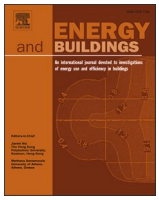
Energy & Buildings 322 (2024) 114691

Contents lists available at ScienceDirect

Energy & Buildings

journal homepage: www.elsevier.com/locate/enb



Exploring automated energy optimization with unstructured building data: A multi-agent based framework leveraging large language models

Tong Xiao, Peng Xu *

School of Mechanical Engineering, Tongji University, Shanghai 201804, PR China

ARTICLE INFO

Keywords:
Automated energy optimization
Unstructured data
Generative Artificial Intelligence
Energy audit
Energy efficiency diagnosis

ABSTRACT

The building sector is a significant energy consumer, making building energy optimization crucial for reducing energy demand. Automating energy optimization tasks eases the workload on engineers and hastens energy savings. More than 85% of building data is unstructured and diverse, concealing energy insights that demand laborious extraction. We propose an LLM-based multi-agent framework to explore automated tasks using these data. The framework includes three stages: building information processing, performance diagnosis, and retrofit recommendation, where LLMs injected with domain expertise act as agents for the roles of planner, researcher and advisor. We develop knowledge databases with retriever tools to inject knowledge and validate through experiments. In case studies, our framework delivered reliable results with only $5.15, effectively handling diverse inputs and tasks across cases. This demonstrates its potential to significantly reduce repetitive human labor and costs. We also discuss the potential of LLM-based multi-agent systems as trustworthy, generalized automated task solvers.

1. Introduction

1.1. Automated tasks in building energy optimization

The building sector accounted for approximately 30 % of global energy consumption in 2022 [1]. Building energy systems play an important role in the decarbonization and electrification of the energy sector, thereby aiding in the fight against global warming. Reducing the energy use of building energy systems while maintaining a favorable indoor environment is an important goal of building management. Energy performance diagnosis aims to identify poor energy performance in a building and determine the causes [2]. It can help advise the building operator on repairs and maintenance to keep the building running in an energy-efficient state. In addition, building energy audits and retrofits aim to improve the poor energy use behavior of buildings and effectively reduce the energy use of building energy systems. Therefore, building energy audits and retrofits are also considered as an efficient method of building energy efficiency [3].

Emerging research directions in the building energy optimization extensively utilize Artificial Intelligence (AI) technologies to automate tasks or discover hidden knowledge from building data. The ultimate goal of AI is to achieve Artificial General Intelligence (AGI) [4], which refers to a system capable of performing human tasks [5]. AI agents have long been considered a key step towards the realization of AGI as they have the potential to carry out a wide range of intelligent activities [6]. In the context of the building energy management, AI agents automate the relevant tasks using different data throughout the entire lifecycle of an energy system. Fig. 1 illustrates the conceptual form of automating building energy audits using an envisioned multi-agent team. The agents will automatically plan tasks, act with tools, cooperate and discuss with each other, and make decisions during the task. In specific engineering applications, the advice given by AI agents should be trustworthy and

Abbreviations: AGI, Artificial General Intelligence; AI, Artificial Intelligence; API, Application Programming Interface; ASHRAE, American Society of Heating, Refrigerating, and Air-Conditioning Engineers; BAS, Building automation system; BIM, Building information modeling; BEMS, Building energy management system; BEM, Building energy modeling; CoT, Chain-of-thought; COP, coefficient of performance; EER, energy efficiency ratio; EPI, Energy performance indicator; EUI, Energy use intensity; GRU, Gated recurrent neural network; HVAC, Heating, Ventilating and Air Conditioning; ICE, Indoor Climate and Energy; IE, Information extraction; LLM, Large Language Model; LM, Language modelling; LSTM, Long short-term memory neural network; NLP, Natural language processing; PLM, Pre-trained Language Model; RAG, Retrieval-Augmented Generation; ReAct, Reasoning and Acting; REPL, Read-Eval-Print Loop; RLHF, Reinforcement learning with human feedback; Seq2seq, Sequence-to-sequence; SFT, Supervised fine-tuning; SOTA, State-of-the-art; Word2vec, Word-to-vector; WTF, water transportation factor.

* Corresponding author.
E-mail address: xupeng@tongji.edu.cn (P. Xu).

https://doi.org/10.1016/j.enbuild.2024.114691
Received 27 June 2024; Received in revised form 2 August 2024; Accepted 17 August 2024
Available online 22 August 2024

needs to be supervised by humans at this stage [7]. The implementation of task automation will significantly reduce the amount of duplicated work required of engineers and allow them to focus on using more energy-efficient and environmentally-friendly systems to maintain a better indoor environment [8]. Current research has paid much attention to automating tasks with tabular building data. However, more than 85 % of the building data is in unstructured format [9] and a large amount of energy use information is hidden in these data. Effective use of the unstructured data will help further improve energy efficiency and thus reduce building energy use. However, since there are no fully recognized data specifications for unstructured data, unstructured data generated in different countries, regions and units will always in different language and structures. At the same time, the personalization of building energy systems between buildings allows for greater variability in unstructured data. Currently, tasks involving unstructured building data still rely on a great deal of human labor and are far from being automated.

1.2. Opportunities for LLM-based agents

Language plays an important role in people's communication and collaboration. The nature of language is thought to originate from prelinguistic intentionality, encompassing perception, belief, desire, memory, intention, etc [10]. Both written and spoken language convey a significant amount of information [11]. Language modelling (LM) is a major approach to enhance the machine's ability to understand and communicate in human language [12]. In 2022 and 2023, OpenAI released two large language models (LLM), ChatGPT [13] and GPT-4 [14], attracting worldwide attention. Subsequently, LLMs, the newest stage in LM, began to have a significant impact on the AI community, leading researchers to regard LLMs as sparks for realizing AGI [15]. Many researchers are beginning to work on developing LLM-based agents that will allow them to replace repetitive human work, such as coding and documentation in software development [16]. As the design and realization of building energy systems are systematic projects requiring collaboration across multiple trades, language also plays an important role in conveying information, rules, and expert experience. For example, the design optimization of a system requires synergies between different simulation software, such as IDA Indoor Climate and Energy (ICE) and TRNSYS [17]. Engineers need to communicate with each other in language to complete simulations. This role of language offers an opportunity to approach the realization of AGI through the development of LLM-based agents.

LLMs are a subset of data-driven deep learning algorithms, and their application can be divided into three categories [18], as shown in Fig. 2. We can directly apply LLM models to static tasks that do not interact with the environment (Fig. 2(a)), like most machine learning and deep learning tasks in the building energy domain (some examples are listed in Table 1). In addition, LLMs can be further developed into LLM-based agents that dynamically interact with the environment (Fig. 2(b)). This will allow the LLM to perform tasks that require interaction with specific environments, such as the control of specific chillers, the processing of data collected from specific energy systems, and so on. Considering that LLMs have demonstrated new emergent capabilities such as memory and reasoning in many tasks, we can further develop cognitive LLM-based agents (Fig. 2(c)). Building energy systems are complex and diverse, and meter-level building energy data are usually heterogeneous across buildings [33]. Data-driven models developed on these data are often difficult to generalize across scenarios [21]. Agents with reasoning skills and domain knowledge memory will be able to plan tasks in different scenarios and transfer the knowledge learned in previous scenarios, thereby coping with more complex and heterogeneous tasks in building energy optimization.

Furthermore, LLMs are highly black-boxed and integrated, with most access provided through the prompting interface or the Application Programming Interface (API). We need to format our tasks in a way that LLMs can follow [34] and explore the performance limits of generic LLMs in our tasks. For LLMs to automate building energy optimization tasks, they must accurately grasp domain concepts, utilize data inference, and generate code. Although generic LLMs possess such

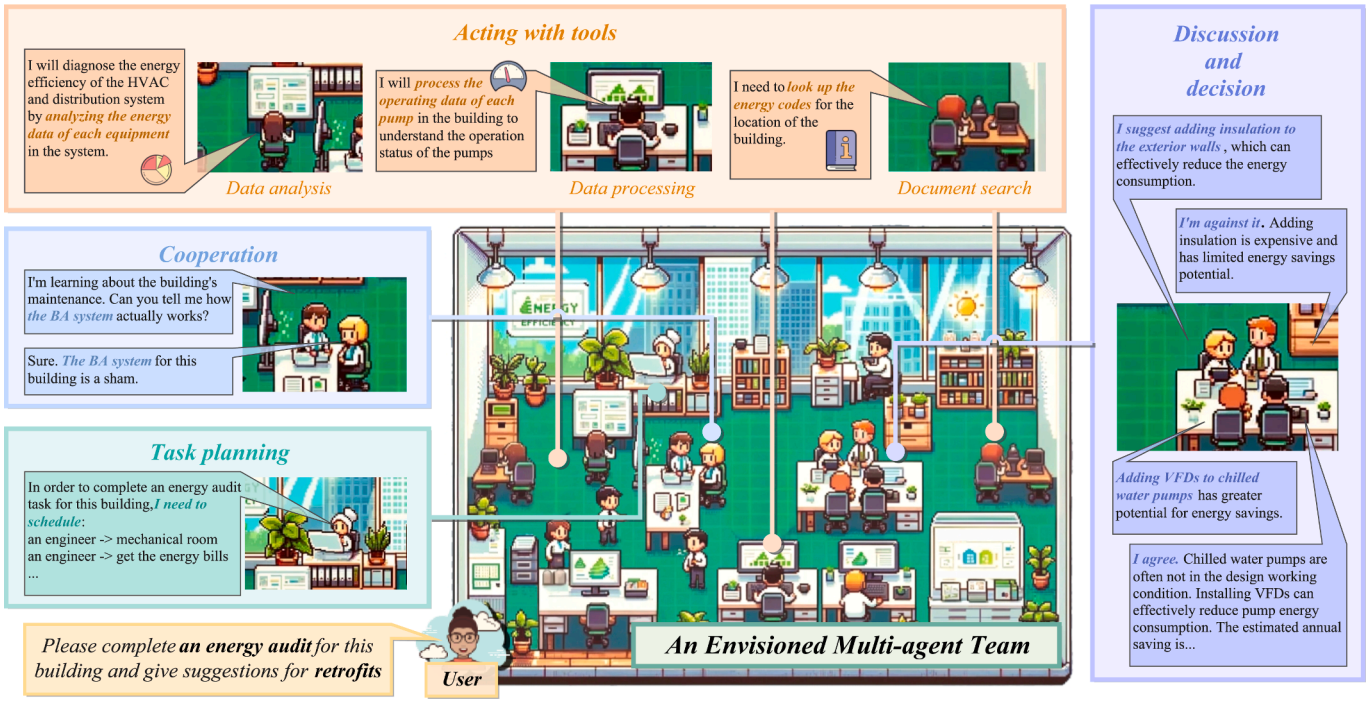


Fig. 1. Conceptual form of a multi-agent team for building energy audit and retrofit. (The conceptual graphics without textual notes and explanations are generated by DALL-E). Multiple agents individually complete tasks such as planning tasks, using tools, and so on. They communicate and discuss to finalize the decision on the retrofit proposal. The user only needs to provide the team with the necessary building information. The team will then automatically offer suggestions for audit-based energy retrofits for the building.

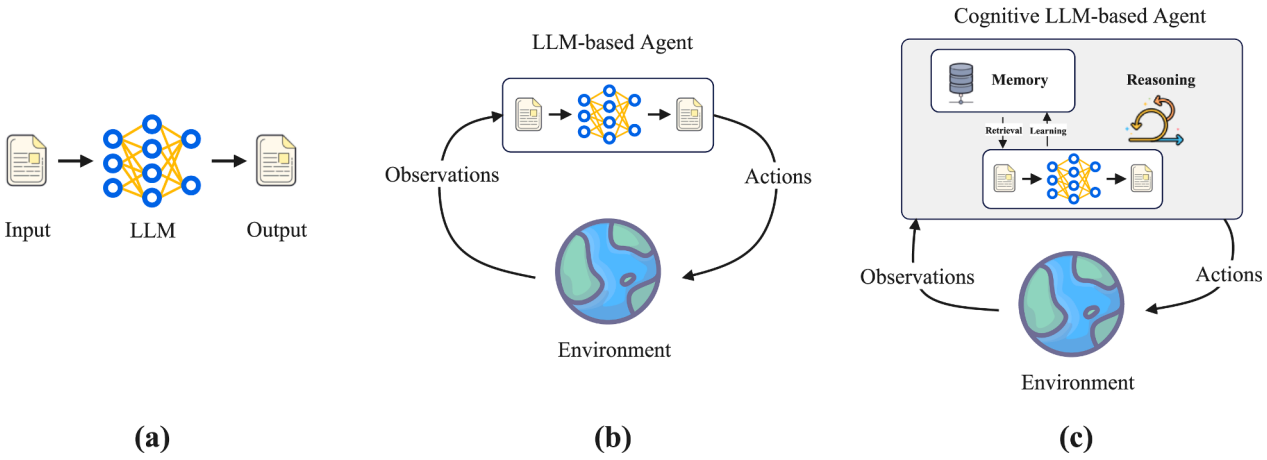


Fig. 2. Three kinds of uses of LLMs [18]. (a) Directly apply LLM models to static tasks that do not interact with the environment. The LLM simply takes text as input and outputs text. (b) Use LLMs to directly interact with the external environment by transforming observations into text and using the LLM to choose actions. (c) Use LLMs to interact with the external environment and further elicit the LLM's memory and reasoning abilities to autonomously plan observations and actions.

Table 1
Some examples for the different uses of LLMs.

The use of LLMs	Similar examples of other data-driven approaches in the building energy domain	Examples of LLM application
(a)	• Building energy forecasting based on energy meter data [19,20] • Building energy forecasting with insufficient energy data [21,22] • Non-intrusive measurements for thermal comfort [23] • Equipment fault detection and diagnosis [24]	• Data-mining for building energy conservation using GPT models [27]. • Pure natural language processing (NLP) tasks, such as academic text generation [28], IDF text file generation [29] and modification [30].
(b)	• Equipment operation control [25]; Model-free control methods for building energy systems [26]	• HVAC terminal control with GPT-4 [31]
(c)	No example (traditional data-driven algorithms do not have emergent capabilities such as reasoning)	• BIMS-GPT [32]

capabilities, they lack specific knowledge about building energy. Knowledge data in the building energy domain is diverse, including handbooks, codes, personal experience, product manuals, national policies, and more. These data, especially design, operation, and maintenance experience, lie outside the public domain (being personalized and decentralized) and are highly heterogeneous. Generic LLMs often fail to learn adequately during training to manage building energy tasks due to the data's complexity. Thus, proper organization of task inputs and outputs, along with injecting domain knowledge, is essential when developing LLM-based agents for specific domain tasks.

1.3. Aims and objectives

In this study, we aim to explore the potential of employing an LLM-based multi-agent framework for automating tasks with unstructured building data in energy optimization. We focus on knowledge-driven performance diagnosis and retrofit recommendations for diverse buildings using personalized energy audit reports. Specifically, we will extract building metadata, perform knowledge-driven yearly performance diagnosis, and provide richer but also reasonable retrofit methods for engineers to choose from. These tasks are pivotal for enhancing energy efficiency in buildings, yet automation in this area remains limited.

Currently, energy audit reports are common and important building data that are currently underutilized. These audit reports are completed by different audit units describing different buildings with different forms of data organization, which makes them highly heterogeneous. At the same time, different buildings have different system forms and energy usage characteristics, which also makes the energy optimization task for the same purpose need to be accomplished through different processes in different buildings. The diversity of input data and the personalization of task paths make it still a challenge to design a framework to automatically extract information from different reports and accomplish energy optimization tasks.

The development of a multi-agent framework based on the LLM model is a potential approach to address this challenge, but currently still unexplored. While LLMs have shown effectiveness in various knowledge tasks, they have insufficient knowledge of specialized domains (e.g., building energy domain) and may provide results accompanied by knowledge hallucination. The current outcomes produced by LLMs for specific building energy optimization tasks are unreliable due to the black-box nature of the answer generation process. In order to design a framework and enable LLMs to perform building energy optimization tasks, we need to explore ways to inject expertise into LLMs. At the same time, we also need to design a rational framework for LLMs to provide explicit knowledge sources and clear rationale when completing tasks in order to minimize the impact of knowledge hallucinations on task outputs.

To fill these research gaps, we discussed the following research questions during the development of our LLM-based multi-agent framework:

- How can we inject knowledge in the field of building energy optimization into generic LLM?
- How can we design an LLM-based multi-agent framework to automate performance diagnosis and retrofit recommendations? What is the performance?
- Is the LLM-based multi-agent system able to handle diverse inputs?

To address these questions, we designed the framework with the following basic considerations:

- The agents used in our framework should be cognitive agents developed based on LLMs. Thus, the framework we design will be able to handle heterogeneous data and accomplish personalized tasks.
- The energy optimization tasks will be divided into several subtasks and completed by multiple agents. These subtasks include language

generation, computation, language-based reasoning, and decision-making. Such language-intensive tasks can be efficiently handled by current LLMs, which excel in such tasks compared to other modalities.
- Special attention should be paid to injecting domain knowledge into generic LLMs. We believe that LLMs with domain knowledge injection can better understand the concepts in building energy optimization, thereby helping engineers accurately accomplish their tasks.

The expected value of our work is substantial. For automating building energy tasks, our work explores the effectiveness of developing new LLM-based approaches to address the critical challenge of diverse inputs and aims to reduce the workload of engineers in language-intensive tasks. In the context of solving energy optimization problems, our work aims to explore the combination of data-driven and knowledge-driven approaches using LLMs' ability to handle linguistic tasks. From an algorithmic application perspective, our work is part of the LLM application framework design, an attempt to develop a new LLM-based multi-agent framework for specialized tasks. Our work is a novel attempt, and we hope it can provide inspiration and promote the development of LLM-based agent systems in the building energy optimization.

The remaining part is organized as follows: Section 2 reviews the related work in LLM application and knowledge injection, and energy performance diagnosis and retrofit recommendation using language. Section 3 provides a comprehensive overview of the technical background for LLM application, including the fundamentals, basic concepts of LLM, and practical methods and applications of LLM-based task solvers. Section 4 describes our proposed multi-agent based framework and the knowledge injection methods used in the framework. Section 5 describes the experiments focus on the effectiveness of knowledge injection we conducted before developing the framework. Section 6 describes the case study of the proposed framework, which is divided into two parts, one on the performance evaluation and the other on the robust test on diverse engineering data. Section 7 discusses the potential of further developing LLM-based multi-agent systems as automated task solvers in building energy optimization. Finally, Section 8 concludes our work and looks forward to future work.

2. Related work

2.1. Applications of LLM and knowledge injection

Since the release of ChatGPT and GPT-4, researchers have been interested in applying LLMs in building construction industry and energy management. They discussed the potential applications of LLMs in building energy efficiency and decarbonization studies [35], as well as in the construction industry [36]. Several studies have investigated the application of LLMs in building energy management. Zhang, et al. [27] developed a data-mining framework for building energy conservation using GPT models. Rysanek, et al. [28] explored the data inference and prediction capabilities of GPT-4 in building science and generated academic language with GPT-4. Gang, et al. [29] fine-tuned an LLM model to generate IDF files for building energy modeling (BEM) with EnergyPlus. Song, et al. [31] tested the performance of using GPT-4 to control the HVAC terminal of a building. They implemented the interaction application type in Fig. 2(b) by controlling the HVAC terminal in the simulation environment directly using the actions provided by GPT-4, collecting the environment state, and then feeding it back to GPT-4. Zheng, et al. [32] developed BIMS-GPT to automatically search information in building information modeling (BIM) data to meet the user requirements. BIMS-GPT directly interacts with users and plans the search tasks to complete accurate BIM searches (i.e. application type in Fig. 2(c)). Previous discussions and research aimed to complete domain tasks with a single LLM model or single LLM-based agent. LLM-based multi-agent systems can combine the capabilities of multiple LLMs to accomplish more complex tasks. Zhang, et al. [30] utilized an LLM-based multi-agent system to modify the IDF files. In their work, multiple agents took on the tasks of modifying text in different fields of the IDF files. Table 1 summarizes the types of LLM applications for the above tasks. Currently, not enough attention has been paid to the development of LLM-based multi-agent system for building energy management tasks. Furthermore, LLM-based multiple cognitive agent systems remain an unexplored field in building energy management research.

In applying LLMs in building energy management, the performance boundaries of using generic LLMs have also received attention. Lu, et al. [37] explored the mastery of knowledge and skills in HVAC systems of different generic LLMs. Their work suggests that GPT4 and GPT3.5 have acquired some domain knowledge, but it is still insufficient for accomplishing domain-specific tasks. Some researchers have attempted to inject domain knowledge into generic LLMs to make them outperform zero-shot generic LLMs in tasks. Zheng, et al. [32] designed prompts with domain context for BIMS-GPT to help align GPT models with BIM information searches. Song, et al. [30] developed an expert demonstration dataset to add expert demonstrations in prompts and evaluate their impact on GPT performance when applied to HVAC control. Gang, et al. [29] injected BEM knowledge into the LLM model through fine-tuning. The above work demonstrates the importance of knowledge injection for generic LLMs to accomplish tasks in the field of building energy management. In order to better apply LLMs to domain tasks, knowledge injection methods deserve further exploration.

2.2. Performance diagnosis and retrofit recommendations using natural language

The main approach for energy performance diagnosis can be categorized into two types: knowledge-driven and data-driven [38]. The development of data-driven methods requires a large amount of building-specific operational data [39]. Knowledge-driven methods, on the other hand, are developed based on diagnosis rules and expert knowledge, and thus require less operational data. Knowledge-driven methods include developing energy simulation models and using energy benchmarking with energy performance indicators (EPI). The former is detailed but time-consuming [40], while the latter is easy to use and especially suitable when building layouts are not available. A typical EPI is energy use intensity (EUI) for building-level performance diagnosis. It has been benchmarked by a number of researchers [41] and is widely used in national programs [42,43] and codes [44]. To specifically identify the causes that affect energy efficiency, researchers have proposed EPIs for system-level [2], equipment-level (component-level) [45,46], and room-level [39] to achieve detailed performance diagnosis. Knowledge-driven methods, especially the energy benchmarking with EPI, perform diagnostics based on rules [40]. These rules can be accurately described using language, which opens up opportunities to automate the diagnostic process using LLM. Currently, research on automated energy efficiency diagnosis with LLMs remains a gap.

Retrofit recommendations in audit reports, focused on energy efficiency retrofits for buildings, require a great deal of expert experience and knowledge [47]. Traditional knowledge-based methods require abundant research time, whereas AI methods are less time-consuming but less interpretative. Case-based approaches are considered to be an effective combination of the two, utilizing retrofit cases from similar buildings to aid in decision-making [3]. In the case-based approach, researchers use NLP methods to find similar case buildings and leverage the tacit knowledge hidden in these cases [48]. Many studies have shown that the case-based approach is effective in early remodeling recommendations [49], indicating that the knowledge contained in natural language is beneficial for early remodeling recommendations. The popularity of LLMs gives hope for using them in knowledge-based retrofit recommendations. Rysanek, et al. [28] used LLM to provide retrofit recommendations, but without clear reasons, and the knowledge on which the decisions were based was not explicit due to the highly

black-boxed nature of LLMs. Not enough attention has been paid to retrofit recommendations that utilize explicit knowledge.

3. Technical background for LLM application

3.1. Fundamentals of LLM

3.1.1. LLM model structure

The development of LLM has gone through several stages: basic language modelling, pre-trained language models (PLMs), and then LLMs. In the basic language modeling stage, the crucial technique was Word2vec (i.e., word embedding) [50,51]. Word2vec vectorizes words to ensure that those with similar meanings are proximate in a lower-dimensional vector space, and thus enabling computers to address language-related issues. It describes the essential foundations of machine language modelling problems.

PLMs were developed based on a key model, the Transformer, and employed a key training method: pre-training and fine-tuning. The Transformer [52] was proposed to solve sequence-to-sequence tasks and is now widely used in various fields [53]. Unlike traditional models [55,56], the Transformer excels at capturing dependencies over long distances [54] by introducing self-attention into a general Seq2seq structure (i.e. encoder-decoder structure) [57,58]. Pre-training and fine-tuning [59] enhance model's performance without extensive labeled data for the downstream task. Pre-training enables the model to obtain generalization performance in language modelling [60]. Fine-tuning further tailors the model to specific tasks, ensuring optimal performance [61].

Latter, researchers found that scaling the model size or the training data size of PLMs can improve model capacity and enable them to solve complex tasks [62]. For example, GPT-3 (175 billion parameters) [63] can solve downstream tasks with simple instructions before fine-tuning, whereas GPT-2-XL (1.5 billion parameters) [64] cannot [53]. These large PLMs became known as LLMs [65]. Depending on the part of Transformers and the type of modification used in model structure, LLM models can be divided into two categories: encoder-decoder or prefix-decoder language models (also known as non-GPT style models) and causal-decoder language models (also known as GPT style models) [66]. Models with different architectures specialize in different tasks, but a recent study stated that scaling the model size may reduce the ability differences caused by model architecture [67]. Currently, most state-of-the-art LLMs are causal-decoder models, e.g., GPT-4, LLaMA [68], Claude, and Bard, while T5 [69] is an encoder-decoder model and GLM [70] is a prefix-decoder model.

3.1.2. Adaptation methods for LLM

Most access to LLMs is currently through the prompting interface or API, such as GPT-4, Claude, and Bard, while some open-source LLMs (e.g., LLaMA) can be accessed by loading model checkpoints. None of these access methods require changes to the model structure, making LLMs highly black-boxed and integrated in application. Methods for adapting them to specific tasks are engineering-oriented, focusing on the process rather than requiring extensive AI expertise. Furthermore, these methods can be generalized across various domains [34].

Current adaptation methods are inextricably linked to three emergent abilities of pre-trained LLMs.

- Instruction following: After fine-tuning using a mixture of multiple downstream task datasets (i.e., instruction tuning), LLMs perform well on unseen downstream tasks [71].
- Few-shot learning (in-context learning) [63]: By providing a few natural language commands and/or one task demonstration (i.e., one-shot, zero-shot), LLMs can perform the desired tasks without additional training or gradient updates.
- Multi-step reasoning: Small language models cannot solve complex reasoning tasks [72]. LLMs can solve such tasks with the chain-of-thought (CoT) prompting strategy [73].

These abilities reflect that LLMs can adapt to downstream tasks without gradient updates. Fine-tuning is no longer the only option for adapting to downstream tasks, providing new possibilities for LLM adaptation methods [74]. The main approaches to model adaptation can be divided into two groups.

- Supervised fine-tuning (SFT): Fine-tuning the pre-trained model to enable it to follow instructions. Reinforcement learning with human feedback (RLHF) is a significant type of SFT method that aims to align the LLM with human values, and is used in the development of InstructGPT [71] and ChatGPT [13].
- Prompt engineering: Prompt engineering aims to design suitable task instructions or specific in-context learning strategies to help elicit the emergent abilities of pre-trained LLMs [34] without extra gradient updates [74]. The LLM can be instructed to perform the desired task by specifying in a prompt the role it needs to play, the formatting requirements for the output, etc [75]. Besides, CoT [73], Reasoning and Acting (ReAct) [76], and Retrieval-Augmented Generation (RAG) [77] are popular prompting strategies used in prompt engineering tasks. The CoT prompting strategy achieves complex reasoning capabilities through intermediate reasoning steps. The ReAct prompting strategy combines the reasoning and action capabilities of LLMs, enabling them to solve more complex reasoning and decision-making tasks. The RAG strategy incorporates retrieved documents into the prompt to provide additional information for LLM generation.

3.1.3. Current shortcomings of LLM

LLMs have been proven capable of addressing a wide range of real-world problems that extend beyond the traditional definition of downstream tasks in NLP [66]. The abilities of generic LLMs offers hope for the development of LLM-based task solvers to replace large amounts of repetitive work in domain-specific tasks (practical methods and applications can be found in Section 3.2). It should be noted, however, that current LLMs still have the following shortcomings which may affect task performance.

- Hallucination: Hallucination refers to the fact that LLMs sometimes generate text that contains fictitious, meaningless, or incorrect information [78]. For example, when asked about the specific standard suitable for designing an air-conditioning system in China, the LLM may mislead user into adopting incorrect standards.
- Knowledge cutoff: Because LLMs are trained on data that is time-sensitive, they may generate responses that are outdated. For example, when users inquire about a multi-zone infection risk assessment model of airborne viruses on cruise ships proposed in 2023 [79], an LLM trained with data before 2023 may provide outdated or irrelevant details due to its training data limitations.

For further development and application, knowledge injection techniques are considered to be effective in addressing the effects brought by such shortcomings (detail descriptions can be found in Section 3.2.1).

3.2. Practical applications of LLM-based task solvers

3.2.1. Knowledge injection techniques

Model adaptation methods can be used to inject domain knowledge into LLMs, reducing the impact of the above shortcomings when applying LLMs to specific domain tasks. For instance, ChatDoctor [80] injects medical knowledge into LLaMA-7B by SFT with a patient-physician conversation dataset, effectively improving the LLM's ability to provide

medical advice. Instead of updating model parameters to learn domain knowledge, a common method is to combine queries with relevant documents and feed them into the LLM [81]. The RAG strategy with proper prompt design has been proven to be a potent method for bolstering the LLM's capabilities in generating factually grounded responses [77]. This method involves augmenting the encoding process with extensive documents or passages retrieved from unstructured data, thereby improving the LLM's performance in outputting accurate answers, reducing hallucinations, and keeping the knowledge up-to-date [82].

SFT can inject domain-styled knowledge into the model and enhance the specialization of the model output, but the need for gradient update training makes it difficult to update frequently. RAG, on the other hand, has the advantage of injecting real-time updated knowledge data into the model by simply updating the knowledge source. At the same time, RAG supports models that output results based on explicit knowledge sources, which makes it applicable to a wider range of application scenarios. Recent research by Microsoft has demonstrated that RAG performs better than SFT in the injection of domain knowledge into LLMs [83]. However, SFT also has the advantage of enabling models with domain style output capabilities. Therefore, most domain models are built following a roadmap that initially involves injecting domain knowledge based on RAG, and then combining this with SFT to further align with the requirements for domain-specific tasks (e.g., legal counseling in the field of law). ChatLaw [84] and ChatDoctor [80] were developed using SFT to enhance the style and ability to solve domain-specific tasks, and using RAG to ensure that knowledge is up-to-date and accurately.

3.2.2. Agent and multi-agent systems

The task planning and reasoning capabilities emerging from LLMs give them great potential for further development into AI agents [85]. Researchers use LLMs as the major component of the brain or controller of the agents and expand their perceptual and action space through strategies such as multi-modal perception and tool utilization [86]. These agents can exhibit reasoning, planning abilities and the ability to interact with the environment through prompting strategies for LLMs, such as CoT and ReAct [18]. With LLM's in-context learning capabilities, agents can have short-term memory and learn to utilize tools through structured prompts with instructions and examples [87]. LLM-based agents have been applied to various real-world tasks, such as travel planning [88], web automation [89], math theorem proving [90], and chemical scientific discovery [91].

Since LLMs have natural language understanding and generation capabilities, they can interact seamlessly, enabling collaboration and competition between multiple agents [92]. Through collaboration, multiple agents can handle more dynamic and complex tasks than a single agent. In an LLM-based multi-agent system, multiple agents are given different profiles and collaborate on a task in a shared task operational environment. They are defined with their roles, traits, actions, and skills through prompt engineering and can interact with the operating environment using tools (calls to the agents-environment interface) [86]. Currently, several LLM-based multi-agent systems have been developed for multiple applications, such as automated software development [16], automated 3D modeling [93], and human society simulation [94].

3.2.3. Model evaluation methods

Many LLM applications, agent systems, and multi-agent systems are developed to automate human work. Since work automation tasks vary in reality, the evaluation methods for these systems are varied and context-specific. Some work automation tasks have a research base and public benchmark datasets, such as HumanEval [95] for automated software development [16]. Researchers can evaluate systems using these public benchmark datasets. However, some tasks do not have benchmark datasets and require subjective evaluation with human annotation [85]. For example, DesignGPT [96] is a multi-agent system developed for design collaboration in which two experts independently score each design solution. Additionally, LLMs are sometimes asked to be the judge to carry out subjective assessments in evaluation [85]. Chemcrow [97] is a multi-agent system designed for chemical research, with one evaluator being an LLM and one human expert conducting the assessment.

4. Proposed framework

4.1. Multi-agent based framework overview

4.1.1. Workflow description

The framework aims to use the building energy audit report as the input and then complete energy efficiency diagnosis and audit recommendations for the building. Firstly, the energy audit report is loaded to construct a knowledge database. Meanwhile, a retriever tool (Fig. 3 (d)) is developed to retrieve the needed information from the knowledge database and then inject it into LLMs. Other documents required to complete the task, such as energy codes, handbooks, and engineering experience reports, are loaded in the same way to build the knowledge database and develop the corresponding retrievers. Then an LLM-based multi-agent team completes the task through three stages, as shown in Fig. 3 (a). The three stages are building information processing, performance diagnosis, and retrofit recommendation.

- Building information processing: This stage aims to extract building metadata from the audit report and create a structured summary of building metadata to provide contextual information for subsequent tasks.
- Performance diagnosis: In this stage, we enable multilevel diagnosis of energy performance by calculating the EPI.
- Retrofit recommendation: Based on the first two stages, we try to combine explicit knowledge to make retrofit recommendations for the given building.

Detail methods used in three stages are described in Section 4.3.

4.1.2. Agent roles and functions

We divided task completion in every stage into three steps: plan, execute, and summarize. Thus, we defined three kinds of roles in the framework: Planner, Researcher, and Advisor. The Planners is responsible for dividing a complex task into several parallel tasks. Researchers are responsible for acting with different tools to accomplish each parallel task. The Advisor is responsible for summarizing the results of all parallel tasks. In every stage, multiple agents with these three roles cooperate and act with tools to complete subtasks. Since the division of tasks in the building information processing stage can be determined by the composition of the schema, the roles of Planner and Advisor are omitted.

As shown in Fig. 3 (b), every agent is developed based on an LLM model, with different roles and tasks defined through structured prompts. In addition, CoT and ReAct prompting strategies were employed to enable the agents to engage in task-planning, self-reflection and acting with tools. Agents can automatically plan tasks based on the information and data available for each building, allowing them to handle tasks with different inputs. The prompts used in our framework are categorized into two main types based on the prompting strategy employed: prompts that use only the CoT strategy (marked as CoT-only prompts), and prompts that use both the CoT and ReAct strategies (marked as CoT-ReAct prompts). Detailed information and examples of these prompts can be found in Appendix E.

4.1.3. Tool integration

Two kinds of tools are developed to help agents complete their tasks: the Python sandbox tool and the retriever tool. The Python sandbox tool, as shown in Fig. 3 (c), can run the Python codes generated by the LLM and print the results. Agents can perform complex calculations and data analysis by generating Python codes and then using the Python sandbox

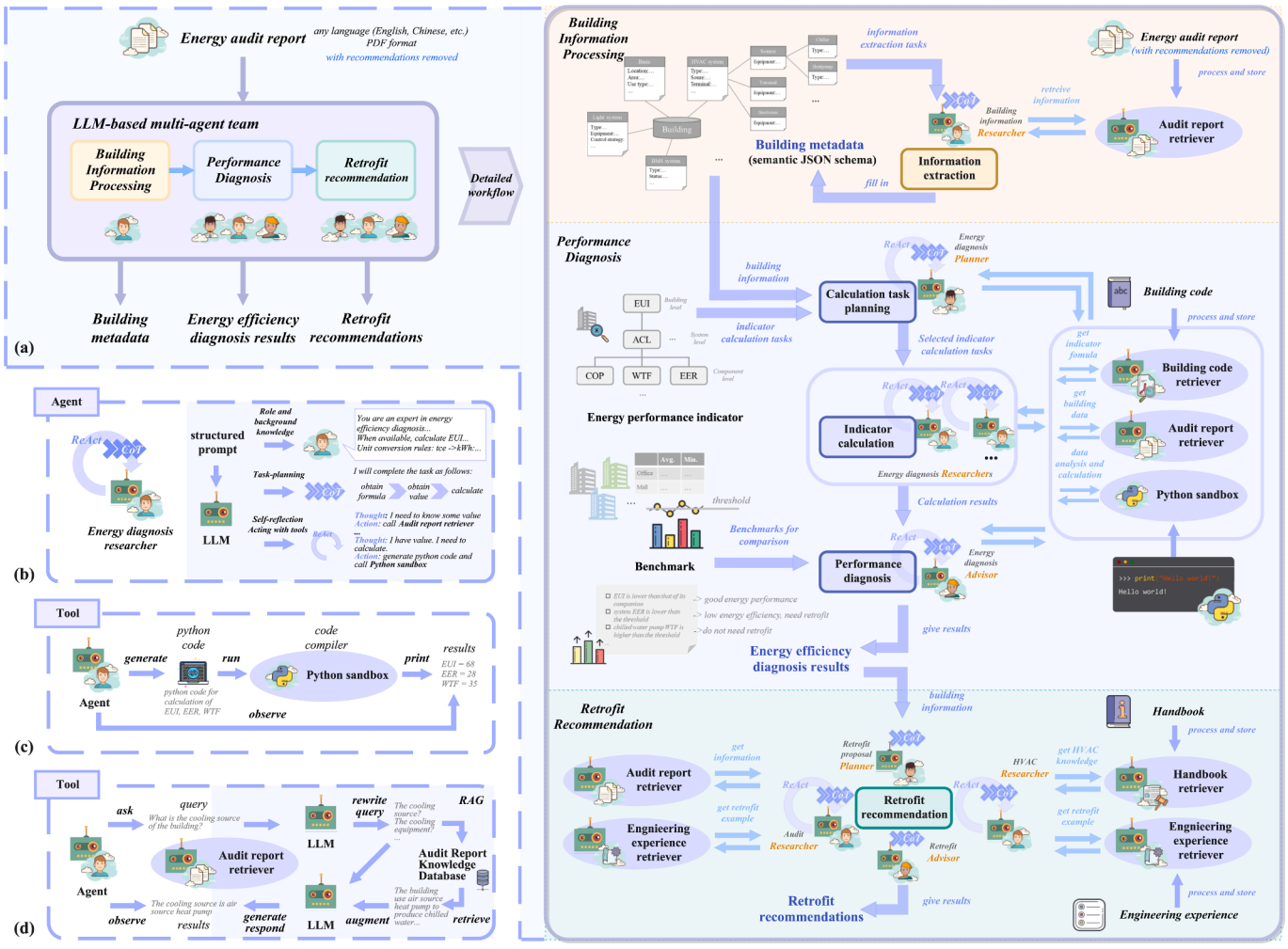


Fig. 3. The proposed multi-agent based framework. (a) The entire workflow. The entire workflow consists of three stages. Each stage is completed by agents with different roles and operating different tools. (b) An example of the LLM-based agent development. Use a prompt to specialize the role and the task of an LLM to become an agent. When completing a task, the agent will plan the steps needed, determine the tools to use, and judge whether the task has been completed. (c) An interaction example between an agent and the Python sandbox tool. The agent generates Python code and observes the results printed by the Python sandbox after running the code. (d) An interaction example between an agent and the retriever tool. The agent asks a query, and then the retriever retrieves the knowledge and provides the result. The retriever was developed based on the RAG strategy.

tool [98]. For example, when an agent needs to calculate the annual operating hours of a chiller based on schedules for different seasons and different day types, it can generate code and then call the Python sandbox tool to run the code and complete the calculation. The retriever tool is developed by injecting knowledge into an LLM through the RAG prompting strategy and can answer document-specific questions. The interaction example for the retriever tool in Fig. 3 (d) applies to all retriever tools in the workflow, such as the audit report retriever, handbook retriever, and so on. Agents can collect accurate information from specific documents through a retrieval tool developed for those documents. Detailed methods used in the retriever tool development are illustrated in Section 4.2 and Fig. 4.

4.2. Knowledge database and retriever tool

4.2.1. Injected multi-modal knowledge

The generic LLM usually doesn't have enough knowledge for specific tasks in building energy optimization. Therefore, further development of the LLM-based agent requires knowledge injection. The aim of developing the knowledge database and the retriever tool is to help the LLM-based agent acquire more knowledge when completing tasks. The knowledge that needs to be injected and the knowledge files we used are listed in Table 2. To support knowledge injection, knowledge should be embedded into a form in which machines can compute distances between knowledge meanings. We use natural language to describe knowledge and embed the knowledge into vectors using a dynamic text embedding model, text-embedding-ada-002 from OpenAI [100,101].

4.2.2. Development of the knowledge database

The knowledge files we used are in multiple modalities (including images, text, tables, formulas, etc.). Therefore, we loaded the unstructured documents in different modalities separately and used Markdown formatting to preserve the structure of the document. Text data could be used directly after loading, while data in other modalities required further processing. For example, table and formula data may be structured incorrectly, and image data required additional processing to retain information in text format. We specified the target format for processing, invoked an LLM with vision capability to process the data into informative text, and performed necessary manual checks. All the informative texts were filled back into the Markdown structure with necessary delimiters. The Markdown texts were split into chunks based on the delimiters (i.e., "#") while preserving hierarchical structure and separating different original modalities of the data. All the text chunks were embedded as vectors using the text embedding model, and then

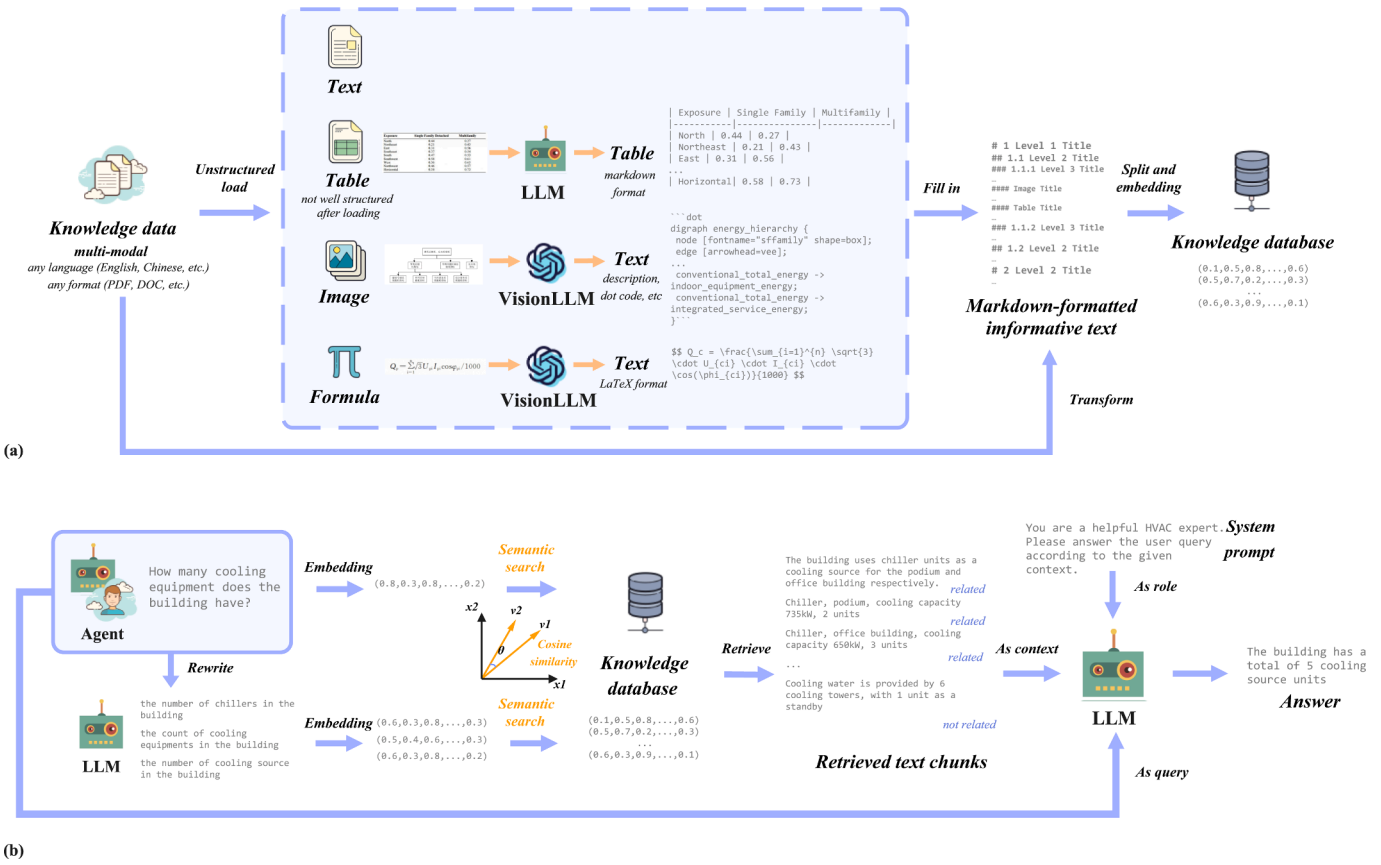


Fig. 4. The development workflow for knowledge database and retriever tool. (a) The knowledge database development workflow. Detailed descriptions can be found in Section 4.2.2 and Appendix A. (b) An example of the workflow when the retriever tool is called. An agent calls the tool with a query and then an LLM rewrite the query to enrich its expression. Cosine similarity is used to measure the semantic similarity between vectors, and similar vectors are retrieved from the knowledge database using both the raw and the rewritten queries. Retrieved vectors are then recovered to be retrieved text chunks, in which most of the information is related to the query while some may not be relevant. Subsequently, roles and tasks are assigned to an LLM using a prompt. All retrieved text chunks are used as context, with the raw query for the LLM to generate an answer.

Table 2
Knowledge needed for energy efficiency diagnosis and retrofit decision.

Retriever tool name	Needed information	Knowledge document
Handbook retriever	Accurate fundamental knowledge of building energy systems, especially heating, ventilating and air conditioning (HVAC) systems, related principles, etc.	• ASHRAE fundamental handbook
Audit report retriever	Basic information about the building, information about the building energy system, information about the building energy use, etc.	• Energy audit report for the specific building
Engineering experience retriever	Engineering experience and case experience related to energy-saving retrofits. Including common energy-saving retrofit measures, corresponding energy savings and empirical data on recovery cycle, etc.	• An engineering technical report: Effectiveness of energy retrofit methods in public buildings [99]
Building code retriever	Detailed information on building energy efficiency design parameters, performance indicator calculation methods and thresholds for the region where the building is located.	• Economic operation of air-conditioning systems [44]

stored in a vector database. With knowledge embedding stored, the vector database was acknowledged as the knowledge database. The knowledge database development workflow is shown in Fig. 4 (a). Detailed information and examples in this process can be found in A ppendix A.

4.2.3. Retriever tool development

The retriever tool completes the retrieval task and generates an answer when an agent asks for information. It is developed based on the RAG prompting strategy, and the detailed example of the workflow is illustrated in Fig. 4 (b). A rewrite process is added to the workflow to enrich the expression of input queries and thus improve retrieval performance [101]. Cosine similarity is used to measure the semantic similarity of the knowledge and queries. A structured prompt with a prompt template is used to call an LLM to give the final answer. The structured prompt includes the input user query and the related retrieved knowledge in text chunks. To further reduce the LLM's hallucination [102], the structured prompt also includes a role instruction, such as having the LLM act as an HVAC expert and honestly admitting when it lacks the relevant knowledge.

4.3. Framework stages

4.3.1. Building information processing

Determining a structured framework (i.e., semantic data schema) for information extraction can ensure consistency and standardization of information summaries, thus facilitating subsequent tasks [103]. We adapt

the information form from the Energy Audit Standards for Public Buildings DG/TJ08-2114–2020 [104] to an expandable JSON semantic schema as the target structure for information extraction. This schema includes the building metadata required for an energy audit and offers common options. The required information includes basic information, building envelope, energy system type, equipment performance, and more.

The extraction of building information falls under the information extraction (IE) tasks in NLP, which aim to extract structural knowledge (e.g., entities, relationships, and events) from pure natural language text [105]. Our task is more complex than traditional IE tasks since the semantic schema is expandable according to the system type of the building, and the amount of information extracted varies among cases. Meanwhile, some information needs to be inferred according to the context if it is not directly mentioned, such as the distribution system type. To address these complexities, we designed a CoT-only prompt and fill it with different one-shot examples and necessary background knowledge for different information extraction task (detailed information on the prompt components can be found in A ppendix E). A Researcher agent is tasked with accomplishing this IE task using these prompts.

4.3.2. Performance diagnosis

The energy performance diagnosis method used in our work is multi-level diagnosis based on EPI. The EPIs used in our work are listed in Table 3 [44,2]. The EPI at the building level applies to all buildings, whereas the applicable EPIs at the HVAC component level vary from building to building depending on the system type. The benchmark for comparison of the EPI at the building level is the statistical distribution of buildings of the same type in the local area. The benchmark for an EPI at the HVAC component level is the threshold value, which we use as provided by [44].

To calculate these EPIs, relevant data needs to be collected from audit reports. However, the data available for calculation varies from case to case due to the heterogeneity of the audit reports [106]. For example, the total power consumption is always reported, while the power consumption of a specific device may not always be available. Meanwhile, missing data situations are widespread among audit reports and they prevent accurate energy efficiency diagnostics in many buildings. To make further retrofit recommendations, a quantitative performance diagnosis would be helpful. Thus, designing a robust performance diagnosis method to cope with different missing data scenarios is a major challenge for the automated workflow. In our work, estimation methods (listed in Table 4) are used when the directly needed data is not available. These estimation methods were adapted from the Energy Audit Standards for Public Buildings DG/TJ08-2114–2020 [104]. Although these estimation methods may not reflect the real state of the building system very accurately, they can provide engineers with a reference result when there is insufficient data. At the same time, we required the model to provide the calculation process along with the output results and clearly indicate whether the estimation has been carried out or not, in order to minimize the impact caused by the estimation. As shown in Fig. 3 (a), several agents with three distinct roles collaborate to complete the task. The flow of planning, subtasks, and actions of an agent within a calculation task is shown in Fig. 5. Due to the various system types and data heterogeneity of the building, every action flow of the agents is different.

4.3.3. Retrofit recommendation

Retrofit recommendations should be appropriate to the building and supported by sufficient background knowledge. Thus, we excite the reasoning abilities of LLMs with CoT prompting strategy to allow LLMs to analyze and make specific retrofit recommendations in the context of the current energy use of the building. We also design a multi-agent collaboration framework that allows LLMs to give recommendations together with reasons based on explicit knowledge and reliable knowledge sources. As shown in Fig. 3 (a), the Planner agent at this stage will analyze the previously generated building metadata and performance diagnosis results and plan the direction of retrofit recommendations based on a set of commonly used retrofit directions derived from the literature [47]. The needed information, including previous results and retrofit direction options, will be provided to the agents as a component of the CoT-ReAct prompt and thus guide their planning. Several Researcher agents will then collect relevant HVAC background knowledge, engineering experience, and the necessary building information by interacting with the handbook, engineering experience, and audit report retriever tools (detailed information of the documents is listed in Table 2). Agents will communicate with each other and collaborate to collect information for comprehensive analysis. Finally the Advisor agent will integrate all the collected information to make specific recommendations for the building with corresponding reasons and knowledge sources within the original directions proposed by the Planner agent. The framework will provide recommendations under the ASHRAE energy audit Level I requirements along with a description of the rationale, advantages and disadvantages, thus helping human experts to accomplish further decision making.

It is worth mentioning that currently the aim of this stage is to provide richer but also reasonable retrofit choices for engineers and thus help them speed up their work without having to spend extra effort searching for information. The results do not need to be totally certain and consistent among runs, but outputs that differ too much between runs can make the framework unreliable. The Planner, Researcher, and Advisor agent structure in this stage is designed to reduce the randomness of the recommendation output. The Planner agent is asked to choose suitable and sufficient directions from a set of options and the downstream agents will make recommendations within the direction choice. Thus, each run of the multi-agent framework will take into full consideration the retrofit directions applicable to the building system and then output retrofit recommendations that cover consistent directions.

5. Experiment: Effectiveness of knowledge injection with the retriever tool

5.1. Experimental setup

To test the knowledge injection effectiveness of the retriever tool, we took one knowledge document as an example, developed the knowledge

Table 3
Energy performance indicators (EPI) for multi-level diagnosis.

Level	Description	EPI
Building	Energy use of the whole building	$\mathrm{EUI} = \frac{\textit{Total power consumption}(\mathrm{kWh})}{\textit{Building area}(\mathrm{m}^2)}$
Building	Cooling load allocation for building area	$\mathrm{ACL} = \frac{\textit{Configured cooling load}(\mathrm{W})}{\textit{Building area}(\mathrm{m}^2)}$
HVAC component	Energy performance of the cold and heat source	$\mathrm{COP} = \frac{\textit{Accumulated cooling load}(\mathrm{kWh})}{\textit{Total power consumption of the source equipments}(\mathrm{kWh})}$
HVAC component	Energy performance of the distribution system	$\mathrm{WTF} = \frac{\textit{Accumulated cooling load}(\mathrm{kWh})}{\textit{Total power consumption of the distribution system}(\mathrm{kWh})}$
HVAC component	Energy performance of the HVAC terminal	$\mathrm{EER} = \frac{\textit{Accumulated cooling load}(\mathrm{kWh})}{\textit{Total power consumption of the HVAC terminals}(\mathrm{kWh})}$

Table 4
Estimation method for the unavailable data.

Needed data	Estimation method
Configured cooling load	$Configured\ cooling\ load = \sum_{\substack{all\ non-standby \\ cold\ sources}} Rated\ cooling\ capacity\ of\ each\ cold\ source$(kW)
Accumulated cooling load	$Accumulated\ cooling\ load = \eta \times Configured\ cooling\ load$(kW) where η is a factor to estimate the ratio of a building's total annual load to its annual peak load. The exact value is related to the region where the building is located. In our work, the value is taken as the statistical average obtained from the simulation for the Shanghai region.
Total power consumption of the source equipment / Total power consumption of the distribution system/ Total power consumption of the HVAC terminals ($E_{subsystem}$)	$E_{subsystem} = \sum_{\substack{all\ relevant\ equipment \\ in\ the\ subsystem}} Energy\ consumption\ of\ the\ equipment$(kWh) When *Energy consumption of the equipment* is not available from the report, $Energy\ consumption\ of\ the\ equipment$(kWh) $= Rated\ power\ of\ the\ equipment$(kW) $\times$ $Annual\ operating\ hour\ of\ the\ equipment$(h) When *Annual operating hour of the equipment* is not available from the report, $Annual\ operating\ hour\ of\ the\ equipment = \sum_{s\in S}\left(\sum_{d\in D_{workday}} h_{s,d}^{workday} + \sum_{d\in D_{weekend}} h_{s,d}^{workeend}\right)$ where S is the set of seasons; $D_{workday}$ and $D_{weekend}$ represent the sets of weekdays and weekends, respectively; $h_{s,d}^{workday}$ and $h_{s,d}^{workeend}$ represent the number of operating hours of the equipment on weekdays and weekends, respectively.

* The subscript symbol *subsystem* represent heat and cold source equipment, distribution systems, and HVAC terminal equipment, depending on the scenario.

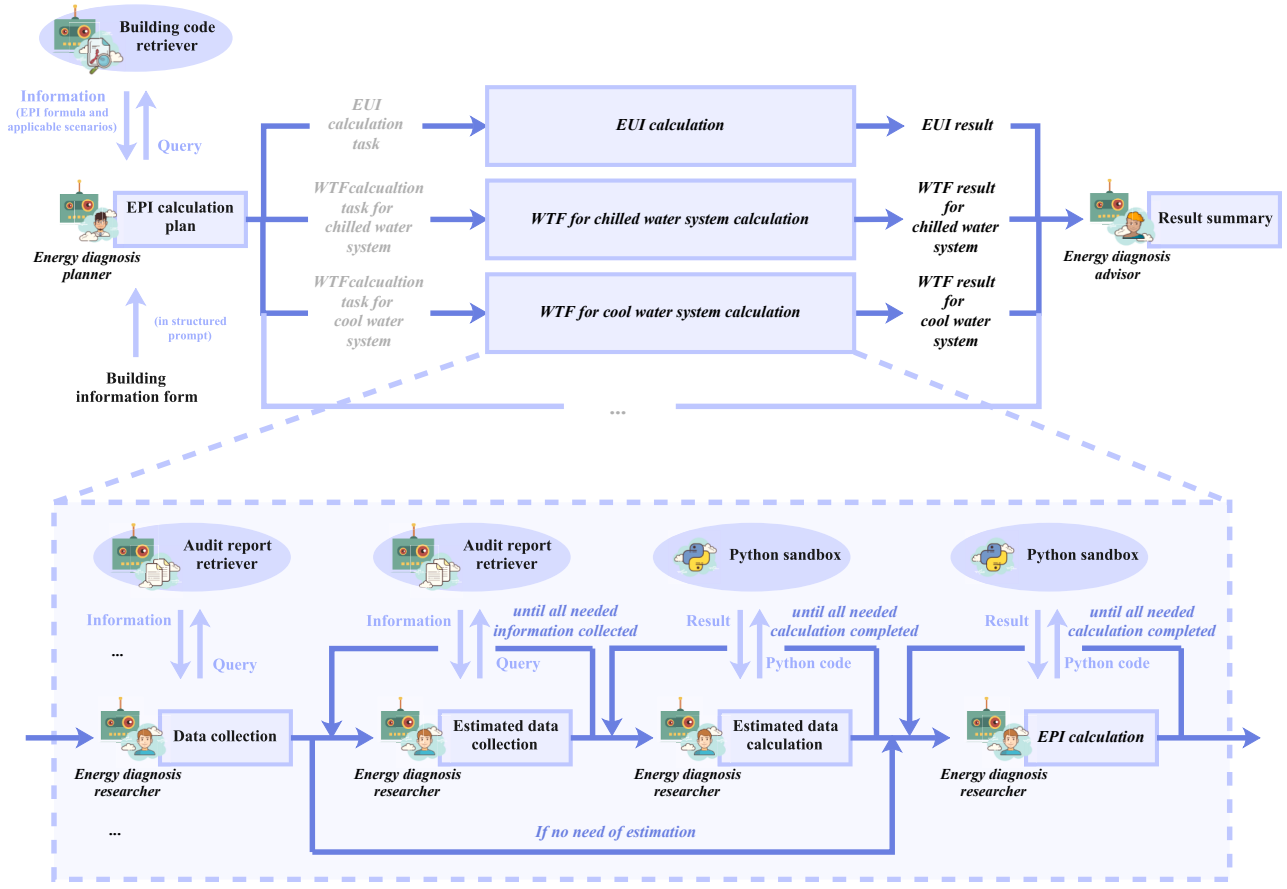


Fig. 5. The flow of planning, subtasks, and actions of the agents within an efficiency diagnosis task. When given a building, an agent will first plan several EPI calculation tasks based on the building information form and the applicable scenarios of EPIs. For each EPI calculation task, an agent will call the Audit report retriever tool to get the required data and generate the calculation code to call the Python sandbox tool to perform the calculations. If the required data is not available, the agent will perform the estimation, calling the same tools. The actions of acting with tools may be repeated multiple times until the corresponding task is completed. Finally, an agent will summarize the results of all EPI calculations.

database, and created a retrieval tool individually. The knowledge document we chose for experiment is the ASHRAE Fundamental Handbook. We selected the ChatGLM-std model (marked as GLM-std) and the GPT3.5-Turbo model as the base models. We then developed the knowledge database in Chroma and created two retriever tools using Langchain [107] in Python. Additionally, we chose the GPT 4-Turbo model, which is regarded as the state-of-the-art (SOTA) LLM, as the benchmark model for comparison. All models were called using the official APIs provided by the model issuers (OpenAI and ZhipuAI). Test cases were tested in zero-shot settings. Other settings for the retriever tool development can be found in Appendix B.

To evaluate the knowledge injection effectiveness of the retriever tool, we generated an evaluation dataset. The dataset was constructed with multiple statements, and the LLM was asked to judge whether each

statement was right or wrong. The accuracy of the judgement indicates whether the LLM captured the related knowledge or not [108]. We used ASHRAE Fundamental Handbook and a textbook used in China to generate the evaluation dataset (detailed information can be found in Appendix B). The dataset contains 7,549 statements, making its size comparable to the benchmark datasets commonly used in the field of LLM applications [109]. We used metrics for classification problems to evaluate the models. The metrics include Accuracy, F1 score, and Balanced Accuracy (formulas can be found in Appendix B).

5.2. Results and analysis

Table 5 lists the accuracy on the evaluation dataset. Two retriever tools (marked as RAG-GLM and RAG-GPT 3.5, respectively) achieve better performance in all metrics on the evaluation dataset than the base models and the SOTA model (GPT 4-Turbo). Due to the limitation of parameter quantity, the GPT series models performed better than GLM model on the evaluation dataset. According to the results of F1 score (negative), the GPT 4-Turbo model is stronger than the GPT 3.5-Turbo model in judging incorrect statements.

We use Balanced Accuracy to be the main metric in performance analysis (results for other metrics can be found in A ppendix C). As shown in Fig. 6 (a), the two retriever tools show better performance than the three generic LLMs on each theme of the evaluation dataset, with the RAG-GPT 3.5 achieving the best performance on most themes. The retriever tools achieve better accuracy on all themes, demonstrating the effectiveness of knowledge injection. In Fig. 6 (b) and 6 (c), we take the SOTA model as the benchmark and calculate the ratio to the SOTA model's balanced accuracy of the two adaptation cases before and after the adaptation. In each theme, the retriever tools perform better than the SOTA model, while their base models perform worse than the SOTA model. These results indicate the effectiveness of the knowledge injection, regardless of the theme of the knowledge and the performance of the basic models.

6. Case study

6.1. Framework development and evaluation metrics

We developed the multi-agent framework with Langchain and CrewAI [110] in Python. All agents are built with structured prompts based on the CrewAI agent. The multi-agent interaction was implemented using CrewAI sequential processes combined with text file writing and reading for key information (detailed descriptions can be found in A ppendix E). The implementation of the Python sandbox tool was adapted from the Python REPL (Read-Eval-Print Loop) tool in Langchain. The retriever tools were developed in the same way as Section 5, we use GPT 3.5-Turbo as the base model for retriever development. In the workflow realization of the first two stages, we used GPT 4-Turbo as the LLM backbone of the agent because most of the tasks at these stages were reasoning tasks, and GPT 4 was evaluated to be the current SOTA in terms of reasoning abilities [111]. Since there were no complex tasks in the retrofit recommendation stage, we used GPT 3.5-Turbo-16 k as the backbone to reduce costs. The temperature parameters of the LLMs used in the framework development were set to the lowest randomness setting (i.e., 0 for the GPT series).

Table 5
Overview of the accuracy on the evaluation dataset.

	Accuracy	F1 score (positive)	F1 score (negative)	Balanced Accuracy
GLM-std	0.7824	0.8305	0.6959	0.7728
GPT 3.5-Turbo	0.8114	0.8590	0.7153	0.7857
GPT 4-Turbo	0.8183	0.8581	0.7474	0.8145
RAG-GLM	0.8559	**0.8894**	0.7933	0.8485
RAG-GPT 3.5	**0.8567**	0.8873	**0.8031**	**0.8607**

To explore the end-to-end performance of the framework, we manually labeled the baseline results and defined metrics for the building information extraction and performance diagnosis process. For the building information extraction task, we used two evaluation metrics: Precision (accuracy of target extracted information) and Recall (correctness of extracted information). For performance diagnosis, we defined a score to evaluate the degree of task completion, inspired by [97]. Detailed information can be found in A ppendix D. For retrofit recommendations, we compared the results to the original recommendations given in the report and defined the coverage of the original recommendations by the results as a metric (Coverage). We also invited domain experts to judge whether the generated recommendations and reasons are reasonable. Considering LLMs are probabilistic models and may generate different results each time, we ran the framework three times for each case to investigate the average performance. Every trial is conducted together with an AgentOps [112] session so that all behaviors of the agents can be observed and replayed.

6.2. End-to-end performance analysis

6.2.1. Case setup

We collected 63 energy audit reports (20 of which were electronic) from building owners and auditors. From these, we chose an informative and well-structured audit report as input to test the effectiveness of the framework (detailed information can be found in Appendix D). After careful manual checking, the text contained no obvious grammatical errors and was easy to read without too many long and complex sentences. To avoid information leakage, we manually removed the energy efficiency calculations and retrofit recommendations from the report. In the following discussion, this case is marked as Case #0.

6.2.2. Results and performance analysis

Table 6 summarizes the results of the validation metrics for each trial. In the building information extraction stage, the Recall metric reaches an average of 0.9627, indicating that the metadata extracted into the schema is mostly accurate. The Precision metric results suggest that the schema is well-expanded and most of the required metadata is extracted. This indicates that the building metadata generated by the framework is accurate enough, requiring only a few manual adjustments to achieve complete accuracy. This greatly reduces the amount of manual work involved in organizing the metadata from the text data.

The performance diagnosis score results show that most of the indicator calculation processes were planned correctly, information was extracted correctly, and most calculations were accurate. The main reason for not achieving a full score was the defective presentation of the text data in the report. The building uses forms of energy other than electricity, but the expression "*annual energy consumption*" was used to present the annual electricity consumption, leading to the extraction of incorrect information. Fig. 7 (c) provides an example of agent behaviors during this stage, showing how agents with three different roles cooperate, act with tools, and complete the task step by step.

In the three trials, the framework gave 15–30 recommendations, with varying coverage of the original recommendations. The unmentioned recommendations were the direct replacement of heating and cooling source equipment and the addition of solar power. Instead, recommendations with similar objectives, such as adding inverters for heating and cooling source equipment, adjusting temperature differentials and flow rates to optimize the operation of chiller systems, adding energy storage devices, and participating in demand response programs, were mentioned. Because the current framework does not support retrofit cost analysis with reliable data sources, it is unable to rank the retrofit recommendations or accomplish decision-making for recommendations with similar objectives. Therefore, we can consider the

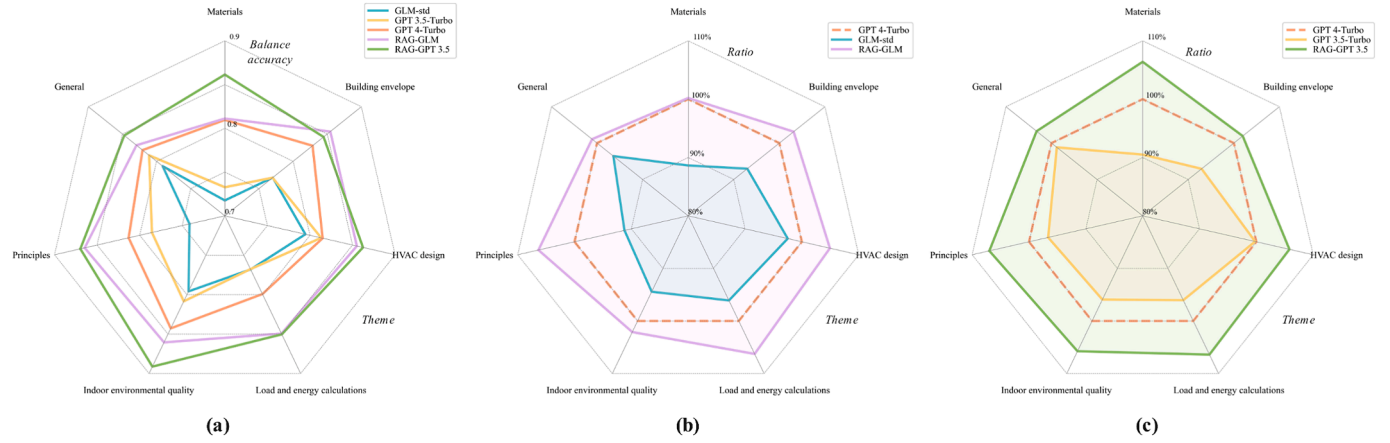


Fig. 6. Result comparison (a) Balanced accuracy for each theme on the evaluation dataset. The categories of themes are inspired by the parts of the ASHRAE fundamental handbook; (b) Comparison of the balanced accuracy of GLM-std model before and after adaptation with GPT 4-Turbo; (c) Comparison of the balanced accuracy of GPT 3.5-Turbo model before and after adaptation with GPT 4-Turbo.

Table 6
Overview of the results (the results in brackets are the average results of three times).

Number of the trial	Building information extraction		Performance diagnosis	Retrofit recommendation
	Precision	**Recall**	**Score**	**Coverage**
1	0.9647	0.9794	0.9750	0.8182
2	0.9294	0.9400	0.9125	0.5454
3	0.9176	0.9688	0.9125	0.6363
Result	0.9647 (0.9373)	0.9794 (0.9627)	0.9500(0.9250)	0.8182(0.6667)

current results to be reasonable and the directions of the retrofit recommendations to be consistent enough. The framework provides reasons along with the data sources when giving recommendations, as shown in Fig. 7(b). The recommendations were also reviewed and considered reasonable by an expert we invited.

The average token and money cost for the framework is summarized in Table 7. While it costs about $5 to complete the entire process, this is far less than the cost of a team of laborers and shows potential to significantly reduce repetitive human labor and costs.

6.3. Robust testing on diverse engineering data

6.3.1. Case setup

To test the performance of the framework on diverse cases, we chose energy audit reports of three buildings as inputs (marked as Case #1, #2, and #3). The three audit reports were completed by different audit units with different report structures and data organization. They include different available energy data with different data units. At the same time, the three buildings have different forms of energy systems, which require different information to be extracted and different indicators to be applied for energy efficiency diagnosis. Detailed information of the buildings can be found in A ppendix D.

The reports were simply processed to remove the retrofit recommendation information. Several data quality problems can be observed from the reports, as listed in Table 8. Readability refers to the ease with which a text can be read and understood based on linguistic features within the text. It is one of the key features of a text. We used the readability metrics provided by cntext [113] to evaluate the readability of the textual content of the reports. The readability metric was calculated by combining the length of subordinate clauses and the proportion of adverbs and conjunctions in the sentences. A higher value of the indicator means that the text is more complex and less readable. As listed in Table 9, all three reports were less readable than Case#0 and each had several quality problems.

6.3.2. Performance analysis

Table 10 lists the results of the robustness tests. In the building information extraction, the accuracy of the extracted information and the schema extension are mostly correct although they are degraded by the quality of the report text. Only a small amount of adjustment is required to achieve complete correctness. The performance diagnosis stage is somewhat affected by contextual inconsistencies and unclear references in sentences. These issues affect the accuracy of information extraction in some of the trials, thus impacting the final results. Although the planning and calculations in the tasks are mostly correct, proper preprocessing of the unstructured data is necessary and may further help improve performance. The framework made 15–18 recommendations in each trial. The retrofit recommendations cover consistent directions, though there is some deviation in the specific expressions. These recommendations basically covered the original recommendations. And some recommendations were richer and more detailed than the original ones, along with verifiable reasons, which can provide reference for engineers. Overall, it can be suggested that the framework can handle the heterogeneous tasks brought about by diverse engineering data well and has good robustness.

7. Discussion: LLM-based multi-agent systems as automated task solvers

LLMs are of significant value in language-intensive tasks. Our framework exploits unstructured building data while assigning different professional roles to different LLMs and injecting linguistic knowledge into LLMs, providing a solution that effectively bridges the gap between data-driven and knowledge-driven approaches to building energy optimization tasks. There is a large amount of unstructured data in the building energy optimization process, and unstructured data often carries a large amount of expert knowledge. In the future, a building energy optimization automation framework that combines data and knowledge can be further developed based on LLM utilizing building unstructured data such as work orders. At the same time, the multi-agent model structure has a clearer and more explainable structure than a single LLM model, and thus may help engineers to better understand the results and developers to further improve the model performance. Implementing the data-driven and knowledge-driven hybrid approach with LLM framework is a promising direction for the building and energy engineering domains that require trustworthy automated task solvers.

Meanwhile, in our framework, we developed the LLM models as cognitive agents, exploiting their ability to use tools, plan, learn and

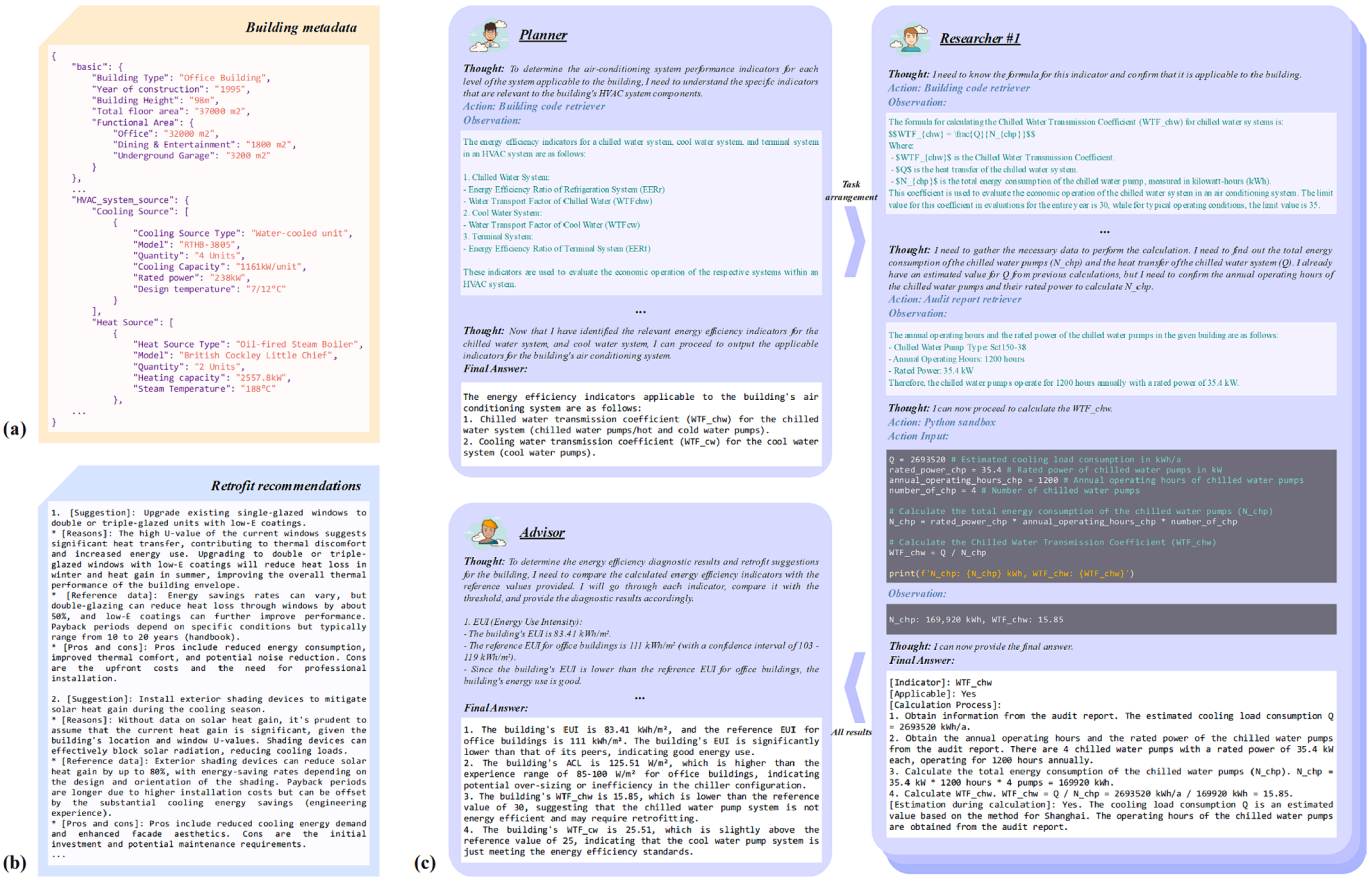


Fig. 7. Examples of the results. (a) An excerpt example of the building metadata in JSON semantic schema generated by the framework. (b) An excerpt example of the retrofit recommendations generated by the framework. (c)An excerpt example of the agent behaviors during the performance diagnosis stage. A Planner agent (on the top left side) first plans the EPI calculation tasks. Then several Researcher agents (on the right side) act with tools to complete the tasks and output the calculation result in a given format. Finally, an Advisor agent (on the bottom left side) summarizes all results and provides diagnostic advice. All the agents are acting follow the ReAct strategy.

Table 7
Overview of the cost.

Stage	Average token cost	Average money cost
Building information extraction	104,610	$ 1.16
Performance diagnosis	303,987	$ 2.91
Retrofit recommendation	1,193,992	$ 1.08
Total	1,602,589	$ 5.15

Table 8
Example data quality problems.

Problem	Examples	Examples from
Inconsistency in context	• The table lists 4 pumps, while the text says there are 3 pumps. • The rated power of the pump in the table is 15 kW, while the text says it is 22 kW.	Case #1
Overly informative table	• Include information on substations, chillers, air handling units in a single table	Case #2
Sentences with unclear references	• In 'There are 6 sets of cooling towers, 6 sets of circulating water pumps, 3 sets of chilled water pumps, 3 sets of cool water pumps, total power; 419.5kw.', the subject that the total power belongs to is unclear.	Case #3

Table 9
Overview of the report qualities.

Case	Sentence readability score	Observed problems
#0	22.75	
#1	32.67	• Inconsistency in context
#2	29.19	• Overly informative table • Sentences with unclear references
#3	24.50	• Overly informative table • Inconsistency in context

Table 10
Overview of the results in robust test (the results in brackets are the average results of three trials).

Case building	Building information extraction		Performance diagnosis	Retrofit recommendation
	Precision	**Recall**	**Score**	**Coverage**
#1	0.9153 (0.8908)	0.9775 (0.9554)	1.0000 (0.8889)	1.0000(0.9583)
#2	0.9375 (0.8906)	1.0000 (0.9624)	0.9454(0.9121)	1.0000(1.0000)
#3	0.8961 (0.8355)	0.9463 (0.8992)	0.8488(0.8345)	1.0000(0.9444)

interact with the environment, thus extending the boundaries of the capabilities of LLM as pure language models. In the future, researchers can further enable the LLM-based agents to use specialist tools in the field of building energy optimization. This will not only enable the LLM to handle tasks that go beyond plain text but also connect it closely to existing work, leading to the further development of automated workflows that effectively reduce human labor. Using LLM's memory capabilities, it is also possible to combine it with the original case-based research to optimize and improve the memory flow, allowing the agent

to continually learn from cases. In addition to this, using LLM's planning capabilities, we can cope with changing data structures and scenarios and develop generic solutions. In conclusion, LLM-based multi-agent systems will hopefully free engineers from complex case-based processing, and LLM-based agent systems have a wide range of potential for development in automation tasks.

8. Conclusions and future work

8.1. Conclusions

Automating tasks in building energy optimization will help reduce the engineers' workloads and allow them to focus on creating environmentally friendly indoor environments. Over 85 % of building data is in unstructured format and a large amount of energy use information is hidden in these data. Effective use of unstructured data can help further improve energy efficiency and thus reduce building energy use. Currently, tasks involving unstructured building data rely on a great deal of human labor and are far from being automated. Inspired by the outstanding performance of LLMs in solving general language-intensive tasks, we proposed an LLM-based multi-agent framework for automating tasks with unstructured building data. The framework includes three main stages: building information processing, performance diagnosis, and retrofit recommendation. In each stage, multiple agents with three kinds of roles: Planner, Researcher and Advisor, cooperate and act with tools to complete tasks. To equip generic LLMs with necessary knowledge to complete specialized tasks, we utilized the RAG prompting strategy and developed knowledge databases together with retriever tools to inject knowledge into the LLM. We completed experiments to test the knowledge injection effectiveness and explored the performance and robustness of the entire framework on various cases. With the findings from the case studies, we discuss the opportunity for developing LLM-based multi-agent systems as automated task solvers.

The main conclusions of our work are as follows:

- Knowledge injection is vital in adapting generic LLMs to solve domain tasks. Our work shows that the RAG prompting strategy is efficient in domain knowledge injection to LLMs.
- Our proposed framework can automatically extract information from unstructured audit reports to generate building metadata in a given schema, perform knowledge-based energy efficiency diagnosis, and provide retrofit recommendations with verifiable reasons. In the case studies, our framework delivered mostly reliable results at a cost of only about $5 and showed robustness in dealing with various inputs and heterogeneous tasks. It can greatly reduce the repetitive human work in unstructured data-based building energy optimization tasks.
- The results of our case study show that the data quality affects the further utilization of unstructured data. We suggest that research on normalization and preprocessing of unstructured data is necessary and important.
- Our work tests the ability of LLMs to handle unstructured knowledge of buildings and the ability of LLM-based cognitive agents to plan when facing heterogeneous tasks. We believe that LLM-based multi-agent systems have great potential for developing trustworthy and generalized automated task solvers in building energy optimization.

8.2. Future directions

Although the current framework performs well in the case study and passes the robust test, some work can be done to improve the performance and capabilities of the framework.

(1) Knowledge injection

Generic LLMs do not have sufficient domain task knowledge, and the knowledge injection process is important and necessary. In our work, the RAG approach effectively completes the knowledge injection into the LLM, enabling it to perform specialized tasks. The RAG approach is lightweight and supports real-time knowledge updates to the model, but the token cost is high, and long-term use is costly. Fine-tuning, as another LLM knowledge injection method, also deserves attention and experimentation. More discussions can be carried out on different knowledge injection methods for different task types.

(2) Data quality

Audit reports completed in different countries, regions and audit units will always in different language and structures, and possibly have different data quality problems. Our work enumerates three data quality issues present in audit reports. These quality problems affected the performance of the framework in some of the test trials in the robust test of the framework. While the performance of the framework is generally good, implementing effective data preprocessing will help develop a more robust framework.

(3) Data exchange

Energy audits of buildings are a very important process during the operational phase of a building. A large amount of unstructured data from energy audit reports has not been fully utilized. The first stage of our proposed framework explores the information extraction of audit reports into building metadata in the form of an expandable JSON semantic schema and demonstrates the potential of LLM in extracting information and normalizing unstructured data. Further development of the LLM framework to normalize different audit reports into a more universally applicable schema [106] will be valuable in advancing the building data exchange during energy optimization tasks.

(4) Equipped with specialist tools

By acting with different retriever and Python sandbox tools, our framework allows LLMs to provide automated yearly energy efficiency diagnosis and qualitative retrofit recommendations based on audit reports. However, in practical energy optimization tasks, the wider application scenarios are daily and hourly energy efficiency diagnosis and retrofit decision-making combined with energy-saving potential analysis. Many specialist tools have been developed to facilitate some of the automated energy optimization tasks. For example, building performance simulation tools (e.g., EnergyPlus) could simulate the hourly energy performance of a given building, and thus benefit building performance optimization and retrofit decision [114]. Further development of the LLM framework to enable them to utilize specialized tools can further improve their capabilities in solving actual energy optimization tasks.

CRediT authorship contribution statement

Tong Xiao: Writing – review & editing, Writing – original draft, Methodology, Conceptualization. **Peng Xu:** Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgement

This research is funded by National Natural Science Foundation of China (No.52161135202).

Appendix A:. Detailed information on knowledge document processing

Most of the knowledge documents in the field of building energy optimization are in PDF format, which is well-suited for communication because it supports accurate display in different systems. Since the underlying structure of a PDF file does not map to the logical structure of a document, we need to perform additional processing. Figure A1 shows four example pages from knowledge documents in PDF format. The PDF files are in diverse layouts and often include textual, tabular and visual data. In addition, there are significant differences in the layout and structure of different PDF documents, especially audit reports issued by different audit units (as shown in Figure A1(b) and (c)). Thus, in order to preserve as much information as possible about the document, we need to process the data in different modalities and preserve the document structure.

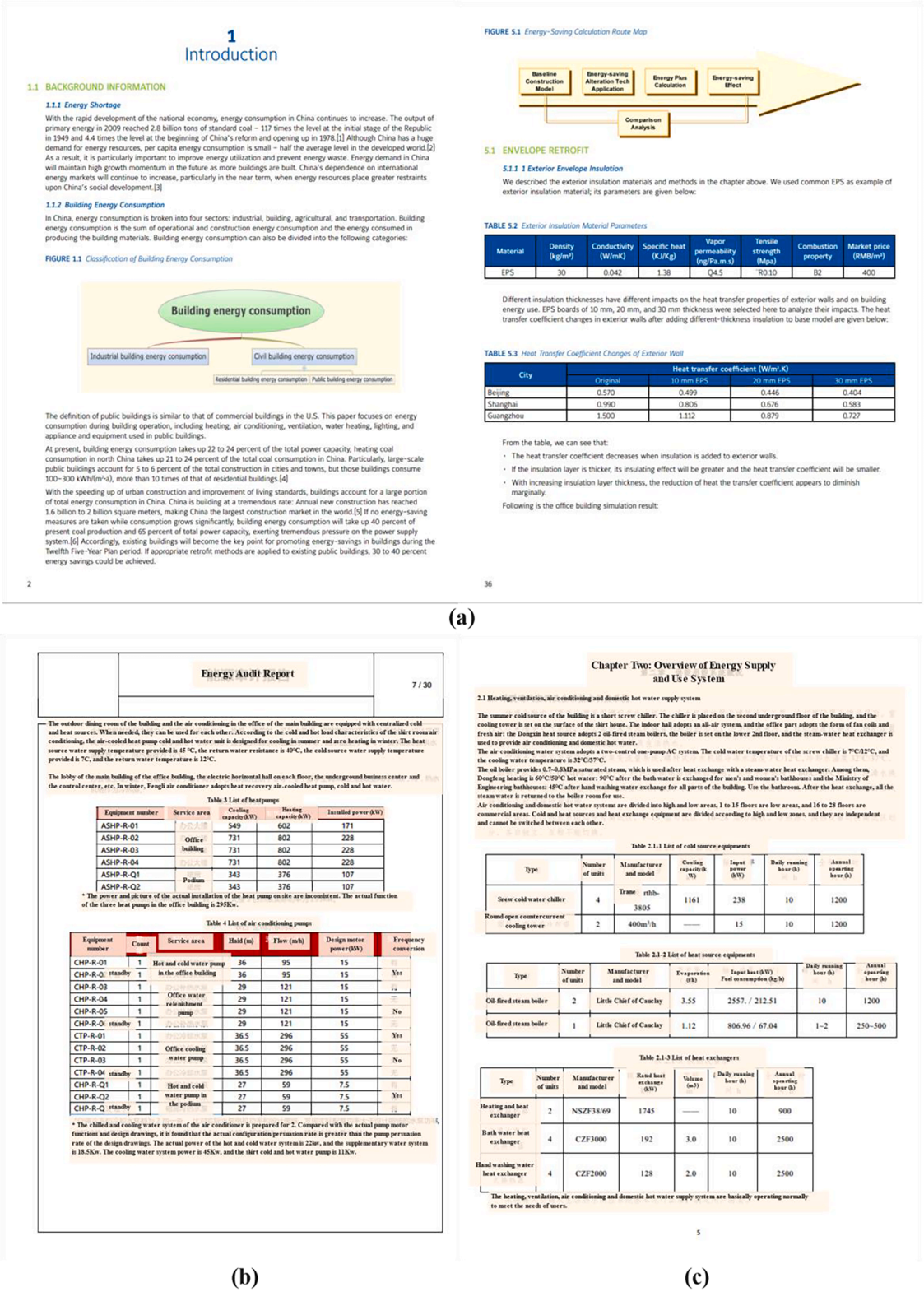


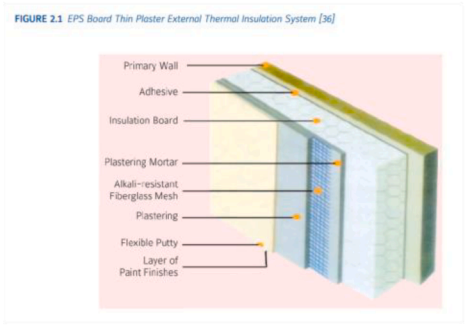
Fig. A1. Four example pages from PDF documents. (a) Two pages from the engineering technical report used in our framework. (b) A page from the audit report of the Case # 1 building, translated from Chinese into English. (c) A page from the audit report of the Case # 0 building, translated from Chinese into English.

As shown in Fig. 4 (a), we first loaded the documents unstructuredly using the Python package PDFPlumber and PDFMiner. Data in different modalities within a single document was loaded separately during this process. We then used an Optical Character Recognition (OCR) model along with regular expression matching to transform the document into Markdown format to preserve the structure of the document. The OCR model we used is Nougat, which is open-soured by Meta. Due to the variable document structure, this process required some manual review.

Next, we invoked an LLM (e.g., moonshot-v1-32 k) and a VisionLLM (e.g., GPT-4 Vision), specifying the target format for processing tables, formulas and images into informative texts. The target format for table processing is the Markdown-formatted table, and for formula processing is LaTeX-formatted text. Image processing targets are categorized into two scenarios: for flowcharts, the processing target is GraphViz dot code, while for other images, the processing target is descriptive text. It should be noted that the OCR model we used (Nougat) can directly convert most tables and formulas written in English into our target formats when transforming the document into Markdown format. LLMs and VisionLLMs are primarily used for data processing that Nougat cannot handle (e.g., processing of Chinese tables, formulas interspersed with Chinese text, and all images).

Due to the limitations of current processing technology, the document data processing involved a small amount of manual checking. We performed manual checks to ensure the preservation of the document structure and the processing results of the VisionLLMs. Manual checks included verifying that the rendering results of LaTeX and Graphviz dot code are consistent with the original content, and that the descriptive text expresses the core content of the image. For the processing of tables, formulas, and flowcharts, the model we used (GPT-4 Vision) performs well, requiring little manual adjustment in our work. However, the process of generating descriptive text for images requires more manual adjustments. Since most of the images in the audit reports were visualizations of other modal information in the original text, we did not process them in the automation process. For the ASHRAE Fundamental Handbook and the engineering technical report, the processing of pictures was carefully checked manually. Such checks ensure the stability of the constructed tool and minimize the impact on the performance of the automation framework. Several results of the data processing can be found in Fig. A2.

All information texts are then split into text chunks for storage in the vector database. Texts were split based on hierarchical delimiters (i.e., “#”) to preserve hierarchical structure. Text of the same level and the same original modality was split into the same chunk, and the hierarchical structure to which the text belongs was expressed in the field and stored. Chunking by level avoids the loss of information caused by cutting, while preserving the hierarchical information of the document. It also helps to keep data from different original modalities stored in the knowledge database in the form of text chunks separately. Additionally, we limited the size of the text chunk to avoid overly long text (i.e., more than 1000 words) at the same level, which could result in an oversized text chunk and thus prevent subsequent prompts from exceeding the API access limit of the LLM model. When splitting text chunks by size, we set the overlap parameter to minimize the loss of information due to splitting. The splitting method according to chunk size was mainly used in the processing of the ASHRAE Fundamental Handbook, as this document contains a lot of text within a level. The settings for the chunk size and the overlap parameters in the framework implementation can be found in Appendix B. It should be noted that the chunk size and the overlap parameters are tunable hyperparameters. However, splitting by chunk size was not widely used in our practice because it was uncommon for the text to be too long. Meanwhile, quantitatively assessing the impact of these two parameters on the overall model is expensive. Thus, we did not further optimize the settings for the chunk size and the overlap parameters. In future work, the data processing process can be further improved by optimizing these two parameters.



(a)

```
#### Figure 2.1 EPS Board Thin Plaster External Thermal Insulation System
```

The diagram is illustrating a typical construction process for a primary (or exterior) wall that includes insulation. Here's a step-by-step description of the message conveyed by the diagram:

1. **Primary Wall**: This is the main structural element of the building, which provides support and separates the interior from the exterior environment.

2. **Adhesive**: A bonding agent is applied to the primary wall to ensure that the insulation board will adhere properly to the wall surface.

3. **Insulation Board**: This is a panel made from insulating materials designed to reduce heat transfer and improve the energy efficiency of the building. It is attached to the primary wall using the adhesive.

4. **Plastering Mortar**: A layer of plastering mortar is applied over the insulation board. This is a mixture of cement, sand, and water that serves to create a smooth and even surface on top of the insulation.

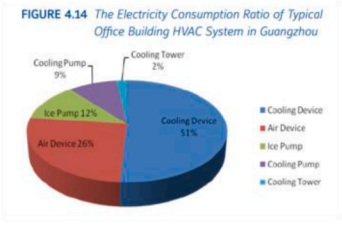
5. **Alkali-resistant Fiberglass Mesh**: This mesh is placed over the plastering mortar to reinforce the wall and prevent cracking. It is resistant to the alkaline substances that can be found in the mortar, ensuring the longevity of the wall system.

6. **Plastering**: Additional layers of plastering mortar are applied over the fiberglass mesh to further smooth out the wall surface and integrate the mesh into the wall structure.

7. **Flexible Putty**: This is a type of sealant used to fill any gaps or holes in the wall, ensuring a seamless and airtight finish.

8. **Layer of Paint Finishes**: The final step is to apply paint or other finishes to the wall. This not only enhances the aesthetic appeal of the wall but also provides additional protection against environmental factors.

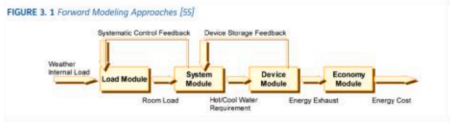
Overall, the diagram is showing a sequence of steps taken to insulate a primary wall and prepare it for final finishing, with a focus on thermal insulation, structural integrity, and aesthetic appeal.
```




(b)

```
#### Figure 4.14 The Electricity Consumption Ratio of Typical Office Building HVAC System in Guangzhou

| Category    | Percentage |
|-------------|------------|
| Cooling Device | 51%     |
| Air Device  | 26%        |
| Ice Pump    | 12%        |
| Cooling Pump | 9%        |
| Cooling Tower | 2%       |
```



(c)

```
#### Figure 3. 1 Forward Modeling Approaches [55]
```dot
digraph {
LM [label="Load Module"]
SM [label="System Module"]
DM [label="Device Module"]
EM [label="Economy Module"]

// Process flow
LM -> SM [label="Room Load"]
SM -> DM [label="Hot/Cool Water Requirement"]
DM -> EM [label="Energy Exhaust"]

// Feedback loops
SM -> LM [label="Systematic Control Feedback"]
DM -> SM [label="Device Storage Feedback"]

// Aesthetics: invisible edges to align feedback above the modules
LM -> EM [style=invis]
SM -> DM [style=invis]

// External influences
node [shape=plaintext]
weather [label="Weather"]
internal [label="Internal Load"]
weather -> LM
internal -> LM
}
```
```

**TABLE 4.2** *Thermal Characteristics Parameters of Enclosure*

| Material | Density (kg/m³) | Conductivity (W/m K) | Specific heat (KJ/KgK) |
|---|---|---|---|
| Steel-reinforced concrete | 2500 | 1.740 | 0.92 |
| Aerated concrete | 700 | 0.220 | 1.05 |
| Crushed stone concrete | 2300 | 1.510 | 0.92 |
| Cement mortar | 1800 | 0.930 | 1.05 |
| Lime-and-cement mortar | 1700 | 0.870 | 1.05 |
| Confined clay brick masonry | 1800 | 0.810 | 1.05 |
| Cement expanded perlite | 800 | 0.260 | 1.17 |
| EPS | 30 | 0.042 | 1.38 |
| XPS | 35 | 0.034 | 1.40 |

(d)

```
#### Table 4.2 Thermal Characteristics Parameters of Enclosure

| Material                | Density (kg/m³) | Conductivity (W/m K) | Specific heat (KJ/KgK) |
|-------------------------|-----------------|----------------------|------------------------|
| Steel-reinforced concrete | 2500          | 1.740                | 0.92                   |
| Aerated concrete        | 700             | 0.220                | 1.05                   |
| Crushed stone concrete  | 2300            | 1.510                | 0.92                   |
| Cement mortar           | 1800            | 0.930                | 1.05                   |
| Lime-and-cement mortar  | 1700            | 0.870                | 1.05                   |
| Confined clay brick masonry | 1800        | 0.810                | 1.05                   |
| Cement expanded perlite | 800             | 0.260                | 1.17                   |
| EPS                     | 30              | 0.042                | 1.38                   |
| XPS                     | 35              | 0.034                | 1.40                   |
```

The contributions of occupants, lighting, and appliance gains to peak sensible and latent loads can be estimated as

$$q_{ig,s} = 136 + 2.2A_{cf} + 22N_{oc} \quad (30)$$

$$q_{ig,l} = 20 + 0.22A_{cf} + 12N_{oc} \quad (31)$$

where

$q_{ig,s}$ = sensible cooling load from internal gains, W
$q_{ig,l}$ = latent cooling load from internal gains, W
A_{cf} = conditioned floor area of building, m²
N_{oc} = number of occupants (if unknown, estimate as $N_{br} + 1$)

(e)

```
The contributions of occupants, lighting, and appliance gains to peak sensible and latent loads can be estimated as

$$ \[q_{ig,s}=136+2.24_{df}+22N_{oc} \tag{30}\] $$

$$ \[q_{ig,l}=20+0.22.4_{df}+12N_{oc} \tag{31}\] $$

_where_

$$\[q_{ig,s}=\text{sensible cooling load from internal gains, W}\] \[q_{ig,l}=\text{latent cooling load from internal gains, W}\] \[A_{df}=\text{ conditioned floor area of building, m}^{2}\] \[N_{oc}=\text{ number of occupants (if unknown, estimate as $N_{br}+1$)}\]$$
```

Fig. A2. A few examples of processing images, tables, and formulas. On the left is a screenshot of the example data from the original file, and on the right is the corresponding results in the informative text after processing. The text in blue is the processing result by VisionLLM. (a)&(b) Two examples of non-flowchart image processing; (c) An example of flowchart processing; (d) An example of table processing; (e) Three examples of formula processing.

Appendix B:. Detailed settings for the experiment

Table B1 lists the detailed settings of the knowledge database and tool development in Case 1. The reason why we choose the GLM-std and the GPT3.5-Turbo is because the former is the most commonly used non-GPT style LLM (>7 billion parameters) and the latter is the most commonly used
```

LLM today. All the models we chose have more than 7 billion parameters and are called using the official APIs.

Table B1
Detailed settings of the experiment.

Setting	Value
Text embedding model	OpenAI text-embedding-ada-002
Text chunk size	800
Text chunk overlap	50
Text embedding vector dimension	1536
Vector database	Chroma
Vector similarity metric	Cosine similarity
Related text chunks count	5

In evaluation dataset generation, we used ASHRAE fundamental handbook and a textbook used in China as the text source. We split the text into chunks and generate questions for each chunk to allow as much knowledge as possible to be brought to attention. The text chunk size is 1000 and the text chunk overlap is 0. Inspired by the outstanding ability on language editing and rewriting of LLM, we called the GPT3.5-Turbo-16 k API to generate a specific number of statements under a CoT prompt based on the provided chunk. The proportions of correct and incorrect statements were also controlled by prompts to avoid the extremely imbalance of correct and incorrect statements. A total of 7763 statements were generated by LLM, and after a simple cleaning (removing statements that need to be judged by context), 6808 statements remained. Besides, 741 statements were generated from the textbook and thus the dataset has a total of 7549 statements. The overview is listed in Table B2, with the categories inspired by the parts of the ASHRAE fundamental handbook.

Table B2
Overview of the evaluation dataset.

Category	Correct	Incorrect	Total
Principles	677	279	956
Indoor environmental quality	778	487	1265
Load and energy calculations	1487	917	2404
HVAC design	490	257	747
Building envelope	272	121	393
Materials	311	117	428
General	1003	353	1356
Total	**5018**	**2531**	**7549**

The following metric for classification problem is used in our work.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

$$Accuracy\ of\ Positive\ Class = \frac{TP}{TP + FP}$$

$$Accuracy\ of\ Negative\ Class = \frac{TN}{TN + FN}$$

$$Balanced\ Accuracy = \frac{Accuracy\ of\ Positive\ Class + Accuracy\ of\ Negative\ Class}{2}$$

where TP denotes the number of samples in which the positive class was correctly predicted; TN denotes the number of samples in which the negative class was correctly predicted; FP denotes the number of samples in which the negative class was incorrectly predicted to be a positive class; FN denotes the number of samples in which the positive class was incorrectly predicted to be negative.

Appendix C:. Detailed results for the experiment

Table C1 lists the results of different metrics for each theme on the evaluation dataset.

Table C1
Accuracy for each theme on the evaluation dataset.

Accuracy	GLM-std	GPT 3.5-Turbo	GPT 4-Turbo	RAG-GLM	RAG-GPT 3.5
Principles	0.7563	0.8128	0.8232	**0.8828**	0.8766
Indoor environmental quality	0.8032	0.8253	0.8490	0.8696	**0.8870**
Load and energy calculations	0.7781	0.7910	0.8043	**0.8519**	0.8473
HVAC design	0.7949	0.8338	0.8070	**0.8633**	0.8606
Building envelope	0.7809	0.7985	0.8338	**0.8665**	0.8438
Materials	0.7407	0.7897	0.8201	0.8224	**0.8528**
General	0.7956	0.8316	0.8118	**0.8346**	0.8338
F1 score (positive)	GLM-std	GPT 3.5-Turbo	GPT 4-Turbo	RAG-GLM	RAG-GPT 3.5
Principles	0.8187	0.8655	0.8703	**0.9165**	0.9105
Indoor environmental quality	0.8380	0.8613	0.8764	0.8946	**0.9047**
Load and energy calculations	0.8190	0.8370	0.8388	**0.8781**	0.8719
HVAC design	0.8357	0.8740	0.8428	**0.8942**	0.8896
Building envelope	0.8337	0.8524	0.8745	**0.9013**	0.8812
Materials	0.8115	0.8558	0.8706	0.8725	**0.8927**
General	0.8529	0.8825	0.8632	**0.8826**	0.8795
F1 score (negative)	GLM-std	GPT 3.5-Turbo	GPT 4-Turbo	RAG-GLM	RAG-GPT 3.5
Principles	0.6284	0.6919	0.7225	**0.8035**	0.8013
Indoor environmental quality	0.7492	0.7641	0.8061	0.8290	**0.8610**
Load and energy calculations	0.7131	0.7089	0.7512	**0.8113**	0.8110
HVAC design	0.7273	0.7559	0.7500	0.8068	**0.8109**
Building envelope	0.6790	0.6825	0.7537	**0.7938**	0.7721
Materials	0.5843	0.6121	0.7050	0.7077	**0.7658**
General	0.6651	0.7030	0.6981	0.7198	**0.7322**
Balanced Accuracy	GLM-std	GPT 3.5-Turbo	GPT 4-Turbo	RAG-GLM	RAG-GPT 3.5
Principles	0.7415	0.7856	0.8130	0.8646	**0.8697**
Indoor environmental quality	0.7958	0.8084	0.8427	0.8605	**0.8912**
Load and energy calculations	0.7680	0.7676	0.7992	0.8493	**0.8503**
HVAC design	0.7946	0.8132	0.8149	0.8551	**0.8623**
Building envelope	0.7702	0.7699	0.8285	**0.8545**	0.8446
Materials	0.7176	0.7327	0.8096	0.8112	**0.8614**
General	0.7911	0.8109	0.8205	0.8294	**0.8473**

Appendix D:. Detailed information of the case reports and tasks

We collected a total of 63 audit reports, all corresponding to buildings located in Shanghai, China. All the reports are written in Chinese. 43 of these reports are in paper format, which cannot be automated at this time, but can help us understand the data characteristics of the audit reports. Among the reports in electronic form, we have selected four as case studies. Table D1 lists basic information about the four case buildings, the number of information items in the manually labeled metadata, and the number of original recommendations provided in the report. As shown in the table, the four reports vary in terms of building function and level of detail of content. In addition, the audit units that completed these reports differed (we are not permitted to disclose the audit unit information). The data organization characteristics of these reports vary, and the metadata information is scattered throughout different parts of the reports. Figure A1 (b)&(c) in Appendix A shows segments of two reports (translated into English for ease of reading). As can be seen from the figure, the two reports have different levels of detail in the information about the heat and cold source equipment, and the structure of the tables representing the heat and cold sources is different. Extracting metadata automatically and accurately with the same framework poses a challenge due to these variations. Table D2 shows the EPIs available in the performance diagnosis of each case building and the estimates required in the calculations due to missing relevant data in the report. The table demonstrates that the EPI calculation tasks applied and the estimates required for different case buildings vary, which intuitively reflects the data heterogeneity and missing data across the different audit reports. Such situations are widespread across the audit reports we have.

Table D1
Overview of the case buildings and the corresponding reports.

Case	#0	#1	#2	#3
Primary use type	Office	Office and mall	Mall	Office and mall
Location	Shanghai, China	Shanghai, China	Shanghai, China	Shanghai, China
Gross area (m^2)	37,000	77,543	32,144	67,477
Height (m)/floor	98/28	110/22	Not available/8	Not available/24
Number of metadata information items	85	177	64	154
Number of recommendations from the report	11	8	3	6

Table D2
Applicable EPI calculations for each case building.

Case	EPI	Need estimation	Estimated value	Note
#0	EUI	No	–	Need unit conversion (tce to kWh)
	ACL	Yes	Configured cooling load	
	WTF	Yes	Accumulated cooling load; energy consumption of the pumps	For chilled water system and cool water system, respectively.
	EER	–	–	Not enough data provided
#1	EUI	No	–	
	ACL	Yes	Configured cooling load	
	WTF	Yes	Accumulated cooling load; energy consumption of the pumps; annual operating hour	For cool water system. Other systems don't have enough data.
	EER	–	–	Not enough data provided
#2	EUI	No	–	Need unit conversion (tce to kWh)
	ACL	No	–	
	WTF	Yes	Accumulated cooling load; energy consumption of the pumps; annual operating hour	For chilled water system and cool water system, respectively.
	EER	Yes	Accumulated cooling load; energy consumption of the terminal units; annual operating hour	
#3	EUI	No	–	
	ACL	No	–	
	WTF	Yes	Accumulated cooling load; energy consumption of the pumps	For chilled water system and cool water system, respectively.
	EER	–	–	Not enough data provided

We designed a scoring metric for the performance diagnostic phase, and Table D3 illustrates the scoring elements. The scoring elements consisted of plan reasonableness (broken down into four score points), information extraction accuracy, calculation accuracy, ability to follow instructions (format), and accuracy of the final result. This indicator is designed to assess the framework's ability to plan, calculate, and extract information in the face of diverse tasks.

Table D3
Scoring components for the performance diagnosis stage.

	Evaluation rules	Fraction
Planning	• Correctly determine that the current indicator should be calculated (25 %) • Correctly determine the estimation steps (25 %) • Correctly plan the information extraction (25 %) • Correctly plan the calculation process (25 %)	0.4
Extraction	$score = \frac{The\ number\ of\ correct\ extraction}{The\ number\ of\ extraction\ needed}$	0.2
Calculation	$score = \frac{The\ number\ of\ correct\ calculation}{The\ number\ of\ calculation\ needed}$	0.2
Instruction following	• Correctly follow the output format	0.1
Final result	• Correct final results (50 %) • Correct diagnostic conclusions (50 %)	0.1

Appendix E:. Technical supplement for framework development

In our framework, agents are specialized with roles and tasks through structured prompts. These prompts are categorized into two main types based on the prompting strategy used: prompts that use only the Chain-of-Thought (CoT) strategy (marked as CoT-only prompts), and prompts that use both the CoT and ReAct strategies (marked as CoT-ReAct prompts). The CoT-only prompts apply to tasks that do not require dynamic planning or interaction with tools, while the CoT-ReAct prompts are used for tasks that require real-time selection and interaction with tools based on task completion. In Fig. 3(a), the type of prompt used by different agents is represented by the form of different icons. An agent using a CoT-only prompt is indicated by an icon with only CoT characters, while an agent using a CoT-ReAct prompt is represented by an icon with both CoT and ReAct characters. Both CoT-only and CoT-ReAct prompts are highly structured and are constructed using the components listed in Table E1.

The implementation of multiple agent interaction is crucial for multi-agent system development. In our framework, the form of multiple agent interaction primarily takes the shape of a one-way flow of tasks in the form of a pipeline. As outlined in Section 4.1.2, during each stage, the Planner agent divides a complex task into several parallel tasks. Then, several Researcher agents accomplish each of these parallel tasks. The Advisor agent summarizes the results of all parallel tasks. In practice, the Planner agent distributes the task into multiple subtasks, thereby initiating multiple pipeline branches. Within each pipeline branch, the task is completed by the Researcher agents. Finally, the Advisor agent summarizes the results of multiple pipeline branches. The interaction within each pipeline branch is facilitated by a CrewAI sequential process, while the interaction between the Planner, Researcher, and Advisor agents is realized through writing and reading files. The sequential process in CrewAI implements a dynamic pipeline workflow, progressing through a predefined list of tasks. Task execution follows the order in the task list, with the output of one task serving as context for the next. Consequently, within every pipeline branch, the downstream Researcher agent can continue the pipeline with the necessary information provided by the upstream Researcher agent. Meanwhile, the Planner, Researcher, and Advisor agents write out to files and read in key results (such as tasks planned by the Planner agent and task completion results for each Researcher agent) to exchange necessary information and cooperate to complete the entire task of the current stage.

Table E1
Components of the structured prompt.

Component name	Function	Example	Use in
Role and goal	Describe the roles and goals the agent needs to play.	You are a helpful HVAC expert, specialized in energy audit.	CoT-only, CoT-ReAct
Task description	Describe the current task that the agent needs to accomplish.	You will be given a brief description of a building and its mechanical and electrical system. The given information will start and end with the delimiter ‘‘‘. Please provide 3–5 direction proposals that you believe are appropriate for the further energy savings potential analysis/ energy retrofit program for this building.	CoT-only, CoT-ReAct
Background information for the task	Describe the necessary background information such as relevant domain knowledge (e.g., commonly used retrofit methods in China), and outputs from the upstream task (e.g., the building metadata, the energy diagnosis results).	The given information is: ‘‘‘{building_metadata}‘‘‘Here are some reference common retrofit method direction (not detailed) : − Building envelope:* Building fabric insulation (i.e. roof, wall, etc.)* Windows retrofits (i.e. multiple glazing, low-E coating, etc.) * Add shading systems − HVAC system:* Using energy-storage cooling/heating source (i.e. water storage, ice storage, eutectic salt storage, etc.)* Using ground source heat pumps (i.e. water source, soil source, etc.)* Free cooling (i.e. fresh air free cooling, water-side free cooling, etc.) * Heat recovery system (i.e. exhaust air heat recovery, condensing heat recovery, etc.) * Large temperature difference water distribution * Variable-frequency control of pumps and fans − Lighting system: * Lighting device upgrade * Daylighting − BA system: * Chilled/cooling water pump frequency and operating number optimization* Chiller and cooling tower operating number optimization* Fresh air volume optimization	CoT-only, CoT-ReAct
Step guide for task accomplishment	Describe the path to completion of the current task step-by-step. This is the key component to realizing the CoT strategy.	Please provide the proposals following these steps, 1. Based on the basic information about the building, suggest directions where further information of the building is needed to further determine the energy savings potential.2. Suggest some retrofit methods according to the directions. You can refer to common energy efficiency retrofit methods for every direction. Then you can select the proper methods from the common methods for the building. Notice that the methods you provide are not limited to the common methods.4. Prepare the direction proposal with number and the mark “[System/Object/Equipment]: ” at the beginning of the proposal. For example, “	CoT-only, CoT-ReAct
Additional requirements for task accomplishment	Additional requirements to supplement the completion of tasks.	When you are providing the proposals,1. Remember the proposal should not fall outside the scope. The scope of the building audit included: building envelope (including wall, roof, window, and shading), HVAC system (including source, recovery, and distribution), lighting system, building automatic control (BA) system, BEMS system, substation system, etc. 2. Remember that the retrofit method should meet ASHRAE’s recommendations for energy audit Level I.3. Avoid overly brief descriptions.4. The energy diagnosis results may be very helpful.	CoT-only, CoT-ReAct
Output format requirement	Describes the format requirements for the output result. If the task requires JSON output, specify the available JSON fields and what each field should contain. If the task output is regular text, specify the text segments and what each segment needs to contain.	Please provide the proposals with number as follow: 1. [System/Object/Equipment]: xxx 2. [System/Object/Equipment]: xxx 3. [System/Object/Equipment]: xxx	CoT-only, CoT-ReAct
Examples	Provide one or more sample task outputs.	1. [Lighting System]: Need to know more about the lighting energy. Retrofit all fixtures to energy efficient fixtures“.	CoT-only, CoT-ReAct
Tool list	Describe the tools available to the agent together with what every tool can do and how to use them.	You ONLY have access to the following tools, and should NEVER make up tools that are not listed here: Search_report: useful for when you need to know some information about the target building. The input to this tool should be a question directly. python_repl_ast: Python Code Interpreter Tool. ALWAYS PRINT VARIABLES TO SHOW THE VALUE. The environment is long running and exists across multiple executions. You must send the whole script every time and print your outputs. When printing results, print all results on the same line.Script should be pure python code that can be evaluated. It should be in python format NOT markdown. Remember that the key of the input dictionary is ’query’. The code should NOT be wrapped in backticks. Here is a example of input:{ ’query’:“‘’# begin’’print(’hello world!’) ’“‘}All python packages including requests, matplotlib, scipy, numpy, pandas, etc are available. Create and display chart using ‘plt.show() ‘.Search_system_energy_efficiency_evaluation_indicators_and_methods: useful when you have energy data of an air conditioning system or its equipment and want to know how to diagnose the energy efficiency of the system. The input to this tool should be a question directly.Search_energy_audit_code: useful when you have collected some basic data from a building and want to understand how it can be further processed for audit or diagnose. The input to this tool should be a question directly.	CoT-ReAct

References

[1] World Energy Outlook, – Analysis [WWW Document], IEA, n.d, 2023 https://www.iea.org/reports/world-energy-outlook-2023 (accessed 1.4.24.

[2] H. Wang, P. Xu, X. Lu, D. Yuan, Methodology of comprehensive building energy performance diagnosis for large commercial buildings at multiple levels, Appl. Energy 169 (2016) 14–27, https://doi.org/10.1016/j.apenergy.2016.01.054.

[3] Y. Li, H. Du, S.B. Kumaraswamy, Case-based reasoning approach for decision-making in building retrofit: A review, Build. Environ. 248 (2024) 111030, https://doi.org/10.1016/j.buildenv.2023.111030.

[4] Morris, M.R., Sohl-dickstein, J., Fiedel, N., Warkentin, T., Dafoe, A., Faust, A., Farabet, C., Legg, S., 2023. Levels of AGI: Operationalizing Progress on the Path to AGI. https://doi.org/10.48550/arXiv.2311.02462.

[5] S. Legg, M. Hutter, Universal Intelligence: A Definition of Machine Intelligence, Mind. Mach. 17 (2007) 391–444, https://doi.org/10.1007/s11023-007-9079-x.

[6] M. Wooldridge, N.R. Jennings, Intelligent agents: theory and practice, Knowl. Eng. Rev. 10 (1995) 115–152, https://doi.org/10.1017/S0269888900008122.

[7] Choi, S., Jain, R., Emami, P., Wadsack, K., Ding, F., Sun, H., Gruchalla, K., Hong, J., Zhang, H., Zhu, X., Kroposki, B., 2024. eGridGPT: Trustworthy AI in the Control Room (No. NREL/TP-5D00-87740, 2352232, MainId:88515). https://doi.org/10.2172/2352232.

[8] H. Sha, P. Xu, Z. Yang, Y. Chen, J. Tang, Overview of computational intelligence for building energy system design, Renew. Sustain. Energy Rev. 108 (2019) 76–90, https://doi.org/10.1016/j.rser.2019.03.018.

[9] S. Baek, W. Jung, S.H. Han, A critical review of text-based research in construction: Data source, analysis method, and implications, Autom. Constr. 132 (2021) 103915, https://doi.org/10.1016/j.autcon.2021.103915.

[10] Searle, J.R., 2007. What is language: some preliminary remarks, in: John Searle's Philosophy of Language: Force, Meaning and Mind. Cambridge University Press, pp. 15–46. https://doi.org/10.1017/CBO9780511619489.002.

[11] A.M. Turing, Computing machinery and intelligence, Mind LIX 433–460 (1950), https://doi.org/10.1093/mind/LIX.236.433.

[12] Y. Bengio, R. Ducharme, P. Vincent, C. Janvin, A neural probabilistic language model, J. Mach. Learn. Res. 3 (2003) 1137–1155.

[13] Introducing ChatGPT [WWW Document], n.d. URL https://openai.com/blog/chatgpt (accessed 11.9.23).

[14] OpenAI, 2023. GPT-4 Technical Report. https://doi.org/10.48550/arXiv.2303.08774.

[15] Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y.T., Li, Y., Lundberg, S., Nori, H., Palangi, H., Ribeiro, M.T., Zhang, Y., 2023. Sparks of Artificial General Intelligence: Early experiments with GPT-4. https://doi.org/10.48550/arXiv.2303.12712.

[16] Qian, C., Cong, X., Liu, W., Yang, C., Chen, W., Su, Y., Dang, Y., Li, J., Xu, J., Li, D., Liu, Z., Sun, M., 2023. Communicative Agents for Software Development. https://doi.org/10.48550/arXiv.2307.07924.

[17] X. Yuan, L. Heikari, J. Hirvonen, Y. Liang, M. Virtanen, R. Kosonen, Y. Pan, System modelling and optimization of a low temperature local hybrid energy system based on solar energy for a residential district, Energ. Conver. Manage. 267 (2022) 115918, https://doi.org/10.1016/j.enconman.2022.115918.

[18] Sumers, T.R., Yao, S., Narasimhan, K., Griffiths, T.L., 2024. Cognitive Architectures for Language Agents. https://doi.org/10.48550/arXiv.2309.02427.

[19] Z. Chen, Y. Chen, T. Xiao, H. Wang, P. Hou, A novel short-term load forecasting framework based on time-series clustering and early classification algorithm, Energ. Buildings 251 (2021) 111375, https://doi.org/10.1016/j.enbuild.2021.111375.

[20] C. Fan, D. Yan, F. Xiao, A. Li, J. An, X. Kang, Advanced data analytics for enhancing building performances: From data-driven to big data-driven approaches, Build. Simul. 14 (2021) 3–24, https://doi.org/10.1007/s12273-020-0723-1.

[21] T. Xiao, P. Xu, R. He, H. Sha, Status quo and opportunities for building energy prediction in limited data Context—Overview from a competition, Appl. Energy 305 (2022) 117829, https://doi.org/10.1016/j.apenergy.2021.117829.

[22] X. Fang, G. Gong, G. Li, L. Chun, P. Peng, X. Shi, Transferability investigation of a Sim2Real deep transfer learning framework for cross-building energy prediction, Energ. Buildings 287 (2023) 112968, https://doi.org/10.1016/j.enbuild.2023.112968.

[23] B. Yang, S. Yang, X. Zhu, M. Qi, H. Li, Z. Lv, X. Cheng, F. Wang, Computer Vision Technology for Monitoring of Indoor and Outdoor Environments and HVAC Equipment: A Review, Sensors 23 (2023) 6186, https://doi.org/10.3390/s23136186.

[24] R. He, P. Xu, Z. Chen, W. Luo, Z. Su, J. Mao, A non-intrusive approach for fault detection and diagnosis of water distribution systems based on image sensors, audio sensors and an inspection robot, Energ. Buildings 243 (2021) 110967, https://doi.org/10.1016/j.enbuild.2021.110967.

[25] X. Yuan, Y. Pan, J. Yang, W. Wang, Z. Huang, Study on the application of reinforcement learning in the operation optimization of HVAC system, Build. Simul. 14 (2021) 75–87, https://doi.org/10.1007/s12273-020-0602-9.

[26] Z. Wang, T. Hong, Reinforcement learning for building controls: The opportunities and challenges, Appl. Energy 269 (2020) 115036, https://doi.org/10.1016/j.apenergy.2020.115036.

[27] C. Zhang, J. Zhang, Y. Zhao, J. Lu, Automated data mining framework for building energy conservation aided by generative pre-trained transformers (GPT), Energ. Buildings 113877 (2024), https://doi.org/10.1016/j.enbuild.2023.113877.

[28] A. Rysanek, Z. Nagy, C. Miller, A.D. Dilsiz, How good is the advice from ChatGPT for building science? Comparison of four scenarios, J. Phys.: Conf. Ser. 2600 (2023) 082006, https://doi.org/10.1088/1742-6596/2600/8/082006.

[29] G. Jiang, Z. Ma, L. Zhang, J. Chen, EPlus-LLM: A large language model-based computing platform for automated building energy modeling, Appl. Energy 367 (2024) 123431, https://doi.org/10.1016/j.apenergy.2024.123431.

[30] Zhang, L., Chen, Z., Ford, V., 2024. Advancing Building Energy Modeling with Large Language Models: Exploration and Case Studies. https://doi.org/10.48550/arXiv.2402.09579.

[31] Song, L., Zhang, C., Zhao, L., Bian, J., 2023. Pre-Trained Large Language Models for Industrial Control. https://doi.org/10.48550/arXiv.2308.03028.

[32] J. Zheng, M. Fischer, Dynamic prompt-based virtual assistant framework for BIM information search, Autom. Constr. 155 (2023) 105067, https://doi.org/10.1016/j.autcon.2023.105067.

[33] Bottaccioli, L., Aliberti, A., Ugliotti, F., Patti, E., Osello, A., Macii, E., Acquaviva, A., 2017. Building Energy Modelling and Monitoring by Integration of IoT Devices and Building Information Models, in: 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC). Presented at the 2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC), pp. 914–922. https://doi.org/10.1109/COMPSAC.2017.75.

[34] Zhao, W.X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., Liu, P., Nie, J.-Y., Wen, J.-R., 2023. A Survey of Large Language Models. https://doi.org/10.48550/arXiv.2303.18223.

[35] Zhang, L., Chen, Z., 2023. Opportunities and Challenges of Applying Large Language Models in Building Energy Efficiency and Decarbonization Studies: An Exploratory Overview. https://doi.org/10.48550/arXiv.2312.11701.

[36] P. Ghimire, K. Kim, M. Acharya, Opportunities and Challenges of Generative AI in Construction Industry: Focusing on Adoption of Text-Based Models, Buildings 14 (2024) 220, https://doi.org/10.3390/buildings14010220.

[37] J. Lu, X. Tian, C. Zhang, Y. Zhao, J. Zhang, W. Zhang, C. Feng, J. He, J. Wang, F. He, Evaluation of large language models (LLMs) on the mastery of knowledge and skills in the heating, ventilation and air conditioning (HVAC) industry, Energy and Built Environment (2024), https://doi.org/10.1016/j.enbenv.2024.03.010.

[38] Y. Lian, Research on energy-saving diagnosis method of air-conditioning system in large public buildings based on engineering practice (master), Trans. Tianjin Univ. (2020).

[39] X. Zhou, Y. Mei, L. Liang, Z. Fan, J. Yan, D. Pan, A dynamic energy benchmarking methodology on room level for energy performance evaluation, Journal of Building Engineering 42 (2021) 102837, https://doi.org/10.1016/j.jobe.2021.102837.

[40] L. Chen, G. Li, J. Liu, L. Liu, C. Zhang, J. Gao, C. Xu, X. Fang, Z. Yao, Fault diagnosis for cross-building energy systems based on transfer learning and model interpretation, Journal of Building Engineering 109424 (2024), https://doi.org/10.1016/j.jobe.2024.109424.

[41] W. Chung, Y.V. Hui, Y.M. Lam, Benchmarking the energy efficiency of commercial buildings, Appl. Energy 83 (2006) 1–14, https://doi.org/10.1016/j.apenergy.2004.11.003.

[42] S.E. Lee, P. Rajagopalan, Building energy efficiency labeling programme in Singapore, Energy Policy 36 (2008) 3982–3992, https://doi.org/10.1016/j.enpol.2008.07.014.

[43] Benchmark Your Building With Portfolio Manager | ENERGY STAR [WWW Document], n.d. URL https://www.energystar.gov/buildings/benchmark (accessed 5.8.24).

[44] Standardization Administration of China (SAC), 2007. Economic operation of air-conditioning systems GB/T 17981-2007.

[45] H. Li, J.E. Braun, An overall performance index for characterizing the economic impact of faults in direct expansion cooling equipment, Int. J. Refrig 30 (2007) 299–310, https://doi.org/10.1016/j.ijrefrig.2006.07.026.

[46] A. Taal, L. Itard, P&ID-based symptom detection for automated energy performance diagnosis in HVAC systems, Autom. Constr. 119 (2020) 103344, https://doi.org/10.1016/j.autcon.2020.103344.

[47] Z. Ma, P. Cooper, D. Daly, L. Ledo, Existing building retrofits: Methodology and state-of-the-art, Energy and Buildings, Cool Roofs, Cool Pavements, Cool Cities, and Cool World 55 (2012) 889–902, https://doi.org/10.1016/j.enbuild.2012.08.018.

[48] T. Liu, G. Ma, D. Wang, X. Pan, Intelligent green retrofitting of existing buildings based on case-based reasoning and random forest, Autom. Constr. 162 (2024) 105377, https://doi.org/10.1016/j.autcon.2024.105377.

[49] X. Zhao, Y. Tan, L. Shen, G. Zhang, J. Wang, Case-based reasoning approach for supporting building green retrofit decisions, Build. Environ. 160 (2019) 106210, https://doi.org/10.1016/j.buildenv.2019.106210.

[50] Mikolov, T., Chen, K., Corrado, G. s, Dean, J., 2013. Efficient Estimation of Word Representations in Vector Space. Proceedings of Workshop at ICLR 2013.

[51] Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J., 2013. Distributed representations of words and phrases and their compositionality, in: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, NIPS'13. Curran Associates Inc., Red Hook, NY, USA, pp. 3111–3119.

[52] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is All you Need. Advances in Neural Information Processing Systems, Curran Associates Inc., 2017.

[53] T. Lin, Y. Wang, X. Liu, X. Qiu, A survey of transformers, AI Open 3 (2022) 111–132, https://doi.org/10.1016/j.aiopen.2022.10.001.

[54] Kolen, J.F., Kremer, S.C., 2001. Gradient Flow in Recurrent Nets: The Difficulty of Learning LongTerm Dependencies, in: A Field Guide to Dynamical Recurrent

Networks. Presented at the A Field Guide to Dynamical Recurrent Networks, IEEE, pp. 237–243. https://doi.org/10.1109/9780470544037.ch14.
[55] S. Hochreiter, J. Schmidhuber, Long Short-term Memory, Neural Comput. 9 (1997) 1735–1780, https://doi.org/10.1162/neco.1997.9.8.1735.
[56] Chung, J., Gulcehre, C., Cho, K., Bengio, Y., 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. NIPS 2014 Workshop on Deep Learning, December 2014.
[57] Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y., 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation, in: Moschitti, A., Pang, B., Daelemans, W. (Eds.), Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Presented at the EMNLP 2014, Association for Computational Linguistics, Doha, Qatar, pp. 1724–1734. https://doi.org/10.3115/v1/D14-1179.
[58] Sutskever, I., Vinyals, O., Le, Q.V., 2014. Sequence to sequence learning with neural networks, in: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, NIPS'14. MIT Press, Cambridge, MA, USA, pp. 3104–3112.
[59] S.J. Pan, Q. Yang, A Survey on Transfer Learning, IEEE Trans. Knowl. Data Eng. 22 (2010) 1345–1359, https://doi.org/10.1109/TKDE.2009.191.
[60] Howard, J., Ruder, S., 2018. Universal Language Model Fine-tuning for Text Classification, in: Gurevych, I., Miyao, Y. (Eds.), Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Melbourne, Australia, pp. 328–339. https://doi.org/10.18653/v1/P18-1031.
[61] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai, X. Huang, Pre-trained models for natural language processing: A survey, Sci. China Technol. Sci. 63 (2020) 1872–1897, https://doi.org/10.1007/s11431-020-1647-3.
[62] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, E.H. Chi, T. Hashimoto, O. Vinyals, P. Liang, J. Dean, W. Fedus, Emergent Abilities of Large Language Models. Transactions on Machine Learning, Research (2022).
[63] T. Brown, B. Mann, N. Ryder, M. Subbiah, J.D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, D. Amodei, Language Models are Few-Shot Learners, Adv. Neural Inf. Proces. Syst. 33 (2020) 1877–1901.
[64] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., n.d. Language Models are Unsupervised Multitask Learners. OpenAI blog. https://openai.com/blog/better-language-models.
[65] Shanahan, M., 2023. Talking About Large Language Models. https://doi.org/10.48550/arXiv.2212.03551.
[66] Yang, J., Jin, H., Tang, R., Han, X., Feng, Q., Jiang, H., Yin, B., Hu, X., 2023. Harnessing the Power of LLMs in Practice: A Survey on ChatGPT and Beyond. https://doi.org/10.48550/arXiv.2304.13712.
[67] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H.W. Chung, C. Sutton, S. Gehrmann, P. Schuh, K. Shi, S. Tsvyashchenko, J. Maynez, A. Rao, P. Barnes, Y. Tay, N. Shazeer, V. Prabhakaran, E. Reif, N. Du, B. Hutchinson, R. Pope, J. Bradbury, J. Austin, M. Isard, G. Gur-Ari, P. Yin, T. Duke, A. Levskaya, S. Ghemawat, S. Dev, H. Michalewski, X. Garcia, V. Misra, K. Robinson, L. Fedus, D. Zhou, D. Ippolito, D. Luan, H. Lim, B. Zoph, A. Spiridonov, R. Sepassi, D. Dohan, S. Agrawal, M. Omernick, A.M. Dai, T. S. Pillai, M. Pellat, A. Lewkowycz, E. Moreira, R. Child, O. Polozov, K. Lee, Z. Zhou, X. Wang, B. Saeta, M. Diaz, O. Firat, M. Catasta, J. Wei, K. Meier-Hellstern, D. Eck, J. Dean, S. Petrov, N. Fiedel, PaLM: Scaling language modeling with pathways, J. Mach. Learn. Res. 24 (2023) 1–113.
[68] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., Lample, G., 2023. LLaMA: Open and Efficient Foundation Language Models.
[69] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, J. Mach. Learn. Res. 21 (2020), 140:5485–140:5551.
[70] Zeng, A., Liu, X., Du, Z., Wang, Z., Lai, H., Ding, M., Yang, Z., Xu, Y., Zheng, W., Xia, X., Tam, W.L., Ma, Z., Xue, Y., Zhai, J., Chen, W., Zhang, P., Dong, Y., Tang, J., 2023. GLM-130B: An Open Bilingual Pre-trained Model. https://doi.org/10.48550/arXiv.2210.02414.
[71] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P.F. Christiano, J. Leike, R. Lowe, Training language models to follow instructions with human feedback, Adv. Neural Inf. Proces. Syst. 35 (2022) 27730–27744.
[72] Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., Schulman, J., 2021. Training Verifiers to Solve Math Word Problems.
[73] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E.H. Chi, Q.V. Le, D. Zhou, Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. Presented at the Advances in Neural Information Processing Systems, 2022.
[74] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, G. Neubig, Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing, 195:1–195:35, ACM Comput. Surv. 55 (2023), https://doi.org/10.1145/3560815.
[75] DLAI - Learning Platform Beta [WWW Document], n.d. URL https://learn.deeplearning.ai/chatgpt-prompt-eng/lesson/1/introduction (accessed 11.22.23).
[76] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K.R. Narasimhan, Y. Cao, ReAct: Synergizing reasoning and acting in language models. In: the Eleventh International Conference on Learning Representations, ICLR 2023, 2023.
[77] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., Kiela, D., 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks, in: Advances in Neural Information Processing Systems. Curran Associates, Inc., pp. 9459–9474.
[78] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y.J. Bang, A. Madotto, P. Fung, Survey of Hallucination in Natural Language Generation, 248:1–248:38, ACM Comput. Surv. 55 (2023), https://doi.org/10.1145/3571730.
[79] Z. Xia, H. Guan, Z. Qi, P. Xu, Multi-Zone Infection Risk Assessment Model of Airborne Virus Transmission on a Cruise Ship Using CONTAM, Buildings 13 (2023) 2350, https://doi.org/10.3390/buildings13092350.
[80] Y. Li, Z. Li, K. Zhang, R. Dan, S. Jiang, Y. Zhang, ChatDoctor: A Medical Chat Model Fine-Tuned on a Large Language Model Meta-AI (LLaMA) Using Medical Domain Knowledge, Cureus (2023), https://doi.org/10.7759/cureus.40895.
[81] Feng, Z., Ma, W., Yu, W., Huang, L., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., liu, T., 2023. Trends in Integration of Knowledge and Large Language Models: A Survey and Taxonomy of Methods, Benchmarks, and Applications. https://doi.org/10.48550/arXiv.2311.05876.
[82] J. Li, X. Cheng, X. Zhao, J.-Y. Nie, J.-R. Wen, in: HaluEval: A Large-Scale Hallucination Evaluation Benchmark for Large Language Models, Association for Computational Linguistics, Singapore, 2023, pp. 6449–6464.
[83] O. Ovadia, M. Brief, M. Mishaeli, O. Elisha, Fine-Tuning or Retrieval? Comparing Knowledge Injection in LLMs. Https:// (2023) https://doi.org/10.48550/arXiv.2312.05934.
[84] Cui, J., Li, Z., Yan, Y., Chen, B., Yuan, L., 2023. ChatLaw: Open-Source Legal Large Language Model with Integrated External Knowledge Bases. https://doi.org/10.48550/arXiv.2306.16092.
[85] L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin, W.X. Zhao, Z. Wei, J.-R. Wen, A Survey on Large Language Model based Autonomous Agents, Front. Comput. Sci. 18 (2024) 186345, https://doi.org/10.1007/s11704-024-40231-1.
[86] Xi, Z., Chen, W., Guo, X., He, W., Ding, Y., Hong, B., Zhang, M., Wang, J., Jin, S., Zhou, E., Zheng, R., Fan, X., Wang, X., Xiong, L., Zhou, Y., Wang, W., Jiang, C., Zou, Y., Liu, X., Yin, Z., Dou, S., Weng, R., Cheng, W., Zhang, Q., Qin, W., Zheng, Y., Qiu, X., Huang, X., Gui, T., 2023. The Rise and Potential of Large Language Model Based Agents: A Survey. https://doi.org/10.48550/arXiv.2309.07864.
[87] Guo, T., Chen, X., Wang, Y., Chang, R., Pei, S., Chawla, N.V., Wiest, O., Zhang, X., 2024. Large Language Model based Multi-Agents: A Survey of Progress and Challenges. https://doi.org/10.48550/arXiv.2402.01680.
[88] Xie, J., Zhang, K., Chen, J., Zhu, T., Lou, R., Tian, Y., Xiao, Y., Su, Y., 2024. TravelPlanner: A Benchmark for Real-World Planning with Language Agents. https://doi.org/10.48550/arXiv.2402.01622.
[89] I. Gur, H. Furuta, A.V. Huang, M. Safdari, Y. Matsuo, D. Eck, A. Faust, A real-world WebAgent with planning, long context understanding. And Program Synthesis, in: the Twelfth International Conference on Learning Representations, 2024.
[90] Yang, K., Swope, A., Gu, A., Chalamala, R., Song, P., Yu, S., Godil, S., Prenger, R. J., Anandkumar, A., 2023. LeanDojo: Theorem proving with retrieval-augmented language models, in: Oh, A., Naumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (Eds.), Advances in Neural Information Processing Systems. Curran Associates, Inc., pp. 21573–21612. https://doi.org/10.48550/arXiv.2306.15626.
[91] D.A. Boiko, R. MacKnight, B. Kline, G. Gomes, Autonomous chemical research with large language models, Nature 624 (2023) 570–578, https://doi.org/10.1038/s41586-023-06792-0.
[92] Li, G., Hammoud, H.A.A.K., Itani, H., Khizbullin, D., Ghanem, B., 2023. CAMEL: Communicative Agents for "Mind" Exploration of Large Language Model Society. https://doi.org/10.48550/arXiv.2303.17760.
[93] Sun, C., Han, J., Deng, W., Wang, X., Qin, Z., Gould, S., 2023. 3D-GPT: Procedural 3D Modeling with Large Language Models.
[94] Park, J.S., O'Brien, J.C., Cai, C.J., Morris, M.R., Liang, P., Bernstein, M.S., 2023. Generative Agents: Interactive Simulacra of Human Behavior.
[95] Chen, M., Tworek, J., Jun, H., Yuan, Q., Pinto, H.P. de O., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., Ray, A., Puri, R., Krueger, G., Petrov, M., Khlaaf, H., Sastry, G., Mishkin, P., Chan, B., Gray, S., Ryder, N., Pavlov, M., Power, A., Kaiser, L., Bavarian, M., Winter, C., Tillet, P., Such, F.P., Cummings, D., Plappert, M., Chantzis, F., Barnes, E., Herbert-Voss, A., Guss, W.H., Nichol, A., Paino, A., Tezak, N., Tang, J., Babuschkin, I., Balaji, S., Jain, S., Saunders, W., Hesse, C., Carr, A.N., Leike, J., Achiam, J., Misra, V., Morikawa, E., Radford, A., Knight, M., Brundage, M., Murati, M., Mayer, K., Welinder, P., McGrew, B., Amodei, D., McCandlish, S., Sutskever, I., Zaremba, W., 2021. Evaluating Large Language Models Trained on Code. https://doi.org/10.48550/arXiv.2107.03374.
[96] Ding, S., Chen, X., Fang, Y., Liu, W., Qiu, Y., Chai, C., 2023. DesignGPT: Multi-Agent Collaboration in Design. https://doi.org/10.48550/arXiv.2311.11591.
[97] M. Bran, A., Cox, S., Schilter, O., Baldassari, C., White, A.D., Schwaller, P., 2024. Augmenting large language models with chemistry tools. Nat Mach Intell 6, 525–535. https://doi.org/10.1038/s42256-024-00832-8.
[98] Zheng, T., Zhang, G., Shen, T., Liu, X., Lin, B.Y., Fu, J., Chen, W., Yue, X., 2024. OpenCodeInterpreter: Integrating Code Generation with Execution and Refinement. https://doi.org/10.48550/arXiv.2402.14658.
[99] P. Xu, Y. Shen, J. Hua, Effectiveness of energy retrofit methods in public buildings in China, Heating Ventilating & Air (2012). Conditioning.
[100] New and improved embedding model [WWW Document], n.d. URL https://openai.com/blog/new-and-improved-embedding-model (accessed 11.23.23).

[101] Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, H., 2023. Retrieval-Augmented Generation for Large Language Models: A Survey. https://doi.org/10.48550/arXiv.2312.10997.
[102] Xie, J., Zhang, K., Chen, J., Lou, R., Su, Y., 2023. Adaptive Chameleon or Stubborn Sloth: Revealing the Behavior of Large Language Models in Knowledge Conflicts. https://doi.org/10.48550/arXiv.2305.13300.
[103] Zhang, L., Chen, J., Zou, J., 2023. Taxonomy, Semantic Data Schema, and Schema Alignment for Open Data in Urban Building Energy Modeling. https://doi.org/10.48550/arXiv.2311.08535.
[104] Shanghai Municipal Commission of Housing and Urban-Rural Development, 2020. Energy audit standards for public buildings DG/TJ08-2114-2020.
[105] G. Li, P. Wang, W. Ke, Revisiting Large Language Models as Zero-shot Relation Extractors, in: H. Bouamor, J. Pino, K. Bali (Eds.), Findings of the Association for Computational Linguistics: EMNLP 2023, Association for Computational Linguistics, Singapore, 2023, pp. 6877–6892, https://doi.org/10.18653/v1/2023.findings-emnlp.459.
[106] N. Long, K. Fleming, C. CaraDonna, C. Mosiman, BuildingSync: A schema for commercial building energy audit data exchange, Developments in the Built Environment 7 (2021) 100054, https://doi.org/10.1016/j.dibe.2021.100054.
[107] Chase, H., 2022. LangChain. https://github.com/langchain-ai/langchain.
[108] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, J. Steinhardt, Measuring massive multitask language understanding. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, 2021.
[109] Zhong, W., Cui, R., Guo, Y., Liang, Y., Lu, S., Wang, Y., Saied, A., Chen, W., Duan, N., 2023. AGIEval: A Human-Centric Benchmark for Evaluating Foundation Models. https://doi.org/10.48550/arXiv.2304.06364.
[110] Moura, J., 2024. crewAI (version 0.30.11). https://github.com/joaomdmoura/crewAI.
[111] Lin, Z., Gou, Z., Liang, T., Luo, R., Liu, H., Yang, Y., 2024. CriticBench: Benchmarking LLMs for Critique-Correct Reasoning. https://doi.org/10.48550/arXiv.2402.14809.
[112] AgentOps-AI, 2024. AgentOps.ai (version 0.2.3). https://github.com/AgentOps-AI/agentops.
[113] Deng, X., Nan, P. 2022. cntext: a Python tool for text mining (version 1.7.9). https://doi.org/10.5281/zenodo.7063523 https://github.com/hiDaDeng/cntext.
[114] Y. Pan, M. Zhu, Y. Lv, Y. Yang, Y. Liang, R. Yin, Y. Yang, X. Jia, X. Wang, F. Zeng, S. Huang, D. Hou, L. Xu, R. Yin, X. Yuan, Building energy simulation and its application for building performance optimization: A review of methods, tools, and case studies, Advances in Applied Energy 10 (2023) 100135, https://doi.org/10.1016/j.adapen.2023.100135.