LLM-in-the-Loop: Replicating Human Insight with LLMs for Better Machine Learning Applications

Anonymous ACL submission

Abstract

Building on the success of human-in-the-loop, where human wisdom is integrated into the development of machine learning algorithms, this position paper takes the initiative to envision an innovative and promising paradigm, **LLM-in-the-loop (LLM-ITL)**, which leverages the unique advantages of LLMs to replicate human involvement and offer a more flexible and cost-efficient solution to real-world challenges. Through a comprehensive review of LLM research from 2020 to 2025, we reveal that many existing LLM applications inherently align with LLM-ITL, with researchers rapidly claiming their superiority over machine learning baselines and LLM-native solutions; however, no universal definition exists, hindering its further advancement and application. In this paper, we define and categorize LLM-ITL methodologies for data, model, and task-centric applications, discuss their underlying rationale, and highlight emerging areas where LLMs can be further integrated into the loop. Furthermore, we present opportunities for developing better LLM-ITL solutions with technical advancements, such as LLM crowdsourcing and text-to-solution, establishing the proposed paradigm as a promising avenue for the future of LLM applications and machine learning research.

1 Introduction

Human-in-the-loop has gained increasing popularity for solving real-world problems by integrating human knowledge and expertise into the development of machine learning models (Wu et al., 2022; Fang et al., 2023). With the recent emergence of Large Language Models (LLMs) and their products, such as ChatGPT and Claude, many researchers argue that LLMs not only significantly outperform traditional machine learning baselines, but also surpass human experts in many tasks (Pu et al., 2023; Törnberg, 2023; Gilardi et al., 2023). As LLMs evolve to become more agent-like and with the proven effectiveness of the "in-the-loop" techniques, a novel application paradigm, "**LLM-in-the-loop**" (abbreviated as **LLM-ITL**), has emerged as a focal point of interest for both academia and industry.

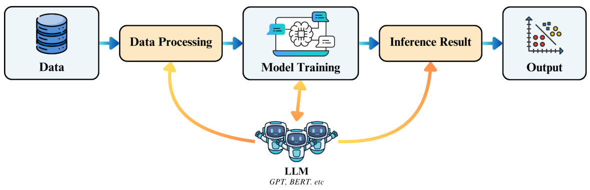

Figure 1: Overview: LLM-in-the-loop Paradigm

The term "LLM-in-the-loop" attracted considerable attention due to the expanding capabilities and popularity of LLMs, yet no universal definition exists in the current research landscape. Interpretations vary from narrowly defining it for specific tasks or methods (Yang et al., 2024b; Kholodna et al., 2024) to adopting an overly broad scope that might generalize the concept (Sudhakar et al., 2024; Zhang et al., 2024b; Bartolo et al., 2020). This concept has also become a catchphrase to align with LLM application trends (Wu et al., 2024; Keles et al., 2024), leading to ambiguity and confusion. Appendix A presents a comprehensive list of research papers collected up to May 1, 2025, featuring the keywords LLM-in-the-loop" or LLM-ITL" in their titles or abstracts. These examples underscore the increasing interest in LLM-ITL applications. However, without a clear definition, there is a lack of understanding of how to effectively utilize LLMs. This lack of clarity limits their generalizability and hinders the recognition of their broader potential to enhance various stages of the problem-solving pipeline.

In this position paper, we envision the future of LLM applications and position that **the LLM-in-the-loop paradigm, which harnesses the strengths of both LLMs and conventional**

machine learning algorithms, presents prevailing research opportunities and practical advantages. Through comprehensive literature reviews and detailed case study analyses, we demonstrate the growing popularity and effectiveness of this framework, as evidenced by the widespread, although often unconscious, application of its methodologies and the resulting state-of-the-art performances. However, this increased visibility also highlights concerns about a lack of public understanding, motivating our efforts in providing the first formal definition, various motivations, and a comprehensive taxonomy of methodologies.

Contribution. This paper is the first to provide an in-depth discussion on the LLM-in-the-loop paradigm, establishing it as a promising framework for the future of LLM applications in addressing real-world problems. The key contributions include: 1) We present practical scenarios where directly applying LLMs for problem-solving results in suboptimal outcomes, highlighting the importance of integrating conventional machine learning algorithms in the era of LLMs[1]; 2) By synthesizing insights from related concepts and examining the implementation of existing in-the-loop methodologies, we formulate the LLM-in-the-loop framework from three perspectives, providing a foundation for future research; 3) We identify challenges in developing effective LLM-in-the-loop solutions and present promising avenues for future research and impactful applications, guiding the research community towards an underexplored landscape of LLM application and machine learning research.

2 Background

2.1 LLMs: Trends and Challenges

LLM Applications. Among diverse areas of LLM research, the study of "Applications of LLMs/ChatGPT" has emerged as the fastest-growing topic since 2023 (Movva et al., 2024). These applications increasingly adopt a **Model-as-a-Service (MaaS) paradigm** (Sun et al., 2022; Gan et al., 2023), also referred to as LLM-native solutions (Liang et al., 2024), which deliver a broad range of functionalities through easily accessible interfaces. As generative language models, LLMs excel in tasks that are inherently sequence-to-sequence (seq2seq) (Vaswani et al., 2017), such as natural language comprehension, translation, and generation (Sottana et al., 2023; Bahdanau et al., 2015; Sutskever et al., 2014; Lewis et al., 2020a). However, extending their application to real-world problem-solving presents significant challenges (Chen et al., 2025a), as these tasks often diverge from the fundamental nature of language modeling and extend beyond the scope of NLP (Srivatsa and Kochmar, 2024; Chen et al., 2024d). Even for tasks that appear NLP-relevant, such as text clustering and topic modeling, the underlying processes do not naturally conform to a seq2seq setting, often relying more on representation learning and optimization rather than generative capabilities (Bengio et al., 2013).

While much of the application-driven research advancements focus on developing better LLMs and innovative engineering techniques (Chen et al., 2023), such as prompt engineering (Song et al., 2024; Brown et al., 2020), model fine-tuning (Hu et al., 2022), and Retrieval Augmented Generation (Lewis et al., 2020b), commendable research efforts are also being made to explore the use of existing state-of-the-art LLMs or smaller, more cost-efficient models (Xu et al., 2024), within **better-designed problem-solving workflows**, such as LLM-chaining (Grunde-McLaughlin et al., 2024) and multi-agent collaboration (Hong et al., 2024c). Task decomposition techniques have further emerged as a promising solution for complex, multi-step tasks (e.g., planning a wedding) (Yuan et al., 2025; Huang et al., 2023), where prompting-based LLMs and machine learning algorithms collaborate effectively in solving well-structured sub-tasks (Khot et al., 2023).

LLM vs. Human. With LLMs demonstrating increasing capabilities across various benchmark evaluations, especially when provided with clear instructions and demonstrations, He et al. (2024) pose a critical and significant inquiry: **Can LLMs potentially replace crowdsourced annotators?** Törnberg (2023) finds that GPT-4 achieves higher accuracy, greater reliability, and equal or lower bias than human classifiers when given the same instructions for tweet classification. This emphasizes the relatively low technical requirements of deploying LLM, as the instructions initially provided to human workers can be reused. Similarly, Gilardi et al. (2023) demonstrates that zero-shot GPT-3.5 outperforms certified "MTurk Masters" high-ability crowd workers in text-annotation tasks. Cegin et al. (2023) suggests that ChatGPT can per-

[1] Code and reading list are available at GitHub repository.

form data augmentation with greater lexical and syntactic diversity than human workers, resulting in reliable downstream performance where models trained on ChatGPT-generated data exhibit comparable robustness to those trained on data from human crowds. With comparable performance, the resource efficiency of LLM demonstrates substantial advantages. Gilardi et al. (2023) reveals that employing an LLM for data labeling is cost-effective, with the per-annotation cost of ChatGPT being 30 times cheaper than MTurk. Additionally, Cegin et al. (2023) claims that substituting human workers with LLMs for generating new data instances is 600 times cheaper.

Incapabilities of LLM. While LLMs excel in numerous tasks, practical scenarios exist where they either underperform or prove infeasible compared to traditional machine learning methods (Liu et al., 2024b). Besides common issues like hallucination and bias, LLMs also face issues in generating answers within a deterministic space (Kholodna et al., 2024). This has been observed in many studies (see example in Section 3) but remains largely unexplored by the research community due to a lack of clear problem formulation. We argue that the misbehavior of LLM is largely due to the absence of a hard-coded solution space, which is often weakly specified through instruction prompts (Zeng et al., 2024), unlike traditional machine learning that strictly binds the solution space and model behavior. To formally define this limitation and motivate further research, we formulate the problem abstraction as follows:

Definition 2.1. Given input data $\mathcal{D}$, targeted solution space $\mathcal{S}$, and an instruction prompt $\mathcal{P}(\mathcal{S})$ specifying solution space, the failure occurs when:

$$\begin{aligned} LLM(\mathcal{P}(\mathcal{S}), \mathcal{D}) \subseteq \mathcal{R} \\ \text{where} \quad ||\mathcal{R} - \mathcal{S}||^2 > \epsilon \end{aligned} \tag{1}$$

where the generated result space deviates significantly from the targeted solution space, exceeding a threshold ϵ, which can be numerical discrepancies (e.g., answer counts or ranges) or qualitative inconsistencies (e.g., misalignment in format).

2.2 In-the-loop Methodologies

Human-in-the-loop. Human-in-the-loop, with "loop" generally implies the problem-solving process, is a well-established approach for incorporating human expertise (Agarwal et al., 2023) into automated modeling processes to enhance the accuracy of predictive models (Kumar et al., 2019), with proven performance improvement and enhanced interpretability in various tasks such as sentence parsing (He et al., 2016), topic modeling (Kumar et al., 2019), and text classification (Arous et al., 2021). Extensive research efforts have explored HITL workflows in machine learning, focusing on data preprocessing, model training, and system-independent application (Wu et al., 2022). Moreover, HITL is particularly beneficial when machine learning models encounter difficulties with complex, nuanced, or ambiguous tasks that demand prior knowledge (Diligenti et al., 2017) and contextual understanding (Mosqueira-Rey et al., 2022).

Definition of LLM-in-the-loop. Drawing inspiration from the close relationship with human-in-the-loop, **the LLM-in-the-loop paradigm is defined as the integration of LLM interaction, intervention, and judgment to guide or modify the training and inference processes of a machine learning model**. While it mirrors the human-in-the-loop process by substituting human participation with LLM agents, the inference remains the responsibility of the machine learning model rather than the LLM agent, distinguishing it from LLM-native or LLM-ML collaboration, where the LLM plays the central role. Notably, given the widespread availability and scalability of LLM agents compared to human workers, we argue that **LLM-in-the-loop offers broader applicability across training, inference, and deployment stages, positioning it as a more general framework that encompasses and extends existing in-the-loop methodologies**. In the following discussion, we demonstrate how LLMs can effectively replace the human role and provide additional benefits to the development of machine learning algorithms.

3 Case Study: LLM-ITL Text Clustering

Human-in-the-loop methodologies have been extensively applied in clustering problems to integrate prior knowledge into unsupervised learning (Coden et al., 2017; Srivastava et al., 2016; Holzinger, 2016). Recently, the development of LLM-in-the-loop solutions for text clustering has rapidly emerged, achieving state-of-the-art performance by leveraging the language understanding capabilities of LLMs. This serves as a great starting point in analyzing existing methods for guiding the future design of LLM-ITL solutions.

Observation and Motivation. The research community appears inherently aware of the limitations in directly applying LLMs for text clustering, as evidenced by the observation that existing studies rarely consider LLM-native baselines but compare solely with conventional machine learning algorithms when developing LLM-ITL solutions (Viswanathan et al., 2024; Hong et al., 2024a; Zhang et al., 2023b). To fill in the gap of missing LLM-native results, we present an empirical study in Appendix B. Notably, the clustering problem has a strict solution space defined by n instances k candidate labels. Our findings reveal that over 90% of the LLM-generated results fail to capture the targeted number of labels and are misaligned with the input instances. Both the instruction prompt and input data affect inference behavior, yet the problem remains unsolved even with state-of-the-art prompt tuning techniques (Agarwal et al., 2024) and in simple clustering settings. This motivates the development of LLM-in-the-loop solutions that rely on machine learning algorithms to produce cluster assignments under the targeted solution space.

LLM-in-the-loop Solutions. ClusterLLM represents a pioneering LLM-in-the-loop solution for text clustering (Zhang et al., 2023b), addressing the limitations of LLM-native approaches in having restricted access to embedding vectors. API-based LLM is prompted to respond to pairwise preference questions structured as a triplet, consisting of two candidate instances and a reference anchor. These preferences are used to fine-tune an embedder, ensuring the input corpus is mapped to a refined embedding space for better clustering. This outlines a typical in-the-loop methodology where **the input data is preprocessed before the modeling process**. For instance, Viswanathan et al. (2024) augmented the input data through a keyphrase expansion strategy, generating a set of keyphrases that could describe document intent with LLM. The sentence and keyphrase embeddings are then concatenated to create a task-dependent data representation for better intent clustering. Similarly, Pattnaik et al. (2024) prompted a fine-tuned LLM to generate a concise cluster name and description for each cluster, then combining these embeddings with the cluster centroid embedding to create weighted multi-view representations, enhancing the performance of the agglomerative clustering algorithm in deriving topical categories within the documents.

Besides incorporating LLMs into the data preprocessing phase, Hong et al. (2024a) proposed the idea of iterative clustering with LLMs feedback, where initial cluster assignments obtained from K-means are evaluated by a fine-tuned LLM based on semantic coherence, and the poorly formed clusters are refined to enhance the final result. Similarly, Viswanathan et al. (2024) prompted LLM to select data instances that *must* be linked or *cannot* be linked, forming a pairwise constraint clustering with the PCKMeans algorithm. These approaches transform the original nature of unsupervised learning into an interactive or semi-supervised learning process, embodying a philosophy of designing LLM-in-the-loop solutions that **modify the modeling process with LLM-driven utilities**.

Furthermore, developing task-specific applications requires a task-oriented design. In the intent clustering problem, Hong et al. (2024a) proposed using LLMs to name clusters in the "action-objective" form, which enhances the usability of the clustered results and allows for further refinement based on either the action or the objective. Likewise, Viswanathan et al. (2024) utilized the reasoning capability of LLMs to assess whether a given low-confidence point belongs to the current cluster, performing post-correction on relocating the data point based on the LLM's judgment. These methods enable **further refinement of the modeling results with task-dependent LLM utilities**.

4 LLM-in-the-loop Methodologies

Based on the case study of LLM-in-the-loop solutions in text clustering, the methodologies can be categorized according to the specific purposes of LLM integration, namely: **data-centric, model-centric, or task-centric**. This framework enables a comprehensive exploration of the associated techniques and highlights opportunities for applying LLM-ITL methods in underutilized domains. Further discussions are presented in Appendix D, and extra case study is presented in Appendix C.

4.1 Data-Centric LLM-in-the-loop

The data-centric approach employs LLMs during the data preprocessing stage of machine learning modeling, with the goal of improving data quality, diversity, and representation to facilitate effective model training and address challenges inherent in traditional data preparation workflows.

Definition 4.1. Given an original dataset $\mathcal{D}_0$, learning function F, and a LLM-driven transformation function Φ_{LLM} guided by prompt $\mathcal{P}$, the data-centric approach aims to improve the task-specific loss $\mathcal{L}$ through data enhancement:

$$\begin{aligned} \text{Preprocess:} \quad & \mathcal{D}_{\text{tf}} = \Phi_{\text{LLM}}(\mathcal{D}_0, \mathcal{P}), \\ \text{Train:} \quad & M_{\text{tf}} = F(\mathcal{D}_{\text{tf}}), \\ \text{Target:} \quad & \mathcal{L}(M_{\text{tf}}) < \mathcal{L}(M_0) \end{aligned} \tag{2}$$

where the preprocessed dataset $\mathcal{D}_{\text{tf}}$ enables the training of model M_{tf} to achieve superior performance compared to the model trained on the original dataset, denoted as M_0.

Data Annotation. Data annotation is a fundamental step in supervised machine learning; however, the annotation process is labor-intensive and often suffers from inconsistent quality due to inherent biases and a lack of expertise (Pandey et al., 2022; Hettiachchi et al., 2021). Recent advancements demonstrate the potential of LLMs to revolutionize this process by offering efficient, high-quality, and scalable annotation solutions (Tan et al., 2024b), often matching or exceeding the quality achieved by crowdsourced annotators and domain experts (Gilardi et al., 2023; Kuzman et al., 2023; Törnberg, 2023). For instance, Chen et al. (2024c) showcased their effectiveness in event extraction, and Kuzman et al. (2023) highlighted ChatGPT's superior performance in automatic genre identification on unseen datasets. Innovative strategies, such as Chain-of-Thought (CoT) prompting combined with explain-then-annotate workflows (He et al., 2024), and CoT with majority voting (Choi et al., 2024), have further advanced LLM-based annotation methods, enabling human-like precision in complex tasks. Moreover, Smith et al. (2024) introduced the Prompted Weak Supervision, which leverages LLMs to generate probabilistic labels, reducing the need for manual intervention while maintaining high annotation quality.

Data Augmentation. Data augmentation is a critical yet complex task that goes beyond basic labeling, requiring the generation of diverse fundamental and auxiliary information tailored to specific task requirements (Rebuffi et al., 2021; Hong et al., 2024b). Although crowdsourcing can be used to address this need, producing reliable and high-quality augmented data poses a far greater challenge than data annotation, whereas conventional generative models also fall short of meeting these demands (Yang et al., 2023). In this context, LLMs present a transformative solution by generating diverse, contextually enriched synthetic datasets, significantly reducing the dependence on manual data collection. For example, Yu et al. (2024) introduced the use of attributed prompts to generate attribute-specific synthetic data, while Zou et al. (2024) proposed a collaborative framework utilizing multiple LLMs to create high-quality synthetic datasets. In addition, Choi et al. (2024) demonstrated the capability of LLMs to create domain-agnostic datasets, paving the way for universal domain generalization. Ba et al. (2024) also illustrated how synthetic data generation with LLMs can reduce calibration errors and improve accuracy on real-world test datasets.

Feature Engineering. Feature engineering transforms raw data into interpretable representations that enhance model performance (Hollmann et al., 2024). Traditional methods rely primarily on domain expertise, but the combinatorial complexity of manually exploring feature spaces renders this approach impractical (Gu et al., 2024). Recent advances leverage LLMs to automate and refine feature generation, producing semantically rich, context-aware features aligned with dataset characteristics and task objectives. For instance, Zhang et al. (2024c) introduced an LLM-driven framework for iterative feature generation and performance-guided refinement. Balek et al. (2024) further demonstrated that LLMs generate interpretable textual features surpassing traditional representations like bag-of-words or dense embeddings in discriminative power. Beyond text, LLMs can align diverse representations for structured learning tasks, such as converting environmental data into structured domain-specific language for agent learning (Spiegel et al., 2024) or encoding conversational turns into canonical forms to support domain-general dialogue policies (Sreedhar et al., 2024). Furthermore, Yang et al. (2024a) emphasized LLMs' versatility to generate task-relevant, linguistically grounded features, such as extracting subject-object pairs.

> **Our position:** From a data-centric perspective, LLM-in-the-loop benefits model training by alleviating data scarcity and enriching data features. The integration of LLMs in a crowdsourcing-like fashion has proven particularly effective, providing a valuable framework for developing "labor-free" in-the-loop solutions. Future research should focus on 1) identifying innovative approaches to integrate prior knowledge from LLMs into data features and 2) designing robust crowdsourcing approaches with LLM agent collaboration. These advancements hold the potential to significantly address the long-standing challenges of data availability and quality assurance.

4.2 Model-Centric LLM-in-the-loop

Incorporating rich human knowledge into machine learning models has been a longstanding research focus, as machine learning alone cannot fully capture the depth of human domain expertise (Wu et al., 2022). To address this, human-in-the-loop approaches integrate human insights by iteratively refining the model for knowledge-enhanced learning. In this section, we explore how LLMs can substitute for the human role to provide model-centric support.

Definition 4.2. Given a trained machine learning model M and LLM-driven utility Φ_{LLM} guided by instruction prompt $\mathcal{P}$, the model-centric approach aims to improve the task-specific loss $\mathcal{L}$ through model refinements:

$$\begin{aligned} \text{Refine:} \quad & M_{\text{tf}} = \Phi_{\text{LLM}}(M, \mathcal{P}), \\ \text{Target:} \quad & \mathcal{L}(M_{\text{tf}}) < \mathcal{L}(M) \end{aligned} \tag{3}$$

such that the refined model M_{tf} outperforms the original model M.

Active Learning and Iterative Refinement. Active learning is a crucial technique for integrating human wisdom and prior knowledge into iterative learning frameworks, especially in low-resource learning settings (Zhang et al., 2023a). Recently, there has been a growing interest in leveraging LLMs for both annotation and uncertainty estimation in an integrated active learning setting across various NLP tasks, such as text classification (Rouzegar and Makrehchi, 2024), named entity recognition, and relation extraction (Zhang et al., 2023a). Unlike data augmentation with LLMs, active learning is a model-based approach that focuses on uncertainty sampling - selecting data points where the model is most uncertain, thus allowing it to learn from challenging instances (Rouzegar and Makrehchi, 2024). While sample selection can be complex and necessitates human judgment, the concept of LLM confidence estimation offers a valuable alternative (Xiong et al., 2024; Geng et al., 2024), enabling verbalized confidence scores to assist the sampling process.

Beyond direct annotation, LLMs also provide a feedback mechanism in an iterative setting, addressing limitations in tasks where direct annotation is challenging (e.g., clustering). For instance, An et al. (2024) queried LLMs to identify true neighbors of selected samples from multiple candidates, leveraging this information for contrastive learning to improve base model representation. Similarly, Hong et al. (2024a) employed LLMs to iteratively refine poorly formed clusters through coherence evaluation at each iteration. In topic modeling, Yang et al. (2024b) used LLMs to refine topics generated by the base model, aligning the model with LLM-provided refinements through fine-tuning. These applications share the commonality of involving LLMs not only in the model training process but also in the inference and deployment stages, as most discussed applications pertain to unsupervised learning. This underscores another unique advantage of LLM-in-the-loop: its inherent model-in-the-loop nature, which offers deployment flexibility and facilitates application across diverse scenarios.

Reinforcement Learning. Reinforcement learning (RL) is a crucial segment of machine learning that seeks to align model behaviors with human expectations through a feedback mechanism (Cao et al., 2024). As LLM agents are increasingly calibrated to human behaviors and preferences through alignment techniques (Liu et al., 2024a; Wang et al., 2023), LLM-in-the-loop reinforcement learning has gained significant momentum. Existing research suggests that the prior knowledge of LLMs can be integrated into the RL process by serving as dynamic feedback sources, such as natural language instructions, demonstrations, evaluative signals, and informative guidance (Laleh and Ahmadabadi, 2024). For instance, Du et al. (2023) leveraged pre-trained LLMs to provide intrinsic motivation for RL agents by setting exploration goals and issuing rewards upon their completion. Similarly, Kwon and Michael (2023) employed LLMs as reward functions, where agent behaviors are evaluated against desired outcomes, generating corresponding reward signals. Barj and Sautory (2024) used LLM feedback to refine RL policies, particularly in scenarios where agents struggled to generalize to out-of-distribution environments.

In addition to reward setting, Karimpanal et al. (2023) utilized LLMs to generate decision-making behaviors, thereby accelerating the learning process. Similarly, Prakash et al. (2023) guided agent exploration by evaluating actions and behaviors based on observed states and task descriptions. In scenarios where RL agents need access to confidential information, Moradi et al. (2023) proposed integrating human-in-the-loop with Federated Learning. However, human involvement may still compromise data privacy and increase the cost of preventive measures. By introducing LLM-in-the-loop with locally deployed open-source LLMs, data privacy can be significantly enhanced, ensuring compliance with the principle of "keeping original data within the domain and making data available and invisible" (Yang et al., 2019). This approach further highlights the unique advantage of having a (large language) model-in-the-loop in constrained scenarios where human involvement is not preferred.

> **Our position:** LLMs demonstrate transformative potential in supporting knowledge-enhanced machine learning with iterative updating. They offer scalable and cost-efficient alternatives to traditional human involvement, facilitating deployable solutions due to their automated nature. However, the limitations of LLMs can be amplified by their direct interaction with the modeling process, leading to issues such as 1) poorly calibrated LLMs generating biased feedback and 2) failures in data sampling and labeling that create outliers in the iterative refinement process. These issues are difficult for machine learning models to unlearn and are hard to detect, unlike errors in data preprocessing.

4.3 Task-Centric LLM-in-the-loop

The task-centric approach employs LLMs as versatile and powerful utilities tailored for specific tasks or applications, focusing on enhancing task performance (e.g., prediction accuracy and interpretability). This section examines how LLMs can be strategically integrated into the inference and post-inference stages of problem-solving.

Definition 4.3. Given a trained machine learning model M, inference task T, and LLM-driven utility Φ_{LLM} guided by prompts $\mathcal{P}$, the task-centric approach aims to enhance task-specific performance evaluation U (e.g., accuracy, coherence) by incorporating LLMs during inference or post-inference evaluation stage:

$$\begin{aligned} \text{Inference:} \quad & \mathcal{O} = M(T), \\ \text{Support:} \quad & \mathcal{O}^* = \Phi_{\text{LLM}}(M, \mathcal{O}, \mathcal{P}) \\ \text{Target:} \quad & U(\mathcal{O}^*) > U(\mathcal{O}) \end{aligned} \tag{4}$$

where the LLM integration interacts with the model output and provides task-specific support, such as post-correction and explainability enhancement.

Post-Correction. Post-correction aims to improve machine learning predictions after the training process by refining model outputs with minimal local changes, a task where traditional methods often fall short due to their limited contextual understanding and scalability (Wei et al., 2024). With the extensive pre-trained knowledge in LLMs, Zhong et al. (2024) proposed using LLMs with in-context learning as post-hoc correctors to propose corrections for the predictions of machine learning models, enabling them to integrate contextual knowledge and deliver dynamic, context-aware corrections. In automatic speech recognition (ASR), CHEN et al. (2023) demonstrated the utility of LLMs in leveraging N-best hypothesis lists to predict the final output and found that LLM can correct errors even for tokens absent from the hypothesis list. Similarly, Hu et al. (2024) employed LLMs to synthesize diverse translation outputs from multiple N-best hypotheses, resulting in a substantial enhancement in translation quality. Beyond ASR, LLMs have been applied in clustering, where Viswanathan et al. (2024) re-ranked low-confidence points by querying their correctness against representative points, and Hong et al. (2024a) refined clusters by generating descriptive names and summaries using LLMs. In topic modeling, Chang et al. (2024) used LLMs to iteratively refine topics by identifying misaligned terms and replacing them with contextually appropriate alternatives. These attempts effectively integrate LLM in enhancing the task performance.

Model Interpretability. Machine learning models frequently struggle with interpretability, especially when generating natural language explanations or extracting actionable insights from outputs. Conventional techniques like feature importance scores and attribution maps focus on explaining model decisions but lack the capacity to interpret outputs through human-intuitive narratives (Pang et al., 2024). LLMs mitigate this gap by synthesizing their natural language understanding and generative capabilities to contextualize model outputs. For instance, Pattnaik et al. (2024) employed LLMs to generate descriptive cluster labels and summaries, while Hong et al. (2024a) and An et al. (2024) assign semantically meaningful names to clusters. In social media analysis, Islam and Goldwasser (2024) leveraged LLMs to summarize high-impact instances within clusters, producing cohesive "talking points" that directly supported downstream tasks like stance detection and demographic inference. Liu et al. (2023) explored the application of LLMs in evaluating text quality and open-ended responses, providing enriched insights by extracting additional features for metric evaluation. Additionally, Bhattacharjee et al. (2024) enabled causal explainability via LLMs by generating counterfactual explanations in black-box text classifiers, enhancing interpretability across complex ML workflows.

Our position: Designing better task-centric LLM-ITL solutions is becoming a scientific endeavor, presenting numerous new challenges and research opportunities. These include 1) replicating human-in-the-loop strategies while adapting to the unique characteristics of LLMs and 2) innovating LLM techniques to enhance their involvement in task-centric applications. Notably, LLMs often struggle with tasks involving token-level manipulation (Chen et al., 2024d), self-reflection (Xiong et al., 2024), and perceiving physical worlds (Fu et al., 2025), such as complex counting and verbalized confidence. These capabilities are believed to play an important role in developing trustworthy and explainable LLM-ITL solutions.

5 Discussion: Where Next

While LLMs have demonstrated significant potential in "in-the-loop" solutions, persistent limitations hinder their effectiveness in specialized sub-tasks. For instance, they struggle with direct computational tasks such as optimization and quantitative trading (Zhao et al., 2024), where precise numerical reasoning is critical. Furthermore, studies suggest that single LLM agents may underperform human experts in forecasting accuracy (Schoenegger and Park, 2023) and exhibit reliability concerns due to inherent model variability and biases (Kholodna et al., 2024). These limitations raise questions about the consistency of generated outputs - such as rewards or feedback - in high-stakes applications (Cegin et al., 2023). Motivated by these challenges, we highlight key future research directions to advance LLM-in-the-loop frameworks and bridge gaps in reliability and adaptability.

Crowdsourcing with LLM. In human-in-the-loop applications, crowdsourcing is often employed to leverage the "wisdom of the crowd" in solving problems through collaborative efforts (Tong et al., 2019; Zhang et al., 2013, 2014). With the increasing use of ChatGPT by crowd workers on MTurk (Veselovsky et al., 2023), we argue that the emergence of LLM-driven crowds, such as "LMTurk" (Zhao et al., 2022), offers a promising foundation for developing more robust LLM-in-the-loop solutions and benefiting the implementation of the aforementioned techniques and applications. This approach harnesses diverse knowledge from different LLMs, helping to reduce biases and errors that might occur when relying on a single model (Kholodna et al., 2024). Recognizing the growing popularity of multi-agent LLM systems (Guo et al., 2024; Hong et al., 2024c), designing LLM crowdsourcing solutions from a multi-agent perspective is a promising research avenue (Jiang et al., 2018). Additionally, leveraging well-established theories in crowdsourcing, such as crowd selection, task decomposition, and result aggregation (Zhang et al., 2024a; Bhatti et al., 2020), provides a comprehensive framework to guide future research directions and technical advancements in LLM multi-agent systems and the "science of LLM-in-the-loop."

Text-to-Solution with LLM. Recent advancements in text-to-code generation (natural language to code) have demonstrated its efficacy in automating problem-solving through code synthesis, requiring minimal programming expertise (Guo et al., 2023; Nijkamp et al., 2023; Ni et al., 2023). However, designing effective LLM-ITL solutions demands significant domain knowledge, such as creating optimal LLM utilities and integrated workflows. Automating this process via a novel "Text-to-Solution" framework could significantly enhance the accessibility of the LLM-ITL methodologies.

As shown in Appendix E, under a zero-shot setting, the LLM is capable of: 1) capturing the concept of LLM integration and LLM-ITL without explicit definition, 2) identifying suitable phases of LLM integration, and 3) deriving concrete implementation plans. However, the generated code quality remains inconsistent, and there is a lack of sufficient understanding of in-the-loop techniques, which limits the diversity of solutions and still necessitates human experts to design the high-level framework. Inspired by the success of AutoML in automatically designing machine learning applications (Lindauer et al., 2024), further research is encouraged to explore **Automated In-the-loop (AutoITL)** as a promising "text-to-solution" framework to automate LLM utility selection and workflow construction, streamlining the creation of effective LLM-ITL solutions.

6 Conclusion

This paper introduces a novel paradigm, LLM-in-the-loop (LLM-ITL), offering the first formal definition, motivations, and application scenarios to guide future advancements and exploration. We present a comprehensive taxonomy of methodologies for integrating LLMs into machine learning development, highlighting underexplored techniques and underutilized domains. As the research community refines LLM-ITL methodologies, this paper establishes a foundation for leveraging the full potential of LLMs, not only in direct problem-solving but through their combined efforts with machine learning models to tackle complex problems.

Limitations

Although this paper provides an extensive overview of the LLM-in-the-loop paradigm and organizes methodologies into three well-defined categories, it is important to acknowledge certain limitations that future research could address further.

First, this paper primarily focuses on text clustering as a significant case study where LLM-in-the-loop methods have been effectively applied to enhance performance and interpretability. However, despite an additional case study on time series forecasting provided in Appendix C, there remains limited empirical evidence to demonstrate the superiority of LLM-in-the-loop solutions. Current research, such as LLM-based data augmentation and reward generation, typically emphasizes component-specific performance rather than holistic task applications. This gap underscores the necessity for future studies to develop LLM-in-the-loop solutions tailored to diverse applications across various domains, enabling a deeper investigation into task-dependent performance and further revealing the strengths and weaknesses of the LLM-in-the-loop paradigm, which is only partly discussed in this paper.

Second, this paper predominantly concentrates on the LLM-in-the-loop machine learning application, similar to the conventional human-in-the-loop setting, where LLMs replace the human role in assisting machine learning models. However, given the advanced capabilities of LLMs, exploring the concept of "LLM-in-the-loop LLM" presents a promising avenue for research and application development, which this paper overlooks. In this scenario, smaller LLMs could address sub-tasks they are particularly suited for or trained on, while a larger LLM manages the overarching tasks. This approach could enhance LLM-native solutions by incorporating the design philosophy of LLM-in-the-loop. Moreover, integrating human involvement in the LLM-in-the-loop framework opens new opportunities to study the dynamics between LLMs, machine learning models, and human input, a topic not discussed in this paper but holds potential in expanding the scope of future research.

References

Eshaan Agarwal, Joykirat Singh, Vivek Dani, Raghav Magazine, Tanuja Ganu, and Akshay Nambi. 2024. Promptwizard: Task-aware prompt optimization framework. *Preprint*, arXiv:2405.18369.

Nikhil Agarwal, Alex Moehring, Pranav Rajpurkar, and Tobias Salz. 2023. Combining human expertise with artificial intelligence: Experimental evidence from radiology. Working Paper 31422, National Bureau of Economic Research.

Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. 2024. Large language models for mathematical reasoning: Progresses and challenges. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 225–237.

Wenbin An, Wenkai Shi, Feng Tian, Haonan Lin, QianYing Wang, Yaqiang Wu, Mingxiang Cai, Luyan Wang, Yan Chen, Haiping Zhu, and Ping Chen. 2024. Generalized category discovery with large language models in the loop. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 8653–8665, Bangkok, Thailand. Association for Computational Linguistics.

Ines Arous, Ljiljana Dolamic, Jie Yang, Akansha Bhardwaj, Giuseppe Cuccu, and Philippe Cudré-Mauroux. 2021. Marta: Leveraging human rationales for explainable text classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(7):5868–5876.

Yang Ba, Michelle V Mancenido, and Rong Pan. 2024. Fill in the gaps: Model calibration and generalization with synthetic data. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17211–17225, Miami, Florida, USA. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Vojtěch Balek, Lukáš Sỳkora, Vilém Sklenák, and Tomáš Kliegr. 2024. Llm-based feature generation from text for interpretable machine learning. *arXiv preprint arXiv:2409.07132*.

Houda Nait El Barj and Théophile Sautory. 2024. Reinforcement learning from llm feedback to counteract goal misgeneralization. *arXiv preprint arXiv:2401.07181*.

Max Bartolo, Alastair Roberts, Johannes Welbl, Sebastian Riedel, and Pontus Stenetorp. 2020. Beat the AI: Investigating adversarial human annotation for reading comprehension. *Transactions of the Association for Computational Linguistics*, 8.

Yoshua Bengio, Aaron Courville, and Pascal Vincent. 2013. Representation learning: A review and new perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(8):1798–1828.

Amrita Bhattacharjee, Raha Moraffah, Joshua Garland, and Huan Liu. 2024. Towards llm-guided causal explainability for black-box text classifiers. In *AAAI 2024 Workshop on Responsible Language Models, Vancouver, BC, Canada*.

Shahzad Sarwar Bhatti, Xiaofeng Gao, and Guihai Chen. 2020. General framework, opportunities and challenges for crowdsourcing techniques: A comprehensive survey. *Journal of Systems and Software*, 167:110611.

Zhen Bi, Ningyu Zhang, Yida Xue, Yixin Ou, Daxiong Ji, Guozhou Zheng, and Huajun Chen. 2024. OceanGPT: A large language model for ocean science tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3357–3372, Bangkok, Thailand. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Yuji Cao, Huan Zhao, Yuheng Cheng, Ting Shu, Yue Chen, Guolong Liu, Gaoqi Liang, Junhua Zhao, Jinyue Yan, and Yun Li. 2024. Survey on large language model-enhanced reinforcement learning: Concept, taxonomy, and methods. *IEEE Transactions on Neural Networks and Learning Systems*.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. *arXiv preprint arXiv:2003.04807*.

Jan Cegin, Jakub Simko, and Peter Brusilovsky. 2023. ChatGPT to replace crowdsourcing of paraphrases for intent classification: Higher diversity and comparable model robustness. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1889–1905, Singapore. Association for Computational Linguistics.

Shuyu Chang, Rui Wang, Peng Ren, and Haiping Huang. 2024. Enhanced short text modeling: Leveraging large language models for topic refinement. *arXiv preprint arXiv:2403.17706*.

Banghao Chen, Zhaofeng Zhang, Nicolas Langrené, and Shengxin Zhu. 2024a. Unleashing the potential of prompt engineering in large language models: a comprehensive review. *Preprint*, arXiv:2310.14735.

CHEN CHEN, Yuchen Hu, Chao-Han Huck Yang, Sabato Marco Siniscalchi, Pin-Yu Chen, and Eng-Siong Chng. 2023. Hyporadise: An open baseline for generative speech recognition with large language models. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Lekai Chen, Ashutosh Trivedi, and Alvaro Velasquez. 2024b. Llms as probabilistic minimally adequate teachers for dfa learning. *arXiv preprint arXiv:2408.02999*.

Lingjiao Chen, Matei Zaharia, and James Zou. 2023. Frugalgpt: How to use large language models while reducing cost and improving performance. *Preprint*, arXiv:2305.05176.

Ruirui Chen, Chengwei Qin, Weifeng Jiang, and Dongkyu Choi. 2024c. Is a large language model a good annotator for event extraction? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17772–17780.

Xiang Chen, Chaoyang Gao, Chunyang Chen, Guangbei Zhang, and Yong Liu. 2025a. An empirical study on challenges for llm application developers. *ACM Transactions on Software Engineering and Methodology*.

Yuan Chen, Zi-han Ding, Ziqin Wang, Yan Wang, Lijun Zhang, and Si Liu. 2025b. Asynchronous large language model enhanced planner for autonomous driving. In *European Conference on Computer Vision*, pages 22–38. Springer.

Yulong Chen, Yang Liu, Jianhao Yan, Xuefeng Bai, Ming Zhong, Yinghao Yang, Ziyi Yang, Chenguang Zhu, and Yue Zhang. 2024d. See what LLMs cannot answer: A self-challenge framework for uncovering LLM weaknesses. In *First Conference on Language Modeling*.

Juhwan Choi, JungMin Yun, Kyohoon Jin, and YoungBin Kim. 2024. Multi-news+: Cost-efficient dataset cleansing via LLM-based data annotation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15–29, Miami, Florida, USA. Association for Computational Linguistics.

Anni Coden, Marina Danilevsky, Daniel Gruhl, Linda Kato, and Meena Nagarajan. 2017. A method to accelerate human in the loop clustering. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 237–245. SIAM.

Shih-Chieh Dai, Aiping Xiong, and Lun-Wei Ku. 2023. LLM-in-the-loop: Leveraging large language model for thematic analysis. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9993–10001, Singapore. Association for Computational Linguistics.

Michelangelo Diligenti, Soumali Roychowdhury, and Marco Gori. 2017. Integrating prior knowledge into deep learning. In *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 920–923.

Bowen Ding, Qingkai Min, Shengkun Ma, Yingjie Li, Linyi Yang, and Yue Zhang. 2024. A rationale-centric counterfactual data augmentation method for cross-document event coreference resolution. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1112–1140, Mexico City, Mexico. Association for Computational Linguistics.

Yuqing Du, Olivia Watkins, Zihan Wang, Cédric Colas, Trevor Darrell, Pieter Abbeel, Abhishek Gupta, and Jacob Andreas. 2023. Guiding pretraining in reinforcement learning with large language models. In *International Conference on Machine Learning*, pages 8657–8677. PMLR.

Zheng Fang, Lama Alqazlan, Du Liu, Yulan He, and Rob Procter. 2023. A user-centered, interactive, human-in-the-loop topic modelling system. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 505–522, Dubrovnik, Croatia. Association for Computational Linguistics.

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A Smith, Wei-Chiu Ma, and Ranjay Krishna. 2025. Blink: Multimodal large language models can see but not perceive. In *European Conference on Computer Vision*, pages 148–166. Springer.

Wensheng Gan, Shicheng Wan, and Philip S. Yu. 2023. Model-as-a-service (maas): A survey. In *2023 IEEE International Conference on Big Data (BigData)*, pages 4636–4645.

Jiahui Geng, Fengyu Cai, Yuxia Wang, Heinz Koeppl, Preslav Nakov, and Iryna Gurevych. 2024. A survey of confidence estimation and calibration in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6577–6595, Mexico City, Mexico. Association for Computational Linguistics.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30):e2305016120.

Madeleine Grunde-McLaughlin, Michelle S. Lam, Ranjay Krishna, Daniel S. Weld, and Jeffrey Heer. 2024. Designing llm chains by adapting techniques from crowdsourcing workflows. *Preprint*, arXiv:2312.11681.

Yang Gu, Hengyu You, Jian Cao, and Muran Yu. 2024. Large language models for constructing and optimizing machine learning workflows: A survey. *arXiv preprint arXiv:2411.10478*.

Daya Guo, Canwen Xu, Nan Duan, Jian Yin, and Julian McAuley. 2023. Longcoder: a long-range pre-trained language model for code completion. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. 2024. Large language model based multi-agents: A survey of progress and challenges. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pages 8048–8057. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Luheng He, Julian Michael, Mike Lewis, and Luke Zettlemoyer. 2016. Human-in-the-loop parsing. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2337–2342, Austin, Texas. Association for Computational Linguistics.

Xingwei He, Zhenghao Lin, Yeyun Gong, A-Long Jin, Hang Zhang, Chen Lin, Jian Jiao, Siu Ming Yiu, Nan Duan, and Weizhu Chen. 2024. AnnoLLM: Making large language models to be better crowdsourced annotators. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*, pages 165–190, Mexico City, Mexico. Association for Computational Linguistics.

Danula Hettiachchi, Mark Sanderson, Jorge Goncalves, Simo Hosio, Gabriella Kazai, Matthew Lease, Mike Schaekermann, and Emine Yilmaz. 2021. Investigating and mitigating biases in crowdsourced data. In *Companion Publication of the 2021 Conference on Computer Supported Cooperative Work and Social Computing*, CSCW '21 Companion, page 331–334, New York, NY, USA. Association for Computing Machinery.

Noah Hollmann, Samuel Müller, and Frank Hutter. 2024. Large language models for automated data science: Introducing caafe for context-aware automated feature engineering. *Advances in Neural Information Processing Systems*, 36.

Andreas Holzinger. 2016. Interactive machine learning for health informatics: when do we need the human-in-the-loop? *Brain informatics*, 3(2):119–131.

Mengze Hong, Yuanfeng Song, Di Jiang, Wailing Ng, Yanjie Sun, and Chen Jason Zhang. 2024a. Dial-in llm: Human-aligned dialogue intent clustering with llm-in-the-loop. *arXiv preprint arXiv:2412.09049*.

Mengze Hong, Yuanfeng Song, Di Jiang, Lu Wang, Zichang Guo, and Chen Jason Zhang. 2024b. Expanding chatbot knowledge in customer service: Context-aware similar question generation using large language models. *Preprint*, arXiv:2410.12444.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. 2024c. MetaGPT: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Yuchen Hu, Chen Chen, Chao-Han Yang, Ruizhe Li, Dong Zhang, Zhehuai Chen, and EngSiong Chng. 2024. GenTranslate: Large language models are generative multilingual speech and machine translators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 74–90, Bangkok, Thailand. Association for Computational Linguistics.

Xiang Huang, Sitao Cheng, Yiheng Shu, Yuheng Bao, and Yuzhong Qu. 2023. Question decomposition tree for answering complex questions over knowledge bases. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press.

Tunazzina Islam and Dan Goldwasser. 2024. Uncovering latent arguments in social media messaging by employing llms-in-the-loop strategy. *Preprint*, arXiv:2404.10259.

Jiuchuan Jiang, Bo An, Yichuan Jiang, Donghui Lin, Zhan Bu, Jie Cao, and Zhifeng Hao. 2018. Understanding crowdsourcing systems from a multiagent perspective and approach. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 13(2):1–32.

Yushan Jiang, Wenchao Yu, Geon Lee, Dongjin Song, Kijung Shin, Wei Cheng, Yanchi Liu, and Haifeng Chen. 2025. Explainable multi-modal time series prediction with llm-in-the-loop. *Preprint*, arXiv:2503.01013.

Thommen George Karimpanal, Laknath Buddhika Semage, Santu Rana, Hung Le, Truyen Tran, Sunil Gupta, and Svetha Venkatesh. 2023. Lagrseq: Language-guided reinforcement learning with sample-efficient querying. *arXiv preprint arXiv:2308.13542*.

Bunyamin Keles, Murat Gunay, and Serdar I. Caglar. 2024. Llms-in-the-loop part-1: Expert small ai models for bio-medical text translation. *Preprint*, arXiv:2407.12126.

Nataliia Kholodna, Sahib Julka, Mohammad Khodadadi, Muhammed Nurullah Gumus, and Michael Granitzer. 2024. Llms in the loop: Leveraging large language model annotations for active learning in low-resource languages. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 397–412. Springer.

Tushar Khot, Harsh Trivedi, Matthew Finlayson, Yao Fu, Kyle Richardson, Peter Clark, and Ashish Sabharwal. 2023. Decomposed prompting: A modular approach for solving complex tasks. In *The Eleventh International Conference on Learning Representations*.

Varun Kumar, Alison Smith-Renner, Leah Findlater, Kevin Seppi, and Jordan Boyd-Graber. 2019. Why didn't you listen to me? comparing user control of human-in-the-loop topic models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6323–6330, Florence, Italy. Association for Computational Linguistics.

Taja Kuzman, Igor Mozetič, and Nikola Ljubešić. 2023. Chatgpt: Beginning of an end of manual linguistic data annotation? use case of automatic genre identification. *Preprint*, arXiv:2303.03953.

Minae Kwon and Sang Michael. 2023. Reward design with language models. In *International Conference on Learning Representations (ICLR)*.

Alireza Rashidi Laleh and Majid Nili Ahmadabadi. 2024. A survey on enhancing reinforcement learning in complex environments: Insights from human and llm feedback. *arXiv preprint arXiv:2411.13410*.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2019. An evaluation dataset for intent classification and out-of-scope prediction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1311–1316, Hong Kong, China. Association for Computational Linguistics.

Krzysztof Lebioda, Viktor Vorobev, Nenad Petrovic, Fengjunjie Pan, Vahid Zolfaghari, and Alois Knoll. 2024. Towards single-system illusion in software-defined vehicles–automated, ai-powered workflow. *arXiv preprint arXiv:2403.14460*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Chengsi Liang, Hongyang Du, Yao Sun, Dusit Niyato, Jiawen Kang, Dezong Zhao, and Muhammad Ali Imran. 2024. Generative ai-driven semantic communication networks: Architecture, technologies and applications. *IEEE Transactions on Cognitive Communications and Networking*, pages 1–1.

Jianzhe Lin, Maurice Diesendruck, Liang Du, and Robin Abraham. 2024. Batchprompt: Accomplish more with less. In *The Twelfth International Conference on Learning Representations*.

Marius Lindauer, Florian Karl, Anne Klier, Julia Moosbauer, Alexander Tornede, Andreas C Mueller, Frank Hutter, Matthias Feurer, and Bernd Bischl. 2024. Position: A call to action for a human-centered autoML paradigm. In *Forty-first International Conference on Machine Learning*.

Wenhao Liu, Xiaohua Wang, Muling Wu, Tianlong Li, Changze Lv, Zixuan Ling, Zhu JianHao, Cenyuan Zhang, Xiaoqing Zheng, and Xuanjing Huang. 2024a. Aligning large language models with human preferences through representation engineering. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10619–10638, Bangkok, Thailand. Association for Computational Linguistics.

Xingkun Liu, Arash Eshghi, Pawel Swietojanski, and Verena Rieser. 2021. Benchmarking natural language understanding services for building conversational agents. In *Increasing naturalness and flexibility in spoken dialogue interaction: 10th international workshop on spoken dialogue systems*, pages 165–183. Springer.

Yanchen Liu, Srishti Gautam, Jiaqi Ma, and Himabindu Lakkaraju. 2024b. Confronting LLMs with traditional ML: Rethinking the fairness of large language models in tabular classifications. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3603–3620, Mexico City, Mexico. Association for Computational Linguistics.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2511–2522.

Morteza Moradi, Mohammad Moradi, and Dario Calogero Guastella. 2023. Experience sharing and human-in-the-loop optimization for federated robot navigation recommendation. In *ICIAP Workshops (2)*, pages 179–188.

Eduardo Mosqueira-Rey, Elena Hernández-Pereira, David Alonso-Ríos, José Bobes-Bascarán, and Ángel Fernández-Leal. 2022. Human-in-the-loop machine learning: a state of the art. *Artif. Intell. Rev.*, 56(4):3005–3054.

Rajiv Movva, Sidhika Balachandar, Kenny Peng, Gabriel Agostini, Nikhil Garg, and Emma Pierson. 2024. Topics, authors, and institutions in large language model research: Trends from 17K arXiv papers. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1223–1243, Mexico City, Mexico. Association for Computational Linguistics.

Ansong Ni, Srini Iyer, Dragomir Radev, Ves Stoyanov, Wen-tau Yih, Sida I. Wang, and Xi Victoria Lin. 2023. Lever: learning to verify language-to-code generation with execution. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Erik Nijkamp, Bo Pang, Hiroaki Hayashi, Lifu Tu, Huan Wang, Yingbo Zhou, Silvio Savarese, and Caiming Xiong. 2023. Codegen: An open large language model for code with multi-turn program synthesis. In *The Eleventh International Conference on Learning Representations*.

Matthew J Page, Joanne E McKenzie, Patrick M Bossuyt, Isabelle Boutron, Tammy C Hoffmann, Cynthia D Mulrow, Larissa Shamseer, Jennifer M Tetzlaff, Elie A Akl, Sue E Brennan, Roger Chou, Julie Glanville, Jeremy M Grimshaw, Asbjørn Hróbjartsson, Manoj M Lalu, Tianjing Li, Elizabeth W Loder, Evan Mayo-Wilson, Steve McDonald, Luke A McGuinness, Lesley A Stewart, James Thomas, Andrea C Tricco, Vivian A Welch, Penny Whiting, and David Moher. 2021. The prisma 2020 statement: an updated guideline for reporting systematic reviews. *BMJ*, 372.

Amie Paige, Adil Soubki, John Murzaku, Owen Rambow, and Susan E. Brennan. 2024. Training LLMs to recognize hedges in dialogues about roadrunner cartoons. In *Proceedings of the 25th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 204–215, Kyoto, Japan. Association for Computational Linguistics.

Rahul Pandey, Hemant Purohit, Carlos Castillo, and Valerie L. Shalin. 2022. Modeling and mitigating human annotation errors to design efficient stream processing systems with human-in-the-loop machine learning. *International Journal of Human-Computer Studies*, 160:102772.

Andrew Pang, Hyeju Jang, and Shiaofen Fang. 2024. Generating descriptive explanations of machine learning models using llm. In *2024 IEEE International Conference on Big Data (BigData)*, pages 5369–5374. IEEE.

Anup Pattnaik, Cijo George, Rishabh Kumar Tripathi, Sasanka Vutla, and Jithendra Vepa. 2024. Improving hierarchical text clustering with LLM-guided multi-view cluster representation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural*

Language Processing: Industry Track, pages 719–727, Miami, Florida, US. Association for Computational Linguistics.

Bharat Prakash, Tim Oates, and Tinoosh Mohsenin. 2023. LLM augmented hierarchical agents. In *2nd Workshop on Language and Robot Learning: Language as Grounding*.

Xiao Pu, Mingqi Gao, and Xiaojun Wan. 2023. Summarization is (almost) dead. *Preprint*, arXiv:2309.09558.

Sylvestre-Alvise Rebuffi, Sven Gowal, Dan Andrei Calian, Florian Stimberg, Olivia Wiles, and Timothy Mann. 2021. Data augmentation can improve robustness. In *Advances in Neural Information Processing Systems*.

Hamidreza Rouzegar and Masoud Makrehchi. 2024. Enhancing text classification through llm-driven active learning and human annotation. In *Proceedings of The 18th Linguistic Annotation Workshop (LAW-XVIII)*, pages 98–111.

Philipp Schoenegger and Peter S. Park. 2023. Large language model prediction capabilities: Evidence from a real-world forecasting tournament. *Preprint*, arXiv:2310.13014.

Ryan Smith, Jason A Fries, Braden Hancock, and Stephen H Bach. 2024. Language models in the loop: Incorporating prompting into weak supervision. *ACM/JMS Journal of Data Science*, 1(2):1–30.

Yuan-Feng Song, Yuan-Qin He, Xue-Fang Zhao, Han-Lin Gu, Di Jiang, Hai-Jun Yang, and Li-Xin Fan. 2024. A communication theory perspective on prompting engineering methods for large language models. *Journal of Computer Science and Technology*, 39(4):984–1004.

Andrea Sottana, Bin Liang, Kai Zou, and Zheng Yuan. 2023. Evaluation metrics in the era of GPT-4: Reliably evaluating large language models on sequence to sequence tasks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 8776–8788, Singapore. Association for Computational Linguistics.

Benjamin Adin Spiegel, Ziyi Yang, William Jurayj, Ben Bachmann, Stefanie Tellex, and George Konidaris. 2024. Informing reinforcement learning agents by grounding language to markov decision processes. In *Workshop on Training Agents with Foundation Models at RLC 2024*.

Makesh Narsimhan Sreedhar, Traian Rebedea, and Christopher Parisien. 2024. Unsupervised extraction of dialogue policies from conversations. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 19029–19045.

Akash Srivastava, James Zou, and Charles Sutton. 2016. Clustering with a reject option: Interactive clustering as bayesian prior elicitation. In *33rd International Conference on Machine Learning: ICML 2016*, pages 16–20.

Saurabh Srivastava, Chengyue Huang, Weiguo Fan, and Ziyu Yao. 2024. Instances need more care: Rewriting prompts for instances with llms in the loop yields better zero-shot performance. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 6211–6232.

Kv Aditya Srivatsa and Ekaterina Kochmar. 2024. What makes math word problems challenging for LLMs? In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1138–1148, Mexico City, Mexico. Association for Computational Linguistics.

Arjun V Sudhakar, Prasanna Parthasarathi, Janarthanan Rajendran, and Sarath Chandar. 2024. Language model-in-the-loop: Data optimal approach to recommend actions in text games. In *ICML 2024 Workshop on Foundation Models in the Wild*.

Tianxiang Sun, Yunfan Shao, Hong Qian, Xuanjing Huang, and Xipeng Qiu. 2022. Black-box tuning for language-model-as-a-service. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 20841–20855. PMLR.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA. MIT Press.

Mingtian Tan, Mike A Merrill, Vinayak Gupta, Tim Althoff, and Thomas Hartvigsen. 2024a. Are language models actually useful for time series forecasting? In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Zhen Tan, Dawei Li, Song Wang, Alimohammad Beigi, Bohan Jiang, Amrita Bhattacharjee, Mansooreh Karami, Jundong Li, Lu Cheng, and Huan Liu. 2024b. Large language models for data annotation and synthesis: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 930–957, Miami, Florida, USA. Association for Computational Linguistics.

Yihong Tang, Zhaokai Wang, Ao Qu, Yihao Yan, Zhaofeng Wu, Dingyi Zhuang, Jushi Kai, Kebing Hou, Xiaotong Guo, Jinhua Zhao, Zhan Zhao, and Wei Ma. 2024. ItiNera: Integrating spatial optimization with large language models for open-domain urban itinerary planning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1413–1432, Miami, Florida, US. Association for Computational Linguistics.

Yongxin Tong, Zimu Zhou, Yuxiang Zeng, Lei Chen, and Cyrus Shahabi. 2019. Spatial crowdsourcing: a survey. *The VLDB Journal*, 29(1):217–250.

Petter Törnberg. 2023. Chatgpt-4 outperforms experts and crowd workers in annotating political twitter messages with zero-shot learning. *arXiv preprint arXiv:2304.06588*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Veniamin Veselovsky, Manoel Horta Ribeiro, and Robert West. 2023. Artificial artificial artificial intelligence: Crowd workers widely use large language models for text production tasks. *Preprint*, arXiv:2306.07899.

Vijay Viswanathan, Kiril Gashteovski, Kiril Gashteovski, Carolin Lawrence, Tongshuang Wu, and Graham Neubig. 2024. Large language models enable few-shot clustering. *Transactions of the Association for Computational Linguistics*, 12:321–333.

Yufei Wang, Wanjun Zhong, Liangyou Li, Fei Mi, Xingshan Zeng, Wenyong Huang, Lifeng Shang, Xin Jiang, and Qun Liu. 2023. Aligning large language models with human: A survey. *Preprint*, arXiv:2307.12966.

Victor Junqiu Wei, Weicheng Wang, Di Jiang, Yuanfeng Song, and Lu Wang. 2024. Asr-ec benchmark: Evaluating large language models on chinese asr error correction. *Preprint*, arXiv:2412.03075.

Xingjiao Wu, Luwei Xiao, Yixuan Sun, Junhang Zhang, Tianlong Ma, and Liang He. 2022. A survey of human-in-the-loop for machine learning. *Future Generation Computer Systems*, 135:364–381.

Yuwei Wu, Yuezhan Tao, Peihan Li, Guangyao Shi, Gaurav S Sukhatmem, Vijay Kumar, and Lifeng Zhou. 2024. Hierarchical llms in-the-loop optimization for real-time multi-robot target tracking under unknown hazards. *arXiv preprint arXiv:2409.12274*.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, YIFEI LI, Jie Fu, Junxian He, and Bryan Hooi. 2024. Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation in LLMs. In *The Twelfth International Conference on Learning Representations*.

Canwen Xu, Yichong Xu, Shuohang Wang, Yang Liu, Chenguang Zhu, and Julian McAuley. 2024. Small models are valuable plug-ins for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 283–294, Bangkok, Thailand. Association for Computational Linguistics.

Cheng Yang, Puli Chen, and Qingbao Huang. 2024a. Can chatgpt's performance be improved on verb metaphor detection tasks? bootstrapping and combining tacit knowledge. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1016–1027.

Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019. Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol.*, 10(2).

Shiping Yang, Renliang Sun, and Xiaojun Wan. 2023. A new benchmark and reverse validation method for passage-level hallucination detection. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Xiaohao Yang, He Zhao, Weijie Xu, Yuanyuan Qi, Jueqing Lu, Dinh Phung, and Lan Du. 2024b. Neural topic modeling with large language models in the loop. *arXiv preprint arXiv:2411.08534*.

Yue Yu, Yuchen Zhuang, Jieyu Zhang, Yu Meng, Alexander J Ratner, Ranjay Krishna, Jiaming Shen, and Chao Zhang. 2024. Large language model as attributed training data generator: A tale of diversity and bias. *Advances in Neural Information Processing Systems*, 36.

Quan Yuan, Mehran Kazemi, Xin Xu, Isaac Noble, Vaiva Imbrasaite, and Deepak Ramachandran. 2025. Tasklama: probing the complex task understanding of language models. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press.

Zhiyuan Zeng, Jiatong Yu, Tianyu Gao, Yu Meng, Tanya Goyal, and Danqi Chen. 2024. Evaluating large language models at evaluating instruction following. In *The Twelfth International Conference on Learning Representations*.

Chen Jason Zhang, Lei Chen, H. V. Jagadish, and Chen Caleb Cao. 2013. Reducing uncertainty of schema matching via crowdsourcing. *Proc. VLDB Endow.*, 6(9):757–768.

Chen Jason Zhang, Yunrui Liu, Pengcheng Zeng, Ting Wu, Lei Chen, Pan Hui, and Fei Hao. 2024a. Similarity-driven and task-driven models for diversity of opinion in crowdsourcing markets. *The VLDB Journal*, pages 1–22.

Chen Jason Zhang, Yongxin Tong, and Lei Chen. 2014. Where to: crowd-aided path selection. *Proc. VLDB Endow.*, 7(14):2005–2016.

Han Zhang, Akram Bin Sediq, Ali Afana, and Melike Erol-Kantarci. 2024b. Generative ai-in-the-loop: Integrating llms and gpts into the next generation networks. *Preprint*, arXiv:2406.04276.

Ruoyu Zhang, Yanzeng Li, Yongliang Ma, Ming Zhou, and Lei Zou. 2023a. LLMaAA: Making large language models as active annotators. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 13088–13103, Singapore. Association for Computational Linguistics.

Xinhao Zhang, Jinghan Zhang, Banafsheh Rekabdar, Yuanchun Zhou, Pengfei Wang, and Kunpeng Liu. 2024c. Dynamic and adaptive feature generation with llm. *arXiv preprint arXiv:2406.03505*.

Yuwei Zhang, Zihan Wang, and Jingbo Shang. 2023b. ClusterLLM: Large language models as a guide for text clustering. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13903–13920, Singapore. Association for Computational Linguistics.

Huaqin Zhao, Zhengliang Liu, Zihao Wu, Yiwei Li, Tianze Yang, Peng Shu, Shaochen Xu, Haixing Dai, Lin Zhao, Hanqi Jiang, Yi Pan, Junhao Chen, Yifan Zhou, Gengchen Mai, Ninghao Liu, and Tianming Liu. 2024. Revolutionizing finance with llms: An overview of applications and insights. *Preprint*, arXiv:2401.11641.

Mengjie Zhao, Fei Mi, Yasheng Wang, Minglei Li, Xin Jiang, Qun Liu, and Hinrich Schuetze. 2022. LMTurk: Few-shot learners as crowdsourcing workers in a language-model-as-a-service framework. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 675–692, Seattle, United States. Association for Computational Linguistics.

Zhiqiang Zhong, Kuangyu Zhou, and Davide Mottin. 2024. Harnessing large language models as post-hoc correctors. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14559–14574, Bangkok, Thailand. Association for Computational Linguistics.

Tianyuan Zou, Yang Liu, Peng Li, Jianqing Zhang, Jingjing Liu, and Ya-Qin Zhang. 2024. Fusegen: Plm fusion for data-generation based zero-shot learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 2172–2190, Miami, Florida, USA. Association for Computational Linguistics.

Alexandra Zytek, Sara Pidò, and Kalyan Veeramachaneni. 2024. Llms for xai: Future directions for explaining explanations. *Preprint*, arXiv:2405.06064.

A Survey Methodology and Statistics

This paper primarily bases its supporting claims on systematic literature reviews. With reference to the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) (Page et al., 2021), we outline the paper selection criteria in detail. Our review scope is strictly defined to include (1) the application of LLMs and (2) machine learning research. Research papers are sourced from a variety of channels, including peer-reviewed journals and conference proceedings. Our search strategy combines keyword searches with regulated filtering, focusing on publications from 2020 to 2024 to capture the latest advancements in LLM research. We prioritize papers from highly recognized peer-reviewed venues, specifically targeting top conferences and journals. An overview of LLM-ITL taxonomy is presented in Figure 2, and the survey statistics are presented in Table 2.

In recognition of the growing trend of disseminating emerging research through non-peer-reviewed preprints, we also collected studies submitted to e-print archive platforms such as arXiv. Our analysis of these preprints focused on extracting key insights, including novel definitions, design principles, optimization strategies, and newly proposed problems. Given the preliminary nature of these works, we emphasize their innovative ideas and concepts rather than their quantitative performance, acknowledging their lack of formal verification and the absence of the peer review process.

Venue	Year: Count	Total
Arxiv	2016:1, 2019:1, 2023: 9, 2024: 22, 2025:1	32
ACL	2019:1, 2020:3, 2023:1, 2024:12	17
EMNLP	2016:1, 2019:1, 2023:7, 2024:7	16
ICLR	2015: 1, 2022:1, 2023:3, 2024:4	9
NeurIPS	2014:1, 2017:1, 2020:2, 2012:1, 2023:1, 2024:2	8
ICML	2016:1, 2022:1, 2023:3, 2024:2	7
NAACL	2022:1, 2024:1	7
AAAI	2021:1, 2023:1, 2024:2, 2025:1	5
VLDB	<2020: 3, 2024:1	4
Other	<2022:10, 2022:3, 2023:4, 2024:9, 2025:3	29
Total		**134**

Table 2: Summary of surveyed papers by publication venue and year: "Others" include venues each with fewer than 2 papers included.

B Empirical Study on LLM-Native Text Clustering

Experimental Setup. In this empirical study, the goal is to group n sentences into K clusters by directly prompting LLMs. Three widely adopted benchmark datasets are evaluated, namely

Category	Title	Year	Task
Task-Specific LLM-ITL	Neural Topic Modeling with **Large Language Models in the Loop** (Yang et al., 2024b)	2024	Topic Modeling
	LLMs as Probabilistic Minimally Adequate Teachers for DFA Learning (Chen et al., 2024b) *(...providing a theoretical foundation for automata learning with **LLMs in the loop**.)*	2024	DFA Learning
	Asynchronous Large Language Model Enhanced Planner for Autonomous Driving (Chen et al., 2025b) *(...we introduce AsyncDriver, a new asynchronous **LLM-enhanced closed-loop** framework)*	2024	Autonomous Driving
	Language Models in the Loop: Incorporating Prompting into Weak Supervision (Smith et al., 2024)	2022	Weak Supervision
	Dial-In LLM: Human-Aligned Dialogue Intent Clustering with **LLM-in-the-loop** (Hong et al., 2024a)	2024	Dialogue Clustering
	LLM-in-the-loop: Leveraging Large Language Model for Thematic Analysis (Dai et al., 2023)	2023	Thematic Analysis
Over-generalized ITL	Uncovering Latent Arguments in Social Media Messaging by Employing **LLMs-in-the-Loop Strategy** (Islam and Goldwasser, 2024)	2024	Social Media Analysis
	LLMs in the Loop: Leveraging Large Language Model Annotations for Active Learning in Low-Resource Languages (Kholodna et al., 2024)	2024	Active Learning
	Generalized Category Discovery with Large **Language Models in the Loop** (An et al., 2024)	2024	Category Discovery
	Generative AI-in-the-loop: Integrating LLMs and GPTs into the Next Generation Networks (Zhang et al., 2024b)	2024	Network Integration
Referential Works	Hierarchical **LLMs In-the-loop Optimization** for Real-time Multi-Robot Target Tracking under Unknown Hazards (Wu et al., 2024)	2024	Robotics
	Training LLMs to Recognize Hedges in Spontaneous Narratives (Paige et al., 2024) *(...we used an **LLM-in-the-Loop** approach to improve the gold standard coding)*	2024	Narrative Analysis
	LLMs-in-the-loop Part-1: Expert Small AI Models for Bio-Medical Text Translation (Keles et al., 2024)	2024	Bio-Medical Translation
	A Rationale-centric Counterfactual Data Augmentation Method for Cross-Document Event Coreference Resolution (Ding et al., 2024) *(...we develop a rationale-centric counterfactual data augmentation method with **LLM-in-the-loop**)*	2024	Coreference Resolution
	Towards Single-System Illusion in Software-Defined Vehicles – Automated, AI-Powered Workflow (Lebioda et al., 2024) *(...inclusion of modern generative AI, specifically **Large Language Models (LLMs), in the loop**)*	2024	Workflow Automation
	Instances Need More Care: Rewriting Prompts for Instances with **LLMs in the Loop** Yields Better Zero-Shot Performance (Srivastava et al., 2024)	2023	Zero-Shot Learning

Table 1: Existing works that explicitly mention "LLM-in-the-loop" in their titles or abstracts can be categorized as follows: "task-specific" includes studies that employed LLM-ITL for a single specific task, "over-generalized" encompasses works with a broad scope extending beyond LLMs, and "referential works" comprises publications that simply referenced the term without applying the methodology.

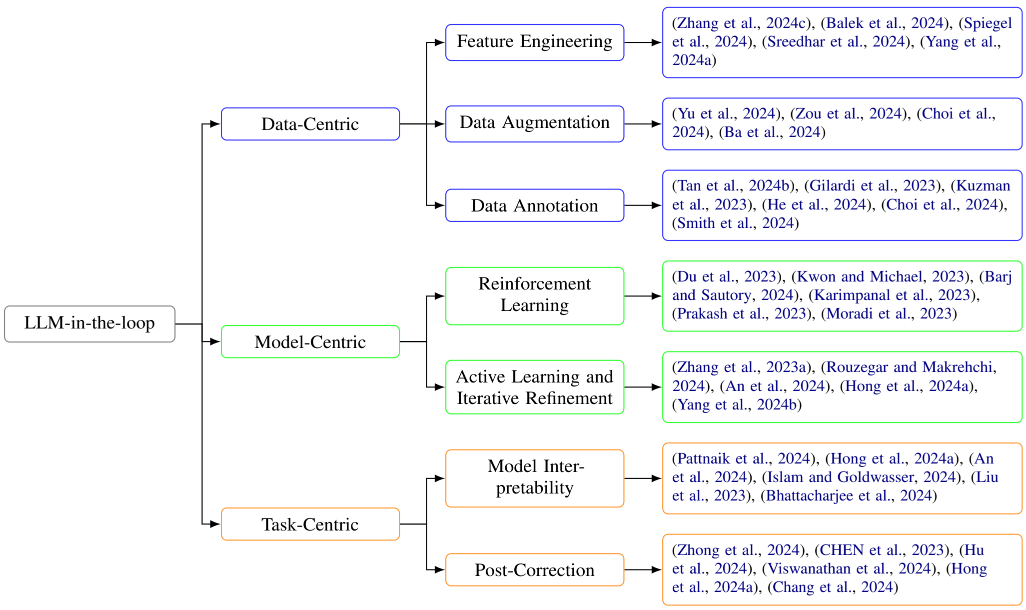

Figure 2: Taxonomy of LLM-in-the-loop Methodologies

CLINC150 (Larson et al., 2019), Banking77 (Casanueva et al., 2020), and HWU64 (Liu et al., 2021). The GPT-4o is employed via the OpenAI API for its broad accessibility, facilitating the reproducibility of results. To mitigate the inherent variability of LLMs while ensuring the significance of the findings, a "resampling" technique, as proposed in (Chen et al., 2024a), is implemented. The model is run 50 times with the same prompt and input data, with the temperature set to 0.5 to balance randomness and consistency in the outputs.

B.1 LLM-native Text Clustering with Prompt Engineering

An exploratory analysis shows that the LLM cannot handle the entire dataset due to input token constraints. Therefore, a subset of the dataset is sampled, consisting of 240 sentences divided into 8 clusters. The objectives of this experiment are twofold: 1) to assess the extent to which LLMs exhibit incapabilities under different prompts, as indicated by discrepancies in the generated solution space and the targeted space defined by the task requirement, and 2) to evaluate the clustering performance of usable LLM-generated cluster assignments. Three hand-crafted prompts were designed: a vanilla instruction prompt with the hint "each label corresponds to a sentence," based on the setup from (Kholodna et al., 2024); a few-shot prompt; and a chain-of-thought prompt. Additionally, the state-of-the-art prompt tuning method, PromptWizard (Agarwal et al., 2024), was used to generate two tailored prompts - one with reasoning steps and one without - specifically tuned to align solution space. Details of the tuning process and the experimented prompts are available on GitHub repository.[2]

Based on the results presented in Table 3, it is evident that the LLM-naive approach underperform in the clustering task, with up to 98% of responses from the standard prompt and 90% from the best-performing prompt failing to align with the targeted label count, making these outputs largely ineffective and a waste of tokens. The adoption of more advanced prompting techniques shows a slight improvement, with prompt tuning without reasoning (i.e., "pw_wo_reasoning") providing the highest number of usable clustering results. While the expected generation of 240 labels remains problematic, the second requirement of clustering into 8 distinct clusters (i.e., adhering to the output space) is well met, with the best-performing prompt successfully generating a list with exactly 8 labels without any error. However, the prompt tuning process incurs substantial costs, both during tuning and at inference time, where the instruction prompt becomes excessively lengthy, posing additional challenges. Additionally, a notable number of samples exceeded the targeted label count, contradicting the "laziness" or "output truncation" behavior of LLMs, which typically outputs less when asked for more.

With the few correct samples obtained, the clustering performance was further evaluated against K-means, which achieved a perfect Normalized Mutual Information (NMI) score of 1. Analyzing the best-performing result from each prompting technique revealed that LLM-based clustering performs reasonably well for this simple task, with the top method achieving performance comparable to K-means clustering. The poorest performance was observed in the reasoning-based prompt, specifically tuned to instruction following, suggesting a potential trade-off between strictly following instructions to ensure usability of results and optimizing for task-solving performance. Despite this, concerns remain about the practicality of using LLMs for text clustering, as the number of usable results for this simple task is still significantly low, which raises doubts about their capability to manage increasing task complexity.

B.2 Input Data and Task Complexity

The next step involves evaluating the impact of input data size and task complexity on the performance of the LLM-natie solution. The input data size varies, ranging from 60 to 600 sentences, with the objective of examining both the emergence of output failure and the variance of the solution space, measured by the difference between the target label count and the predicted label count. The best-performing prompt identified in the previous discussion (i.e., pw_wo_reasoning) is utilized.

From the clustering results in Table 4, we show that a simpler task with $n = 60$ can be easily solved with only one error occurring out of 50 runs. As task complexity rises, output failures increase significantly, appearing in a random pattern when the number of sentences exceeds 120, corresponding to approximately 5200 input tokens plus 4300 tokens from the instruction prompt. Although this is well below the maximum input token limit, the lengthy inputs to the LLM present significant challenges for instruction following during the inference process. By analyzing the variance of the generated clustering results, we observe from Figure 3 that as task complexity grows, the variance also increases. This leads to more outliers, i.e., results that significantly deviate from the majority, resulting in more uninterpretable behavior. **These observations explain why existing research rarely considers LLM-native baselines, mainly due to the infeasibility and unpredictable behaviors of LLMs, motivating future investigation into underlying causes.**

Note that the discussed problem is significantly different from the Batch Prompt (Lin et al., 2024). In Batch Prompt, while the input to the LLM contains n instances and expects n outputs, the tasks being solved are independent and can be easily decomposed into individual prompts. For example, solving 10 math problems in a single prompt or across ten separate prompts. The main goal of Batch Prompt is to reduce the cost of repeated instructions. In contrast, for tasks like clustering and NER, the input must contain n instances, and the solution space is strictly bound by the input data.

[2] The complete code and data are available at https://anonymous.4open.science/r/LLM-in-the-loop-4F42/.

	CLINC150					Banking77					HWU64				
Prompt	L	E	G	OOS	NMI	L	E	G	OOS	NMI	L	E	G	OOS	NMI
vanilla	13	1	36	2	0.976	6	0	44	17	-	12	1	37	20	0.789
cot	19	1	30	1	0.909	13	2	35	15	0.763	7	0	43	12	-
fewshot	15	2	32	5	1	12	2	36	44	**0.858**	19	2	29	16	0.794
pw_wo_reasoning	15	**4**	31	**0**	**1**	6	**3**	41	25	0.760	0	2	48	17	**0.823**
pw_w_reasoning	14	2	34	3	0.896	5	0	45	24	-	6	0	44	27	-

Table 3: Summary of clustering results generated using various prompts, each repeated 50 times, under the clustering setting of $n = 240$ and $k = 8$. The statistics include counts of cases that are Less Than (L), Equal to (E), or Greater Than (G) the target number of clusters n; Out of Set (OOS) denotes misaligned label sets; and Normalized Mutual Information (NMI) measures the clustering quality for results with correct cluster counts and label sets, when applicable. The best results are highlighted in bold.

Task Setting	L	E	G	OOS
$n = 60$	0	49	1	0
$n = 120$	0	1	49	0
$n = 180$	12	8	30	8
$n = 240$	15	4	31	0
$n = 300$	12	1	37	2
$n = 360$	10	3	37	3
$n = 420$	10	1	39	6
$n = 480$	16	0	34	10

Table 4: Summary of clustering results generated with different clustering settings.

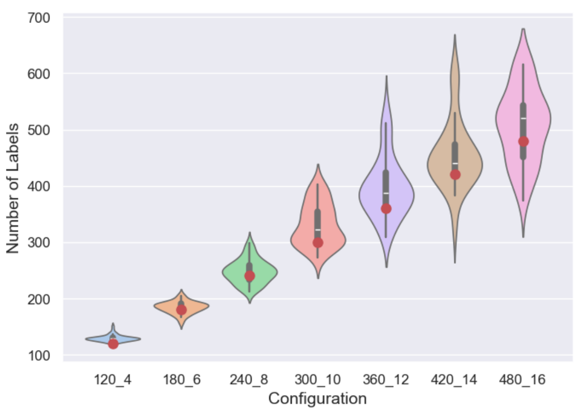

Figure 3: Variance of clustering results from the targeted solution space (i.e., for each specified number of clusters, n).

C Additional Empirical Evidence on the Superiority of LLM-ITL

Besides the widely researched text clustering problem, which benefits from LLM-ITL methodologies due to the involvement of natural language input and semantic comprehension capabilities, we provide additional empirical evidence for the application of LLM-ITL in time series forecasting. Previous studies have demonstrated that using LLMs directly for time series forecasting does not outperform traditional methods (Tan et al., 2024a), highlighting limitations of the LLM-native approach and motivating the use of LLM-ITL. A recent study showed that applying the LLM-ITL framework for time series prediction enhances interpretability and accuracy by leveraging LLMs to reason over multimodal encoder outputs and refine predictions iteratively, exemplifying a typical model-centric and task-centric approach (Jiang et al., 2025). Building on these findings, we explored a data-centric method where LLMs augment input time series data with textual explanations of peaks and troughs within a window size of 15, achieving a 3.29% performance improvement over the baseline that ignores data augmentation.

D Further Discussions

This section analyzes the design philosophy behind each LLM-ITL approach and provides deeper insights for future research. It also explores when to apply LLM-ITL and how to advance existing methods to better leverage the benefits of LLM integration.

Discussion on Data-Centric LLM-in-the-loop The integration of LLMs into data preprocessing offers undeniable advantages in mitigating labor-intensive workflows, and the research question of **how to make LLMs better data annotators** represents a prominent research direction combining LLMs and data science. The development of in-the-loop solutions poses new challenges, requiring both model-specific adaptations (e.g., augmenting data embeddings to fit the particular optimization mechanism) and task-specific customizations (e.g., crafting specific features for intended purposes). This introduces a high level of diversity in how data can be enhanced. While LLMs demonstrate emerging capabilities with in-context learning and can provide domain-specific knowledge often lacking in machine learning, the exploration of applying LLMs in a typical in-the-loop solution to fully leverage these capabilities remains largely underexplored, with limited research combining LLM-driven data preprocessors and machine learning models to solve real-world problems. Additionally, the use of LLMs poses new concerns in assessing data integrity and detecting underlying biases and false information caused by potential hallucinated generations (Tan et al., 2024b), thereby motivating further research into two perspectives: how to design better LLM utility for data augmentation and how to design better in-the-loop solution with more effective LLM integration.

D.1 Discussion on Model-Centric LLM-in-the-loop

Integrating prior knowledge into learning frameworks is crucial for enhancing model performance, especially in data-scarce scenarios where common sense is vital. Task-specific models excel at predefined objectives but struggle to incorporate broad human-like knowledge due to resource-intensive training requirements. Traditionally, human expertise has guided model behavior through active and reinforcement learning. However, LLMs, pre-trained on vast human-generated content, offer a scalable and efficient alternative by providing diverse feedback to refine models. They mitigate data imbalance by enhancing generalization in rare scenarios and address sparse reward issues in reinforcement learning by delivering tailored signals to guide exploration. Additionally, LLMs provide significant time and cost efficiencies over human-driven processes, offering high-quality annotations and context-aware feedback at scale. By interacting with ML models through structured prompts, LLMs distill general knowledge into specialized models, improving sample efficiency and learning trajectories. While LLM-ITL emphasizes the importance of using machine learning model for problem-solving, LLMs facilitate the transfer of general knowledge, enabling models to handle rare, complex, and evolving tasks with greater adaptability and robustness.

D.2 Discussion on Task-Centric LLM-in-the-loop

Traditional inference workflows often underutilize intermediate outputs, such as hypotheses, embeddings, or raw predictions, leaving valuable information unexplored. Rule-based or heuristic post-processing methods lack the adaptability and contextual understanding needed to handle complex or ambiguous scenarios effectively (CHEN et al., 2023). Similarly, traditional interpretability techniques, such as feature importance scores or attribution maps, provide limited insights and fail to produce human-interpretable explanations or actionable feedback (Zytek et al., 2024). LLMs address these limitations by leveraging extensive pre-trained knowledge and few-shot capabilities to dynamically refine outputs, aligning them with task-specific requirements (Viswanathan et al., 2024). Moreover, LLMs can generate high-level abstractions, such as descriptive summaries (Pattnaik et al., 2024) and novel metrics (Liu et al., 2023), surpassing the rigid constraints of conventional approaches and enabling more flexible insights.

D.3 Addressing LLM Limitations in LLM-in-the-loop Solutions

Understanding the limitations of LLMs is crucial for effectively harnessing their strengths while mitigating potential drawbacks. Although LLMs demonstrate remarkable capabilities across a range of tasks, they also encounter issues such as hallucination, bias, and inconsistent instruction adherence, which can affect their reliability. **These limitations form the basis for our proposed paradigm**, where LLMs act as assistants, working alongside machine learning algorithms that primarily focus on task resolution. By deploying LLMs strategically in areas where they are most effective, such as data annotation and text summarization, we can minimize their weaknesses and develop solutions that leverage the strengths of both LLMs and machine learning models.

Like human workers, LLMs can exhibit bias. Human-in-the-loop systems typically use crowdsourcing and the "wisdom of the crowd" to ensure diverse opinions, with majority voting helping to mitigate individual biases. Similarly, we emphasize the concept of LLM crowdsourcing in Section 5, which involves employing multiple LLMs with diverse models and prompts to reduce individual bias. We argue that bias in LLM-in-the-loop systems is significantly lower than in LLM-native solutions since they incorporate machine learning models that operate more deterministically. In contrast, LLM-native approaches are entirely dependent on the limitations of a single LLM. Additionally, LLM-ITL naturally addresses decomposed tasks on a smaller scale, which reduces the impact of LLM bias on final task performance compared to LLM-native solutions that approach the task as a whole.

D.4 Should LLM-in-the-loop replace Human-in-the-loop in the future?

The idea of replacing human participation with LLMs is appealing due to several advantages they offer. LLM-ITL provides broader applicability across various stages, including training, inference, and deployment, making it a more general and encompassing framework. This approach extends existing in-the-loop methodologies by effectively enabling the development of labor-free solutions. When LLMs perform comparably to human workers in tasks such as data annotation, reward generation, and text processing, they present a viable alternative. The transition to LLM-ITL can enhance capabilities at inference time, offer significant cost savings, and potentially protect data privacy. Therefore, when LLMs match or exceed human performance, adopting LLM-ITL could be ideal, offering extended capabilities and efficiencies. For tasks where humans demonstrate significantly superior performance, such as those involving creativity or specific low-resource domain knowledge, utilizing human-in-the-loop would be more beneficial.

D.5 Should LLM-in-the-loop replace LLM-native solutions in the future?

LLM-driven utilities facilitate in-the-loop development by serving as **LLM-native components** tailored for sub-tasks (e.g., evaluation, annotation) rather than solving entire problems directly, referred to as LLM-native solutions. While we acknowledge the limitations of LLM-native applications in many problem-solving scenarios due to the inherent limitations of LLMs, they remain a feasible and predominant choice for less-restricted tasks like code generation and machine translation. LLM-native solutions are particularly effective for tasks involving multiple input sources and modalities (Tang et al., 2024), complex reasoning (Ahn et al., 2024), and heavy reliance on domain knowledge (Bi et al., 2024). These are areas where traditional machine learning algorithms, even with human or LLM in-the-loop, struggle to perform effectively, highlighting the need for LLM-native solutions instead of investing significant efforts in developing suitable machine learning algorithms.

Regarding computational cost, while LLM-ITL may be more expensive than traditional ML methods, it is more controllable than LLM-native solutions, which rely solely on LLMs. For example, using LLMs directly for text clustering requires N inputs and N outputs, while the semi-supervised method only requires the LLM to process sampled data points or derive pairwise constraints, which is significantly cheaper than fully relying on LLM inference.

E Demonstration: Text-to-Solution for Intent Clustering

In this demonstration, we aim to assess the practicality of generating LLM-in-the-loop solutions with LLMs and evaluate whether the LLM can understand the concept of LLM-in-the-loop based on its existing knowledge. Two state-of-the-art models, DeepSeek-R1 and GPT-4o, are tested under zero-shot settings and applied to solve the task of intent clustering (see Figure 4 and 5). The discussion on LLM generated response is presented in Section 5.

You are tasked with writing an intent clustering algorithm where the base machine learning algorithm is K-means. You should implement a LLM-in-the-loop solution, where LLM request should be integrated into the problem-solving workflow. **Zero-shot Prompting**

Okay, let me try to figure out how to approach this problem. The user wants an intent clustering algorithm using K-means as the base, but also integrating an LLM into the workflow. Hmm. So, the main idea here is to use K-means for clustering, but enhance it with an LLM to maybe improve the feature extraction or help with determining the number of clusters, or perhaps interpreting the results. Let me think about the typical steps in K-means clustering.

First, K-means requires feature vectors. Text data isn't directly usable, so usually, we convert text into embeddings. That's where the LLM could come in. Maybe using an LLM to generate embeddings for each text input. Alternatively, maybe the LLM could help in generating labels or refining the clusters after K-means is done. Another thought: determining the optimal K value is a challenge in K-means. Maybe the LLM could assist in evaluating cluster quality or suggesting K based on the data. **Instruction Understanding**

Wait, the problem says "LLM-in-the-loop", which implies that the LLM is part of the iterative process, similar to the "Human-in-the-loop" design. So maybe during the clustering, the LLM is used to process the data in some way. Let me outline possible steps:

1. Preprocessing: Use LLM to generate embeddings for each text input. This would convert text into numerical vectors that K-means can process. Models like OpenAI's text-embedding-ada-002 are commonly used for this.

2. Clustering: Apply K-means on the embeddings to get clusters.

3. Post-processing: Use LLM to interpret cluster centroids or generate labels for each cluster based on the centroid's nearest points or representative samples. **Similar to Hong et. al. (2024)**

Alternatively, maybe the LLM could help in refining the clusters by analyzing the initial clusters and suggesting adjustments. Or perhaps dynamically determining K by having the LLM analyze the data distribution.

Another angle is that K-means might not capture semantic similarities well, so using LLM embeddings could provide better semantic features, leading to more meaningful clusters.

Let me structure this step by step. First, the data needs to be embedded. Then K-means clusters them. Then, to assign meaningful labels or validate the clusters, use the LLM again. For example, take the top terms in each cluster or some representative samples and ask the LLM to generate a label or verify if the cluster makes sense. **Implementation Planning**

Also, maybe the number of clusters K could be determined by the LLM. For instance, generate a few candidate K values using elbow method or silhouette scores, then use the LLM to evaluate which is more appropriate based on the data's intent.

Figure 4: Text-to-solution with DeepSeek-R1 for LLM-in-the-loop Intent Clustering

Implementing an intent clustering algorithm with K-means and incorporating a large language model (LLM)-in-the-loop solution involves several steps. Below is a high-level outline of the process, followed by a more detailed Python implementation.

High-Level Outline

1. **Data Collection and Preprocessing**: Collect and preprocess the data to be clustered. This might involve cleaning text data, tokenization, and embedding.
2. **Initial K-means Clustering**: Perform initial clustering using K-means.
3. **LLM Integration for Refinement**: Use an LLM to refine cluster assignments or provide feedback for improving clusters. **Implementation Planning**
4. **Iterative Improvement**: Iterate the process, incorporating LLM feedback to improve clustering.

Figure 5: Text-to-solution with GPT-4o for LLM-in-the-loop Intent Clustering