Probing the Decision Boundaries of In-context Learning in Large Language Models

Anonymous Author(s)
Affiliation
Address
email

Abstract

In-context learning in large language models enables them to generalize to new tasks by prompting with a few exemplars without explicit parameter updates. In this work, we propose a new mechanism to probe and understand in-context learning from the lens of decision boundaries for in-context classification. Decision boundaries qualitatively demonstrate the inductive biases of standard classifiers. Surprisingly, we find that the decision boundaries learned by current LLMs in simple binary classification tasks are irregular and non-smooth. We investigate factors influencing these boundaries and explore methods to enhance their generalizability. Our findings offer insights into in-context learning dynamics and practical improvements for enhancing its robustness and generalizability.

1 Introduction

A key emergent behavior of recent transformer-based language models is in-context learning, which allows the model to learn tasks by conditioning on a set of demonstrations without training [Wei et al., 2022, Brown et al., 2020]. Recent studies on understanding in-context learning explore theoretical links to gradient descent [Akyürek et al., Von Oswald et al., 2023, Dai et al., 2023] and practical factors affecting performance, such as demonstration accuracy [Min et al., 2022b, Shi et al., 2023], prompt structure, model size [Wei et al., 2023, Webson and Pavlick, 2022], and example order [Chen et al., 2024]. Garg et al. [2022], Nguyen and Grover [2022] demonstrate that even small transformers can learn unseen function classes in-context.

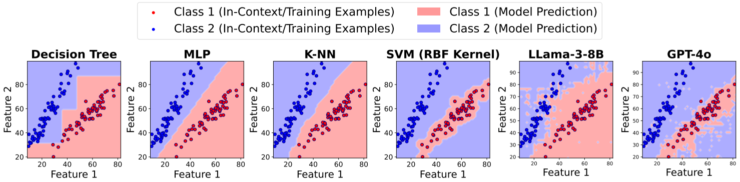


Figure 1: Decision boundaries of LLMs and traditional machine learning models on a linearly separable binary classification task. The background colors represent the model's predictions, while the points represent the in-context or training examples. LLMs exhibit non-smooth decision boundaries compared to the classical models. See Appendix C for model hyperparameters.

Our study offers a new perspective by viewing in-context learning in LLMs as a unique machine learning algorithm, leveraging decision boundary analysis in classification tasks to gain insights into their performance. This method probes the inductive biases and generalization capabilities of LLMs, providing a comprehensive assessment of their robustness. Surprisingly, recent LLMs struggle to provide smooth decision boundaries across classification tasks we tested, regardless of model size, in-context example number and order, and label semantics. This issue persists even in simple binary

Submitted to Workshop on Scientific Methods for Understanding Deep Learning, NeurIPS 2024.

linear classification tasks where classical methods like SVM achieve smooth boundaries with fewer examples as in Figure 1. To explore this, We experimented with various open-source LLMs (Llama series [Touvron et al., 2023, Xia et al., 2023], Mistral [Jiang et al., 2023]) and state-of-the-art closed-source LLMs (GPT-4o and GPT-3.5). We then investigated methods to smooth decision boundaries, including fine-tuning and adaptive prompting strategies. Our contributions can be summarized as follows: 1) Introduced a novel mechanism to probe and understand in-context learning in LLMs by visualizing and analyzing the decision boundaries on classification tasks. 2) Discovered that SoTA LLMs exhibit non-smooth, irregular decision boundaries even on simple linearly separable tasks, unlike classical ML models. 3) Identified several factors impacting decision boundary smoothness, including model size, number of in-context examples, quantization levels, label semantics, and order of examples. 4) Evaluated methods to improve decision boundary smoothness, such as fine-tuning earlier layers and active learning with uncertainty-based sample selection. 5) Demonstrated that fine-tuning LLMs on simple tasks can generalize to complex ones, and training transformers from scratch for in-context learning can lead to smoother boundaries.

2 Methodology

In-Context Classification & Decision Boundary Visualization. In a K-class classification task with data distribution $p_{\text{data}}(\mathbf{x}, y)$, where $\mathbf{x}$ is the input feature and $y \in \{1, \ldots, K\}$ is the class label, we construct an in-context prompt by sampling n examples $(\mathbf{x}_i, y_i) \sim p_{\text{data}}$ for $i = 1, \ldots, n$. Given a new test point $\mathbf{x}_{\text{test}}$, the prompt $P = (\mathbf{x}_1, y_1, \ldots, \mathbf{x}_n, y_n, \mathbf{x}_{\text{test}})$ is fed to the LLM π, which predicts a class $\hat{y}$ for $\mathbf{x}_{\text{test}}$. The LLM predicts by choosing the most likely class in the next token distribution. The class prediction is $\hat{y} = \arg\max_{i \in \{1,\ldots,K\}} l_{c(i)}$, where $l_{c(i)}$ are the logit values for each class label converted to unique token ids. To visualize the decision boundary of model π, we generate a grid of points covering the feature space defined by the in-context examples set $\mathcal{S} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_k, y_k)\}$. We create a uniform grid with G points along each dimension. The grid points are denoted as $\mathbf{X}_{\text{grid}} = \{\mathbf{x}_{\text{query}} \mid \mathbf{x}_{\text{query}} \in [\mathbf{x}_{\min}, \mathbf{x}_{\max}]^d, \mathbf{x}_{\text{query}} = \mathbf{x}_{\min} + i\Delta\mathbf{x}, i \in \{0, 1, \ldots, G-1\}\}$ where $\Delta\mathbf{x} = \frac{1}{G-1}(\mathbf{x}_{\max} - \mathbf{x}_{\min})$ is the grid spacing along each dimension. Each point $\mathbf{x}_{\text{query}} \in \mathbf{X}_{\text{grid}}$ is a query input, and model π is prompted with the sequence $(\mathbf{x}_1, y_1, \ldots, \mathbf{x}_k, y_k, \mathbf{x}_{\text{query}})$ to predict the class label $\hat{y}$. The decision boundary is visualized by plotting the predicted labels $\hat{y}$ over the grid $\mathbf{X}_{\text{grid}}$.

3 Experiments

We examine existing LLMs through the lens of decision boundaries by conducting a series of binary classification tasks under varying conditions. Our experiments aim to address the following key questions: (1) How do existing pretrained LLMs perform on binary classification tasks? §3.1 (2) How do different factors influence the decision boundaries of these models? §3.2 (3) How can we improve the smoothness of decision boundaries through finetuning or prompting? §3.3

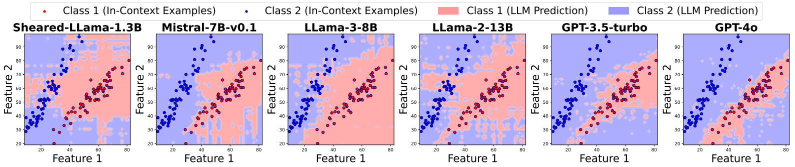


Figure 2: Visualizations of decision boundaries for various LLMs on a linearly seperable binary classification task. The 128 in-context data points are shown as scatter points and the colors indicate the label determined by each model.

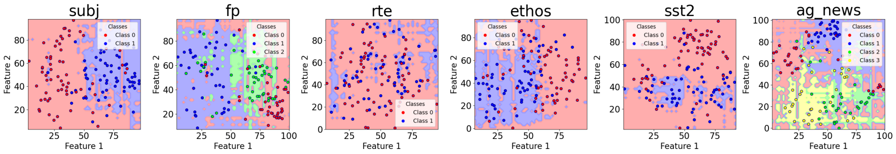


Figure 3: Decision boundaries of Llama-3-8b on six NLP tasks, ranging from binary to multi-class classification. Since text embeddings are natively high-dimensional, we projected text embeddings onto a 2D space using t-SNE. The irregular, non-smooth behaviors are also seen in these tasks.

Experiment Setup. We investigate the decision boundary of LLMs by prompting them with n in-context examples of binary classification tasks, with an equal number of examples for each class. We generate synthetic classification datasets with three types of linear and non-linear classification tasks: linear, circle, and moon, each describing different shapes of ground-truth decision boundaries. Detailed information on the dataset generation can be found in Appendix D. In addition to the in-context examples, we calculate the in-context learning accuracy on a held-out test set of size 100. We sample in-context examples and test points from classification task and convert them into prompt, with an example shown in Appendix K. We study an extensive range of models, with sizes ranging from 1.3B to 13B parameters, including open-source models such as Llama2-7B, Llama3-8B, Llama2-13B, Mistral-7B-v0.1, and sheared-Llama-1.3B as well as closed-source LLMs, including GPT-4o and GPT-3.5-turbo. We generate decision boundaries using 8-bit quantization for open-source models, with a 50x50 grid (2500 queries per boundary). Open-source model predictions use the method in 2, while closed-source models use next token generation.

3.1 Non-Smooth Decision Boundaries of LLMs in synthetic and NLP classification tasks.

Figure 2 compares the decision boundaries of 6 LLMs when provided with 128 in-context examples on synthetic binary classification task. All of them exhibit non-smooth decision boundaries. The decision boundaries vary significantly across models, indicating that these models have different reasoning abilities to interpret the same in-context data. All models show fragmented decision regions, which means small changes in the input features can result in different classifications. The non-smoothness are also observed with experiments on multi-class NLP classification tasks as shown in Figure 3, raising concerns about the reliability of LLMs and their practical deployment, as even when test accuracy for classification is high (shown in Figure 8), the underlying decision boundary lacks generalization.

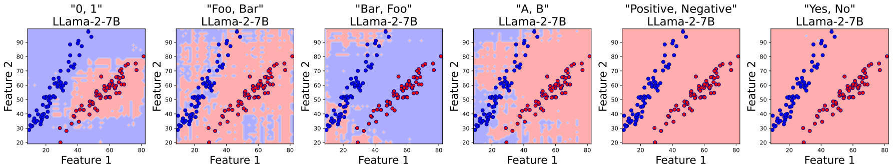


Figure 4: The decision boundaries of LLama-2-7B and LLama-3-8B, across various class labels. Each row corresponds to a model, and each column represents a different class label, shown in quotes.

3.2 How Do Different Factors Influence the Decision Boundaries?

Impact of Model Size on Decision Boundary and Accuracy From Figure 2, model sizes increase from left to right, yet there is no clear correlation between model size and the smoothness of the decision boundary. Even GPT-4o, demonstrates fragmented decision regions. **Increasing In-Context Examples Does Not Guarantee Smoother Decision Boundaries** While classification accuracies tend to improve with more in-context examples, Figure 10 reveals that this does not translate to smoother decision boundaries even as the number of examples increase from 2^3 to 2^6. **How Quantization Affects the Decision Boundary?** Figure 9(a) illustrates the decision boundaries of the LLaMA-2-7B model under different quantization levels Dettmers et al. [2022]. This indicates that the reduced precision from 4-bit quantization significantly affects points near the decision boundary or areas where the model is most uncertain. We plotted the probability prediction for class 1 (Figure 9(b)). This suggests varying quantization levels can flip the LLM's decisions in the regions of highest uncertainty. **Are Decision Boundaries Sensitive to the Prompt Format?** Yes, decision boundaries are sensitive to the labels' names, as shown in Figure 4. Using semantically unrelated labels, such as "Foo" and "Bar" as suggested in [Wei et al., 2023], results in flipped predictions compared to using reversed class names like "Bar" and "Foo". **Are Decision Boundaries Sensitive to the Order of In-Context Learning Examples?** Yes. In Figure 11, we demonstrate that the model's decision boundaries vary with different shuffles of the in-context examples.

3.3 How to Improve the Decision Boundary Smoothness?

Can We Finetune LLMs on the In-Context Examples to Achieve Smoother Decision Boundaries? No. Our experiments indicate that finetuning LLMs on in-context examples does not result in smoother decision boundaries in Appendix J.

How to Finetuning LLMs for Smoother Decision Boundaries in Classification Tasks? We explore two finetuning approaches: (1) Using LoRA [Hu et al., 2021] to finetune the pretrained LLM's attention layers, token embedding layer, or linear head layer; (2) Modifying the LLM architecture (we term as CustomLLM) by freezing the transformer backbone and attaching new embedding layers and prediction head, trained using objective (2). This approach leverages task-specific layers to utilize the backbone's pattern-matching capabilities. Experimental details are in Appendix H. Results in Figures 5 and 13 indicate that finetuning intermediate and earlier embedding layers produces smoother decision boundaries than finetuning the top prediction head.

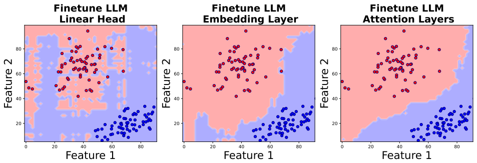
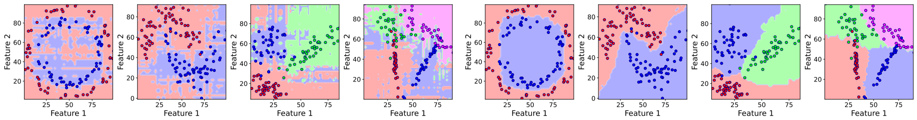


Figure 5: LLM finetuning ablations. Decision boundary after finetuning the linear head, embedding layer and the attention layers.

Can LLMs finetuned on one in-context learning task generalize to more complex in-context learning tasks? Yes, as shown in Figure 14, we found it generalizes to unseen non-linear tasks as well as 3-class and 4-class classification tasks, despite only being trained on a binary linear task.

Can we train a transformer from scratch to learn smooth decision boundaries in-context? We investigate whether pretraining affects decision boundaries by training Transformer Neural Processes (TNPs) [Nguyen and Grover, 2022] from scratch. TNPs are designed for in-context learning, predicting query labels $y_{i>m}$ given query inputs $x_{i>m}$ and context pairs $\{(x_i, y_i)\}_{i=1}^m$. We trained four TNP models of different sizes (Table 1). Figure 15 shows how decision boundaries change as more in-context examples are added. Results indicate that TNPs learn smooth decision boundaries for non-linear tasks. Interestingly, we didn't observe a clear scaling law relating transformer size to decision boundary smoothness; smaller models often generalized better than larger ones.

Using Uncertainty-aware Active Learning to Smooth Decision Boundaries. We explore smoothing LLM decision boundaries using uncertainty-aware active learning. After an initial decision boundary is obtained, we query the LLM to identify uncertain points based on the entropy of class probabilities. We select the top-k most uncertain points, ensuring spatial diversity via greedy sampling, and add these as new in-context examples. As in Figure 16, this method yields smoother decision boundaries and higher test accuracies compared to random sampling.



(a) Decision boundaries **before** SFT on linear data of Llama3-8b across 4 tasks.

(b) Decision boundaries **after** SFT on linear data of Llama3-8b across 4 unseen tasks.

Figure 6: Generalization of Llama-3-8B after fine-tuning on a single binary linear classification task. After training, it generalizes to non-linear classification and 3-class and 4-class classification tasks.

4 Conclusion

We propose a novel approach to understanding in-context learning in LLMs by probing their decision boundaries in in-context learning in binary classification tasks. Despite achieving high test accuracy, we observe that the decision boundaries of LLMs are often irregularly non-smooth. Through extensive experiments, we identify factors that affect this decision boundary. We also explore fine-tuning and adaptive sampling methods, finding them effective in improving boundary smoothness. Our findings provide new insights into the mechanics of in-context learning and suggest pathways for further research and optimization.

References

R. Agarwal, A. Singh, L. M. Zhang, B. Bohnet, S. Chan, A. Anand, Z. Abbas, A. Nova, J. D. Co-Reyes, E. Chu, et al. Many-shot in-context learning. *arXiv preprint arXiv:2404.11018*, 2024.

K. Ahn, X. Cheng, H. Daneshmand, and S. Sra. Transformers learn to implement preconditioned gradient descent for in-context learning. *Advances in Neural Information Processing Systems*, 36, 2024.

E. Akyürek, D. Schuurmans, J. Andreas, T. Ma, and D. Zhou. What learning algorithm is in-context learning? investigations with linear models. In *The Eleventh International Conference on Learning Representations*.

Y. Bai, F. Chen, H. Wang, C. Xiong, and S. Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. *Advances in neural information processing systems*, 36, 2024.

A. Bertsch, M. Ivgi, U. Alon, J. Berant, M. R. Gormley, and G. Neubig. In-context learning with long-context models: An in-depth exploration. *arXiv preprint arXiv:2405.00200*, 2024.

T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

X. Chen, R. A. Chi, X. Wang, and D. Zhou. Premise order matters in reasoning with large language models. *arXiv preprint arXiv:2402.08939*, 2024.

A. Conneau and D. Kiela. Senteval: An evaluation toolkit for universal sentence representations. *arXiv preprint arXiv:1803.05449*, 2018.

D. Dai, Y. Sun, L. Dong, Y. Hao, S. Ma, Z. Sui, and F. Wei. Why can gpt learn in-context? language models secretly perform gradient descent as meta-optimizers. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4005–4019, 2023.

T. Dettmers, M. Lewis, Y. Belkada, and L. Zettlemoyer. Llm.int8(): 8-bit matrix multiplication for transformers at scale, 2022.

S. Garg, D. Tsipras, P. S. Liang, and G. Valiant. What can transformers learn in-context? a case study of simple function classes. *Advances in Neural Information Processing Systems*, 35:30583–30598, 2022.

E. J. Hu, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, et al. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2021.

A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

A. Lampinen, I. Dasgupta, S. Chan, K. Mathewson, M. Tessler, A. Creswell, J. McClelland, J. Wang, and F. Hill. Can language models learn from explanations in context? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 537–563, 2022.

M. Li, S. Gong, J. Feng, Y. Xu, J. Zhang, Z. Wu, and L. Kong. In-context learning with many demonstration examples. *arXiv preprint arXiv:2302.04931*, 2023a.

Y. Li, M. E. Ildiz, D. Papailiopoulos, and S. Oymak. Transformers as algorithms: Generalization and stability in in-context learning. In *International Conference on Machine Learning*, pages 19565–19594. PMLR, 2023b.

P. Malo, A. Sinha, P. Korhonen, J. Wallenius, and P. Takala. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, 65(4):782–796, 2014.

S. Min, M. Lewis, L. Zettlemoyer, and H. Hajishirzi. Metaicl: Learning to learn in context. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2791–2809, 2022a.

S. Min, X. Lyu, A. Holtzman, M. Artetxe, M. Lewis, H. Hajishirzi, and L. Zettlemoyer. 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, 2022b.

I. Mollas, Z. Chrysopoulou, S. Karlos, and G. Tsoumakas. Ethos: an online hate speech detection dataset. *arXiv preprint arXiv:2006.08328*, 2020.

S. Müller, N. Hollmann, S. P. Arango, J. Grabocka, and F. Hutter. Transformers can do bayesian inference. In *International Conference on Learning Representations*, 2021.

T. Nguyen and A. Grover. Transformer neural processes: Uncertainty-aware meta learning via sequence modeling. In *International Conference on Machine Learning*, pages 16569–16594. PMLR, 2022.

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. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

F. Shi, X. Chen, K. Misra, N. Scales, D. Dohan, E. H. Chi, N. Schärli, and D. Zhou. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pages 31210–31227. PMLR, 2023.

R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642, 2013.

H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

J. Von Oswald, E. Niklasson, E. Randazzo, J. Sacramento, A. Mordvintsev, A. Zhmoginov, and M. Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pages 35151–35174. PMLR, 2023.

A. Wang, Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, and S. Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32, 2019.

A. Webson and E. Pavlick. Do prompt-based models really understand the meaning of their prompts? In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2300–2344, 2022.

J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, et al. Emergent abilities of large language models. *Transactions on Machine Learning Research*, 2022.

J. Wei, J. Wei, Y. Tay, D. Tran, A. Webson, Y. Lu, X. Chen, H. Liu, D. Huang, D. Zhou, et al. Larger language models do in-context learning differently. *arXiv preprint arXiv:2303.03846*, 2023.

N. Wies, Y. Levine, and A. Shashua. The learnability of in-context learning. *Advances in Neural Information Processing Systems*, 36, 2024.

J. Wu, D. Zou, Z. Chen, V. Braverman, Q. Gu, and P. Bartlett. How many pretraining tasks are needed for in-context learning of linear regression? In *The Twelfth International Conference on Learning Representations*, 2023.

M. Xia, T. Gao, Z. Zeng, and D. Chen. Sheared llama: Accelerating language model pre-training via structured pruning. *arXiv preprint arXiv:2310.06694*, 2023.

R. Zhang, S. Frei, and P. L. Bartlett. Trained transformers learn linear models in-context. *arXiv preprint arXiv:2306.09927*, 2023.

X. Zhang, J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28, 2015.

A Background

Given the limited space in the main text, we provide background section in the Appendix here.

A.1 Training Large Language Models

Large Language Models (LLMs) are trained on vast corpora of text using unsupervised learning. During training, these models learn to predict the next token in a sequence. Given a sequence of tokens $(x_1, x_2, \ldots, x_{t-1})$, the model predicts the next token x_t by maximizing the likelihood $P(x_t|x_1, x_2, \ldots, x_{t-1})$. The training objective typically involves minimizing the cross-entropy loss:

$$L = -\sum_{i=1}^{N} \sum_{t=1}^{T_i} \log P(x_t|x_1, x_2, \ldots, x_{t-1}) \tag{1}$$

where T_i is the number of tokens in the i-th sequence and N is the total number of sequences in the corpus. During training, teacher forcing is often employed, where the model receives the ground truth token x_t as input at each time step instead of its own prediction, enabling parallel training.

A.2 In-Context Learning in LLMs

After training, LLMs can generalize to new tasks through a mechanism known as in-context learning. Let $\mathcal{S} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$ represent the set of n input-output pairs provided as examples in the prompt, where $\mathbf{x}_i$ is an input and y_i is the corresponding output. Given a new input $\mathbf{x}_{\text{new}}$, the LLM is turned into a task-specific model that predicts the output $\hat{y}_{\text{new}}$ by conditioning on the given examples: $P(\hat{y}_{\text{new}}|\mathbf{x}_{\text{new}}, \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\})$. In-context learning allows the LLM to perform tasks by leveraging the context provided by these examples, thereby inferring the task and generating appropriate responses for new inputs. This approach utilizes the model's ability to recognize patterns and apply learned knowledge without additional training or fine-tuning.

B Related Works

Understanding in-context learning in transformers and LLMs is an active area of research, with existing works approaching this problem from both theoretical and practical perspectives.

Theoretical understanding of in-context learning Recent works aim to establish a theoretical connection between in-context learning and gradient descent (GD). The pioneering work by Akyürek et al. proves transformers can implement learning algorithms for linear models based on GD and closed-form ridge regression by construction. Von Oswald et al. [2023] proves the equivalence between linear self-attention and GD on linear regression by construction. Similarly, Dai et al. [2023] shows that attention in transformers has a dual form of GD and views transformers as meta-optimizers. Subsequent works extend these ideas to characterize the global optimum of single-layer linear transformers. Ahn et al. [2024] observe that with the optimal parameters, the transformer implements a single step of preconditioned gradient descent, while Zhang et al. [2023] shows that at the global optimum, the transformer achieves competitive prediction error with the best linear predictor on a new prediction task. In addition to theoretical connections to GD, a complementary direction aims to establish statistical complexity and generalization bounds of in-context learning in transformers [Bai et al., 2024, Li et al., 2023b, Wies et al., 2024, Wu et al., 2023]. The common limitation of these existing theoretical frameworks is the reliance on strong assumptions about the transformer architecture or the functional form of the in-context learning tasks which may not reflect real-world practices.

Practical understanding of in-context learning More relevant to our paper is a line of works focusing on understanding the practical aspects of in-context learning in LLMs. Many existing works investigate the roles of in-context examples and prompts. Min et al. [2022b] show a surprising result that ground-truth demonstrations are not required for in-context learning, while other factors such as the label space, input text distribution, and overall sequence format play an important role. Shi et al. [2023] investigate the distractibility of LLMs and shows that their performance dramatically drops when irrelevant context is included. Subsequently, Wei et al. [2023] characterize these behaviors of LLMs with respect to model size, and show that larger language models perform in-context learning differently in the presence of flipped or semantically unrelated labels. Webson and Pavlick [2022] argue against the current practice of prompt engineering, showing that intentionally irrelevant or even pathologically misleading prompts achieve similar downstream performance to instructively good prompts. Orthogonally, Lampinen et al. [2022] find that including explanations in the in-context examples significantly improves the few-shot performance of LLMs. Finally, given the expanded context windows of modern LLMs, recent works have explored in-context learning in the many-shot setting with hundreds or thousands of examples [Agarwal et al., 2024, Li et al., 2023a, Bertsch et al., 2024].

Learning to learn in-context In contrast to the emergent in-context capabilities of LLMs, existing works have also studied methods that learn to perform in-context learning explicitly. Min et al. [2022a] propose MetaICL, a meta-training framework for finetuning pretrained LLMs to perform in-context learning on a large and diverse collection of tasks. MetaICL outperforms several baselines including emergent in-context learning and multi-task learning followed by zero-shot transfer. Going beyond the text domain, TNP [Nguyen and Grover, 2022] and PFNs [Müller et al., 2021] are two concurrent works that propose to train transformer models to perform in-context prediction for a family of functions, which allows in-context generalization to unseen functions after training. Similarly, [Garg et al., 2022] show that autoregressive transformers can be trained from scratch to learn function classes such as linear functions and 2-layer ReLU networks. These works present an interesting set of baselines for our work to examine the in-context learning ability of LLMs.

C Traditional Classifiers Model Details

In our experiments, we used several classical machine learning models with the following hyperparameters:

- **Decision Tree Classifier:** We set the maximum depth of the tree to 3.
- **Multi-Layer Perceptron:** The neural network consists of two hidden layers, each with 256 neurons, and the maximum number of iterations is set to 1000.
- **K-Nearest Neighbors:** The number of neighbors is set to 5.
- **Support Vector Machine (SVM):** We used a radial basis function (RBF) kernel with a gamma value of 0.2.

D Classification Datasets Creation Details

We use three types of classification tasks from `scikit-learn` [Pedregosa et al., 2011] to probe the decision boundary of LLMs and transformers: linear, circle, and moon classification problems. For linear classification tasks, we utilize the `make_classification` function, which generates random classification problems by creating clusters of points normally distributed around the vertices of a hypercube with sides of length $2 \times$ class_sep. Circle classification tasks are generated using the `make_circles` function, creating a binary classification problem with a large circle containing a smaller circle. The `factor` parameter controls the scale of the inner circle relative to the outer circle. Moon classification tasks are generated using the `make_moons` function, creating a binary classification problem with two interleaving half circles. The `noise` parameter controls the standard deviation of Gaussian noise added to the data points.

For training tasks, the `class_sep` parameter is randomly sampled from the range $[1.5, 2]$, and the `factor` parameter for circular tasks is sampled from $[0.1, 0.4]$. For testing tasks, the `class_sep` parameter is sampled from $[1, 1.4]$, and the `factor` parameter from $[0.5, 0.9]$, ensuring that testing tasks differ from training tasks. The `noise` parameter for moon-shaped tasks is sampled from $[0.05, 0.1]$ for training and $[0.1, 0.2]$ for testing, introducing varying levels of complexity in the classification problems.

E Classification Results on Multi-Class NLP Classification Tasks

We extend our analysis to multi-class NLP classification tasks using high-dimensional real-world datasets. To address the challenge of visualizing high-dimensional text embeddings, we project them onto a 2D space using t-SNE. While any dimensionality reduction technique inevitably introduces confounding factors, this approach allows us to extend our analysis to more complex, real-world scenarios. Our experiments encompass six widely-used NLP classification tasks, covering both binary and multi-class settings. These include Subjective/Obejective sentence classification (SUBJ) [Conneau and Kiela, 2018], financial sentiment analysis (FP) [Malo et al., 2014], textual entailment recognition (RTE) [Wang et al., 2019], hate speech detection (ETHOS) [Mollas et al., 2020], sentiment analysis (SST-2) [Socher et al., 2013] and news topic classification (AG_NEWS) [Zhang et al., 2015]. We provide a broader perspective on the applicability of our approach. The results, presented in Figure 7, demonstrate that the non-smooth decision boundary characteristics observed in our synthetic datasets persist in these more complex NLP tasks.

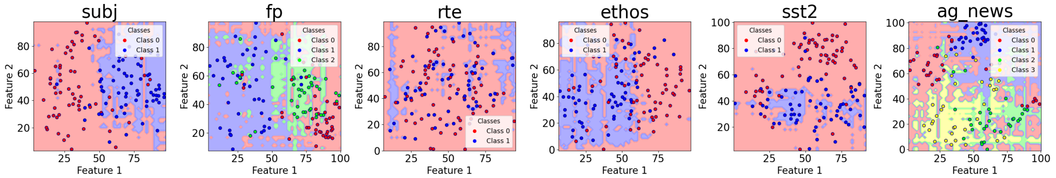


Figure 7: Decision boundaries of Llama-3-8b on six NLP tasks, ranging from binary to multi-class classification. Since text embeddings are natively high-dimensional, we projected text embeddings onto a 2D space using t-SNE. The irregular, non-smooth behaviors are also seen in these tasks.

F Factors affecting the decision boundary

F.1 How number of in-context examples affect the classification accuracy?

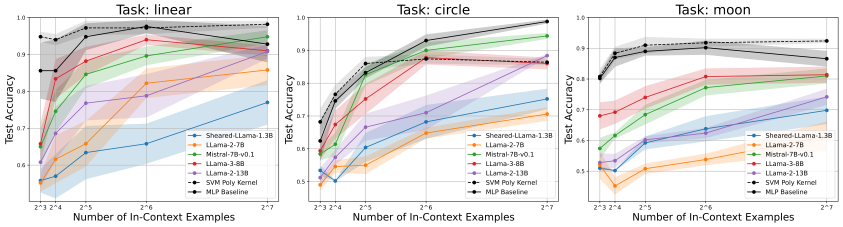
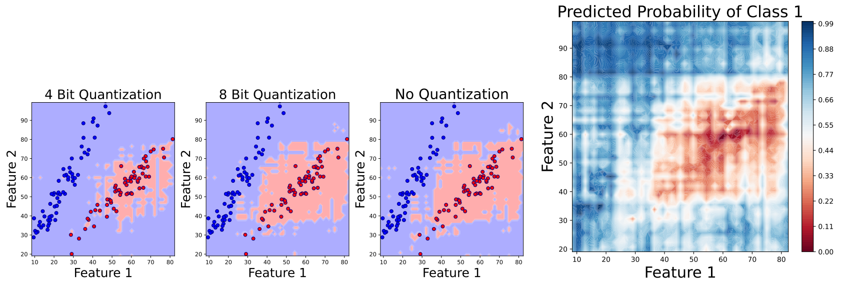


Figure 8: In-context test accuracy for LLMs and baselines across three classification tasks (linear, circle, and moon), with each subplot illustrating the test accuracy as the number of in-context examples increases. The baselines are the SVM with a polynomial kernel and the MLP with two hidden layers. Shaded regions represent the standard error of the mean accuracy across 5 seeds.

F.2 How Quantization Affects the Decision Boundary?



(a) Decision boundaries of Llama-2-7b with different quantization choices on a linearly separable task.

(b) Prediction of probability of class 1 with 8-bit quantization.

Figure 9: Impact of quantization on Llama2-7b's decision boundaries and probability predictions.

F.3 How decision boundary scale with more in-context learning examples?

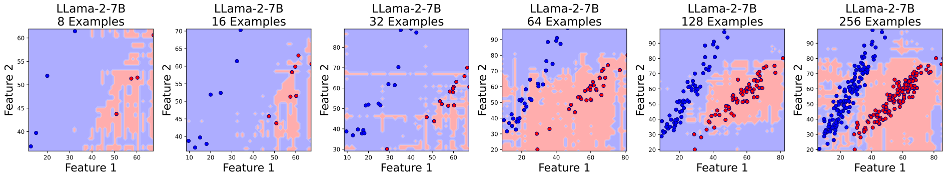


Figure 10: Decision boundary of Llama2-7b with increasing in-context examples from 8 to 256.

F.4 Are Decision Boundaries Sensitive to the Order of In-Context Learning Examples?

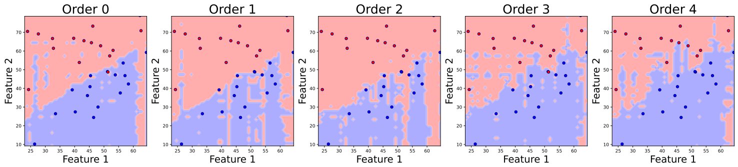


Figure 11: The sensitivity of the Llama3-8b model's decision boundary to the order of in-context examples. Each subplot (Order 0 to Order 4) shows the model's decision boundary with the same 32 examples shuffled differently.

G Pretrained LLMs decision boundary on linear and non-linear classification tasks

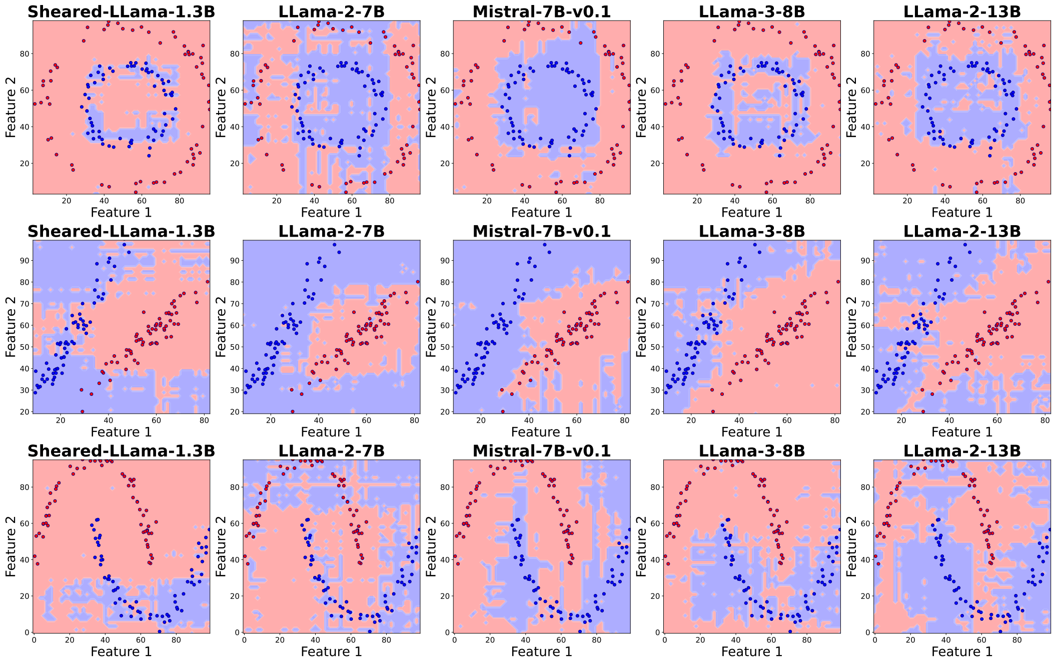


Figure 12: Visualizations of decision boundaries for various LLMs, ranging in size from 1.3B to 13B, on three classification tasks. The tasks are, from top to bottom, circle, linear, and moon classifications. Note that the circle and moon tasks are not linearly separable. The in-context data points are shown as scatter points and the colors indicate the label determined by each model. These decision boundaries are obtained using 128 in-context examples. The visualization highlights that the decision boundaries of these language models are not smooth.

H Fintuening Experiment Details

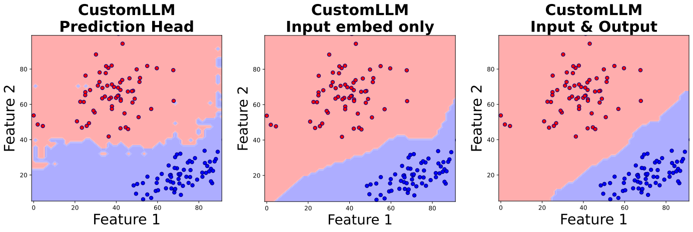
H.1 Can We Finetune LLMs on a Dataset of Classification Tasks to Achieve Smoother Decision Boundaries?

Previous works have shown that finetuning a pretrained LLM on a large collection of tasks improves its in-context learning performance on unseen tasks [Min et al., 2022a]. In this section, we investigate if the same paradigm helps improve the decision boundary smoothness of LLMs. To do this, we finetune a pretrained Llama model [Touvron et al., 2023] on a set of 1000 binary classification tasks generated from `scikit-learn` [Pedregosa et al., 2011], where the ground-truth decision boundary is either linear, circle-shaped, or moon-shaped, with equal probabilities. For each task, we sample randomly $N = 256$ data points $x \sim \mathbf{X}_{\text{grid}}$ and their corresponding label $y's$. We then sample the number of context points $m \sim \mathcal{U}[8, 128]$, and finetune the LLM to predict $y_{i>m}$ given $x_{i>m}$ and the preceding examples:

$$\mathcal{L}(\pi) = \mathbb{E}\left[\sum_{i=m+1}^{N} \log p(y_i \mid x_i, x_{1:i-1}, y_{1:i-1})\right], \tag{2}$$

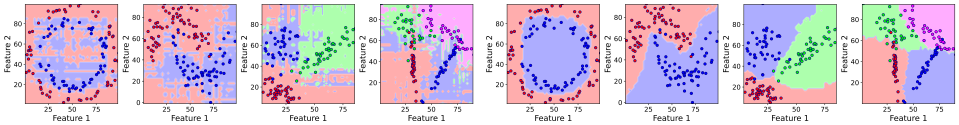
where the expectation is with respect to task, data points $\{(x_i, y_i)\}_{i=1}^{N}$, and the number of context points m. After training, we evaluate the same finetuned model on various binary classification tasks with varying numbers of context points. To ensure the test tasks are unseen during training, we use different parameters in creating the datasets, such as the separateness between two classes and the scale between the inner and outer circles in the circle task. See Appendix D for more details.

We consider several finetuning settings for ablation studies. 1) In the first setting, we finetune the pretrained LLM using LoRA [Hu et al., 2021] and finetune the attention layers. 2) We finetune only the token embedding layer of LLM. 3) We finetune only the linear head layer of LLM. Then we consider modifying the architecture of the LLM: In this setting, we keep the core transformer backbone of the LLM frozen, attach randomly initialized embedding layers and prediction head to the model, and train the entire model using objective (2). This stems from the intuition that task-specific embedding and prediction layers allow the model to maximally utlize the general pattern-matching capabilities of the transformer backbone for the new task. We refer to this model as CustomLLM, and consider its three variants, which add 1) a new embedding layer for x, 2) a new prediction head for y, and 3) new embedding layers for x, y, and a new prediction head for y. The embedding layers and prediction head are MLPs with one hidden layer. We embed the raw numerical values instead of the text representation of x whenever a new embeddding layer for x is used (same for y), and predict directly the binary class values instead of text labels whenever the new prediction head is used. Results of Finetuning LLM and CustomLLM in Figure 5 and Figure 13 show that finetuning the intermediate and earlier embedding layers leads to smoother decision boundary compared to finetuning the top prediction head.



(a) Prediction head (b) Input embedding layer (c) Both layers

Figure 13: CustomLLM finetuning ablations. Decision boundary after finetuning the prediction head, input embedding layer, and both layers for the CustomLLM.



(a) Decision boundaries **before** SFT on linear data of Llama3-8b across 4 tasks.

(b) Decision boundaries **after** SFT on linear data of Llama3-8b across 4 unseen tasks.

Figure 14: Generalization ability of Llama-3-8B after supervised fine-tuning on a single binary linear classification task. The first two columns show the model's performance on non-linear classification tasks before and after fine-tuning, while the last two columns demonstrate its ability to generalize to 3-class and 4-class classification tasks.

H.2 Can LLMs finetuned on one in-context learning task generalize to more complex in-context learning tasks?

We demonstrated that SFT on the dataset can smooth the decision boundary on that dataset. In this section, we further explore whether a LLM fine-tuned only on a linear task can achiever smoother decision boundaries on unseen and more complex tasks. As shown in Figure 14, we compare the decision boundaries of Llama3-8b before and after SFT on the linear task only. Unexpectedly, we found it generalizes to unseen non-linear tasks as well as 3-class and 4-class classification tasks, despite only being trained on a binary linear task. The smoother decision boundaries observed in these unseen tasks suggest that fine-tuning on a synthetic in-context learning task can have downstream benefits for other tasks, enabling the model to be more robust in in-context learning.

H.3 Can we train a transformer from scratch to learn smooth decision boundary in-context?

One may wonder whether a small transformer trained from scratch can provide smooth decision boundaries. To answer this, we train TNPs [Nguyen and Grover, 2022] , a transformer-based model specifically designed for in-context learning. For each sequence of data points $\{(x_i, y_i)\}_{i=1}^{N}$ from a task C, TNPs learn to predict the query labels $y_{i>m}$ given the query inputs $x_{i>m}$ and the context pairs, assuming conditional independence among the queries given the context:

$$\mathcal{L}(\theta) = \mathbb{E}\left[\sum_{i=m+1}^{N} \log p(y_i \mid x_i, x_{1:m}, y_{1:m})\right], \tag{3}$$

where the expectation is with respect to task C, data points $\{(x_i, y_i)\}_{i=1}^{N}$, and the number of context points m. TNPs employ a specialized mask to ensure the conditional independence assumption.

We trained TNP models of four different sizes as shown in the Table 1 below. We plot how does the TNP models decision boudnary changes as more in-context examples are added in Figure 15. TNP models learn smooth deicision boundary for this moon-shaped non-linear task. And we did not observe a scaling law of transformer sizes versus the decision boundary smoothness. In contrast the smaller model generalize better than the larger model.

Table 1: TNP transformers model sizes and architectures.

Model	**Parameters (M)**	**Input embed dim**	**feedforward dim**	**num heads**	**num layers**
Small	0.1	64	64	2	3
Medium	0.6	128	128	4	6
Large	1.6	128	256	8	12
Larger	9.7	256	512	16	18

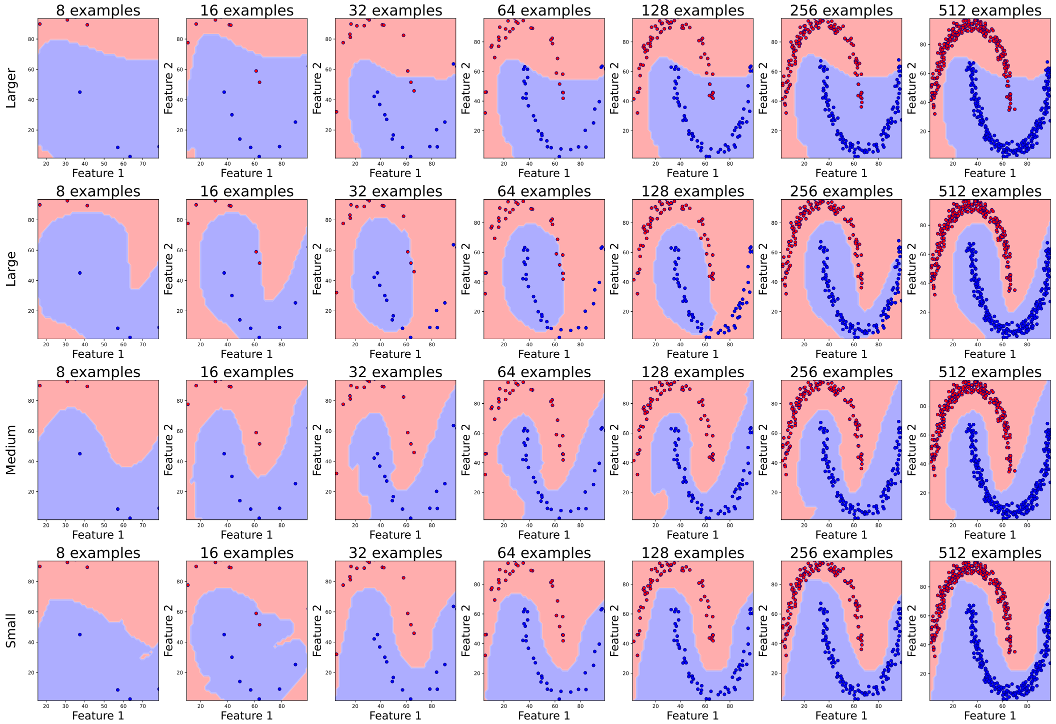


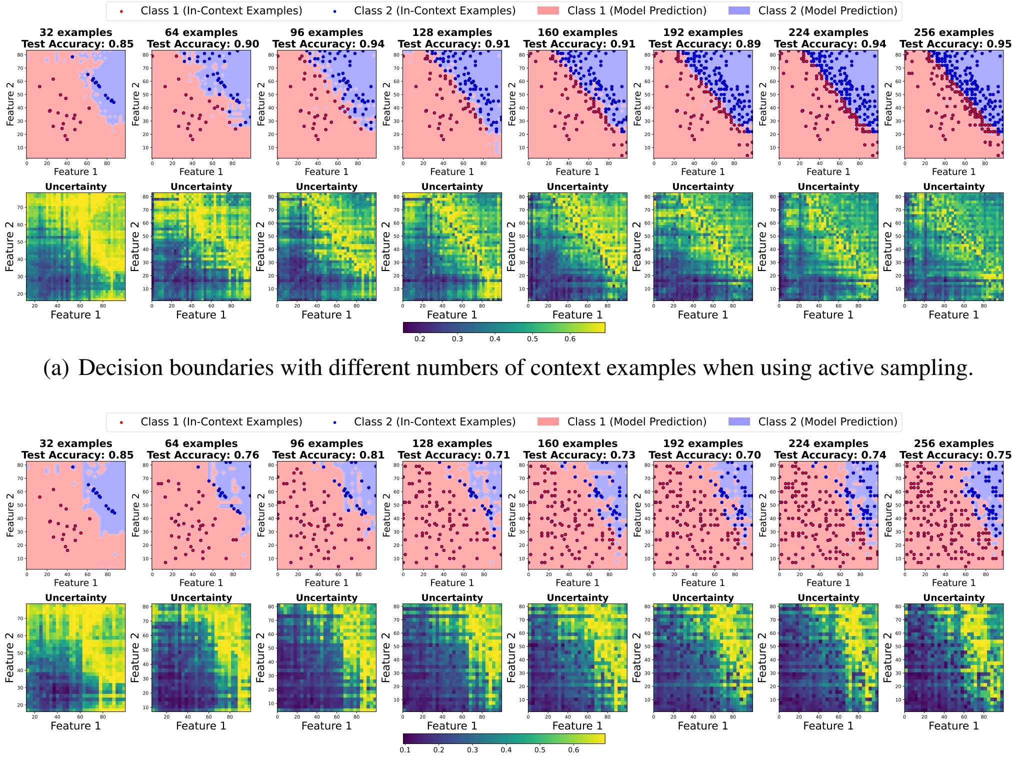
Figure 15: Decision boundary of TNP models of different model sizes.

H.4 How to Use Uncertainty-aware Active Learning to Learn Decision Boundaries

We investigate whether the decision boundary can be smoothed by providing the LLM with labels of the most uncertain points on the grid as additional in-context examples. Uncertainty is measured as the entropy of the probability distribution of the two classes after softmax normalization of the logits. Our study focuses on an active learning scheme where new in-context examples are incrementally added based on the LLM's current uncertainty. Initially, we obtain the decision boundary conditioned on the existing in-context examples. To refine this boundary, we query the LLM over a grid and select the top-k most uncertain points, ensuring they are spatially distant from each other using a greedy sampling approach. For labeling these uncertain points, we use a logistic regression model well-trained on a larger dataset with perfect accuracy as the ground truth decision boundary. As shown in Figure 16, this uncertainty-aware active sampling method results in a smoother decision boundary over iterations compared to random sampling. The iterative refinement enhances the model's generalization capabilities, leading to higher test set accuracies and greater sample efficiency, requiring fewer additional in-context examples to achieve performance gains. These findings indicate that leveraging the LLM's uncertainty measurements is valuable for selecting new in-context examples in resource-constrained settings where labeled data is scarce. We show more examples in below:

I SFT LLMs for in-context classification

We used LoRA [Hu et al., 2021] to supervise fine-tune the Llama series models on both non-linear and linear classification tasks, including circle, linear, and moon datasets. The models fine-tuned are Sheared-Llama-1.3B, Llama2-7B, Llama2-13B, and Llama3-8B. Visualization in Figure 19 demonstrates that these language models produce smoother decision boundaries after training on the classification datasets using SFT.



(a) Decision boundaries with different numbers of context examples when using active sampling.

(b) Decision boundaries with different numbers of context examples when using random sampling.

Figure 16: Comparison of active and random sampling methods. We plot the decision boundaries and uncertainty plot across different number of in-context examples from 32 to 256, where the in-context examples are gradually added to the prompt using active or random methods. Active sampling gives smoother decision boundary and the uncertain points lie on it. The test set accuracies is plotted in the titles.

J Finetune on in-context examples only

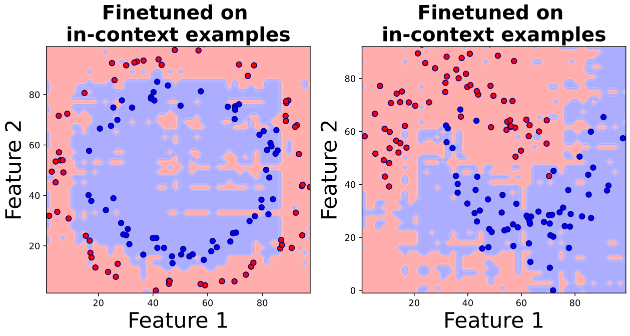


Figure 20: Two examples of Llama2-7B finetuned on the in-context examples points, which are scattered points in the plot.

K Prompt Format for binary classification

The prompt format we used in our experiments to query the classification result is shown in Figure 21.

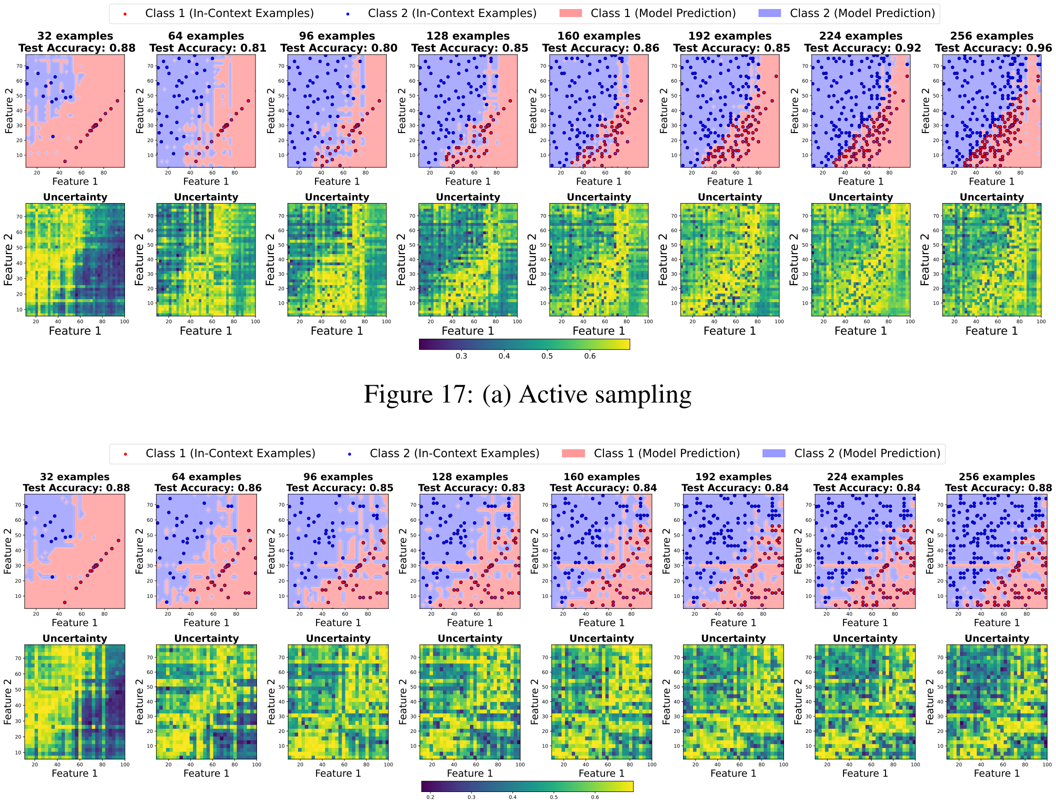


Figure 17: (a) Active sampling

(b) Random sampling

Comparison of decision boundaries of uncertainty-based actively sampling and randomly sampling in-context examples. Example 1.

L Limitation

One limitation of our study is the focus on demonstrating mainly binary classification tasks. Limited by the available compute, we chose binary tasks and also for better qualitative reasoning. However, we also extended our experiments to tasks with four classes and found that our methods generalize to multi-class classification and other more complex tasks. Additionally, the exploration of fine-tuning and adaptive sampling methods, although effective in our experiments, may not be universally applicable across closed-source LLMs that do not allow access to logits. Future work should consider a broader range of tasks and datasets, as well as a more diverse set of LLM models, to validate and extend our findings.

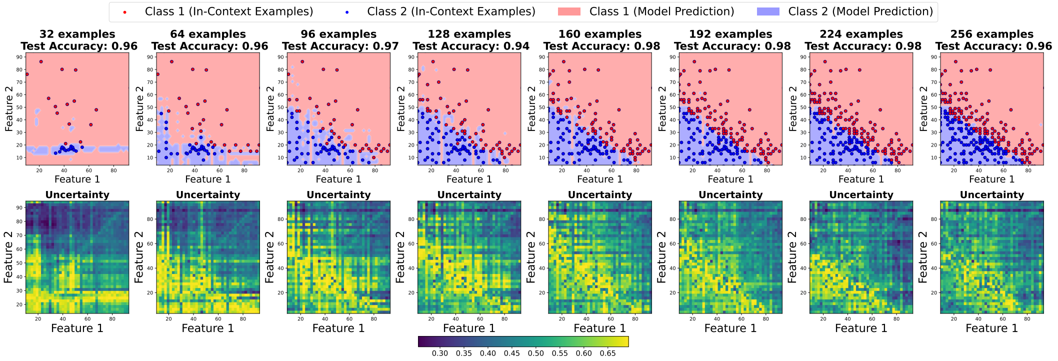
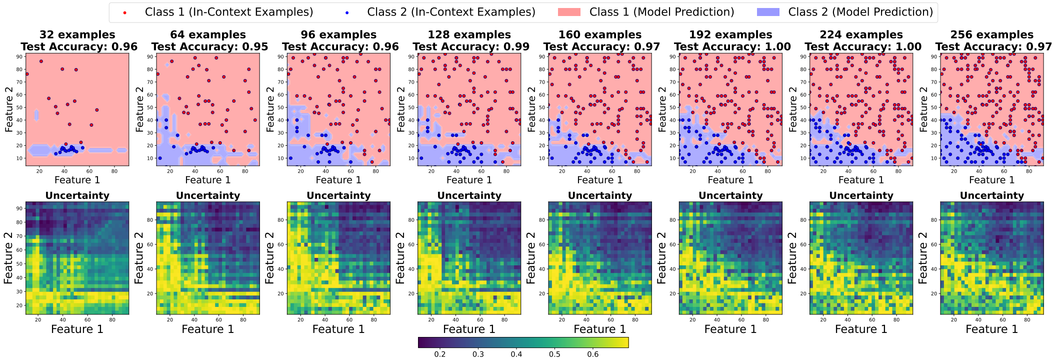
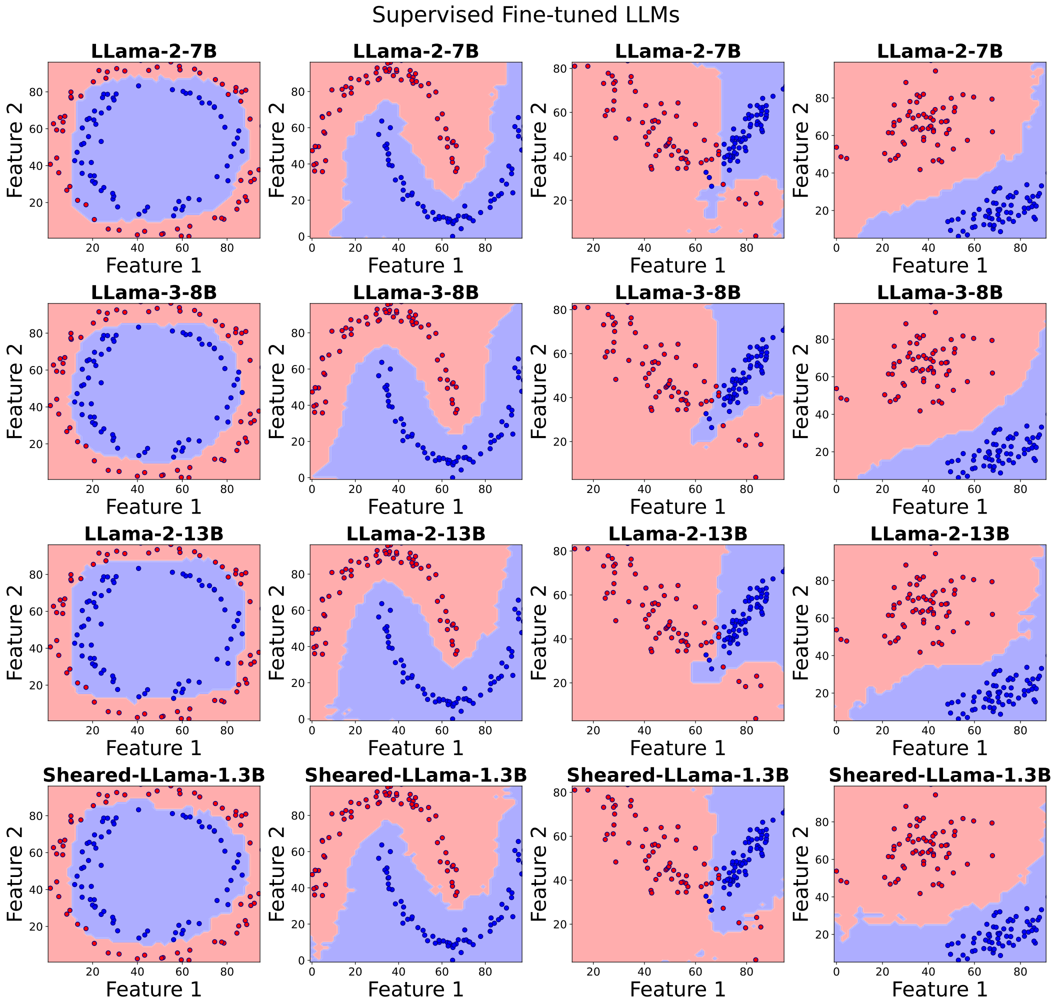


Figure 18: (a) Active sampling



(b) Random sampling

Comparison of decision boundaries of uncertainty-based actively sampling and randomly sampling in-context examples. Example 2.



Supervised Fine-tuned LLMs

LLama-2-7B

LLama-3-8B

LLama-2-13B

Sheared-LLama-1.3B

Figure 19: Decision boundary of in-context learning on 128 examples across Llama series models after supervised finetuning with LoRA.

```
Given pairs of numbers and their labels, predict the label for a new
input pair of numbers based on the provided data.
Answer with only one of the labels ‘Foo’ and ‘Bar’:

Input:  64 24
Label:  Bar
Input:  34 41
Label:  Bar
Input:  71 66
Label:  Bar
...
Input:  96 49
Label:  Foo
Input:  21 56
Label:  Foo

What is the label for this input?
Input:  2 3
Label:
```

Figure 21: Few-shot in-context prompt with n context questions.