Faithful Latent Feature Mining with Large Language Models

Bingxuan Li[1], **Pengyi Shi**[2]
[1]UCLA
[2]Purdue University
bingxuan@ucla.edu, shi178@purdue.edu

Abstract

Predictive modeling often encounters significant challenges in domains with limited data availability and quality. This is particularly true in areas like healthcare, where collected features may be weakly correlated with outcomes, and gathering additional features is constrained by ethical considerations or practical limitations. Traditional machine learning (ML) models struggle to incorporate unobserved yet critical factors. In this work, we introduce an effective approach to formulate latent feature mining as text-to-text propositional logical reasoning. We propose *FLAME* (**F**aithful **L**atent Fe**A**ture Mining for Predictive **M**odel **E**nhancement), a framework that leverages large language models (LLMs) to augment observed features with latent features and enhance the predictive power of ML models in downstream tasks. Our framework is generalizable across various domains with necessary domain-specific adaptation, as it is designed to incorporate contextual information unique to each area, ensuring effective transfer to different areas facing similar data availability challenges. We validate our framework with a case study using the MIMIC data. Our results show that inferred latent features significantly enhance the downstream classifier over 10%.

Introduction

Prediction plays a crucial role in decision-making across many domains. While traditional machine learning (ML) models are powerful, they are often constrained by the availability of observed data features. Contrary to the common belief that we are in a "big data era," this is not always the case, especially in areas where decisions have profound impacts on human lives. In areas like healthcare, data availability is often constrained, with ethical limitations further restricting the features that can be collected and used (Lu, Dou, and Nguyen 2021; Yuan et al. 2023). As a result, many critical decisions must rely on a limited set of features, some of which may have weak correlations with the prediction target. This presents significant challenges for achieving accurate predictions.

To overcome the challenges posed by limited feature availability and quality, latent feature mining is a common approach. However, traditional techniques face two key limitations in domain-specific applications. First, inferring domain-specific latent features often requires contextual information beyond the available data, such as expert input, public information, or crowd-sourcing. This information is typically in natural language, which ML models like neural networks struggle to process and encode into proper embeddings. Second, many latent feature mining techniques, such as deep-learning based auto-encoders and the Expectation-Maximization (EM) algorithm, lack interpretability. They extract features in abstract mathematical formats that are difficult to explain in human terms. This is especially problematic in high-stakes domains like healthcare or criminal justice, where explaining and justifying a model's predictions is crucial for building trust and ensuring ethical decision-making. The black-box nature of these methods makes it harder to gain confidence in the model's outputs in these domains.

Human experts can infer additional latent features that go beyond the explicit data provided by drawing on their experience. For example, predicting a patient's discharge destination is critical for optimizing hospital resource allocation and planning appropriate post-acute care. Available data for this prediction task typically includes basic demographics (e.g., age and gender) and clinical information such as diagnosis codes and length of stay. However, factors like family support systems, home/community environment, and financial resources also play a significant role in discharge locations. These social-economic factors, combined with standardized clinical data, can help to predict whether a patient is ready to be discharged home with sufficient support, or needs to be transferred to a skilled nursing facility. That said, collecting such social-economic data raises privacy concerns and is often not systematically captured in electronic health records. Instead, experienced healthcare managers could rely on their understanding of these social determinants of health to make informed discharge recommendations. While effective, this human-based approach is difficult to scale, as it relies on tacit human knowledge that is hard to formalize into standardized processes. Additionally, the human reasoning process is both time- and labor-intensive, limiting its application to large populations.

Recent advancements in large language models (LLMs) present a promising new avenue with their advanced reasoning capability (Brown et al. 2020; Ouyang et al. 2022; Achiam et al. 2023). LLMs have potential to process and generate information in ways that mimic human thought processes (Ji et al. 2024). Building on this insight, we propose *FLAME*, a framework that leverages LLMs to augment observed features with latent features and enhance the predictive power of ML models in downstream tasks like classification. *FLAME* offers two key advantages over traditional latent feature mining methods: (1) it seamlessly integrates contextual information provided in natural language, and (2) by emulating human reasoning, it produces more interpretable outputs, making it particularly valuable in high-stakes domains requiring explainability. We summarize our main contributions as follows.

1. We introduce a new approach that LLMs to formulate latent feature mining as a reasoning task using text-to-text propositional logic. This method effectively infers latent features from observed data and provides significant improvements in downstream prediction accuracy and interpretability over traditional techniques.
2. We develop a four-step versatile framework that integrates domain-specific contextual information with minimal customization efforts. This framework is highly adaptable across various domains, particularly those with limited observed features and ethical constraints on data collection.
3. We empirically validate our framework through a case study. The results demonstrate that the extracted latent features enhance the performance of predictive models by over 10%.

Related Works

Data Augmentation versus Latent Feature Mining. Data augmentation is a technique widely employed to provide more data samples to improve the predictive power of ML models (Van Dyk and Meng 2001). Generative models such as Generative Adversarial Networks (GANs) learn data patterns and generate synthetic data to augment training sample sizes (Goodfellow et al. 2014; Kingma and Welling 2013). In contrast, latent features are hidden characteristics in a dataset that are not directly observed but can be inferred from available data. Incorporating meaningful latent features can enhance the performance of downstream applications (Zhai and Peng 2016; Jiang et al. 2023). Methods such EM and Variational Autoencoders (VAEs) offer alternative techniques to infer latent features from observed data. However, EM algorithms, while estimating latent variable assignments and updating model parameters to maximize data likelihood, often produce results that are difficult to interpret and require strong parametric assumptions. Similarly, VAEs use probabilistic approaches to describe data distribution with latent variables, but the learned mappings can also be hard to interpret. Another related approach is dimension reduction such as Principal Component Analysis, which reduces the size of the feature space while preserving the most important information. However, dimension reduction is less effective when the input feature set is already limited.

We summarize a comparison in Table 1 to further distinguish the difference between *FLAME* and existing approaches for enhancing predictive model from data/features perspective.

Supervised Fine-tuning with Synthetic Training Data Supervised Fine-tuning (SFT) is an effective method for LLMs to reduce hallucinations and better align outputs with real-world data and human preferences (Tonmoy et al. 2024; Qiao et al. 2022; Hu et al. 2021). Synthetic data offers a low-cost way to enhance LLM reasoning across domains (Liu et al. 2024; Zelikman et al. 2022; Wang et al. 2022). In this work, *FLAME* generates synthetic "rationales" in a self-instruct fashion for the reasoning process to infer latent features, followed by SFT to enhance alignment and reduce hallucinations.

Note that we distinguish between augmenting the feature space and augmenting training data. Our primary goal is to enrich the feature space by inferring and adding latent features to improve downstream predictions. As part of the steps in *FLAME* to achieve this goal, we also augment training data with synthetic samples during the fine-tuning process for LLMs.

The Problem Setting

In this section we formally describe our problem setting that leverages latent features to enhance downstream tasks. The downstream task we focus on is a multi-class classification problem, but the framework can easily extend to other downstream prediction tasks such as regression problems.

In a standard multi-class classification problem setting, suppose we have a dataset $D = (x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, where x_i is a d-dimensional vector representing the input features $X \in \mathcal{X}$ and $y_i \in \mathcal{Y} = \{1, 2, \ldots, C\}$ denotes the corresponding class label Y for individual $i = 1, \ldots, n$. The goal is to learn a classifier $f : \mathcal{X} \rightarrow \mathcal{Y}$ that accurately predicts the class labels. Consider the following scenarios in which f struggles to capture the relationship between X and Y: (1) The number of input features X is small relative to the complexity of the classification task. (2) When X are weakly correlated with class labels Y, they may not provide discriminating information to accurately predict the corresponding class labels.

To address these challenges, we can use additional informative features to enhance the classifier's ability to capture the relationship between X and Y. Latent features can serve such a purpose: Latent features, denoted as Z, represent underlying attributes that are not directly observed within the dataset but are correlated with both the observed features X and the class labels Y. We use a function g with $Z = g(X)$ to denote the correlations between the latent features and the observed features X. Latent features Z are correlated with X and Y. One can learn the latent features from the original features X and augment the features $f(\mathbf{X}, \mathbf{Z})$ to learn the classifier Y.

While this approach seems beneficial intuitively, it is important to note that adding more features is not always

Approach	Additional Features Creation Capability	Interpretability	Contextual Information Integration Capability
Data Augmentation (GANs, VAEs)	×	×	×
Latent Feature Mining (EM Algorithm)	✓	×	×
Dimension Reduction	✓	×	×
FLAME	✓	✓	✓

Table 1: Comparison of *FLAME* and related methods: Unlike data augmentation, which increases sample size, *FLAME* expands the feature space by training LLMs to infer latent variables from existing features. Compared to traditional latent feature mining methods, *FLAME* mimics human expert reasoning and incorporates domain-specific context, offering improved interpretability. Unlike dimension reduction methods, *FLAME* enriches the dataset by adding latent features that capture key aspects of the underlying phenomena.

helpful if the extracted features are not meaningful and introduce noise. In the following lemma, we show in a simple logistic regression setting that while adding features can reduce in-sample loss, it does not always reduce out-of-sample loss if the added features are not informative. We use the log-loss (the cross-entropy loss) of the logistics regression for binary outcome $Y \in \{0, 1\}$. We denote the optimal coefficients that minimize the in-sample log-loss function as β^* for the original features and $\tilde{\beta}^*$ for the augmented features.

Lemma 1. *The in-sample log-loss always follows* $\mathcal{L}^{in}(\tilde{D}, \tilde{\beta}^*) \leq \mathcal{L}^{in}(D, \beta^*)$. *When the added features are non-informative, there exist instances such that the out-of-sample log-loss* $\mathcal{L}^{out}(\tilde{D}, \tilde{\beta}^*) > \mathcal{L}^{out}(D, \beta^*)$.

The results in the lemma can be generalized to multi-class labels. Since augmenting the feature space is not necessarily beneficial unless the added features are meaningful, a major part of our case study is to empirically test whether the extracted features from our framework indeed improve downstream prediction. If the added features significantly enhance downstream prediction accuracy, this provides strong evidence that the inferred latent features are meaningful.

Latent Feature Mining with LLMs

We propose a new approach, *FLAME*, to efficiently and accurately extract latent features and augment observed features to enhance the downstream prediction accuracy. It extracts the latent features Z from the original features X to capture complex patterns and relationships that individual features may overlook, especially when some of the X's are weakly correlated with the outcome Y. At a high level, our approach transform this latent feature extraction process as a text-to-text propositional reasoning task, i.e., infer the relationship $Z = g(X)$ through logical reasoning with natural language. Figure 1 provides an example of the extract process with the steps elaborated on below.

Following the framework established in previous work (Zhang et al. 2022), we denote the predicates related to the observed features as $P_1, P_2, \ldots, P_m$. Consider a propositional theory S that contains rules that connect P's to the latent feature Z. We say Z can be deduced from S if the logic implication $(P_1 \wedge P_2 \wedge \ldots \wedge P_m) \to Z$ is covered in S. For potentially complicated logical connections between P's and Z, we also introduce intermediate predicates O's and formulate a logical chain (a sequence of logical implications) that connects X to the latent features Z as follows:

$$X \to (P_1 \wedge P_2 \wedge \ldots \wedge P_m) \to (O_1 \wedge O_2 \wedge \ldots \wedge O_\ell) \to Z. \quad (1)$$

Our approach formulates this logical chain as a multi-stage Chain of Thoughts (CoT) prompt template, and then guide LLMs to infer Z from X using the prompt template. Specifically, we first extract predicates P's from X. Then we infer intermediate predicates with a rule $(P_1 \wedge P_2 \wedge \ldots \wedge P_m) \to O_l$ for $l = 1, \ldots, \ell - 1$, and forward the intermediate predicates into the next stage to infer O_{l+1}. Finally, we infer latent features with $(O_1 \wedge O_2 \wedge \ldots \wedge O_\ell) \to Z$. With the formulated multi-stage CoT prompt template, we then generate synthetic training data to fine-tune LLMs to enhance the logical reasoning ability of LLMs in the self-instruct manner (Wang et al. 2022).

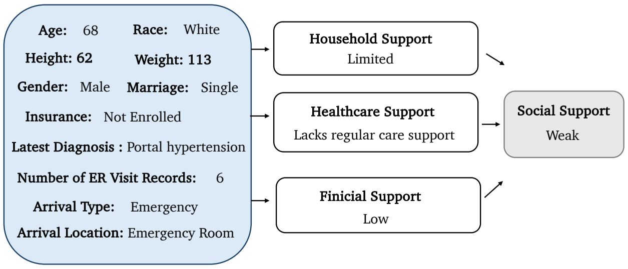


Figure 1: Example of latent feature mining through chain of reasoning. The latent feature "Social Supports" (Z) is inferred from the observed input features (X) via intermediate predicates (O), and is then used alongside X to improve the prediction for outcome (Y).

We use a hypothetical example from our healthcare case study setting to illustrate the formulation of the logic chain. The blue (leftmost) box in Figure 1 shows the observed feature X for one individual. Examples for the predicates P's formulated from X could be:

> P_1 : *"the patient is uninsured"*, P_2 : *"the patient arrived through emergency room"*, P_3 : *"the patient is single"*, P_4 : *"the patient has portal hypertension"*, P_5 : *"the patient has multiple ER visits (10 records)"*, P_6 : *"the patient is 68 years old"*, P_7 : *"the patient has no listed emergency contacts"*, ...

To infer the latent feature Z – in this example, the level of support available at home/community after being discharged – we go through a multi-stage reasoning to infer the intermediate predicates O's; see the white (middle) boxes in Figure 1. One example logic that connects P's to O's could be:

> P_6 = *”The patient is 68 years old”*
> P_3 = *”The patient is single”*
> P_2 = *”The patient arrived through emergency room”*
> O_1 = *”The patient has limited home support”*
> If $(P_6 \wedge P_3 \wedge P_2 \rightarrow O_1) \in S$, then O_1 is True.

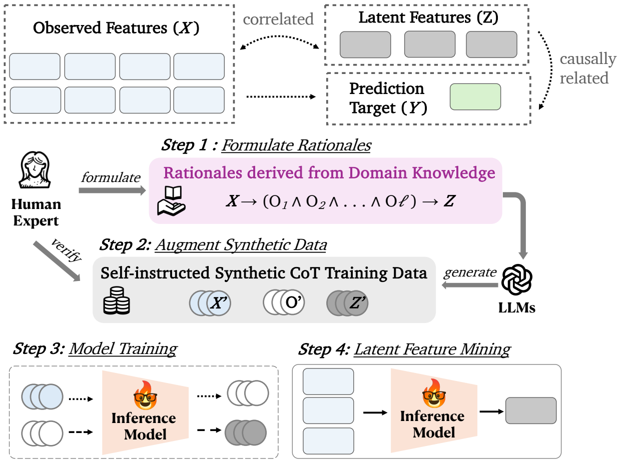
Finally, with P's and O's, we can connect X with Z through the logic chains:

> *"The patient's age of 68 and single status, combined with ER presentation, indicates limited support at home. Additionally, being uninsured and having multiple ER visits (6 records) for portal hypertension demonstrates lacks of regular care support. The combination of no insurance and repeated emergency visits suggests low financial support. Given these circumstances, the patient shows significant gaps across all support dimensions - household, healthcare, and financial resources - indicating weak overall social support that will likely impact post-discharge outcomes."*

Here, *being 68 years old and single status"* and *being uninsured with 6 ER visits and portal hypertension"* are P's extracted from the features X, while *limited household support"* and *lacks regular care support"* and *low financial support"* are O's. Finally, the rationales *the patient shows significant gaps across all support dimensions - household, healthcare, and financial resources - indicating weak overall social support that will likely impact post-discharge outcomes"* connect the intermediate predicates to the latent variable Z we want to infer, i.e., Z='weak social support'.

Figure 2 illustrates the full process of of `FLAME` with four steps.

(1) Formulate baseline rationales: The first step is to formulate baseline rationales, whic serve as guidelines for LLMs to infer latent features from observed ones. This involves two sub-steps:



Observed Features (X)
correlated
Latent Features (Z)
causally related
Prediction Target (Y)
Step 1 : Formulate Rationales
Human Expert
formulate
Rationales derived from Domain Knowledge
$X \rightarrow (O_1 \wedge O_2 \wedge \ldots \wedge O\ell) \rightarrow Z$
verify
Step 2: Augment Synthetic Data
Self-instructed Synthetic CoT Training Data
X'
O'
Z'
generate
LLMs
Step 3: Model Training
Inference Model
Step 4: Latent Feature Mining
Inference Model

Figure 2: Overview of latent feature inference framework.

– The first sub-step is to develop some baseline rationales, i.e., identify observed features potentially correlated with latent features and formulate their relationships – the logic chain that connects X to Z. Sources to help formulate these baseline rationales include established correlations (e.g., readmission risk score formulas), expert input, and other contextual information like socio-economic status in the neighborhood. This is also a critical step in our framework that allows the **integration of domain-specific contextual information** in the format of natural language.

– In the second sub-step, we craft prompts with interactive alignment. This is a critical component to establish correct reasoning steps for prompts used in Step 2 to generate synthetic rationales. We involve experienced human in the domain to provide a prompt template for LLMs to generate rationales aligned with the baseline rationales, then test the prompt template on a few examples using zero-shot. If the LLM fails to certain example, we provide the ground truth back to the LLM, allowing it to revise the prompt template (Miao, Teh, and Rainforth 2023). This process iteratively refines the template until LLMs consistently generate the desired output for all selected examples.

(2) Enlarge data with synthetic rationales for fine-tuning: We generate synthetic training data in self-instruct fashion (Wang et al. 2022). With a handful of examples of the baseline rationales as a reference, we guide the LLMs via in-context learning to generate similar rationales to enlarge the training data samples. To ensure the quality and diversity of the generated dataset, we introduce human-in-the-loop interventions to filter out low-quality or invalid data based on heuristics. We also leverage automatic evaluation metrics for quality control, e.g., removing data that lack essential keywords.

(3) Fine-tuning LLMs: To enhance the reasoning capabilities of the LLMs and better align their outputs in specific domains, we leverage the fine-tuning process with processed dataset from the previous step (Qiao et al. 2022).

Fine-tuning not only boosts the accuracy and reliability of the LLMs, but also significantly improves their ability to reason with complex inputs and reduce hallucination (Tonmoy et al. 2024).

(4) Latent feature inference: The fine-tuned model mirrors the nuanced reasoning process of human experts. We use it to infer latent features, which are then fed into downstream prediction tasks to improve accuracy.

Experiments Setup

We test the efficacy of `FLAME` on the discharge location prediction task in the healthcare domain.

Task Description. The discharge location prediction task involves using individual patient-level data to predict the most likely discharge destination for patients upon their discharge from the hospital inpatient units. We apply `FLAME` to extract (new) latent features to enhance the prediction accuracy for this discharge location task. Specifically, we create a new feature, "social support," which captures the extent of healthcare, familial, and community support available to the patient after being discharged.

Dataset. MIMIC (Medical Information Mart for Intensive Care) dataset (Johnson et al. 2016) is a comprehensive dataset containing detailed de-identified patient clinical data and is widely used for various prediction tasks in the machine learning literature.

Implementation Details. We implement our proposed framework as follows [1]. All prompt templates are available in Appendix C.

- Step 0. Profile writing: In this pre-processing step, we translate structured data X into text that can be better handled by LLMs, i.e., formulating predicates P's from the features X. Then we formulate the intermediate predicates O's, where we prompt LLMs to extract and summarize underlying information such as background and socio-economic status in two or three sentences. We then merge these sentences into the patient's profile. We use zero-shot prompting with GPT-4.

- Step 1. Formulating rationales: Using human input, established risk score calculations, and auxiliary information available publicly, we establish the logic chains from P's and O's to Z.

- Step 2. Enlarge fine-tuning data: With the 40 baseline rationales, we generate additional synthetic rationales. We sample patient features and corresponding ground truth risk scores from the dataset, using one of the 40 rationales as an example, to prompt LLMs to produce similar narratives with CoT prompts. In total, we got 3000 rationales for the training data.

- Step 3. Fine-tune LLMs: Our framework is designed to be plug-and-play, allowing the synthetic data generated in the previous step to be used across different language models. We fine-tune two pre-trained language models for cross-validation purposes: GPT-3.5 and Llama2-13b (OpenAI 2021). We use OpenAI API to fine-tune GPT-3.5-turbo-0125 (Touvron et al. 2023; OpenAI). We fine-tune Llama2-13b-chat using LoRA (Hu et al. 2021).

- Step 4. Inference with LLMs: We prompt fine-tuned LLMs to infer $\hat{Z}_i$ from features X_i for each patient i in the test data.

Variable	Categories	Percentage
Discharge Location	Home	40.19
	Other	40.19
	Died	19.62
Gender	Female	51.53
	Male	48.47
Race	White	61.09
	Black/African American	11.70
	Other	11.45
	Asian	2.49
	Hispanic or Latino	1.89
	White - Other European	1.69
Marital Status	Married	43.05
	Single	35.29
	Widowed	11.01
	Other	10.65
Insurance	Other	58.24
	Medicare	34.53
	Medicaid	7.23
Language	English	90.84
	Other	9.16
Admit Type	Emergency	56.95
	Other	41.60
	Elective	1.45

Table 2: Categorical Variables Summary Statistics of MIMIC dataset

Evaluation. We train an ML classifier to predict outcomes with and without the inferred latent features, i.e., $\hat{Y}_i \sim f(X_i, \hat{Z}_i)$ versus $\hat{Y}_i \sim f(X_i)$ and then evaluate their out-of-sample accuracy. The dataset is split into a 7:3 ratio for training and testing, respectively. We use five different random seeds to run each experiment five times and then average the results to ensure the reliability of our findings.

Experiments Result

In this section, we demonstrates the experiment result of discharge location prediction task. We also conduct ablation experiments to further investigate our advantage and limitations (Please see Appendix B).

Table 3 demonstrates the result of discharge location prediction task. The results show an average improvement of approximately 8.64% in accuracy and 8.64% in F1 score when latent features are added to the models. This is similar to the percentage increase reported in Table 5(a). Specifically, the GBT model achieves the highest accuracy after in-

[1] Code and Implementations are available here: https://bit.ly/3XMi8QN

Model	Accuracy (std.)	F1 score (std.)
LR	65.22% (0.01)	65.46% (0.01)
MLP	63.19% (0.02)	63.19% (0.02)
GBT	64.84% (0.01)	65.09% (0.01)
RF	65.11% (0.01)	65.44% (0.01)
LR w/ Latent Feature	**71.22% (0.01)**	**71.26% (0.01)**
MLP w/ Latent Feature	**74.40% (0.01)**	**74.50% (0.01)**
GBT w/ Latent Feature	**75.56% (0.02)**	**75.38% (0.02)**
RF w/ Latent Feature	**75.31% (0.01)**	**75.22% (0.01)**

Table 3: The experiment result. We use five different random seeds to run experiment five times and report the average.

corporating the latent features. The results demonstrate another strong evidence of using our framework to improve downstream prediction power with the addition of latent features.

Furthermore, the inferred variable “Social Support” shows strong correlation with the discharge location. This finding suggests that *FLAME* can uncover meaningful latent variables that might otherwise be overlooked in traditional data collection methods in the healthcare settings. More importantly, this experiment on a different dataset from a different domain demonstrates the effectiveness and generalizability of *FLAME*.

Discussion

What is required to generalize *FLAME* for each new application? *FLAME* has broad potential across various domains, particularly those with limited observed features and ethical constraints. Steps 2-4 primarily rely on the adaptability of LLMs and allow flexible application across different domains. However, Step 1 – identifying and formulating baseline domain-specific rationales – requires domain expertise and involves additional manual effort. This effort is worthwhile because our framework is intentionally designed to be domain-specific. We believe this is actually the critical step that drives the improved downstream prediction accuracy demonstrated in Section . By leveraging contextual information that traditional methods cannot, *FLAME* significantly enhances model performance.

Future work. As we continue to refine our *FLAME* framework, we are actively pursuing avenues to enhance its fidelity and reliability. First, we are streamlining the process to reduce the need for human intervention and increase the scalability of our approach, thus minimizing the potential for subjective influences and increasing the scalability of our approach. This involves automating feature selection and validation processes, leveraging the LLM’s capabilities to self-verify and iterate on its outputs. Second, we acknowledge that LLMs can inadvertently perpetuate existing biases present in their training data, and how to mitigate such bias remains an open question in the field (Wan et al. 2023; Gallegos et al. 2024). *FLAME* attempts to minimize such biases by leveraging domain-specific data and expert input during the fine-tuning process. Furthermore, the training dataset is curated to include a diverse range of scenarios, and the model’s inferences are continually tested against ground truth data where available. Nevertheless, we are implementing more sophisticated error control mechanisms to diminish the impact of potential inaccuracies in the generated features. For example, we are in the process of hiring human annotators to verify the output from the LLMs reasoning. Other possible options include developing confidence scoring systems for generated features (Detommaso et al. 2024).

Conclusion

In conclusion, *FLAME* provides a novel solution to the challenges of limited feature availability in high-stakes domains by using LLMs to augment observed data with interpretable latent features. This framework improves downstream prediction accuracy while enhancing explainability, which makes it valuable for sensitive decision-making in areas like healthcare.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

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

Campos, R.; Mangaravite, V.; Pasquali, A.; Jorge, A.; Nunes, C.; and Jatowt, A. 2020. YAKE! Keyword extraction from single documents using multiple local features. *Information Sciences*, 509: 257–289.

Detommaso, G.; Lopez, M. B.; Fogliato, R.; and Roth, A. 2024. Multicalibration for confidence scoring in LLMs. In *ICML 2024*.

Gallegos, I. O.; Rossi, R. A.; Barrow, J.; Tanjim, M. M.; Kim, S.; Dernoncourt, F.; Yu, T.; Zhang, R.; and Ahmed, N. K. 2024. Bias and fairness in large language models: A survey. *Computational Linguistics*, 1–79.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. *Advances in neural information processing systems*, 27.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; and Chen, W. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Ji, B.; Liu, H.; Du, M.; and Ng, S.-K. 2024. Chain-of-Thought Improves Text Generation with Citations in Large Language Models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 18345–18353.

Jiang, Q.; Chen, C.; Zhao, H.; Chen, L.; Ping, Q.; Tran, S. D.; Xu, Y.; Zeng, B.; and Chilimbi, T. 2023. Understanding and constructing latent modality structures in multi-modal representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7661–7671.

Johnson, A. E.; Pollard, T. J.; Shen, L.; Lehman, L.-w. H.; Feng, M.; Ghassemi, M.; Moody, B.; Szolovits, P.; Anthony Celi, L.; and Mark, R. G. 2016. MIMIC-III, a freely accessible critical care database. *Scientific data*, 3(1): 1–9.

Kingma, D. P.; and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Liu, R.; Wei, J.; Liu, F.; Si, C.; Zhang, Y.; Rao, J.; Zheng, S.; Peng, D.; Yang, D.; Zhou, D.; et al. 2024. Best Practices and Lessons Learned on Synthetic Data for Language Models. *arXiv preprint arXiv:2404.07503*.

Lu, Q.; Dou, D.; and Nguyen, T. H. 2021. Textual Data Augmentation for Patient Outcomes Prediction. In *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2817–2821.

Miao, N.; Teh, Y. W.; and Rainforth, T. 2023. Selfcheck: Using llms to zero-shot check their own step-by-step reasoning. *arXiv preprint arXiv:2308.00436*.

OpenAI. ???? Fine-tuning. https://platform.openai.com/docs/guides/fine-tuning. Accessed: 2024-05-22.

OpenAI. 2021. GPT-3.5. https://platform.openai.com/docs/models/gpt-3.5. Accessed: 2024-05-22.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744.

Qiao, S.; Ou, Y.; Zhang, N.; Chen, X.; Yao, Y.; Deng, S.; Tan, C.; Huang, F.; and Chen, H. 2022. Reasoning with language model prompting: A survey. *arXiv preprint arXiv:2212.09597*.

Tonmoy, S.; Zaman, S.; Jain, V.; Rani, A.; Rawte, V.; Chadha, A.; and Das, A. 2024. A comprehensive survey of hallucination mitigation techniques in large language models. *arXiv preprint arXiv:2401.01313*.

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

Van Dyk, D. A.; and Meng, X.-L. 2001. The art of data augmentation. *Journal of Computational and Graphical Statistics*, 10(1): 1–50.

Wan, Y.; Pu, G.; Sun, J.; Garimella, A.; Chang, K.-W.; and Peng, N. 2023. "Kelly is a Warm Person, Joseph is a Role Model": Gender Biases in LLM-Generated Reference Letters. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Findings of the Association for Computational Linguistics: EMNLP 2023*, 3730–3748. Singapore: Association for Computational Linguistics.

Wang, Y.; Kordi, Y.; Mishra, S.; Liu, A.; Smith, N. A.; Khashabi, D.; and Hajishirzi, H. 2022. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*.

Yuan, J.; Tang, R.; Jiang, X.; and Hu, X. 2023. Llm for patient-trial matching: Privacy-aware data augmentation towards better performance and generalizability. In *American Medical Informatics Association (AMIA) Annual Symposium*.

Zelikman, E.; Wu, Y.; Mu, J.; and Goodman, N. 2022. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35: 15476–15488.

Zhai, C.; and Peng, J. 2016. Mining Latent Features from Reviews and Ratings for Item Recommendation. In *2016 International Conference on Computational Science and Computational Intelligence (CSCI)*, 1119–1125.

Zhang, H.; Li, L. H.; Meng, T.; Chang, K.-W.; and Broeck, G. V. d. 2022. On the paradox of learning to reason from data. *arXiv preprint arXiv:2205.11502*.

Zhang, J.; Zhao, Y.; Saleh, M.; and Liu, P. J. 2019. PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. arXiv:1912.08777.

Appendix

A. Proof of Lemma 1

We use the log-loss, defined as

$$\mathcal{L}(D, \beta) = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log(p_i) + (1-y_i)\log(1-p_i)\right] \tag{2}$$

for given data $D = \{(x_i, y_i)\}_{i=1}^{n}$ and $p_i = 1/\bigl(1 + e^{-(\beta_0+\beta_1 x_i)}\bigr)$. When using the augmented feature $\tilde{x}_i = (x_i, z_i)$, we denote the data as $\tilde{D} = \{\bigl((x_i, z_i), y_i\bigr)\}_{i=1}^{n}$.

For the first part of the lemma, we note that the in-sample log-loss for the original features follows

$$\mathcal{L}^{\text{in}}(D, \beta) = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log(p_i) + (1-y_i)\log(1-p_i)\right], \tag{3}$$

and the in-sample log-loss for the augmented features follows

$$\mathcal{L}^{\text{in}}(\tilde{D}, \beta) = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log(\tilde{p}_i) + (1-y_i)\log(1-\tilde{p}_i)\right], \tag{4}$$

where $p_i = 1/\bigl(1 + e^{-(\beta_0+\beta_1 x_i)}\bigr)$ and $\tilde{p}_i = 1/\bigl(1 + e^{-(\beta_0+\beta_1 x_i+\beta_2 z_i)}\bigr)$.

We denote the optimal coefficients that minimize the log-loss in (3) as $\beta^* = (\beta_0^*, \beta_1^*)$, and the coefficients that minimize the log-loss in (4) as $\tilde{\beta}^* = (\tilde{\beta}_0^*, \tilde{\beta}_1^*, \tilde{\beta}_2^*)$. Note that $\check{\beta} = (\beta_0^*, \beta_1^*, 0)$ is a feasible solution for the log-loss in (4). Therefore, using the optimization property, we have

$$\mathcal{L}^{\text{in}}(\tilde{D}, \tilde{\beta}^*) \leq \mathcal{L}^{\text{in}}(\tilde{D}, \check{\beta}) = \mathcal{L}^{\text{in}}(D, \beta^*),$$

which completes the first part of the lemma.

For the second part of the lemma, we first assume that for the given data D, $\mathcal{L}^{\text{in}}(\tilde{D}, \tilde{\beta}^*) = \mathcal{L}^{\text{in}}(D, \beta^*) - \epsilon/n$ where $\epsilon \geq 0$ from the first part of the lemma. We now construct an instance with an out-of-sample dataset D' that contains $n+1$ samples, where D' consists of (i) the n data points that exactly match with D (or $\tilde{D}$) for the first n samples, and (ii) one additional sample (x_{i+1}, y_{i+1}) (or $((x_{i+1}, z_{i+1}), y_{i+1})$ when using the augmented features). Without loss of generality, assume that $y_{i+1} = 1$. Then we have

$$\mathcal{L}^{\text{out}}(D', \beta^*) = \frac{1}{n+1}\left(n\mathcal{L}^{\text{in}}(D, \beta^*) - \log(p_{i+1}))\right)$$

and

$$\mathcal{L}^{\text{out}}(\tilde{D}', \tilde{\beta}^*) = \frac{1}{n+1}\left(n\mathcal{L}^{\text{in}}(\tilde{D}, \tilde{\beta}^*) - \log(\tilde{p}_{i+1}))\right).$$

When the added features Z's are non-informative, we consider the scenarios that they are noise and the additional term $\tilde{\beta}_2^* Z$ also contributes noise to the predictions. In other words, the coefficients $\tilde{\beta}^*$ do not generalize well to the test data. Therefore, there exists an instance where the realization of Z, z_{i+1} deviates from the predicted probability significantly, such that

$$\tilde{p}_{i+1} < p_{i+1}/\exp(\epsilon) \leq p_{i+1}.$$

Note that this instance exists since the noise terms do not correspond to any actual pattern in the test data, causing incorrect predictions, and in our construction, a smaller predicted probability would be less accurate as the label $y_{i+1} = 1$. Therefore,

$$-\log(\tilde{p}_{i+1}) > -\log(p_{i+1}) + \epsilon,$$

and

$$\begin{aligned}\mathcal{L}^{\text{out}}(\tilde{D}', \tilde{\beta}^*) &= \frac{1}{n+1}\left(n\mathcal{L}^{\text{in}}(D, \beta^*) - \epsilon - \log(\tilde{p}_{i+1}))\right) \\ &> \frac{1}{n+1}\left(n\mathcal{L}^{\text{in}}(D, \beta^*) - \log(p_{i+1}))\right) = \mathcal{L}^{\text{out}}(D', \beta^*).\end{aligned}$$

B. Ablation Study

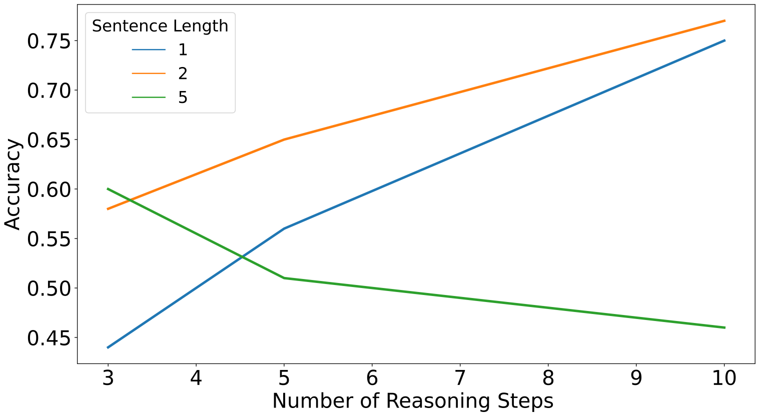
Do the inherent biases of LLMs influence the inference process of latent features? To assess whether the reasoning processes within generated texts exhibit biases, we conducted the following experiments. First, we utilized the pretrained keyword extraction model YAKE (Campos et al. 2020) to search for racial terms within the reasoning steps of the generated text. The analysis showed that such keywords were absent, indicating no explicit racial bias in this context. Second, we closely examined the race distribution in the ground-truth data versus the distribution in the predictions made by the model. The analysis revealed that the race distributions between the ground-truth and the predicted outcomes were similar. This similarity suggests that the model does not introduce additional racial biases in its predictions and accurately reflects the distributions present in the input data. Both results validate that the LLMs' inherent biases are not carried into the inference process. Other types of bias, such as bias in lexical context, are beyond the scope of this paper and are left for future research.

How sensitive is our approach to the quality of human guidelines? *FLAME* is sensitive to human guidelines, specifically the baseline rationales and prompt templates formulated in Step 1. We have conducted an ablation study to determine the optimal level of details required in the prompts. As shown in Figure 3 (b), the best performance was achieved with the most reasoning steps and a sentence length of two per step. In other words, increasing the number of reasoning steps allows us to decompose the task into simpler components and enhances the performance of LLMs. More importantly, while human guidelines are important, **the interactive self-revise alignment strategy can significantly help** during the sub-step of Step 1 (prompt crafting). By providing ground truth and encouraging self-reflection, GPT-4 can revise the prompt template to include crucial details, ensuring a more accurate evaluation.

How important is the fine-tuning step in *FLAME*? We have conducted another ablation study, where we repeated the risk-level prediction task with zero-shot, one-shot, and three-shot prompting to compare with our fine-tuned model. In zero-shot, we provided only the task description. In one-shot and three-shot, we included randomly selected human-verified examples. Accuracy rankings from lowest to highest were: three-shot (40%), zero-shot (55%), one-shot (60%), and the fine-tuned model (75%); see Figure 3 (a). The three-shot's poor performance may be due to information loss from long inputs. Zero-shot responses are highly variable and not well-suited for downstream tasks. Although one-shot showed improvement, the fine-tuned model significantly outperformed all others. Hence, the answer to the question is that **fine-tuning is necessary**. Additionally, the fine-tuning process incorporates feedback loops with domain experts to adjust and correct the model's reasoning pathways, ensuring that the latent features inferred, such as the need for substance abuse treatment, are aligned with nuanced real-world outcomes rather than broad statistical correlations.

Setting	Accuracy
Zero-shot	55%
One-shot	60%
Three-shot	40%
Fine-tune	75%

(a) Risk level prediction results across different setting



(b) Risk level prediction results across different strategy

Figure 3: Ablation study results: (a) Experiments on risk level prediction task using GPT4 with different prompting setting. (b) Experiments using GPT4 with different prompting setting different prompting strategies.

How does *FLAME* perform compared to the baseline? We implemented a baseline approach using an MLP classifier on embeddings from a text summarization model, and repeat the risk level prediction experiment: We use few-shot to generate a profile that contains all information related to the client (See Figure **??** for more detail on prompt template), then we extract the embedding from the encoder of Pegasus (Zhang et al. 2019). These embeddings serve as input to an MLP classifier for risk level prediction. We evaluated on a balanced validation set of 50 samples per class, and this baseline achieved only 52% accuracy on the three-class classification task, while our proposed approach is able to reach over 75% accuracy.

C. Compute Resources

For all experiments, we split data into training and testing dataset with ratio of 8:2.

We use OpenAI offical API to finetune GPT3.5 model, which requires no GPUs. Each finetune job takes about 2 hours. We repeat 6 times for different sub tasks. Additionally, we also run Machine Learning baseline model on CPU (Intel i7). We run grid search for each classifier.

All other experiments (e.g. sensitive experiment) are conducted on ChatGPT, which requires no GPU.

D. Prompts

Inference Prompt Template

```
[System Prompt] You are a healthcare
professional assessing a patient's
social support to understand their
overall well-being and potential
factors influencing their final
discharge plan.

[User Prompt]
Here are principles of social support
assesment: <principles>
---
Here are questions to consider when
assessing social support: <questions>
---
Given patient profile:
<profile>

Now, please assess the social support
for the this patient, return in the
required format of response:
```

Self-Instruct Prompt Template

```
Please generate similiar response as
the example below. Follow the exact
same format of example response.

---

Here is an example input:

Age: 67
Marital Status: Widowed
Number of Previous Visits: 12 visits
Arrival Type: Emergency
Arrival Way: Emergency Room

---

Here is an example response:

Assessment:
1. How can this patient's age
impact support?
At 67, the patient is a
senior, which might indicate decreased
social support due to retirement,
potential loss of peers, and
a possible reduction in physical
ability. Seniors often rely on
family or community services, and
being widowed can further reduce
their social network.

2. How might marital status
impact social support?
Being widowed likely indicates a
significant loss of close social
support, particularly the loss of
a primary caregiver. This could
mean the patient has fewer people
to rely on for emotional support,
which might contribute to increased
hospital visits and emergency
situations.

3. How might previous visits
impact social support?
With 12 visits in the past
year, this patient appears to
have frequent health issues that
require medical attention. This
frequency might suggest insufficient
social support to manage their
health.

4. How might arrival type
impact social support?
The emergency arrival type indicates
that the patient likely faced
an urgent health issue. This
might suggest that their social
support system is inadequate.

5. How might arrival way
impact social support?
Arriving through the emergency
room indicates an immediate need
for medical attention, possibly
pointing to a lack of social
support to manage their health.

Summary:
This patient, a 67-year-old widowed
individual with 12 visits in
the past year, arriving via
emergency room, likely has limited
social support. The combination of
age, marital status, and frequent
visits suggests they may lack
adequate support.
```

Result (Only choose from Weak or Strong. Force to choose one): Weak

----

Now, please follow the EXACT same format and generate rationales assessing social support for the following patient with discharge location of <<DISCHARG>>:

Age: <<AGE>>
Marital Status: <<MaritalStatus>>
Number of Previous Visits: <<Number>>
Arrival Type: <<ArrivalType>>
Arrival Way: <<ArrivalWay>>