ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 2 - 2026

Evaluating Generative AI Programs through Aristotle's Four Causes: Integrating Critical Political Economy and Realist Evaluation

Xihan Gong[1] *and Haonan Cui*[2,*]

[1] School of Marxism, Northeastern University, Shenyang, 110169, Liaoning, China
[2] School of Information Engineering, Shenyang Institute of Science and Technology, Shenyang, 110167, Liaoning, China

SUMMARY: *The wide application of generative AI programs in text generation, question answering, platform services and organizational decision-making has changed the evaluation problem from simple model performance measurement to the collaborative analysis of data, algorithms, platforms and targets. This paper introduces Aristotle's four factors, integrates critical political economy and realist evaluation, and constructs a comprehensive evaluation framework covering training data quality, computing power support, model architecture, generation mechanism, MLOps process, API collaboration, log audit and value alignment. Amazon Hiring, Google Bard and Microsoft Tay are taken as case studies. The results show that the program bias score is 0.18 under the condition of "low data bias-low model fluctuation", and rises to 0.81 under the condition of "high data bias-high model fluctuation". In Amazon Hiring, material and form reasons contributed a total of 61%; in Google Bard, form reasons contributed 38%; in Microsoft Tay, motivation and purpose reasons contributed 58%. The research shows that the failure of generative AI programs has a significant four-factor coupling feature. The framework can provide an evaluation basis with both computer technology depth and social interpretation power for algorithm governance, responsibility tracking and platform regulation, which has methodological significance.*

KEYWORDS: *Generative AI program; Aristotle's four causes; Critical political economy; Evaluation of realism*

1 Introduction

The rapid penetration of generative artificial intelligence in natural language processing, content generation, education support, platform services, and organizational decision-making is promoting the transformation of the technical object "program" from a simple tool system to a composite socio-technical system with data dependence, model autonomy, platform collaboration, and value orientation [1]. The continuous evolution of large language models, diffusion models, and multi-modal generation frameworks makes generative AI programs show strong adaptability and deployment potential in tasks such as text writing, question answering, code generation, content review, intelligent recommendation, and auxiliary governance. However, the widespread application of generative AI programs does not automatically bring stable and reliable practical results [2, 3]. In reality, after importing relevant programs, different organizations often face problems such as output distortion, fact illusion, bias amplification, responsibility ambiguity, scene mismatch and goal drift, resulting

*cuihaonan@syist.edu.cn
https://doi.org/10.65102/is2026789
Published: 30 April, 2026

in obvious tension between technical performance, governance requirements and social effects [4]. Therefore, the research on generative AI programs should not only stay at the level of model capability comparison or tool efficiency discussion, but also need to establish a systematic evaluation framework that can explain "why it works, why it fails, who is responsible for it, and what purpose it serves".

Existing research on generative AI mostly focuses on technical indicators such as accuracy, robustness, bias, toxicity, reasoning ability, and interpretability. These works provide an important foundation for model performance measurement, but their focus usually falls on algorithm output and benchmark results. It is insufficient to reveal the causal chain after the program is embedded in the specific institutional environment, organizational process and platform structure [5-7]. On the one hand, technical evaluation often treats generative AI programs as relatively static model entities, which is difficult to fully explain the linkage relationship between training data, computing power resources, development process, deployment mechanism and usage goals. On the other hand, although traditional project evaluation attaches importance to the correspondence between context, mechanism and results, it does not pay enough attention to computerized elements such as algorithm architecture, data labor, platform capital and technical power, and it is difficult to respond to the characteristics of high complexity, strong coupling and cross-subjectivity of generative AI programs [8, 9]. Therefore, how to establish a bridging mid-level framework between technical analysis and social evaluation has become an urgent problem to be solved in generative AI research.

Aristotle's theory of four Causes provides a theoretical entrance with explanatory power for this problem. The introduction of material factors, formal factors, dynamic factors and purpose factors into generative AI program evaluation can correspond to training data and computing power basis, model architecture and generation mechanism, development subject and platform collaboration, task goal and value orientation respectively, so as to integrate originally scattered data, algorithm, organization and specification problems into the same causal analysis structure [10]. On this basis, critical political economy can reveal the power relations behind corpus sources, data annotation, platform governance and capital control, and realist evaluation can help analyze how mechanisms are triggered and form differentiated results under different situations [11, 12]. The combination of the two methods and the four-factor method can not only enhance the depth of interpretation of generative AI program evaluation, but also help to improve its diagnostic ability for practical problems such as technology failure, responsibility diffusion and value deviation.

Based on this, this paper takes generative AI programs as the research object, attempts to construct a comprehensive evaluation framework that integrates Aristotle's four factors, critical political economy and realist evaluation, and analyzes the operation logic and failure mechanism of generative AI programs from the four dimensions of data, model, subject and goal. Through typical cases, it discusses the process deviation, governance dilemma and value conflict under the interaction of the four factors. This paper aims to provide an evaluation idea with both computer technology perspective and social explanation ability for the interdisciplinary research of generative AI programs, and also provide a scalable theoretical pivot for subsequent research on algorithm governance, platform regulation, and responsibility tracking.

2 Related work

In recent years, the research on generative AI programs has gradually expanded from the discussion of model performance to the analysis of responsibility attribution, institutional

constraints and social consequences. Bartsch et al. reviewed the evolution of AI system accountability research from the perspective of bibliometrics, and pointed out that algorithm application has shifted from single point of technical responsibility to cross-subject collaborative governance, and the accountability framework needs to cover the whole process of design, deployment, use and supervision [13]. Dastani and Yazdanpanah further discussed the responsibility allocation logic of AI systems, and believed that under the condition of increasing degree of autonomous decision-making, the traditional responsibility determination method with a single actor as the core is no longer suitable for the operation reality of complex AI systems [14]. From the perspective of expert responsibility, Hedlund and Persson proposed that the roles of developers, technical teams and evaluators in model training, parameter setting and risk communication should not be diluted by the organizational structure of the platform, which provided important inspiration for the motivation analysis of generative AI programs [15].

In terms of norms and ethics research, Cheong et al. focused on the response of the American legal system to the social impact of generative AI, emphasizing that the existing regulations still lag behind in value protection, rights relief and public risk regulation [16]. Ning et al. proposed an ethical review checklist through a review of medical scenarios, indicating that once a generative AI program is embedded in a high-risk application scenario, accuracy, bias, privacy, and interpretability must be included in the pre-assessment process [17]. Gregor further introduced responsible artificial intelligence into the discussion of academic publishing, indicating that the governance problem of generative AI is not limited to industrial deployment, but has entered the field of knowledge production and science communication [18].

From the perspective of social impact, Formosa et al. pay attention to the reshaping of democratic citizenship by generative AI, pointing out that its role in information access, opinion expression and public participation is not a neutral technical enhancement, but accompanied by value guidance and system shaping [19]. Munoriyarwa and de-Lima-Santos analyzed the problem of initiative, power and authority of generative AI in news research, and revealed that content generation programs are changing the news production process and its authority structure [20].

Critical political economy research provides stronger explanatory power for understanding the deep operation logic of generative AI programs. Meng started from the political rhetoric and national narrative of artificial intelligence, revealing the close relationship between AI development discourse and power construction [21]. Lagna discussed the combination of asset management capitalism and artificial intelligence, and pointed out that there was an obvious trend of capital concentration and platform control behind the expansion of AI [22]. Starting from AI supply chain capitalism, Valdivia emphasizes that algorithm harm cannot be understood only as output layer deviation, but should also be traced back to longer chains such as data collection, labeling labor, energy consumption and environmental costs [23]. Dauvergne's discussion on AI and the environmental cost of the global supply chain, as well as Shibata's analysis on the relationship between digitization and flexibility of the platform service department, also show that there is a complex tension between the goal setting, resource organization and actual consequences of technical procedures [24, 25]. At the level of public discourse and governance, Mao and Shi-Kupfer investigated the content structure and communication characteristics of AI ethics discussions in the Chinese context, and showed that social cognition itself would in turn shape the direction of technology governance [26]. Amoore et al. proposed the concept of "hint word politics" to further reveal that generative AI programs do not only run inside the model, and their input design, interaction logic and governance methods also constitute a new power

space [27].

In general, existing studies have provided an important foundation from the aspects of responsibility, ethics, law, media, and political economy, but most of them still remain in a single dimension, lacking a unified analysis framework for the causal linkage between training data, model structure, development process, platform governance, and goal orientation. Based on this, we introduce Aristotle's four factors to comprehensively evaluate generative AI programs, with a view to establishing a more integrated research path between computer technical analysis and socially critical interpretation.

3 Construction of generative AI program evaluation method based on Aristotle's four factors

3.1 Training data quality computing power support and corpus source evaluation of material factor dimension

In Aristotle's method of four causes, material causes correspond to the underlying materials on which generative AI programs are formed and run. For generative AI programs, this "material" is a composite support system composed of training data, annotation results, pre-training samples, instruction fine-tuning corpus, human feedback data, and computing power resources. The knowledge boundary, semantic stability, and deviation distribution of the program output are often preset at the input end. Therefore, the quality evaluation cannot stay at the statistical level of data volume, but should carry out systematic analysis from three aspects: the quality of training data, the supply intensity of computing power, and the credibility of corpus sources. If the data has class imbalance, semantic duplication, timeliness lag and annotation noise, the model is easy to amplify the original bias in the inference process even if it adopts a more advanced generation architecture, which is manifested as fact illusion, value skew and scene mismatch. If the computing power support is insufficient, the depth of parameter update, the scope of context modeling and the intensity of multi-round alignment training will be limited, resulting in the generation instability and response drift of the program in complex tasks.

From the perspective of training data quality, generative AI programs usually rely on large-scale heterogeneous data to complete knowledge compression and distribution learning, so it is necessary to evaluate sample validity, class balance, label consistency and time series freshness at the same time. To improve the computability of this process, the training data quality index can be expressed as follows:

$$Q_d = \alpha \frac{N_e}{N_t} + \beta \left(1 - \frac{\sigma_c}{\mu_c + \varepsilon}\right) + \gamma A_l + \delta R_t \tag{1}$$

where, Q_d represents the quality index of training data, N_e represents the number of effective samples, N_t represents the total number of samples, σ_c and μ_c represent the standard deviation and mean of category distribution respectively, Al represents the consistency coefficient of labeling, Rt represents the timeliness matching degree of corpus, α, β, γ, δ are weight parameters. This formula integrates data integrity, balance and reliability into a unified evaluation result, which helps to avoid judging the quality of training basis only by sample size.

High-quality data alone is not enough to ensure the stable ability of the program, because the material is also directly constrained by the support conditions of computing power. Generative AI programs rely on GPU clusters, video memory capacity, parallel scheduling

mechanism and continuous computing time in the training and inference stages. Insufficient computing power will compress the training rounds of the model, insufficient parameter updates, and reduce the ability of long-term context learning, thereby weakening the effect of complex semantic mapping. Based on this logic, the computing power support strength can be defined as follows:

$$C_s = \eta \ln(1 + G \cdot M) + \theta P_e + \kappa S_r - \lambda E_c \quad (2)$$

where C_s represents the computing power support index, G represents the number of parallel computing nodes, M represents the available memory capacity of a single node, P_e represents the parallel training efficiency, S_r represents the stability of the training process, E_c represents the energy consumption cost per unit sample, η, θ, κ, λ are the adjustment coefficients.

In addition to data and computing power, the transparency and traceability of the source of the corpus also determine whether the material is reliable. The training corpus of current generative AI programs often comes from open web pages, platform logs, commercially licensed texts, user interaction data, and artificially generated samples. Different sources have obvious differences in copyright status, geographical coverage, language style, value stance, and noise level. If the source is unclear, the authorization is unclear, or the sensitive content is not filtered enough, the program may carry hidden risks in the generation phase. In order to describe the credibility of the source of the corpus, the source reliability can be expressed as follows:

$$S_p=\mu U_a+\nu D_v+\xi T_r+\rho F_s \quad (3)$$

where, S_p represents the reliability of corpus sources, U_a represents the degree of authorization compliance, D_v represents the diversity of sources, T_r represents the traceability of sources, F_s represents the effectiveness of sensitive information filtering, μ, ν, ξ and ρ are the weight coefficients. This formula can transform the problem of corpus source from qualitative description to quantitative evaluation, and provide a basis for subsequent responsibility tracking and risk identification.

In summary, the assessment of the material factor dimension is an overall test of whether the training material is of high quality, whether the computing power is matched, and whether the source is credible. The quality of training data determines the boundary of knowledge input, the support of computing power affects the depth of model learning, and the source of corpus shapes the value background of program generation. Together, the three constitute the underlying technical foundation of generative AI programs.

3.2 Model architecture generation mechanism and interpretability evaluation of formal factor dimension

The formalism focuses on the internal organization and output implementation path of generative AI programs, and the core lies in model architecture, generation mechanism, and interpretability configuration. For large language models and multi-modal generation systems, the number of network layers, hidden dimensions, context Windows, attention distribution and decoding strategy jointly determine the semantic modeling ability and output stability. If the structure design does not match the task requirements, the program is prone to problems such as high fluency but insufficient authenticity, weakened context correlation and broken reasoning chain. Therefore, formal factor evaluation needs to be carried out from three aspects: structural adaptation, generative stability and interpretative validity.

Model architecture evaluation should examine the matching degree between structural

capacity and task complexity. Architecture fit can be expressed as follows:

$$M_a = \alpha \ln(1 + L \cdot H) + \beta \frac{W}{W + \varepsilon} + \gamma C_m - \delta O_r \tag{4}$$

where, M_a represents architecture adaptability, L is the number of network layers, H is the dimension of hidden layer, W is the length of context window, C_m is the multi-modal synergy coefficient, Or is the structural redundancy. This index is used to measure whether the model capacity, context carrying capacity and structure utilization efficiency meet the task requirements.

The generation mechanism evaluation mainly analyzes the probability distribution stability of the decoding stage. Generative AI programs usually output tokens in an autoregressive manner. Temperature coefficient, sampling range, and repetition penalty affect text coherence and risk level. Generation stability can be defined as follows:

$$G_s = 1 - \frac{1}{T} \sum_{t=1}^{T} \frac{H(p_t)}{\log V} \tag{5}$$

where G_s is the generation stability, T is the output sequence length, $H(p_t)$ is the information entropy of the TTH position prediction distribution, and V is the size of the vocabulary. A higher G_s means that the generation path is more concentrated and the output fluctuates less.

In knowledge-intensive tasks, it is also necessary to investigate the semantic fit between the output content and the input context. The semantic consistency coefficient can be constructed as follows:

$$S_c = \frac{1}{T} \sum_{t=1}^{T} \cos(h_t^{in}, h_t^{out}) \tag{6}$$

where, S_c represents the semantic consistency coefficient, h_t^{in} and h_t^{out} are the latent vector representations of the input context and the output result at the t-th position, respectively. This formula can reflect whether the model deviates from the original context during the generation process, and the lower the value is, the greater the risk of semantic drift.

Interpretability evaluation needs to determine whether the internal attention path of the model is clear, and whether the explanation results can correspond to the real decision basis. Based on the attention weight distribution, the explanatory indicators can be expressed as follows:

$$I_a = \frac{1}{L} \sum_{l=1}^{L} \left(1 - \frac{H(A^{(l)})}{\log N}\right) \tag{7}$$

where, I_a is the attention interpretability index, $A^{(l)}$ is the attention distribution of the LTH layer, and N is the number of input tokens. The higher this index is, the more focused the model is on the key semantic position and the clearer the explanation path is.

In general, the evaluation of the form factor dimension needs to integrate the model structure, generation process and interpretation ability into a unified framework. Architecture fit reflects the quality of model design, generates stability and semantic consistency coefficients to identify output risks, and attention interpretability index provides technical basis for fault localization and governance intervention. Through these indicators, we can

systematically evaluate whether the internal mechanism of generative AI programs is stable, transparent, and suitable for specific task scenarios.

3.3 Development process platform collaboration and responsibility tracking and evaluation of dynamic factor dimension

Motivation factors focus on the question of "who drives a generative AI program, through what process, and how to form a traceable chain of responsibility". In generative AI programs, motivation is mainly reflected in the development team, data engineering link, model training process, platform deployment mechanism, and post-run feedback loop. For large model applications, program performance is not only determined by training data and model structure, but also continuously affected by engineering steps such as data cleaning, parameter fine-tuning, alignment training, interface orchestration, access control, and log auditing. If the development process lacks standardized constraints and the platform side lacks collaborative scheduling and security control, the system is prone to problems such as version drift, interface mismatch, audit lag and responsibility attribution difficulties after going online. Therefore, the dynamic factor evaluation needs to put the MLOps process, platform collaboration capability and responsibility tracking mechanism in the same framework. The generative AI program development process and the platform collaborative evaluation framework are shown in Figure 1.

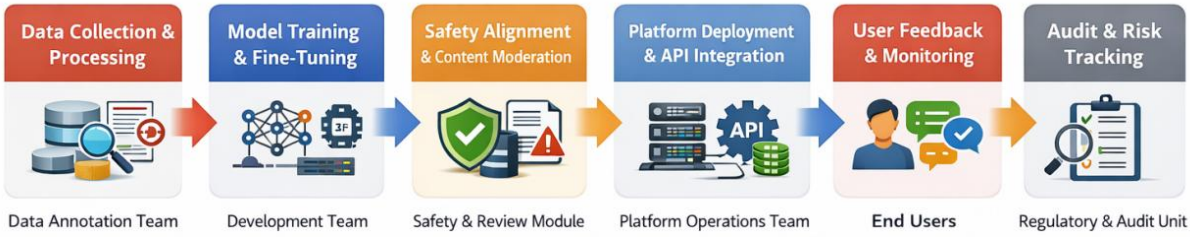

Figure 1: Framework diagram of generative AI program development process and platform collaborative evaluation

As shown in Figure 1, the generative AI program development process and platform collaborative evaluation framework can be divided into six links: data preparation, model training and fine-tuning, security alignment and content review, platform deployment and interface call, user feedback and backflow, responsibility recording and risk tracking. Each link corresponds to data engineering team, algorithm development team, platform operation and maintenance team, audit module and audit node, and forms linkage through model registration, version control, API gateway, log link and alarm system. The framework can correspond the technical path of the program from training to service output with the responsibility chain, and provide a clear reference for subsequent evaluation.

The maturity of the development process determines whether the program is stable in iterations and can be defined as follows:

$$D_m = \alpha \frac{N_a}{N_p} + \beta T_c + \gamma V_s + \delta Q_g \tag{8}$$

where, D_m is the maturity of development process, $\frac{N_a}{N_p}$ represents the proportion of process nodes that have been automated, T_c represents the coverage of continuous integration and continuous deployment, V_s represents the integrity of data and model version management, Q_g

represents the effectiveness of quality access control, α, β, γ, δ are weight coefficients. The higher the index, the more standardized the engineering process of the program from training, testing to production, and the lower the risk of model drift and deployment error.

Platform collaboration capability reflects the linkage efficiency among model service, audit module, storage node and call interface, which can be expressed as follows:

$$P_c = \eta S_r + \theta \frac{1}{1 + L_{avg}} + \kappa F_b + \lambda U_o \tag{9}$$

where, P_c is the platform collaboration efficiency, S_r represents the success rate of service request, L_{avg} represents the average response delay, F_b represents the feedback reflux synchronization rate, U_o represents the resource orchestration utilization, η, θ, κ, λ are the regulation parameters. This formula can comprehensively measure the running quality of interface call, module collaboration and feedback synchronization, and is suitable for evaluating the stability of generative AI platforms in multi-service scenarios.

The key of responsibility tracing evaluation is to determine whether program output can be traced back to specific development and operation steps. To this end, the completeness of the chain of responsibility can be defined as follows:

$$R_t = \mu L_g + \nu M_r + \xi A_u + \rho C_h \tag{10}$$

where, R_t is the integrity of responsibility chain, L_g represents the integrity of log retention, M_r Represents the completeness of model registration information, Au represents the verifiability of access identity, C_h represents the validity of call chain hash verification, μ, ν, ξ and ρ are weight parameters. This metric can be used to consolidate the version number, training batch, deployment time, call subject, and output records into the audit scope, thereby enhancing the ability to locate responsibility.

For risk governance after going online, it is also necessary to investigate the response efficiency of the system from anomaly detection to closed-loop disposal, which can be expressed as follows:

$$G_r = \frac{\omega E_d + \phi W_a + \psi H_p}{1 + \tau_r} \tag{11}$$

where, G_r is the risk response index, Ed is the accuracy of anomaly detection, W_a is the effectiveness of alarm triggering, H_p is the arrival rate of manual intervention, τ_r is the average disposal delay, ω, ϕ and ψ are the weight coefficients. The formula can describe the governance efficiency of the platform in the face of harmful output, ultra vious call and model exception, and it is easy to identify the specific performance of power cause failure in the operation phase.

On the whole, the evaluation of the motivation dimension should focus on "whether the process is standardized, whether the platform is coordinated, whether the responsibility is traceable, and whether the risk is controllable". Development process maturity is used to identify the stability of engineering links, platform collaboration efficiency is used to measure the quality of service operation, and the integrity of responsibility chain and risk response index support audit tracking and governance intervention. By embedding these computer technical indicators into the dynamic cause analysis, the development subject, platform mechanism and responsibility chain of generative AI programs can form a clearer evaluation structure, and also provide a methodological basis for the problem of responsibility diffusion and governance failure in the case analysis of Chapter IV.

3.4 Assessment of task-target scene adaptation and value alignment for objective dimensions

Objective This study focuses on the question of "why to run, what task to serve, and whether the output conforms to the preset value boundaries" of generative AI programs. In generative AI programs, purpose is usually represented by task goal setting, scenario deployment constraints, policy optimization directions, and value alignment rules. For large language models, retrieval enhanced generation systems, and multi-modal agents, if the task goal is vaguely defined, the model may have problems such as response offset, scene mismatch, and governance overstep even if it has strong generation ability. Therefore, it is necessary to integrate task semantics, scenario constraints and value rules into the technical analysis framework.

Task goal consistency can be used to measure whether the system output conforms to the predetermined functional boundaries, which can be defined as:

$$G_o = \frac{z_t \cdot z_y}{\|z_t\|\|z_y\|} \times (1 - \omega E_d) \tag{12}$$

where, G_o is the task target consistency coefficient, z_t represents the target task semantic vector, z_y represents the model output semantic vector, Ed represents the target offset rate, and ω is the penalty coefficient. This formula combines the semantic fit degree with the deviation risk, and can reflect whether the program is stably generated around the established goal, which provides a basis for subsequent scene evaluation.

Scenario adaptation capability mainly examines the deployment effect of the program in the specific business environment, which can be expressed as follows:

$$A_s = \alpha D_a + \beta R_h + \gamma \frac{1}{1 + L_r} + \delta(1 - F_e) \tag{13}$$

where, A_s is the scene adaptation index, D_a represents the domain task accuracy, R_h represents the retrieval hit rate, L_r represents the average response delay, F_e represents the abnormal output rate, and α, β, γ, δ are the weight parameters. The index can simultaneously describe domain knowledge invocation, real-time response and exception control capabilities, which is helpful to judge the applicability of programs in educational Q&A, medical assistance, government services and other scenarios. Through this calculation, "whether it is suitable to go online" can be transformed into a quantifiable judgment.

Value alignment assessment focuses on whether the output satisfies safety, compliance and ethical constraints, which can be defined as:

$$V_a = \mu P_s + \nu H_p + \xi C_r - \rho T_x \tag{14}$$

where, V_a is the value alignment index, P_s represents the policy rule passing rate, H_p represents the artificial preference consistency, C_r represents the compliance response rate, T_x represents the harmful content triggering rate, μ, ν, ξ and ρ are the weight coefficients. The formula integrates the policy model, human feedback and content security detection into the same evaluation structure, which can better reveal whether the program keeps the value boundary stable in the generation stage. It can be seen that the purpose factor assessment is not only related to the output effect, but also related to the system governance intensity.

In order to facilitate the induction of the evaluation objects and technical indicators of the purpose factor dimension, Table 1 gives the specific evaluation content of task objectives,

scene adaptation and value alignment.

Table 1: Evaluation Indicator Design for Task Objectives, Scenario Adaptation, and Value Alignment under the Final Cause Dimension

Evaluation Subdimension	Technical Object	Core Indicators	Reference Threshold
Task Objective Consistency	Task planning module, prompt template, output controller	Semantic similarity, objective deviation rate, task completion rate	Semantic similarity ≥ 0.85, deviation rate ≤ 0.10
Scenario Adaptation Capability	Retrieval-augmented module, domain adapter, inference service interface	Domain accuracy, retrieval hit rate, response latency	Accuracy ≥ 0.88, hit rate ≥ 0.80, latency ≤ 2.0 s
Value Alignment Level	Policy model, RLHF module, content safety filter	Rule pass rate, preference consistency, harmful content trigger rate	Pass rate ≥ 0.95, trigger rate ≤ 0.03
Overall Deployment Availability	Scheduling gateway, monitoring module, audit feedback chain	Online stability rate, exception recovery rate, manual review fulfillment rate	Stability rate ≥ 0.97, recovery rate ≥ 0.90

Table 1 shows that the purpose factor dimension cannot only discuss abstract goals, but also needs to be implemented with the help of computer technology modules such as task control, scenario deployment, and security alignment. Through the joint analysis of goal consistency coefficient, scene adaptation index and value alignment index, the goal realization degree and governance controllability level of generative AI programs in a specific application environment can be systematically judged, which also provides a method basis for effect differentiation and value conflict identification in subsequent case discussions.

3.5 Generative AI program failure mode recognition and diagnosis method under four-factor coupling

In the running process of generative AI programs, the failure is usually not directly triggered by a single factor, but is the result of the combined effect of training data bias, model structure defects, development process oversight, and goal constraint deviation. The material cause determines the knowledge input boundary, the form cause affects the semantic generation path, the dynamic cause relates to the deployment collaboration and the responsibility closed loop, and the purpose cause constrict the output direction and the value boundary. When a local anomaly occurs in any link of the four dimensions, the risk will be transmitted to the output along the data flow, parameter flow and call chain, and eventually show up as fact illusion, bias amplification, response instability, ultra vivious generation and governance failure. Therefore, failure mode recognition needs to establish a diagnostic method for multi-dimensional feature fusion, which integrates data anomaly detection, model behavior analysis, platform log audit and target deviation monitoring into the same computing framework.

In order to measure the overall failure risk under the effect of four-factor coupling, the coupling failure index can be defined as follows:

$$F_c = 1 - \prod_{i=1}^{4}(1 - r_i)^{w_i} \tag{15}$$

where, F_c represents the coupling failure index, r_i represents the local risk intensity corresponding to the four factor dimensions, and w_i represents the weight of each dimension. The formula can map the training data noise, generation offset, process anomaly and target mismatch into a comprehensive risk value. When the risk of a certain dimension increases rapidly, the overall failure index will be amplified synchronously, which is helpful to identify the evolution trend of the program from local instability to systemic failure.

In engineering implementation, the four-cause failure identification can construct an abnormal propagation intensity index based on multi-source logs and model behavior characteristics, which can be used to describe the degree of risk diffusion between modules:

$$P_f = \frac{1}{N}\sum_{j=1}^{N}\left(\lambda_1 a_j + \lambda_2 b_j + \lambda_3 c_j + \lambda_4 d_j\right) \tag{16}$$

where, P_f represents the anomaly propagation intensity, a_j is the data quality outlier, b_j is the generation stability outlier, c_j is the platform call chain outlier, d_j is the target offset outlier, N is the number of monitoring samples, and λ_1 to λ_4 are the normalized weights. The index can be linked with log audit, online monitoring and security alarm modules to locate whether the risk originates from the input end, generation end, deployment end or target control end, so as to improve the pertinence of diagnosis.

After the completion of anomaly aggregation, it is also necessary to discriminate the failure type. The diagnostic confidence can be expressed as follows:

$$D_c = \frac{\exp(s_k)}{\sum_{m=1}^{M}\exp(s_m)} \tag{17}$$

where, D_c represents the diagnostic confidence of a certain failure category, s_k represents the comprehensive discrimination score of the KTH failure mode, and M represents the total number of failure types. With the help of this formula, the typical failure modes such as data bias, illusion generation, liability diffusion and target drift can be transformed into comparable classification results, and the dominant risk category can be output. In this way, the diagnosis results can directly serve the subsequent governance decisions and accountability tracking.Figure 2 shows the failure mode identification process of generative AI programs under four-factor coupling.

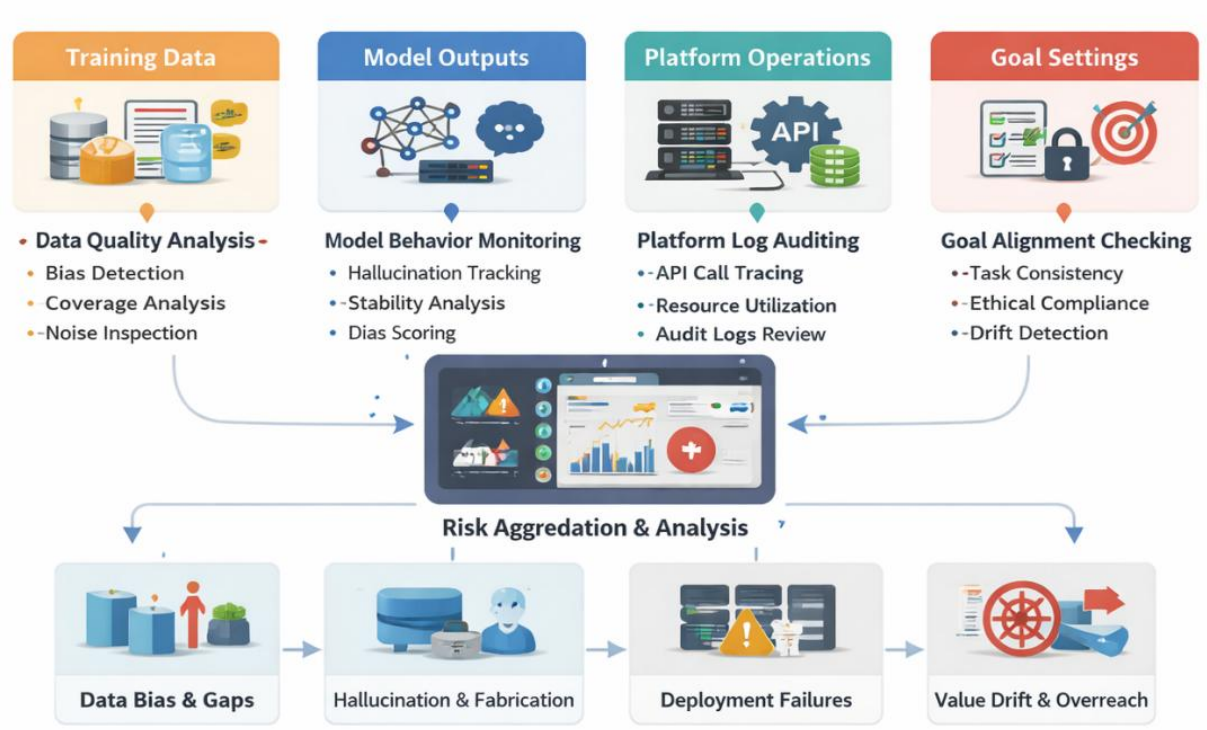

Figure 2: Flowchart of generative AI program failure mode recognition under four-factor coupling

Through the above method, the failure identification no longer stays at the single point of fault location level, but can start from the four-factor coupling relationship, and make a structural diagnosis of the anomaly source, propagation path, and final performance of the generative AI program. This method not only strengthens the operability of technical monitoring, but also provides a unified discrimination basis for the analysis of procedural deviation, responsibility imbalance and value conflict in the case evaluation of Chapter IV.

4 Case evaluation and result discussion

4.1 Data source of case selection and process design of four-factor assessment

In order to ensure that the case evaluation is representative and comparable, this paper selects Amazon Hiring, Google Bard and Microsoft Tay as the analysis objects, corresponding to three typical generative AI program scenarios of intelligent recruitment screening, public information generation and open social interaction, respectively. The three types of cases have obvious differences in training data organization, model generation mechanism, platform collaboration level and target constraints, which can more completely cover the main observation content of four dimensions of material, formal, dynamic and objective factors. Amazon Hiring mainly reflects the structure deviation of training samples and the imbalance of screening logic. Google Bard better reflects the fact checking risk and deployment response pressure in knowledge generation. Microsoft Tay focuses on exposing the platform review gap, abnormal feedback diffusion and governance failure in the open interactive environment. In this paper, a total of 36 case related materials were collected, including 9 enterprise public instructions and product information, 14 mainstream media investigations and reports, and 13

academic research and review documents. The data span from 2016 to 2025. Combined with the four-factor assessment method constructed in Chapter 3, this paper sets up a total of 12 core indicators under four dimensions, and carries out subsequent analysis according to the process of "case data collation - index mapping - risk feature extraction - comprehensive assessment and comparison". In order to ensure the comparability between different cases and different dimension indicators, this paper conducts 0-1 standardization on each indicator. The corresponding relationship between sample sources of each case and four factors evaluation indicators is shown in Table 2.

Table 2: Corresponding table of case sample sources and four factors evaluation indicators

Case Sample	Scenario Type	Number of Sources	Main Data Sources	Material Cause Evaluation Indicators	Formal Cause Evaluation Indicators	Efficient Cause Evaluation Indicators	Final Cause Evaluation Indicators
Amazon Hiring	Intelligent recruitment screening	12	Corporate public information, media investigative reports, academic discussion materials	Gender bias rate in data, sample coverage, corpus source transparency	Stability of model screening rules, output consistency	Standardization of development process, completeness of responsibility tracing	Consistency of recruitment objectives, level of fairness constraints
Google Bard	Public information generation	11	Product demonstration records, media reports, technical review materials	Timeliness of knowledge corpus, reliability of factual samples	Generation stability, factual verification capability, semantic consistency	Platform review response speed, deployment coordination efficiency	Fitness of information service objectives, degree of value alignment
Microsoft Tay	Open social interaction	13	Platform interaction records, corporate responses, research literature	Corpus contamination rate in interaction data, input noise intensity	Degree of unstable content generation, frequency of abnormal outputs	Risk response index, adequacy of manual intervention	Goal retention in dialogue, level of safety boundary control

Table 2 shows that the selected cases have strong differences in data basis, model mechanism, platform governance and task objectives, which can provide a unified assessment entry for the subsequent deviation analysis of material and form causes, responsibility diffusion analysis of dynamic causes and value conflict analysis of purpose causes, and also lay a structured foundation for the expansion of subsequent chart results in Chapter 4.

4.2 Analysis on the generation mechanism of program deviation under the effect of material and form factors

The procedural bias in this case is not triggered by a single technical link, but the result of the combined effect of the quality of the training data, the source structure of the corpus, and the model generation mechanism. Taking Amazon Hiring and Google Bard as the main analysis objects, and combining with Microsoft Tay for comparison, it can be found that the data

sample of the former is affected by the historical recruitment structure for a long time. Google Bard is jointly influenced by the timeliness of corpus and the fluctuation of decoding mechanism in the process of knowledge generation, while Microsoft Tay is influenced by input noise pollution and generation instability in an open interactive environment. All of them show obvious coupling characteristics of material and form factors. The program deviation scores under different combinations of material and form factors are shown in Figure 3.

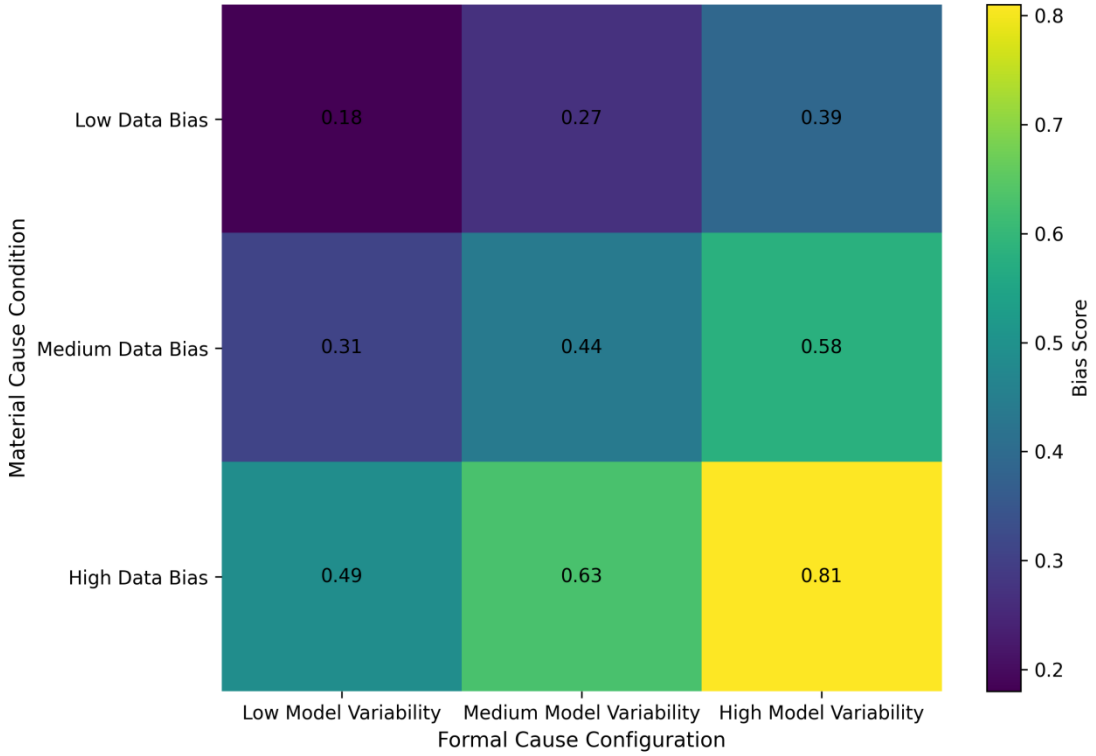

Figure 3: Heat maps of program deviation scores under different combinations of material and form factors

Figure 3 shows that the program bias score keeps increasing as the degree of data bias and the degree of model fluctuation synchronously increase. In the "low data bias-low model volatility" combination, the bias score is only 0.18. When transferred to the "medium data bias-medium model fluctuation" combination, the bias score increased to 0.44. In the "high data bias-high model fluctuation" condition, the bias score reaches 0.81. This shows that the defects of material causes will be further explicit under the amplification of formal causes, especially when the degree of freedom of model generation is high and the fact constraint is weak, the deviation output is easier to accumulate and spread.

In order to further compare the influence degree of key indicators of material and form factors on deviation output, Table 3 shows the comparison results of various indicators.

Table 3: Comparison table of the influence of key indicators of material and form factors on deviation output

Indicator Category	Key Indicator	Amazon Hiring	Google Bard	Microsoft Tay	Impact on Biased Output
Material Cause	Data bias rate	0.72	0.34	0.61	The higher the data bias rate, the more pronounced the structural bias in the output
Material Cause	Corpus source reliability	0.58	0.76	0.49	The less stable the source, the more likely erroneous knowledge and abnormal expressions are to enter the output
Material Cause	Sample coverage	0.63	0.81	0.57	Insufficient coverage weakens the program's ability to adapt to marginal scenarios
Formal Cause	Architectural fitness	0.69	0.74	0.62	When the architecture is poorly matched to the task, output consistency declines
Formal Cause	Generation stability	0.71	0.59	0.46	Lower stability increases the probability of hallucinations, distortion, and abnormal responses
Formal Cause	Semantic consistency	0.65	0.57	0.43	The lower the consistency, the more likely off-topic outputs, mismatches, and improper expansions become

Table 3 shows that the data deviation rate of Amazon Hiring is the highest, reaching 0.72, indicating that the historical sample structure has a strong traction effect on the screening results. The source reliability of Google Bard is relatively high, with 0.76, but the generation stability is only 0.59, reflecting that its problems are more concentrated in the generation mechanism level. The generated stability and semantic consistency of Microsoft Tay are only 0.46 and 0.43, respectively, indicating that the form is more likely to trigger abnormal output due to instability in an open interactive environment. In general, the material factor determines the initial strength of the bias input, and the form factor determines the amplification degree of the bias at the generation end. The superposition of the two factors will significantly increase the risk of program bias.

4.3 Analysis of responsibility diffusion and governance failure driven by dynamic factors

The governance failure of generative AI programs is often directly related to the dispersion of development subjects, insufficient platform collaboration, and broken responsibility chain. Although Amazon Hiring, Google Bard and Microsoft Tay have different scenarios, they all show a common problem caused by imbalance of power, that is, when the program output is abnormal, it is difficult to quickly locate the responsibility between the development team, the platform operator, the review mechanism and the user interaction end. The multi-agent responsibility tracing capability is shown in Figure 4.

Figure 4: Radar chart of multi-agent responsibility tracking capability

As can be seen from Figure 4, Google Bard is relatively high in process normalization and platform collaboration, reaching 0.79 and 0.76 respectively, indicating that its engineering link and platform support are relatively complete. The log completeness and responsibility traceability of Amazon Hiring are 0.71 and 0.66, respectively, which are at a medium level. Microsoft Tay has a number of indicators that are significantly lower, among which the responsibility traceability is only 0.39 and the response speed is only 0.35. This shows that in an open interactive environment, if the platform lacks a stable review and rapid intervention mechanism, the motivation is more likely to evolve to governance failure due to risk, and eventually magnifies the problem of responsibility diffusion.

4.4 Analysis of program effect alienation and value conflict under objective deviation

Objective Offset is mainly manifested as the actual output of the program gradually deviates from the established task boundary, and further leads to effect alienation and value conflict at the level of scene adaptation and value alignment. Amazon Hiring maintains a high goal focus under the efficiency orientation, but due to the insufficient embedding of fairness constraints, there is a deviation between the consistency of task goals and value realization. Google Bard has a strong ability of knowledge generation in the public information service scenario, but there is still a tension between fact checking and real-time response, which is manifested as that the scene adaptation ability and value robustness are not fully synchronized. However, Microsoft Tay quickly deviated from the original dialogue goal in the open interactive environment, which not only showed weak scene control ability, but also showed continuous instability in the security boundary and value constraint level. It can be seen that the purpose cause problem is not only reflected by the decline of value alignment score, but also by the weakening of goal retention ability and the decline of scene adaptation level. The distribution of alignment scores for different case values is shown in Figure 5.

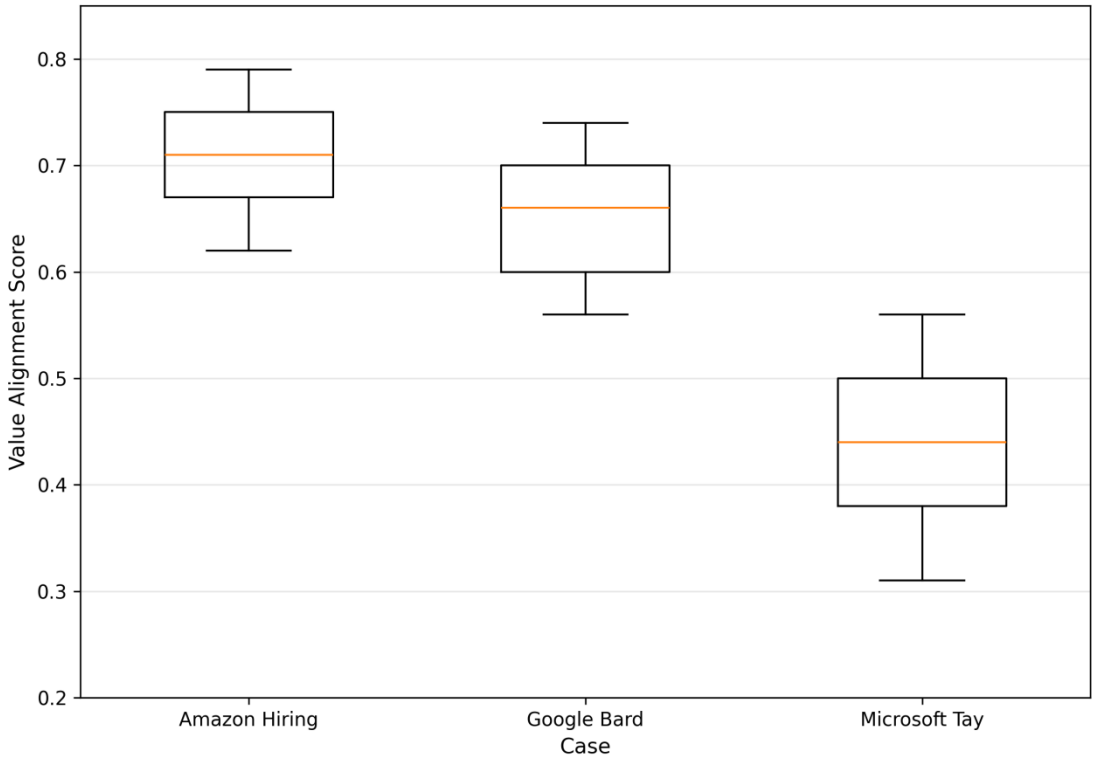

Figure 5: Boxplots of the distribution of alignment scores for different case values

As can be seen from Figure 5, the value alignment score of Amazon Hiring is generally high, with a median of about 0.71 and a interquartile range of 0.65 to 0.77. Google Bard has a median of about 0.66, a slightly lower distribution than Amazon Hiring; The median value of Microsoft Tay is only 0.44, and the overall range is concentrated between 0.35 and 0.53. This shows that once the purpose deviation occurs, the program output will shift from task completion to effect alienation, and further induce value conflict. Among them, the fluctuation range and low value range of Microsoft Tay are the most obvious, indicating that insufficient target constraints are more likely to amplify the risk of value imbalance in open scenarios.

4.5 Comprehensive discussion on the evaluation results of generative AI programs under four-factor interactions

Based on the above analysis, it can be found that the risk formation of generative AI programs does not follow a single technical path, but is the result of material factors, formal factors, dynamic factors and purpose factors superimposed and cross-amplified in different scenarios. Amazon's Hiring problem is more focused on the coupling between the historical sample structure and the screening logic, while Google Bard is mainly reflected in the misalignment between the knowledge generation mechanism and the information service goal. Microsoft Tay shows the synchronization instability of platform governance, responsibility tracking and value boundary in an open interactive environment. Figure 6 shows the proportion of failure contribution of the four-cause dimension.

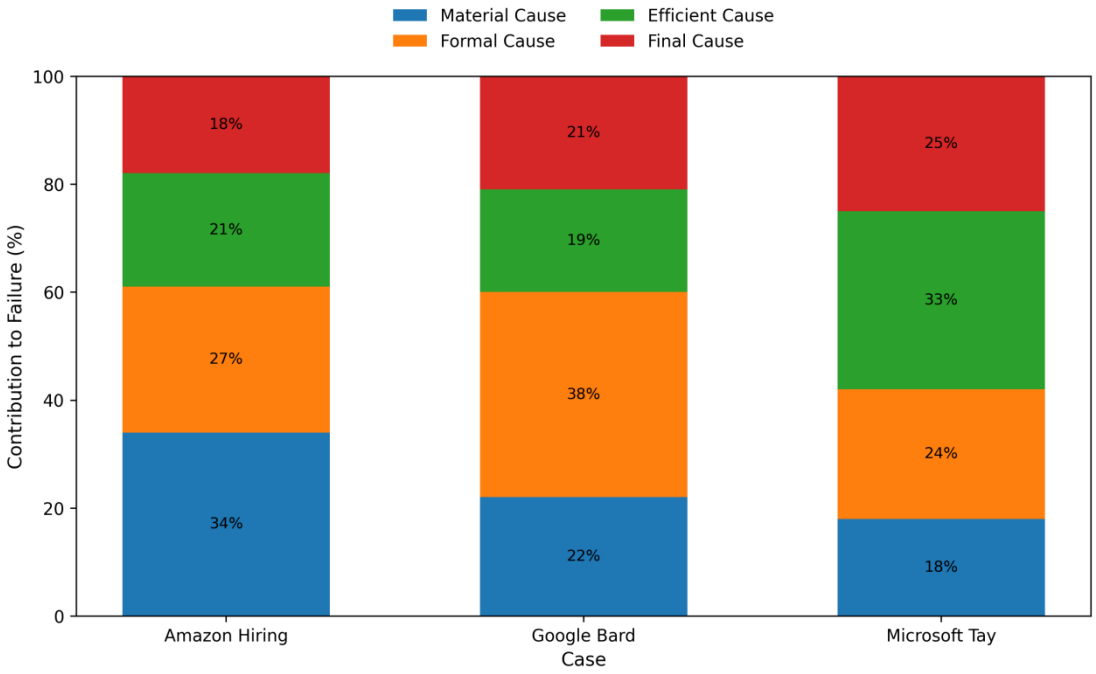

Figure 6: Stacked bar plot of proportion of failure contributions for four dimensions

As can be seen from Figure 6, the failure contribution of Amazon Hiring mainly focuses on material and formal reasons, accounting for 34% and 27% respectively, and the total reaches 61%, indicating that the coupling of data structure deviation and screening mechanism is the main source of risk. Google Bard has the highest contribution of form factor (38%), which is significantly higher than that of material factor (22%) and dynamic factor (19%), indicating that its core problems appear more in the generation mechanism, fact checking and output control links. The contribution of motivation and purpose factors of Microsoft Tay is 33% and 25%, respectively, and the total reaches 58%, indicating that insufficient collaborative governance and target boundary loss in the open platform are the key causes of its abnormal diffusion. The results show that although different cases show program failure, the dominant factors are not the same, and the hierarchical diagnosis must be carried out by combining the four-factor structure. The comprehensive assessment scores of four factors in different cases are shown in Figure 7.

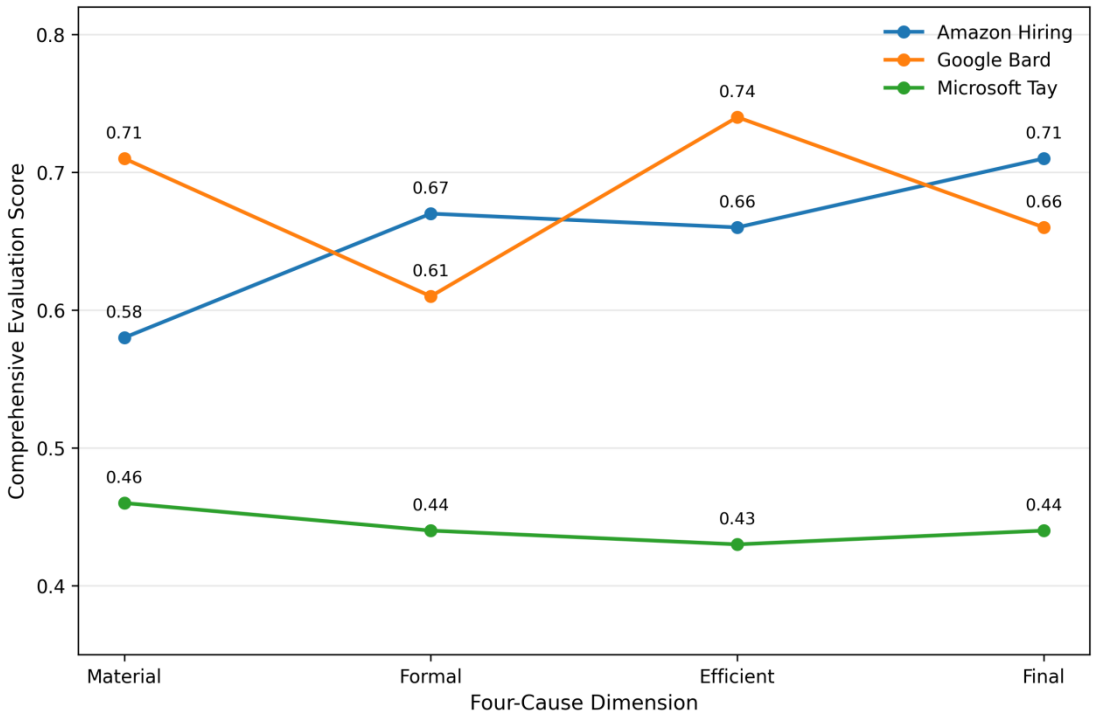

Figure 7: Line chart comparing the comprehensive assessment scores of four factors in different cases

As can be seen from Figure 7, Amazon Hiring has a relatively high score of 0.71 in the objective factor dimension, but only 0.58 in the material factor dimension, indicating that its program goal setting is clear, and the defects in the input sample structure significantly weaken the overall evaluation performance. Google Bard has the highest score of 0.74 in the dynamic factor dimension, indicating that the platform collaboration and engineering process are relatively complete, but the score of form factor is only 0.61, indicating that the stability of the generation mechanism is still a weak link. The scores of Microsoft Tay in the four dimensions are all low, among which the scores of motive cause are the lowest, only 0.43, formal cause and purpose cause are 0.44 and 0.44, respectively, showing that its failure has obvious characteristics of multi-cause superposition. To sum up, the evaluation centers of Amazon Hiring, Google Bard and Microsoft Tay are biased towards "input bias dominant", "mechanism imbalance dominant" and "governance failure dominant" respectively, which indicates that the evaluation of generative AI programs under the interaction of four factors cannot stay at the level of single indicator judgment. Instead, the data basis, generation mechanism, platform process and target constraints should be combined for overall identification.

5 Conclusion

Focusing on the training data, computing power foundation, model architecture, generation mechanism, platform collaboration, responsibility chain and value alignment of generative AI programs, this paper constructs a comprehensive evaluation method based on the four-factor method, and verifies the explanatory ability of the framework through three typical cases. The case results show that Amazon Hiring achieves 0.72 in the data bias rate, Google Bard only achieves 0.59 in the generation stability, and Microsoft Tay only achieves 0.39 and 0.44 in the accountability traceability and value alignment median respectively. It indicates that program

risk will continuously accumulate in the interaction of material, formal, dynamic and purpose factors. Furthermore, the heat map, radar chart, box plot and comprehensive comparison chart jointly show that the core problems of generative AI programs focus on four main lines: input bias, mechanism imbalance, governance failure and goal drift. The contribution of this paper is to incorporate training data evaluation, decoding stability analysis, platform log audit, and value alignment detection into a unified framework, which provides a scalable analysis path for engineering optimization, risk diagnosis, and institutional governance of subsequent generative AI programs.

About the Author

Xihan Gong was born in Shenyang, Liaoning Province, P.R.China, in 1995. She received the master's degree from Northeastern University, P.R.China. Currently, she is pursuing her doctoral studies at the School of Marxism, Northeastern University. Her research focuses on philosophy of science, philosophy of technology (with an emphasis on the philosophical implications of generative artificial intelligence and data analysis).

Haonan Cui received the B.S. degree in Wind Energy and Power Engineering from Shenyang University of Technology, Shenyang, China, in 2015. He obtained the M.S. degree in Control Engineering from Northeastern University, Shenyang, China, in 2018. Currently, he is a lecturer at Shenyang Institute of Science and Technology, and he is also pursuing his Ph.D. degree in Electrical Engineering at Shenyang University of Technology. His research interests include deep learning, artificial intelligence algorithms.

References

[1] Chang, Y., Wang, X., Wang, J., Wu, Y., Zhu, K., Chen, H., Yi, X., Wang, C., Wang, Y., Ye, W., Zhang, Y., Chang, Y., Yu, P. S., Yang, Q., Xie, X., & Huang, H. A Survey on Evaluation of Large Language Models. ACM Transactions on Intelligent Systems and Technology, 2024, 15(3): 1–45. DOI: 10.1145/3641289.

[2] Zhao, W., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Z., Jiang, H., Liu, R., Fang, Y., Zhao, Y., Yang, X., Zhang, Y., et al. Explainability for Large Language Models: A Survey. ACM Transactions on Intelligent Systems and Technology, 2024, 15(2): 1–38. DOI: 10.1145/3639372.

[3] Gallegos, I. O., Rossi, R. A., Barrow, J., Tanjim, M. M., Kim, S., Dernoncourt, F., Yu, T., Zhang, R., & Ahmed, N. K. Bias and Fairness in Large Language Models: A Survey. Computational Linguistics, 2024, 50(3): 1097–1179. DOI: 10.1162/coli_a_00524.

[4] Taeihagh, A. Governance of Generative AI. Policy and Society, 2025, 44(1): 1–22. DOI: 10.1093/polsoc/puaf001.

[5] Janssen, M., van den Hoven, J., Helberger, N., & Menssen, L. Responsible governance of generative AI: conceptualizing AI governance as a complex adaptive system(s). Policy and Society, 2025, 44(1): 38–51. DOI: 10.1093/polsoc/puae040.

[6] Khanal, Y. P., Zhang, Y., & Taeihagh, A. Why and how is the power of Big Tech increasing in the policy process? The case of generative AI. Policy and Society, 2025,

44(1): 52–69. DOI: 10.1093/polsoc/puae012.

[7] Ulnicane, I. Governance fix? Power and politics in controversies about governing generative AI. Policy and Society, 2025, 44(1): 70–84. DOI: 10.1093/polsoc/puae022.

[8] Judge, R., Nitzberg, M., & Russell, S. When code isn't law: rethinking regulation for artificial intelligence. Policy and Society, 2025, 44(1): 85–97. DOI: 10.1093/polsoc/puae020.

[9] Martin, K., & Waldman, A. E. Are Algorithmic Decisions Legitimate? The Effect of Process and Outcomes on Perceptions of Legitimacy of AI Decisions. Journal of Business Ethics, 2023, 183(3): 653–670. DOI: 10.1007/s10551-021-05032-7.

[10] Gerdon, F., Bach, R. L., Kern, C., & Kreuter, F. Social impacts of algorithmic decision-making: A research agenda for the social sciences. Big Data & Society, 2022, 9(1): 1–13. DOI: 10.1177/20539517221089305.

[11] Savolainen, L. The shadow banning controversy: perceived governance and algorithmic folklore. Media, Culture & Society, 2022, 44(6): 1091–1109. DOI: 10.1177/01634437221077174.

[12] Thomas, M., et al. The case for a broader approach to AI assurance: addressing "hidden" harms in the development of artificial intelligence. AI & Society, 2025, 40: 1469–1484. DOI: 10.1007/s00146-024-01950-y.

[13] Bartsch, S., et al. The Present and Future of Accountability for AI Systems: A Bibliometric Analysis. Information Systems Frontiers, 2025, 27: 2463–2484. DOI: 10.1007/s10796-025-10636-9.

[14] Dastani, M., & Yazdanpanah, V. Responsibility of AI Systems. AI & Society, 2023, 38: 843–852. DOI: 10.1007/s00146-022-01481-4.

[15] Hedlund, J., & Persson, E. Expert responsibility in AI development. AI & Society, 2024, 39: 453–464. DOI: 10.1007/s00146-022-01498-9.

[16] Cheong, M., Caliskan, A., & Kohno, T. Safeguarding human values: rethinking US law for generative AI's societal impacts. AI and Ethics, 2025, 5(2): 1433–1459. DOI: 10.1007/s43681-024-00451-4.

[17] Ning, Y., et al. Generative artificial intelligence and ethical considerations in health care: a scoping review and ethics checklist. The Lancet Digital Health, 2024, 6(11): e848–e856. DOI: 10.1016/S2589-7500(24)00143-2.

[18] Gregor, S. Responsible Artificial Intelligence and Journal Publishing. Journal of the Association for Information Systems, 2024, 25(1): 48–60. DOI: 10.17705/1jais.00863.

[19] Formosa, P., Kashyap, B., & Sahebi, S. Generative AI and the future of democratic citizenship. Digital Government: Research and Practice, 2025, 6(2): 1–10. DOI: 10.1145/3674844.

[20] Munoriyarwa, A., & de-Lima-Santos, M.-F. Generative AI and the future of news:

examining AI's agency, power, and authority. Journalism Practice, 2025, 19(10): 2177–2188. DOI: 10.1080/17512786.2025.2545448.

[21] Meng, B. "This is China's Sputnik Moment": The Politics and Poetics of Artificial Intelligence. Interventions, 2023, 25(3): 351–369. DOI: 10.1080/1369801X.2021.2003227.

[22] Lagna, A. Asset manager capitalism and the political economy of artificial intelligence. Review of International Political Economy, 2025, 32(2): 512–528. DOI: 10.1080/09692290.2024.2432393.

[23] Valdivia, A. The supply chain capitalism of AI: a call to (re)think algorithmic harms and resistance through environmental lens. Information, Communication & Society, 2025, 28(12): 2118–2134. DOI: 10.1080/1369118X.2024.2420021.

[24] Dauvergne, P. Is artificial intelligence greening global supply chains? Exposing the political economy of environmental costs. Review of International Political Economy, 2022, 29(3): 696–718. DOI: 10.1080/09692290.2020.1814381.

[25] Shibata, S. Digitalization or flexibilization? The changing role of technology in the political economy of Japan's platform service sector. Review of International Political Economy, 2022, 29(5): 1549–1576. DOI: 10.1080/09692290.2021.1935294.

[26] Mao, Y., & Shi-Kupfer, K. Online public discourse on artificial intelligence and ethics in China: context, content, and implications. AI & Society, 2023, 38(1): 373–389. DOI: 10.1007/s00146-021-01309-7.

[27] Amoore, L., Bennett, S., & Campolo, A. Politics of the prompt: Government in the age of generative AI. Economy and Society, 2025, 54(3): 1–24. DOI: 10.1080/03085147.2025.2560177.