LLMartini: Seamless and Interactive Leveraging of Multiple LLMs through Comparison and Composition

YINGTIAN SHI, Georgia Institute of Technology, USA
JINDA YANG, Simon Fraser University, Canada
YUHAN WANG, Wuhan university, China
YIWEN YIN, Tsinghua University, China
HAOYU LI, Tsinghua University, China
KUNYU GAO, Tsinghua University, China
CHUN YU, Tsinghua University, China

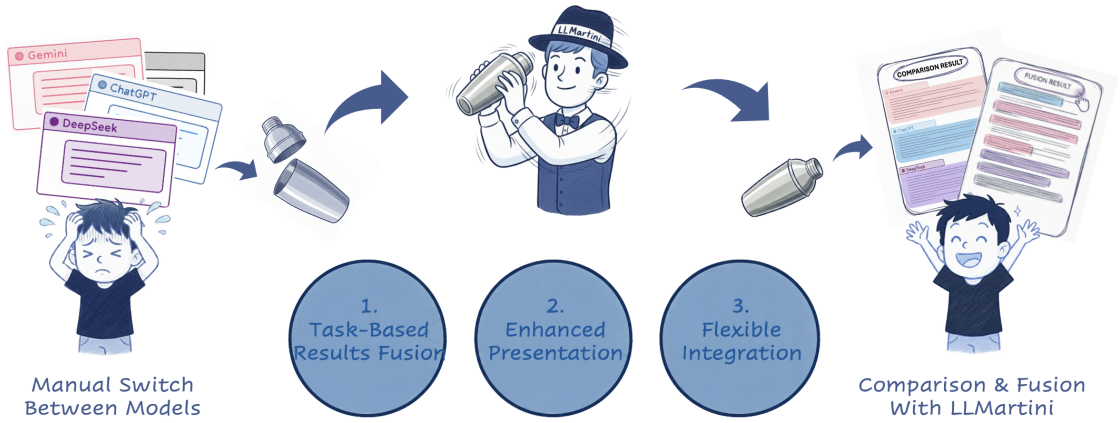

Fig. 1. LLMartini is a novel interactive system that supports seamless comparison, selection, and intuitive cross-model composition. The three key features of LLMartini are: Task-based purpose recognition and model results fusion; two enhanced interfaces for results comparison and fusion; and flexible fusion tools and functions.

The growing diversity of large language models (LLMs) means users often need to compare and combine outputs from different models to obtain higher-quality or comprehensive responses. However, switching between separate interfaces and manually integrating outputs is inherently inefficient, leading to a high cognitive burden and fragmented workflows. To address this, we present LLMartini, a novel interactive system that supports seamless comparison, selection, and intuitive cross-model composition tools. The system

Authors' Contact Information: Yingtian Shi, yshi457@gatech.edu, Georgia Institute of Technology, USA; Jinda Yang, Simon Fraser University, Canada; Yuhan Wang, Wuhan university, China; Yiwen Yin, Tsinghua University, China; Haoyu Li, Tsinghua University, China; Kunyu Gao, Tsinghua University, China; Chun Yu, Tsinghua University, China.

Manuscript submitted to ACM

decomposes responses into semantically-aligned segments based on task-specific criteria. It automatically merges consensus content and highlights model differences through color coding, all the while preserving unique contributions. In a user study (N=18), LLMartini significantly outperformed conventional manual methods across all measured metrics, including task completion time, cognitive load, and user satisfaction. Our work highlights the importance of human-centered design in enhancing the efficiency and creativity of multi-LLM interactions and offers practical implications for leveraging the complementary strengths of various language models.

CCS Concepts: • **Human-centered computing → Human computer interaction (HCI)**; **Interactive systems and tools**; *User models*.

Additional Key Words and Phrases: Large Language Models, Model Ensemble, User Experience Design, Human-AI Collaboration

ACM Reference Format:
Yingtian Shi, Jinda Yang, Yuhan Wang, Yiwen Yin, Haoyu Li, Kunyu Gao, and Chun Yu. 2018. LLMartini: Seamless and Interactive Leveraging of Multiple LLMs through Comparison and Composition. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym ')*. ACM, New York, NY, USA, 26 pages.

1 Introduction

The rapid advancement of powerful large language models (LLMs) has fundamentally transformed how users access information and accomplish tasks [1, 7]. Today, a growing diversity of models is available, each exhibiting unique strengths in areas such as factual accuracy, creative generation, or stylistic variation [8, 42]. Faced with this expanding ecosystem, users often lack a clear understanding of each model's specific capabilities. This uncertainty has prompted users to adopt a new strategy: querying multiple LLMs with the same prompt, then comparing and combining their outputs to produce higher-quality or more comprehensive responses. Empirical observations [30] and user studies (Section 3) indicate that this is not a niche activity. In fact, 93% of surveyed users actively compare outputs across different models and often engage in manual fusion, such as copying a paragraph from one model and combining it with a table from another, to construct a unified response.

Despite its practical benefits, this process remains highly inefficient. It requires users to constantly switch between interfaces, mentally track sources and quality of responses, and manually integrate outputs. This fragmented process imposes a significant cognitive load, creates operational friction, and makes the integration of outputs tedious and error-prone [68]. Prior work has established valuable insight for Human-LLM Interaction, but remains largely confined to single-model scenarios [58, 73, 89]. These insights can not be directly used to support the more complex scenario of multi-LLM interaction. Although multi-agent frameworks [27, 79] employ multiple models, they typically constrain user interaction within predefined, automated workflows that prioritize system-level optimization over user agency. Similarly, LLM ensemble systems [50, 60] are designed to produce a single, optimized output through algorithmic fusion. The ensemble system often suppresses the diversity and differences that users seek when manually exploring responses across models.

To bridge this gap, we first conducted user studies to identify core needs and derive design principles for multi-LLM interaction. Informed by these findings, we present LLMartini, a novel interactive system designed to support seamless and intuitive comparison, selection, and composition of outputs from multiple LLMs. LLMartini replaces the fragmented multi-tab workflow with a unified interface for accessing diverse models. It introduces a dual-mode interaction paradigm: a visual comparison view for divergent responses and a fusion interface for similar ones. The system automatically segments responses into semantically aligned units, merges consensus content, highlights discrepancies via color coding, and preserves unique contributions. Through this system, users can efficiently select, combine, and refine content while significantly reducing cognitive load. In a user study (N=18), LLMartini significantly outperformed manual

workflows across all measured metrics, including task completion time, cognitive load, and user satisfaction. Our work underscores the importance of human-centered design in facilitating efficient and creative multi-LLM interactions and offers practical insights for harnessing the complementary strengths of diverse models.

This work makes the following contributions:

- We identify key user needs, behaviors, and pain points in multi-LLM interaction and derive actionable design guidelines for future multi-model systems.
- We present LLMartini, a novel system centered on a visual comparison and user-controlled fusion paradigm, effectively reducing cognitive load and interaction friction.
- We conduct a user study demonstrating that LLMartini significantly improves task efficiency, increases user satisfaction, and promotes model exploration compared to conventional multi-tab workflows.

2 Related Work

2.1 Human-LLM Interaction

With the growing capabilities of AI systems such as Large Language Models (LLMs), user interaction with AI has become more frequent and widespread. Human-LLM interaction has emerged as a critical research area in human-centered artificial intelligence. Early studies [3, 6, 82] provided foundational principles and guidelines for designing interactive AI systems, emphasizing that building effective human-AI partnerships requires enhancing transparency, controllability, and human agency. However, the uncertain capabilities and complex outputs from LLMs pose significant challenges for users [39, 83, 86]. Researchers have attempted to improve user interaction experiences through various approaches, including enhancing the interpretability and controllability of AI outputs [2, 38, 43, 61, 73], developing collaborative frameworks [58], and optimizing prompt design methods[77, 89]. Extensive exploration has also been conducted on user-LLM collaboration in specific contexts [22] such as healthcare [47, 62, 75], education [26, 35, 54], management [55], writing [37], and programming [63, 85].

However, the interaction paradigms proposed in these studies still focus on interaction with a single LLM. Although current research on powerful models like GPT-4 highlights their broad capabilities, it also underscores that no single model is universally suitable for all tasks [10, 57]. Interacting with only one model limits users' ability to leverage the strengths of different models and dynamically select the most appropriate one.

To enhance the flexibility of user-LLM interaction, Luminate [67] assists users in exploring diverse outputs from a single LLM. Meanwhile, emerging frameworks such as AutoGen [79] and MetaGPT [27] support solving more complex tasks by composing multiple LLMs through structured dialogues, demonstrating the great potential of multi-model collaboration [23, 80]. While these multi-model frameworks provide users with more reliable outputs, they remain confined to predetermined architectures for LLM collaboration, sacrificing users' agency in comparing outputs and their control over the results of multiple models. In practical multi-LLM interactions, users tend to actively compare and combine outputs from various models—a characteristic that has not been fully explored. LLMartiniaims to address this gap by providing specialized interaction techniques for comparing and integrating outputs from different LLMs.

2.2 LLM Ensemble Systems

LLMs exhibit significant disparities in performance both on standardized benchmarks [8, 14, 53, 90] and in real-world user data [40, 44], reflecting an uneven distribution of capabilities across different domains and tasks. To enhance overall performance and reliability, methods that integrate multiple models have gained increasing attention. Ensemble

learning has a long-established history in machine learning [19] and has been proven effective in improving both the comprehensive ability and output consistency of models. Correspondingly, combining multiple LLMs to achieve complementary strengths, improve answer accuracy, and enhance robustness has become an important research direction.

Various multi-model fusion strategies have been proposed in existing studies[16, 46, 50]. The Mixture of Agents (MoA) framework employs a hierarchical structure that integrates generated outputs from different models—such as GPT, Claude, and Gemini—through multi-round collaborative reasoning and output fusion, significantly enhancing the accuracy and overall performance of the final output [76]. LLM-Blender introduces a two-stage integration framework to improve response quality and diversity [34]. In addition to direct output fusion methods, some studies have explored model knowledge fusion [74] and parameter fusion [18, 28, 84], both of which have demonstrated improvements in effectiveness and efficiency. Other approaches include self-integration through single-model multi-sample reasoning [41], or dynamic model selection via routing mechanisms, such as in FrugalGPT, to balance performance and cost [13].

With the advancement of multi-model systems, scalable multi-LLM platforms—such as OpenAgents [81] and AgentVerse [15]—have further facilitated the deployment, simulation, and evaluation of multi-model systems in real-world environments, providing support for research on model interoperability and collective intelligent behavior. Some studies suggest that well-integrated combinations of open-source models can even match or surpass the performance of proprietary models [69].

However, existing methods still tend to provide unified outputs, often failing to adequately account for the alignment between user preferences and task characteristics. For instance, in creative tasks, users may prefer diverse candidate results rather than a single "optimal" output. Current mainstream approaches typically rely on aggregation mechanisms—such as ranking, fusion, or voting—to achieve quality optimization or consensus [34, 49, 87]. While these strategies perform well in factual or objective tasks, they may suppress output diversity and even deviate from users' subjective intents. Therefore, in certain scenarios, it may be necessary to intentionally maximize the diversity of generated results [70].

To better adapt to user needs, we propose LLMartini, which incorporates a more flexible model fusion mechanism—such as preference-aware routing and diversity-preserving fusion strategies—along with an optimized user interaction design. This makes the integration process more transparent and controllable. The system can dynamically adjust ensemble strategies based on explicit user feedback or implicit contextual signals, thereby delivering a more human-centric model fusion experience.

2.3 Information Comparison and Decision Support Tools

Decision Support Systems (DSS) have a long history of aiding users in making informed choices by presenting and analyzing relevant information [64]. Traditional DSS often includes comparison features, allowing users to evaluate different options side by side. In recent years, machine learning (ML) and large language models (LLMs), have been widely integrated into DSS to enhance their analytical and decision-making capabilities across complex domains such as healthcare [5, 31, 45, 56, 72], business [66], resource management [29, 32], and scientific research and engineering [4, 88].

Within the emerging framework of Evaluative AI, the focus has shifted from providing a single answer toward presenting evidence, enabling users to make decisions themselves through the comparison and evaluation of information [51]. This shift underscores the central role of information comparison in the decision-making process. The form of DSS is being reshaped, with changes in how information is aggregated, accessed, and shared [11, 24]. The importance

of information comparison in decision-making is well-established. Research indicates that comparing multiple pieces of information can reduce uncertainty and lead to better decisions [36, 48]. This is particularly valuable in complex fields such as environmental protection, where single sources may be incomplete or biased. Furthermore, DSS incorporating information comparison has been widely applied in optimizing machine learning methods [59] and supporting human resource decisions [33], among other specialized domains. The effective comparison and utilization of multi-source information also depend significantly on its fusion and presentation. Methods for multi-source information fusion have been developed to integrate diverse information sources [20, 71, 78], and pattern-based approaches have been proposed for context-aware knowledge fusion within DSS [65].

However, existing comparison and fusion techniques were not designed for the generative nature of LLM outputs. Traditional side-by-side viewers lack specialized functionalities to compare textual responses that vary in structure, style, and content. Similarly, current DSS do not address the unique challenges of comparing and integrating outputs from multiple LLMs, such as handling differing response formats, identifying complementary information, and manually consolidating content. Inspired by current DSS design principles, we developed LLMartini, an LLM Fusion and Comparison Platform. It provides users with a systematic tool to weigh the value of different LLM outputs. Our design embodies the philosophy of Evaluative AI [51] by refraining from providing a single "correct answer." Instead, based on the result analysis, it highlights differences among models, supporting users in comparison, editing, and final decision-making.

LLMartiniaddresses the research gap by introducing the first interactive system specifically designed to support users in comparing and fusing multiple LLM outputs. Our work integrates insights from HAI research on human control and transparency, ensemble methods for combining model outputs, decision support principles for information comparison, and evaluation methods for assessing model capabilities—creating a novel interaction paradigm that places the user at the center of multi-LLM workflows.

3 Study1: Understanding User Behaviors and Needs in Multi-LLM Interactions

We have observed that users need to interact with multiple models simultaneously. To address this, we first conducted a questionnaire survey to quantify the scale of this demand and explore current user habits in LLM interactions. We collected 383 valid responses (159 male, 221 female, 3 prefer not to say, mean age = 23.96, SD = 3.87) from participants of diverse age groups, educational backgrounds, and professions.

The survey results (Fig. 2) reveal that 72% of users interact with LLMs more than five times per day, with 16.4% engaging in more than 20 interactions daily. Additionally, 93% of users reported using two or more models concurrently, among which 64.9% use exactly two models simultaneously. These findings highlight not only the significant demand for LLM interactions but also a strong preference for multi-model engagement.

To further investigate user motivations and needs in multi-model interactions, we developed a website that allows users to interact with multiple LLMs at the same time and conducted a user behavior analysis experiment. Through this experiment, we aim to address the following research questions:

- RQ1: What are the core motivations for users to interact with multiple large language models?
- RQ2: How do task types and model characteristics influence users' behavioral patterns when interacting with multiple models?
- RQ3: How do users compare, evaluate, and integrate outputs from different models? What are the main challenges and pain points encountered in this process?

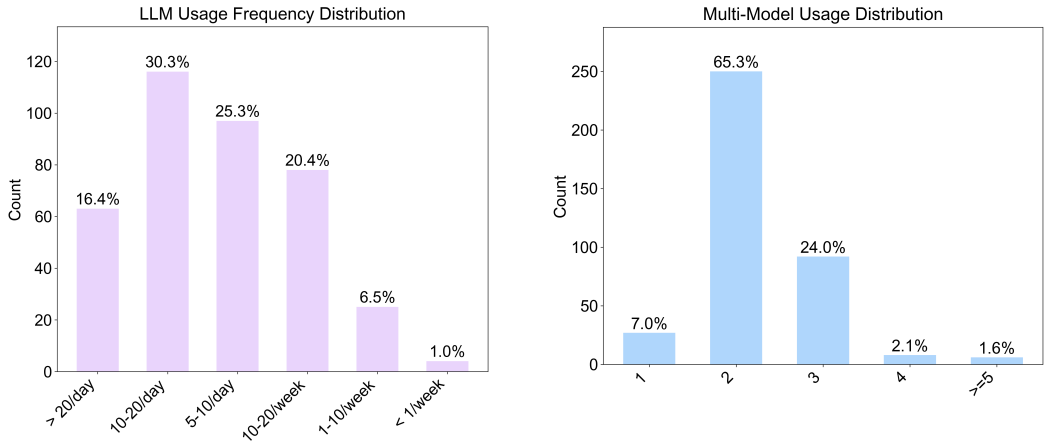

Fig. 2. The frequency distribution of users' interactions with LLM and the number of models used simultaneously by users in daily interactions.

3.1 Apparatus: The Basic Multi-LLM Prototype

We developed an experimental prototype website designed to enable users to obtain responses from multiple large language models (LLMs) simultaneously on the same page and provide feedback on the model outputs. The system uses FastAPI as the backend framework and React for the frontend interface, integrating APIs from several mainstream models. All user interaction requests and model responses are persistently stored in a local PostgreSQL database.

Based on the four most frequently used models identified in our preliminary survey—DeepSeek [17], ChatGPT [52], Doubao [12], and Gemini [21]—we included them as the supported models in the system.

In terms of page layout, we drew inspiration from mainstream LLM products (such as DeepSeek and ChatGPT), adopting a navigation bar to display historical conversation records and a main chat interface. This design aims to closely mimic users' familiar interaction environment and reduce the learning cost.

The output of each model is presented in two forms: a summary card and an expanded view. The summary card displays the model name and the first 50 characters of the generated content, with four action buttons at the top (Fig. 3 b.): "Cite", "Like", "Dislike", and "Regenerate". Expanding the panel reveals the full response, along with support for adding comments and copying the text (Fig. 3 d.).

After a user submits a query, summary cards for all models are displayed side by side. By default, the first model in the list generates a response automatically; users can also manually trigger other models to run. If unhappy with a result, they can click "Regenerate" to refresh that model's output. All responses are streamed progressively and can be fetched simultaneously.

Users can quickly evaluate model outputs via "Like" or "Dislike" buttons, or provide more detailed textual feedback in the expanded panel. The "Cite" (Fig. 3 b.) feature allows users to select one or more results as context for subsequent dialogue, and cited model names are displayed next to the input box (Fig. 3 a.). If no model is explicitly selected, the response from the first model is used as the default context. Additionally, users can specify a preferred model (Fig. 3 c.)before the conversation begins; its card will be automatically pinned to the top and set as the default for generating responses.

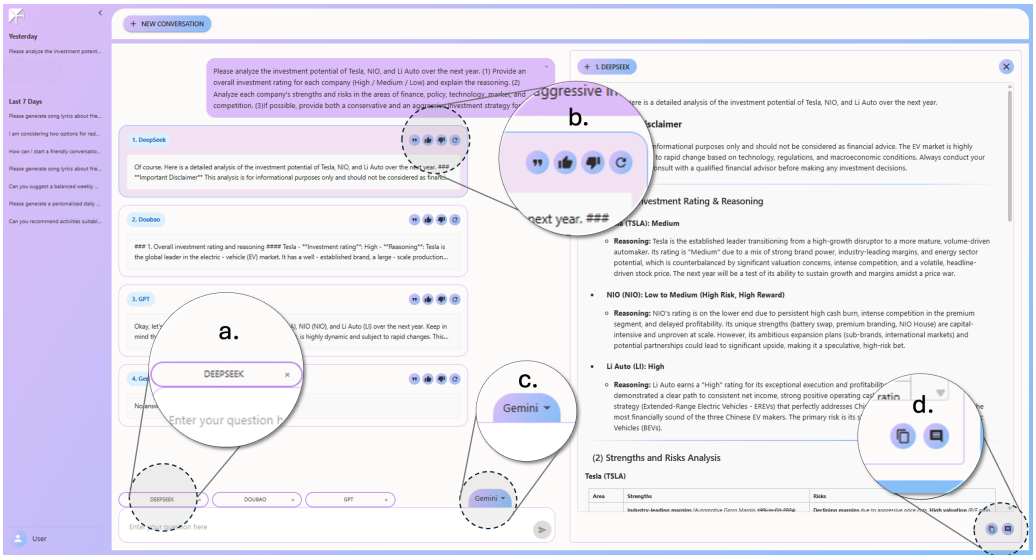

Fig. 3. Website prototype, with the history bar on the left, user interaction input and multi-model cards in the middle, and model output details on the right. Zoomed highlights show: (a) cited model name, (b) four action buttons ("Cite", "Like", "Dislike", and "Regenerate") on summary card, (c) dropdown model selector, (d) "Copy" and "Comment" buttons on expanded panel.

3.2 Study Procedure

We invited 19 users (9 male, 10 female, mean age = 25.37, SD = 4.54) to participate in the experiment. The participants had diverse backgrounds and varied significantly in their frequency of LLM usage and typical task scenarios. The experiment lasted for one week, during which users were asked to use our platform following their natural interaction habits. While we did not impose strict restrictions on usage scenarios, we encouraged users to cover as many different types of tasks as possible.

To ensure data quality and sufficiency, we required each user to complete at least ten distinct types of tasks and submit no fewer than 70 requests in total. Additionally, after each request, users were required to provide quick feedback (like or dislike ratings) for at least one model's output and submit at least one written comment.

At the end of the experiment, we conducted approximately 30-minute semi-structured interviews with each user to gain in-depth insights into their experience, main pain points, and their perspectives and suggestions regarding the multi-model interaction paradigm. As a token of appreciation, each participant received a gift card worth between $15 to $30, based on the amount of data they contributed.

3.3 Study Results

We cleaned the collected data by excluding invalid records caused by system errors or model invocation failures, ultimately obtaining 1893 valid user queries and 8058 model interaction records (each successful retrieval of a model output was counted as one interaction, including regeneration). For user feedback, we filtered out short comments containing fewer than 10 characters, retaining 2049 instances of quick feedback (likes/dislikes) and 1856 valid textual comments. To further analyze user behavior and model performance in depth, we conducted a detailed examination of user queries and their corresponding model interaction records.

3.3.1 Task-Related Analysis. In terms of task segmentation, we distinguished task types based on user requests and their contextual information: significantly different topics within the same session were treated as independent tasks, while follow-up questions on the same topic were grouped under the same task.

Task categorization employed a heuristic iterative method: First, we selected 50 diverse tasks as seed samples for manual pre-classification. Subsequently, this classification result was used as a prompt to iteratively categorize the remaining tasks using the DeepSeek model. The model determined whether a new task belonged to an existing category; if not, it generated a new category and dynamically updated the prompt. After multiple iterations, we manually verified and adjusted the classification results to enhance inter-category distinction. Finally, all tasks were reclassified based on the stabilized prompt from the iterative process to ensure consistency. We established a classification system along two distinct dimensions: based on task content (the specific topics involved), resulting in 66 categories; and based on task purpose (the user's core intent behind the query), resulting in 4 broad categories and 17 subcategories (One of the categories is "Others"). The purpose-based classification system includes the following four typical types (each broad category contains several subcategories):

- Content Generation(G): Refers to creating new text, code, or creative ideas from scratch, emphasizing originality and creativity. Includes "Creative Writing", "Opinion Expression", "Simulated Dialogue", "Code Generation", "Explanation", and "Question Generation".
- Content Editing(E): Involves modifying, polishing, reorganizing, or optimizing existing text to improve its readability, accuracy, and expressiveness. Includes "Text Editing", "Copywriting Editing", and "Emotional Editing".
- Information Retrieval (R): Aims to retrieve, filter, and integrate information from various sources based on user queries to fulfill factual inquiries, comparisons, or extraction needs. Includes "Information Query", "Comparative Analysis", and "Information Extraction".
- Problem Solving (S): Provides practical solutions or operational guidance through analysis, reasoning, and suggestions tailored to specific user needs. Includes "Recommendation", "Q&A Analysis", "Exercise Solving", and "Solution Design".

The results in Figure 4 indicate that the topics of the tasks are widely distributed across multiple domains. Among them, health and diet-related questions were the most frequent topics of concern for general users. In terms of interaction rounds (number of user queries per task), complex tasks such as programming, business, and event planning significantly exceeded other types, demonstrating users' deeper interactive demands in these scenarios.

From the distribution (Fig. 5), Problem Solving tasks accounted for 61.91%, making them the primary purpose of user interaction, indicating that users tend to use large language models as tools for addressing practical problems. However, Content Generation tasks had a relatively lower proportion; their average number of interaction turns was significantly higher, suggesting that these tasks often require multiple rounds of iteration and refinement.

3.3.2 Model-Related Analysis. In addition to task-based analysis, we also examined user habits regarding model usage. On average, users used 3.26 models per request (excluding regeneration), and 58.5% of requests involved using all four models. Figure 6 shows that each user's model preferences and usage frequency varied. We measured model performance using like and dislike rates, normalizing the scores within each task type (by dividing a single model's performance score by the sum of all performance scores for that task type). The result (Fig. 7) shows that model performance indeed differed across task types. While Deepseek demonstrates superior performance in the domains of Philosophy and Family Education, ChatGPT holds an advantage in the areas of Event Planning and Data Processing. We also validated this finding through user feedback during the interviews.

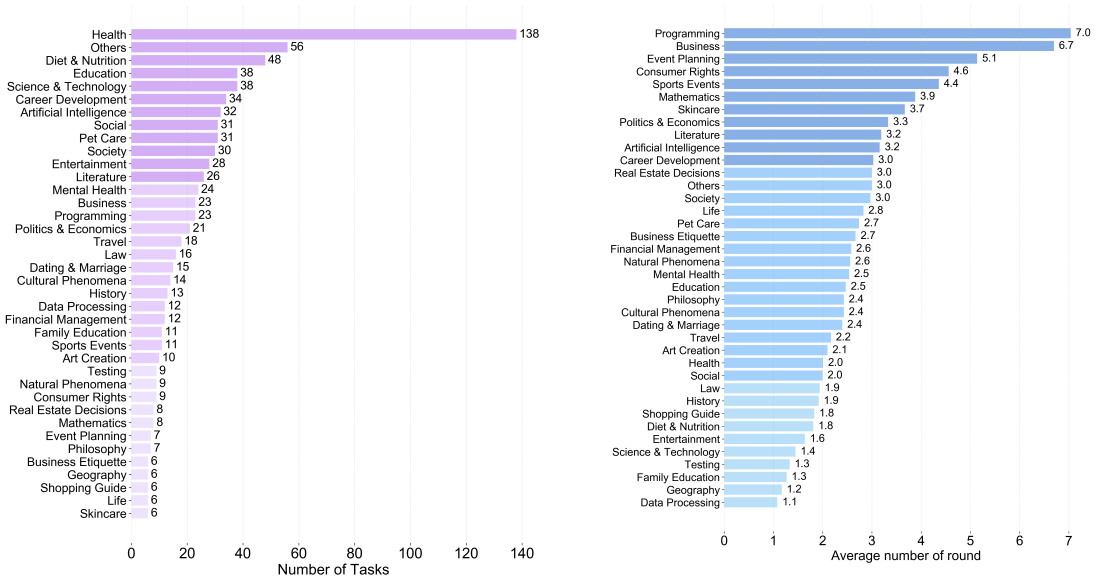

Fig. 4. Based on the statistical results of the task topic, the left figure shows the average total number of tasks for each topic, and the right figure shows the average number of user request rounds required to complete a task for the topic.

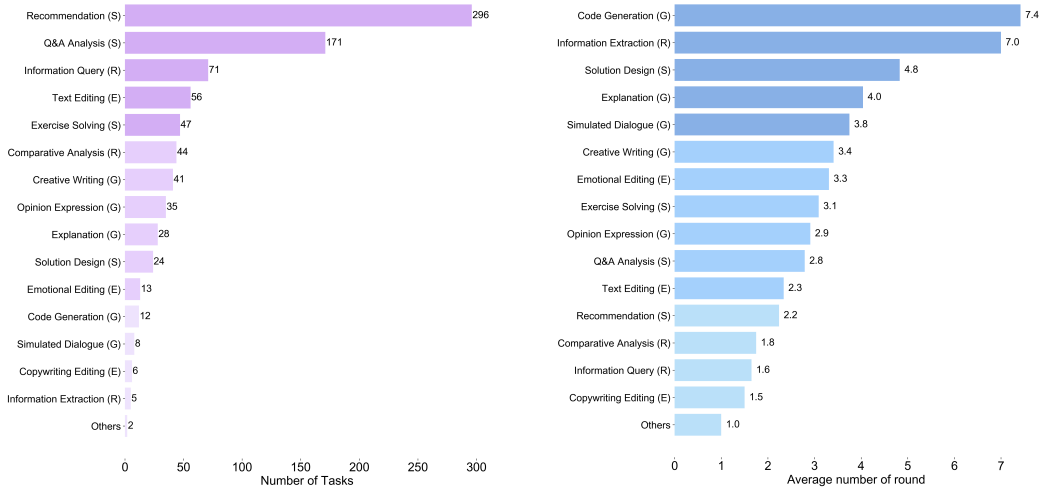

Fig. 5. The statistical results based on the task purpose are as follows: the left side shows the average number of tasks for each topic, and the right side shows the average number of user request rounds required to complete a task for that topic. The capital letters after the task classification represent the major categories to which it belongs.

3.4 Findings

Based on our data analysis and user interviews, we derived the following conclusions:

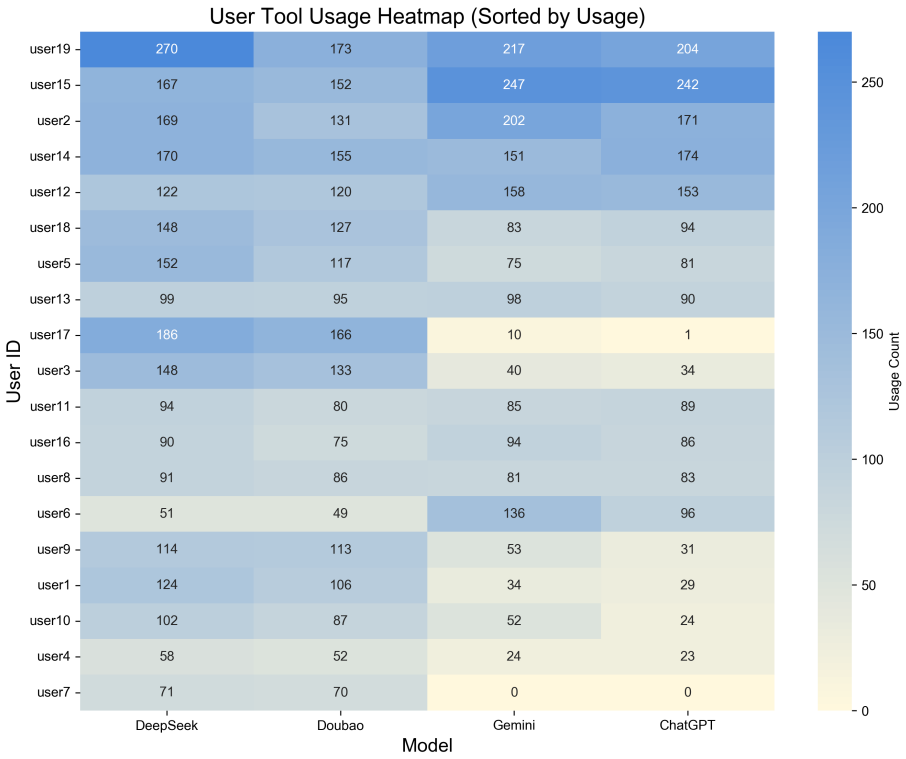

Fig. 6. The heat map of model interaction (Darker blue indicates more usage, Lighter yellow indicates less usage). The results are sorted from top to bottom according to the total number of user interactions. The number in each block represents the number of interaction times.

1. Information breadth, depth, and output differences drive users to try different models. Our data and user feedback indicate that model outputs vary in style and content across different usage scenarios, and user preferences and habits also differ. Some users found DeepSeek's responses detailed and structured, meeting their expectations, while others considered its answers overly verbose and preferred Gemini's concise style. Normal users typically lack prior knowledge about model characteristics and cannot determine which model is better suited for a specific problem or which output aligns more closely with their expectations before getting the results. Consequently, to obtain more comprehensive or suitable results, users tend to use multiple models simultaneously for comparison or attempt to integrate outputs from different models. This fundamentally addresses the user motivations for multi-LLM usage explored in RQ1. During interviews, we asked users to rate the necessity of using multiple models simultaneously on a 7-point Likert scale, yielding an average score of 5.53 (SD = 1.46).

2. Task purpose influences the number of models users compare and their behavior. When asked about their expected number of models for comparison, 63.16% of users stated that three to five models were sufficient for their needs, but also pointed out that this number is closely related to the task purpose. For tasks with relatively fixed or consistent answers (e.g., information retrieval questions), results from different models show little significant variation. Users only need one or two models to verify the results. However, for highly subjective tasks with diverse outputs, such

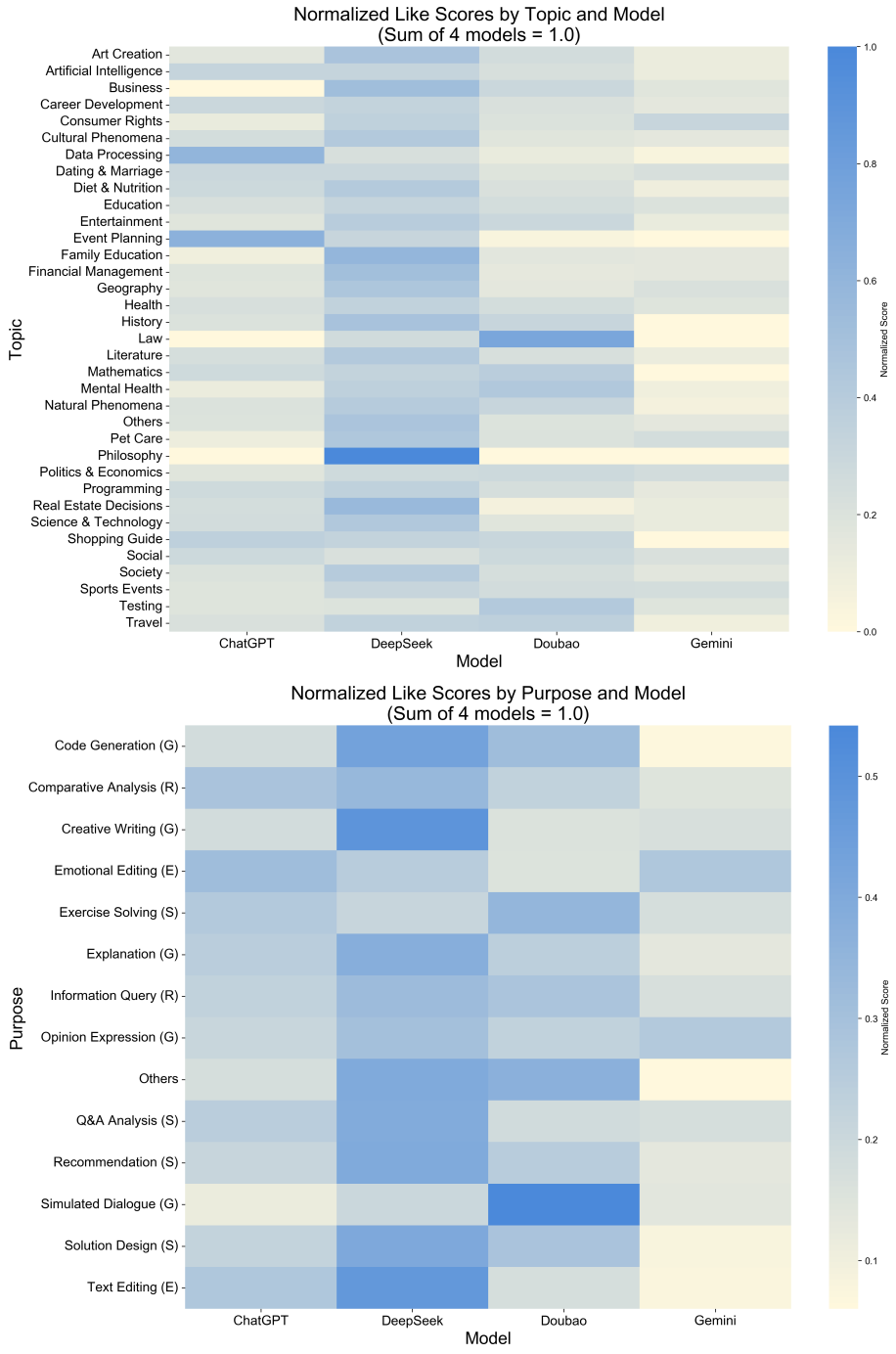

Fig. 7. Normalized model likelihood heatmap based on task type (Darker blue indicates higher score, Lighter yellow indicates lower score). A higher normalized score indicates a higher likelihood rate for the model on related task type.

as content generation, users prefer to examine results from multiple models to gain richer information. Furthermore, regarding the handling of model results, users expressed that if results differ significantly, they wish to expand all results simultaneously for quick browsing. Conversely, if model outputs contain overlapping or similar content, they prefer an intuitive way to identify the differences.

3. The "Surprise" is a key factor enhancing user evaluation and acceptance. Combining user textual comments and interview feedback, we found that users particularly value "surprise" in model outputs—i.e., unexpected thorough considerations or unique information beyond their expectations. Users stated that if a model's output contains such "surprise," they are more inclined to adopt its results because "it makes me feel the model considered more comprehensively, and the output is more reliable." The pursuit of "surprise" also motivates users to proactively try multiple models and integrate results to obtain more comprehensive responses. Points 2 and 3 collectively address the influence of task and model characteristics on user behavior explored in RQ2.

4. Users exhibit a "Browse-Compare-Integrate" behavior pattern. Users generally reported that after obtaining outputs from multiple models, they typically first browse the overall structure and then compare the similarities and differences between various outputs. When adopting results, besides directly selecting output from a specific model, many users actively integrate outputs from multiple models to form the final desired result. In existing commercial LLM interactions, they often need to switch between different pages or applications and finally integrate the results in an editing application, which significantly increases their operational and cognitive load. Although our current platform reduces the need for multi-page switching, users still expressed a strong demand for efficient comparison and integration features. This clearly identifies the multi-model interaction paradigms and existing pain points concerned in RQ3.

3.5 Design Guideline

Based on our experimental observations and user interviews, we derived three design guidelines aimed at enhancing the user experience and effectiveness of multi-LLM interaction systems:

1. Provide context-aware intent recognition and model recommendation The system should be capable of automatically identifying the task intent (e.g., content generation, content editing, problem-solving) and dynamically adjust the default number and type of models responding accordingly. For instance, models more proficient in the detected task type should be prioritized. Multiple models could be activated by default for creative tasks, while a single model might suffice for factual queries.

2. Support differentiated parallel responses and flexible comparison schemes The system should support parallel response generation from multiple models within a unified interface and allow users to easily expand, compare, and focus on different results. For tasks with significantly divergent outputs, parallel display should be provided to facilitate quick browsing. For tasks with similar results, the system should automatically summarize consensus points and highlight key differences to reduce users' cognitive load.

3. Enhance result actionability and integrability Effective tools and interfaces should be provided for users to act on model outputs, such as supporting segment selection and one-click merging of answers across models. Users should be able to flexibly combine outputs from different models to independently construct a final result that better meets their needs, thereby optimizing the "Browse-Compare-Integrate" behavior and reducing operational overhead.

4 Platform Design

Based on the observations and interviews from Experiment 1, we summarized three key design guidelines (Section 3.5). With these guidelines as our design objectives, we aimed to develop a next-generation multi-LLM interaction platform that not only enhances interaction efficiency but also effectively reduces users' cognitive load. Building upon the prototype from Experiment 1, we introduced the following three new features:

- 1. Task Type Recognition and Model Recommendation: Automatically identifies the type of user query and recommends suitable models.
- 2. Enhanced Model Output Presentation Mechanism: Visually presents differences and consensus among models based on their generated content.
- 3. Flexible Multi-Model Result Integration: Provides both automatic and manual result integration functionalities.

4.1 Workflow

After a user submits a query, the system backend first performs real-time intent recognition and task type classification. This type of label is displayed in the interaction interface (Fig. 8 b.) in real-time to enhance the explainability of the system's behavior and users' perceptual transparency. Based on the identified task type, the system dynamically adjusts its model dispatching strategy. This includes recommending the order for invoking models and automatically executing the corresponding number of models according to the task purpose. Recommendation reasons (Fig. 8 c.) are attached to the summary cards generated by each model. In addition to the models invoked automatically by the system, users can also manually trigger other models to run.

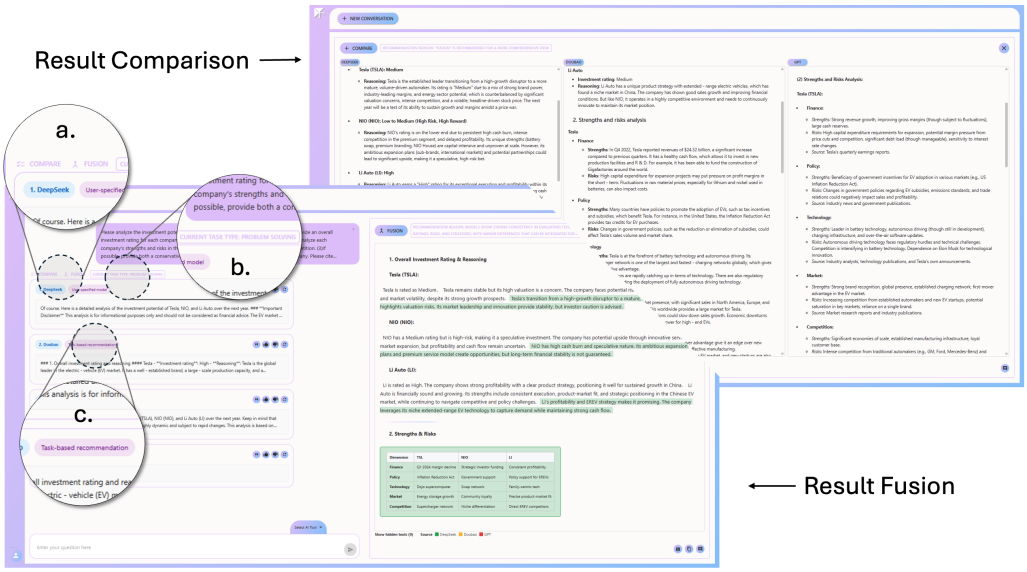

Fig. 8. LLMartini provides two comparison modes. The top image shows the comparison results, and the bottom left image shows the fusion results. Zoomed view highlighting (a) fusion/comparison buttons, (b) current task type (e.g., "Problem-Solving"), and (c) reason for tool recommendations

After obtaining the model results, the system provides an "Fusion" function (Fig. 8 a.). Upon clicking the integration button, users can select the models whose outputs they wish to integrate in a pop-up interface. The system will then synthesize the user's selected results, the original query, and the task type to algorithmically recommend an appropriate presentation method, specifically divided into two modes:

- Result Comparison: Suitable for situations where outputs differ significantly. It presents the detailed results of multiple models in a side-by-side, synchronized scrolling format, facilitating intuitive comparison.
- Result Fusion: Suitable for results with overlapping content or similar semantics. It processes and integrates the outputs, explaining at the top of the interface for recommending this mode.

Furthermore, we retained a direct button for switching to the result comparison view, allowing users to easily access the comparison page if desired.

4.2 Implementation

4.2.1 Task Objective Awareness and Dynamic Model Dispatching. Results from Study 1 indicate that the task category significantly influences users' behavioral patterns when interacting with multiple models. To address this, we introduced a mechanism for dynamically dispatching models based on the task objective. When a user submits a request, the system utilizes the task classifier constructed in Experiment 1 to identify the task objective (including major and minor categories) in real time and records it. The model dispatching strategy is primarily formulated based on the performance ranking of models under each major task category, which is calculated from the aggregated user data of Study 1. We chose to make recommendations at the major category level (rather than minor categories) because some minor categories have limited sample sizes and uneven user distribution, which could introduce user bias and affect the generalizability of recommendations.

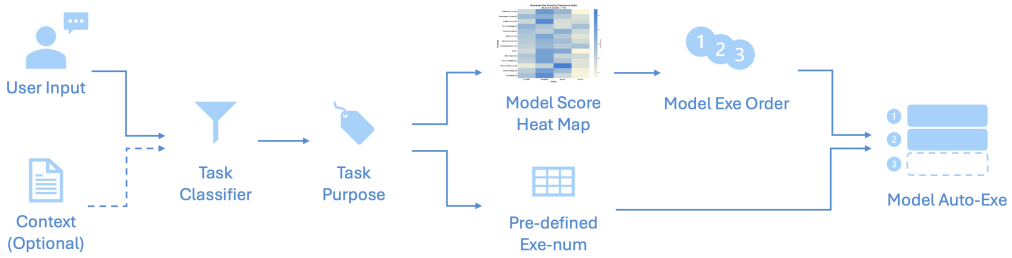

Fig. 9. LLMartini Dynamic Model Dispatching. First, the task purpose is analyzed based on user input and the current context. Then, the number and order of model executions are matched according to the task purpose. Finally, the model is automatically executed.

During the dispatching process, models actively specified by the user still receive the highest priority and are placed at the top of the queue. The system predefines the number of models to be automatically executed for each major task category: the "Content Generation" category, which demands the highest creativity, automatically invokes all four models, while the other major categories default to executing the top two best-performing models.

4.2.2 Adaptive Integration and Comparative Display. After the user selects the model results to be integrated, the system initiates an adaptive integration process by synthesizing the user's request, task objective, and model outputs. The integration process is divided into two stages: preliminary integration and integration enhancement.

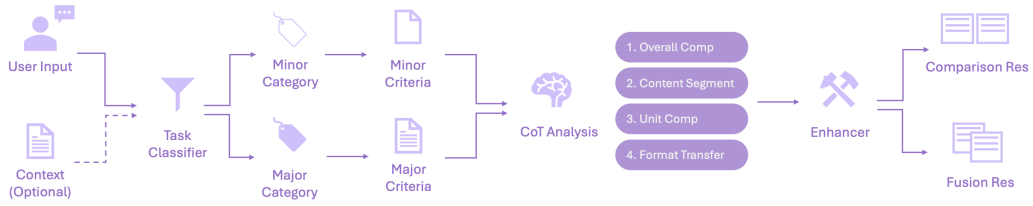

Fig. 10. LLMartini Fusion and Comparison Pipeline. First, the task purpose is analyzed based on user input and the current context. Then, the corresponding evaluation criteria are obtained based on the task purpose. A CoT approach decomposes the model output into comparison units, which are then integrated under the guidance of the evaluation criteria to produce the final result.

We employ a Chain-of-Thought prompting approach to achieve preliminary integration. Since users have different focuses for different tasks, the system maintains a two-tiered set of integration criteria: major category integration criteria and minor category integration criteria. The major category integration criteria define the core focus points for each type of task. For example, the "Problem Solving" category emphasizes solutions and logic, while the "Content Editing" category focuses on modification quality and style consistency. The minor category integration criteria are refined to specific subcategories, providing standards for determining differences between results and their priorities.

In the preliminary integration, the integrator first evaluates the overall differences between results based on the major category criteria. If the differences are too significant, it directly recommends a comparative display; otherwise, it attempts integration. For model outputs requiring integration, the system first structures the outputs into blocks according to the task type. For instance, in information retrieval tasks, model outputs can be divided into modules such as "Task Comprehension," "Information Provision," "Summary," and "Resource References." The system allows for differences in module completeness across models, as varying levels of completeness are an important reflection of differences between models. Content within each module is further segmented semantically into units that can be compared or integrated.

Subsequently, the integrator judges whether semantic units generated by different models are similar based on the minor category criteria. Units originating from a single model are retained; units generated by multiple models are either merged or retained in parallel based on semantic similarity. Finally, the system converts the output into a structured JSON format, where each unit corresponds to a list. The list elements annotate the source model and the corresponding content.

To implement this process, we developed dedicated prompt templates for each major task category, embedding the integration criteria and segmentation logic for that category. We also introduced preferences for integration versus comparison (e.g., content generation tasks prefer comparison, while content editing tasks lean towards integration). Additionally, we included relevant examples to help the integrator understand how to segment and differentiate model differences. Minor category criteria are dynamically concatenated into the major category prompt during integration to provide a more refined judgment basis.

After preliminary integration, direct concatenation of some units may lead to a disjointed expression. Therefore, we introduce an enhancer to optimize and validate the results. Firstly, it improves textual coherence and readability, enhancing the quality of the integrated output. Secondly, it performs secondary validation to ensure the rationality of each unit segmentation, splitting or merging units as necessary to improve display effectiveness and user experience.

4.2.3 Multi-LLM Result Integration Interface. In the integration display view, the system divides content generated by different models into paragraph or sentence-level fragments based on semantic units. The source model of each fragment is distinguished by color highlighting, with a legend at the bottom indicating the color-model correspondence. If a segment of content is generated by only a single model, it is highlighted in the corresponding color. If multiple models generate semantically similar content, it is automatically merged and not highlighted to reduce visual clutter and indicate high consensus and reliability. If different models generate relevant but semantically distinct content, both versions are retained, and a switching mechanism is provided—users can browse different expressions of the same content using left and right arrows that appear upon hovering.

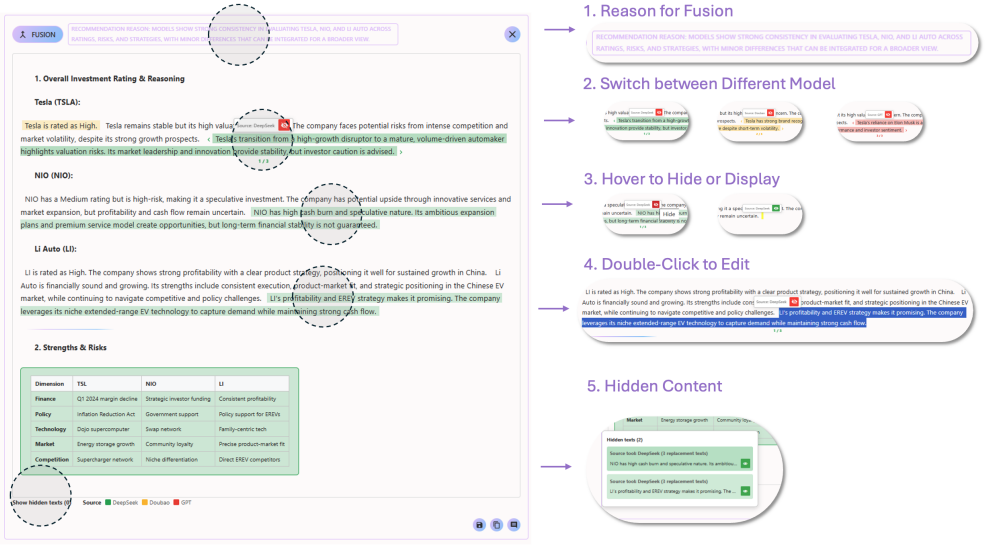

Fig. 11. LLMartini Fusion Interface. We provide users with rich fusion interactions. Users can view fusion reasons (1), switch results from different model sources (2), hide or show models (3), edit results (4), and uniformly view hidden content (5).

Additionally, hovering displays the model name and allows users to hide the current paragraph. Hidden content is marked with a yellow placeholder, and users can restore it at any time via hover actions or the "Hidden Content" panel in the lower left corner of the page. Users can also double-click any paragraph to enter edit mode directly, allowing free adjustment of the content. Through these features, users can quickly identify differences and unique contributions from various model outputs, conveniently switch, hide, or edit content, and efficiently construct an output that meets their needs. When the user clicks the "Copy" button, the system automatically integrates all currently visible and edited content in the interface and writes it to the clipboard for subsequent use.

5 Study2: System Usability and User Evaluation

To evaluate the effectiveness, usability, and user satisfaction of LLMartini in real-world usage scenarios, we conducted a user experiment to assess system usability and gather user feedback. We aimed to address how LLMartini performs compared to traditional multi-tab or multi-window interaction methods in terms of task efficiency, user experience, and cognitive load.

5.1 Study Design

5.1.1 Participants. We recruited 18 participants (8 male, 10 female, mean age = 24.8, SD = 3.3). All participants were selected from a sample pool distinct from Study 1 to ensure evaluation independence. To cover as much task diversity as possible, we selected participants with diverse backgrounds and varying levels of LLM experience.

5.1.2 Tasks and Procedure. We defined the method where users manually switch between tabs or applications to interact with different models and manually combine the outputs as the baseline. The LLMartini platform served as the experimental condition.

The experiment employed a within-subject design; each user experienced both interaction methods. To counterbalance order effects, half of the users started with the baseline method, while the other half began with LLMartini.

Each participant was required to complete a series of predefined tasks. The experiment consisted of two sessions, each using one interaction method. Within each session, users were asked to complete three distinct tasks with different topics under each of the four different task purposes (Content Generation, Content Editing, Information Retrieval, Problem Solving), resulting in 12 tasks per session. Regarding task topics, those obtained from Study 1 were divided into three sets based on their frequency of occurrence: high(more than 25 times), medium(10-25 times), and low(less than 10 times). The topic for each distinct task was randomly selected from one of these sets according to the user's own background, ensuring minimal topic similarity between users while guaranteeing coverage of all topics. Users needed to conceptualize the specific task based on the given objective and topic and complete it through interaction. In total, each user completed 2 methods × 4 task objectives × 3 task topics = 24 interactive tasks.

At the beginning of the experiment, the facilitator introduced the background and task content. Before using LLMartini, the facilitator demonstrated its main features, including multi-model response generation, result comparison, and result integration, using a simple query. Users then proceeded through the two experimental sessions. Task completion time was recorded by the facilitator, excluding the time spent conceptualizing the question and waiting time for model output. There is 10 10-minute break between sessions. After completing all the tasks, users filled out the NASA-TLX [25] questionnaire and the system SUS [9] questionnaire for both methods. This was followed by a semi-structured interview, inviting users to share their experiences and feedback. The entire experiment typically lasted from 90 to 120 minutes. Participants were compensated with a $20-30 gift card upon completion.

5.2 Results and Analysis

5.2.1 Quantitative Analysis Results. We calculated usability and cognitive load metrics for both methods. The average task completion times for LLMartini and the baseline were 2.59 minutes (SD = 1.80) and 3.40 minutes (SD = 1.48), respectively. Our system significantly reduced the time required to complete tasks ($p = 0.015 < 0.05$). This indicates that LLMartini effectively enhances task efficiency by reducing context switching and providing an integrated workflow.

The SUS scores for LLMartini and the baseline were 75.56 (SD = 11.32) and 49.72 (SD = 11.84) (out of 100), respectively. According to the SUS rating scale, LLMartini's score falls within the Good range and is significantly higher than that of the baseline method ($p < 0.05$). LLMartini shows no significant difference from the baseline method in terms of Learn Quickly and Need to Learn. This is because the baseline method is generated from users' daily usage habits, resulting in a lower learning cost. The absence of a significant difference between the two methods also indicates that LLMartini itself has a very low learning cost. In contrast, LLMartini scores significantly higher than the baseline in Need Support. This is because the baseline method requires almost no guidance, whereas LLMartini introduces a new fusion interface, and users who are new to the platform may require some functional introductions to help them quickly understand

its capabilities. Additionally, LLMartini significantly outperforms the baseline method across all individual criteria of usability.

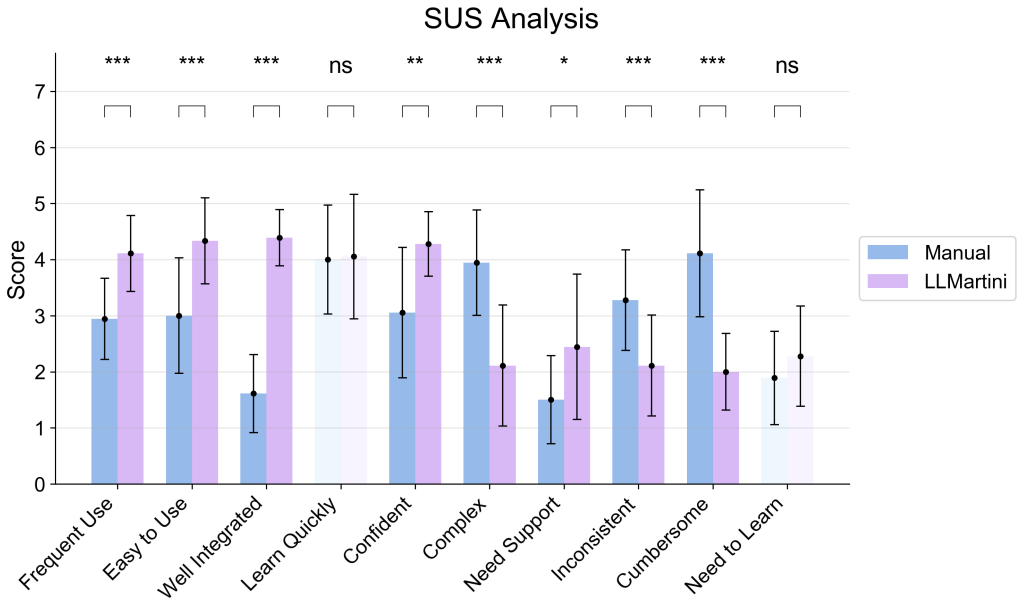

Fig. 12. SUS scores and significance analysis results of LLMartini and baseline.

Regarding cognitive load, the NASA-TLX scores for LLMartini and the baseline were 40.58 (SD = 14.65) and 59.66 (SD = 17.13) (out of 100), respectively. LLMartini significantly reduced users' cognitive load ($p < 0.05$). Users reported that "not having to switch between apps" and visualization of differences greatly eased their comparison burden. For the Performance score, there was no significant difference between the baseline method and LLMartini. This is because the task completion criterion was defined as users having obtained a model fusion result they found acceptable for their task. Consequently, users were able to achieve their desired output with both methods, leading to comparable scores in the performance category. In contrast, LLMartini demonstrated a significant reduction in burden across all other individual burden metrics.

5.2.2 User Preferences and Behavioral Patterns. During the interviews, we recorded users' usability evaluations and willingness to use both systems on a 7-point Likert scale. LLMartini scored significantly higher than the baseline in both usability (5.89 vs 2.67) and willingness to use (6.17 vs 4.06) ($p < 0.05$).

All participants agreed that LLMartini's approach of interacting with multiple models simultaneously and automatically integrating their outputs was an exciting design. Participant 14 further commented that the platform offered more than just combining results from four models: "Through functions such as quoting, simultaneous invocation, and fusion, I can achieve collaboration among multiple language models—for example, using outputs from two models as context for the next round, then fuse the output from the next round." This made users feel that LLMartini provides greater potential for multi-model collaboration.

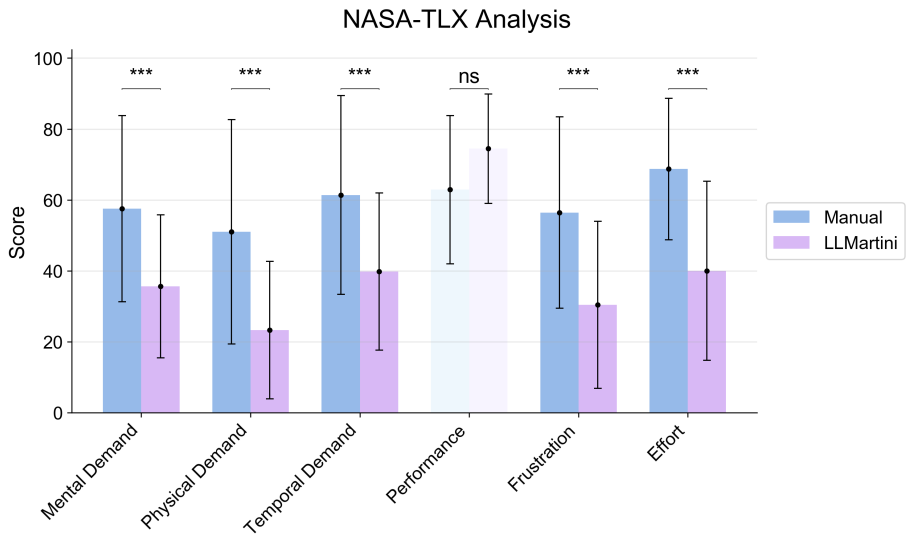

Fig. 13. NASA-TXL scores and significance analysis results of LLMartini and baseline

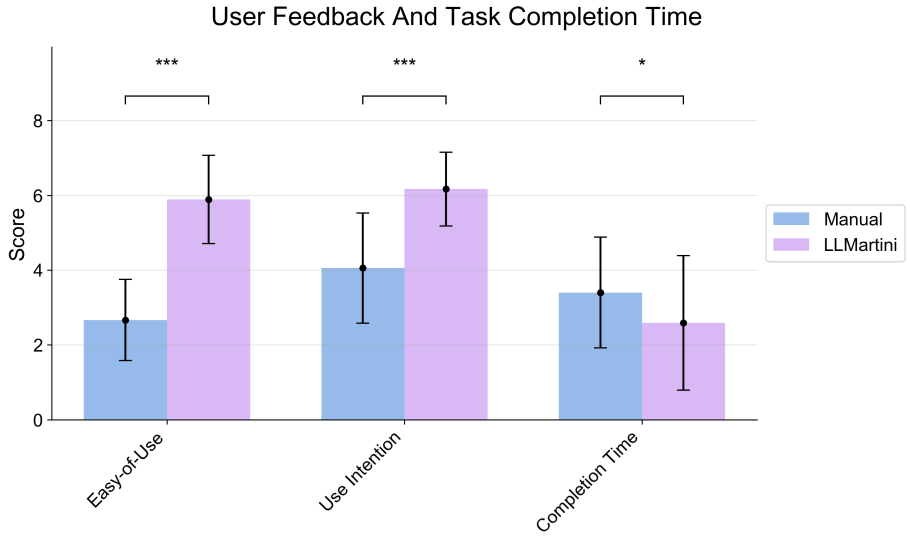

Fig. 14. User feedback scores and Completion time significance analysis results of LLMartini and baseline.

We observed that different users focused on different aspects when using LLMartini. Users with less experience in LLM interaction tended to appreciate the speed and convenience of the fusion function. Participant 9 mentioned that LLMartini's automatic fusion reduced their decision-making burden, eliminating the need to manually compare and select model results, while quickly delivering more comprehensive outcomes. On the other hand, users who frequently use LLMs valued the flexibility offered by the system, which enhanced their efficiency when combining model outputs.

However, some users noted that when fusing lengthy model outputs, LLMartini might omit certain details. This loss of information could occasionally increase their cognitive load, sometimes even requiring them to manually review the original results. They expressed a desire for the ability to configure parameters of the fusion function—such as increasing the weight of a specific model or adjusting the granularity of fusion.

5.2.3 Unexpected Usage Behaviors and Design Implications. In addition to the user usage we expect, we observed several interesting, unexpected usage patterns:

1. "Exploratory Comparison" Behavior: Some users intentionally selected models with the most divergent output styles for comparison, not to solve a specific problem but to "understand the performance differences between different models" or "test the boundary of comparison and fusion".
2. Novel Use of Integration Feature: Some users utilized the integration feature as a "quality filter," judging the reliability of information by observing which content was commonly generated by multiple models (the automatically merged sections).
3. Session Knowledge Accumulation: Some users employed the fusion results as contextual information in subsequent interactions. They viewed LLMartini as a platform for simultaneous collaboration with multiple LLMs, where each interaction generated collective knowledge through fusion and propagated it to all models.

Some expert users suggested the system should record historical comparison results to form a personalized "model characteristic knowledge base." They expected the system to record their interaction habits and learn how to achieve better fusion from their interactions. These findings offer new insights for multi-LLM interaction: a) There is a need to support more flexible comparison and fusion strategies; b) A more adaptable contextual reference mechanism should be provided; c) The introduction of long-term learning and personal adaptation capabilities should be considered.

The results of Study 2 confirm the effectiveness of LLMartini in enhancing the efficiency of multi-model interactions. Compared to the traditional method, our system performed better not only in objective metrics (task time, error rate) but also showed significant improvement in subjective user experience (satisfaction, cognitive load).

6 Discussion

6.1 From Tool to Collaborator

The initial inspiration for designing LLMartini came from users' need to utilize multiple models simultaneously. However, as our research progressed, we gradually discovered that LLMartini is no longer just a tool for improving efficiency in user interactions with multiple models.

When users interact with multiple models, beyond comparing and integrating results, there are more complex collaboration patterns at play. Certain complex tasks require users to select different models at different stages to achieve better outcomes. In such scenarios, the user's interaction resembles working with a team of individuals skilled in different areas. This "team collaboration" style of interaction prompted us to rethink LLMartini's system positioning—it is no longer merely a platform for multi-model invocation and integration but rather a mediator of cognitive collaboration. Through LLMartini, users gain insights into the capabilities of different models, dynamically assign tasks based on objectives, and continuously adjust collaboration and information sharing among models during execution. For example, in a creative generation task, a user might first use one model for brainstorming, then another for logical validation, and finally call upon a third model to optimize the expression style. The entire process is no longer a simple parallel invocation or result integration but a highly dynamic, temporally structured intelligent collaborative behavior.

This discovery also hints at new possibilities for future AI systems in supporting complex tasks: systems should not only focus on improving model performance but also pay attention to how to help users flexibly organize multiple model capabilities, making them a natural extension of the user's thought process. The paradigm of user interaction with multiple models should shift from "using tools" to "leading a team," thereby unleashing the collaborative potential of human and model collectives in more complex intelligent tasks.

6.2 More Powerful Model vs Human-Centered System

While attempting to help users improve the efficiency of integrating model outputs, a question has consistently lingered around us: Could a single, more powerful model change this? A phenomenon observed in the first user study helped us answer this question: two users formed opposite judgments about the same model's capabilities. This revealed the significant variability that user preferences and personalization bring to the interaction experience. Although AI capabilities continue to advance across various aspects, it is challenging for any single model to meet the needs of every user.

The "power" of a single model does not equate to comprehensive coverage of user needs. Users' expectations and evaluation criteria for models are often highly subjective, stemming from their task contexts, cognitive habits, and even aesthetic preferences. For example, one user might prioritize the interpretability of generated results, while another may value creativity or execution efficiency more. Such differences mean that even when facing the same model, different users can have vastly different experiences and levels of satisfaction.

This further validates the fundamental significance of human-centered interaction systems like LLMartini: their goal is not merely to pursue the "optimal" performance of AI but to provide support from the user's perspective, enabling users to autonomously construct results that best suit their current tasks and personal preferences. During user interviews, we also received suggestions about a personalized fusion method. Therefore, future interaction designs should place greater emphasis on incorporating personalized factors into the system architecture, such as introducing more fine-grained user preference perception mechanisms and providing adaptive guidance for model recommendation and combination strategies. Only in this way can a deeper alignment between the diverse needs of different users and the technical capabilities of models be achieved.

6.3 Modalities beyond Text

Current large models have progressively achieved multimodal and even Agent capabilities. Although LLMartini has achieved effective integration and comparison results in the textual modality, it has also prompted us to delve deeper into the integration methods for other modalities. Compared to the structured and stable nature of text, the fusion mechanisms for modalities such as images, audio, and even video are far more complex.

From the perspective of output form, text inherently possesses a strong structural quality, allowing semantic units to be segmented and reorganized through methods like sentence or paragraph division, typically without compromising the overall meaning. In contrast, content such as images and audio often exists within a continuous, high-dimensional, and implicitly semantic structure. Simple cutting or fragmentation can easily disrupt their informational integrity and aesthetic consistency. For example, separating regions of an image may result in the loss of global compositional intent, while segmenting audio may break its emotional coherence.

Furthermore, from the perspective of user interaction, the editing and integration of multimodal content present entirely new challenges. Users not only need to address alignment issues between different modalities (such as image-text associations or audio-visual synchronization) but also must manage the manipulation and organization of multiple

media types simultaneously within the interface. Traditional text-centric interaction models are difficult to directly apply to multimodal contexts. How to design an intuitive and efficient multimodal workflow—enabling users to flexibly compare, combine, and even create cross-modal content—has become a critical issue requiring urgent exploration.

Therefore, future multi-model interaction platforms like LLMartini must not only architecturally support the input and output of heterogeneous modalities but also rethink the design of the entire interaction paradigm. Potential directions include developing interfaces that support multimodal annotation and dynamic preview, introducing non-destructive editing operations (such as layering, masking, and timeline editing), and even providing visual representations of cross-modal semantic associations. These advancements would help users effectively exercise creative judgment and maintain control in significantly more complex environments.

6.4 Limitation and Future Work

While LLMartini presents considerable advantages for facilitating multi-LLM interactions, our current research is subject to several limitations that suggest productive avenues for further investigation.

First, the existing framework for task classification and model recommendation is derived from a relatively small and heterogeneous sample (Study 1, N=19), which may introduce selection bias and constrain representativeness. Due to substantial variation in participant backgrounds, several task categories were underpinned by limited responses, potentially allowing the preferences of a narrow user subset to disproportionately sway model recommendations. A more robust and personalized approach should integrate a triad of factors: task characteristics, objectively evaluated model capabilities, and individualized user preferences. Future efforts will focus on expanding the dataset to improve the validity and generalizability of the classification and recommendation mechanisms.

Second, the evaluation of the system was conducted under controlled laboratory conditions (N=18). Although the findings yielded encouraging outcomes, they necessitate validation through larger-scale longitudinal studies conducted in authentic settings. Such research is critical to capturing emergent patterns of adoption, adaptive use over time, and the evolving needs of users when interacting with multi-LLM systems in real-world contexts.

Finally, the present study concentrates primarily on core functionalities such as output comparison and composition. However, participant feedback throughout the study underscored a compelling demand for more sophisticated interfaces that support advanced collaborative workflows among multiple LLMs. Moreover, extending this work to incorporate multimodal outputs (e.g., text, image, audio) and agent-based interactive frameworks represents a promising yet underexplored territory with substantial theoretical and practical implications.

7 Conclusion

In conclusion, to address the inefficiencies of manually comparing and combining outputs from multiple LLMs, we introduced LLMartini, a novel interactive system that enables seamless comparison and user-controlled composition through semantic segmentation and visual highlighting. A user study demonstrated that LLMartini significantly outperforms conventional methods by reducing task time, cognitive load, and increasing satisfaction. Our work highlights the importance of human-centered design in leveraging the complementary strengths of diverse AI models, providing a foundation for more efficient and creative multi-LLM interactions.

Acknowledgments

To Robert, for the bagels and for explaining CMYK and color spaces.

References

[1] 2024. GPT-4 Technical Report. arXiv:2303.08774 [cs.CL] https://arxiv.org/abs/2303.08774

[2] Sajid Ali, Tamer Abuhmed, Shaker El-Sappagh, Khan Muhammad, J. Alonso-Moral, R. Confalonieri, Riccardo Guidotti, J. Ser, N. Díaz-Rodríguez, and Francisco Herrera. 2023. Explainable Artificial Intelligence (XAI): What we know and what is left to attain Trustworthy Artificial Intelligence. *Information Fusion* (2023). doi:10.1016/j.inffus.2023.101805

[3] Saleema Amershi, Daniel S. Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi T. Iqbal, Paul N. Bennett, K. Quinn, J. Teevan, Ruth Kikin-Gil, and E. Horvitz. 2019. Guidelines for Human-AI Interaction. *International Conference on Human Factors in Computing Systems* (2019). doi:10.1145/3290605.3300233

[4] Daniel Barthel, Jonathan D. Hirst, Jacek Blazewicz, Edmund K. Burke, and Natalio Krasnogor. 2007. ProCKSI: a decision support system for Protein (Structure) Comparison, Knowledge, Similarity and Information. *BMC Bioinformatics* (2007). doi:10.1186/1471-2105-8-416

[5] Manuela Benary, Xing David Wang, Max Schmidt, D. Soll, G. Hilfenhaus, M. Nassir, C. Sigler, Maren Knödler, Ulrich Keller, D. Beule, Ulrich Keilholz, Ulf Leser, and D. Rieke. 2023. Leveraging Large Language Models for Decision Support in Personalized Oncology. *JAMA Network Open* (2023). doi:10.1001/jamanetworkopen.2023.43689

[6] W. Bingley, Caitlin Curtis, Steven Lockey, Alina Bialkowski, N. Gillespie, S. Haslam, R. Ko, Niklas K. Steffens, Janet Wiles, and Peter Worthy. 2022. Where is the human in human-centered AI? Insights from developer priorities and user experiences. *Computers in Human Behavior* (2022). doi:10.1016/j.chb.2022.107617

[7] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark Krass, Ranjay Krishna, Rohith Kuditipudi, Ananya Kumar, Faisal Ladhak, Mina Lee, Tony Lee, Jure Leskovec, Isabelle Levent, Xiang Lisa Li, Xuechen Li, Tengyu Ma, Ali Malik, Christopher D. Manning, Suvir Mirchandani, Eric Mitchell, Zanele Munyikwa, Suraj Nair, Avanika Narayan, Deepak Narayanan, Ben Newman, Allen Nie, Juan Carlos Niebles, Hamed Nilforoshan, Julian Nyarko, Giray Ogut, Laurel Orr, Isabel Papadimitriou, Joon Sung Park, Chris Piech, Eva Portelance, Christopher Potts, Aditi Raghunathan, Rob Reich, Hongyu Ren, Frieda Rong, Yusuf Roohani, Camilo Ruiz, Jack Ryan, Christopher Ré, Dorsa Sadigh, Shiori Sagawa, Keshav Santhanam, Andy Shih, Krishnan Srinivasan, Alex Tamkin, Rohan Taori, Armin W. Thomas, Florian Tramèr, Rose E. Wang, William Wang, Bohan Wu, Jiajun Wu, Yuhuai Wu, Sang Michael Xie, Michihiro Yasunaga, Jiaxuan You, Matei Zaharia, Michael Zhang, Tianyi Zhang, Xikun Zhang, Yuhui Zhang, Lucia Zheng, Kaitlyn Zhou, and Percy Liang. 2022. On the Opportunities and Risks of Foundation Models. arXiv:2108.07258 [cs.LG] https://arxiv.org/abs/2108.07258

[8] Rishi Bommasani, Percy Liang, and Tony Lee. 2023. Holistic Evaluation of Language Models. *Trans. Mach. Learn. Res.* (2023). doi:10.1111/nyas.15007

[9] John Brooke. 1996. SUS: A quick and dirty usability scale. In *Usability evaluation in industry*, Patrick W. Jordan, Bruce Thomas, Bernard A. Weerdmeester, and Ian L. McClelland (Eds.). Taylor & Francis, London, England, 189–194.

[10] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, J. Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Y. Lee, Yuan-Fang Li, Scott M. Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. 2023. Sparks of Artificial General Intelligence: Early experiments with GPT-4. *arXiv.org* (2023). doi:10.48550/arxiv.2303.12712

[11] Jason W. Burton, Ezequiel Lopez-Lopez, Shahar Hechtlinger, Zoe Rahwan, S. Aeschbach, Michiel A. Bakker, Joshua A Becker, Aleks Berditchevskaia, Julian Berger, Levin Brinkmann, Lucie Flek, Stefan M Herzog, Saffron Huang, Sayash Kapoor, Arvind Narayanan, Anne-Marie Nussberger, T. Yasseri, P. Nickl, Abdullah Almaatouq, Ulrike Hahn, Ralf H J M Kurvers, Susan Leavy, Iyad Rahwan, Divya Siddarth, Alice Siu, A. Woolley, Dirk U. Wulff, and Ralph Hertwig. 2024. How large language models can reshape collective intelligence. *Nature Human Behaviour* (2024). doi:10.1038/s41562-024-01959-9

[12] ByteDance. 2024. Doubao (Powered by Doubao LLM) [Large language model]. https://www.doubao.com/. Accessed: 2024-07-10.

[13] Lingjiao Chen, M. Zaharia, and James Y. Zou. 2023. FrugalGPT: How to Use Large Language Models While Reducing Cost and Improving Performance. *arXiv.org* (2023). doi:10.48550/arxiv.2305.05176

[14] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harrison Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth A. Barnes, Ariel Herbert-Voss, William H. Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew M. Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Samuel McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating Large Language Models Trained on Code. *arXiv: Learning* (2021).

[15] Weize Chen, Yusheng Su, Jingwei Zuo, Cheng Yang, Chenfei Yuan, Cheng Qian, Chi-Min Chan, Yujia Qin, Ya-Ting Lu, Ruobing Xie, Zhiyuan Liu, Maosong Sun, and Jie Zhou. 2023. AgentVerse: Facilitating Multi-Agent Collaboration and Exploring Emergent Behaviors in Agents. *arXiv.org* (2023). doi:10.48550/arxiv.2308.10848

[16] Zhijun Chen, Jingzheng Li, Pengpeng Chen, Zhuoran Li, Kai Sun, Yuankai Luo, Qianren Mao, Dingqi Yang, Hailong Sun, and Philip S. Yu. 2025. Harnessing Multiple Large Language Models: A Survey on LLM Ensemble. *arXiv.org* (2025). doi:10.48550/arxiv.2502.18036

[17] DeepSeek. 2024. DeepSeek Chat (Sep 10 version) [Large language model]. https://www.deepseek.com/. Accessed: 2024-09-10.

[18] Prasenjit Dey, S. Merugu, and S. Kaveri. 2025. Uncertainty-Aware Fusion: An Ensemble Framework for Mitigating Hallucinations in Large Language Models. *arXiv.org* (2025). doi:10.1145/3701716.3715523

[19] Thomas G. Dietterich. 2000. Ensemble Methods in Machine Learning. *multiple classifier systems* (2000). doi:10.1007/3-540-45014-9_1

[20] Liguo Fei, Tao Li, and Weiping Ding. 2026. Adaptive multi-source information fusion for intelligent decision-making in emergencies: Integrating personal, event, and environmental factors. *Information Fusion* 125 (2026), 103512. doi:10.1016/j.inffus.2025.103512

[21] Google. 2024. Gemini (1.5 Flash version) [Large language model]. https://gemini.google.com/. Accessed: 2024-06-01.

[22] Jiajing Guo, V. Mohanty, Jorge Piazentin Ono, Hongtao Hao, Liang Gou, and Liu Ren. 2024. Investigating Interaction Modes and User Agency in Human-LLM Collaboration for Domain-Specific Data Analysis. *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems* (2024). doi:10.1145/3613905.3651042

[23] Taicheng Guo, Yanjiao Chen, Yaqi Wang, Rui Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. 2024. Large Language Model based Multi-Agents: A Survey of Progress and Challenges. *Proceedings of the Thirty-ThirdInternational Joint Conference on Artificial Intelligence* (2024). doi:10.24963/ijcai.2024/890

[24] Shivam Gupta, Sachin Modgil, Samadrita Bhattacharyya, and Indranil Bose. 2021. Artificial intelligence for decision support systems in the field of operations research: review and future scope of research. *Annals of Operations Research* (2021). doi:10.1007/s10479-020-03856-6

[25] Sandra G. Hart and Lowell E. Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Advances in psychology*, P. A. Hancock and N. Meshkati (Eds.). Vol. 52. North-Holland, Amsterdam, 139–183. doi:10.1016/S0166-4115(08)62386-9

[26] Kenneth Holstein and Vincent Aleven. 2021. Designing for human-AI complementarity in K-12 education. *arXiv: Human-Computer Interaction* (2021). doi:10.1002/aaai.12058

[27] Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Ceyao Zhang, Jinlin Wang, Zili Wang, Steven Ka Shing Yau, Z. Lin, Liyang Zhou, Chenyu Ran, Lingfeng Xiao, Chenglin Wu, and Jürgen Schmidhuber. 2023. MetaGPT: Meta Programming for A Multi-Agent Collaborative Framework. *International Conference on Learning Representations* (2023).

[28] Yi-Chong Huang, Xiaocheng Feng, Baohang Li, Yang Xiang, Hui Wang, Bing Qin, and Ting Liu. 2024. Ensemble Learning for Heterogeneous Large Language Models with Deep Parallel Collaboration. *Neural Information Processing Systems* (2024).

[29] M. Iswahyudi, Nofirman Nofirman, I Ketut Adi Wirayasa, S. Suharni, and Ita Soegiarto. 2023. Use of ChatGPT as a Decision Support Tool in Human Resource Management. *Jurnal Minfo Polgan* (2023). doi:10.33395/jmp.v12i1.12869

[30] Maurice Jakesch, Jeffrey T. Hancock, and Mor Naaman. 2023. Human heuristics for AI-generated language are flawed. *Proceedings of the National Academy of Sciences* 120, 11 (2023), e2208839120. arXiv:https://www.pnas.org/doi/pdf/10.1073/pnas.2208839120 doi:10.1073/pnas.2208839120

[31] S. Jayatilake and G. U. Ganegoda. 2021. Involvement of Machine Learning Tools in Healthcare Decision Making. *Journal of Healthcare Engineering* (2021). doi:10.1155/2021/6679512

[32] L. Jian. 2021. Design of enterprise human resources decision support system based on data mining. *Soft Computing - A Fusion of Foundations, Methodologies and Applications* (2021). doi:10.21203/rs.3.rs-805084/v1

[33] Li Jian. 2022. Design of enterprise human resources decision support system based on data mining. *Soft Comput.* 26, 20 (Oct. 2022), 10571–10580. doi:10.1007/s00500-021-06659-4

[34] Dongfu Jiang, Xiang Ren, and Bill Yuchen Lin. 2023. LLM-Blender: Ensembling Large Language Models with Pairwise Ranking and Generative Fusion. *arXiv.org* (2023). doi:10.48550/arxiv.2306.02561

[35] Sanna Järvelä, Andy Nguyen, and Allyson Hadwin. 2023. Human and artificial intelligence collaboration for socially shared regulation in learning. *British Journal of Educational Technology* (2023). doi:10.1111/bjet.13325

[36] Yonggyun Kim. 2023. Comparing information in general monotone decision problems. *Journal of Economic Theory* (2023). doi:10.1016/j.jet.2023.105679

[37] Mina Lee, Percy Liang, Yang, Qian, Mina Lee, Percy Liang, Yang, and Qian. 2022. CoAuthor: Designing a Human-AI Collaborative Writing Dataset for Exploring Language Model Capabilities. *International Conference on Human Factors in Computing Systems* (2022). doi:10.1145/3491102.3502030

[38] Bo Li, Peng Qi, Bo Liu, Shuai Di, Jingen Liu, Jiquan Pei, Jinfeng Yi, and Bowen Zhou. 2021. Trustworthy AI: From Principles to Practices. *Comput. Surveys* (2021). doi:10.1145/3555803

[39] Jie Li, Hancheng Cao, Laura Lin, Youyang Hou, Ruihao Zhu, and Abdallah El Ali. 2023. User Experience Design Professionals' Perceptions of Generative Artificial Intelligence. *International Conference on Human Factors in Computing Systems* (2023). doi:10.1145/3613904.3642114

[40] Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. 2024. From Crowdsourced Data to High-Quality Benchmarks: Arena-Hard and BenchBuilder Pipeline. *arXiv.org* (2024). doi:10.48550/arxiv.2406.11939

[41] Wenzhe Li, Yong Lin, Mengzhou Xia, and Chi Jin. 2025. Rethinking Mixture-of-Agents: Is Mixing Different Large Language Models Beneficial? *arXiv.org* (2025). doi:10.48550/arxiv.2502.00674

[42] Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue Wang, Keshav Santhanam, Laurel Orr, Lucia Zheng, Mert Yuksekgonul, Mirac Suzgun, Nathan Kim, Neel Guha, Niladri Chatterji, Omar Khattab, Peter Henderson, Qian Huang, Ryan Chi, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. 2023. Holistic Evaluation of Language Models. arXiv:2211.09110 [cs.CL] https://arxiv.org/abs/2211.09110

[43] Q. Vera Liao, Daniel M. Gruen, and Sarah Miller. 2020. Questioning the AI: Informing Design Practices for Explainable AI User Experiences. *International Conference on Human Factors in Computing Systems* (2020). doi:10.1145/3313831.3376590

[44] Bill Yuchen Lin, Yuntian Deng, K. Chandu, Faeze Brahman, Abhilasha Ravichander, Valentina Pyatkin, Nouha Dziri, R. L. Bras, and Yejin Choi. 2024. WildBench: Benchmarking LLMs with Challenging Tasks from Real Users in the Wild. *arXiv.org* (2024). doi:10.48550/arxiv.2406.04770

[45] Siru Liu, A. Wright, B. Patterson, J. Wanderer, R. Turer, S. Nelson, A. McCoy, Dean F. Sittig, and Adam Wright. 2023. Using AI-generated suggestions from ChatGPT to optimize clinical decision support. *JAMIA Journal of the American Medical Informatics Association* (2023). doi:10.1093/jamia/ocad072

[46] Jinliang Lu, Ziliang Pang, Min Xiao, Yaochen Zhu, Rui Xia, and Jiajun Zhang. 2024. Merge, Ensemble, and Cooperate! A Survey on Collaborative Strategies in the Era of Large Language Models. *arXiv.org* (2024). doi:10.48550/arxiv.2407.06089

[47] Ming Y. Lu, Bowen Chen, Drew F. K. Williamson, Richard J. Chen, Melissa Zhao, Aaron K Chow, Kenji Ikemura, Ahrong Kim, Dimitra Pouli, Ankush Patel, Amr Soliman, Chengkuan Chen, Tong Ding, Judy J. Wang, Georg K. Gerber, Ivy Liang, L. Le, Anil V. Parwani, Luca L Weishaupt, and Faisal Mahmood. 2024. A multimodal generative AI copilot for human pathology. *Nature* (2024). doi:10.1038/s41586-024-07618-3

[48] Amelie Luhede, Prateek Verma, and Thorsten Upmann. 2025. Value of Information analysis for marine conservation decision making. *Environmental Science & Policy* (2025). doi:10.1016/j.envsci.2025.104164

[49] Bo Lv, Chen Tang, Yanan Zhang, Xin Liu, Ping Luo, and Yue Yu. 2024. URG: A Unified Ranking and Generation Method for Ensembling Language Models. In *Findings of the Association for Computational Linguistics: ACL 2024*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 4421–4434. doi:10.18653/v1/2024.findings-acl.261

[50] Ibomoiye Domor Mienye and Theo G. Swart. 2025. Ensemble Large Language Models: A Survey. *Information* (2025). doi:10.3390/info16080688

[51] Tim Miller. 2023. Explainable AI is Dead, Long Live Explainable AI! (2023). doi:10.1145/3593013.3594001

[52] OpenAI. 2023. ChatGPT (Mar 20 version) [Large language model]. https://chat.openai.com/chat. Accessed: 2023-09-15.

[53] Alicia Parrish, Angelica Chen, Nikita Nangia, Vishakh Padmakumar, Jason Phang, Jana Thompson, Phu Mon Htut, and Sam Bowman. 2021. BBQ: A hand-built bias benchmark for question answering. *Findings* (2021). doi:10.18653/v1/2022.findings-acl.165

[54] Pat Pataranutaporn, Valdemar Danry, Joanne Leong, Parinya Punpongsanon, Dan Novy, Pattie Maes, and Misha Sra. 2021. AI-generated characters for supporting personalized learning and well-being. *Nature Machine Intelligence* (2021). doi:10.1038/s42256-021-00417-9

[55] Sebastian Raisch and Kateryna Fomina. 2024. Combining Human and Artificial Intelligence: Hybrid Problem-Solving in Organizations. *Academy of Management Review* (2024). doi:10.5465/amr.2021.0421

[56] Arya S Rao, John Kim, Meghana Kamineni, Michael Pang, Winston Lie, Keith J. Dreyer, and Marc D. Succi. 2023. Evaluating GPT as an Adjunct for Radiologic Decision Making: GPT-4 Versus GPT-3.5 in a Breast Imaging Pilot. *Journal of the American College of Radiology* (2023). doi:10.1016/j.jacr.2023.05.003

[57] Lee Reicher, Guy Lutsker, Nadav Michaan, Dan Grisaru, and Ido Laskov. 2024. Exploring the role of artificial intelligence, large language models: Comparing patient-focused information and clinical decision support capabilities to the gynecologic oncology guidelines. *International journal of gynaecology and obstetrics: the official organ of the International Federation of Gynaecology and Obstetrics* (2024). doi:10.1002/ijgo.15869

[58] Jeba Rezwana and M. Maher. 2022. Designing Creative AI Partners with COFI: A Framework for Modeling Interaction in Human-AI Co-Creative Systems. *International Conference on Innovative Computing and Cloud Computing* (2022). doi:10.1145/3519026

[59] Riccardo Rosati, Ville Kyrki, Germana Cecchini, Flavio Tonetto, Paolo Viti, Adriano Mancini, Emanuele Frontoni, Riccardo Rosati, Ville Kyrki, Germana Cecchini, Flavio Tonetto, Paolo Viti, Adriano Mancini, and Emanuele Frontoni. 2022. From knowledge-based to big data analytic model: a novel IoT and machine learning based decision support system for predictive maintenance in Industry 4.0. *Journal of Intelligent Manufacturing* (2022). doi:10.1007/s10845-022-01960-x

[60] Omer Sagi and Lior Rokach. 2018. Ensemble learning: A survey. *Wiley interdisciplinary reviews: data mining and knowledge discovery* 8, 4 (2018), e1249.

[61] Zhang Shao, Xihuai Wang, Wenhao Zhang, Yongshan Chen, Lei Gao, Dakuo Wang, Weinan Zhang, Xinbing Wang, and Ying Wen. 2024. Mutual Theory of Mind in Human-AI Collaboration: An Empirical Study with LLM-driven AI Agents in a Real-time Shared Workspace Task. *arXiv (Cornell University)* (2024). doi:10.48550/arxiv.2409.08811

[62] Ashish Sharma, Inna Wanyin Lin, Adam S. Miner, David C. Atkins, and Tim Althoff. 2023. Human–AI collaboration enables more empathic conversations in text-based peer-to-peer mental health support. *Nature Machine Intelligence* (2023). doi:10.1038/s42256-022-00593-2

[63] Yingtian Shi, Xiaoyi Liu, Chun Yu, Tianao Yang, Cheng Gao, Chen Liang, and Yuanchun Shi. 2024. Bridging the gap between natural user expression with complex automation programming in smart homes. arXiv:2408.12687 [cs.HC] https://arxiv.org/abs/2408.12687

[64] Jung P. Shim, Merrill Warkentin, James F. Courtney, Daniel J. Power, Ramesh Sharda, and Christer Carlsson. 2002. Past, present, and future of decision support technology. *Decision Support Systems* (2002). doi:10.1016/s0167-9236(01)00139-7

[65] Alexander V. Smirnov, Tatiana Levashova, and Nikolay Shilov. 2015. Patterns for context-based knowledge fusion in decision support systems. *Information Fusion* (2015). doi:10.1016/j.inffus.2013.10.010

[66] Roland Somlai. 2022. Integrating decision support tools into businesses for sustainable development: A paradoxical approach to address the food waste challenge. *Business Strategy and the Environment* (2022). doi:10.1002/bse.2972

[67] Sangho Suh, Meng Chen, Bryan Min, T. Li, and Haijun Xia. 2023. Luminate: Structured Generation and Exploration of Design Space with Large Language Models for Human-AI Co-Creation. *International Conference on Human Factors in Computing Systems* (2023). doi:10.1145/3613904.3642400

[68] John Sweller. 2011. CHAPTER TWO - Cognitive Load Theory. Psychology of Learning and Motivation, Vol. 55. Academic Press, 37–76. doi:10.1016/B978-0-12-387691-1.00002-8

[69] Shengji Tang, Jianjian Cao, Weihao Lin, Jiale Hong, Bo Zhang, Shuyue Hu, Lei Bai, Tao Chen, Wanli Ouyang, and Peng Ye. 2025. Open-Source LLMs Collaboration Beats Closed-Source LLMs: A Scalable Multi-Agent System. arXiv:2507.14200 [cs.CL] https://arxiv.org/abs/2507.14200

[70] Selim Furkan Tekin, Fatih İlhan, Tiansheng Huang, Sihao Hu, and Ling Liu. 2024. LLM-TOPLA: Efficient LLM Ensemble by Maximising Diversity. (2024). doi:10.18653/v1/2024.findings-emnlp.698

[71] Yan Tu, W. J. Wang, Zhuang Ma, Zongmin Li, and Benjamin Lev. 2025. Multi-Agent Reinforcement Learning-based consensus building for Large-Scale Group Decision Making. *Information Fusion* (2025). doi:10.1016/j.inffus.2025.103629

[72] Salih Tutun, Marina Johnson, Abdulaziz Ahmed, Abdullah Albizri, Sedat Irgil, Ilker Yesilkaya, Esma Ucar, Tanalp Sengun, Antoine Harfouche, Salih Tutun, Marina Johnson, Abdulaziz Ahmed, Abdullah Albizri, Sedat Irgil, Ilker Yesilkaya, Esma Ucar, Tanalp Sengun, and Antoine Harfouche. 2022. An AI-based Decision Support System for Predicting Mental Health Disorders. *Information Systems Frontiers* (2022). doi:10.1007/s10796-022-10282-5

[73] Maria Virvou, G. Tsihrintzis, and Evangelia-Aikaterini Tsichrintzi. 2024. VIRTSI: A novel trust dynamics model enhancing Artificial Intelligence collaboration with human users – Insights from a ChatGPT evaluation study. *Information Sciences* (2024). doi:10.1016/j.ins.2024.120759

[74] Fanqi Wan, Xinting Huang, Deng Cai, Xiaojun Quan, Wei Bi, and Shuming Shi. 2024. Knowledge Fusion of Large Language Models. *arXiv.org* (2024). doi:10.48550/arxiv.2401.10491

[75] Hao Wang, Qingling Ding, Yibing Luo, Zixuan Wu, Jiahao Yu, Huizhi Chen, Yubin Zhou, He Zhang, Kai Tao, Xiaoliang Chen, Jun Fu, and Jin Wu. 2023. High-Performance Hydrogel Sensors Enabled Multimodal and Accurate Human–Machine Interaction System for Active Rehabilitation. *Advances in Materials* (2023). doi:10.1002/adma.202309868

[76] Junlin Wang, Jue Wang, Ben Athiwaratkun, Ce Zhang, and James Zou. 2024. Mixture-of-Agents Enhances Large Language Model Capabilities. *arXiv.org* (2024). doi:10.48550/arxiv.2406.04692

[77] Wei, Jason, Wang, Xuezhi, Schuurmans, Dale, Bosma, Maarten, Chi, Ed, Le, Quoc, Zhou, and Denny. 2022. Chain of Thought Prompting Elicits Reasoning in Large Language Models. *Neural Information Processing Systems* (2022). doi:10.48550/arxiv.2201.11903

[78] Peng Wu, Yutong Xie, Ligang Zhou, Muhammet Deveci, and Luis Martínez. 2025. Multi-source data driven decision-support model for product ranking with consumer psychology behavior. *Information Fusion* (2025). doi:10.1016/j.inffus.2025.103014

[79] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. 2023. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation Framework. *arXiv.org* (2023). doi:10.48550/arxiv.2308.08155

[80] Tongshuang Wu, Michael Terry, Carrie J. Cai, Tongshuang Wu, Michael Terry, and Carrie J. Cai. 2022. AI Chains: Transparent and Controllable Human-AI Interaction by Chaining Large Language Model Prompts. *International Conference on Human Factors in Computing Systems* (2022). doi:10.1145/3491102.3517582

[81] Tianbao Xie, Fan Zhou, Zhoujun Cheng, Peng Shi, Luoxuan Weng, Yitao Liu, Toh Jing Hua, Junning Zhao, Qian Liu, Che Liu, Leo Z. Liu, Yiheng Xu, Hongjin Su, Dongchan Shin, Caiming Xiong, and Tao Yu. 2023. OpenAgents: An Open Platform for Language Agents in the Wild. *arXiv.org* (2023). doi:10.48550/arxiv.2310.10634

[82] Wei Xu, M. Dainoff, Liezhong Ge, and Zaifeng Gao. 2021. From Human-Computer Interaction to Human-AI Interaction: New Challenges and Opportunities for Enabling Human-Centered AI. *arXiv.org* (2021).

[83] Wei Xu, Marvin J. Dainoff, Liezhong Ge, and Zaifeng Gao. 2022. Transitioning to Human Interaction with AI Systems: New Challenges and Opportunities for HCI Professionals to Enable Human-Centered AI. *International journal of human computer interactions* (2022). doi:10.1080/10447318.2022.2041900

[84] Yangyifan Xu, Jinliang Lu, and Jiajun Zhang. 2024. Bridging the Gap between Different Vocabularies for LLM Ensemble. *North American Chapter of the Association for Computational Linguistics* (2024). doi:10.48550/arxiv.2404.09492

[85] Jackie (Junrui) Yang, Yingtian Shi, Chris Gu, Zhang Zheng, Anisha Jain, Tianshi Li, Monica S. Lam, and James A. Landay. 2025. GenieWizard: Multimodal App Feature Discovery with Large Language Models. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. Association for Computing Machinery, New York, NY, USA, Article 936, 17 pages. doi:10.1145/3706598.3714327

[86] Qian Yang, Aaron Steinfeld, C. Rosé, and J. Zimmerman. 2020. Re-examining Whether, Why, and How Human-AI Interaction Is Uniquely Difficult to Design. *International Conference on Human Factors in Computing Systems* (2020). doi:10.1145/3313831.3376301

[87] Yuxuan Yao, Han Wu, Mingyang Liu, Sichun Luo, Xiongwei Han, Jie Liu, Zhijiang Guo, and Linqi Song. 2024. Determine-Then-Ensemble: Necessity of Top-k Union for Large Language Model Ensembling. *arXiv.org* (2024). doi:10.48550/arxiv.2410.03777

[88] Yifei Yu, D. Yazan, Veronica Junjan, and M. Iacob. 2022. Circular economy in the construction industry: A review of decision support tools based on Information & Communication Technologies. *Journal of Cleaner Production* (2022). doi:10.1016/j.jclepro.2022.131335

[89] J.D. Zamfirescu-Pereira, Richmond Y. Wong, Bjoern Hartmann, and Qian Yang. 2023. Why Johnny Can't Prompt: How Non-AI Experts Try (and Fail) to Design LLM Prompts. *International Conference on Human Factors in Computing Systems* (2023). doi:10.1145/3544548.3581388

[90] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, E. Xing, Haotong Zhang, Joseph Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. *arXiv.org* (2023). doi:10.48550/arxiv.2306.05685

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009