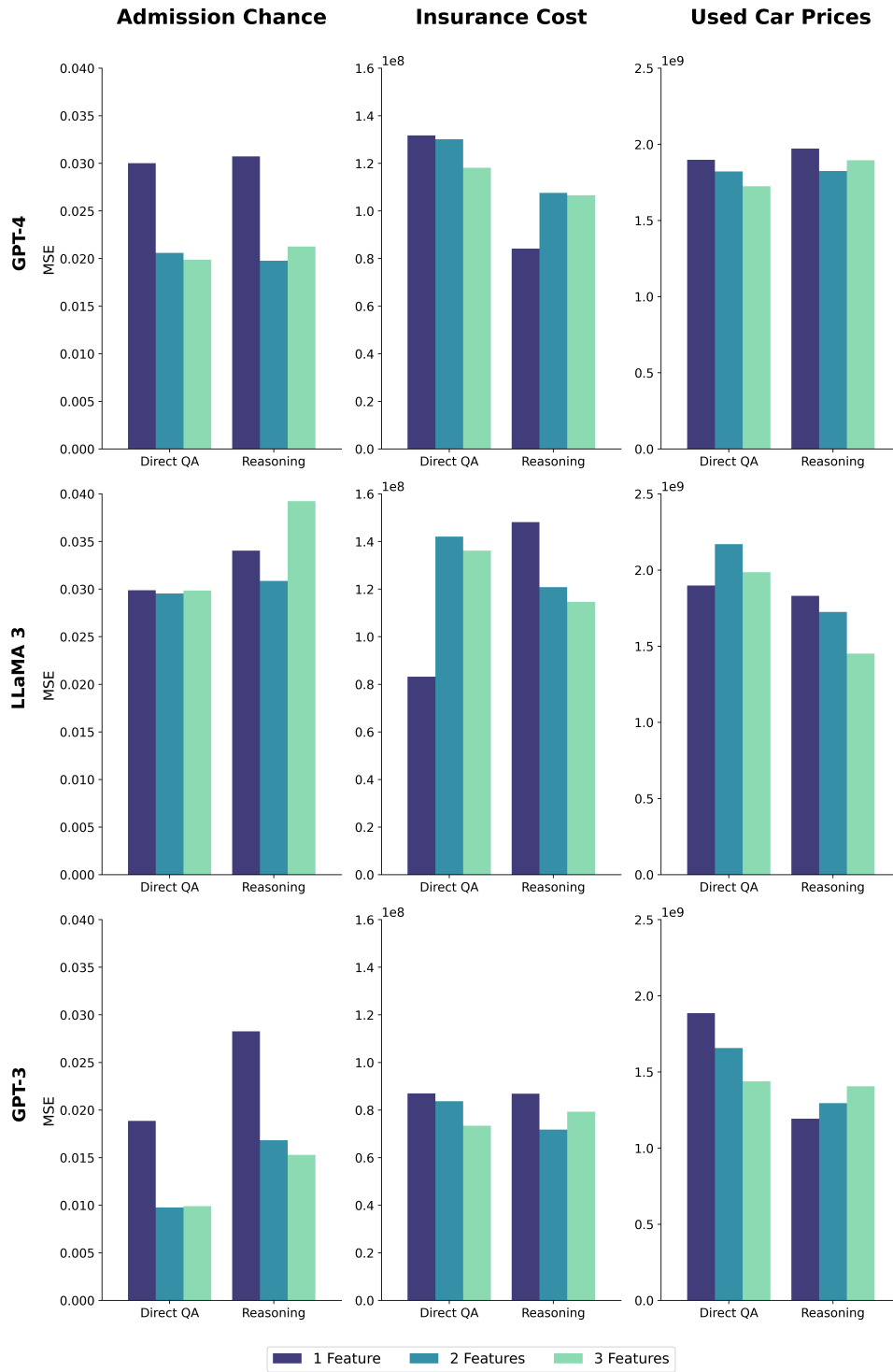


Figure 23: Comparison between Direct QA and Reasoning prompt configurations between dataset, LLMs and the number of features using MSE metric.

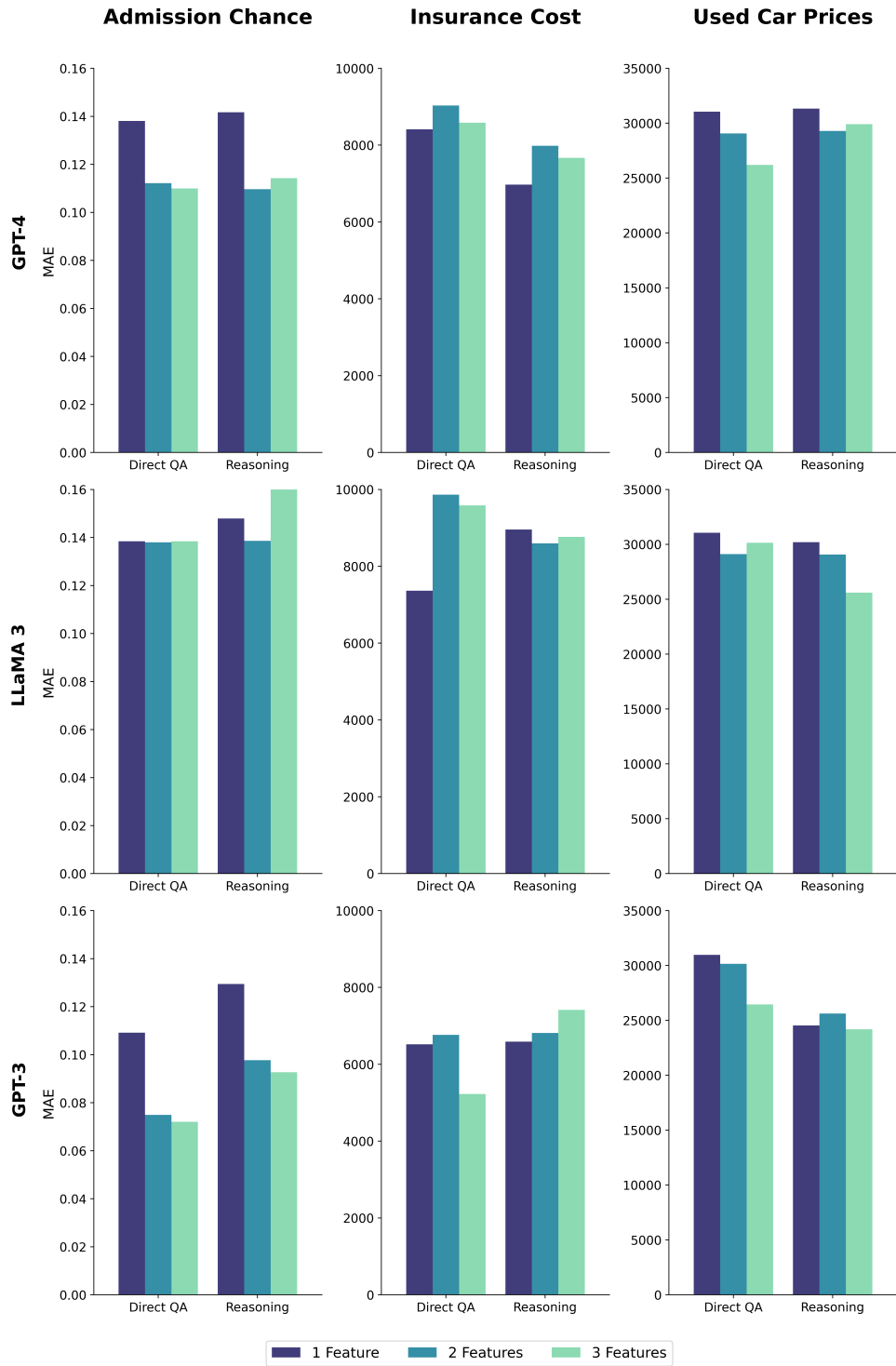


Figure 24: Comparison between Direct QA and Reasoning prompt configurations between dataset, LLMs and the number of features using MAE metric.

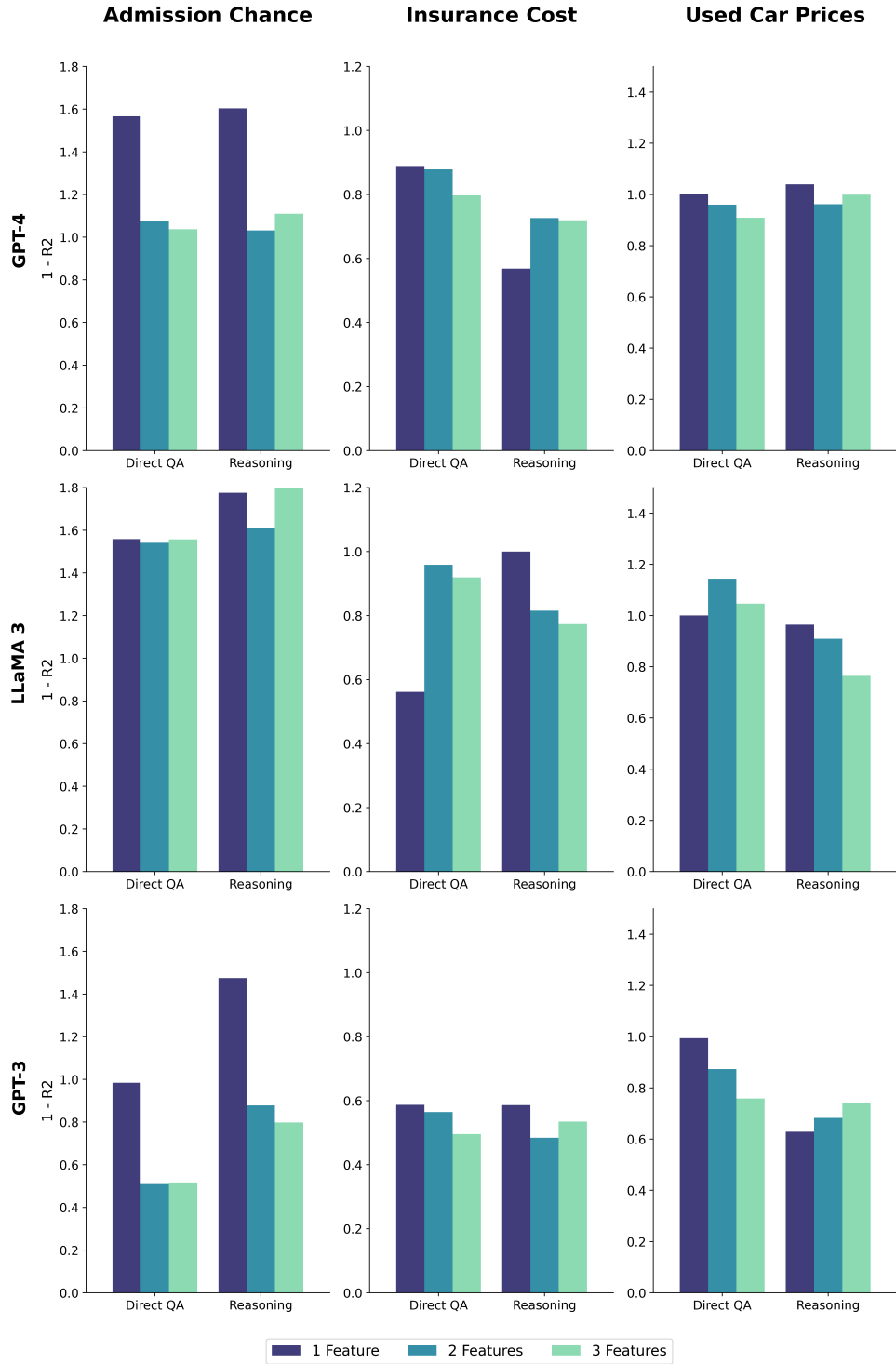


Figure 25: Comparison between Direct QA and Reasoning prompt configurations between dataset, LLMs and the number of features using $1 - R^2$ metric.