applied sciences

MDPI

Article

A Large Language Model-Driven Strategic Evaluation Framework via Time-Series Directed Acyclic Graphs

Mingyin Zou, Xiaomin Zhu, Yanqing Ye *, Guangrong You and Li Ma

Strategic Assessments and Consultation Institute, Academy of Military Science, Beijing 100091, China; zoumingyin20@alumni.nudt.edu.cn (M.Z.)

* Correspondence: yeyanqing09@alumni.nudt.edu.cn; Tel.: +86-186-0096-9825

Abstract

Strategic evaluation is essential for decision-making under uncertainty. Yet existing qualitative and quantitative methods—including chat-oriented large language model (LLM) evaluations—are difficult to deploy in complex, dynamic environments. They often fail to represent nonlinear causal dependencies among indicators, account for temporal lags, or support scalable reasoning. To address these limitations, we propose an LLM-driven strategic evaluation framework with three innovations. First, the framework integrates LLMs across the evaluation lifecycle and couples their qualitative reasoning with quantitative model computation, improving both efficiency and deployability. Second, we introduce a Time-Series Directed Acyclic Graph (TS-DAG) indicator system that explicitly encodes causal structure and time-lagged interdependencies. Third, we develop an LLM-driven procedure that automatically derives the TS-DAG architecture and instantiates its computational parameters, reducing reliance on expert-only construction. We validate the framework through an empirical study of the new energy vehicle market, complemented by baseline algorithm comparisons and sensitivity analyses. The results show that the proposed framework can uncover core indicators, capture competitive dynamics, and explain long-term strategic outcomes across varying environmental conditions. Overall, the framework provides a robust solution for strategic evaluation in complex settings, bridging qualitative strategic reasoning and quantitative, data-driven analysis.

Keywords: strategic evaluation; temporal-sequential directed acyclic graph; large language model; human–AI co-reasoning

1. Introduction

As global competition accelerates and decision environments grow more complex, strategic evaluation has become increasingly important. Strategic evaluation provides a structured way to analyze, quantify, and predict the consequences of alternative strategies, thereby supporting decision-making [1]. However, strategic settings typically involve multiple interacting agents, nonlinear causal dependencies, and substantial uncertainty. These factors make rigorous evaluation challenging, yet essential for modern governance and enterprise planning.

Strategic evaluation typically involves two coupled mechanisms. First, strategic actions directly affect target indicators through resource allocation, capability development, market positioning, and related levers. Second, these direct effects propagate through an indicator network, where changes in one variable induce downstream adjustments in others. For example, increased Research and Development (R&D) investment may reduce

Academic Editor: Paulo Santos

Received: 4 January 2026
Revised: 11 February 2026
Accepted: 12 February 2026
Published: 18 February 2026

short-term profitability but later improve innovation output and market share [2]. Crucially, these dependencies are both causal and time-lagged: strategic effects rarely appear immediately and often emerge after heterogeneous delays. This high-dimensional, nonlinear propagation makes strategic evaluation methodologically demanding, yet essential for anticipating long-term consequences and managing uncertainty.

Traditional strategic evaluation methods span both qualitative and quantitative paradigms. On the qualitative side, SWOT (Strengths, Weaknesses, Opportunities, and Threats) is commonly used to organize internal and external factors [3,4]. PESTEL (Political, Economic, Social, Technological, Environmental, and Legal) analysis supports macro-environmental scanning [5,6], and the Balanced Scorecard links strategic objectives to multidimensional performance indicators [7]. Despite their widespread use, these approaches rely heavily on manual expert input. This reliance introduces subjectivity, constrains scalability, and limits the granularity and data-driven interpretability needed to model complex strategic dynamics.

In the quantitative domain, multi-criteria decision-making (MCDM) methods are widely used to provide numerical assessments [8]. Representative examples include AHP (Analytic Hierarchy Process), which derives weights from hierarchical comparisons [9]; TOPSIS (Technique for Order of Preference by Similarity to Ideal Solution), which ranks alternatives by their distance to ideal reference points [10]; and ELECTRE (Elimination and Choice Expressing Reality), which supports decision-making via outranking relations [11]. In practice, when these methods are applied to qualitative strategy evaluation, experts must still specify how each strategy affects the underlying indicators [12,13]. The model then propagates these expert-defined impacts to produce final evaluation scores. Despite their usefulness, most MCDM pipelines rely on linear aggregation and compress complex strategic dynamics into a single scalar. As a result, they struggle to represent nonlinear causal dependencies and time-lagged effects that often govern how strategic influence propagates in real settings.

Large language models (LLMs) have recently reshaped how complex problems are formulated and solved. Built on the Transformer architecture [14], these models use self-attention and are pre-trained on large-scale corpora via self-supervised learning [15,16]. This progress has also changed the landscape of strategic evaluation. LLMs provide strong semantic reasoning and contextual understanding, which helps translate high-level strategic concepts into operational, measurable indicators. When integrated with evaluation pipelines, LLMs can structure qualitative considerations into quantitative representations and automate the interpretation of unstructured inputs. This capability complements traditional models by better handling nonlinear interactions and supporting more precise, data-driven strategic assessment.

Accordingly, integrating LLMs into both qualitative [17–19] and quantitative [20–22] evaluation paradigms has become an active research direction. A common line of work treats LLMs as virtual domain experts and uses dialog-driven prompting to elicit strategic judgments or simulate decision-making [21]. However, many existing studies remain primarily conversational and are not tightly coupled with a rigorous evaluation pipeline. Using LLMs as standalone evaluators raises practical concerns: their generative nature limits interpretability, and stochastic outputs can reduce reliability. As a result, LLM-only assessments may be fluent yet fail to meet the precision and robustness requirements of high-stakes strategic evaluation.

In summary, despite progress in both traditional methods and LLMs, strategic evaluation still faces three key limitations: (1) Modeling efficiency and integration. Strategic evaluation is knowledge-intensive and highly complex, which makes model construction costly. Traditional approaches rely on prolonged expert intervention, while many LLM-based

efforts remain at the level of ad hoc dialog. A systematic framework that embeds LLMs across the evaluation lifecycle—while maintaining scientific rigor and interpretability—has yet to be established. (2) Dynamic causality and temporal lags. Strategic indicators interact through nonlinear causal mechanisms and time-lagged dependencies. However, many existing paradigms rely on static structures or linear aggregation, which cannot faithfully represent how strategic influence propagates over time. (3) Scalability of indicator-system construction. Building high-dimensional indicator systems and specifying their relationships is difficult to scale when it depends primarily on domain experts. This motivates automated methods that can derive indicator architectures and instantiate their parameters in a way that better reflects underlying strategic dynamics.

To address these challenges, we propose a methodology that couples LLM-based reasoning with explicit structural modeling. The contributions are summarized as follows:

1. LLM-Driven Strategic Evaluation Framework: We present a framework that integrates LLMs across the evaluation lifecycle to support the analysis of complex strategic environments. Evaluation elements are elicited and refined by the LLM, and candidate indicators are derived to align with the evaluation purpose and rules.
2. TS-DAG Indicator System Architecture: A time-series directed acyclic graph (TS-DAG) representation is introduced. Acyclicity is enforced for zero-lag dependencies while causal and time-lagged relationships among indicators are encoded, enabling sequential computation and interpretable propagation of strategic influence over time.
3. LLM-Driven TS-DAG Instantiation: We propose an automated procedure in which the TS-DAG structure is derived and node-level computational forms and parameters are instantiated by an LLM, reducing reliance on expert-only modeling. The resulting indicator system captures nonlinear relationships, causal chains, and delayed effects.
4. Empirical Validation in the NEV Market: The proposed framework is validated in a new energy vehicle (NEV) market case study. Competitive dynamics among three representative participants (legacy automaker, technology innovator, and price leader) are analyzed, and the influence of key variables (e.g., consumer confidence, subsidy level, and battery breakthroughs) on market-share outcomes is evaluated. In addition, we conduct comparative experiments against two baselines (AHP-based and prompt-based) to assess the robustness of market-share evolution.

2. Materials and Methods

2.1. LLM-Driven Strategic Evaluation Framework

Figure 1 provides an overview of the proposed LLM-driven strategic evaluation framework. The workflow comprises four interconnected phases: (i) evaluation-element specification, (ii) indicator-system construction, (iii) strategic deduction and evaluation, and (iv) post hoc data analysis. All LLMs used as algorithmic components are denoted by Φ, and the corresponding prompts are provided in the Supplementary Materials. By coupling expert input with LLM-based reasoning at each phase, the framework forms a closed-loop pipeline that maps problem definitions to quantitative insights.

Evaluation-Element Determination: The workflow begins with human–AI collaboration. Domain experts and the LLM jointly specify the evaluation constraints, including (i) the evaluation topic (strategic context), (ii) the evaluation objects (entities under review), (iii) the evaluation purpose (target outcomes), and (iv) the evaluation rules (boundaries and logical constraints).

Indicator System Construction: This phase uses LLM-based semantic reasoning to derive the indicator system topology (e.g., causal and hierarchical relations) and to instantiate the functional dependencies and parameters needed for quantitative computation.

Strategic Deduction and Evaluation: Given perturbations in key environmental variables, the framework generates strategy sequences and evaluates their impacts. The LLM acts as an inference engine that maps each strategic action to quantitative perturbations on the indicator system, enabling influence to propagate through the network over time.

Data Analysis: After deduction, the framework aggregates simulation traces and extracts actionable insights through (i) trend analysis (temporal evolution), (ii) correlation analysis (interdependencies), and (iii) importance analysis (key drivers).

The following subsections detail the core components of the framework, including the TS-DAG indicator system, its automated construction, the dynamic deduction mechanism, and the post hoc data analysis methods.

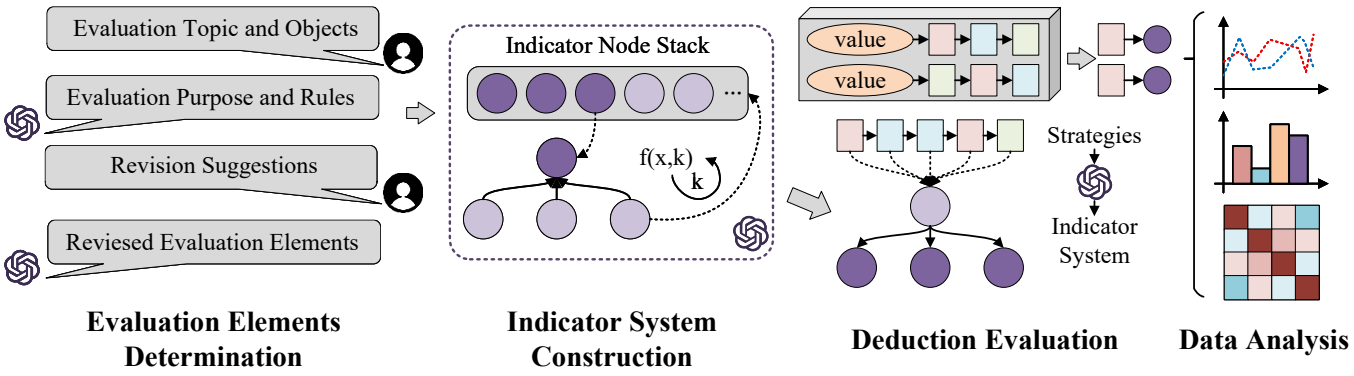


Figure 1. Schematic overview of the LLM-driven strategic evaluation framework.

2.2. Temporal-Sequential Directed Acyclic Graph

Traditional indicator systems are often implemented as hierarchical trees, general network structures, or hybrid tree–network structures. While effective for static settings, these representations are ill-suited to strategic evaluation, where indicators span multiple domains (e.g., politics, economics, military, and diplomacy) and frequently exhibit cross-domain dependencies. Tree-structured systems typically compute node values in a strict bottom-up order. This design makes it difficult to represent lateral dependencies among indicators at the same level. Network-based systems can express such dependencies but may introduce cycles. Cycles can create ambiguous update orders and redundant computation; for example, if A influences B, B influences C, and C influences A, then A may be updated multiple times within a single iteration. Moreover, strategic influence is rarely instantaneous: causal effects often emerge with temporal delays. Conventional tree and network representations do not explicitly encode these time-lagged dependencies, limiting their ability to model how strategic effects propagate over time.

To address these challenges, we propose a Temporal-Sequential Directed Acyclic Graph (TS-DAG) representation for strategic indicator systems (Figure 2). The TS-DAG contains three node types: input nodes, constant nodes, and compute nodes. Each compute node derives its state from a function of its predecessors' current values and time-lagged (historical) values.

2.2.1. Mathematical Modeling

A TS-DAG contains heterogeneous node types, each serving a distinct role in the computation graph. Unlike standard static graphs, its dependencies include temporal offsets: a node at time t may depend on the value of another node from $t - \tau$.

To represent this structure and ensure computability, we formalize a TS-DAG as a triple $\mathcal{G} = (V, E, \mathcal{F})$, where V is the node set, E is the directed edge set with temporal lags, and $\mathcal{F} = \{f_v \mid v \in V_{\text{comp}}\}$ is the collection of computation functions for the compute nodes.

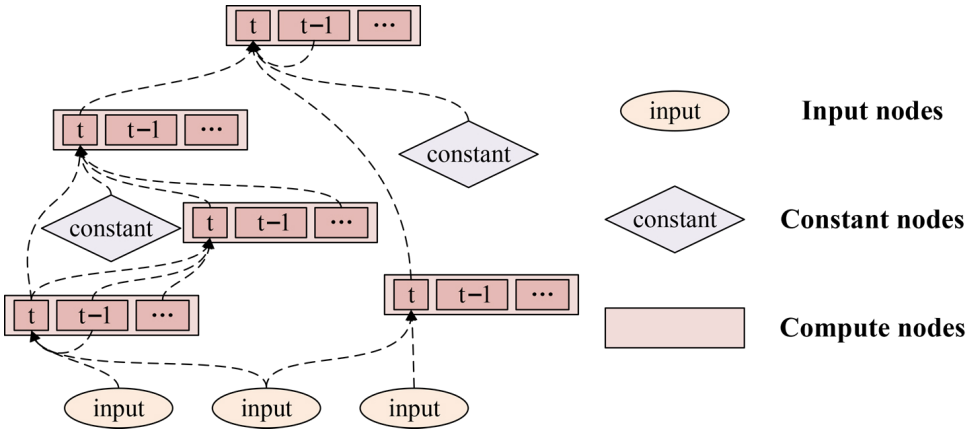


Figure 2. An example of a TS-DAG.

Node Model. Nodes in strategic evaluation systems play heterogeneous roles: some encode fixed parameters, some ingest exogenous time-series signals, and others compute derived states. To capture this heterogeneity, we partition V into three pairwise disjoint subsets:

$$V = V_{\text{const}} \cup V_{\text{input}} \cup V_{\text{comp}}, \quad V_i \cap V_j = \emptyset \ \ \forall\, i \neq j \tag{1}$$

where V_{const} denotes constant nodes with time-invariant values, V_{input} denotes input nodes driven by external time series, and V_{comp} denotes compute nodes whose states are derived via functions in $\mathcal{F}$.

Edge Model. Temporal dependencies often include explicit time lags: a node's state at time t may depend on another node's state at time $t - \tau$. We therefore define the directed edge set as $E \subseteq V \times V \times N \geq 0$. Each edge $e = (u, v, \tau) \in E$ indicates that node v consumes node u with lag τ.

Function Model. The function model specifies how node states evolve and ensures that a TS-DAG is executable. The update rule depends on the node type. For constant nodes $v \in V_{\text{const}}$, the state is time-invariant:

$$x_v(t) = c_v, \quad \forall\, t \geq 0 \tag{2}$$

where $c_v \in \mathbb{R}$ is a predefined constant.

For input nodes $v \in V_{\text{input}}$, the state is given by an exogenous time series:

$$x_v(t) = I_v(t), \quad \forall\, t \geq 0 \tag{3}$$

where $I_v : \mathbb{N}_{\geq 0} \to \mathbb{R}$ denotes the external input signal.

For each compute node $v \in V_{\text{comp}}$, we define the predecessor set $S_v = \{(u, \tau) \mid (u, v, \tau) \in E\}$ and its cardinality $k_v = |S_v|$. We then introduce an indexing bijection $\sigma_v : \{1, \ldots, k_v\} \to S_v$ such that $\sigma_v(j) = (u_j, \tau_j)$.

At time t, we assemble the input vector $\mathbf{z}_v(t) \in \mathbb{R}^{k_v}$ by retrieving each predecessor value at the required lag. To keep the update rule well-defined when $t - \tau_j < 0$, we enforce the following boundary condition:

$$z_{v,j}(t) = \begin{cases} x_{u_j}(t - \tau_j) & \text{if } t - \tau_j \geq 0 \\ \Gamma(u_j) & \text{otherwise} \end{cases} \tag{4}$$

where $\Gamma : V \to \mathbb{R}$ provides a default value for each node.

Given $\mathbf{z}_v(t)$, the node state is computed by applying the update function $f_v \in \mathcal{F}$:

$$x_v(t) = f_v(\mathbf{z}_v(t)), \quad f_v : \mathbb{R}^{k_v} \to \mathbb{R} \tag{5}$$

2.2.2. Topological Constraints and Sequential Computation

A key property of TS-DAGs is that temporal edges can form feedback loops across time. Specifically, a past state of node u can influence node v, and a past state of node v can, at a different lag, influence node u. In contrast, we forbid instantaneous mutual dependence within the same time step; otherwise, the update order would be undefined and the computation ill-posed. We enforce this constraint by defining the zero-lag subgraph and requiring it to be acyclic.

Zero-Lag Subgraph. We define the zero-lag subgraph $G_0 = (V, E_0)$, where $E_0 = \{(u, v) \mid (u, v, 0) \in E\}$ contains exactly the edges with zero temporal lag. We enforce that G_0 is a directed acyclic graph (DAG). This constraint eliminates instantaneous circular dependence and yields a well-defined computation order at each time step.

Topological Ordering. To derive an executable computation sequence, Kahn's algorithm [23] is applied on the zero-lag subgraph G_0. This procedure constructs a linear ordering $\mathcal{L} = (v_1, v_2, \ldots, v_{|V|})$ that satisfies the precedence constraint:

$$\forall (u, v, 0) \in E : \quad \text{pos}(u) < \text{pos}(v) \tag{6}$$

where $\text{pos}(v)$ denotes the position of node v in $\mathcal{L}$. As a result, when $x_v(t)$ is computed for any $v \in V_{\text{comp}}$, all required zero-lag predecessor states at time t have already been computed.

We define the zero-lag in-degree of node v as $d^0_{\text{in}}(v) = |\{u \mid (u, v, 0) \in E\}|$. Kahn's algorithm initializes a queue with the root node set $R = \{v \in V \mid d^0_{\text{in}}(v) = 0\}$. This set includes V_{const}, V_{input}, and any compute node that depends only on historical states. The algorithm then repeatedly removes a node from the queue, appends it to $\mathcal{L}$, and deletes its outgoing zero-lag edges. These deletions decrement the corresponding in-degrees and may expose new roots.

We validate acyclicity by checking whether the algorithm produces a full ordering. If $|\mathcal{L}| < |V|$ at termination, then G_0 contains a directed cycle and the TS-DAG specification is invalid. Algorithm 1 details the full procedure.

Algorithm 1 Topological Sort for TS-DAG

Require: TS-DAG $\mathcal{G} = (V, E, \mathcal{F})$
Ensure: A linear ordering $\mathcal{L}$, or $\perp$ if cycle detected

$E_0 \leftarrow \{(u, v) \mid (u, v, 0) \in E\}$
$\mathcal{L} \leftarrow [\,]$; $\mathcal{Q} \leftarrow [\,]$
$d_{\text{in}}[v] \leftarrow 0$, $\forall v \in V$
for each $(u, v) \in E_0$ **do**
    $d_{\text{in}}[v] \leftarrow d_{\text{in}}[v] + 1$
for each $v \in V$ **with** $d_{\text{in}}[v] = 0$ **do**
    $\mathcal{Q}$.ENQUEUE(v)
while $\mathcal{Q} \neq \emptyset$ **do**
    $u \leftarrow \mathcal{Q}$.DEQUEUE(); $\mathcal{L}$.APPEND(u)
    for each $(u, v) \in E_0$ **do**
        $d_{\text{in}}[v] \leftarrow d_{\text{in}}[v] - 1$
        if $d_{\text{in}}[v] = 0$ **then**
            $\mathcal{Q}$.ENQUEUE(v)
return $|\mathcal{L}| = |V|$? $\mathcal{L} : \perp$

Because the graph topology is fixed during simulation (only node states $x_v(t)$ evolve), $\mathcal{L}$ is computed once at initialization. It is then cached and reused for all time steps $t \in \{0, 1, \ldots, T\}$, which amortizes the $\mathcal{O}(|V| + |E_0|)$ sorting cost across the full horizon.

2.2.3. TS-DAG State Propagation

Given the topological ordering $\mathcal{L}$, the TS-DAG is evaluated sequentially over the simulation horizon. At each time step t, nodes are processed in the order specified by $\mathcal{L}$. This ordering resolves all instantaneous (zero-lag) dependencies before any dependent computation is executed.

For each compute node $v \in V_{\text{comp}}$, we first assemble its input vector $\mathbf{z}_v(t) \in \mathbb{R}^{k_v}$ using Equation (4). We then instantiate the node state by applying its transfer function, as specified in Equation (5).

Algorithm 2 summarizes the full propagation procedure. We establish correctness via two invariants. (i) For any zero-lag edge $(u, v, 0) \in E$, the ordering places u before v, so $x_u(t)$ is available when we construct $\mathbf{z}_v(t)$. (ii) For any lagged edge with $\tau > 0$, temporal causality ensures that $x_u(t - \tau)$ has already been computed in an earlier time step.

The algorithm runs in $\mathcal{O}(T \cdot (|V| + |E|))$ time. It uses $\mathcal{O}(|V| \cdot T)$ space to store node trajectories.

Algorithm 2 TS-DAG State Propagation

Require: TS-DAG $\mathcal{G} = (V, E, \mathcal{F})$, horizon T, default mapping Γ
Ensure: State trajectories $\{x_v(t)\}_{v \in V, t \in [0,T]}$

$\mathcal{L} \leftarrow$ TopologicalSort($\mathcal{G}$) (Algorithm 1)
if $\mathcal{L} = \perp$ **then return** Error
for $t = 0$ **to** T **do**
  for each v in $\mathcal{L}$ **do**
    if $v \in V_{\text{const}}$ **then**
      $x_v(t) \leftarrow c_v$
    else if $v \in V_{\text{input}}$ **then**
      $x_v(t) \leftarrow I_v(t)$
    else
      for $j = 1$ **to** k_v **do**
        $(u_j, \tau_j) \leftarrow \sigma_v(j)$
        $z_{v,j}(t) \leftarrow x_{u_j}(t - \tau_j)$ **if** $t \geq \tau_j$ **else** $\Gamma(u_j)$
      $x_v(t) \leftarrow f_v(\mathbf{z}_v(t))$
return $\{x_v(t)\}_{v \in V, t \in [0,T]}$

2.3. Automated Construction of the Strategic TS-DAG

Section 2.2 specifies the TS-DAG formally and establishes its formal foundation. However, manually instantiating a TS-DAG for a complex strategic scenario is labor-intensive and prone to cognitive bias. To mitigate these issues, we propose an automated construction framework that couples the semantic reasoning capability of LLMs with the structural rigor of graph theory. The framework derives the network topology and instantiates node-level parametric logic in a single pipeline.

2.3.1. LLM-Driven TS-DAG Structure Generation

Manually constructing TS-DAGs for complex strategic evaluation is impractical. The indicator space is combinatorial, and temporal interdependencies further increase the modeling burden.

We propose an LLM-driven, top-down recursive decomposition procedure. The LLM derives candidate indicators, decomposes composite indicators into sub-indicators, and assigns temporal lags to the resulting dependencies. This design preserves semantic alignment with high-level strategic intent while enabling systematic graph construction.

Strategic evaluation is specified by four elements: (i) the evaluation theme (domain of interest), (ii) the evaluation object (entity under evaluation), (iii) the evaluation pur-

pose (decision intent), and (iv) the evaluation rule (domain constraints and normative criteria). Together, these elements define the evaluation scope and the criteria used to judge outcomes.

We encode them as a Strategic Context Tuple $\mathcal{C} = \langle \text{Theme}, \text{Object}, \text{Purpose}, \text{Rule} \rangle$. This tuple serves as the semantic anchor that guides subsequent TS-DAG construction.

Given $\mathcal{C}$, an LLM-driven agent derives the indicator hierarchy via top-down recursive decomposition. It starts with a root concept node derived from $\mathcal{C}$ and then iteratively expands the graph.

For each node v, the agent classifies it as either atomic (directly observable) or composite (derived from other indicators). It assigns atomic nodes to V_{input}. It assigns composite nodes to V_{comp} and recursively decomposes them until all leaves are atomic. Figure 3 illustrates this structure-generation process.

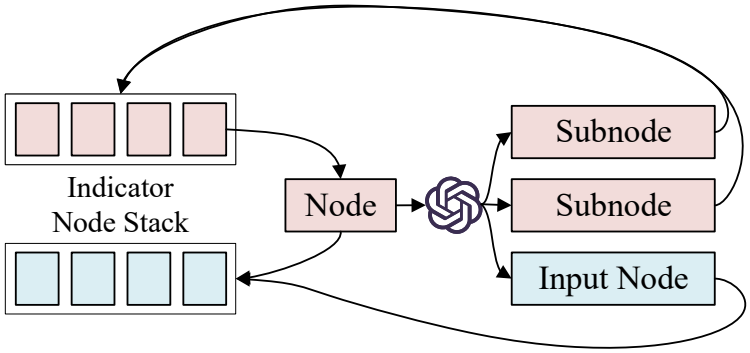


Figure 3. Schematic diagram of the structure generation of TS-DAG based on the LLM-driven approach.

The decomposition is inherently temporal. Instead of assuming instantaneous dependencies and repairing cycles post hoc, the agent infers a temporal lag τ for each predecessor by reasoning about the underlying causal mechanism. This design captures feedback across time while enforcing acyclicity in the zero-lag subgraph G_0.

Concurrently, the agent synthesizes the symbolic form of each transfer function f_v. It leaves parameters unspecified for subsequent calibration. Algorithm 3 formalizes the overall procedure.

2.3.2. Data-Driven Parameter Identification

The symbolic transfer functions generated in Algorithm 3 must be instantiated with concrete parameters before we can perform quantitative evaluation. However, manual tuning is impractical, and labeled strategic data are typically unavailable.

We therefore adopt an iterative refinement procedure. In this procedure, the LLM-driven agent generates diagnostic scenarios and evaluates the resulting model outputs against domain knowledge.

For each compute node $v \in V_{\text{comp}}$ with predecessor set $S_v = \{(u_j, \tau_j)\}_{j=1}^{k_v}$, we instantiate its parameters via the following iterative procedure.

Step 1 (Initialize). The agent generates a scenario set $\mathbf{X}_v \in \mathbb{R}^{N \times k_v}$ that covers boundary cases, typical ranges, and extreme conditions. It also proposes an initial parameter vector $\theta_v^{(0)}$:

$$\mathbf{X}_v, \theta_v^{(0)} \leftarrow \Phi(\textsc{Initialize}, v, S_v, \mathcal{C}) \tag{7}$$

Step 2 (Forward evaluation). At iteration t, we evaluate the symbolic function on all scenarios:

$$\hat{y}_v^{(t,n)} = f_v(\mathbf{x}_v^{(n)}; \theta_v^{(t)}), \quad n = 1, \ldots, N \tag{8}$$

Algorithm 3 LLM-Driven TS-DAG Structure Generation

Require: Strategic Context $\mathcal{C}$, LLM-driven Agent Φ
Ensure: TS-DAG structure (V, E) with symbolic functions $\{f_v\}_{v \in V_{\text{comp}}}$

$V \leftarrow \{v_{\text{root}}\}$; $V_{\text{input}} \leftarrow \emptyset$; $V_{\text{comp}} \leftarrow \emptyset$; $E \leftarrow \emptyset$
$S \leftarrow [v_{\text{root}}]$
while $S \neq \emptyset$ **do**
  $v \leftarrow S.\text{POP}()$
  if $\text{type}_v =$ `atomic` **then**
    $V_{\text{input}} \leftarrow V_{\text{input}} \cup \{v\}$
  else
    $V_{\text{comp}} \leftarrow V_{\text{comp}} \cup \{v\}$
    $\{(u_i, \tau_i)\}_{i=1}^{k} \leftarrow \Phi(\text{DECOMPOSE}, v, \mathcal{C})$
    for $i = 1$ **to** k **do**
      if $u_i \notin V$ **then**
        $V \leftarrow V \cup \{u_i\}$; $S.\text{PUSH}(u_i)$
      $E \leftarrow E \cup \{(u_i, v, \tau_i)\}$
    $f_v \leftarrow \Phi(\text{SYNTHESIZESYMBOLIC}, v, \{(u_i, \tau_i)\})$
return $(V, E, \{f_v\}_{v \in V_{\text{comp}}})$

Step 3 (Feedback). The agent validates each prediction against domain knowledge and returns directional feedback:

$$\mathbf{d}_v^{(t)} = \Phi(\text{EVALUATE}, \mathbf{X}_v, \hat{\mathbf{y}}_v^{(t)}, \mathcal{C}) \tag{9}$$

where $d_v^{(t,n)} \in \{-1, 0, +1\}$ indicates whether $\hat{y}_v^{(t,n)}$ is underestimated, appropriate, or overestimated.

Step 4 (Update). The agent aggregates the feedback and adjusts the parameters:

$$\theta_v^{(t+1)} = \Phi(\text{ADJUST}, \theta_v^{(t)}, \mathbf{d}_v^{(t)}, \mathbf{X}_v, \mathcal{C}) \tag{10}$$

Stopping criterion. We terminate the refinement when either the parameter change is small or the feedback indicates sufficient alignment:

$$\|\theta_v^{(t+1)} - \theta_v^{(t)}\| < \epsilon \quad \text{or} \quad \frac{1}{N} \sum_{n=1}^{N} |d_v^{(t,n)}| < \delta \tag{11}$$

After T iterations, we obtain the final parameter vector $\hat{\theta}_v = \theta_v^{(T)}$. We execute this procedure sequentially along the topological ordering to respect computational dependencies.

2.4. Strategic Action Quantification and Propagation

The TS-DAG executes on numerical inputs, whereas strategic decisions are naturally expressed in language (e.g., 'launch a diplomatic initiative' or 'accelerate technology deployment'). A principled mechanism is therefore required to translate each qualitative action into quantitative perturbations of the exogenous input nodes $\mathcal{V}_{in}$.

We use an LLM-driven agent as a domain-aware evaluator. Given a chronologically ordered action sequence $\mathcal{S} = (s_1, s_2, \ldots, s_K)$, where action s_k is enacted at time t_k, the agent infers the impact of s_k on each input node. This inference is context-dependent. Specifically, it conditions on both the prior action history $(s_1, \ldots, s_{k-1})$ and the current system state $\mathbf{x}^{(t_k-1)} = \{x_v^{(t_k-1)}\}_{v \in \mathcal{V}}$. As a result, identical actions can yield different outcomes under different histories and states.

For each action s_k and input node $v \in \mathcal{V}_{in}$, the agent returns a discrete impact label $I_v^{(k)} \in \mathcal{I}$, where $\mathcal{I} = \{\texttt{L+}, \texttt{S+}, \texttt{N}, \texttt{S-}, \texttt{L-}\}$ denotes large positive, small positive, neutral, small negative, and large negative effects.

We convert this qualitative label into a multiplicative factor via a predefined mapping $M : \mathcal{I} \to \mathbb{R}^+$, with values $\{1.4, 1.1, 1.0, 0.9, 0.6\}$. We then update each input node as

$$x_v^{(t_k)} = x_v^{(t_k-1)} \cdot M\Big(\Phi(s_k, (s_1, \ldots, s_{k-1}), \mathbf{x}^{(t_k-1)}, v)\Big), \quad \forall v \in \mathcal{V}_{in} \tag{12}$$

where $\Phi(\cdot)$ denotes the agent's impact-evaluation function. After we update all input nodes, Algorithm 2 propagates the perturbations through the TS-DAG. This propagation yields the resulting direct and indirect consequences across the indicator system.

We implement strategic deduction as an interaction between two roles: decision-makers and evaluators. Decision-makers propose strategic actions, whereas evaluators assess their implications. This separation mirrors real-world practice, where strategy formulation and impact evaluation require distinct expertise.

Within the TS-DAG architecture, we operationalize this interaction as a three-phase cyclic process: (i) strategic gaming, (ii) strategic evaluation, and (iii) graph propagation. Figure 4 illustrates the overall procedure.

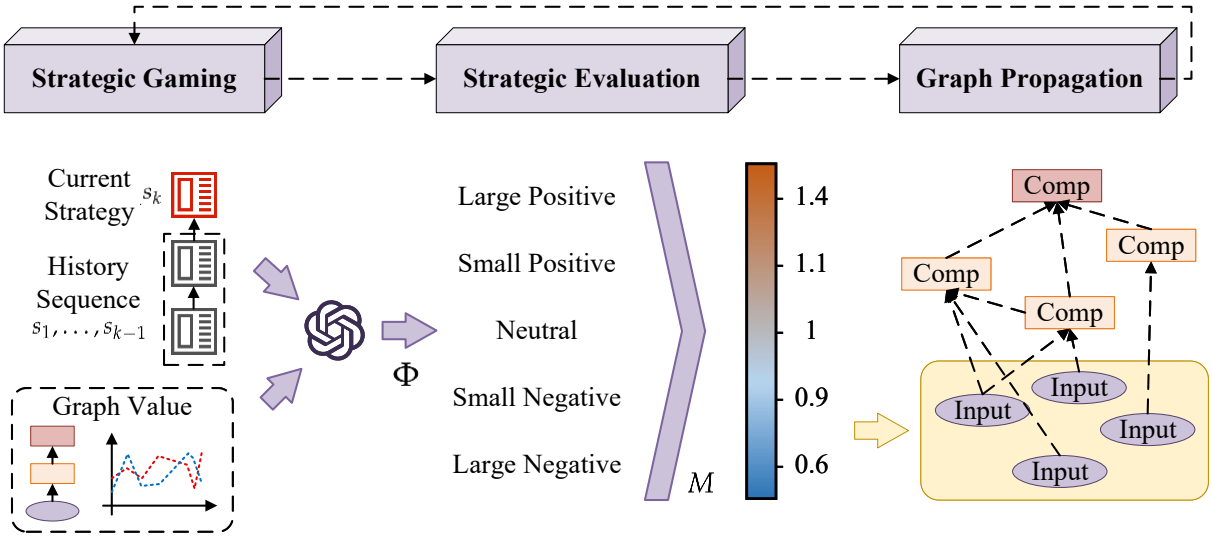


Figure 4. Schematic diagram of strategic deduction based on TS-DAG.

Strategic gaming. Multiple agents propose strategic actions, each representing a distinct stakeholder within the simulated environment. Each agent derives its proposal from the current system state and its anticipation of other agents' responses.

Strategic evaluation. This phase quantifies the joint effect of the proposed (and potentially conflicting) actions. Given the current actions, the action history, and each participant's capability metrics, the evaluator derives a net perturbation for each input node. The evaluation accounts for interaction effects, i.e., actions may reinforce, offset, or amplify one another.

Graph propagation. We then propagate the evaluated perturbations through the TS-DAG. Starting from the perturbed input nodes $\mathcal{V}_{in}$, the TS-DAG executes its causal update rules to compute all downstream node states. This propagation exposes both immediate effects and delayed, higher-order consequences.

2.5. Extracting Strategic Insights Through Comparative Analysis

The goal of strategic deduction is to derive actionable insights. However, a single simulation run may mainly reflect idiosyncratic conditions. To extract robust patterns, results are compared across deduction runs under different key-variable configurations. This comparative design helps reveal the underlying logic of strategic dynamics rather than scenario-specific outcomes.

Our workflow proceeds in two steps. First, key-variable combinations that define the strategic landscape are identified. Second, for each combination, multiple deduction rounds are run, and the resulting trajectories are analyzed. Insights are then extracted using three complementary analyses: trend analysis, correlation analysis, and feature importance analysis.

2.5.1. Trend Analysis

Trend analysis characterizes how each agent's indicators evolve over time within a fixed deduction scenario. To make this analysis precise, we first define the data generated by each deduction round.

Deduction definition. Fix a key-variable configuration $\mathcal{K} = \{k_1, k_2, \ldots, k_m\}$. N independent deduction rounds are run. The r-th round ($r \in \{1, 2, \ldots, N\}$) is denoted by the tuple:

$$\mathcal{R}_{\mathcal{K}}^{(r)} = (\mathcal{K}, \mathcal{C}^{(r)}, \mathcal{D}_{\mathcal{K}}^{(r)}) \tag{13}$$

where $\mathcal{C}^{(r)}$ specifies the contextual background (e.g., initial conditions and environmental parameters), and $\mathcal{D}_{\mathcal{K}}^{(r)}$ stores the resulting trajectories of all agents' indicators.

Strategic Indicator Trend Analysis. For participant i in the r-th deduction round under key variable combination $\mathcal{K}$, the strategic indicators form a time-indexed vector $\mathbf{x}_{i,\mathcal{K}}^{(r)}(t) = [x_{i,\mathcal{K},1}^{(r)}(t), x_{i,\mathcal{K},2}^{(r)}(t), \ldots, x_{i,\mathcal{K},p}^{(r)}(t)]^\top \in \mathbb{R}^p$, where p is the number of tracked indicators and $t \in \{0, 1, \ldots, T\}$ represents discrete time steps within the deduction horizon T.

Analyzing the changing trends of strategic indicators of different agents over time can reveal the conflicts or mutual benefits among these indicators, as well as the competitive and cooperative trends among different agents under the same indicator.

2.5.2. Correlation Analysis

Correlation analysis quantifies two types of relationships: (i) correlations among indicators and (ii) correlations between indicators and key variables. These analyses capture both intrinsic dependencies in the indicator system and sensitivity to external conditions.

Inter-indicator correlation. To derive patterns that generalize across scenarios, we aggregate indicator values across all key-variable configurations and deduction rounds. For indicator j, we define the aggregated set:

$$\mathcal{X}_j = \{x_{i,\mathcal{K},j}^{(r)}(t) \mid \mathcal{K} \in \mathbb{K}, r \in \{1, \ldots, N\}, i \in \mathcal{P}, t \in \{0, \ldots, T\}\} \tag{14}$$

where $\mathbb{K}$ denotes the set of key-variable configurations and $\mathcal{P}$ denotes the participant set.

The Pearson correlation between indicators j and l is then computed as

$$\rho_{j,l} = \frac{\mathrm{Cov}(\mathcal{X}_j, \mathcal{X}_l)}{\sigma_{\mathcal{X}_j} \sigma_{\mathcal{X}_l}} \tag{15}$$

The coefficient is interpreted as follows: $\rho_{j,l} > 0$ indicates positive co-movement, $\rho_{j,l} < 0$ indicates negative co-movement, and $\rho_{j,l} \approx 0$ indicates weak dependence.

Indicator-variable correlation. To quantify how each participant responds to external conditions, we compute the correlation between its indicators and the key variables.

One-hot encoding. Each key-variable configuration $\mathcal{K} = \{k_1, k_2, \ldots, k_m\}$ is one-hot encoded. Let $\mathcal{V}_q$ be the value set of the q-th key variable, with cardinality $|\mathcal{V}_q| = d_q$. For a value $v \in \mathcal{V}_q$, we define the one-hot vector $\mathbf{e}_q^{(v)} \in \{0, 1\}^{d_q}$, where the entry corresponding to v is 1 and all others are 0. We then concatenate all one-hot vectors to obtain

$$\mathbf{h}_{\mathcal{K}} = [\mathbf{e}_1^{(k_1)}, \mathbf{e}_2^{(k_2)}, \ldots, \mathbf{e}_m^{(k_m)}] \in \{0, 1\}^D, \quad D = \sum_{q=1}^{m} d_q \tag{16}$$

Paired dataset. For participant i and indicator j, we construct the paired dataset:

$$\mathcal{Z}_{i,j} = \{(x^{(r)}_{i,\mathcal{K},j}(t), \mathbf{h}_{\mathcal{K}}) \mid \mathcal{K} \in \mathbb{K},\ r \in \{1,\dots,N\},\ t \in \{0,\dots,T\}\} \tag{17}$$

Correlation. The Pearson correlation between indicator values and the s-th one-hot component of key variables is computed as

$$\rho_{i,j,s} = \frac{\sum_{(x,\mathbf{h})\in\mathcal{Z}_{i,j}} (x-\mu_{i,j})(h_s-\mu_s)}{|\mathcal{Z}_{i,j}| \cdot \sigma_{i,j}\sigma_s} \tag{18}$$

where h_s denotes the s-th component of $\mathbf{h}_{\mathcal{K}}$. The coefficient $\rho_{i,j,s}$ identifies which participants are most sensitive to specific key variables.

2.5.3. Feature Importance Analysis

Feature importance analysis quantifies how strategic actions and exogenous conditions contribute to the ultimate outcome, represented by the top-level node in the TS-DAG. Because the underlying relationships can be nonlinear and interaction-driven, we use Random Forest (RF) regression as a flexible, nonparametric estimator.

Experimental protocol and randomness control. A set of independent simulation trajectories indexed by scenario-action pairs $(\mathcal{K}, r)$ is analyzed. Here, $\mathcal{K} \in \mathbb{K}$ denotes a key-variable configuration and r indexes independently generated action sequences. All analyses use trajectories sampled over a common horizon $t \in \{0,\dots,T\}$. To ensure reproducibility, the pseudo-random seed is fixed for RF training and for the permutation procedure.

Top-level objective. Let $\mathcal{N}_{\text{top}}$ denote the set of top-level nodes in the TS-DAG. For a target $j^* \in \mathcal{N}_{\text{top}}$, we denote its value for participant i under scenario $\mathcal{K}$ in run r at time t by $x^{(r)}_{i,\mathcal{K},j^*}(t)$.

Block-wise RF for action importance (lagged features). The observations form a panel time series. To avoid temporal leakage, lagged predictors are used and run-wise (block) splits are performed so that all time steps from the same run remain within a single fold.

Let $g(i,\mathcal{K},r)$ denote the run identifier. For $t \geq 1$, we define the feature vector as

$$\mathbf{f}_{i,\mathcal{K},r,t} = [\mathbf{a}_{i,\mathcal{K},r}(t-1),\ \mathbf{c}(\mathcal{K})]^{\top} \tag{19}$$

where $\mathbf{a}_{i,\mathcal{K},r}(t-1)$ stacks the lag-1 strategic action variables (e.g., *RD_Money*, *MKT_Money*, *Price*, $Profit$for each agent), and $\mathbf{c}(\mathcal{K})$ encodes the macro-environmental variables (Consumer Confidence, Subsidy Level, Battery Technology). Samples at $t = 0$ are excluded because lagged values are not available.

The feature–target dataset is constructed as

$$\mathcal{D}_{\text{RF}} = \bigcup_{i\in\mathcal{P}} \bigcup_{\mathcal{K}\in\mathbb{K}} \bigcup_{r=1}^{N} \bigcup_{t=1}^{T} \{(\mathbf{f}_{i,\mathcal{K},r,t},\ y_{i,\mathcal{K},r,t})\}, \quad y_{i,\mathcal{K},r,t} = x^{(r)}_{i,\mathcal{K},j^*}(t) \tag{20}$$

An RF regressor $\mathcal{F}_{\text{RF}}$ is trained using a fixed hyperparameter setting shared across targets. Generalization is assessed via run-wise (block) cross-validation (GroupKFold grouped by $g(i,\mathcal{K},r)$). Predictive performance is summarized using R^2 and mean absolute error (MAE).

Permutation importance and rank stability. To mitigate known biases of impurity-based importance under correlated predictors, permutation importance is computed on each held-out fold, and the resulting distributions are aggregated. For feature j, we define its importance as the expected degradation in validation score after permuting f_j:

$$\mathcal{I}_j = \mathbb{E}[\Delta\mathcal{S}_j], \quad \Delta\mathcal{S}_j = \mathcal{S}(\hat{\mathbf{y}}, \mathbf{y}) - \mathcal{S}(\hat{\mathbf{y}}^{\pi(j)}, \mathbf{y}) \tag{21}$$

where $\mathcal{S}$ denotes the validation metric (we use R^2), and $\hat{\mathbf{y}}^{\pi(j)}$ is the prediction after permuting feature j in the validation set.

3. Results

3.1. Case Description

To validate the framework, the new energy vehicle (NEV) market is used as a case study. We model a competitive landscape with three archetypal participants: a legacy automaker, a technology innovator, and a price leader. Across diverse market scenarios, their strategic performance is simulated and evaluated.

3.1.1. Evaluation Elements

Evaluation Topic. We identify the key drivers that determine market share across different types of NEV manufacturers.

Evaluation Objects. The three agents represent distinct enterprise archetypes.

- Legacy Automaker (Legacy). A traditional premium brand with strong brand equity, mature manufacturing capability, and extensive offline channels. It is comparatively slower in adopting intelligent and software-defined features.
- Technology Innovator (Tech). A technology-centric player whose competitive advantage lies in intelligent driving and user experience. It maintains an avant-garde brand image and typically adopts a direct-sales model.
- Price Leader (Price). A cost-performance-oriented player targeting mass-market households with aggressive pricing and relatively high specifications.

Evaluation Purpose. Each participant follows a distinct objective. Legacy aims to preserve leadership in the premium segment and maintain its EV market share at least at the level achieved in the internal combustion engine era. Tech aims to become the leading smart-EV brand, monetize technological advantages through sustained profitability, and cultivate a high-end user community. Price aims to rapidly expand market share and achieve economies of scale.

Evaluation Rules. We apply three principles. First, we evaluate balanced performance across multiple dimensions rather than optimizing a single metric (e.g., market share, financial health, and brand equity). Second, we emphasize sustained performance over time to distinguish short-term gains from long-term advantages. Third, we evaluate robustness across environmental conditions and rank strategies by their performance across scenarios, rather than by peak performance in a single favorable setting.

3.1.2. Key Variables

We consider three macro-level variables that shape the NEV market: the Consumer Confidence Index (CCI), the Government Subsidy Level (GSL), and the Battery Technology Bottleneck (BTB).

Consumer Confidence Index (CCI) captures market sentiment with three levels (High, Medium, Low). It indicates consumers' willingness to purchase high-end consumer goods.

Government Subsidy Level (GSL) specifies the intensity of government subsidies for high-end consumption (Strong, Medium, Weak). It directly affects vehicle prices and manufacturers' profits.

Battery Technology Bottleneck (BTB) denotes whether a critical battery technology has achieved a breakthrough (Yes/No). Such breakthroughs can fundamentally alter cost structures and performance.

3.2. Experimental Setup

3.2.1. LLM Parameters

We use `qwen3-max-20260123` [24] for all LLM-driven components, including (i) deriving the TS-DAG structure, (ii) instantiating node parameters, and (iii) evaluating strategic actions online during deduction. We set the temperature to 0 to ensure deterministic outputs, and the maximum context length to 256k tokens.

To handle malformed outputs, we implement an automatic repair policy that focuses strictly on syntactic validation (i.e., JSON schema compliance). The repair loop follows a feedback-based refinement logic: upon detecting a formatting error, the system constructs a new prompt that contains (1) the original task, (2) the specific error message, and (3) the required output schema. This composite input is re-submitted to the LLM to guide it toward producing a valid structure. We allow up to three retries per query.

3.2.2. Random Forest Parameterization

For feature-importance analysis, we train a Random Forest regressor with `n_estimators`=500, `max_depth`=12, `min_samples_leaf`=4, and `max_features=sqrt`. We set `random_state`=42 to control randomness. We compute permutation importance with `n_repeats`=5.

We assess generalization via run-wise (block) 5-fold GroupKFold cross-validation. The grouping variable is the run identifier (`run_id`).

3.2.3. Strategic Deduction Settings

Experiments are designed to cover the macro-environmental space induced by the key variables. Consumer Confidence has 3 levels, Government Subsidy has 3 levels, and Battery Technology has 2 levels. Together, they define $3 \times 3 \times 2 = 18$ scenarios.

For each scenario, 10 independent strategic action sequences are generated. This yields 180 simulation runs in total. These runs form the dataset used for correlation and feature-importance analyses.

3.2.4. Comparison Algorithms

To benchmark the indicator system derived in Section 2.3, we compare TS-DAG against two baselines. The first baseline is an AHP-based direct evaluation method (AHP-based). The second baseline generates a TS-DAG via direct prompting with $\tau = 0$ (prompt-based). Implementation details are provided in the Supplementary Materials.

3.2.5. Comparison Indicators

We evaluate the proposed method and the baselines using three metrics that jointly quantify competitive separation and temporal stability.

- Discriminability ($\mathcal{D}$): We quantify how well a method separates competing agents. We compute $\mathcal{D}$ as the temporal mean of the across-agent standard deviation of market shares. Larger $\mathcal{D}$ indicates clearer separation and avoids degenerate uniform outcomes.
- Volatility ($\mathcal{V}$): We quantify short-term instability in the simulated trajectories. We define $\mathcal{V}$ as the temporal mean of the absolute market-share differences between consecutive time steps. Smaller $\mathcal{V}$ implies smoother trajectories and supports reliable long-horizon analysis.
- Stable Discriminability Score (SDS): We combine separation and stability via $\text{SDS} = \mathcal{D}/(\mathcal{V} + \epsilon)$. This metric rewards methods that preserve separation (high $\mathcal{D}$) while suppressing jitter (low $\mathcal{V}$). Larger SDS indicates better overall performance.

3.2.6. Parameter Settings for Sensitivity Analysis

To evaluate robustness with respect to action-impact scaling, we conduct a sensitivity analysis on the mapping M in Equation (12). We consider three configurations: Baseline, Conservative, and Aggressive.

Baseline is the default setting used in the main experiments. Conservative compresses the magnitude of action impacts toward the neutral factor (1.0). Aggressive expands the impact range to stress-test the deduction dynamics under stronger perturbations. Table 1 reports the numerical values used for each impact level.

Table 1. Parameter settings for the mapping M under different sensitivity scenarios.

Setting	Large Positive	Small Positive	Neutral	Small Negative	Large Negative
Baseline	1.4	1.1	1.0	0.9	0.6
Conservative	1.2	1.05	1.0	0.95	0.8
Aggressive	1.6	1.2	1.0	0.8	0.4

3.3. Automated Generation of the Evaluation Indicator System

Using the LLM-driven TS-DAG structure-generation procedure, the framework derives an evaluation indicator system for the NEV case. The generated specification explicitly defines node types, initializes node states, and instantiates symbolic transfer functions that govern temporal market evolution.

The resulting TS-DAG comprises three node categories: Exogenous Inputs (decision variables), State Variables (accumulated assets with temporal memory), and Computational Nodes (instantaneous derivations). In this formulation, $i \in \{\text{Legacy}, \text{Tech}, \text{Price}\}$ indexes the agent and t denotes the discrete time step.

To parameterize the simulator, we set a unified constant configuration. Specifically, we set the monthly Total Market Demand (TMD) to 50,000 units, the reference Consumer Confidence Index (CCI) to 1.0, and the base Net Promoter Score (NPS_{base}) to 150. We encode temporal persistence using decay factors ($k_{decay} = 0.98, k_{br} = 0.95, k_{nps} = 0.97$) and model efficiencies using coefficients ($RDF = 1, MKT = 0.5$). We compute the Total Attraction Index with fixed component weights: $W_p = 0.4$ (Product), $W_c = 0.3$ (Price), and $W_b = 0.3$ (Brand). The full graph specification is provided in Table 2.

Table 2. The automatically generated indicator system for NEV market simulation.

Node Name	Description	Type	Initial Values (t = 0)	Generated Transfer Function/Formula
Exogenous Decision Inputs (Per Step)				
RD_Money_i	R&D Investment	Input	L: 50, T: 30, P: 15	Exogenous (Action)
MKT_Money_i	Marketing Investment	Input	L: 45, T: 30, P: 15	Exogenous (Action)
$Price_i$	Average Selling Price	Input	L: 40, T: 35, P: 25	Exogenous (Action)
$Profit_i$	Profit Per Unit	Input	L: 8, T: 6, P: 4	Exogenous (Action)
Core Competitiveness Metrics (State Variables)				
PPS_i	Product Power Score	State	L: 120, T: 130, P: 100	$PPS_i[t-1] \cdot k_{decay} + \ln(1 + \frac{RD\_Money_i}{RDF})$
BPS_i	Brand Power Score	State	L: 150, T: 110, P: 80	$BPS_i[t-1] \cdot k_{br} + \ln(1 + \frac{MKT\_Money_i}{MKT}) + (NPS_i - NPS_{base}) \cdot 0.1$
NPS_i	Net Promoter Score	State	L: 160, T: 180, P: 150	$NPS_i[t-1] \cdot k_{nps} + (PPS_i - \overline{PPS}_{total}) \cdot 2$
Intermediate & Market Logic				
$PPS2_i$	Price Power Score	Comp	-	$PPS_i[t] / Price_i[t]$
$Total_i$	Total Attraction Index	Comp	-	$PPS_i \cdot W_p + PPS2_i \cdot W_c + BPS_i \cdot W_b$
$MKTS_i$	Market Share	Comp	-	$Total_i[t] / \sum_j Total_j[t]$
$Sales_i$	Sales Volume	Comp	-	$TMD \cdot MKTS_i[t]$
$Online_i$	Online Buzz Index	Comp	-	$10 \ln(1 + MKT\_Money_i) + 5 \ln(1 + Sales_i)$
Resource Constraints (Budget States)				
RDB_i	RD Fund Balance	State	L: 500, T: 300, P: 150	$RDB_i[t-1] - RD\_Money_i + \frac{Profit_i \cdot Sales_i}{10{,}000}$
$MKTB_i$	MKT Budget Balance	State	L: 300, T: 200, P: 100	$MKTB_i[t-1] - MKT\_Money_i$

This representation links macro conditions to micro decisions within a single causal–temporal model. The key variables (e.g., CCI, GSL, BTB) act as scenario-level modulators that are exogenous to individual agents and shared across participants. In contrast, the input nodes encode agent-specific decision levers (e.g., investment, pricing, and profit-related choices). During deduction, these inputs propagate through meso-level state variables (e.g., brand power and NPS) via TS-DAG state-update equations. The propagation ultimately determines top-level outcomes such as market share. Overall, the TS-DAG provides an interpretable mechanism in which macro conditions define the operating regime, while agent-level decisions drive endogenous competitive dynamics through structured intermediate states.

The automatically generated TS-DAG captures several nuanced mechanisms of business operations (Figure 5). First, the transfer functions for PPS and BPS combine a decay factor ($k < 1$) with a logarithmic investment return. This design enforces continuous spending to sustain competitiveness, which operationalizes the "Red Queen" effect.

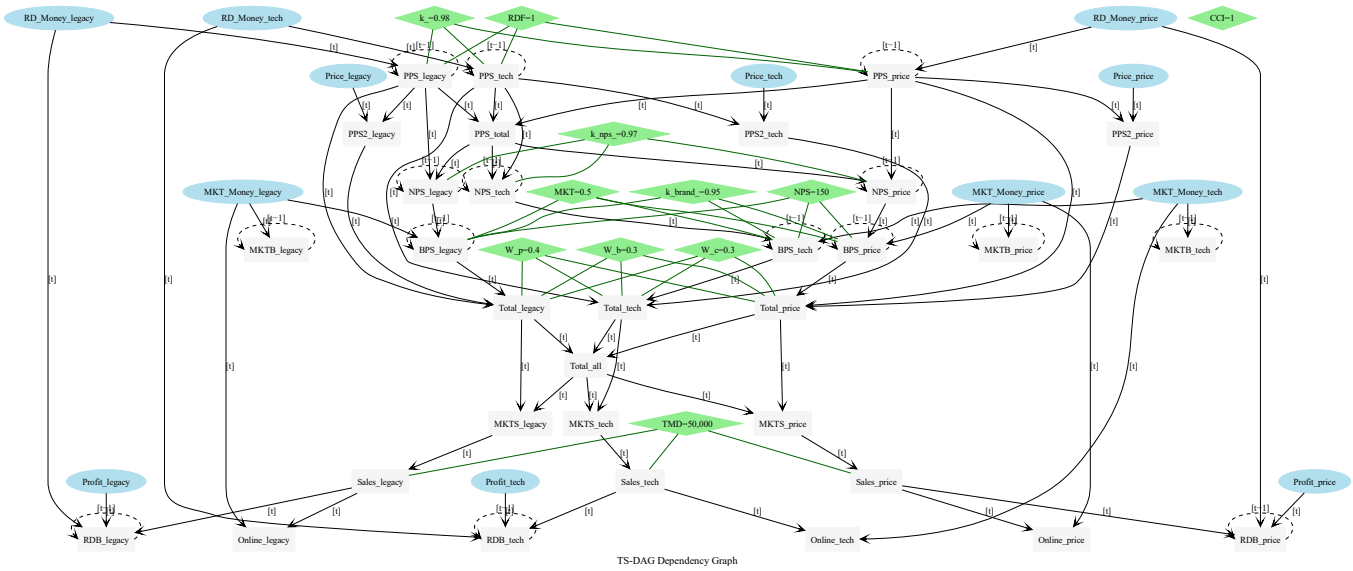


Figure 5. Indicator system diagram structure of Table 2.

Second, the framework derives $PPS2$ (Price Power Score) as the ratio $PPS/Price$. This formulation induces an explicit trade-off: increasing $Price$ improves unit profit ($Profit$) but reduces $PPS2$. A lower $PPS2$ decreases the Total Attraction Index and can therefore reduce market share ($MKTS$).

Third, the TS-DAG instantiates a feedback loop between product quality and brand perception. Specifically, NPS increases with relative product superiority, $(PPS_i - \overline{PPS}_{total})$. A higher NPS then increases brand power (BPS), capturing the long-term effect of word-of-mouth. This mechanism is particularly important for the Tech agent.

Finally, the resource constraints (RDB, $MKTB$) impose a survival boundary. The RDB update rule includes a replenishment term derived from $Sales$ and $Profit$. This closes the loop between commercial success and future innovation capacity.

3.4. Analysis of Single-Run Evolutionary Dynamics

To examine whether the generated strategies yield coherent temporal dynamics, we analyze key indicators from a representative single simulation run. Figure 6 plots the resulting trajectories for the three interacting agents.

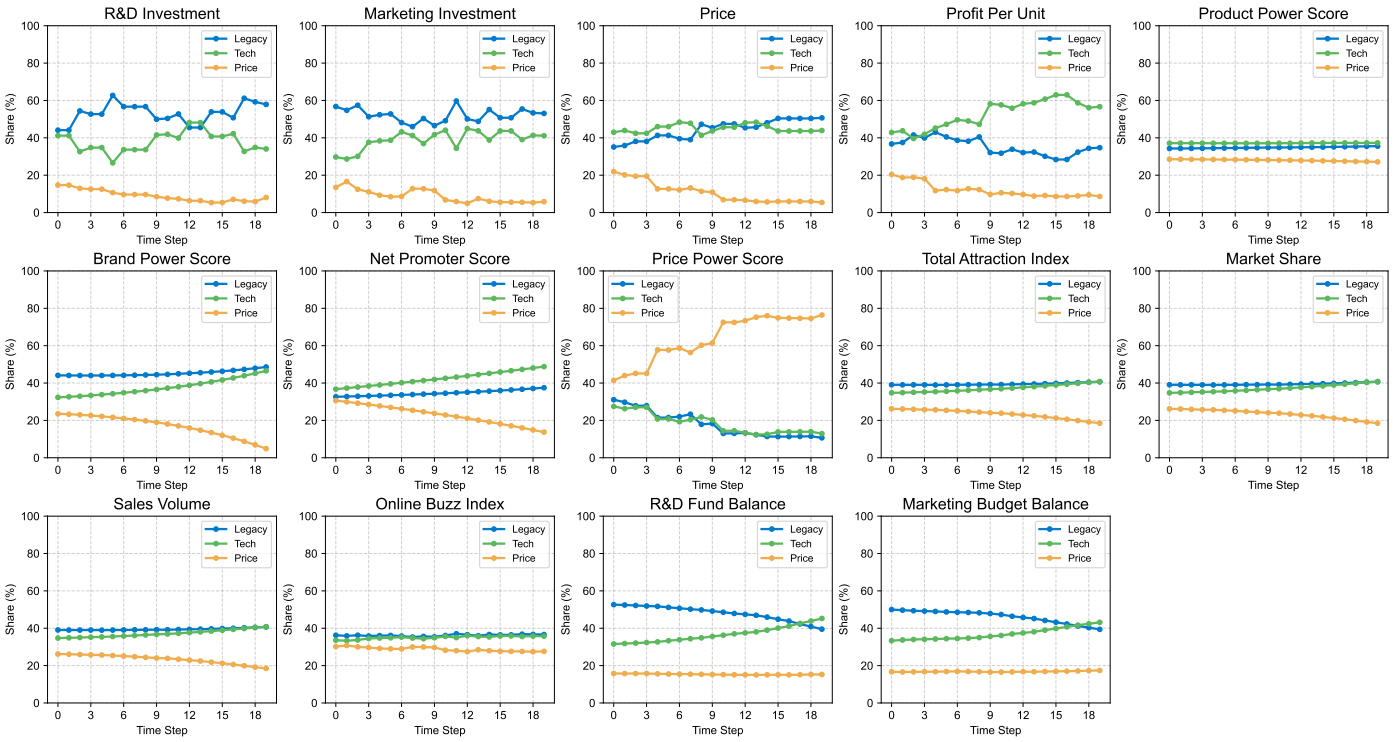


Figure 6. An example of the indicator trajectories in a single simulation run.

The market-share trajectory ($MKTS$) shows a pronounced shift in competitive leadership. At initialization, Legacy leads the market due to its higher initial brand power and accumulated resources. Over time, Tech exhibits sustained growth and eventually overtakes Legacy, resulting in a clear crossover.

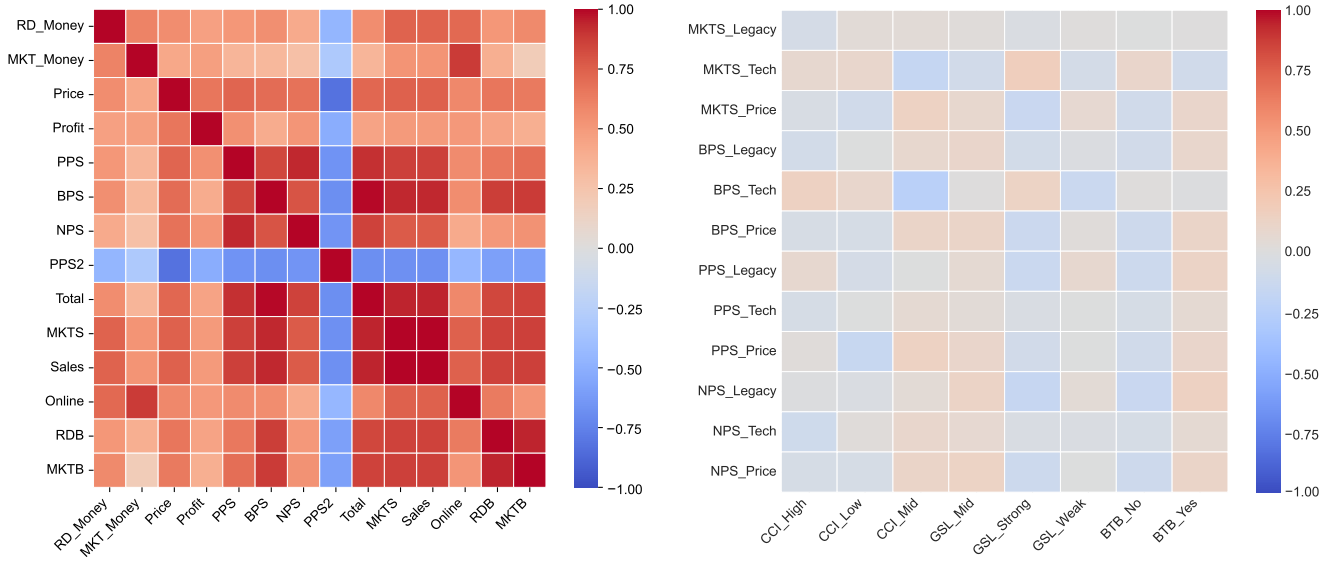
The underlying drivers are consistent with the TS-DAG mechanisms. Tech achieves higher $Profit$ and NPS. Its larger R&D investment improves product quality (PPS), which increases NPS. In turn, higher NPS feeds back to strengthen brand power (BPS). Moreover, higher unit profitability replenishes the R&D budget (RDB), enabling sustained reinvestment and preventing competitiveness from decaying.

In contrast, Price highlights the limitations of a purely cost-driven strategy in this setting. It maintains the highest Price Power Score ($PPS2$), i.e., the strongest cost-performance ratio. However, its market share declines over time. This decline is primarily driven by thin margins, which limit RDB accumulation, and by a lower brand ceiling. Overall, the simulation suggests that cost-performance alone ($PPS2$) is insufficient to secure market dominance when competitors jointly optimize profitability, reputation (NPS), and product innovation.

3.5. Statistical Correlation Analysis of Multi-Run Simulations

To extract insights that generalize beyond a single run, we perform a correlation analysis across all simulation runs. We quantify statistical dependencies (i) among generated indicators and (ii) between indicators and macro-environmental key variables. Figure 7 reports the resulting Pearson correlation heatmaps.

Figure 7a reports inter-indicator correlations aggregated across all agents. By pooling agents, we highlight structural market mechanisms rather than agent-specific artifacts. We observe a pronounced negative correlation between the Price Power Score ($PPS2$) and most other performance indicators. This pattern suggests that emphasizing cost-performance (low price relative to product power) can trade off against profitability and brand accumulation, which slows long-term asset growth.



(**a**) (**b**)

Figure 7. Correlation analysis of evaluation indicators and environmental variables. (**a**) shows inter-indicator correlations aggregated across all agents. (**b**) shows the sensitivity of core indicators to macro-environmental key variables.

Excluding direct computational links, Brand Power Score (BPS) shows the strongest positive correlation with Market Share ($MKTS$). Net Promoter Score (NPS) is also strongly aligned with market dominance. These results are consistent with the single-run dynamics (Figure 6): Tech overtakes Legacy primarily through higher NPS and sustained brand accumulation (BPS), rather than through price competition.

Figure 7b relates macro-environmental key variables (Consumer Confidence, Subsidy Level, and Battery Technology) to core indicators. Although these correlations are weaker than intra-system correlations, they still reveal interpretable policy and technology effects. For example, strong subsidies correlate positively with $MKTS_{tech}$ but negatively with $MKTS_{price}$. This indicates that financial incentives disproportionately benefit premium, innovation-oriented players.

We also observe that moderate Consumer Confidence and Subsidy levels correlate positively with core indicators across all agents. Finally, battery-technology breakthroughs correlate positively with most global performance indicators, but negatively with $MKTS_{tech}$. This suggests that an industry-wide technology leap can reduce the innovation leader's relative advantage and narrow the competitive gap.

3.6. Feature Importance Analysis via Random Forest Regression

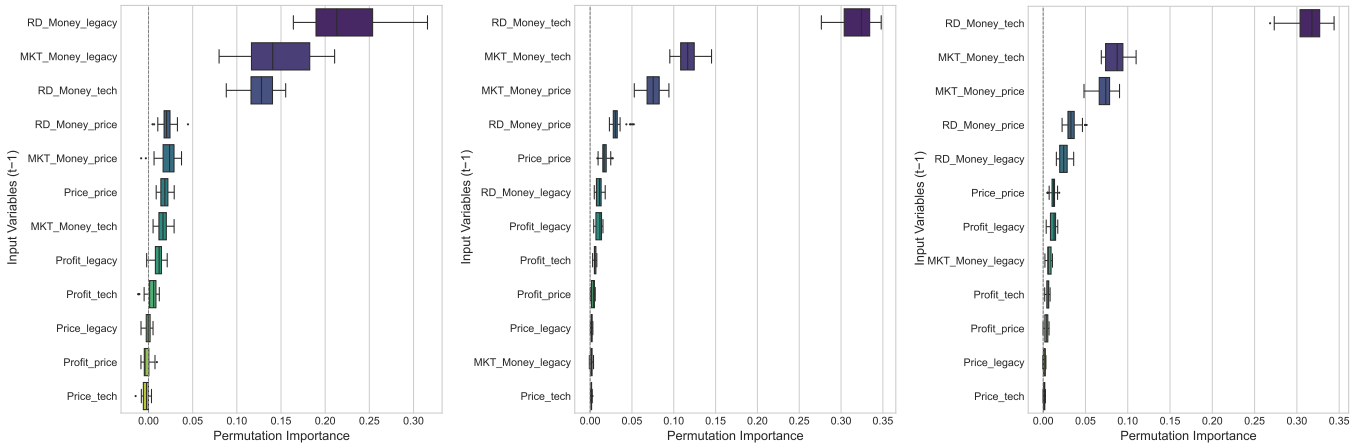
Figure 8 summarizes permutation-importance estimates obtained under run-wise (block) cross-validation. Across all three agents, lagged R&D investment (RD_Money) ranks as the most important predictor of subsequent market share. This indicates that sustained innovation input is the primary lever for competitive advantage in this market.

Lagged marketing investment (MKT_Money) is consistently the second most important feature. This highlights the role of demand shaping and brand exposure in converting technological capability into realized market share.

The stable ordering across $MKTS_{legacy}$, $MKTS_{tech}$, and $MKTS_{price}$ suggests that the dominance of R&D over marketing is structural, rather than an artifact of a specific participant.

Table 3 reports cross-validated predictive performance. The positive R^2 values and low MAE across all three targets indicate that the learned mapping from lagged actions to

subsequent market share is statistically meaningful. This supports using permutation-based importance to interpret the dominant strategic drivers.



(**a**) (**b**) (**c**)

Figure 8. Permutation-based feature importance for market-share prediction under a Random Forest model. Panels (**a**–**c**) correspond to the Legacy, Tech, and Price agents, respectively, and show cross-validated importance distributions for lagged strategic actions. Across all three agents, *RD_Money* is consistently the most important driver of *MKTS*, followed by *MKT_Money*.

Table 3. Cross-validated predictive performance of the Random Forest models for market share prediction.

	R^2	MAE
MKTS_legacy	5.33×10^{-1}	1.52×10^{-3}
MKTS_tech	8.86×10^{-1}	4.53×10^{-3}
MKTS_price	8.64×10^{-1}	5.54×10^{-3}

3.7. Comparison Algorithm Analysis

To assess the quality of the generated indicator system, we benchmark TS-DAG against two baselines. The first baseline is an AHP-based direct evaluation method (AHP-based). The second baseline generates a TS-DAG via direct prompting (prompt-based).

Table 4 summarizes the results. TS-DAG achieves the lowest volatility, $\mathcal{V} = 2.49 \times 10^{-3}$ (var 6.11×10^{-8}), while maintaining competitive separation, $\mathcal{D} = 8.52 \times 10^{-2}$ (var 1.22×10^{-5}). As a result, TS-DAG attains the highest stable discriminability score, SDS $= 3.44 \times 10^{1}$ (var 6.35). It substantially outperforms the AHP-based method (SDS $= 5.77$; var 7.25) and the prompt-based method (SDS $= 4.47$; var 3.67). The definitions of $\mathcal{V}$, $\mathcal{D}$, and SDS are given in the preceding subsection.

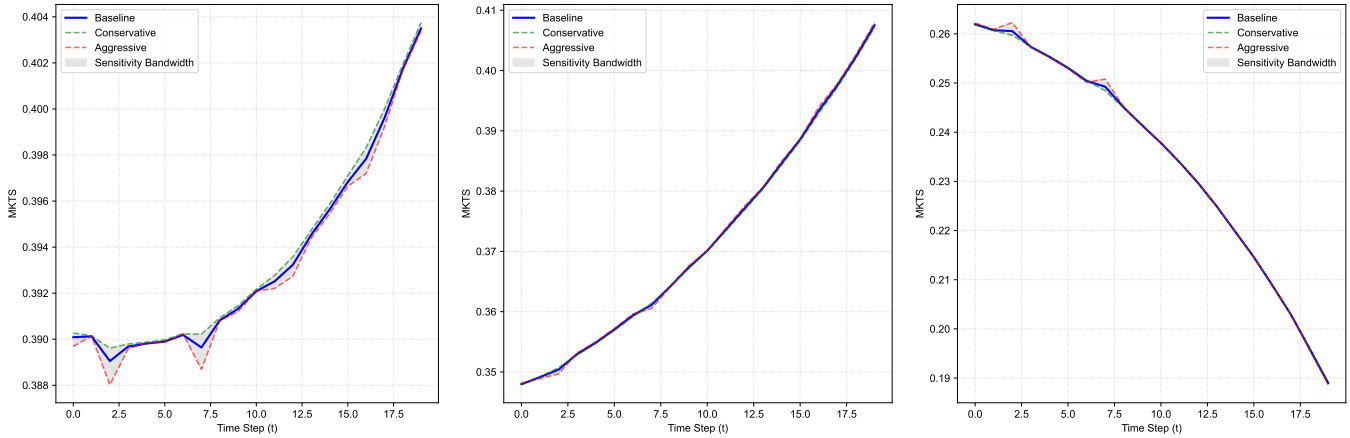
Table 4. Comparative performance of TS-DAG and baseline methods in terms of volatility, discriminability, and stable discriminability score.

	V		D		SDS	
	Mean	**Var**	**Mean**	**Var**	**Mean**	**Var**
TS-DAG	2.49×10^{-3}	6.11×10^{-8}	8.52×10^{-2}	1.22×10^{-5}	$3.44 \times 10^{+1}$	6.35
AHP-based	4.20×10^{-2}	3.35×10^{-4}	2.20×10^{-1}	7.65×10^{-3}	5.77	7.25
prompt-based	4.00×10^{-2}	4.39×10^{-5}	1.73×10^{-1}	3.23×10^{-3}	4.47	3.67

Table 4 indicates that TS-DAG yields substantially lower volatility while preserving competitive separation. This combination leads to the highest SDS among the compared methods. Overall, the results suggest that TS-DAG produces trajectories that are both stable and discriminative, which is critical for reliable strategic evaluation.

3.8. Algorithm Sensitivity Analysis

Figure 9 compares three configurations of the impact mapping M in Equation (12): Baseline, Conservative, and Aggressive. Aggressive amplifies action-induced perturbations. This setting disproportionately benefits the Price agent by strengthening short-term market acquisition under cost-competitive positioning. In contrast, Conservative compresses action-to-indicator effects. This stabilizes trajectories and favors Legacy by preserving its accumulated brand and resource advantages. Overall, Baseline lies between these extremes and yields the most stable market-share evolution.



(**a**) (**b**) (**c**)

Figure 9. Sensitivity analysis under different impact-weight configurations. Panels (**a**–**c**) report the resulting market share trajectories for the Legacy, Tech, and Price agents, respectively.

To further quantify this stability, we examine ranking consistency under parameter perturbations. Across 180 runs with different parameter settings, the MKTS ranking changes in only 5 runs (97.2% unchanged), indicating that the qualitative conclusions are robust to the impact-weight configuration. We therefore adopt Baseline as the default setting in the main experiments.

4. Conclusions

This paper introduces an LLM-driven strategic evaluation framework that couples large language models with a time-series directed acyclic graph (TS-DAG). The TS-DAG enforces an explicit causal structure at each time step while capturing time-lagged propagation, and the LLM helps derive the indicator system, instantiate model components, and quantify action impacts during deduction. Experiments in the new energy vehicle market demonstrate that the framework yields coherent competitive dynamics and supports interpretable, scenario-dependent analysis. However, the framework inherits LLM risks (hallucination, prompt sensitivity, and bias), which can affect indicator definitions and action-to-impact mappings. External validity is constrained by the stylized simulator and the finite scenario set. In addition, performance may degrade under distribution shift, and uncertainty is not explicitly quantified.

Future work will extend the framework in three directions. First, evaluation will be coupled with explicit decision and optimization modules so that strategies can be selected, refined, and validated in a closed loop. Second, robustness and external validity will be improved by calibrating TS-DAG components with real-world evidence, expanding scenario coverage, and stress-testing under distribution shift with uncertainty-aware analysis. Third, we will develop domain-specialized models and safer prompting/guardrail

mechanisms to reduce hallucination, mitigate bias, and improve the reliability of action-to-impact mappings.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/app16042007/s1, Section S1: Prompts for LLM-Driven Agent, Section S1.1: LLM-Driven TS-DAG Structure Generation-Generate Input Node, Section S1.2: LLM-Driven TS-DAG Structure Generation-Generate Formula, Section S1.3: Data-Driven Parameter Identification-Initialize Input Parameter, Section S1.4: Data-Driven Parameter Identification-Evaluate Feedback, Section S1.5: Data-Driven Parameter Identification-Adjust Parameters, Section S1.6: Strategic Action Quantification and Propagation, Section S2: Comparison Algorithms Description, Section S2.1: AHP-based Algorithm, Section S2.2: Prompt-based Algorithm.

Author Contributions: Conceptualization, M.Z. and X.Z.; methodology, M.Z. and Y.Y.; software, M.Z.; formal analysis, M.Z.; writing—original draft preparation, M.Z.; writing—review and editing, X.Z. and Y.Y.; visualization, M.Z.; supervision, G.Y. and L.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The code and raw data will be made public if necessary.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Punt, A.E.; Butterworth, D.S.; de Moor, C.L.; De Oliveira, J.A.; Haddon, M. Management strategy evaluation: Best practices. *Fish Fish.* **2016**, *17*, 303–334. [CrossRef]
2. Xin, Z.; Kui, G.; Xie, Z. Modes of Research-oriented Strategy Deduction and Its Intelligent Decision-making Assistance Methods. In *2024 36th Chinese Control and Decision Conference (CCDC)*; IEEE: Xi'an, China, 2024; pp. 3882–3885.
3. Puyt, R.W.; Lie, F.B.; Madsen, D.Ø. From SOFT approach to SWOT analysis, a historical reconstruction. *J. Manag. Hist.* **2025**, *31*, 333–373. [CrossRef]
4. Benzaghta, M.A.; Elwalda, A.; Mousa, M.M.; Erkan, I.; Rahman, M. SWOT analysis applications: An integrative literature review. *J. Glob. Bus. Insights* **2021**, *6*, 54–72. [CrossRef]
5. Yüksel, I. Developing a multi-criteria decision making model for PESTEL analysis. *Int. J. Bus. Manag.* **2012**, *7*, 52. [CrossRef]
6. Azis, S.S.A.; Salleh, K.M.; Ab Rahman, N.H.; Aziz, N.; Abdullah, S. Asset Management using PESTLE-SWOT Analysis: Strategic Aspects for Integrated Dam Catchment Management. *Plan. Malays.* **2025**, *23*, 790–804. [CrossRef]
7. Chehimi, M.; Naro, G. Balanced Scorecards and sustainability Balanced Scorecards for corporate social responsibility strategic alignment: A systematic literature review. *J. Environ. Manag.* **2024**, *367*, 122000. [CrossRef]
8. Mahdiraji, H.A.; Kazimieras Zavadskas, E.; Kazeminia, A.; Abbasi Kamardi, A. Marketing strategies evaluation based on big data analysis: A CLUSTERING-MCDM approach. *Econ. Res. Ekon. Istraživanja* **2019**, *32*, 2882–2892. [CrossRef]
9. Darko, A.; Chan, A.P.C.; Ameyaw, E.E.; Owusu, E.K.; Pärn, E.; Edwards, D.J. Review of application of analytic hierarchy process (AHP) in construction. *Int. J. Constr. Manag.* **2019**, *19*, 436–452. [CrossRef]
10. Behzadian, M.; Otaghsara, S.K.; Yazdani, M.; Ignatius, J. A state-of the-art survey of TOPSIS applications. *Expert Syst. Appl.* **2012**, *39*, 13051–13069. [CrossRef]
11. Govindan, K.; Jepsen, M.B. ELECTRE: A comprehensive literature review on methodologies and applications. *Eur. J. Oper. Res.* **2016**, *250*, 1–29. [CrossRef]
12. Chou, S.Y.; Chang, Y.H.; Shen, C.Y. A fuzzy simple additive weighting system under group decision-making for facility location selection with objective/subjective attributes. *Eur. J. Oper. Res.* **2008**, *189*, 132–145. [CrossRef]
13. Edwards, W.; Barron, F. SMARTS and SMARTER: Improved Simple Methods for Multiattribute Utility Measurement. *Organ. Behav. Hum. Decis. Processes* **1994**, *60*, 306–325. [CrossRef]
14. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.u.; Polosukhin, I. Attention is All you Need. In *Advances in Neural Information Processing Systems*; Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: New York, NY, USA, 2017; Volume 30.

15. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv* **2019**, arXiv:1907.11692.
16. OpenAI; Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F.L.; Almeida, D.; Altenschmidt, J.; Altman, S.; et al. GPT-4 Technical Report. *arXiv* **2024**, arXiv:2303.08774.
17. Puyt, R.W.; Madsen, D.Ø. Evaluating ChatGPT-4's historical accuracy: A case study on the origins of SWOT analysis. *Front. Artif. Intell.* **2024**, *7*, 1402047. [CrossRef]
18. Brath, R.; Bradley, A.; Jonker, D. Strategic management analysis: From data to strategy diagram by LLM. *arXiv* **2024**, arXiv:2409.06643. [CrossRef]
19. Kharlashkin, L.; Morooka, E.; Tereshchenko, Y.; Hämäläinen, M. ORACLE: Time-Dependent Recursive Summary Graphs for Foresight on News Data Using LLMs. *arXiv* **2025**, arXiv:2512.15397. [CrossRef]
20. Svoboda, I.; Lande, D. Enhancing Multi-Criteria Decision Analysis with AI: Integrating Analytic Hierarchy Process and GPT-4 for Automated Decision Support. *arXiv* **2024**, arXiv:2402.07404. [CrossRef]
21. Wang, H.; Zhang, F.; Mu, C. One for All: A General Framework of LLMs-based Multi-Criteria Decision Making on Human Expert Level. *arXiv* **2025**, arXiv:2502.15778.
22. Phoka, T.; Wathahong, T.; Boriwan, P. An LLM–MCDM Framework with Lin's Concordance Correlation Coefficient for Recommendation Systems: A Case Study in Food Preference. *Appl. Sci.* **2026**, *16*, 117. [CrossRef]
23. Kahn, A.B. Topological sorting of large networks. *Commun. ACM* **1962**, *5*, 558–562. [CrossRef]
24. Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; et al. Qwen3 Technical Report. *arXiv* **2025**, arXiv:2505.09388. [CrossRef]