# An LLMs-Enhanced Multi-Agent Feedback Modeling Framework for Intelligent Public Funding Allocation

**Junqiao Gong[1], Jiaxiao Wang[2], Gang Wang[1], Jian Ma[2*], Xuansong Tam[3]**
[1]*School of Management, Hefei University of Technology, Hefei, China,*
[2]*Department of Information Systems, City University of Hong Kong, Hong Kong, China,*
[3]*Department of Economics and Finance, City University of Hong Kong, Hong Kong, China*
**Corresponding Author*

**Abstract: With the continuous improvement of e-government platforms and the rapid development of artificial intelligence technology, an innovative, government-led model for public funding allocation has emerged, which achieves precise allocation of public fundings through automatic filling of application forms and intelligent verification of enterprise qualifications. However, existing studies still lacks systematic modeling of the implementation mechanism of this public fund allocation model. In response to the inherent complexity of the operating logic of this model and the multi-source heterogeneity of policy texts and enterprise data, this paper innovatively proposes a large language models-based multi-agent feedback modeling framework for intelligent public funding allocation. This method consists of three core intelligent agents: the policy agent, enterprise agent, and matching agent. Among them, the policy agent is responsible for parsing policy texts and simulating government decision-making processes. The enterprise agent automatically generates funding application forms by integrating structured and unstructured enterprise data. The match agent is responsible for outputting the matching results and their decision reasons. Experimental evaluation conducted on a real-world dataset confirms the efficacy and practical value of the proposed framework, demonstrating its superiority over the baseline methods.**

**Keywords: Public Funding Allocation; Multi-Agent Feedback Modeling; Large Language Models**

## 1. Introduction

Public funding allocation serves as a key governmental instrument for stimulating enterprise development and promoting industrial upgrading [1-3]. In practice, governments typically issue business support policies and provide eligible firms with financial, informational, or service-oriented support to enhance growth, innovation capabilities, and competitiveness [2]. The prevailing allocation process follows a policy release–enterprise application model. Specifically, the government publishes policy documents and application templates, and then enterprises assess their eligibility and internally prepare application materials, and government agencies subsequently organize expert committees to evaluate applications and determine funding recipients [4]. Despite its widespread use, this traditional enterprise-initiated application model suffers from several limitations that weaken the effectiveness of public fundings in supporting enterprise development. Firstly, many enterprises do not employ dedicated grant-application specialists, instead, non-specialist staff members are temporarily assigned to prepare application materials. Due to their limited understanding of policy requirements and application criteria, the submitted materials are often incomplete or imprecise, reducing the likelihood of successful funding [4]. Secondly, to increase their chances of success, some firms hire third-party intermediaries to prepare applications. These intermediaries typically charge a percentage of the awarded fundings as compensation, meaning that a portion of the public fundings intended for enterprise development is diverted, thereby lowering the effective utilization of government subsidies. To ensure that public fundings genuinely reach enterprises that need them and to improve the overall efficiency of resource allocation, an emerging alternative model has been proposed, which can be called the government-led public funding allocation model [1]. Specifically, governments publish business support policies, but enterprises no longer need to initiate the application process. Instead, governments proactively identify eligible firms using existing enterprise data and automatically allocate fundings. This model has the potential to reduce procedural frictions, eliminate intermediary distortions, and achieve more precise and effective policy implementation. Therefore, this model has become a research focus in both academic and industrial domains.

The allocation of public fundings necessitates that governments deploy financial resources in accordance with specific policy objectives and within defined budgetary constraints [3, 5]. Existing public funding allocation methods can generally be classified into traditional methods and data-driven methods [6-8]. Traditional methods primarily include the baseline methods, incremental budgeting methods, and formula-based methods [7]. The baseline methods are determined by adjusting historical funding patterns [6]. The incremental budgeting allocation methods are derived by applying predetermined proportional changes to the previous year's budget [9]. For example, Newman [5] examined the underlying factors that contributed to the breakdown of the norm-based budgeting system, as well as the deeper forces that have driven higher education institutions in Ghana to adopt incremental budgeting practices. Reddick [9] proposed a hybrid budgeting method that integrates incremental budgeting with rational, analytical methods in order to enhance the effectiveness and rigor of budgetary decision-making. Both methods rely heavily on administrative experience and discretionary judgment, thereby exhibiting a considerable degree of subjectivity. In contrast, formula-based allocation methods enhance methodological rigor and transparency through standardized indicators and systematic procedures. As a structured decision-making framework, formula-based allocation typically involves two critical phases [10]. Firstly, they need to identify key variables, such as supply-side, demand-side, and cost-related factors. Secondly, they determine optimal weights for these variables through statistical techniques, including regression analysis [3]. For example, Walters [3] proposed a statistically grounded strategy for weighting and combining variables in the construction of fund allocation formulas, which leverages current, historical, or hypothetical allocation data to automatically generate the structure and weights of the formula, thereby enhancing the systematicity and rigor of the allocation process. Hadar [10] introduced a composite funding formula that not only incorporates traditional equity-based considerations but also integrates a component designed to reward improvements in the distribution of educational achievement across schools. Although formula-based methods have been widely adopted in sectors such as healthcare and education funding, they still face inherent limitations. These include the reinforcement of existing allocation patterns and limited adaptability to dynamic fiscal conditions.

In recent years, with the rapid advancement of artificial intelligence technologies, data-driven allocation methods have attracted increasing scholarly attention due to their potential to address the limitations inherent in traditional methods [11, 12]. These methods leverage machine learning and deep learning techniques to extract salient features from multi-source heterogeneous data, such as policy texts and enterprise operational information, and subsequently construct predictive and interpretable models that enhance both allocative efficiency and distributive equity [13]. For example, Jang [12] introduced a novel budget allocation method, which begins by employing machine learning methods to generate precise forecasts of future research output. Building on these predictions, it then incorporates a robust optimization framework to effectively manage the uncertainties inherent in the estimated output values. Prasetyo and Suharjito [8] proposed a machine learning method for public funding allocation, which first employs principal component analysis to extract and select salient features from budget implementation quality indicator data, and subsequently constructs a series of ensemble learning models using these selected features. Nevertheless, existing data-driven methods often struggle to accommodate emerging government-led public funding allocation model. Their limitations are primarily manifested in the following challenges:

Firstly, in contrast to the traditional enterprise-initiated public funding allocation model, this new allocation model represents a fundamental shift toward a government-led allocation mechanism. The core innovation of this model lies in systematically integrating multi-source heterogeneous data with advanced artificial intelligence methods, thereby reconstructing the traditionally manual application process into an automated, data-driven system. From a systems engineering perspective, implementing this novel public funding allocation model necessitates the deconstruction of the public fund allocation process into multiple logically interconnected functional modules [14, 15]. These primarily include the intelligent comprehension of policy texts, automated generation of application materials, and dynamic verification of enterprise eligibility. To ensure the effective operation of this complex system, specialized technical solutions must be designed for each functional module. Multi-agent systems, with their distinct advantages in decomposing complex tasks and facilitating collaborative problem-solving [16-18], offer a promising research direction for addressing this challenge. Within this framework, a key challenge emerges: how to design mutually independent yet organically coordinated sub-task modules through the task decomposition mechanism of multi-agent systems to construct an intelligent public fund allocation solution.

Secondly, public fund allocation involves processing multi-source heterogeneous data, predominantly unstructured textual data such as policy documents and corporate reports. Traditional deep learning methods demonstrate limited capability in extracting deep semantic relationships from such data [15]. Notably, recent breakthroughs in large language models (LLMs) have demonstrated remarkable text comprehension and generation capabilities, achieving significant success in domains such as financial text analysis and e-commerce review processing [18, 19]. This study proposes to innovatively leverage the semantic understanding and knowledge distillation capabilities of LLMs for policy document analysis and enterprise profiling tasks. However, public funding allocation involves, on the one hand, analyzing policy elements and enterprise characteristics in alignment with allocation objectives, and on the other hand, operates under rigid budgetary constraints, meaning that not all applicants meeting the fundamental eligibility criteria can be funded. Government must further evaluate enterprises based on their qualifications, growth potential, and strategic value within industrial chains and regional synergies, thereby maximizing the utility of limited funds. This presents a core challenge: how to develop an LLMs-based feature analysis framework that can achieve precise policy-enterprise matching by holistically considering the above two aspects.

In this paper, we propose an innovative large language model-based multi-agent feedback modeling framework (LLM-MAFM) to effectively address key challenges in public funding allocation. The framework consists of three core modules: a policy agent, an enterprise agent, and a matching agent. Specifically, first, the policy agent semantically deconstructs policy texts from the perspective of funding allocation tasks, focusing on three dimensions: relevance, quality, and connectiveness. Simultaneously, the enterprise agent integrates the enterprise data to automatically generate funding application forms based on standardized templates, and then maps the application indicators to the three dimensions of relevance, quality, and correlation to facilitate policy-enterprise matching. Subsequently, the matching agent generates interpretable matching results with supporting evidence through multidimensional matching analysis. These results are then fed back to both the policy agent and the enterprise agent to further update them. For experimental validation, we collected a comprehensive dataset of enterprise support policies from various cities in China. The experimental results demonstrate that our proposed method achieves a performance improvement of 6.56% in Precision@30 and 4.15% in Recall@30 compared to the strongest baseline, DeepSeek-V3. This significant enhancement validates the effectiveness of our proposed method.

The contributions of this paper are as follows:

Firstly, we propose a new model for public funding allocation that fundamentally transforms the traditional public funding allocation model, establishing a government-led intelligent allocation paradigm. This contribution provides important insights for the digital transformation of public services.

Secondly, we propose an innovative LLMs-based multi-agent feedback modeling framework for intelligent public funding allocation. The policy agent employs LLMs to decode policy documents and incorporates a government decision-making simulator to accurately capture the underlying policy intent. The enterprise agent integrates enterprise data to automatically complete funding applications. The matching agent employs a three-dimensional evaluation system, encompassing relevance, quality, and connectivity, to not only produce final allocation decisions but also generate interpretable feedback for policy comprehension and optimization feedback for application refinement. This feedback mechanism drives iterative refinements of both the policy and enterprise agents.

Thirdly, we conduct systematic experiments using real-world policy datasets from across China, along with enterprise data from Qichacha's commercial database. The results demonstrate our method's significant advantages in public funding allocation.

The remainder of this paper is organized as follows. Section 2 introduces the methodology, presenting the proposed framework in detail. Section 3 outlines the experimental setup. Section 4 presents the experimental results and discussions. Finally, section 5 summarizes the key findings and outlines future directions for research.

## 2. Methodology

### 2.1 Overall Framework

This section presents a novel LLMs-enhanced multi-agent feedback modeling framework for intelligent public funding allocation, as illustrated in Figure 1.

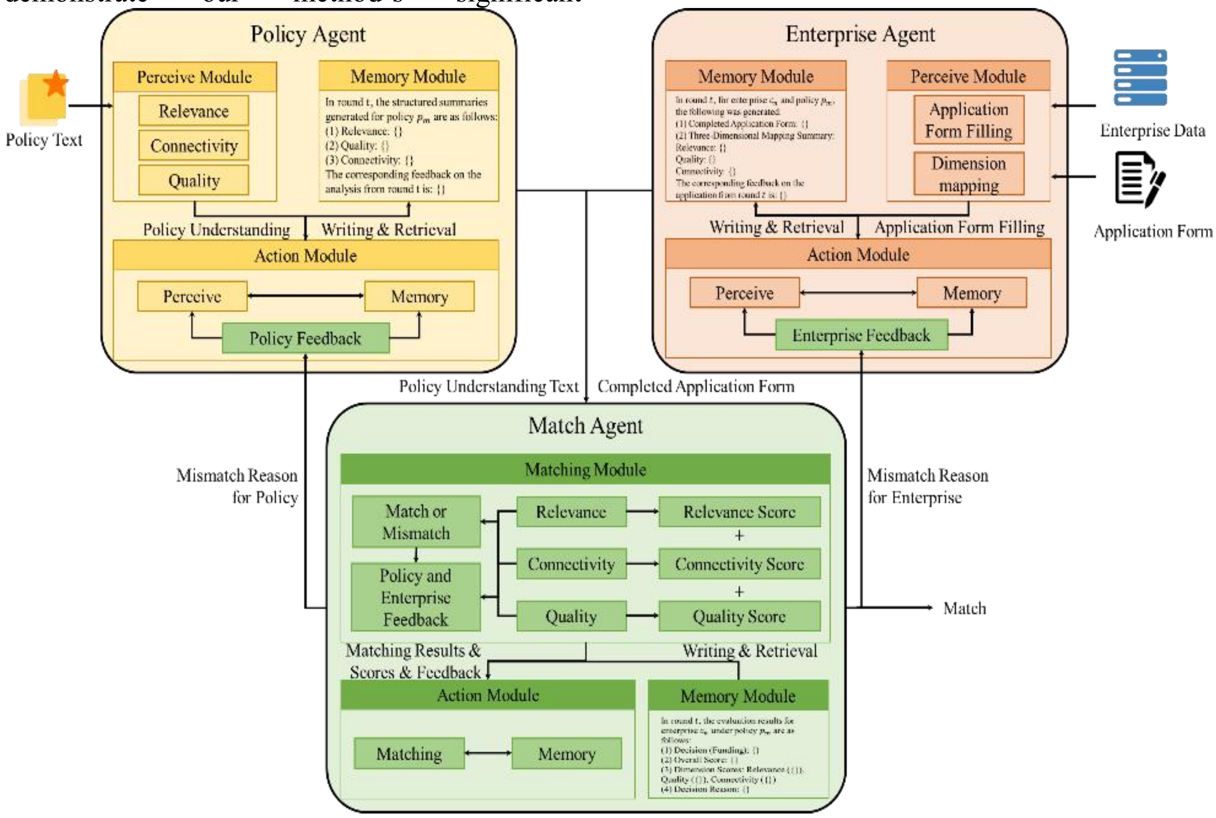


**Figure 1. The Overall Framework of the Proposed LLM-MAFM Method**

The method includes three agents: the policy agent, enterprise agent and matching agent. Specifically, the policy agent is responsible for comprehensively interpreting policy documents by evaluating them along three dimensions, including relevance, connectivity, and quality [20]. This enables the construction of a structured and semantically rich representation of policy intent. The enterprise agent automatically extracts essential information from enterprise data to generate a complete and properly structured application document, thereby replacing manual form filling and reducing human-induced variability. The match agent performs multi-dimension matching between the policy understanding text produced by the policy agent and the application document generated by the enterprise agent. Based on the matching results, it produces targeted feedback for both the policy agent and the enterprise agent. These agents then refine their policy interpretation and application generation processes accordingly, forming a closed-loop multi-agent feedback cycle that continuously enhances system performance.

In this paper, to accurately describe our proposed method, the basic definitions are as follows. We assume that there are $M$ public funding policies, and $N$ enterprises in the public funding allocation. Let $P=\{p_1,p_2,\ldots,p_m,\ldots,p_M\}$ denotes the set of public funding policies, and $p_m,m\in\{1,2,3,\ldots,M\}$ represents the $m$-th public funding policy. The textual content of policy $p_m$ is denoted by $T_m$. Similarly, let $C=\{c_1,c_2,\ldots,c_n,\ldots,c_N\}$ denotes the set of enterprises, and $c_n,n\in\{1,2,3,\ldots,N\}$ represents the $n$-th enterprise. The enterprise data can be represented as $E_m$. $Y\in\mathbb{R}^{M\times N}$ represents the funding matrix, that is if the company $c_n$ is funded by the public fund policy $p_m$, $y_{mn}=1$, else $y_{mn}=0$. This paper aims to predict the values that do not exist in the funding matrix $Y$.

### 2.2 Policy Agent

The policy agent is primarily responsible for analyzing policy texts and progressively refining its understanding based on feedback from the match agent, thereby enabling deeper comprehension of policy content. The agent comprises a perception module, a memory module, and an action module. The perception module conducts in-depth interpretation of policy texts, while the memory module stores the agent's iterative understandings together with the corresponding feedback. The action module formally defines the operational procedures that govern the agent's behavior. The prompt template of the policy agent is presented in Table 1.

**Table 1. Prompt template for the Policy Agent**

| Memory Template |
|---|
| In round $t$, the structured summaries generated for policy $p_m$ are as follows:<br>(1) Relevance: {}<br>(2) Quality: {}<br>(3) Connectivity: {}<br>The corresponding feedback on the analysis from round t is: {} |
| Prompt Template |
| **Role**: As a policy interpretation expert at the Government Funding Office, you are well-versed in policies related to science and technology, industry, and regional development. Your task is to analyze policy documents and provide structured summaries.<br>**Task**: Generate three structured summaries for the following original policy text, taking into account the feedback from the previous round (if any), each approximately 80 to 120 words in length:<br>**(1) Relevance**: Summarize the policy's core objectives, main support areas, and the specific types of entities (e.g., enterprises, institutions) that are eligible.<br>**(2) Quality**: Outline the key qualification standards, financial thresholds, and technical capability requirements that applicants must meet.<br>**(3) Connectivity**: Describe the requirements or expectations for industrial chain collaboration, regional synergy, platform co-construction, or cross-departmental cooperation.<br>**Inputs**:<br>**(1) Original policy text**: {}<br>**(2) Previous round's feedback**: {} (Write "None" if this is the first round or no feedback exists.)<br>**Useful Tips**:<br>(1) Prioritize addressing any specific issues, omissions, or corrections pointed out in the previous round's feedback.<br>(2) Summarize concisely, retaining all key numerical data, dates, technical terms, and proper nouns from the original policy.<br>(3) Focus strictly on extracting and organizing information as per the three dimensions from the given text and feedback.<br>**Output**: Your response must be a valid JSON object with the keys "Relevance", "Quality", and "Connectivity", and nothing else. |

#### 2.2.1 Perceive Module

In the allocation of public funds, governments typically evaluate applicant enterprises across three key dimensions: relevance, quality, and connectivity, thereby ensuring that public fundings are directed toward enterprises that best align with policy objectives and generate the greatest public value. To align with this decision-making logic, this study interprets policy texts from the perspective of how governments assess enterprises, enabling a more precise extraction of policy intent and its mapping onto enterprise characteristics. Specifically, the relevance evaluates the alignment between the policy's intended support direction and its articulated objectives, encompassing elements such as the policy title, stated goals, scope of application, and the industrial domains it targets. The quality captures the policy's substantive eligibility and capability requirements for potential applicants, including eligibility criteria, evaluation standards, compliance conditions, and expectations regarding project implementation capacity. The connectivity reflects the degree to which the policy is embedded within broader industrial ecosystems, regional development strategies, and value-chain coordination frameworks. Policy documents often articulate requirements or incentives related to industrial-chain collaboration, regional cluster development, cross-agency cooperation, or platform-building initiatives. By systematically interpreting policy texts through these three evaluative dimensions, this study seeks to more accurately reconstruct the underlying logic of governmental funding decisions, thereby enhancing the precision of policy understanding and the effectiveness of policy behavior modeling.

Building upon this multi-dimensional framework, we employ the LLMs to process raw policy texts as inputs, facilitating granular semantic parsing and structured representation of policy documents. This modeling method not only enables machine-readable transformation of policy elements but also elucidates the micro-level decision-making mechanisms underlying corporate policy comprehension.

$$(R_m^t, Q_m^t, C_m^t) = f_{p-p}\left(T_m, MP_m^{t-1}\right) \quad (1)$$

where $f_{p-p}$ represents the perceive module of the policy agent, $R_m^t, Q_m^t$ and $C_m^t$ represent the relevance output, quality output and connectivity output in the $t$-th iteration from the policy $p_m$, respectively. $MP_m^{t-1}$ denotes the $t-1$-th iteration memory from the policy agent.

2.2.2 Memory and action Module

The memory module is designed to store the policy-understanding texts generated by the policy agent at each iteration, along with the corresponding feedback provided by the match agent, thereby supplying essential information for the action module to read and update. $MP_m^{t-1}$ includes the three aspects policy-understanding texts $R_m^{t-1}, Q_m^{t-1}, C_m^{t-1}$ and the policy feedback $FP_m^{t-1}$ from match agent. This module is further intended to enable the perception module, during each new round of policy interpretation, to draw upon the previously stored understanding results and feedback, thereby facilitating continuous refinement and iterative enhancement of the agent's comprehension of policy documents.

The action module defines the operational logic of the policy agent. Taking the $t$-th iteration as an example, the inputs to the policy agent include the raw policy text as well as the policy-understanding text and the policy feedback stored in the memory module during iteration $t-1$. In this iteration, the action module first invokes the memory module to retrieve the previously stored policy-understanding results and the policy feedback. It then calls the perceive module, which reinterprets the prior understanding based on both the retrieved feedback and the policy text, thereby producing updated policy-understanding text $R_m^t, Q_m^t, C_m^t$. After the reinterpretation process, the action module writes the newly generated policy-understanding text back into the memory module. Finally, if the matching agent produces new policy feedback during the current iteration, the action module also records this feedback into the memory module, ensuring continuity and completeness in the agent's evolving policy comprehension.

### 2.3 Enterprise Agent

The Enterprise Agent is designed to automate the population of application forms by automatically extracting essential fields from enterprise data and mapping key indicators in the completed forms onto the relevance, quality, and connectivity dimensions to support policy–enterprise matching performed by the matching agent. The operational framework consists of a perception module, a memory module, and an action module. The perception module is responsible for analyzing enterprise data, executing the automated form-filling process, and subsequently mapping the populated fields onto the relevance, quality, and connectivity dimensions. The memory module functions as the system's information repository, storing completed application forms as well as the enterprise feedback returned by the matching agent. The action module serves as the central coordinator, defining and orchestrating the execution sequence across modules. It activates the perception and memory modules in a structured manner to ensure the coherent and efficient completion of the automated form-filling task. The prompt template of the enterprise agent is shown in Table 2.

**Table 2. Prompt template for the Enterprise Agent**

| Memory Template |
|---|
| In round $t$, for enterprise $c_n$ and policy $p_m$, the following was generated:<br>(1) Completed Application Form: {}<br>(2) Three-Dimensional Mapping Summary:<br>Relevance: {}<br>Quality: {}<br>Connectivity: {}<br>The corresponding feedback on the application from round t is: {} |
| Prompt Template |
| **Role:** As an enterprise data analyst, you are proficient in extracting and synthesizing information from diverse sources including business operations, financial reports, intellectual property (e.g., patents), financing records, and public filings. Your task is to accurately complete policy application forms.<br>**Task:** Based on the provided enterprise data and the target policy's application form template, automatically fill in all required fields. Then, generate a concise three-dimensional mapping summary (each summary ≤ 100 words) that aligns the enterprise profile with the policy's evaluation framework, considering any feedback from the previous round.<br>**Inputs:**<br>**(1) Enterprise Data:** {}<br>**(2) Application Form Template:** {}<br>**(3) Previous Round's Feedback:** {} (Write "None" if this is the first round or no feedback exists.)<br>**Useful Tips:**<br>(1) Strictly adhere to the form template's structure. Fill each field with the most accurate and specific information available in the enterprise data. Maintain original units for numerical fields (e.g., "RMB 5 million", "3 patents").<br>(2) Prioritize feedback correction: If the previous round's feedback points out missing fields, ambiguous descriptions, or factual inaccuracies, ensure these are addressed and corrected in this round's output.<br>(3) Prohibition on fabrication: Do not generate, infer, or fabricate any information that cannot be directly supported or logically deduced from the provided enterprise data. If data for a field is unavailable, indicate this clearly (e.g., "Data not available" or leave as instructed by the template).<br>(4) For the Dimension Mapping, synthesize information to reflect:<br>**1) Relevance:** The alignment between the enterprise's core business/activities and the policy's stated objectives and supported areas.<br>**2) Quality:** The enterprise's quantitative and qualitative qualifications, such as financial health (e.g., revenues), R&D investment, certifications, and technical capabilities.<br>**3) Connectivity:** The enterprise's position and collaborative engagements within industrial chains, its involvement in regional initiatives, platforms, or partnerships.<br>**Output:** Your response must be a valid JSON object containing two main keys: "Application Form" and "Dimension Mapping", and nothing else. |

2.3.1 Perceive module

Enterprise data typically encompasses a wide range of information, including fundamental corporate attributes, operational records, and key performance indicators. Within emerging models of public fund allocation, ensuring that public fundings are directed toward enterprises with genuine financial needs requires moving beyond traditional workflows in which firms manually complete application forms. In this context, the enterprise agent automatically extracts the required information from heterogeneous enterprise data sources and generates application forms in a fully automated manner, thereby improving both efficiency and reliability. In practice, the automated form-filling and dimension-mapping workflow is organized into four major stages. Firstly, the perceive module performs schema-guided semantic specification for each application field. By leveraging field names, natural-language descriptions, formatting constraints, and illustrative examples, the module constructs a structured field definition that precisely identifies the extraction target and its associated restrictions. Secondly, the perceive module applies semantic retrieval and LLM-based contextual understanding to filter relevant information from heterogeneous enterprise data sources, extract field-level content, and subsequently conduct normalization and validity checks. Thirdly, the extracted and verified field values are mapped onto predefined application templates, enabling the automated generation of structured form entries or natural-language narrative sections. Finally, the completed application fields are further projected onto the relevance, quality, and connectivity dimensions to produce structured enterprise-side representations along these dimensions, thereby supporting downstream policy–enterprise matching. The mathematic calculate process is shown:

$$AF^{t}_{mn}, RE^{t}_{mn}, QE^{t}_{mn}, CE^{t}_{mn} = f_{e-p}\left(E_n, TE_m, ME^{t-1}_{mn}\right) \quad (2)$$

where $f_{e-p}$ represents the perceive module of the enterprise agent, $TE_m$ represents the m-th policy application form, and $AF^{t}_{mn}$ represents the completed application form of enterprise $c_n$ for the policy $p_m$ in the $t$-th turn. $RE^{t}_{mn}$, $QE^{t}_{mn}$ and $CE^{t}_{mn}$ denotes the relevance text, quality text, and connectivity text. $ME^{t-1}_{mn}$ denotes the memory from the memory module in enterprise agent in the $t-1$-th turn.

2.3.2 Memory and action Module

The memory module is designed to store both the completed application documents and the enterprise feedback generated by the matching agent, while supporting read–write operations when invoked by the action module. Specifically, $ME^{t-1}_{mn}$ consists of two components, which are the completed application form, denoted as $AF^{t-1}_{mn}$, and the enterprise feedback resulting from the policy–enterprise matching process, denoted as $FE^{t-1}_{mn}$. By continuously updating and retrieving these two types of information, the system preserves contextual continuity across iterations and progressively enhances the quality of the application-filling process.

The action module defines the core operational logic of the enterprise agent. Taking the t-th iteration as an example, the inputs to the enterprise agent include the enterprise data, the policy application form, and the completed application form together with enterprise feedback stored in the memory module during iteration $t-1$. In the current iteration, the action module first retrieves from the memory module the previously stored application form and feedback. It then invokes the perceive module, which updates and refines the prior application content based on the retrieved feedback and the existing application draft, thereby producing an updated version of the application form. After generating the revised application, the action module writes the newly produced application form back into the memory module. Finally, if the matching agent generates new enterprise feedback during the current iteration, the action module also records this feedback in the memory module, ensuring that the enterprise agent maintains continuity and accuracy in its multi-round application-filling process.

## 2.4 Matching Agent

The matching agent aims to determine the alignment between a given policy and an enterprise by leveraging the policy-understanding text generated by the policy agent and the completed application form produced by the enterprise agent. It consists of three components: the matching module, the memory module, and the action module, which together support a complete workflow from matching assessment to feedback generation. Firstly, the matching module employs a matching model that evaluates policy–enterprise alignment at both coarse-grained and fine-grained levels, thereby identifying enterprises eligible for funding. Second, the memory module records the selected enterprises, policy feedback, and enterprise feedback generated during the matching process, ensuring continuity and state tracking across iterations. Finally, the action module defines the operational logic of the matching agent, orchestrating the invocation and coordination of all components. The prompt template used by the matching agent is presented in Table 3.

**Table 3. Prompt Template for the Matching Agent**

| Memory Template |
|---|
| In round $t$, the evaluation results for enterprise $c_n$ under policy $p_m$ are as follows:<br>(1) Decision (Funding): {}<br>(2) Overall Score: {}<br>(3) Dimension Scores: Relevance ({}), Quality ({}), Connectivity ({})<br>(4) Decision Reason: {} |
| Prompt Template |
| **Role:** As the chief reviewer of the government funding review committee, you strictly adhere to the principle of "fixed budget, prioritized support" in making all allocation decisions.<br>**Task:** Integrate the following three materials to determine whether enterprise $c_n$ should receive funding from policy $p_m$ and provide a clear, evidence-based reason for your decision.<br>**Inputs:**<br>**(1) Policy Understanding:** The policy is analyzed along three dimensions:<br>**1) Relevance:** {}<br>**2) Quality:** {}<br>**3) Connectivity:** {}<br>**(2) Enterprise Dimension Mapping:** The enterprise's alignment with the policy dimensions is summarized as:<br>**1) Relevance:** {}<br>**2) Quality:** {}<br>**3) Connectivity:** {}<br>**Useful Tips:**<br>**(1) Scoring Rules:** You must score the enterprise on each dimension (0-1 point) based on the alignment between the Policy Understanding and the Enterprise Dimension Mapping. Then, calculate the Overall Score using the formula:<br>Overall Score = 0.4 * Relevance_Score + 0.3 * Quality_Score + 0.3 * Connectivity_Score<br>**(2) Decision Rule:** Recommend funding if the following condition is met:<br>Overall Score ≥ 0.7<br>**If these conditions are not met, do not recommend funding.**<br>**(3) Reasoning:** Your reason (60-100 words) must cite specific evidence, such as key data points from the application form, alignment with policy clauses, or notable strengths/weaknesses in a particular dimension. Avoid generic statements.<br>**(4) Feedback Mapping:** Based on your Reason, generate two actionable feedbacks:<br>**1) Enterprise Feedback:** Provide concrete, actionable suggestions for the enterprise to improve its qualification or application. Focus on gaps or strengths identified in your reason.<br>**2) Policy Feedback:** Provide interpretive or clarificatory feedback for the policy agent. Focus on ambiguities in the policy text or mismatches between policy intent and common enterprise profiles, as revealed by this case.<br>**Output:**<br>Your response must be a valid JSON object containing five main keys: "Matching results", "Overall Score", "Reason", "Enterprise Feedback" and "Policy Feedback", and nothing else. |

2.4.1 Matching Module

After obtaining the three types of policy-understanding texts generated by the policy agent, together with the completed application form produced by the enterprise agent, we leverage the large language model as the core matching model to determine whether an enterprise should receive public funding and to produce both the funding decision and the corresponding justification. Specifically, the matching model evaluates the alignment between the policy and the enterprise along three dimensions: relevance, quality, and connectivity. Among these dimensions, relevance is assigned the highest weight, as insufficient relevance between the policy objectives and the enterprise profile indicates that the enterprise does not meet the fundamental eligibility criteria of the policy and should therefore be excluded at the outset. In contrast, quality and associativity are assigned relatively lower weights. Nevertheless, relevance alone is insufficient to support funding decisions. In practice, government agencies do not simply allocate funding to all enterprises that meet basic eligibility requirements; instead, they prioritize enterprises with stronger growth potential, higher developmental value, or more substantial alignment with broader policy aims. Consequently, supplementary assessments based on quality and associativity are necessary to capture these additional considerations and support a more comprehensive and realistic funding evaluation.

$$(Y^t_{mn}, R^t_{mn}, RE^t_{mn})=f_{m-m}(R^t_m, Q^t_m, C^t_m, AF^t_{mn}) \quad (3)$$

Where, $f_{m-m}$ denotes the matching module within the matching agent. $Y^t_{mn}$ and $R^t_{mn}$ represent the generated matching result and the corresponding reasons, respectively. The reason $R^t_{mn}$ is subsequently propagated as enterprise feedback $EP^t_m$ and policy feedback $FP^t_m$ to the enterprise agent and the policy agent.

$$FE^t_m=\mathrm{h}_e(R^t_{mn}),\ FP^t_m=\mathrm{h}_p(R^t_{mn}) \quad (4)$$

Through this feedback mechanism, both agents iteratively refine the policy-understanding texts and the automatically completed application documents. $RE^t_{mn}$ denotes the matching score, which comprises the relevance score, the quality score, and the connectivity score. The weight distribution of 4:3:3 across dimensions was determined through Delphi expert consultation.

$$RE^t_{mn}=0.4* RE^{(r)t}_{mn}+0.3* RE^{(q)t}_{mn}+0.3* RE^{(a)t}_{mn} \quad (5)$$

Where $RE^{(r)t}_{mn}$ denotes the relevance score, $RE^{(q)t}_{mn}$ denotes the quality score, and $RE^{(a)t}_{mn}$ represents the connectivity score.

2.4.2 Memory Module and Action Module

The memory module is designed to store the policy–enterprise matching outcomes, including the matching score and the corresponding reasons, while supporting read–write operations when invoked by the action module. The action module defines the core operational logic of the matching agent. In the $t$-th iteration, its inputs consist of the policy-understanding texts produced by the policy agent and the completed application form generated by the enterprise agent. During each iteration, the action module first calls the matching module to perform policy–enterprise matching, thereby obtaining the matching score, the matching result, and the explanatory reasons. These outputs are then written into the memory module. If the matching result aligns with the label, the iteration terminates. Otherwise, the action module transforms the rationale into enterprise feedback and policy feedback, which are subsequently delivered to the enterprise agent and the policy agent, respectively, to facilitate iterative refinement of the policy-understanding texts and the application form in subsequent rounds.

### 2.5 Feedback Loop

The feedback loop serves as the core mechanism for continuous optimization in the proposed method. By leveraging the matching results and reasons generated by the matching agent, it establishes a closed-loop system that facilitates iterative refinement from decision output to agent adjustment. The cycle initiates with the initial outputs from the policy agent and enterprise agent in each iteration. Following multi-dimensional matching analysis by the matching agent, the process yields not only funding decisions and matching scores but also interpretable decision rationales. These rationales are subsequently transformed into concrete policy feedback and enterprise feedback, which target the enhancement of policy interpretation accuracy and the improvement of application form completion precision, respectively. The feedback signals are stored in the respective memory modules of the agents and serve as crucial contextual information in subsequent iterations, guiding the perception modules to perform more targeted parsing and generation of the original inputs. Through this iterative “execution-evaluation-feedback-refinement” process, the method emulates expert-level reflective behavior, progressively converging toward optimal decisions. The termination of the loop is governed by both the maximum iteration count and the stability of matching outcomes, ensuring an effective balance between efficiency and performance. The pseudocode for the proposed method is provided in Algorithm 1.

Algorithm 1

INPUT: $P=\{p_1,p_2,\ldots,p_m,\ldots,p_M\}$, $C=\{c_1,c_2,\ldots,c_n,\ldots,c_N\}$, $TE$, $T$

$t \leftarrow 1$, $MP^0_m \leftarrow \Phi$, $ME^0_{mn} \leftarrow \Phi$, $MM^0_{mn} \leftarrow \Phi$, $Y^0_{mn} \leftarrow None$

for $m$ in 1 to M do

for $n$ in 1 to N do<br>
  $(R^t_m, Q^t_m, C^t_m) = f_{p-p}(T_m, MP^{t-1}_m)$ ▹ Policy understanding<br>
  $AF^t_{mn}, RE^t_{mn}, QE^t_{mn}, CE^t_{mn} = f_{e-p}(E_n, TE_m, ME^{t-1}_{mn})$ ▹ Application form filling<br>
  while $t \leq T$ do<br>
    $(Y^t_{mn}, R^t_{mn}, RE^t_{mn}) = f_{m-m}(R^t_m, Q^t_m, C^t_m, AF^t_{mn})$ ▹ Multi-dimension matching<br>
    if $t=T$ or $Y^t_{mn} = Y^{t-1}_{mn}$ then<br>
      break<br>
    else:<br>
      $FE^t_m = h_e(R^t_{mn})$, $FP^t_m = h_p(R^t_{mn})$ ▹ Generate policy and enterprise feedbacks<br>
      $MP^t_m = MP^{t-1}_m \cup \{R^t_m, Q^t_m, C^t_m, R^t_{mn}\}$, ▹ Update policy agent memory<br>
      $ME^t_{mn} = ME^{t-1}_{mn} \cup \{AF^t_{mn}, RE^t_{mn}, QE^t_{mn}, CE^t_{mn}, FE^t_m\}$ ▹ Update enterprise agent memory<br>
      $MM^t_{mn} = MM^{t-1}_{mn} \cup \{Y^t_{mn}, R^t_{mn}, RE^t_{mn}\}$ ▹ Update matching agent memory<br>
      $t \leftarrow t+1$<br>
    end if<br>
  end while<br>
  return $Y^t_{mn}, R^t_{mn}, RE^t_{mn}$<br>
end for<br>
end for

## 3. Experimental Setup

### 3.1 Datasets

To empirically validate the effectiveness of the proposed method, we constructed a comprehensive dataset. The dataset encompasses multi-dimensional records of enterprises and public fund allocation policies, specifically targeting science and technology innovation funding programs at the provincial level in China. The statistics of these datasets are provided in Table 4, and the detailed information on these datasets is as follows:

Enterprise data: We sourced enterprise data from the Qichacha platform, encompassing structured/unstructured enterprise features. This includes the core metadata (registered capital, industry, location), governance structures (shareholders, key personnel, subsidiaries, controlling entities), operations/risk markers (investments, historical changes, penalties), legal/compliance records (litigation, enforcement actions), innovation assets (patents, trademarks, copyrights, certifications), business ecosystem (suppliers, clients, awards), financial activities (financing, equity pledges). This granular profiling forms dynamic feature vectors for our Enterprise Agent.

Policy data: We aggregated official provincial innovation funding policies, comprising full policy documents, structured application templates, historical allocation records with verification labels identifying enterprise-policy funding matches.

**Table 4. Statistics of the Dataset**

| Description | Value |
|---|---|
| The number of policies | 57 |
| The number of enterprises | 3679 |
| The number of allocation records | 7016 |
| Density | 3.34% |

### 3.2 Evaluation Metrics

To quantitatively evaluate the performance of the proposed method, we adopt the Precision@K [21] and Recall@K [22] metrics. Precision@K measures the accuracy of the top-K matching results, reflecting the method's ability to identify the most relevant policies. Recall@K assesses the proportion of all relevant policies that appear within the top-K results, thereby capturing the method's coverage and comprehensiveness.

### 3.3 Baselines

For a systematic evaluation of the proposed method, we selected the following representative baseline models, covering machine learning-based methods, deep learning-based methods and LLM-based methods:

SVM [23]: A classical machine learning method designed to identify an optimal hyperplane that separates samples of different classes within a high-dimensional feature space. Its objective is to maximize the geometric margin between the hyperplane and the nearest data points from each class.

 

XGBoost [8]: An advanced ensemble learning method based on the principle of gradient boosting. It sequentially constructs an ensemble of decision trees, where each subsequent tree is trained to correct the residual errors of its predecessors. The final prediction is derived from a weighted aggregation of the outputs from all trees.
TextCNN [24]: A deep learning method for text classification, which employs one-dimensional convolutional filters of varying sizes to capture salient local patterns and n-gram features from text sequences.
BERT [25]: A transformer-based pre-trained language model that generates deep, context-aware representations through its bidirectional training objectives, namely masked language modeling and next-sentence prediction.
LLaMA-3.1-70B [26]: A high-performance, open-source large language model developed by Meta. In this evaluation, we leverage its capabilities in a few-shot prompting paradigm, where the model is presented with a carefully crafted instruction and a limited number of exemplars. This approach tests its inherent reasoning and task-solving abilities for the public funding allocation task without updating its internal parameters.
DeepSeek-V3 [27]: A state-of-the-art open-source large language model renowned for its robust performance in complex reasoning tasks. It is evaluated under a comparable few-shot prompting setup to LLaMA-3.1-70B, allowing for a direct comparison of different foundational LLMs' zero-shot generalization capabilities within the same application context.

### 3.4 Implementation Detail

All experiments were conducted on servers running Ubuntu 20.04 LTS, utilizing NVIDIA Tesla V100 GPUs with 32GB VRAM as the computational infrastructure. The proposed method employs DeepSeek-v3 as its core LLMs engine, configured with a temperature parameter of 0 to ensure deterministic outputs. The matching agent implements a multi-dimensional matching mechanism based on cosine similarity, which computes semantic alignment between policy requirements and enterprise characteristics across the relevance, quality, and connectivity dimensions, followed by a weighted fusion process. A similarity threshold of 0.7 is applied to filter high-confidence candidate matches. To validate the framework's generalizability, we adopted a zero-shot prompting strategy without performing full-parameter fine-tuning of the base LLMs. The multi-agent feedback cycle was configured with a maximum iteration count of 3, effectively balancing refinement depth and computational efficiency.

## 4. Experimental Results and Discussion

### 4.1 Experimental Results

To validate the effectiveness of the proposed method, we conducted a comprehensive experiment against the baselines. The experimental results are systematically presented in Table 5.

**Table 5. The Performance of LLM-MAFM and Baselines.**

| Methods | Precision (%) | | Recall (%) | |
|---|---|---|---|---|
| | K=30 | K=50 | K=30 | K=50 |
| SVM | 25.26 | 18.82 | 5.17 | 8.78 |
| XGBoost | 30.98 | 22.25 | 5.93 | 9.47 |
| TextCNN | 35.27 | 29.46 | 8.73 | 13.49 |
| BERT | 37.09 | 31.17 | 9.15 | 14.72 |
| LLaMA-3.1-70B | 41.12 | 34.54 | 10.91 | 16.19 |
| DeepSeek-V3 | 42.39 | 35.93 | 11.76 | 16.83 |
| LLM-MAFM | **45.17** | **37.42** | **14.28** | **18.97** |

Firstly, the proposed LLM-MAFM method demonstrates comprehensive superiority over all baseline methods in the task of intelligent public funding allocation. The experimental results indicate that LLM-MAFM achieves a performance improvement of 6.56% in Precision@30 and 4.15% in Recall@30 compared to the strongest baseline, DeepSeek-V3. This significant advantage is primarily attributed to its innovative multi-agent feedback modeling mechanism. Specifically, the policy agent performs a structured analysis of policy documents along the relevance, quality, and connectivity dimensions, enabling precise capture of policy intent and eligibility criteria. Simultaneously, the enterprise agent leverages LLMs to automatically extract essential information from multi-source enterprise data, generates completed application forms, and maps the content onto the same three dimensions to facilitate accurate policy-enterprise matching. The matching agent conducts multi-dimensional matching analysis to produce allocation decisions with interpretable justifications, while its feedback mechanism enables iterative refinement of both policy interpretation and application generation. Notably, as the candidate set size $K$ increases, LLM-MAFM exhibits the most gradual decline in precision alongside the steepest growth in recall.

Secondly, the performance of LLMs-based methods demonstrably surpasses that of both traditional machine learning and deep learning-based methods. Specifically, taking LLaMA-3.1-70B and BERT for example, the experimental results reveal that LLaMA-3.1-70B achieves a significant performance improvement of 10.86% in Precision@10 and 10.81% in Recall@10. This marked enhancement underscores the inherent effectiveness and superior capability of LLMs in comprehending complex policy directives and enterprise profiles for the task of public funding allocation, leveraging their advanced semantic understanding and reasoning abilities. Furthermore, deep learning-based methods consistently outperform traditional machine learning methods. This performance advantage stems from the ability of deep learning models to automatically learn hierarchical feature representations from raw textual data. Unlike traditional machine learning methods that rely on manual feature engineering, deep learning models capture nonlinear relationships and contextual information through multi-layer neural network architectures, thereby exhibiting stronger representational capacity and generalization performance in policy-enterprise matching tasks.

## 4.2 Discussion

In this section, we have systematically investigated the pivotal roles of two core design components within our proposed method, which are the multi-agent feedback modeling mechanism and the multi-dimensional analysis mechanism. Additionally, we have examined the impact of a critical hyperparameter, the maximum number of iterations, on model performance.

### 4.2.1 Effect of the multi-agent feedback modeling

To rigorously evaluate the contribution of the multi-agent feedback modeling mechanism, we conducted a comparative analysis between the proposed method and two critical variants: a variant without the feedback loop (w/o feedback) and a variant without the enterprise agent (w/o enterprise). The experimental results are shown in Figure 2.

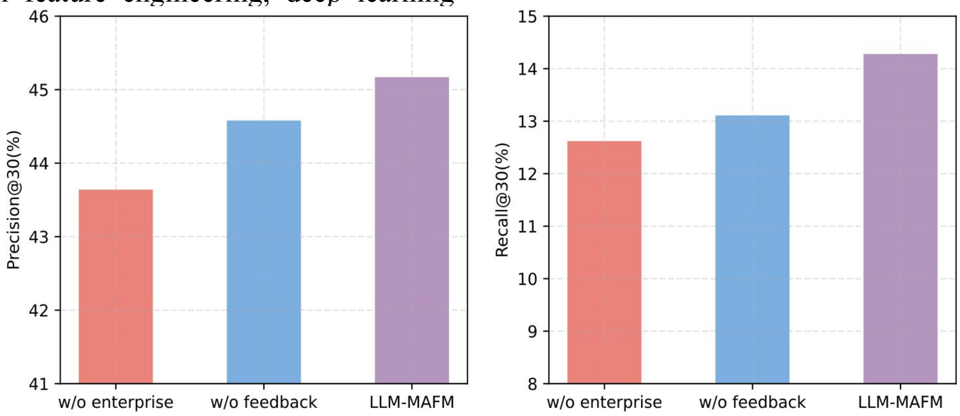


**Figure 2. The Effect of the Multi-Agent Feedback Modeling**

As illustrated in Figure 2, firstly, it is evident that the complete LLM-MAFM method comprehensively surpasses the variant w/o feedback across all evaluation metrics. Specifically, LLM-MAFM achieves a 1.32% increase in Precision@30 and a 8.92% enhancement in Recall@30. This significant performance gain robustly validates the critical role of the multi-agent feedback modeling mechanism in optimizing public funding allocation decisions. The efficacy of this mechanism stems from the dynamic optimization cycle it establishes: the feedback signals generated by the matching agent enable the policy agent and the enterprise agent to iteratively calibrate their outputs in subsequent rounds. This process facilitates a progressively deeper semantic understanding of policy texts and enhances the accuracy of application material generation, ultimately leading to an iterative refinement of policy-enterprise matching precision. Secondly, the results further indicate that w/o feedback variant's performance still consistently exceeds the w/o enterprise variant. This comparative relationship strongly corroborates the necessity of the enterprise agent design. By leveraging LLMs to automatically parse multi-source enterprise data and map structured application fields, this module provides high-quality, specialized enterprise-side information representation for the matching agent. Its inherent capacity for deep modeling of enterprise information confers a significant advantage to the matching task, even in the absence of the feedback loop.

4.2.2 Effect of the multi-dimension analysis

To systematically evaluate the efficacy of the multi-dimensional analysis mechanism, we conducted a comprehensive ablation study. This study compares the performance of the complete LLM-MAFM method against three strategically ablated variants: w/o connect, w/o quality, and w/o con&qua. The w/o connect variant eliminates the connectivity dimension from the matching process. The w/o quality variant removes the quality dimension. The w/o con&qua variant simultaneously excludes both the connectivity and quality dimensions, relying exclusively on the fundamental relevance dimension for policy-enterprise matching. The detailed experimental results are presented in Figure 3.

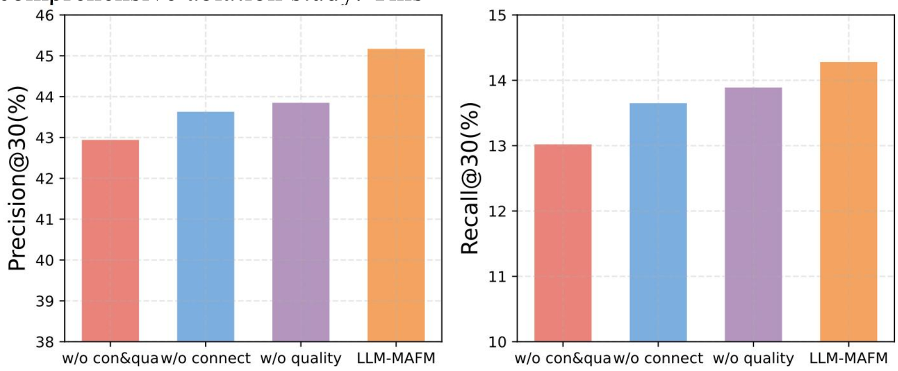


**Figure 3. The Effect of the Multi-Dimension Analysis**

As illustrated in Figure 3, firstly, the complete LLM-MAFM method demonstrates superior performance across all evaluation metrics when compared to the three ablated variants (w/o quality, w/o connect, and w/o con&qua). Specifically, in the comparison with LLM-MAFM and the w/o con&qua variant achieves a 5.19% increase in Precision@30 and a 9.68% rise in Recall@30. This substantial enhancement robustly validates the effectiveness of integrating the relevance, quality, and connectivity dimensions for comprehensive policy-enterprise matching. The underlying rationale stems from the inherent complexity of real-world public funding allocation. In practice, policy implementation is typically constrained by fixed budgets, meaning that not all enterprises aligning with the basic policy direction (relevance dimension) can receive funding. Government agencies must further evaluate applicants based on their operational capabilities and developmental potential (quality dimension), as well as their strategic value within industrial chains and regional coordination (connectivity dimension), to maximize the utility of limited funds. The tripartite analytical framework of LLM-MAFM accurately mirrors this decision-making logic. Secondly, the w/o quality variant and the w/o connect variant still consistently outperform the w/o con&qua variant. This performance strongly corroborates the necessity of incrementally incorporating both quality and connectivity dimensions. It indicates that each additional dimension contributes independent performance gains, and more importantly, that the synergistic evaluation across multiple dimensions is paramount for achieving precise allocation.

4.2.3 Effect of the maximum number of iterations

To systematically investigate the impact of the maximum iteration on model performance, we conducted a comprehensive sensitivity analysis. This experiment was designed to quantitatively evaluate the effects of varying the iteration parameter $T \in \{1,2,3,4\}$ on the performance of the LLM-MAFM method. The experimental results are presented in Figure 4.

As illustrated in Figure 4, firstly, both Precision@30 and Recall@30 metrics exhibit a consistent upward trend as the maximum number of iterations increases. This pattern provides further evidence for the necessity of the iterative feedback mechanism, demonstrating that successive refinement cycles contribute cumulatively to enhancing policy interpretation and policy-enterprise matching accuracy. Secondly, the most substantial gains occur during the initial iterations (particularly from 1 to 3), while the rate of improvement noticeably decelerates in subsequent rounds (e.g., from 3 to 4). This suggests that the early iterations effectively address major discrepancies in understanding and matching, whereas later stages focus on finer-grained adjustments. These findings offer valuable practical implications. Given that additional iterations linearly increase computational costs, a trade-off exists between performance gains and resource consumption. Consequently, for real-world deployment, it may be optimal to set the iteration count to a point such as 3, where the majority of performance benefits are achieved without incurring excessive operational costs. This balance ensures both efficiency and effectiveness in practical applications.

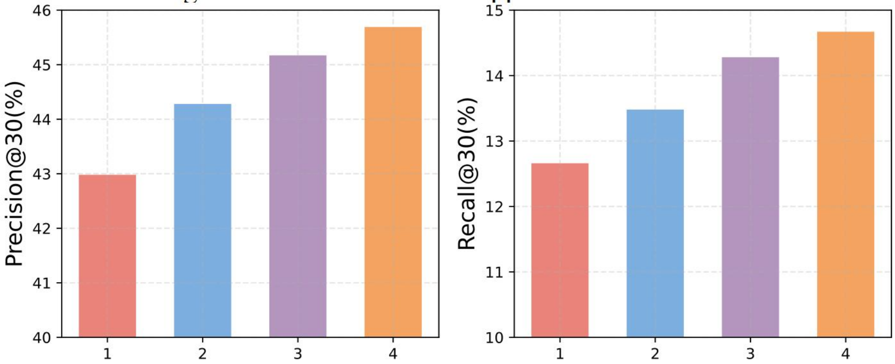


**Figure 4. The Effect of the Maximum Number of Iterations**

## 5. Conclusions

This study proposes LLM-MAFM, a LLMs-based multi-agent feedback modeling method for the emerging government-led public funding allocation model, comprising three core modules: the policy agent, enterprise agent, and matching agent. Specifically, the policy agent parses policy documents and simulates governmental decision-making processes. The enterprise agent automatically generates funding applications by integrating both structured and unstructured data. The matching agent produces allocation results with transparent decision rationales. For experimental validation, we compiled a nationwide dataset of enterprise support policies across Chinese provinces, including policy texts, application materials, fund disbursement records, and corporate information from Qichacha’s commercial database. Experimental results demonstrate the superior performance of our proposed method.

Future research will be pursued along two primary dimensions to advance the practical deployment and societal impact of intelligent public funding allocation systems. Firstly, addressing critical data security concerns in public funding allocation, we will develop distributed training schemes that incorporate privacy-preserving technologies to facilitate cross-domain knowledge fusion while rigorously protecting sensitive enterprise data, thereby enabling secure and trustworthy large-scale data utilization. Secondly, we will explore hybrid human-AI collaborative decision-making models that effectively integrate the analytical capabilities of intelligent agents with the domain expertise of human professionals. This research direction aims to develop robust decision-support systems that synergistically combine algorithmic recommendations with expert judgment, enhancing the reliability and practical applicability of the allocation framework in complex, real-world policy environments.

**Acknowledgments**

This work was supported by the Shenzhen-Hong Kong-Macau Science and Technology Program (Project No: SGDX20201103093201000) and the National Natural Science Foundation of China under grant 72431005.

## References

[1] P. Aroca and P. Fierro, "*Identifying gatekeepers in the allocation of public funds: a spatial econometric analysis*". 2024.

[2] K. Nataliia, N. Mardus, O. Davydov, V. Yevtushenko, O. Melen, and A. Kharchenko, "Devising the policy of funds allocation in

 

the decision-making process under the budget and tax management", *Eastern-European Journal of Enterprise Technologies,* vol. 1, pp. 69-79, 02/28 2022.

[3] W. Walters, "A Regression-based Approach to Library Fund Allocation," *Library Resources & Technical Services,* vol. 51, pp. 263-278, 10/01 2007.

[4] L. Bower, *AI Decision Making for Allocating Government Grant Funds*. 2022.

[5] E. Newman, "Budgeting and Fund Allocation in Higher Education in Ghana," *Journal of Education and Vocational Research,* vol. 4, pp. 275-286, 09/30 2013.

[6] X. Su, "The allocation of public funds in a hierarchical educational system," *Journal of Economic Dynamics and Control,* vol. 28, no. 12, pp. 2485-2510, 2004/12/01/ 2004.

[7] A. A. Talib, "Formula Based Allocation of Public Funds: The Case of Higher Education Research Funding," *Public Money & Management,* vol. 21, no. 1, pp. 57-64, 2001/01/01 2001.

[8] B. Prasetyo and Suharjito, "Optimizing Government Funding Allocation with Ensemble Machine Learning Models," in *2024 10th International Conference on Computing, Engineering and Design (ICCED)*, 2024, pp. 1-5.

[9] C. Reddick, "State resource allocation and budget formats: towards a hybrid model," *Journal of Public Budgeting, Accounting & Financial Management,* vol. 19, pp. 221-244, 03/01 2007.

[10] I. BenDavid-Hadar, "Funding education: developing a method of allocation for improvement," *International Journal of Educational Management,* vol. 32, no. 1, pp. 2-26, 2018/01/01/ 2018.

[11] Y. Wang, N. Liu, Z. Pan, and X. You, "AI-Based Resource Allocation in E2E Network Slicing with Both Public and Non-Public Slices," *Applied Sciences*, vol. 13, no. 22, p. 12505.

[12] H. Jang, "A decision support framework for robust R&D budget allocation using machine learning and optimization," *Decision Support Systems,* vol. 121, pp. 1-12, 2019/06/01/ 2019.

[13] C. Challoumis, "BUILDING A SUSTAINABLE ECONOMY - HOW AI CAN OPTIMIZE RESOURCE ALLOCATION," 10/17 2024.

[14] S. Chen, Y. Liu, W. Han, W. Zhang, and T. Liu, "A Survey on LLM-based Multi-Agent System: Recent Advances and New Frontiers in Application," *arXiv e-prints,* arXiv:2412.17481, 2024.

[15] J. Ma, "Research on the role of LLM in multi-agent systems: A survey," *Appl. Comput. Eng,* vol. 71, pp. 180-186, 2024.

[16] Z. Wang, Y. Yu, W. Zheng, W. Ma, and M. Zhang, "MACRec: a Multi-Agent Collaboration Framework for Recommendation," *arXiv e-prints,* arXiv:2402.15235, 2024.

[17] M. Valentini, "Cooperative and Competitive LLM-Based Multi-Agent Systems for Recommendation," presented at the Advances in Information Retrieval: 47th European Conference on Information Retrieval, ECIR 2025, Lucca, Italy, April 6–10, 2025, Proceedings, Part V, Lucca, Italy, 2025.

[18] K.-T. Tran, D. Dao, M.-D. Nguyen, Q.-V. Pham, B. O'Sullivan, and H. D. J. a. e.-p. Nguyen, "Multi-Agent Collaboration Mechanisms: A Survey of LLMs," p. arXiv:2501.06322Accessed on: January 01, 2025.

[19] S. An, Z. Ma, Z. Lin, N. Zheng, and J.-G. J. a. e.-p. Lou, "Make Your LLM Fully Utilize the Context," p. arXiv:2404.16811Accessed on: April 01, 2024.

[20] J. Gong, J. Wang, G. Wang, J. Ma, and X. Tam, "An LLMs-based Multi-Agent Collaborative Framework for Intelligent Public Fund Allocation," presented at the Proceedings of the 2025 International Conference on Generative Artificial Intelligence for Business, 2025.

[21] S. Cai *et al.*, "Agentic Feedback Loop Modeling Improves Recommendation and User Simulation," p. arXiv:2410.20027Accessed on: October 01, 2024.

[22] C. Gao *et al.*, "CIRS: Bursting Filter Bubbles by Counterfactual Interactive Recommender System," vol. 42, no. 1 %J ACM Trans. Inf. Syst., p. Article 14, 2023.

[23] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt, and B. Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their Applications,* vol. 13, no. 4, pp. 18-28, 1998.

[24] Y. Kim, "Convolutional Neural Networks for Sentence Classification," p.

arXiv:1408.5882Accessed on: August 01, 2014.

[25] J. Devlin, M.-W. Chang, K. Lee, and K. J. a. e.-p. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," p. arXiv:1810.04805Accessed on: October 01, 2018.

[26] A. Grattafiori *et al.*, "The Llama 3 Herd of Models," p. arXiv:2407.21783Accessed on: July 01, 2024.

[27] DeepSeek-AI *et al.*, "DeepSeek-V3 Technical Report," p. arXiv:2412.19437Accessed on: December 01, 2024.

 