Enhancing Predictive Model Learning via Domain-Knowledge Augmented Latent Feature Mining

Bingxuan Li[1, 2], **Pengyi Shi**[3], **Amy R Ward**[4]
[1]University of Illinois Urbana-Champaign
[2]University of California, Los Angeles
[3]Purdue University
[4]University of Chicago
bl61@illinois.edu, shi178@purdue.edu, amy.ward@chicagobooth.edu

Abstract

Predictive modeling in high-stakes domains often suffers from limited observed features due to ethical and practical constraints. To address this challenge, we propose a novel approach that formulates latent feature mining as a text-to-text propositional logic reasoning task, facilitating domain knowledge integration and improving the interpretability of latent features. We design *FLAME* (**F**aithful **L**atent Fe**A**ture Mining for Predictive **M**odel **E**nhancement), a domain knowledge-augmented latent feature mining framework, offering an efficient training paradigm to strengthen the domain-specific reasoning capabilities of large language models (LLMs) for latent feature inference. The goal of our framework is to augment observed features with inferred latent features, enhancing the performance of predictive models in downstream machine learning tasks. We validate our approach through two case studies: (1) the criminal justice system, where data collection is ethically challenging and inherently limited, and (2) the healthcare domain, where patient privacy concerns and the complexity of medical data restrict comprehensive feature collection. Experimental results demonstrate that the inferred latent features significantly enhance the performance of downstream classifiers by over 10%.

1 Introduction

Prediction plays a crucial role in decision making in many domains. Although traditional 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 criminal justice and healthcare, data availability is often limited, 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.

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 such as 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 such as healthcare or criminal justice, where explaining and justifying the predictions of a model is crucial to 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.

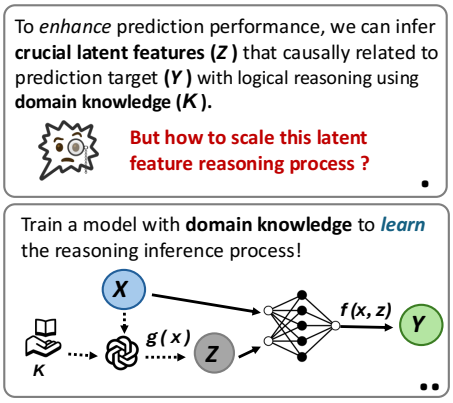

Figure 1: Motivation of *FLAME*: Predictive models suffer from poor performance with limited observable features (X). Domain experts can manually infer crucial latent features (Z) that causally relate to the prediction target (Y) through logical reasoning over observed features (X). We propose an effective learning paradigm to scale and automate this reasoning process by training the model that learns to infer latent features based on domain knowledge.

Figure 1 illustrates the motivation behind our approach to address these two limitations. Human experts can infer additional latent features that go beyond the explicit data

provided by drawing on their experience. For example, in the criminal justice system, predicting an individual's likelihood of in-program recidivism (the probability of committing a new crime during probation) is crucial for determining eligibility for incarceration-diversion programs (Rotter and Barber-Rioja 2015; Li et al. 2024). Typically, available data includes only basic demographic and criminal history information, but domain knowledge suggests that other factors – such as socio-economic status, community support, and psychological profiles – can significantly impact outcomes. Collecting such sensitive data raises ethical concerns, but human case managers can rely on their professional experience to infer these critical yet unrecorded details from observed data. 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.

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 the 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 three key advantages: **(1) Adaptive**: *FLAME* offers a more adaptive way to incorporate domain knowledge. Unlike rigid retrieval-based methods for domain knowledge augmented generation, which require predefined knowledge representations, our approach allows domain experts to directly encode their knowledge in natural language. This lowers the barrier for domain knowledge integration, enabling experts to provide explanations, reasoning patterns, and contextual insights in the form that the model can readily process and apply. **(2) Reliable**: *FLAME* provides an efficient training paradigm to guide LLMs to better interpret and utilize the domain knowledge with minimal human-annotated data. As a result, the model can more effectively align with domain knowledge to infer latent features that are *causally* related to the prediction target. **(3) Interpretable**: *FLAME* produces a human-like reasoning process, which is more interpretable outputs, making it particularly valuable in high-stakes domains requiring explainability.

We summarize our main contributions as follows.

1. We introduce a **novel approach that formulates latent feature mining as a text-to-text propositional logic reasoning task**, enabling the effective inference of interpretable latent features while lowering the *barrier* of domain-knowledge integration for human experts across domains.
2. We propose an **efficient training paradigm** that improves the model's ability to interpret and utilize domain knowledge, enhancing domain-specific reasoning with minimal human-annotated data.
3. We develop a **four-step domain-knowledge augmented framework** for latent feature mining to enhance predictive models across various domains, especially those with limited observed features and ethical constraints on data collection.

We empirically validate our framework through case studies in the criminal justice and healthcare domain, where latent features are crucial for enhancing prediction accuracy. The framework's strong performance across these distinct domains highlights its adaptability and broader applicability to similar challenges.

2 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 as EM and Variational Autoencoders (VAEs) offer alternative techniques to infer latent features from observed data. However, these approaches often produce results that are difficult to interpret and require strong parametric assumptions. We summarize a comparison in table 1 to further distinguish the difference between *FLAME* and existing approaches for enhancing predictive models from a data/features perspective.

Fine-tuning with Synthetic Training Data Fine-tuning 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 finetuning 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 augment training data with synthetic samples.

Domain-Knowledge Augmented Reasoning. Rather than relying solely on the inherent knowledge of LLMs, integrating external knowledge during the reasoning process can improve both robustness and accuracy (Guu et al. 2020; Lewis et al. 2020). Prior research on knowledge-augmented language models has explored leveraging external knowledge bases (Chen et al. 2017; Izacard et al. 2023; Zhang et al. 2023; Borgeaud et al. 2022), such as retrieving relevant knowledge based on the input query (Chen et al. 2017; Kang et al. 2024), and the knowledge distillation technique to enhance retrievers (Kang et al. 2024). Additionally, researchers have investigated injecting domain-specific knowledge to

Methods	Approach	Interpretability	Domain Knowledge Integration Capability
Data Augmentation (GANs, VAEs)	increasing sample size	×	×
Latent Feature Mining (EM)	extracting (new) latent features	×	×
Dimension Reduction	reducing feature size	×	×
FLAME	extracting (new) latent features	✓	✓

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.

improve knowledge alignment in specialized fields (Liang et al. 2024). However, existing methods may fall short in real-world applications: In many professional fields (e.g., criminal justice, medicine, finance), effective reasoning relies on understanding domain-specific rules and abstract contextual information. Retrieval-based methods alone do not enforce alignment between LLMs and these complex knowledge representations. To bridge this gap, we introduce a two-fold approach: (1) constructing an external domain-specific knowledge base with essential contextual information and (2) developing an effective learning paradigm that trains the model to interpret and utilize this contextual information more effectively.

3 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 the 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 $f : \mathcal{X} \to \mathcal{Y}$ 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.

Definition of Latent Features. 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. One can learn the latent features from 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 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 logistic 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 3.1. *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.

4 Method

We propose a new approach, *FLAME*, to efficiently and accurately extract latent features and increase observed features to improve the accuracy of downstream prediction. 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 2 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

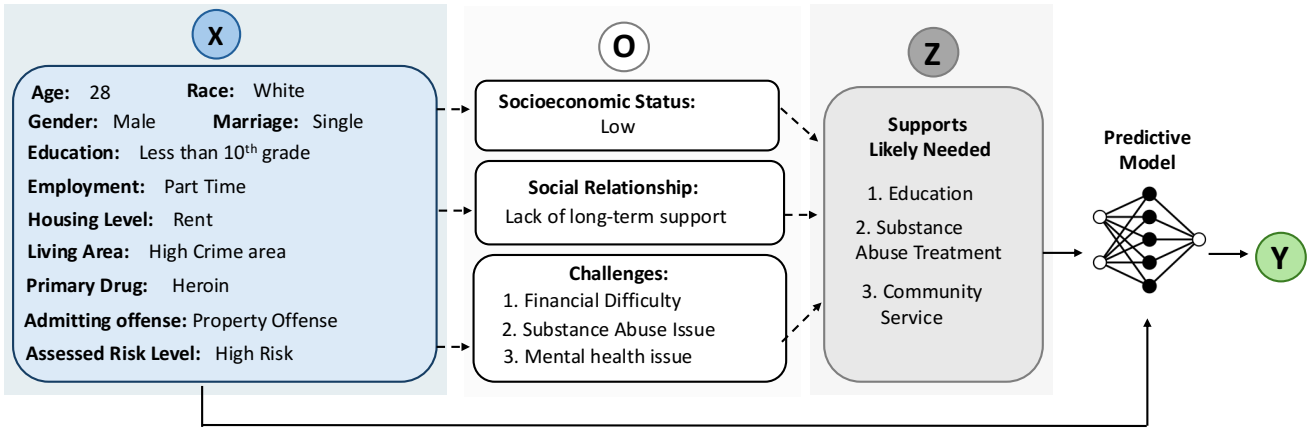

Figure 2: Example of latent feature mining through chain of reasoning. The latent feature "Supports Likely Needed" (Z) is inferred from the observed input features (X) via intermediate predicates (O), and is then used alongside X to improve the prediction for outcome (Y).

logic implication $(P_1 \wedge P_2 \wedge \ldots \wedge P_m) \rightarrow Z$ 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:

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

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 $(P_1 \wedge P_2 \wedge \ldots \wedge P_m) \rightarrow O_l$ 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 $(O_1 \wedge O_2 \wedge \ldots \wedge O_\ell) \rightarrow Z$. 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).

We use a hypothetical example from our case study setting to illustrate the formulation of the logic chain. The blue (leftmost) box in Figure 2 shows the observed feature X for one individual. Examples for the predicates P's formulated from X could be:

> P_1 :*"the client has part-time job"*, P_2 : *" the client hasn't complete high school"*, P_3 :*"the client is single"*, P_4 : *"the client has drug issue"*, P_5 :*" the client lives in high crime area"*, P_6 : *" the client is assessed with high risk" ...*

To infer the latent feature Z – in this example, the support likely needed during probation – we go through a multi-stage reasoning to infer the intermediate predicates O's; see the white (middle) boxes in Figure 2. One example logic that connects P's to O's could be:

> P_1 = *"The client has unstable employment"*
> P_2 = *"The highest education level of client is less than 10th grade"*
> O_1 = *"The client has low socioeconomic status"*
> If $(P_1 \wedge P_2 \rightarrow O_1) \in S$, then O_1 is True.

Finally, with P's and O's, we can connect X with Z though the logic chains. One example of the logical chain is as follows:

> *"The client is grappling with unstable employment and a relatively low educational level, factors that likely contribute to a low socioeconomic status. Additionally, being single, struggling with drug issues, and residing in a high-crime area further exacerbate the lack of positive social support. Given these circumstances, education could be valuable. Community service can be particularly beneficial for someone who is single and may lack a broad support network. Substance abuse treatment is crucial for individuals from lower socioeconomic backgrounds to aid in recovery from substance abuse. Hence this client likely needs support on education, substance abuse treatment, community service."*

Here, *"unstable employment and a relatively low educational level"* and *"being single, struggling with drug issues, and residing in a high-crime area"* are P's extracted from the features X, while *"a low socioeconomic status"* and *"lack of positive social support"* are O's. Finally, the rationales *"education could be valuable ... recovery from substance abuse. Hence this client likely needs support on education, substance abuse treatment, community service"* connect the intermediate predicates to the latent variables Z (supports likely needed) we want to infer, i.e., Z_1='education', Z_2='substance abuse treatment', Z_3='community service'.

Figure 3 illustrates the full process of *FLAME* with four steps.

(1) Rationales Formulation: The first step is to formulate rationales with domain knowledge. This involves two sub-steps: **The first sub-step** develops baseline rationales by identifying observed features potentially correlated with

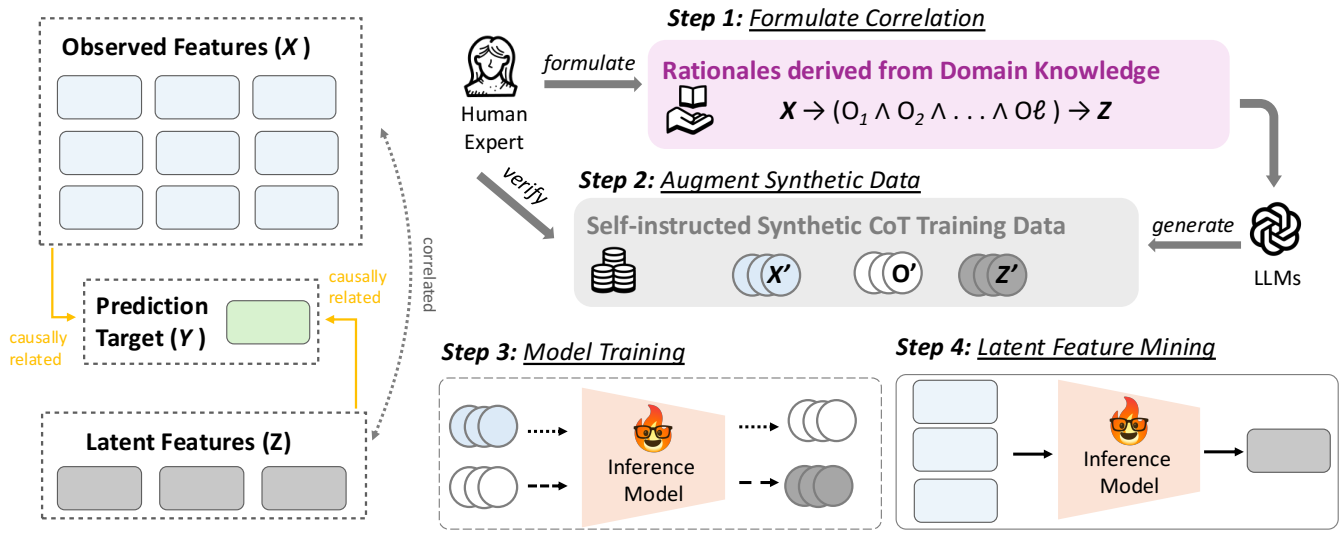

Figure 3: Overview of *FLAME*: The framework consists of four steps: (1) Rationales Formulation. (2) Training Data Augmentation. (3) Model Training. (4) Latent Feature Inference.

latent features and formulating their relationships—i.e., the logical connections linking X to Z. These rationales leverage established correlations (e.g., risk score formulas), expert insights, and domain-specific contextual factors such as neighborhood socio-economic status. This critical step enables our framework to efficiently integrate external domain knowledge, guiding LLMs to accurately infer latent features from observed data. **In the second sub-step**, we iteratively craft a prompt template crucial for establishing accurate reasoning to infer latent features and generate synthetic rationales in Step 2. Initially designed by domain experts to ensure alignment with baseline rationales, the prompt template undergoes iterative zero-shot testing. When the LLM fails on specific examples, ground-truth feedback facilitates prompt refinement (Miao, Teh, and Rainforth 2023). This iterative process continues until the model consistently produces desired outputs, effectively contextually grounding the LLM outputs with domain knowledge.

(2) Training Data Augmentation: We generate synthetic training data using a self-instruct approach (Wang et al. 2022). Starting with a small set of baseline rationales as references, we guide LLMs to generate new rationales using in-context learning. To maintain data quality and diversity, we incorporate human-in-the-loop interventions, filtering out low-quality or invalid samples based on heuristics. Additionally, we apply automatic evaluation metrics for quality control, such as removing samples that lack essential keywords. This step **efficiently augments the training data required for models to further strengthen in-domain reasoning ability at a low cost**.

(3) Model Training: To strengthen the domain-specific reasoning capabilities of LLMs and better align their outputs with specific domains, we fine-tune the model using the processed dataset from the previous steps . This step is both essential and effective for enhancing domain-specific reasoning. By explicitly encoding expert-driven rationales and contextual information, **the model internalizes domain knowledge, allowing it to generalize beyond direct retrieval and reason in a more reliable and adaptive manner.** Without this fine-tuning step, the model would rely solely on retrieval-based methods for domain knowledge augmentation, which limited the understanding of nuanced and complex domain knowledge. In addition, we validate the necessity and impact of this step through an ablation study, demonstrating that it significantly improves in-domain reasoning performance for latent feature inference.

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

5 Experiments Setup

We design two case studies to empirically investigate the following questions: (1) Can *FLAME* accurately mimic human reasoning to infer latent features? (2) Does *FLAME* improve the performance of downstream prediction tasks?

5.1 Case Study 1: Incarceration Diversion Program Management

In this case study, we test the efficacy of *FLAME* in the criminal justice domain. We conduct evaluation on a unique dataset from a state-wide incarceration diversion program.

Task Description. The outcome prediction task involves using individual data to predict the most likely outcome of clients upon termination from the incarceration diversion program. In this task, we treat the "support likely needed" (e.g., substance treatment, counseling) for each client as the latent features Z and use them to augment the original feature X for outcome pre-

diction, which is a multi-classification problem to learn $Y \sim f(X, Z)$ among four labels for the outcome $Y \in \{Completed, Revoked, NotCompleted, Other\}$. Based on domain knowledge, the inferred characteristics are indeed beneficial and not detrimental (recall the results in Lemma 1). The raw dataset does not record this feature; thus, Z in this task is unobservable. Available support program options for this task are detailed in the Supplementary Material.

Implementation Details. We implement our proposed framework as follows. All prompt templates are available in Supplementary Material. We train ML classifiers to predict outcomes with and without the inferred latent features, and then evaluate their out-of-sample accuracy.

- 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, socio-economic status, and challenges in two or three sentences. We then merge these sentences into the client's profile. We use zero-shot prompting with GPT-4.

- Step 1. Formulating rationales: We invite domain experts to formulate 40 baseline rationales to deduce "support likely needed" from the features of the client. We leverage a multi-stage reasoning strategy (Qiao et al. 2022) to decompose the task into three sub-tasks: (1) identify the main challenges from the client's profile, (2) rank these challenges by priority, (3) match the challenges with suitable programs. Particularly, the third task is our main goal, with the first two serving as steps to streamline the process and simplify the task.

- Step 2. Augmenting Training Data: With the 40 baseline rationales, we generate additional synthetic rationales. We sample client features 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. Training domain-specific alignment: We trained the pretrained language model GPT-3.5 (OpenAI 2021). We use the OpenAI API to fine-tune GPT-3.5-turbo-0125.

- Step 4. Latent Feature Inference: We leverage trained LLMs to infer "support likely needed" $\hat{Z}_i$ from features X_i for each client i in the test data.

5.2 Case Study 2: Healthcare Management

In this case study, we test the efficacy of *FLAME* in the healthcare domain. We conducted experiments on the MIMIC dataset (Johnson et al. 2016), a comprehensive dataset containing detailed clinical data from de-identified patients.

Task Description. The discharge location prediction task involves using individual patient-level data to predict the most likely discharge destination for patients upon discharge from 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.

Implementation Details. We repeat the four-step process of our framework: Steps 0 and 2-4 remain almost the same as in the previous task. Step 1 requires a slight adjustment (as discussed in Section 4, this step is the main part of our framework that requires customization). Here, in step 1, we invite domain experts to craft rationales to infer social support in Step 1. We chose GPT-3.5 as the base inference model of *FLAME* for this task. We train ML classifiers to predict outcomes with and without the inferred latent features, and then evaluate their out-of-sample accuracy.

6 Experiments Results

Case study 1: Outcome Prediction We compare the performance of the downstream classifiers that are trained with and without the latent features. As illustrated in Table 2, incorporating latent features significantly improves the performance of the downstream classifiers. Furthermore, the feature importance in Figure 4 shows that the inferred features – 'Support_1', 'Support_2', and 'Support_3' – are among the top-ranked features. This implies the significant relevance of these features on the downstream classification task. Hence, we can conclude that **our approach has the capability of enhancing the downstream classifier's accuracy with inferred latent features**.

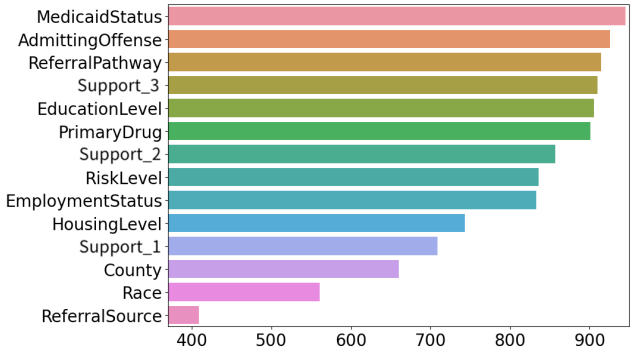

Figure 4: Feature importance plot of the outcome prediction task

Case Study 2: Discharge Location Prediction Table 2 demonstrates the result of the 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. Specifically, the GBT model achieves the highest accuracy after incorporating 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, as shown in Figure G.4 in the appendix, the inferred variable "Social Support" shows a strong correlation with the discharge location. This finding suggests that *FLAME* can uncover meaningful latent features. More importantly, **this experiment on a different dataset from a different domain demonstrates the effectiveness and generalizability**

Model	Outcome Prediction		Discharge Location Prediction	
	ROC_AUC (std.)	F1 Score (std.)	ROC_AUC (std.)	F1 Score (std.)
Logistic Regression (LR)	70% (0.01)	70% (0.01)	65.22% (0.01)	65.46% (0.01)
Multi-layer Perceptron (MLP)	81% (0.01)	70% (0.01)	63.19% (0.02)	63.19% (0.02)
Gradient Boosted Trees (GBT)	84% (0.01)	71% (0.01)	64.84% (0.01)	65.09% (0.01)
Random Forests (RF)	83% (0.01)	70% (0.01)	65.11% (0.01)	65.44% (0.01)
LR w/ Latent Feature	**85% (0.02)**	**75% (0.01)**	**71.22% (0.01)**	**71.26% (0.01)**
MLP w/ Latent Feature	**88% (0.01)**	**73% (0.01)**	**74.40% (0.01)**	**74.50% (0.01)**
GBT w/ Latent Feature	**92% (0.01)**	**77% (0.01)**	**75.56% (0.02)**	**75.38% (0.02)**
RF w/ Latent Feature	**90% (0.01)**	**75% (0.01)**	**75.31% (0.01)**	**75.22% (0.01)**

Table 2: The experiment result for Outcome Prediction and Discharge Location Prediction task. We use five different random seeds to run experiment five times and report the average.

First Evaluation		Second Evaluation	
Metric	**Mean ± Std**	**Category**	**Result**
Correctness	4.48 ± 0.34	Human-crafted misclassified as *FLAME*-generated	75%
Interpretability	4.36 ± 0.41	*FLAME*-generated misclassified as human	60%
Similarity to Human Reasoning	4.22 ± 0.39	—	—

Table 3: Human evaluation results of model-generated rationales. The first evaluation includes human-rated metrics; the second evaluation measures confusion between human and *FLAME*-generated rationales.

of *FLAME*.

7 Trustworthiness Evaluation

In this section, we design two additional studies to evaluate the reliability and trustworthiness of *FLAME*. In Appendix D, we conduct additional **ablation studies** for in-depth analysis to evaluate the sensitivity, effectiveness, and significance of each component of *FLAME*.

7.1 Human Evaluation

Evaluation Setting To assess the quality of *FLAME*-generated rationales, we conducted two human evaluations. In the first evaluation, we randomly selected 50 *FLAME*-generated rationales from Experiment 1 and Experiment 2 of the criminal justice case study. We recruited six criminal justice experts and two students without prior knowledge of this work to annotate each rationale on a 5-point Likert scale. The annotators were asked to evaluate the rationales based on the following three criteria:

- Correctness: Do the rationale leads to a valid and reasonable latent inference?
- Interpretability: How clear and understandable the reasoning is?
- Similarity to Human Reasoning: How closely the rationale replicate that of a human expert?

In the second evaluation, we aimed to determine whether humans could distinguish between rationales generated by *FLAME* and those written by humans. We mixed the 20 sampled *FLAME*-generated rationales with 20 additional human-crafted rationales. The same five annotators were then asked the following question for each rationale: *Do you think this reasoning step was generated by a human or an LLM?*

Evaluation Results. Table 3 presents the results of both evaluations. For the first evaluation, the results indicate that *FLAME* demonstrates a strong ability to generate rationales that align with human reasoning and produce accurate conclusions. For the second evaluation, we can see human evaluators struggled to distinguish them from human-written ones, further validating the quality of *FLAME*-generated rationales. In conclusion, the results from both evaluations suggest that *FLAME* can effectively mimic human-crafted rationales in both reasoning accuracy and indistinguishability from human-generated reasoning.

7.2 Automatic Evaluation

In this study, we consider an individual's observed feature as a latent feature, even though it is recorded in the dataset (i.e., ground truth labels are available). This experiment evaluates whether the inferred latent features, $\hat{Z}$, generated by *FLAME* align with the actual features, Z.

Evaluation Setting We treat an observed feature – Risk Level – as the latent feature to infer. The task is a multi-classification problem to learn $Z \sim g(X)$ among four labels for the latent variable $Z \in \{moderate, high, very\_high\}$ based on each client's profile X. We invited domain experts to craft rationales for this task in step 1. The rest of the steps remain the same as previous experiments. We choose GPT-3.5 ($FLAME_g$) and LLaMA2 ($FLAME_l$) as base models. We compare against (1) traditional machine learning models (2) a strong baseline that uses an MLP classifier trained on embeddings from a text summarization model.

Evaluation Results *FLAME*$_g$ achieves 77.01% accuracy, and *FLAME*$_l$ achieves 63.08% accuracy, outperforming traditional machine learning models : GBT (58.24%), RF (56.02%), MLP (52.03%), and LR (50.12%). To further illustrate the advantage of our approach, we evaluate the embedding-based MLP baseline on a balanced validation set with 50 samples per class. It achieves only 51.88% accuracy.

The relatively poor performance of traditional machine learning models and baseline may be attributed to the weak correlation between the observed features and the target variable (risk level). In contrast, our approaches are able to capture complex and non-obvious relationships between inputs and outcomes. These results indicate that **our method enables more accurate inference of latent features**, which standard models fail to uncover. Here, we emphasize that our LLM-based model, with millions of parameters, is *not* claimed to universally outperform traditional machine learning models. Rather, **our goal is to demonstrate its ability to capture latent features**, which can complement existing methods.

8 Discussion and 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, which makes it valuable for decision-making.

What is required to generalize *FLAME* for new domains? *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—is fundamental to our framework and requires domain expertise and additional manual effort. This effort is not a drawback but a deliberate design choice that sets our approach apart from other methods. ***FLAME* enables human experts to supply external domain knowledge and guide the inference model in interpreting and utilizing this information, strengthening the in-domain reasoning capabilities**.

To elaborate, in the outcome prediction task, we collected publicly available socio-economic data associated with different zip codes as domain-specific contextual information. We evaluated three settings to assess the impact of domain knowledge provided by Step 1 of *FLAME*: First, we provided both zip code and socio-economic data with explicit instructions, enabling the model to extract useful latent features and improve prediction accuracy. In the second setting, we removed the socio-economic information, which significantly harmed the model's ability to extract relevant features. Lastly, we asked the model to generate contextual information for zip codes using only its internal knowledge (GPT-4). Out of 50 zip codes, 5 remained undetermined, 17 contained hallucinated (incorrect) details, and only 33 were accurate (see Appendix E for examples), which aligns with recent research showing that LLMs are unreliable as standalone knowledge bases (He, Wang, and Wang 2024; Zheng, Lapata, and Pan 2024). Overall, these results demonstrate that although our approach requires some manual effort, it is a valuable and worthwhile investment for enhancing predictive modeling with latent features.

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*.

Borgeaud, S.; Mensch, A.; Hoffmann, J.; Cai, T.; Rutherford, E.; Millican, K.; van den Driessche, G.; Lespiau, J.; Damoc, B.; Clark, A.; de Las Casas, D.; Guy, A.; Menick, J.; Ring, R.; Hennigan, T.; Huang, S.; Maggiore, L.; Jones, C.; Cassirer, A.; Brock, A.; Paganini, M.; Irving, G.; Vinyals, O.; Osindero, S.; Simonyan, K.; Rae, J. W.; Elsen, E.; and Sifre, L. 2022. Improving Language Models by Retrieving from Trillions of Tokens. In *International Conference on Machine Learning, ICML*, 2206–2240.

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.

Chen, D.; Fisch, A.; Weston, J.; and Bordes, A. 2017. Reading Wikipedia to Answer Open-Domain Questions. In Barzilay, R.; and Kan, M., eds., *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, 1870–1879. Association for Computational Linguistics.

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.

Guu, K.; Lee, K.; Tung, Z.; Pasupat, P.; and Chang, M. 2020. Retrieval Augmented Language Model Pre-Training. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, 3929–3938. PMLR.

He, Q.; Wang, Y.; and Wang, W. 2024. Can Language Models Act as Knowledge Bases at Scale? *arXiv preprint arXiv:2402.14273*.

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*.

Izacard, G.; Lewis, P.; Lomeli, M.; Hosseini, L.; Petroni, F.; Schick, T.; Dwivedi-Yu, J.; Joulin, A.; Riedel, S.; and Grave, E. 2023. Atlas: Few-shot Learning with Retrieval Augmented Language Models. *Journal of Machine Learning Research*, 24(251): 1–43.

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.

Kang, M.; Lee, S.; Baek, J.; Kawaguchi, K.; and Hwang, S. J. 2024. Knowledge-augmented reasoning distillation for small language models in knowledge-intensive tasks. *Advances in Neural Information Processing Systems*, 36.

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

Lewis, P. S. H.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.; Rocktäschel, T.; Riedel, S.; and Kiela, D. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Li, B.; Castellanos, A.; Shi, P.; and Ward, A. 2024. Combining Machine Learning and Queueing Theory for Data-driven Incarceration-Diversion Program Management. In *Proceedings of the Thirty-Sixth Annual Conference on Innovative Applications of Artificial Intelligence*. AAAI.

Liang, L.; Sun, M.; Gui, Z.; Zhu, Z.; Jiang, Z.; Zhong, L.; Qu, Y.; Zhao, P.; Bo, Z.; Yang, J.; et al. 2024. Kag: Boosting llms in professional domains via knowledge augmented generation. *arXiv preprint arXiv:2409.13731*.

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. 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*.

Rotter, M.; and Barber-Rioja, V. 2015. *Diversion programs and alternatives to incarceration*. Oxford University Press.

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*.

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

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.; Muhamed, A.; Anantharaman, A.; Wang, G.; Chen, C.; Zhong, K.; Cui, Q.; Xu, Y.; Zeng, B.; Chilimbi, T.; and Chen, Y. 2023. ReAugKD: Retrieval-Augmented Knowledge Distillation For Pre-trained Language Models. In Rogers, A.; Boyd-Graber, J. L.; and Okazaki, N., eds., *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, 1128–1136. Association for Computational Linguistics.

Zheng, D.; Lapata, M.; and Pan, J. Z. 2024. Large language models as reliable knowledge bases? *arXiv preprint arXiv:2407.13578*.