Investigating inductive bias in LLMs - A case study in creditworthiness assessment*

Thi Hoang Anh Tran[1,*,†], Emiel Caron[2] and Hans Weigand[3]

[1]Tilburg University, Warandelaan 2, 5037 AB Tilburg, the Netherlands

Abstract

Large Language Models (LLMs) have been increasingly adopted in solving tasks across domains and modalities thanks to their generalization capability. Since LLMs are trained on massive datasets, they achieve a "saturated" test loss close to zero. As a result, two models can behave differently even though they achieve similar loss, meaning that they are functionally non-identifiable. This study introduces a fine-tuning approach to probe LLMs' inductive bias and understand how models learn decision rationale in the context of creditworthiness assessment. We fine-tune an LLM on both prediction and explanation using tabular inputs and explanation instructions generated by the Explainable Boosting Machine (EBM) judge. By decomposing loss for each task and controlling the granularity of explanation instructions, we assess how the inductive bias can be distilled to LLMs through fine-tuning and whether LLMs can learn instance-specific rationales that produce those outputs.

Keywords

Large Language Models, inductive bias, explanation, financial LLMs, creditworthiness

1. Introduction

As artificial intelligence systems become more integrated into decision-making processes, research has increasingly focused on explainable AI (XAI) to enhance human–AI complementary performance, while the recent emergence of Large Language Models (LLMs) such as ChatGPT has further broadened the scope of AI applications across diverse domains. Research on explaining LLM generalization capability could be divided into two regimes: interpolation and saturation. The interpolation regime focuses on overparameterized models that perfectly fit the training data and thus generalize well to in-distribution test data. While the interpolation regime focuses on the transition from training data to test data, the saturation regime assumes the LLMs have already achieved near-zero loss on both training and test sets due to massive training data. Because the test loss is saturated, it becomes "agnostic" to the model's internal logic, meaning that models become functionally non-identifiable [1].

Inductive bias offers one approach to address functional non-identifiability by answering the question: "Which rules did the model learn?" [1] Traditional machine learning (ML) views inductive bias as a preference for simplicity, yet in LLMs, it should be considered throughout the model development pipeline rather than only in the model architecture [2].

This study focuses on probing LLMs' inductive bias to understand "Which rules did the model learn?" during fine-tuning with tabular input. We control the rule by fine-tuning instructions generated by an interpretable machine learning judge (ML judge) and evaluating LLMs' inductive bias by measuring their alignment with the ML judge. Specifically, the instructions are formulated as a dual-task (prediction and explanation), and fine-tuning is implemented in a lightweight manner, preserving the architecture of the pre-trained model. We use a case study in creditworthiness assessment to illustrate the inductive probing pipeline. We aim to answer two research questions:

RQ1: How task-specific inductive bias is distilled to LLMs through supervised fine-tuning?

RQ2: How dual-task objective fine-tuning scheme resolve the functional non-identifiability of LLMs?

Late-breaking work, Demos and Doctoral Consortium, colocated with the 4th World Conference on eXplainable Artificial Intelligence: July 01–03, 2026, Fortaleza, Brazil

*Corresponding author.

t.h.a.tran@tilburguniversity.edu (T. H. A. Tran); e.a.m.caron@tilburguniversity.edu (E. Caron); h.weigand@tilburguniversity.edu (H. Weigand)

0000-0001-7310-9371 (T. H. A. Tran); 0009-0002-0760-7377 (E. Caron); 0000-0002-6035-9045 (H. Weigand)

© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

published 2026-09-10

CEUR-WS.org/Vol-4259/paper9.pdf

This study contributes to a deeper understanding of LLMs' learning capabilities for emulating ensemble decision boundaries. In a previous study [3], inductive bias of complex decision boundaries is implicitly injected into LLMs through the predicted labels in fine-tuning instructions. In this study, we explicitly introduce both predicted labels and decision logic in fine-tuning instructions to examine how well LLMs emulate complex decision rules. The instance-specific reasoning is important in high-stakes domains such as finance, which, to the best of our knowledge, have not been studied empirically [4]. In addition, we provide a fine-tuning scheme to resolve LLMs functional non-identifiability issue. By formulating a dual-task objective and decomposing the training loss for each task, we distinguish whether models learn logical rules during fine-tuning or resort to shortcuts in making predictions afterward.

2. Theoretical Background

Mitchell (1977) [5] defines inductive bias as the set of prior assumptions or hypotheses a ML algorithm uses to predict outputs for inputs it has never encountered. Mitchell distinguishes between restriction bias, i.e., architectural limits on the hypothesis space, and preference bias, i.e., the criteria for choosing one consistent hypothesis over another. Solomonoff [6] formalizes Mitchell's "preference bias" as a mathematical theory for quantifying simplicity [7]. Under this framework, inductive inference is a weighted average of all possible models, favoring the simplest ones. Empirical studies from Dingle et al. (2018) [7] show that real-world data, regardless of their domain, have low algorithmic complexity and are compressible. This low complexity aligns with the inner simplicity bias of neural network models, including Transformer models, enabling them to generalize across seemingly unrelated domains.

Reizinger et al. (2024) [1] argue that statistical generalization, as indicated by low loss, does not always translate into strong downstream performance for LLMs. Specifically, when the loss saturates near zero across the training and test sets and becomes agnostic to the model's internal logic. In other words, multiple models trained with different parameters can achieve the same minimal test loss while exhibiting radically different behavior on out-of-distribution (OOD) data. Inductive bias then acts as a tie-breaker, nudging training toward a particular solution. It is therefore necessary to examine which inductive biases drive transferability and OOD performance. Budnikov et al. (2025) [2] argue that inductive bias is not a static property of the architecture (see [6]) but is shaped throughout different stages of the model's life cycle, including data preparation, training, and inference.

Understanding the connection between inductive bias and generalization has implications for the adoption of LLMs across modalities. Lu et al. (2021) and Dinh et al. (2022) together prove that minimal fine-tuning leverages the inductive bias of pre-trained Transformers to perform diverse tasks such as numerical computation, prediction, vision, and classification [8, 3]. Fine-tuning LLMs has been employed in the financial domain, but is more popular in natural language tasks (sentiment analysis, question answering, named-entity recognition) [9] than in machine learning tasks (credit risk assessment) [4]. Feng et al. (2024) demonstrate that minimal fine-tuning of LLMs on tabular data can achieve generalization on par with traditional expert systems in credit assessment [4].

3. Methodology

The study employs supervised fine-tuning experiments to probe LLMs' inductive bias, as this process alters the model's weights and can affect generalization behavior. In addition, because LLMs always generate explanations, it is difficult to distinguish between task-specific and pre-trained inductive biases without strict controls.

Figure 1 shows the experimental setup for this study, which consists of three phases: constructing fine-tuning prompts, supervised fine-tuning, and inference and evaluation. The fine-tuning prompts include two tasks: (1) prediction and (2) explanation, where the targets of these tasks are provided by an interpretable machine learning model, i.e., ML judge. The prediction task is formulated as a binary classification. The explanation task is formulated as the top 15 features that contributed to the decision, ranked in ascending order of importance. Fine-tuning instructions vary in the level of

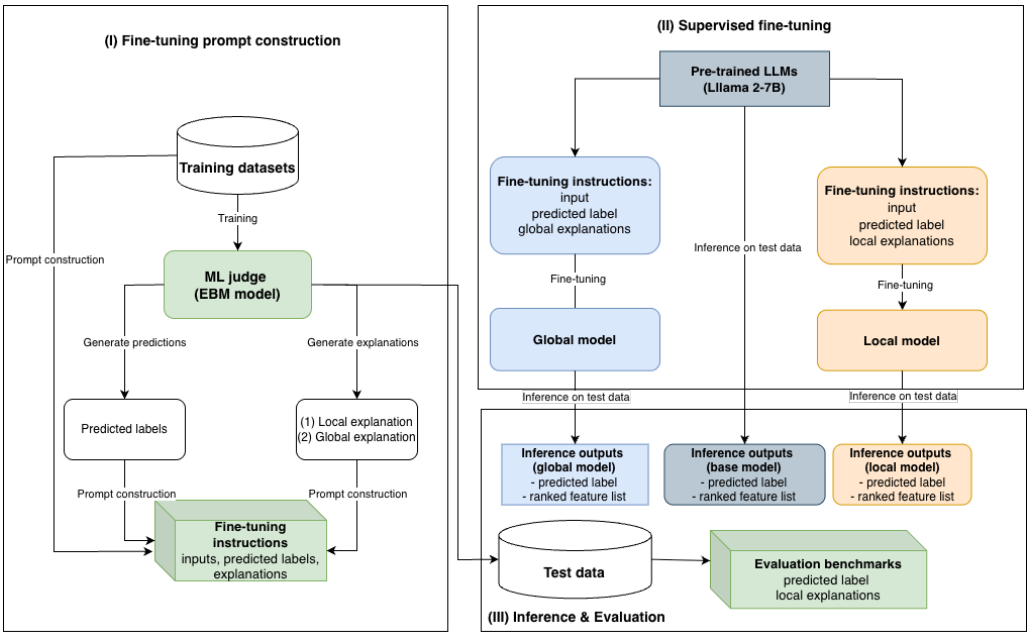


Figure 1: Experiment design. The high-resolution figure is available in this link.

granularity of their explanations, resulting in three model variants: a pre-trained model, a fine-tuning model trained on global explanation instructions (global_model), and a fine-tuning model trained on local explanation instructions (local_model). During the inference phases, three models receive the same inference prompts and are evaluated using the same benchmarks generated by the ML judge.

Data We employ a creditworthiness assessment for implementing the experiment. Three public datasets, namely, German Credit, Australian, and LendingClub [1]. These public datasets have been widely used for credit assessment in prior research on creditworthiness, including fine-tuning credit-scoring LLMs in Feng et al. (2024) [4]. Table 1 summarizes the basic information about these datasets and their use in each experiment.

Table 1
Dataset summary. For the LendingClub dataset target label includes two statuses: "Fully paid" and "Charge Off".

Dataset	# Rows	# Attributes	Attribute types	Target Label	Train/Validation/Test split
German Credit	1,000	20	7 numeric and 13 categorical	700 Good/ 300 Bad	700/100/200
Australian	690	14	6 numerical and 8 categorical	307 Good/ 383 Bad	482/69/139
LendingClub*	13,453	21	14 numeric and 7 categorical	10,767 Good/ 2,686 Bad	9,417/1,345/2,691

The datasets' attributes are classified into three categories: customer soft information (address, age, gender, income, marriage status, employment, etc.), loan information (amount, purpose, due date, interest rates, etc.), and credit history (savings, previous loan, house or car ownership, delinquency, etc.) Each dataset requires an additional step to convert tabular input into meaningful text.

Stage I: Fine-tuning prompt construction Fine-tuning prompts consist of (1) input data, (2) predicted labels, and (3) the explanations (local or global) generated by the ML judge. Figure 2 shows

[1] German Credit and Australian are retrieved from the UCI repository, and LendingClub is retrieved from Kaggle

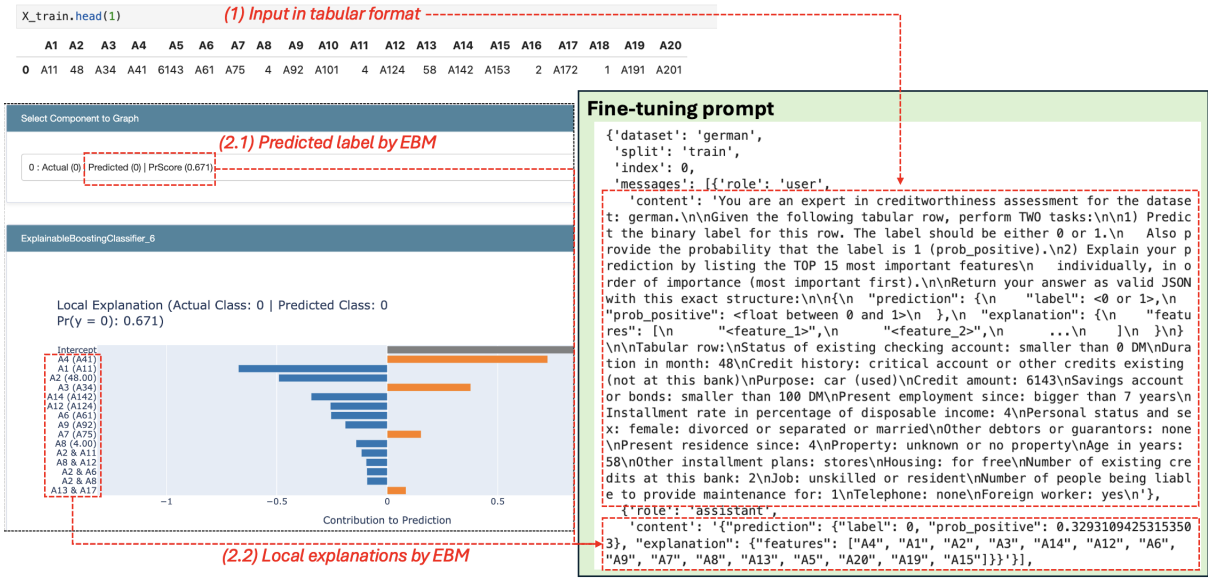


Figure 2: Example fine-tuning prompt for dataset German Credit. Fine-tuning targets include the predicted label and the local EBM explanation. EBM default explanation displays the first 15 terms on the vertical axis, including individual features (e.g., A1, A2, etc) and feature interactions (e.g., "A2 & A11", "A8 & A12"). For the preliminary experiment, only individual feature contributions are considered, excluding feature interactions and the direction of the contribution (positive/negative). In this example, the complete local explanation is [A4, A1, A2, A3, A14, A12, A6, A9, A7, A8, A13, A5, A20, A19, A15]. The high-resolution figure is available in this link.

an example of fine-tuning prompt construction. Tabular input is converted to a textual format by interpreting the column's meaning and the categorical attribute's meaning. The Explainable Boosting Machine (EBM) algorithm is employed to train an ML judge. We do not use post-hoc explanation methods such as LIME [10] or SHAP [11] to generate instructions, because these methods approximate the decision rationale of black-box models using surrogate models. By contrast, EBM is inherently interpretable, thereby ensuring that explanations faithfully represent the decision mechanism. EBM is developed by Lou et al. (2012) [12] and Nori et al. (2019) [13] with the general form:

$$g(E[y]) = \beta_0 + \Sigma f_i(x_i) + \Sigma f_{i,j}(x_i, x_j) \tag{1}$$

Where g is the link function that adapts the model to different settings, such as classification or regression; f_i is the feature function, and $f_{i,j}$ is the function of pairwise interaction between features. EBM combines inductive bias from linear and tree-based algorithms: it learns the feature function f_i using boosted decision trees and retains the linear additive final form. Thus, EBM aligns with the learning capabilities of Transformer models [3]. EBM performs on par with other black-box ensemble algorithms, such as bagged and boosted trees, while generating both global-level and local-level explanations in terms of feature contribution [12]. Local explanations are calculated for each instance (as shown in figure 2), whereas the global explanation is the overall feature contribution across all instances.

Three EBM models are trained separately on three datasets. For fine-tuning, we use predicted labels, local explanation, and global explanation generated on the training partitions. For evaluation, we use the true labels, predicted labels, and local explanations on the testing partition as benchmarks.

Stage II: Supervised fine-tuning Our fine-tuning pipeline adopts previous work by Dinh et al. (2022) - LIFT framework [3] and Feng et al. (2024) - CALM [4]. We adopt the LIFT framework by casting tasks as a causal language modeling objective, using templated prompts and standard token-level cross-entropy. We adopt CALM by using the same pretrained model Llama2-7B-chat [2] and using Low-Rank Adaptation (LoRA) [14] for parameter-efficient fine-tuning.

[2]https://huggingface.co/meta-llama/Llama-2-7b

Our fine-tuning pipeline includes some new extensions. First, because the assistant's responses require both prediction and explanation, we apply a structured JSON schema. The JSON schema for "prediction" includes a binary label and a probability, and that for "explanation" includes an ordered list of 15 features. Neither LIFT nor CALM explicitly supervises structured explanation ranking. In the LIFT framework, authors use a general template for both instruction and inference, whereas in CALM, authors limit answers to only prediction. Second, we keep the training similar to LIFT but perform additional computations to monitor the learning process for each sub-task. Specifically, the optimized objective is the average token-level cross-entropy across all tokens in the assistant's response ($\mathcal{L}_{\text{assistant}}$). Prediction-only loss ($\mathcal{L}_{\text{pred}}$) and explanation-only loss ($\mathcal{L}_{\text{expl}}$) are computed on corresponding subsets of assistant token (inside prediction JSON, explanation JSON). These decomposed losses are logged during training to monitor the evolution of the explanation and prediction tasks.

Finetuning is performed for 1 complete epoch with approximately 10,600 training examples. Inputs are tokenized up to a maximum sequence length of 1,024 tokens. Evaluation and checkpointing are performed every 50 steps, keeping the three most recent checkpoints. We use a LoRA rank r=16, scaling factor α = 32, dropout p = 0.05, with no bias parameters adapted.

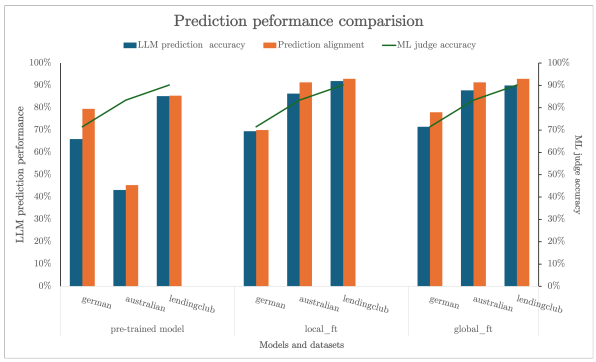
Stage III: Inference and evaluation During inference, the baseline pre-trained model and the two fine-tuned models are given the same prompts, wrapped in the "Instruction ... => Response:" format. The Instruction part of the prompt includes the dataset name, task description, the JSON schema for each task's output, and input information for each case. The raw answers from LLMs are the text generated after "Response:", which undergo further parsing to produce the final output for evaluation. We set the temperature to 0 and top-p to 1 to reduce sampling noise and to make the answers deterministic. In addition, due to computing resource limitations, we perform inference on a sample of 500 instances from the test partition of the LendingClub dataset (2,691 instances). For other datasets, we perform full inference on the test partitions (German Credit: 200 instances; Australian: 139 instances).

We assess the inductive bias distilled to LLMs by how well the fine-tuned model aligns with the ML judge on each task: prediction and explanation. Specifically, we compare LLM predictions with the ML judge's prediction to assess how well LLMs learn the task from the ML judge, using prediction alignment (similar to [3]). In addition, we compare the LLM's predictions with the true labels to assess overall generalization capability, measured by prediction accuracy. For the explanation task, we use four metrics to assess the alignment between retrieval information (LLMs' explanations) and benchmarks (local explanation by ML judge): precision@k, position match@k, Spearman's rank correlation, and NDCG. *Precision@k* measures the fraction of the top-k predicted features that appear in the ground-truth set (order-insensitive). *Position match@k* measures exact agreement at each rank position within the top-k list (strictly order-sensitive). *Spearman's rank correlation* assesses whether the ranking is approximately monotonic with respect to the ML teacher. *NDCG@k* assesses whether the model ranks important features near the top of their explanations.

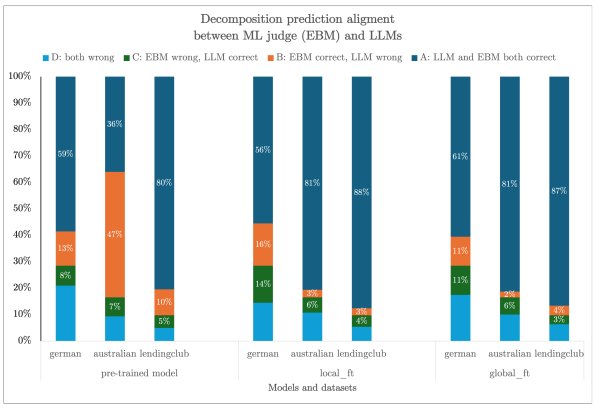
4. Preliminary Results

Prediction performance Figure 3a shows the accuracy performance for three models: a pre-trained model, a fine-tuned model on global explanations, and a fine-tuned model on local explanations. Accuracy is evaluated at the dataset level. Regarding overall LLM prediction accuracy, fine-tuned models outperform the baseline across all datasets. The Australian dataset shows a substantial increase in prediction accuracy, from below 50% in the baseline to above 80% in the fine-tuned models, and both fine-tuned models outperform the ML judge. However, the prediction performance is not substantially different between the two fine-tuned models. This can be explained by the same instructions, i.e., predicted labels, during fine-tuning.

The alignment between LLM's predictions and ML judge predictions, measured by prediction alignment, is inconsistent across the three datasets. Fine-tuning tends to improve alignment when the base model is poorly aligned with the teacher (Australian dataset), but the "mimic" behavior is reduced when



(a) Prediction performance comparison among three models

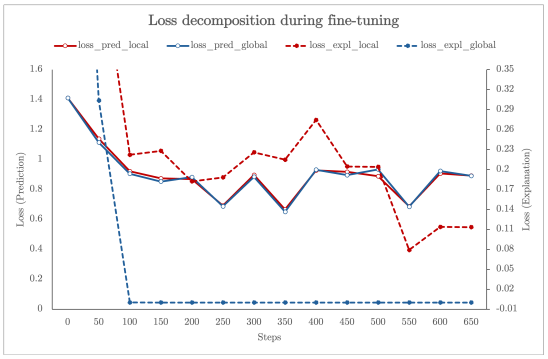


(b) Decomposition prediction alignment

Figure 3: Prediction performance among three LLMs: pretrained model, fine-tuned model on local explanations (local_ft), fine-tuned model on global explanations (global_ft). High-resolution figures are available in this link: *Figure (a)* compares prediction accuracy among three models on three datasets. LLM prediction accuracy (blue columns) compares LLMs' predictions with true labels. Prediction alignment (orange columns) compares the LLM's prediction with the ML judge's prediction. ML judge accuracy (green lines) is the EBM model's prediction accuracy. *Figure (b)* decomposes prediction by LLMs into 4 groups: A (both LLM and EBM correct), B (EBM correct, LLM wrong), C (LLM correct, EBM wrong), and D (both wrong).

Metrics \ Models	local_ft & german	local_ft & australian	local_ft & lendingclub	global_ft & german	global_ft & australian	global_ft & lendingclub
precision@5	97.4%	100.0%	99.4%	97.7%	100.0%	98.7%
precision@10	97.2%	100.0%	98.5%	94.0%	100.0%	96.3%
precision@15	89.9%	93.0%	93.3%	94.1%	93.3%	89.6%
pos_matching_5	30.5%	39.0%	62.2%	20.1%	38.0%	48.6%
pos_matching_10	20.8%	29.1%	43.2%	15.8%	28.4%	30.1%
pos_matching_15	17.3%	29.3%	35.2%	14.4%	25.4%	23.7%
spearman_5	51.3%	56.4%	80.6%	57.3%	40.7%	75.1%
spearman_10	55.7%	66.8%	87.2%	53.7%	67.8%	76.0%
spearman_15	59.5%	76.4%	86.2%	67.1%	79.7%	74.1%
ndcg_5	78.9%	91.0%	96.2%	84.4%	92.5%	94.7%
ndcg_10	84.0%	94.6%	96.1%	88.2%	95.1%	94.4%
ndcg_15	85.8%	95.7%	96.2%	90.2%	96.0%	94.8%

(a) Performance of fine-tune models on the explanation task on three datasets



(b) Loss decomposition during fine-tuning: local model (red) vs. global model (blue)

Figure 4: Figure (a) compares explanation performance between 2 fine-tuned models. Figure (b) decomposes the loss during fine-tuning. High-resolution figures are available in this link.

the baseline model already performs similarly to the teacher (LendingClub and German Credit).

The alignment between LLMs and the ML judge is good only when it improves overall prediction performance. To examine this behavior, we decompose the LLMs' predicted answers into four groups: A (both LLM and EBM correct), B (EBM correct, LLM wrong), C (LLM correct, EBM wrong), and D (both wrong), as shown in Figure 3b. Accordingly, group A indicates that LLMs adopt the teacher's inductive bias, whereas groups B and C indicate derivation from teacher instruction. Across the dataset, fine-tuning increases the number of shared correct predictions (group A, accounting for 60-90% results) and reduces the harmful deviation from the teacher (group B is less representative than group C).

Explanation performance Even with the strict schema, the baseline pre-trained model performs poorly on the explanation task. Despite the JSON instruction template containing only feature names, the pre-trained models generate long explanations and incorrect or incomplete feature names, making them unsuitable for parsing and evaluation. This hallucination is expected, as the explanation task is more difficult than the prediction task. Thus, for explanation performance, we report metrics only for

fine-tuned models, as shown in Figure 4a.

Across all datasets, both fine-tuned LLMs achieve high precision@k and NDCG@k, indicating that fine-tuned models can reproduce a complete set of explanatory features as instructed, with the most important features placed higher rank. Differences arise primarily from ranking fidelity: positional matching (pos_matching@k) is substantially lower than previous measures, suggesting that exact rank alignment is difficult. Spearman rank correlations are better (above 50%), showing moderate alignment in monotonicity of LLMs' explanations with respect to the ML teacher. In other words, given the overlapping features, high-ranked features in benchmarks are also highly ranked in explanations generated by LLMs. Model-wise, local fine-tuning consistently improves ordering fidelity on LendingClub and increases positional matches on German, whereas global fine-tuning yields better NDCG and deeper-rank ordering on German; on Australian, the two models are largely comparable.

Loss decomposition during fine-tuning Loss decomposition during fine-tuning (as shown in Figure 4b) provides additional evidence that LLMs understand the two tasks differently, even though they appear in the same prompts. The loss curves indicate how quickly the model learns each task. Specifically, the prediction losses for the local and global models, as shown by the solid curves, are nearly overlapping, indicating that the explanation complexity in the instruction does not materially affect the model's ability to learn the prediction task. In contrast, explanation learning is highly sensitive to rule complexity, as shown by the dashed curves diverging sharply. Specifically, the global model rapidly drives explanation loss toward zero, consistent with learning a near-deterministic global explanation template, whereas the local model retains a higher residual explanation loss due to the instance-specific variability of local explanations. The loss decomposition shows that explanation complexity governs rule adoption during fine-tuning, whereas predictive performance is insensitive to it under the dual-task fine-tuning setup.

5. Conclusion and Future Work

This study examines LLMs' inductive bias by employing an interpretable machine learning judge to provide fine-tuning instructions at different granularities to probe LLMs' inductive bias. By formulating a dual-task objective and decomposing the loss for each task, the study provides empirical evidence on how LLMs learn decision rationales during fine-tuning and how they perform predictions after fine-tuning.

The preliminary results from a case study on creditworthiness assessment support the LLM's preference for simplicity when learning decision rationales from the ML judge. Global explanations, which are stable across instances, are adopted rapidly during fine-tuning. By contrast, local explanations, which vary across instances, remain substantially harder to learn. During inference, fine-tuned models retrieve many relevant features, but reproducing the exact ranking of local explanations is more difficult. Furthermore, the decomposition of the training loss suggests that LLMs learn the prediction task differently from the explanation task. Two fine-tune models perform similarly on the prediction task and align well with the ML judge. However, they differ in explanation performance.

For future work, we aim to refine the methodology and extend the experiment to different domains to enhance generalization and external validity. First, we will expand the experiment to include different LLMs and evaluate inference on unseen datasets after the pre-trained model's knowledge cut-off date to mitigate data contamination risk. In addition, we are aware of the risk of hallucinations in the structured explanation task, as illustrated in the findings. Thus, for this paper, we simplify the decision rationale of the ML judge by considering only individual feature names. In future research, we will examine feature interaction and the direction of contribution combined with different fine-tuning schemes, such as multi-stage fine-tuning. This extension helps examine LLM learning behavior at higher levels of complexity and how the model propagates from rationale to prediction. Equally important, study results provide behavioral explanations of LLMs rather than their internal mechanisms. It requires mechanistic interpretability to verify the student LLM's internal representations and the causal logic that truly

governs it. Finally, the explanation evaluation in this paper is against ML judge benchmarks, as we want to validate how well LLMs learn the decision rationale. For external validity, it would require both domain instant-specific expert annotations for fine-tuning and their evaluations of the explanations' faithfulness and usefulness.

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] P. Reizinger, S. Ujváry, A. Mészáros, A. Kerekes, W. Brendel, F. Huszár, Position: understanding llms requires more than statistical generalization, in: Proceedings of the 41st International Conference on Machine Learning, ICML'24, JMLR.org, 2024.

[2] M. Budnikov, A. Bykova, I. P. Yamshchikov, Generalization potential of large language models, Neural Computing and Applications 37 (2025) 1973–1997. doi:10.1007/s00521-024-10827-6.

[3] T. Dinh, Y. Zeng, R. Zhang, Z. Lin, M. Gira, S. Rajput, J.-y. Sohn, D. Papailiopoulos, K. Lee, Lift: language-interfaced fine-tuning for non-language machine learning tasks, in: Proceedings of the 36th International Conference on Neural Information Processing Systems, NIPS '22, Curran Associates Inc., Red Hook, NY, USA, 2022.

[4] D. Feng, Y. Dai, J. Huang, Y. Zhang, Q. Xie, W. Han, Z. Chen, A. Lopez-Lira, H. Wang, Empowering many, biasing a few: Generalist credit scoring through large language models, 2024. URL: https://arxiv.org/abs/2310.00566. arXiv:2310.00566.

[5] T. M. Mitchell, Machine learning, volume 1, McGraw-hill New York, 1997.

[6] R. J. Solomonoff, Algorithmic Probability: Theory and Applications, Springer US, Boston, MA, 2009, pp. 1–23. doi:10.1007/978-0-387-84816-7_1.

[7] K. Dingle, C. Q. Camargo, A. A. Louis, Input–output maps are strongly biased towards simple outputs, Nature Communications 9 (2018) 761. doi:10.1038/s41467-018-03101-6.

[8] K. Lu, A. Grover, P. Abbeel, I. Mordatch, Pretrained transformers as universal computation engines, CoRR abs/2103.05247 (2021). URL: https://arxiv.org/abs/2103.05247. arXiv:2103.05247.

[9] M. Golec, M. AlabdulJalil, Interpretable llms for credit risk: A systematic review and taxonomy, 2025. URL: https://arxiv.org/abs/2506.04290. arXiv:2506.04290.

[10] M. T. Ribeiro, S. Singh, C. Guestrin, "why should i trust you?": Explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '16, Association for Computing Machinery, New York, NY, USA, 2016, p. 1135–1144. doi:10.1145/2939672.2939778.

[11] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (Eds.), Advances in Neural Information Processing Systems, volume 30, Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf.

[12] Y. Lou, R. Caruana, J. Gehrke, G. Hooker, Accurate intelligible models with pairwise interactions, in: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '13, Association for Computing Machinery, New York, NY, USA, 2013, p. 623–631. doi:10.1145/2487575.2487579.

[13] H. Nori, S. Jenkins, P. Koch, R. Caruana, Interpretml: A unified framework for machine learning interpretability, arXiv preprint arXiv:1909.09223 (2019).

[14] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen, Lora: Low-rank adaptation of large language models, in: ICLR 2022, 2022. URL: https://www.microsoft.com/en-us/research/publication/lora-low-rank-adaptation-of-large-language-models/.