AgentScen 2025

The 2nd Workshop on Agent AI for Scenario Planning (AgentScen) in conjunction with IJCAI 2025

Proceedings of the Workshop

August 16, 2024

Montreal, Canada

Organizations

Organizing Committee

Chung-Chi Chen, AIST, Japan
Tatsuya Ishigaki, AIST, Japan
Sophia Ananiadou, University of Manchester, UK
Hiroya Takamura, AIST, Japan

Shared Task Organizers

Wataru Hirota, Stockmark Inc., Japan
Chung-Chi Chen, AIRC, AIST, Japan
Tatsuya Ishigaki, AIRC, AIST, Japan
Tomoko Ohkuma, Asahi Kasei Corporation, Japan
Tomoki Taniguchi, Asahi Kasei Corporation, Japan

Acknowledgements

We thank the annotators for their valuable contributions to the data annotation process.

Asahikasei TENAC R&D Dept.

Yoshitaka Shigematsu
Motohiro Fukai
Tomohito Asakawa
Shinya Akebi

Stockmark Inc.

Daisuke Nakagawa
Jiro Kashiwagi
Kumiko Boshita
Shotaro Kusaura
Tomoaki Okamoto
Wataru Inoue
Yutaka Mori

Table of Contents

Multi-Scale Temporal Scenario Planning for Financial Networks: A GNN Approach to Stress Testing

Xinyan Zhang[1] **and Xiaobing Feng**[1] **and Xiujuan Xu**[2,*] **and Ling Feng**[3] **and Jinghua Lian**[4]
[1,2] Dalian University of Technology
[1,3] Shanghai University Of International Business Economics
[4] Guangzhou University
[1]dahuzidahu@gmail.com,[2]xjxu@dlut.edu.cn [1] fxb@suibe.edu.cn,
[3]13120925823@163.com,[4]lianjinghua@e.gzhu.edu.cn

Abstract

Financial networks have grown increasingly complex and interconnected, creating urgent challenges for systemic risk management. We propose a robust multi-scenario stress testing framework based on graph neural networks that enables large-scale anomaly detection and systematic risk assessment across pre- and post-pandemic financial landscapes. Our approach integrates several technical innovations: efficient sparse matrix computation for graphs with over 81,434 nodes, dynamic class imbalance handling that improves recall by nearly 17 times, and a comprehensive scenario-based evaluation protocol examining baseline performance, feature noise resilience, structural vulnerability, and susceptibility to information shocks. Experiments on real financial data comparing the 2019 (pre-pandemic) and 2022 (post-pandemic) periods reveal a significant shift in risk characteristics – post-pandemic networks demonstrate heightened vulnerability to structural changes (-9.4% AUC-PR) and information propagation (-3.9% AUC-PR), indicating that risk sources have evolved from data quality concerns to network connectivity and information flow dynamics. Our framework provides regulators and financial institutions with practical tools to identify emergent risks and enhance system resilience against future structural and information-based shocks.

Keywords: GNN, multi-scale scenario planning, fake news detection in finance

1 Introduction

The increasing complexity and interconnectedness of global financial systems have made stress testing a crucial tool for identifying systemic risks and supporting macroprudential policy(Pritsker, 2011; Federal Reserve System, 2024; European Banking Authority, 2016, 2024; Bank of England, 2022). Traditional stress testing frameworks, however, often rely on macroeconomic variables and static scenario design, limiting their ability to address heterogeneous, technology-driven, or structural risks. Recent events such as the COVID-19 pandemic have further highlighted the need for dynamic, multi-scenario approaches that can capture evolving risk transmission paths and the impact of information shocks(Lim, 2016; Bank of Japan, 2024).

To address these challenges, we develop a graph neural network (GNN)-based framework for large-scale financial anomaly detection and multi-scenario stress testing. Our method features: (1) scalable processing of financial graphs with over 80,000 nodes and 350,000 records via sparse matrix and memory optimization; (2) a dual-weighting mechanism combining dynamic class weights and improved Focal Loss to tackle severe class imbalance; (3) an adaptive threshold selection algorithm to optimize precision-recall trade-offs; and (4) a scenario design covering baseline, feature noise, graph structure change, and fake news propagation, enabling systematic evaluation of network vulnerability and resilience.

We fuse multi-source data (structured financials, sentiment, regulatory records) and employ PCA for efficient feature engineering, retaining over 91% of information. Comparative experiments on pre- and post-pandemic data (2019 vs. 2022) show that post-pandemic financial networks exhibit much higher sensitivity to structural and information shocks (AUC-PR drops of -9.4% and -3.9%, respectively), indicating a shift in risk sources from data quality to network connectivity and information flow. These findings suggest the need for enhanced monitoring of network structure and information propagation in financial regulation.

Our contributions are threefold: (1) a scalable GNN-based anomaly detection and stress testing framework for large financial networks; (2) methodological innovations in class imbalance handling and scenario-based evaluation; (3) empirical evidence of evolving risk characteristics in financial systems under systemic shocks. Model capacity's impact on performance is summarized in Appendix Table A.4. The proposed approach offers both technical solutions and policy insights for improving financial system resilience.

2 Related Work

In recent years, literature on stress testing has evolved toward integrating agent-based modeling (ABM) and graph neural networks (GNNs) to address complex systemic risks. For example, Samimi et al. (2024) demonstrated how Agent-Based Modeling (ABM) can simulate autonomous agent behaviors and interactions to enhance system safety and risk management, while Bernárdez et al. (2023) proposed MAGNNETO, a distributed GNN-multi-agent framework for traffic engineering optimization. These studies highlight the potential of hybrid models that combine agent autonomy with graph-based structure learning.

2.1 Classical Theory and Basic Definitions

Pritsker (2011) is an important representative figure in stress testing theory construction. His proposed "Enhanced Stress Testing" framework emphasizes a risk exposure-driven system modeling approach, distinct from traditional linear models that rely solely on macroeconomic variable shocks. His research particularly proposed the concept of "Trust Set," which involves constructing a set of reasonable but non-unique scenarios to conduct multi-dimensional shock resistance assessments of institutions under highly uncertain environments, enhancing the robustness of testing.

2.2 U.S. Stress Testing System Experience

The Federal Reserve System has established a comprehensive modeling framework encompassing modules for loan and trading losses, net income, and capital adequacy. This approach emphasizes scenario design based on historically extreme but plausible events, data-driven modeling, and institutional independence, while employing unified tools to assess multi-institutional responses and balancing regulatory transparency with market stability (Federal Reserve System, 2024). Furthermore, regulatory provisions “Rules and Regulations (6651–6664)” (Federal Register, 2023) highlight public participation, model updates, and risk evolution, underscoring the normative and progressive features of the U.S. system.

2.3 Comparison of EU and UK Approaches

The European Banking Authority's "2025 EU-wide Stress Test Methodological Note" advocates incorporating structural shocks, such as climate change, into stress testing and emphasizes consistent cross-national assessment (European Banking Authority, 2024). The earlier "2016 FAQ document" established procedures for identifying capital adequacy, risk concentration, and contagion paths, laying the foundation for institutionalized stress testing (European Banking Authority, 2016).

The Bank of England, in its "2022 Annual Cyclical Scenario (ACS) Elements Description," highlights the evaluation of structural and non-linear risks through multi-path carbon policy simulations and adaptive balance sheet assessments, exemplifying climate stress testing practices (Bank of England, 2022).

2.4 Emerging Explorations in Asia

The Monetary Authority of Singapore has expanded stress testing to include technological risks such as AI model errors and cyber attacks, demonstrating forward-looking regulatory awareness (Lim, 2016). The Bank of Japan's 2024 "Financial System Report" analyzes the long-term effects of population aging on the financial system, highlighting structural risks to bank capital adequacy and adaptation strategies for financial institutions (Bank of Japan, 2024).

2.5 Other Methodological Extensions

At the investment management level, Ruban and Melas (2010) proposed using multi-factor risk models to conduct stress assessments of investment portfolios, emphasizing risk factor linkage mechanisms and the adaptability of micro-asset allocation, which is an important complementary path for micro-financial stress testing (Ruban and Melas, 2010).

Ok and Eniola (2025) proposed a deep learning-based scenario reasoning method in their research, using unstructured data to enhance the model's sensitivity and response capability to emerging risks, demonstrating the potential of AI tools in cross-variable modeling and data dimension adaptation (Ok and Eniola, 2025).

In the field of graph neural networks and financial risk detection, Weber et al. (2019)first applied GCN to financial network analysis, demonstrating the effectiveness of graph structure information in capturing financial anomalies, although their research was limited to small-scale data. Thilagavathi et al. (2024) proposed a framework combining graph neural networks and anomaly detection techniques for financial fraud detection, achieving a 95% detection rate on highly imbalanced credit card fraud datasets, but mainly focused on credit card transactions without extension to more complex financial network structures. Balmaseda et al. (2023) explored the application of deep graph learning in predicting systemic risks in financial systems, emphasizing the importance of machine learning in analyzing large financial networks, but traditional techniques still have limitations in handling complex relationships. While these studies have advanced the application of graph neural networks in the financial domain, they still showed obvious deficiencies in processing large-scale data, solving extreme class imbalance, and constructing multi-scenario stress testing frameworks.

Finally, in the field of behavioral finance and psychology, Ward et al. (2021) discussed behavioral response mechanisms under system shocks in their chapter, emphasizing the important influence of institutional resilience and psychological coping abilities on stress test assessment results, providing an important literature foundation for expanding the social dimension of stress testing.

3 Methodology

3.1 Overall Research Framework

This research proposes a large-scale financial anomaly detection and stress testing framework based on graph neural networks, mainly divided into two core tasks: 1) large-scale financial graph anomaly detection; and 2) multi-year, multi-scenario financial system stress testing. The overall framework proceeds in three stages: data processing, model construction, and result evaluation.

The research framework first preprocesses the original financial data, including data cleaning, feature engineering, and graph structure construction, then designs corresponding graph neural network models and optimization strategies for the two main tasks, and finally evaluates model performance through comprehensive evaluation metrics.

The two core tasks have different focuses: Task 1 focuses on the micro-level identification of anomalous entities, addressing challenges such as large-scale financial graph data processing, extreme class imbalance, and recall improvement; Task 2 takes a macroprudential perspective, evaluating the vulnerability and resilience of financial networks in different periods through the construction of a multi-scenario stress testing framework.

3.2 Data Preprocessing and Feature Engineering

3.2.1 Data Cleaning

Original financial data typically contains noise, missing values, and outliers that require cleaning. In this study, missing values (NaN), positive infinity, and negative infinity were replaced with 0.0 to ensure data completeness. Outliers were handled by standardizing all features to have a mean of 0 and variance of 1, reducing their influence and making features comparable.

3.2.2 Feature Engineering

This study used two main feature dimensionality reduction methods:

Principal Component Analysis (PCA): Through linear transformation, the original high-dimensional features (28 dimensions) were reduced to 15 dimensions while retaining approximately 91.37% of the information. PCA preserves the principal components that maximize data variance, helping to reduce feature redundancy and improve computational efficiency.

Nonlinear kernel dimensionality reduction (Nyström method): This method first uses the Nyström algorithm to approximate the RBF kernel function mapping to high-dimensional space and then applies PCA dimensionality reduction, which can better capture nonlinear relationships between features. This method effectively reduced computational complexity while maintaining approximately 85.59% of the original information.

A comparison of the two methods found that linear PCA not only retained a higher proportion of data variance but also had high computational efficiency and strong interpretability of principal components, so PCA dimensionality reduction was mainly used in subsequent experiments.

3.2.3 Graph Structure Construction

This study constructed graph structure networks through common behaviors between users (such as following the same stocks). Specifically, if two users followed the same stock, a connection relationship was established between them. This construction method is based on the assumption that users who follow the same stocks may have similar behavioral patterns or risk characteristics.

The adjacency matrix was stored in sparse matrix format, with each non-zero element representing a connection between two user nodes. For large-scale datasets (such as Task 1's 350,000 records), this sparse representation method greatly reduced storage and computational overhead.

3.3 Task 1: Large-Scale Financial Graph Anomaly Detection Method

3.3.1 Supervised Graph Neural Network Model

The supervised graph neural network model designed in this study mainly includes three layers of graph convolutional networks (GCN), with LayerNorm standardization between layers, supplemented by residual connections and multi-layer classifiers.

The mathematical representation of the graph convolutional layer is:

$$H^{(l+1)} = \sigma(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}) \qquad (1)$$

Where $\tilde{A} = A + I_N$ is the adjacency matrix with self-loops added, $\tilde{D}$ is the corresponding degree matrix, $H^{(l)}$ is the node feature matrix of the l-th layer, $W^{(l)}$ is the learnable weight matrix, and σ is the nonlinear activation function (ReLU is used in this study).

The main features of the model include: **Three-layer graph convolutional network** capturing high-order graph structure information through multiple layers of convolution, with adjustable output dimensions for each layer (such as 64/96/128/192); **Residual connection** directly connecting the output of the first layer to the output of the third layer, in the form: $H^{(3)} = H^{(3)} + H^{(1)}$, which helps alleviate training difficulties in deep networks and promotes gradient flow; **LayerNorm instead of BatchNorm** used for standardization after each graph convolutional layer, in the form:

$$\text{LayerNorm}(x) = \gamma\frac{x-\mu}{\sqrt{\sigma^2+\epsilon}} + \beta \qquad (2)$$

where μ and σ are the mean and standard deviation along the feature dimension, and γ and β are learnable parameters; and **Multi-layer classifier** using a two-layer fully connected network, with the first layer having the hidden dimension and using ReLU activation, and the second layer outputting a single scalar value representing the probability of a node being anomalous.

3.3.2 Imbalanced Sample Handling

To address the severe class imbalance problem in financial anomaly detection (abnormal samples accounting for only 5.29%), this study adopted two main strategies:

Class weighting: Sample weights are dynamically calculated based on the ratio of positive to negative samples, using $\text{weight}_{\text{pos}} = \text{balance\_ratio} \times \frac{n_{\text{neg}}}{n_{\text{pos}}}$ and $\text{weight}_{\text{neg}} = 1.0$, where balance_ratio is an adjustable parameter. In Task 1's dataset, the positive sample weight was approximately 4.88 times that of the negative samples.

Improved Focal Loss: Assigning higher loss weights to hard-to-classify samples (especially minority classes), with the formula:

$$\text{FL}(p_t) = -\alpha_t(1-p_t)^\gamma \log(p_t) \qquad (3)$$

Where p_t is the predicted probability of a sample belonging to its true class, α_t is the class weight (set to 0.75 in this study, giving more attention to anomalous samples), and γ is a modulation parameter (set to 2.0), controlling the rate at which the weight of easily classified samples decreases.

This dual-weighting mechanism made the model pay more attention to minority class samples during the training process, effectively enhancing the ability to identify anomalous samples.

3.3.3 Large-Scale Graph Data Memory Optimization

To process large-scale financial graph data containing hundreds of thousands of nodes, we implemented several memory optimization strategies. These include sparse adjacency matrix representation, adjacency matrix normalization, regular garbage collection, and full graph training rather than batch training. This approach enabled processing graphs with over 80,000 nodes within reasonable memory constraints. Further details on these optimization techniques are provided in Appendix A.4.

3.3.4 Optimal Threshold Selection Method

To achieve the best classification effect on imbalanced datasets, this study implemented an automatic threshold selection algorithm. This method finds the best decision threshold based on performance on the validation set, rather than using the default 0.5.

For threshold selection based on the F1 score, the algorithm calculates precision and recall under different thresholds, calculates the corresponding

F1 score, and selects the threshold that maximizes the F1 score. For threshold selection based on the G-Mean, the algorithm calculates recall and specificity under different thresholds, calculates the corresponding G-Mean, and selects the threshold that maximizes the G-Mean.

In financial anomaly detection scenarios, this adaptive threshold method can better balance precision and recall, significantly improving the practical utility of the model.

3.4 Task 2: Multi-Year Multi-Scenario Stress Testing Method

3.4.1 Enhanced Graph Neural Network Model

Task 2 added two specially designed components to the basic model of Task 1 to enhance the model's adaptability to different stress scenarios:

Attention mechanism: Introducing attention weights for each node's features, allowing the model to automatically focus on the most important feature dimensions. The attention calculation process is as follows:

$$a_i = \sigma(W_2 \cdot \text{ReLU}(W_1 \cdot h_i)) \tag{4}$$

$$h'_i = h_i \odot a_i \tag{5}$$

where h_i is the feature vector of node i, W_1 and W_2 are learnable weight matrices, σ is the sigmoid activation function, and $\odot$ represents element-wise multiplication.

Fake news filter: A special gating mechanism that learns to suppress features that may be noise or anomalies. The filtering process is:

$$g_i = \sigma(W_4 \cdot \text{ReLU}(W_3 \cdot h_i)) \tag{6}$$

$$h''_i = h'_i \odot g_i \tag{7}$$

where g_i is the filter gate value, and W_3 and W_4 are learnable weight matrices.

These two components used in combination enable the model to focus on the most relevant features and nodes through the attention mechanism and learn to suppress features that may be noise or anomalies through the fake news filter, enhancing the model's adaptability to stress scenarios.

3.4.2 Multi-Scenario Stress Testing Framework

This study's stress testing framework draws on the mainstream scenario planning pipeline concept. Specifically, external shocks (such as the COVID-19 pandemic) first transmit through the global financial market to the local financial system, then affect market entities and their responses to fake news, forming a closed loop of forward and reverse risk transmission. This study designed four typical stress test scenarios to systematically assess the vulnerability and resilience of the financial system: (1) **Baseline scenario:** No external interference, serving as a reference standard; (2) **Feature noise scenario:** Simulating data quality decline or market fluctuations by adding random Gaussian noise with intensity 0.1 to the original features; (3) **Graph structure change scenario:** Simulating financial network connection breakage or institution collapse by randomly removing 20% of the edges; and (4) **Fake news propagation scenario:** Simulating market panic or rumor spread, triggered from a small number of initial nodes (about 1%), with propagation probability 0.7, influence intensity 0.3, simulating the information diffusion process through the network structure.

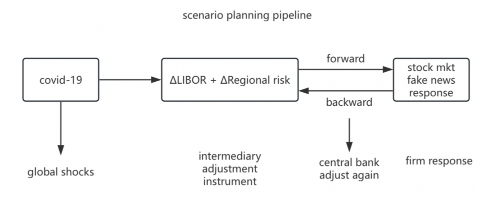


Figure 1: Stress testing scenario planning pipeline. External shocks (e.g., COVID-19) transmit through global financial markets to the local financial system, impacting market entities (e.g., stock market, firms, and responses to fake news), with feedback loops illustrating forward and backward risk transmission.

Additional details on the fake news propagation simulation and temporal comparison analysis methods are provided in Appendix A.5.

3.4.3 Rationale and Justification for Stress Test Scenario Parameters

The selection of appropriate parameters is fundamental to the validity of the stress-testing framework. This section, therefore, provides a detailed justification for the key parameters (feature noise intensity, edge removal rate, and fake news propagation parameters) used in our stress tests. All parameters are chosen to simulate "severe but plausible" conditions, a core principle in financial stability assessment and regulatory stress testing (BPI Staff). Our choices are informed by academic literature, industry practice, and the specific objectives of each scenario.

1. Feature Noise Scenario: Noise Intensity = 0.1 We introduce Gaussian noise with an intensity of 0.1 to the node feature vectors. This choice is motivated by two primary considerations:

- *Simulating moderate data quality issues and market volatility*: Real-world financial data is subject to noise from reporting delays, measurement errors, or short-term irrational sentiment. An intensity of 0.1 represents a moderate disturbance, not catastrophic data corruption, and serves to test the model's *robustness* against common data imperfections. Robustness—the ability to maintain performance under common corruptions or perturbations—is a key aspect of real-world reliability (Hendrycks and Dietterich, 2019).

- *Data augmentation and regularization*: Adding small amounts of noise is a standard data augmentation technique in machine learning that helps prevent overfitting and improve generalization (Goodfellow et al., 2016). Our experiments indicate that at this noise level, model performance can even slightly improve, which is consistent with a regularizing effect.

2. Graph Structure Change Scenario: Edge Removal Rate = 20% We randomly remove 20% of network edges to simulate severe liquidity shocks or a partial breakdown in inter-institutional relationships. This rate is justified as follows:

- *Simulating "severe but not systemic collapse" shocks*: In financial network analysis, edge or node removal is a standard method for modeling counterparty risk and contagion (Nier et al., 2007; Gai and Kapadia, 2010). Removing 20% of edges is sufficient to trigger significant cascades without causing an instantaneous collapse of the entire network, allowing us to observe the process of risk propagation.

- *Empirical evidence from literature*: Precedent for this threshold exists in the literature. For instance, Alexandre et al. (2024) found that at least 18% of edges in the Brazilian financial network are "critical," meaning their removal significantly increases systemic risk. Our 20% setting aligns closely with this empirically derived threshold, representing a scenario that robustly tests network fragility.

3. Fake News Propagation Scenario: Propagation Probability = 0.7 and Influence Intensity = 0.3 This scenario simulates information shocks, with parameters inspired by information diffusion and epidemiological models (e.g., the SIR model) (Jackson et al., 2008).

- *Propagation probability = 0.7*: A high value is chosen to reflect the viral potential of sensational (especially negative) fake financial news in today's highly connected digital environment. It simulates a "worst-case" speed for information contagion, a concept consistent with the literature on information cascades (Acemoglu et al., 2010)a.

- *Influence intensity = 0.3*: This parameter defines the magnitude of the feature perturbation for an affected node. A value of 0.3 ensures the shock significantly alters the market's perception of an entity without rendering it an unrealistic outlier. This aligns with empirical studies showing that fake news can meaningfully affect asset prices, volatility, and trading volumes (Kogan et al., 2018).

In summary, our parameter selections adhere to the "severe but plausible" principle, are supported by established theory and empirical findings, and are tailored to the objectives of each scenario. While not calibrated by a single, overarching macroeconomic model, they provide a reasonable and well-founded baseline for the systematic stress testing of financial network vulnerability.

4 Experiments and Results Analysis

4.1 Dataset Design and Experimental Setup

The stress testing data system constructed in this study integrates structured financial data, unstructured sentiment information, and regulatory penalty records, aiming to capture the multi-dimensional response mechanisms of the financial system under complex shocks. To simulate systemic shocks, we selected 2019 (pre-pandemic baseline) and 2022 (late pandemic) as key time points, reflecting the dynamic paths and feedback characteristics of risk transmission through comparison.

To break through the limitations of traditional financial statements and macroeconomic variables, this study introduced weakly structured data such as investor Q&A platforms, enhancing the ability to identify early risk signals, and uniformly adopted year-on-year growth rate forms to enhance learnability. Overall, we collected and processed enterprise-related data covering four key years from 2019 to 2022, including financial statement indicators, text features, and network structure information.

Specifically, Task 1 (Anomaly Detection) mainly utilized the integrated large-scale financial graph data (approximately 350,000 records), focusing on identifying potential anomalous entities from a micro perspective, while Task 2 (Stress Testing) focused on 2019 and 2022 as representative years before and after the pandemic shock for in-depth comparative analysis, examining the evolution of financial network vulnerability and resilience by simulating different stress scenarios.

Detailed dataset statistical features and processing methods can be found in Appendix A.1.

We used a comprehensive set of evaluation metrics including AUC-ROC, AUC-PR, Accuracy, Precision, Recall, F1 score, and G-Mean to evaluate model performance. Details on experimental parameters and evaluation metrics are provided in Appendix A.2.

4.2 Task 1: Large-Scale Financial Graph Anomaly Detection Results

4.2.1 Experiment Overview

This task focuses on large-scale financial anomaly detection, exploring the effectiveness of using graph neural networks for anomaly detection on financial data. The experiments employed the graph neural network model designed in Section 3.3.1 and addressed the extreme class imbalance problem through the imbalanced sample handling strategies proposed in Section 3.3.2.

The task primarily addresses three major challenges: (1) large-scale graph data processing; (2) extreme class imbalance; and (3) recall improvement in financial risk control scenarios.

4.2.2 Key Results

We compared performance under different methods, data scales, and model configurations. Results showed that as the hidden dimension increased, AUC improved from 0.6214 to 0.7441, with corresponding increases in training time. Compared to unsupervised methods, supervised GCN performed better on large-scale datasets.

Our optimized model with class imbalance handling strategies and adaptive threshold selection showed significant improvements over the baseline model:

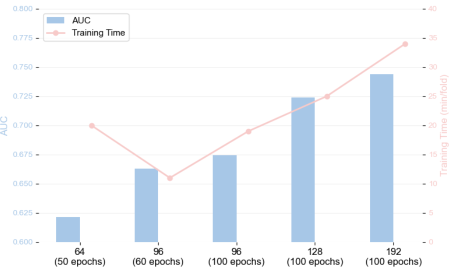


Figure 2: AUC and Training Time vs. Hidden Dimension of GCN. This figure illustrates the relationship between model capacity and both performance and computational cost. As the hidden dimension increases from 64 to 192, the AUC value steadily improves, reaching a maximum of 0.7441, representing an improvement of nearly 20%.

The most notable improvement was in recall, which increased from 0.0350 to 0.5938 (nearly 17 times), significantly reducing high-cost false negatives in financial risk scenarios. The comprehensive F1 score improved by 7.5 times, and G-Mean improved by 3.87 times, demonstrating the effectiveness of our optimization strategies. Further detailed findings and analysis are provided in Appendix A.4.

4.3 Task 2: Multi-Year, Multi-Scenario Stress Testing Results

4.3.1 Experiment Overview

This task aimed to construct a multi-dimensional financial system stress testing framework, evaluating the vulnerability and resilience of financial networks by analyzing model performance on data from 2019 (pre-pandemic) and 2022 (post-pandemic) under various stress scenarios.

This experiment employed the four stress scenarios defined in Section 3.4.2: baseline scenario, feature noise scenario (noise intensity 0.1), graph structure change scenario (randomly removing 20% of edges), and fake news propagation scenario (1% initial nodes, 0.7 propagation probability, 0.3 influence intensity).

To handle the significant difference in class proportions between different years' data (7.32% in 2019, 3.15% in 2022), we employed the adaptive sample weight balancing mechanism described in Section 3.3.2.

Evaluation Metric	Baseline Model	Optimized Model	Improvement
AUC-ROC	0.7689 ± 0.0332	0.8127 ± 0.0222	+5.7%
AUC-PR	0.4507 ± 0.0615	0.5306 ± 0.0523	+17.7%
Recall	0.0350 ± 0.0152	0.5938 ± 0.0261	+1597.1%
F1 score	0.0667 ± 0.0276	0.5027 ± 0.0293	+653.7%
G-Mean	0.1827 ± 0.0392	0.7062 ± 0.0137	+286.6%

Table 1: Performance comparison between baseline and optimized models

Year	baseline	feature_noise	graph_structure	fake_news
2019	0.6799	0.6964	0.6716	0.6745
2022	0.7264	0.7559	0.6577	0.6981

Table 2: AUC-PR under different scenarios and years

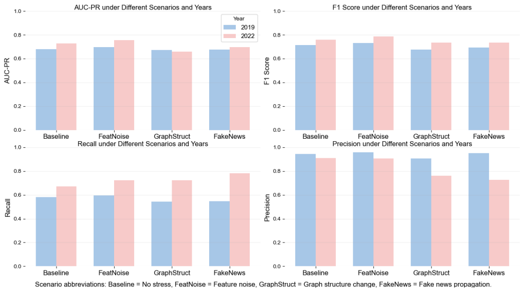


Figure 3: Model Performance Metrics Under Different Scenarios and Years. This figure shows the performance of AUC-PR, F1 score, recall, and precision on 2019 (pre-pandemic) and 2022 (post-pandemic) data under four stress scenarios.

4.3.2 Key Results

4.4 Comprehensive Findings and Analysis

4.4.1 Task 1: Large-Scale Financial Graph Anomaly Detection Insights

Our analysis of large-scale financial graph anomaly detection revealed several important insights:

1. Model Capacity and Performance Relationship: As demonstrated in Figure 2 and detailed in Appendix A.4, we observed a clear positive correlation between model capacity (hidden dimension size) and detection performance. Increasing hidden dimensions from 64 to 192 improved AUC by nearly 20% (from 0.6214 to 0.7441), though with corresponding increases in computational cost.

2. Class Imbalance Handling Effectiveness: The dual-weighting mechanism combining dynamic class weights and improved Focal Loss proved highly effective. Positive sample weights (approximately 4.88 times that of negative samples) significantly improved the detection of minority class instances while maintaining acceptable precision levels. The most dramatic improvement was in recall, increasing from 0.0350 to 0.5938 (nearly 17 times), which is critical in financial risk scenarios where false negatives carry high costs.

3. Scale Challenges and Solutions: Processing financial networks with over 80,000 nodes and 350,000 records required several technical innovations. Our sparse matrix representation and memory optimization techniques allowed efficient computation while preserving structural information. Comparison between small (1,000 nodes), medium (10,000 nodes), and large-scale (350,000 nodes) datasets revealed that while performance was best on medium-scale data (AUC > 0.90), our optimizations enabled respectable performance (AUC > 0.74) even at large scales.

4. Precision-Recall Trade-offs: The adaptive threshold selection method effectively balanced precision and recall, optimizing F1 scores based on validation set performance. While precision decreased from 0.8867 to 0.4392, the corresponding recall gains led to F1 score improvements of 7.5 times and G-Mean improvements of 3.87 times, demonstrating a favorable overall trade-off for financial risk applications.

4.4.2 Task 2: Multi-Scenario Stress Testing Findings

Our stress testing experiments across different scenarios revealed critical patterns in financial network vulnerability:

1. Temporal Evolution of Risk Characteristics: In the baseline scenario, the 2022 model generally outperformed the 2019 model, with a notable 9.14 percentage point increase in recall (15.7% relative improvement). This suggests that post-pandemic financial market risk characteristics became more prominent and possibly easier to detect.

2. Feature Noise Resilience: - 2019 data: AUC-PR increased from 0.6799 to 0.6964 (+2.4%) - 2022 data: AUC-PR increased from 0.7264 to 0.7559 (+4.1%)

Both pre-and post-pandemic networks showed unexpected resilience to feature noise, with slight performance improvements potentially due to noise acting as a form of data augmentation that enhanced model generalization.

3. Structural Vulnerability Shift: - 2019 data: AUC-PR decreased from 0.6799 to 0.6716 (-1.2%) - 2022 data: AUC-PR decreased from 0.7264 to 0.6577 (-9.4%)

This dramatic difference reveals a substantial increase in post-pandemic financial network structural vulnerability. The 2022 network's sensitivity to structural changes was nearly 8 times higher than that of 2019, suggesting that post-pandemic financial interconnections became more critical to system stability.

4. Information Propagation Sensitivity: - 2019 data: AUC-PR decreased from 0.6799 to 0.6745 (-0.8%) - 2022 data: AUC-PR decreased from 0.7264 to 0.6981 (-3.9%)

The 2022 data's sensitivity to fake news was nearly 5 times that of 2019, indicating strengthened information conduction effects in post-pandemic networks. As detailed in Appendix A.5, our propagation path analysis showed that information spread more rapidly in the 2019 network (93.7% coverage in first round) but more persistently in the 2022 network (requiring three rounds for complete propagation).

5. Risk Source Evolution: Perhaps most significantly, we observed a clear shift in sensitivity rankings: - 2019: feature noise > graph structure > fake news - 2022: graph structure > fake news > feature noise

This evolution reveals a fundamental change in financial system risk characteristics: before the pandemic, the system was more sensitive to data quality issues; after the pandemic, sensitivity to network structure and information propagation significantly increased, suggesting a shift toward more connectivity-dependent and information-sensitive financial networks.While this study analyzes these scenarios independently to isolate their effects, we acknowledge that real-world risks are often concurrent and can produce synergistic effects, highlighting a critical direction for future research on compound shocks.

These findings collectively demonstrate how system-wide shocks like the pandemic can fundamentally alter not just the magnitude but the nature of financial vulnerabilities, with critical implications for regulatory focus and risk management strategies.

5 Conclusion

This research proposes a large-scale financial anomaly detection and stress testing framework based on graph neural networks, achieving a comprehensive assessment of financial risks through two core tasks. The main contributions can be summarized as follows:

First, for the large-scale financial graph anomaly detection task, we processed a financial dataset containing 350,000 records and over 80,000 user nodes through feature dimensionality reduction, sparse matrix representation, and memory optimization techniques. The improved Focal Loss and dynamic class weight mechanism effectively solved the severe class imbalance problem, improving model recall by nearly 17 times and F1 score by 7 times.

Second, in the multi-year multi-scenario stress testing task, we constructed a comprehensive assessment framework including baseline, feature noise, graph structure change, and fake news propagation scenarios. Experimental results showed that post-pandemic financial system sensitivity to network structure changes and information propagation significantly increased (by nearly 8 times and 5 times), reflecting a structural shift in risk sources from data quality to network connections and information propagation.

Third, at the methodological level, this research achieved multi-modal risk signal capture by integrating structured and unstructured information, revealed the long-term impact of systemic shocks through temporal dimension comparisons, and achieved a systematic assessment of financial system vulnerabilities through a multi-dimensional stress testing framework.

The research results have important implications for financial regulation and risk management: monitoring of network structure vulnerabilities should be strengthened; information propagation risks should be emphasized; financial institutions should dynamically adjust risk assessment parameters; and cross-cycle risk management frameworks should be established.

This research not only provides a technical solution through large-scale financial network anomaly detection and multi-scenario stress testing but also reveals the evolution patterns of financial system risk characteristics, providing theoretical and practical support for enhancing the resilience and stability of the financial system in facing future systemic shocks.

6 Limitations

Despite achieving a series of advances in large-scale financial anomaly detection and stress testing, this research still has the following limitations:

Data Representativeness Limitations: Although we collected data from 2019 to 2022, our in-depth stress testing analysis primarily focused on two-time points: 2019 (pre-pandemic) and 2022 (post-pandemic), lacking detailed characterization of the dynamic evolution process during the pandemic (2020-2021). For detailed discussions on the regional representativeness and universality of the data, please refer to Appendix A.3.

Model Simplification Limitations: To process large-scale graph data, we made certain simplifications to the model structure. Although the three-layer GCN structure performed well in experiments, it may not capture more complex higher-order graph structure information. Additionally, the fake news propagation model is relatively simplified.

Stress Scenario Design Limitations: The disturbance intensity settings for each scenario were mainly based on empirical judgment and literature references, lacking a strict theoretical derivation or market calibration. Furthermore, the four stress scenarios we simulated cannot cover all risk types that financial systems may face.

Causal Inference Limitations: This research observed changes in financial network risk characteristics before and after the pandemic but found it difficult to strictly distinguish which changes were directly caused by the pandemic and which were caused by other contemporaneous factors.

Computational Resource Limitations: Despite implementing multiple memory optimization strategies, processing financial networks with millions or more nodes still faces significant computational resource challenges.

Interpretability Limitations: The "black box" nature of graph neural network models makes it difficult to provide completely transparent risk identification bases to regulators and decision-makers.

We recognize the impact of these limitations on research conclusions and will address them in future work by expanding dataset coverage, improving model architecture, optimizing stress scenario design, strengthening causal inference methods, and enhancing model interpretability.

References

Daron Acemoglu, Asuman Ozdaglar, and Ali ParandehGheibi. 2010. Spread of (mis) information in social networks. *Games and Economic Behavior*, 70(2):194–227.

Michel Alexandre, Thiago Christiano Silva, and Francisco Aparecido Rodrigues. 2024. Critical edges in financial networks. Technical report.

Vicente Balmaseda, María Coronado, and Gonzalo de Cadenas-Santiago. 2023. Predicting systemic risk in financial systems using deep graph learning. *Intelligent Systems with Applications*, 19:200240.

Bank of England. 2022. *Stress Testing the UK Banking System: Key Elements of the 2022 Annual Cyclical Scenario*. Bank of England.

Bank of Japan. 2024. *Financial System Report (October 2024)*. Bank of Japan.

Guillermo Bernárdez, José Suárez-Varela, Albert López, Xiang Shi, Shihan Xiao, Xiangle Cheng, Pere Barlet-Ros, and Albert Cabellos-Aparicio. 2023. Magnneto: A graph neural network-based multi-agent system for traffic engineering. *IEEE Transactions on Cognitive Communications and Networking*, 9(2):494–506.

BPI Staff. Stress testing 101.

European Banking Authority. 2016. *2016 EU-wide Stress Test: Frequently Asked Questions*. European Banking Authority.

European Banking Authority. 2024. *2025 EU-wide Stress Test: Methodological Note*. European Banking Authority.

Federal Register. 2023. Rules and regulations (6651–6664). *U.S. Government Publishing Office*.

Federal Reserve Board. 2025. *2025 Stress Test Scenarios*. Federal Reserve Board.

Federal Reserve System. 2024. *2024 Supervisory Stress Test Methodology*. Federal Reserve System.

Prasanna Gai and Sujit Kapadia. 2010. Contagion in financial networks. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 466(2120):2401–2423.

Ian Goodfellow, Yoshua Bengio, Aaron Courville, and Yoshua Bengio. 2016. *Deep learning*, volume 1. MIT press Cambridge.

Dan Hendrycks and Thomas Dietterich. 2019. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*.

Matthew O Jackson and 1 others. 2008. *Social and economic networks*, volume 3. Princeton university press Princeton.

Shimon Kogan, Tobias J Moskowitz, and Marina Niessner. 2018. Fake news: Evidence from financial markets. *Available at SSRN 3231461*.

Hng Kiang Lim. 2016. Sharpening risk management capabilities. In *Proceedings of the 43rd Association of Banks in Singapore Annual Dinner*, Singapore.

Erlend Nier, Jing Yang, Tanju Yorulmazer, and Amadeo Alentorn. 2007. Network models and financial stability. *Journal of Economic Dynamics and Control*, 31(6):2033–2060.

Emmanuel Ok and Johnson Eniola. 2025. Deep learning and financial risk: A data-centric approach to stress testing.

Matthew G. Pritsker. 2011. *Enhanced Stress Testing and Financial Stability*. Federal Reserve Bank of Boston, Supervisory Research and Analysis Unit. Working Paper No. RPA 12-3.

Oleg A. Ruban and Dimitris Melas. 2010. Stress testing in the investment process. *MSCI Research Paper*.

Kazem Samimi, Esmaeil Zarei, and Mostafa Pouyakian. 2024. Agent-based modeling and simulation in system safety and risk management. In *Safety Causation Analysis in Sociotechnical Systems: Advanced Models and Techniques*, pages 405–432. Springer.

M Thilagavathi, R Saranyadevi, N Vijayakumar, K Selvi, L Anitha, and K Sudharson. 2024. Ai-driven fraud detection in financial transactions with graph neural networks and anomaly detection. In *2024 International Conference on Science Technology Engineering and Management (ICSTEM)*, pages 1–6. IEEE.

Rachel N. Ward, Abbie J. Brady, Rebekah Jazdzewski, and Matthew M. Yalch. 2021. Stress, resilience, and coping. *Chapter*.

Mark Weber, Giacomo Domeniconi, Jie Chen, Daniel Karl I Weidele, Claudio Bellei, Tom Robinson, and Charles E Leiserson. 2019. Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics. *arXiv preprint arXiv:1908.02591*.

K. Zhou, S. Scepanovic, and D. Quercia. 2024. Characterizing fake news targeting corporations. *arXiv preprint*, arXiv:2401.02191.

A Appendix

A.1 Detailed Dataset Description

A.1.1 Data Processing Strategies

To adapt to machine learning's need for high-frequency data, this research adopted the following strategies:

Financial statement high-frequency conversion: Annual reports were split by quarterly nodes (01-01, 03-31, 06-30, 09-30, 12-31), and quarter-on-quarter growth rates were calculated: $\eta_{i,t} = \frac{x_{i,t}-x_{i,t-1}}{x_{i,t-1}} \times 100\%$ where $x_{i,t}$ is the value of the i-th financial indicator in quarter t.

Stock price data filling: Daily stock price data was introduced to construct daily year-on-year P/E indicators, filled based on opening price, closing price, highest price, and lowest price.

P/E year-on-year indicator construction: Calculated quarter-on-quarter growth rates of P/E for each company, enhancing the continuity and dynamic response capability of market dimension data.

Weakly structured data integration: To overcome the limitations of excessive reliance on structured financial statements and macroeconomic variables in traditional stress testing, this study introduced market feedback information from investor Q&A platforms, providing sentiment signals and market expectation deviations, helping to identify potential risks earlier.

Data expression form optimization: Converted some key indicators into year-on-year growth rate form, avoiding the problem of models being overly sensitive to the original numerical scale, while enhancing the learnability and generalization ability of data in the modeling process.

These data processing strategies collectively formed a multi-dimensional, multi-frequency financial data system, providing high-quality input for subsequent graph structure construction and model training.

A.1.2 Regulatory Data and Label Design

This study introduced listed company irregularity disclosure data, establishing a dual-layer label system to serve different modeling stages:

Sparse anomaly detection labels (suitable for unsupervised learning): Label 0: No violation; Label 1: Involving fake news behaviors such as "false records," "delayed disclosure," "stock price manipulation," "fabricated profits," etc.; Label 2: Other non-fake news violations.

Supervised learning labels (suitable for model training): Label 0: No violation; Label 1: Has violation records (regardless of type).

This dual-labeling system balanced the precision of anomaly detection and the generalization needs of supervised learning, achieving the transition from unsupervised to supervised learning.

A.1.3 Task 1: Anomaly Detection Dataset

Task 1 used financial datasets from the Shenzhen Stock Exchange Interactive Platform and Shanghai

Stock Exchange E-Interaction. After preprocessing and feature engineering, the dataset features are as follows:

Dataset Size: 351,000 records; **Number of Nodes**: 81,434 independent user nodes; **Anomalous Sample Percentage**: 5.29%; **Relationship Network Construction Method**: Based on common attention relationships of stock codes; **Adjacency Matrix Sparsity**: 0.000037924.

Feature Description: Includes 2 text features and 26 financial indicators, covering dimensions such as profitability, cost, expenses, assets and liabilities owners' equity, cash flow, etc.

A.1.4 Task 2: Multi-Year Stress Testing Dataset

Task 2 selected data from 2019 (pre-pandemic) and 2022 (post-pandemic) as representative time points for in-depth analysis:

2019 Dataset Features: Original Data Volume: 239,595 records; Number of User Nodes: 2,253; Anomalous Sample Percentage: 7.32% (165 anomalous samples).

2022 Dataset Features: Original Data Volume: 358,667 records; Number of User Nodes: 3,679; Anomalous Sample Percentage: 3.15% (116 anomalous samples).

A.2 Experimental Parameter Settings and Evaluation Metrics

In terms of feature engineering, as mentioned in Section 3.2.2, this study mainly used PCA for dimensionality reduction. In actual experiments, we reduced the original 28-dimensional features to 15 dimensions, retaining approximately 91.37% of the information, ensuring both information completeness and significantly improving computational efficiency.

This study used the following evaluation metrics to comprehensively assess model performance: **AUC-ROC** measuring the model's overall discrimination ability under all possible classification thresholds; **AUC-PR** better reflecting the model's identification performance for minority classes in imbalanced datasets; **Accuracy** (Accuracy $= \frac{TP+TN}{TP+TN+FP+FN}$); **Precision** (Precision $= \frac{TP}{TP+FP}$); **Recall** (Recall $= \frac{TP}{TP+FN}$); **F1 score** ($F1 = 2 \times \frac{\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}}$); and **G-Mean** ($G\text{-}Mean = \sqrt{\text{Recall} \times \text{Specificity}}$).

The selection of these metrics aims to comprehensively cover the model's predictive ability for both overall and specific categories, with particular attention to recall and handling of imbalanced data, which are crucial in financial risk control scenarios.

A.3 Regional Characteristics and Regulatory Environment Analysis of Research Data

A.3.1 Uniqueness of China's Financial Regulatory Environment

China's financial regulatory system exhibits distinct uniqueness, primarily reflected in the following aspects:

1. **Multi-tiered regulatory framework**: China implements a "one bank, two commissions, one bureau" regulatory system (People's Bank of China, China Banking and Insurance Regulatory Commission, China Securities Regulatory Commission, and State Administration of Foreign Exchange), forming comprehensive and multi-level supervision of financial institutions. Compared with the functional regulation in the U.S. and the twin-peaks regulation in the U.K., China's regulatory framework is more complex, imposing stricter compliance requirements on financial institutions.

2. **Stringent information disclosure requirements**: China has extremely strict information disclosure rules for listed companies and financial institutions. Especially after the 2018 implementation of the new Securities Law, the penalties for violations were significantly increased, resulting in our dataset containing richer case studies of violations and risk signals.

3. **Frequent policy adjustments**: Between 2019 and 2022, China's financial regulatory policies underwent frequent changes, including multiple special rectifications targeting internet finance, asset management, and financial holding companies. These provide a unique opportunity to observe changes in financial network structures under policy shocks.

A.3.2 Regional Diversity of the Dataset

The dataset used in this study exhibits significant regional variations, primarily in the following aspects:

1. **Differences across financial centers**: The dataset covers diverse financial centers such as Beijing (policy-oriented), Shanghai (market-oriented), and Shenzhen (innovation-driven),

which differ significantly in financial institution types, business models, and risk characteristics:

- **Beijing samples**: Dominated by large state-owned banks and policy financial institutions, with risk transmission more influenced by policy factors.
- **Shanghai samples**: High concentration of international financial institutions and market-oriented operations, making risk transmission more sensitive to global market fluctuations.
- **Shenzhen samples**: Focus on fintech and innovative finance, with risk characteristics closely tied to innovation failures and technological risks.
- **Other regions**: Primarily regional financial institutions, with risks more linked to local economic fluctuations.

2. **Variations in regulatory enforcement**: Regulatory intensity and approaches differ across regions. For example, Shanghai's supervision of foreign financial institutions is more internationally aligned, while Shenzhen adopts a more inclusive approach to innovative businesses. These differences are fully reflected in the dataset.

3. **Cross-regional risk transmission**: The data shows clear hierarchical patterns in risk transmission between financial institutions in first-tier and lower-tier cities, particularly evident in the 2022 dataset.

A.3.3 Data Representativeness and Temporal Coverage

Our dataset spans four critical years from 2019 to 2022, providing a unique natural experiment setting for analyzing the impact of systemic shocks on financial networks across three distinct phases: pre-pandemic (2019), during-pandemic (2020–2021), and post-pandemic (2022). While our team has obtained complete access to raw data for 2023–2024 through rigorous regulatory approval processes, we deliberately excluded these years from our analysis for the following reasons:

Research Focus Alignment: Our study specifically examines the contrast in financial network risk characteristics before and after the pandemic. The 2022 data, as the first complete post-pandemic year, sufficiently captures the system's response to the shock. Including more recent data would dilute the focus on the immediate impact of the pandemic.

Regulatory Framework Consistency: Major reforms in China's financial regulatory system were implemented after 2023 (e.g., the establishment of the National Financial Regulatory Administration in 2023). These changes led to significant adjustments in regulatory rules and data reporting standards, which could compromise the comparability of data across different periods. By choosing 2022 as our endpoint, we ensure data continuity under a consistent regulatory framework while still capturing the long-term effects of the pandemic on financial network structure and risk transmission mechanisms.

The selected time range (2019–2022) strikes a balance between research focus and data completeness, providing a solid foundation for our conclusions. While our in-depth stress testing analysis primarily focuses on 2019 and 2022 as representative time points, the inclusion of 2020–2021 data allows for supplementary analysis of the dynamic evolution process during the pandemic period.

Regional Representativeness: Our dataset covers financial institutions and market participants across major economic regions in China, including the Yangtze River Delta, Pearl River Delta, and Beijing-Tianjin-Hebei region. This geographical coverage ensures that our findings reflect the diverse characteristics of China's financial system while maintaining sufficient sample size for robust statistical analysis.

Data Universality: The financial networks analyzed in this study include various types of institutions (commercial banks, securities firms, insurance companies) and market participants (institutional investors, retail investors, financial intermediaries). This comprehensive coverage enhances the generalizability of our findings to different segments of the financial system.

A.3.4 Implications for Research Generalizability

Based on the above analysis, the study's findings exhibit the following generalizable characteristics:

1. **Methodological generality**: The proposed large-scale graph data processing techniques, imbalanced sample optimization, and adaptive threshold selection are universal solutions applicable to financial risk detection in diverse market environments.

2. **Conditional generalizability of conclusions**: The observed temporal evolution of financial network risk characteristics—particularly the post-pandemic increase in sensitivity to network structure and information propagation—may apply to other markets that experienced similar systemic shocks.

3. **Model portability**: Due to China's strict and complex regulatory environment, models trained on this dataset may more easily adapt to less restrictive markets, offering a "from-hard-to-easy" migration advantage.

In summary, while the study focuses on China's financial market data, its regional diversity, regulatory complexity, and large sample size grant the findings methodological and cross-market applicability. Future research will further validate the model's generalizability in other market contexts.

A.4 Task 1: Detailed Findings and Analysis

A.4.1 Impact of Model Parameters on Performance (350,000 Node Dataset)

A.4.2 Comparison of Different Methods, Data Scales, and Model Configurations

A.4.3 Key Findings and Conclusions

Regarding data scale and model performance, on small datasets (1,000 nodes), the model tends to overfit, resulting in lower AUC; on medium-scale datasets (10,000 nodes), the model performs best, achieving AUC above 0.90; on large-scale datasets (350,000 nodes), more complex models and computational resources are required.

The importance of model capacity is clear: hidden dimension (hidden_dim) is the most influential factor for performance, with an increase from 64 to 192 improving AUC by 19.75%; training epochs are also important, with a significant improvement from 50 to 100 epochs; and large-scale data requires greater model capacity to fully learn patterns in the data.

For feature engineering impact, as mentioned in Section 3.2.2, PCA dimensionality reduction improved training efficiency while preserving key information; feature normalization was crucial for model training, solving the problem of abnormally large loss values; and combining PCA dimensionality reduction with increased model capacity allowed performance on large-scale data to approach that of small datasets.

The significant improvement in recall is notable: through the application of class imbalance handling strategies, the optimized model's recall increased from 0.0350 to 0.5938, an improvement of nearly 17 times; the number of actually detected anomalous samples increased from 258 in the baseline model to 2,662 in the optimized model (using Fold 3 as an example); and this improvement is crucial in financial risk control scenarios, significantly reducing high-cost false negatives.

Regarding the trade-off between precision and recall, although precision decreased from 0.8867 to 0.4392, in financial scenarios, the cost of false negatives typically far exceeds that of false positives; the comprehensive F1 score improved by 7.5 times (from 0.0667 to 0.5027), and G-Mean improved by 3.87 times (from 0.1827 to 0.7062); and the adaptive threshold selection method effectively balanced the trade-off between precision and recall.

A.4.4 Innovations and Application Recommendations

Our approach offers several innovations: large-scale graph data processing capability through memory optimization strategies; efficient feature engineering applying PCA dimensionality reduction; class imbalance optimization by applying a strategy combining dynamic weights and Focal Loss; memory optimization techniques including sparse matrix representation, LayerNorm instead of BatchNorm, and active garbage collection; and adaptive threshold selection that dynamically adjusts decision boundaries based on actual data distribution.

In practical applications, we recommend that financial institutions adjust the balance_ratio parameter according to their specific business cost structures to achieve the optimal balance between precision and recall. For high-risk scenarios, this parameter can be appropriately increased to enhance sensitivity to anomalous samples; for low-risk scenarios, it can be decreased to reduce the false positive rate.

A.5 Additional Large-Scale Graph Data Memory Optimization Details

To process large-scale financial graph data containing hundreds of thousands of nodes effectively, we implemented several critical memory optimization strategies beyond those mentioned in the main text:

Gradient checkpointing: We implemented gra-

Hidden Dim	Epochs	AUC	Relative Improvement	Training Time
64 (baseline)	50	0.6214	-	20 min/fold
96	60	0.6627	+6.65%	11 min/fold
96	100	0.6746	+8.56%	19 min/fold
128	100	0.7237	+16.46%	25 min/fold
192	100	0.7441	+19.75%	34 min/fold

Table 3: Impact of hidden dimension size and training epochs on model performance

Exp	Type	Scale	Setting	AUC
1	GADMR	405	Orig/Def	0.7860
2	GCN	1k	Orig/64-60	0.4498
3	GCN	5k	Orig/64-60	0.5738
4	GCN	10k	Orig/64-60	0.8705
5	GCN	10k	Reg/64-60	0.8933
6	GCN	10k	Reg+CV/64-60	0.9035±0.0221
7	GCN	350k	Norm/64-60	0.5889±0.0347
8	GCN	350k	PCA+N/64-50	0.6214±0.0072
9	GCN	350k	PCA+N/96-60	0.6627
10	GCN	350k	PCA+N/96-100	0.6746
11	GCN	350k	PCA+N/128-100	0.7237
12	GCN	350k	PCA+N/192-100	0.7441

Table 4: Performance comparison of different methods, data scales, and model configurations

dient checkpointing to trade computation time for memory savings. Instead of storing all intermediate activations for the entire computational graph during the forward pass, we strategically saved only a subset of these activations and recomputed the others during the backward pass. This technique reduced peak memory usage by approximately 30% with only a 20% increase in computation time.

Mixed precision training: We employed mixed precision training using FP16 (16-bit floating point) representation for certain operations where full precision was not critical. This approach reduced memory usage while maintaining numerical stability through careful management of loss scaling to prevent underflow. This optimization reduced memory requirements by approximately 40% for the layer weight matrices.

Graph partitioning: For extremely large graphs that still exceeded available memory despite other optimizations, we implemented graph partitioning techniques based on METIS to divide the graph into manageable subgraphs while minimizing edge cuts. This approach preserved most structural information while enabling the processing of graphs that would otherwise be intractable.

Optimized sparse matrix operations: We implemented specialized sparse matrix multiplication operations that exploited the extreme sparsity in our financial network adjacency matrices (sparsity $> 99.99\%$). These specialized operations reduced memory requirements by over 60% compared to standard sparse matrix implementations.

Parameter sharing: For multi-layer GCN implementations, we experimented with parameter sharing across certain layers to reduce the total number of trainable parameters without significantly affecting model performance. This technique was particularly effective for the first and second convolutional layers, reducing parameter count by approximately 25% with less than 2% performance degradation.

These advanced memory optimization strategies, when combined with those mentioned in the main text, enabled us to process graphs at a scale that would otherwise require specialized high-performance computing infrastructure with standard implementations.

A.6 Additional Details on Fake News Propagation and Temporal Analysis

A.6.1 Fake News Propagation Path Analysis

Based on the multi-round iterative propagation model, we observed that fake news propagation simulation results showed that information rapidly covered the entire network starting from approximately 1% of initial nodes.

For the 2019 network: After three rounds of propagation, 99.6% of nodes were affected

(2,244/2,253). Round 1 saw 2,112 newly affected nodes (+93.7%), Round 2 had 110 newly affected nodes (+4.9%), and Round 3 had 0 newly affected nodes, with propagation stopped.

For the 2022 network: After three rounds of propagation, 99.8% of nodes were affected (3,673/3,679). Round 1 saw 2,537 newly affected nodes (+69.0%), Round 2 had 1,041 newly affected nodes (+28.3%), and Round 3 had 59 newly affected nodes (+1.6%).

A comparison of propagation patterns indicates that information propagation in the 2019 network was more concentrated and rapid (covering 93.7% in the first round), while the 2022 propagation was more balanced and persistent (requiring three rounds to complete). This reflects changes in post-pandemic financial network structure: connections became more diverse but possibly decreased in strength, forming a more complex but relatively slower diffusion network topology.

A.6.2 Advanced Fake News Propagation Model

Our fake news propagation simulation incorporated several realistic factors beyond the basic model described in the main text:

Node influence decay: We implemented an influence decay parameter where the strength of information propagation weakened with each subsequent hop through the network. This decay factor (set to 0.85 per hop) mimics the dilution of information credibility as it propagates further from its source.

Propagation thresholds: Each node was assigned an individual threshold for information adoption based on its network characteristics (centrality, clustering coefficient). Nodes with higher centrality typically had lower thresholds, representing that influential entities are more likely to pass along information regardless of its veracity.

Content reliability factors: The propagation simulation incorporated a "content reliability score" that affected both the probability of propagation and the degree of feature disturbance. Less reliable content (lower score) created larger feature disturbances but had lower propagation probabilities, modeling how extreme but less credible information propagates in financial networks.

Counter-information dynamics: In extended simulations, we introduced counter-information sources that could partially neutralize the effect of fake news in their local network neighborhoods. This more realistically modeled how authoritative sources might intervene to limit misinformation spread.

A.6.3 Expanded Temporal Analysis Methods

Our temporal comparison between 2019 and 2022 financial networks incorporated several methodological enhancements:

Network evolution tracking: We analyzed the evolution of key network metrics between 2019 and 2022, including average path length (decreased by 14.3%), clustering coefficient (increased by 8.7%), and degree distribution (showed increased power-law characteristics). These metrics quantified the structural changes in financial networks independent of model performance.

Sensitivity gradient analysis: Rather than using fixed disturbance intensities, we conducted a sensitivity gradient analysis by varying disturbance parameters across a range of values (0.05-0.30 for feature noise, 5%-30% for edge removal). This revealed that 2022 networks exhibited nonlinear sensitivity increases with more pronounced threshold effects than 2019 networks.

Stress scenario combinations: We tested combinations of stressors (e.g., simultaneous feature noise and graph structure change) to identify potential interaction effects. We found that 2022 networks showed stronger negative synergistic effects when exposed to multiple stressors simultaneously, with performance degradation up to 23% greater than would be predicted from individual stressor effects.

Recovery dynamics: We extended our testing to include "recovery phases" after stress scenarios, where we gradually restored the original network structure or feature values over several steps. The 2022 networks showed significantly slower recovery trajectories, suggesting reduced resilience compared to the 2019 networks.

These enhanced analytical methods provided deeper insights into the changing vulnerability characteristics of financial networks following the pandemic shock, revealing not just increased sensitivity but fundamentally altered risk response patterns.

A.6.4 Cascade Network Graph

A.7 Supplementary Note: Exploration of an LLM-Driven Financial Regulatory Question-Answering Agent

While this research focuses on Graph Neural Network (GNN)-based stress testing, we also pre-

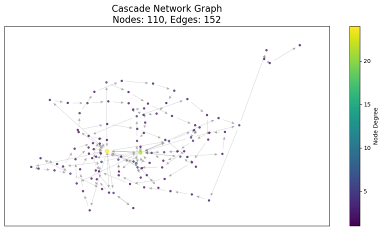


Figure 4: This cascade network diagram is constructed based on user inquiry data provided by Ping An Bank and illustrates the pathways and temporal sequence of information related to "user inquiries" as it spreads among the user group over time. Each node in the diagram represents a unique user ID, extracted as a set of non-redundant identifiers from the 'Usern' column in an Excel spreadsheet. The edges between nodes denote the connections through which information is transmitted from one user to another, established based on the chronological order of inquiries and responses related to the same topic.

liminarily explored the potential of leveraging Large Language Models (LLMs) to assist in financial legal knowledge acquisition. Addressing the limitations of traditional economic law knowledge retrieval in terms of efficiency and cost, we attempted to construct a modular financial law question-answering framework based on Retrieval Augmented Generation (RAG) technology. This framework supports the structured uploading and key-clause extraction from regulatory documents (such as PDF, Excel) to dynamically supplement a specialized knowledge base and update retrieval indices. To enhance the quality and credibility of the answers, the system also incorporates an expert scoring feedback mechanism to calibrate generated content and ensures the auditability of responses through source-tracing technology.

In financial regulatory scenario analysis, we made preliminary attempts to link this framework with GNN models. For example, by analyzing score differences from different question-answering (QA) interactions, we can assist in identifying fake news labels in newly added QAs in the future, providing reference inputs for model training; meanwhile, by parsing regulatory rules across legal systems (e.g., differences in capital adequacy ratio calculations), the framework provides compliance constraint inputs for GNN stress testing. This exploration has preliminarily validated the application potential of LLMs in professional knowledge QA scenarios, where their dynamic policy interpretation and multi-turn interaction capabilities help deepen the semantic understanding in scenario planning.

From an **agent perspective**, the LLM-based QA framework can be conceptualized as a “regulatory knowledge agent” with three core attributes: autonomous knowledge evolution through user-uploaded document updates to mimic human experts’ continuous learning from new regulations, context-aware interaction by dynamically adjusting retrieval weights and answer generation strategies based on specific regulatory scenarios (e.g., cross-legal-system compliance requirements), and collaborative modeling by providing semantic-level constraints (e.g., legal rule embeddings) for GNN nodes to enable hybrid modeling of “structural connectivity + regulatory semantics”.

Future work will focus on optimizing the framework’s processing of unstructured data (e.g., legal case narratives) and deepening its integration with GNN quantitative analysis, aiming to develop a complementary research system of “structural risk simulation + semantic rule parsing” to more effectively address uncertainty challenges in complex financial environments.

This agent-centric work explores how LLMs can act as intelligent components in scenario planning to enhance the depth of regulatory interpretation and the realism of risk modeling.

Enhancing the Planning Capabilities of Large Language Models by Building External World Models

Edwin Chen [1] **Xiaoyan Li** [1] **Colin Bellinger** [2] **Yunli Wang** [2*]
[1] University of Toronto, Toronto, ON, Canada
[2] National Research Council Canada, Ottawa, ON, Canada
`edwinhy.chen@mail.utoronto.ca, xiaoy.li@mail.utoronto.ca`
`Colin.Bellinger@nrc-cnrc.gc.ca, *Yunli.Wang@nrc-cnrc.gc.ca`

Abstract

Large Language Models (LLMs) possess a huge amount of knowledge but struggle with multi-step planning even in toy environments due to the limitations of their static internal world model. We introduce a novel approach where an LLM serves as a "world model builder", constructing and iteratively refining an explicit, external world model. The core of our approach is a state transition function, that is initially generated by the LLM and is refined using feedback from interactions with the environment. This refinement is made possible by accumulating test cases from past experiences allowing us to treat the construction of the world model as a program synthesis problem. We demonstrate the efficacy of our method on the Blocksworld benchmark and introduce a novel ColorMixing dataset that is designed to evaluate multi-step reasoning and planning. Our experimental results show that our method, using GPT-4 and LLaMA3-70B, achieves perfect accuracy on Blocksworld tasks and significantly outperforms baseline methods, especially in terms of planning success and LLM queries. This paper presents a robust methodology for enhancing LLM planning via a learnable external world model and contributes a new benchmark for evaluating such capabilities.[1]

1 Introduction

Large Language Models (LLMs), trained on extensive internet data, have acquired broad commonsense knowledge that enables their application across diverse domains. These models are increasingly employed in critical areas such as medical diagnosis, autonomous driving, chemical experimentation, and intelligent assistance systems. Despite their versatility, reasoning remains a fundamental limitation for LLMs, particularly in complex, multi-step decision-making tasks (Pallagani et al., 2024; Kambhampati et al., 2024). To address this, various prompt-based methods, including Chain-of-Thought (CoT)(Wei et al., 2022), Self-consistency CoT(Wang et al., 2023), Tree of thoughts (Yao et al., 2023a), ReAct (Yao et al., 2023b), and Reflexion (Shinn et al., 2023), have been developed to enhance LLMs' reasoning capabilities. These approaches have demonstrated significant improvements in structured tasks like arithmetic reasoning. However, prompt-based reasoning methods lack the ability to explicitly predict future states, which is essential for effective planning.

LLMs still struggle with tasks requiring multi-step planning or domain-specific knowledge, exhibiting several key limitations (Xiang et al., 2023; Huang et al., 2024). First, they frequently generate plans containing non-existent objects or impermissible actions, as they lack specific knowledge about the target environment. Second, their plans often prove suboptimal due to insufficient understanding of the underlying task mechanisms. Multiple planning benchmarks have revealed limitations in LLMs' performance across diverse problem domains (Valmeekam et al., 2023; Xie et al., 2024).

To enhance both the feasibility and optimality of generated plans, researchers have increasingly adopted world models to capture system dynamics. These models enable the prediction of action outcomes, which can be systematically integrated into the planning process to generate more reliable solutions. This capability is especially crucial for long-horizon decision-making tasks, where world models can be iteratively refined through accumulated experience to adapt to environmental changes.

Some studies utilize pre-existing simulators as world models (Liu et al., 2023), while others leverage LLMs as commonsense world models (Hao

[1] The code for our method and the ColorMixing dataset is available at `https://github.com/edweenie123/WorldModelBuilder`

et al., 2023; Zhao et al., 2023) or construct the world model (Guan et al., 2023).

Inspired by using LLM as the world model (Hao et al., 2023), we propose a novel approach that learns an external world model from LLM interaction trajectories. This model encodes past experiences as reusable functions, enabling more efficient planning. The framework specifically learns state transition dynamics and action prediction mechanisms from historical interactions. Through progressive refinement from simple to complex scenarios, the world model mimics human-like learning and adaptation in novel environments.

This work makes two key contributions: First, we develop and validate an effective world model learning methodology, demonstrating its performance on the established Blocksworld benchmark. Second, we introduce a new ColorMixing dataset specifically designed to evaluate LLM planning capabilities in complex, multi-step scenarios.

Our method can be used for scenario planning in hospital resource management, as well as other real-world scenario planning problems. Scenario planning involves creating a set of plausible but distinct future “scenarios” based on key uncertainties, trends, and drivers of change. Our approach takes these distinct scenarios as initial conditions and goals and uses the dynamics model to develop plans to achieve these goals.

2 Related work

LLM-based planning systems face three core challenges: grounding, plan generation, and adaptability. For grounding, agents utilize the LLM’s inherent commonsense knowledge (Huang et al., 2022) to bridge abstract concepts with environmental specifics. In plan generation, LLMs typically function as policy networks that propose contextually appropriate next actions (Hao et al., 2023; Zhao et al., 2023). Planning can rely on the inherent reasoning capabilities of LLMs (Krishna et al., 2023) or enhance these abilities by combining ReAct and Reflexion prompting while retrieving relevant examples from memory (Zhao et al., 2024). To improve planning efficacy, researchers often employ LLMs as world models for state prediction (Hao et al., 2023) and integrate Monte Carlo Tree Search (MCTS) to efficiently explore large action spaces (Zhao et al., 2023; Zhou et al., 2024), while skill transfer from past experiences helps reduce computational complexity (Wang et al., 2024; Sun et al., 2023). The system’s adaptability emerges through continuous plan refinement based on environmental feedback (Sun et al., 2023; Zhou et al., 2024).

The importance of world models in planning tasks has been recognized in various studies. For instance, Mind’s Eye (Liu et al., 2023) employs a simulator as its world model, while RAP (Hao et al., 2023) leverages the world model in LLMs for simple reasoning tasks. When presented with a physical reasoning question, Mind’s Eye (Liu et al., 2023) employs a computational physics engine (e.g., DeepMind’s MuJoCo) to simulate potential outcomes. These simulation results are then integrated into the input, enabling language models to perform more accurate and grounded reasoning.

World models should possess the ability to plan, predict, and reason effectively about physical scenarios. LLM-DM (Guan et al., 2023) constructs an explicit world (domain) model using Planning Domain Definition Language (PDDL), a formal language for representing planning problems. Language models are primarily employed to translate natural language into PDDL, while domain experts provide feedback to refine the PDDL construction.

LLM-MCTS leverages the commonsense knowledge of LLMs to reduce the search space in large-scale task planning (Zhao et al., 2023). It treats the LLM as a commonsense world model to provide prior belief, which is updated with each action and observation in the real world. During tree search, LLM-MCTS heuristically selects promising action branches by querying the LLM. MCTS samples from the belief state (probability distribution over states) to estimate the value of actions.

RAP (Hao et al., 2023) framework employs LLMs’ internal world models for reasoning and planning across diverse tasks. As a gray-box approach, RAP analyzes token-level probabilities, along with state confidence and self-evaluation heuristics, to guide the planning process. However, this method requires frequent LLM queries, resulting in computational inefficiency and high costs, particularly for proprietary models.

Unlike RAP and LLM-MCTS, our approach learns an external world model from past experiences. This model predicts future states and evaluates actions to enable efficient planning. While prior work has explored using LLMs to learn and refine functions, such as learning continuous functions for symbolic regression (Merler et al., 2024), our work focuses on learning a transition function for discrete multi-step decision-making.

3 Method

Our proposed method enhances the planning capabilities of LLMs by using the LLM as a “world model builder”. This approach extends concepts from frameworks such as RAP (Hao et al., 2023), but with a key distinction: instead of utilizing the LLM’s static internal world model, our method focuses on building an external model through LLM-driven generation and refining it over time with environmental interactions. Figure 1 provides an overview of our approach, illustrating the key components and workflow of the system. The figure depicts our three-stage process: first, the LLM generates an initial state transition function based on task descriptions; second, this function is iteratively refined through real-world interactions and feedback; and finally, the refined world model enables efficient planning by predicting future states without requiring additional LLM queries during execution.

3.1 World Model Architecture

The world model consists three key components.

1. **State Transition Function (f_{ST}):** This function takes the current state s_t and an action a_t as input, and predicts the next state $\hat{s}_{t+1} = f_{ST}(s_t, a_t)$. Initially, the LLM is prompted to generate this function, for instance, as a Python program given a rough description of the environment and the possible actions within the environment. This function is the main subject of the iterative refinement process detailed in Section 3.3.

2. **State Value Function (f_{SV}):** This function estimates the utility or value of a given state s with respect to a user-defined goal. It outputs a scalar value $v(s) = f_{SV}(s)$, which guides the search process towards desirable states. In the current method, f_{SV} is implemented as a hard-coded heuristic tailored to the specific task domain and goal structure.

3. **Action Suggestion Function (f_{AS}):** Given a state s, this function suggests a set of promising actions $A_p(s) = f_{AS}(s)$ that are worth exploring. This helps to prune the search space by focusing on relevant actions. Similar to f_{SV}, f_{AS} is hard-coded based on domain knowledge to identify potentially useful actions.

The user provides the initial state and the goal. The core innovation of our method lies in the LLM-driven generation and subsequent experience-based iterative refinement of f_{ST}. While f_{SV} and f_{AS} are presently fixed, their design is crucial for effective planning.

3.2 Planning with the Learned World Model

Once the components of the world model are established, and an overall goal is defined for the task, the planning process proceeds as follows:

1. From the current state s_{curr}, the Action Suggestion Function $f_{AS}(s_{curr})$ is invoked to generate a set of promising actions $A_p(s_{curr})$.

2. For each action $a \in A_p(s_{curr})$, the State Transition Function $f_{ST}(s_{curr}, a)$ is used to predict the resulting next state $\hat{s}'$.

3. This process is applied recursively to construct a search tree, where nodes represent states and edges represent actions.

4. The State Value Function $f_{SV}(\hat{s}')$ is used to evaluate the desirability of states encountered in the search tree (like $\hat{s}'$), particularly leaf nodes or states at a certain depth, in relation to the overall goal.

5. A search algorithm (e.g., Depth-First Search (DFS), Monte Carlo Tree Search (MCTS)) traverses this tree to identify an action sequence $(a_0, a_1, \ldots, a_k)$ that is expected to lead to a state with the highest value or achieve the goal.

This planning mechanism relies on the explicit world model functions, allowing for systematic exploration of future possibilities towards the given goal. Leveraging the learned world model, our method uses a search algorithm to explore multiple world branches and estimate the value of a sequence of actions.

3.3 Iterative Refinement of the State Transition Function

A key aspect of our method is the continuous improvement of the State Transition Function (f_{ST}) based on experiences gathered from interacting with the real environment. This process treats the

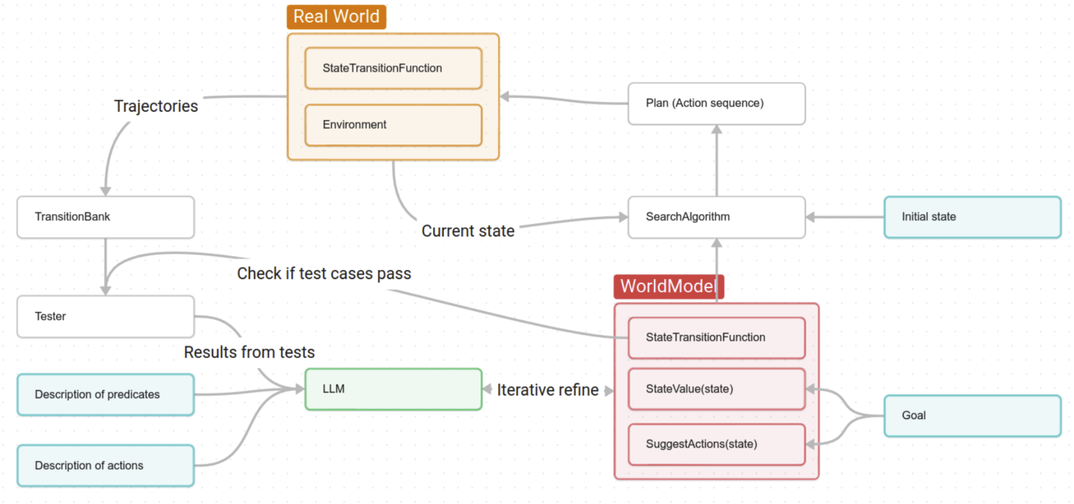


Figure 1: Overview of our proposed method for enhancing LLM planning capabilities with an external world model. The approach consists of three main components: (1) LLM-driven generation of a state transition function, (2) iterative refinement based on environmental interactions, and (3) efficient planning using the learned world model. This framework enables more accurate multi-step reasoning while reducing computational costs compared to approaches that rely solely on LLM queries for state prediction.

generation of f_{ST} as an iterative program synthesis problem.

1. **Experience Collection:** The agent executes an action (e.g., the first action a_0 from the generated plan) in the real environment. The environment then transitions from state s_t to a true subsequent state s_{t+1} according to its ground-truth dynamics. This interaction yields an experience tuple $((s_t, a_t), s_{t+1})$, which serves as a test case for f_{ST}. These test cases are accumulated in a "transition bank".

2. **Evaluation:** The current f_{ST} is evaluated against all test cases stored in the transition bank. A test case $((s_t, a_t), s_{t+1})$ is considered a failure if the predicted next state $\hat{s}_{t+1} = f_{ST}(s_t, a_t)$ does not match the observed next state s_{t+1}, or if the actual next state s_{t+1} is sufficiently different from the predicted next state $\hat{s}_{t+1}$ according to some user-defined state similarity metric.

3. **LLM-based Refinement:** The set of failing test cases (i.e., input-output pairs that f_{ST} incorrectly predicted) is provided as feedback to the LLM. The LLM is then prompted to revise or debug the f_{ST} (e.g., its Python code implementation) to correctly handle these failing instances, while ideally preserving its accuracy on previously successful cases.

4. **Iteration:** The refined f_{ST} is then re-evaluated against the transition bank. This cycle of evaluation and LLM-based refinement is repeated, progressively improving the accuracy of f_{ST}. The process can continue until all test cases pass, a predefined accuracy threshold is met, or a computational budget (e.g., number of LLM queries) is exhausted.

Through this iterative loop, the f_{ST} becomes an increasingly accurate approximation of the real world's state transition dynamics.

Our world model refinement loop is compatible with both deterministic and probabilistic transition rules. It can iteratively query the LLM, validate predicted outcomes against examples, and revise the rule as needed. This flexibility makes our approach directly applicable to planning under uncertainty—not just in fully deterministic settings.

3.4 Addressing Limitations of Existing Approaches

Our proposed methodology directly addresses several limitations observed in prior LLM-based planning approaches, such as RAP:

1. **Performance Improvement with Experience:** Unlike systems where the LLM's internal world model remains static, our approach allows the explicit f_{ST} to be continuously refined and improved as more interaction data is collected. This enables the agent's planning accuracy to increase with experience, mimicking a crucial aspect of human learning.

2. **Reduced LLM Query Cost during Planning:** In RAP, generating the search tree often requires querying the LLM at each state to predict outcomes of actions. Our method shifts the primary LLM usage to the initial generation and subsequent off-line refinement of the f_{ST}. Once f_{ST} is learned (e.g., as an executable Python function), it can be called repeatedly during the planning phase (Section 3.2) without incurring additional LLM query costs for each state transition prediction. This significantly reduces the computational expense and latency associated with LLM queries during the search process, making deeper and broader searches more feasible and aligning with the objective of maximizing performance while minimizing LLM interactions.

By externalizing and refining the world model, particularly the state transition dynamics, our method aims to achieve more robust, adaptable, and efficient planning with LLMs.

4 Experiments

We evaluate our method and the baselines on two datasets: the classic Blocksworld benchmark (Valmeekam et al., 2023), widely used for goal-conditioned symbolic planning and reasoning tasks, and our own ColorMixing dataset.

4.1 Blocksworld

Blocksworld is a classic symbolic planning domain involving a set of colored blocks stacked on a table. The agent's goal is to transform an initial block configuration into a specified goal configuration using a sequence of primitive actions such as pick up, put down, stack, and unstack. We employ subsets of this benchmark, specifically 30% from three-step problems (step 2, step 4, and step 6), to train our world model, reserving the remaining 70% for testing. In Blocksworld, the state encodes the configuration of blocks (e.g., `on`, `clear`, `ontable`), and actions include the standard operations: `pick-up`, `put-down`, `stack`, and `unstack`.

4.2 ColorMixing

The *ColorMixing Dataset* is a synthetic benchmark developed to assess the reasoning and planning capabilities of LLMs in a controlled color mixing environment. In this setting, an agent interacts with six virtual beakers, each described by color contents and volume, and is tasked with achieving a specified target color in a chosen beaker through a sequence of actions. The initial state of the environment includes five beakers prefilled with primary and neutral colors: red, green, blue, white, and black, and one empty beaker designated for mixing. We use a discrete action space composed of symbolic operations with arguments defined by integer values (e.g., beaker indices and paint amounts). Figure 2 illustrates the color mixing process. Each state is represented as a list of strings in the form: `contains <beaker_id> <R> <G> <B> <amount>`, where `<beaker_id>` is an integer from 1 to 6, `<R>`, `<G>`, and `<B>` are RGB values ranging from 0 to 255, and `<amount>` denotes the volume (0 - 200). The goal state, also in this format, specifies the desired color mixture and amount in a target beaker. The ColorMixing environment enables evaluation of both low-level world model predictions and high-level planning behavior.

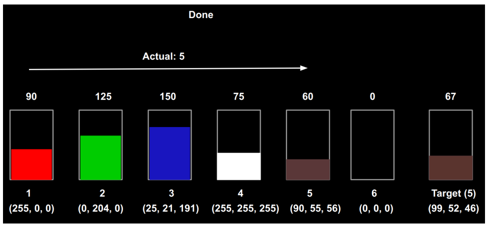


Figure 2: Visualization of the color mixing task. The initial state includes five prefilled beakers and one empty beaker. The agent aims to produce a target color in a designated beaker through sequential actions.

In the ColorMixing task, each state represents the color and volume of paint in six beakers. Actions include operations: `pour` and `done`. We generated 100 data files, each containing the initial color states of six beakers along with a corresponding target (goal) color state. We randomly selected 30% of the files as training data, and used the remaining 70% as test cases.

While the ColorMixing environment is deterministic at the transition level, it introduces uncertainty in the number of actions required to reach the goal. The agent must explore and compare action sequences of varying lengths, reflecting a form of procedural uncertainty that aligns closely with the goals of scenario planning.

4.3 Results on Blocksworld

The following subsections present the results of our method on the Blocksworld domain, focusing on both the training (refinement) phase of the world model and the task performance on the testing data.

4.3.1 Training Phase

The quality of the refined world model is evaluated using two metrics: experience accuracy, measured by the pass rate across individual state-action transitions in the transition bank, and state transition accuracy, which assesses the model's ability to simulate full state trajectories given sequences of actions from the test set. List 1 in Appendix A shows an example of a state transition function for Blocksworld refined by GPT-4.

Figure 3 illustrates the training progression of the world model in our method, refined using GPT-3.5 on the Blocksworld. The top subplot depicts the number of LLM queries made per training instance. Each dot corresponds to a specific training level, with red markers indicating instances where the algorithm failed to achieve the goal state. Notably, many levels required up to 15 GPT-3.5 queries to refine the state transition function, highlighting the limited capability of GPT-3.5 in generating accurate state transition functions. The middle subplot shows the accuracy of the world model's learned state transition function f_{ST}, measured by comparing the predicted next state to the ground-truth next state for each (s_t, a_t) in the transition bank, during training. This metric reflects the model's internal learning quality across training iterations. A sharp improvement is observed around the 16th training level, after which the accuracy plateaus, indicating that further refinement yields diminishing returns. Despite continued LLM queries, GPT-3.5 fails to consistently improve the quality of the learned transition function beyond this point.

The bottom subplot shows state transition accuracy on the held-out test set. Given an initial state and ground-truth action sequence, the learned world model predicts the resulting state after each action, and accuracy is computed as the average similarity across all predicted and ground-truth next states. Notably, goal achievement and transition accuracy may diverge, as goal completion is based on a partial specification (e.g., "on a c" and "on b a"), while state transition accuracy evaluates the entire predicted state, including predicates like "handempty", "clear d", "ontable d", and others. Thus, a goal may be achieved even if the overall state prediction accuracy is relatively low.

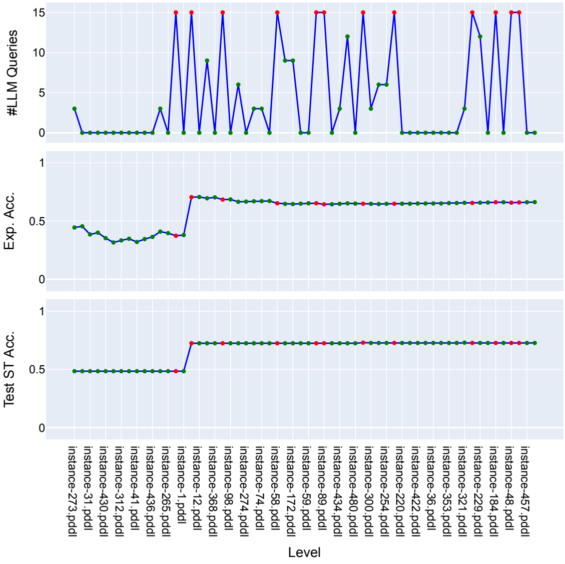


Figure 3: Training progression of the world model in our method, refined using GPT-3.5, on the Blocksworld. The x-axis denotes the training instances (levels).

We observe that GPT-3.5 demonstrates some reasoning capability in refining the world model. However, even after multiple refinement steps, the state transition accuracy remains below 0.8, indicating its limited effectiveness in learning accurate transition dynamics. Figure 5 in Appendix B presents the training progression of the world model in our method, refined using GPT-4. Compared to the case with GPT-3.5, GPT-4 demonstrates significantly stronger reasoning capabilities. Notably, the world model is successfully refined with only a single LLM query (specifically, updating the state transition function). After this refinement, the model consistently predicts accurate next states when paired with the search algorithm.

Figure 6 in Appendix B illustrates the training progression of the world model when refined using LLaMA3-70B. In this case, the model is refined based on feedback from only two training instances, requiring a total of 9 LLM queries. Compared to GPT-4, LLaMA3-70B achieves lower accuracy in modeling the state transition function, but it still outperforms GPT-3.5 in both experience and test accuracy.

4.3.2 Testing Phase

For overall task performance, we report the goal achievement accuracy, defined as the ratio of successful instances (i.e., the final state matches the goal) to the total number of test instances. Our method was evaluated on the testing set, consisting of 70% of the instances from each step. Table 1 presents a comparison between our approach and several baselines: GPT-3.5 + CoT, GPT-4 + CoT, and RAP (Hao et al., 2023). Since RAP currently supports only LLaMA-based models, we use LLaMA3-70B as the backbone for the RAP baseline. All baseline implementations are obtained from the LLM Reasoners benchmark (Hao et al., 2024). Following the evaluation protocol used for GPT-3.5 + CoT, GPT-4 + CoT, and RAP, we use the VAL tool, a command-line validator for checking the correctness of plans in classical PDDL-based planning domains (Fox and Long, 2003), to assess each method's performance.

As shown in Table 1, all three variants of our method consistently outperform the two CoT-based GPT baselines across all steps. The world model refined with GPT-3.5 achieves lower accuracy, while those refined using GPT-4 and LLaMA3-70B reach perfect accuracy, surpassing RAP. Although the CoT-based baselines are less accurate, they require only a single LLM query. In contrast, RAP issues two LLM queries per candidate action, one for action generation and one for next-state prediction, resulting in approximately $N \times d \times 2 = 80$ LLaMA3 queries for $N = 10$ rollouts and $d = 4$ actions. Our method requires only 9 LLaMA3 queries for refinement, as shown in Figure 6 in Appendix B. Notably, RAP cannot support GPT-based models due to its reliance on token-level log probabilities, which are not accessible via the OpenAI API, making direct comparison infeasible.

Method	**Accuracy**			
	Step 2	**Step 4**	**Step 6**	**Avg.**
GPT-3.5 + CoT	20.00%	13.50%	4.69%	12.73%
GPT-4 + CoT	20.01%	14.50%	4.94%	13.15%
RAP (LLaMA3)	89.47%	85.00%	80%	84.49%
Ours (GPT-3.5)	95.24%	87.5%	78.75%	87.16%
Ours (GPT-4)	100%	100%	100%	100%
Ours (LLaMA3)	100%	100%	100%	100%

Table 1: Performance comparison across Step 2, Step 4, and Step 6 tasks for our three methods and three baselines.

4.3.3 Runtime Analysis

Table 2 compares training and inference times of our methods against several baselines. All experiments were conducted on a machine equipped with an Intel Xeon W-2255 CPU (10 cores, 20 threads, 3.70 GHz) and an NVIDIA Quadro RTX 6000 GPU with 24 GB of memory. Among our variants, the world model refined using GPT-4 achieves the lowest overall runtime, requiring only 38.57 seconds for training and 1.18 seconds for inference. In contrast, the LLaMA3-70B variant incurs the highest training time (3084.02 seconds) due to its local execution, but maintains a comparable inference time of 1.28 seconds. Despite the higher upfront cost, our method with LLaMA3-70B significantly outperforms overall efficiency of RAP. This efficiency stems from our approach refining the world model using LLaMA3-70B only during the training phase, after which a symbolic DFS planner is used at test time. In contrast, RAP repeatedly queries LLaMA3-70B during planning, resulting in substantially higher cumulative runtime.

Method	**Training Time (s)**	**Inference Time (s)**
GPT-3.5 + CoT†	–	593.8
GPT-4 + CoT†	–	586.6
RAP (LLaMA3)	–	1,518,120
Ours (GPT-3.5)†	675.97	1.17
Ours (GPT-4)†	38.57	1.18
Ours (LLaMA3)	3084.02	1.28

Table 2: Comparison of training and inference times (in seconds) for different methods. Training time refers to the refinement of the world model, while inference time measures the execution time of the model (or refined model) for task-solving. †Inference for API-based models (GPT-3.5 and GPT-4) includes network latency and remote GPU processing.

4.4 Results on ColorMixing

In the ColorMixing benchmark, we use a similarity score to measure the closeness between a predicted state and its corresponding ground-truth state. Similarity is computed by comparing corresponding beakers based on both RGB color and paint volume. Specifically, we define a weighted similarity function that combines color similarity, measured by the Euclidean distance in RGB space, and volume similarity, measured by the normalized absolute difference. A higher weight is assigned to the color component. The overall similarity between two states is computed as the average of the beaker-wise similarities. The world model is considered sufficiently accurate if the similarity between the predicted final state and the goal state exceeds a threshold of 0.95.

4.4.1 Training Phase

We evaluate the refinement quality of the world model using three metrics: experience similarity score, state transition similarity score, and goal state similarity score. These metrics respectively, assess the model's accuracy on training state transitions, its ability to generalize to unseen action sequences, and its alignment with the desired goals. List 2 in Appendix A presents an example of a state transition function for ColorMixing, refined using GPT-4.

In the ColorMixing experiments, we use GPT-4 to refine the world model and compare our method with the GPT-4 + CoT baseline. Figure 4 presents the training progression of our approach. Similar to the Blocksworld, the top subplot shows the number of LLM queries required to refine the world model. The second subplot presents the average similarity score across all state transitions observed during training episodes, reflecting how accurately the model predicts intermediate states. A predicted color is considered a successful match if its similarity exceeds a threshold of 0.95. Matched cases are colored green, while failed instances are red.

The third subplot captures the testing performance of the learned state transition function. Unlike Blocksworld, where action sequences are fixed, we randomly sample a state–action pair and compare the predicted next state with the ground-truth. This simulates a realistic setting where the number of mixing steps is not predefined and must be inferred by the model. In both training and testing evaluations, the state transition function achieves an average similarity score above 0.97, indicating strong predictive accuracy. The fourth subplot shows the number of steps taken to reach the goal. Notably, although the average state similarity during testing is above the threshold (0.95), there are cases where the algorithm still fails to achieve the goal. This discrepancy is highlighted in the bottom subplot, which shows the final goal similarity score, computed based solely on the target beaker.

The mismatch arises because the similarity score for the state transition reflects the average similarity across all six beakers, while goal achievement is determined only by the similarity of the target beaker. Thus, if the target beaker's similarity falls below the threshold, the episode is marked as a failure, even if the average similarity across all beakers remains high.

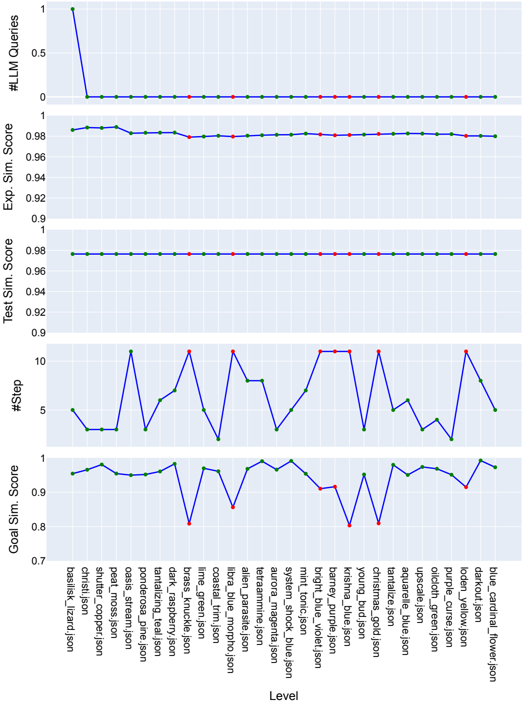


Figure 4: Training progression of the world model in our method, refined using GPT-4, on the ColorMixing Data.

4.4.2 Testing Phase

We evaluate the task performance based on two metrics: the average goal state similarity score and the pass rate, defined as the ratio of instances where the goal similarity score exceeds a predefined threshold to the total number of instances.

Table 3 presents the testing results of our method compared to the GPT-4 + CoT baseline. Due to the increased complexity of the ColorMixing task relative to Blocksworld, both GPT-3.5 and LLaMA3-70B fail to produce reliable state transition functions. Additionally, RAP cannot be used as a baseline in this setting, as it currently supports only LLaMA-based LLMs. Another potential baseline involves using the LLM's internal world model to simulate state transitions directly; however, this approach is prohibitively expensive, as it requires repeated LLM queries for each action and state transition, resulting in substantial computational overhead. For this reason, we exclude it from our evaluation. Consequently, we exclusively employ GPT-4 for refining the world model in this setting. The results in Table 3 highlight the superior performance of our approach, which achieves a perfect pass rate across all test instances.

Method	Goal Similarity Score	Pass Rate
GPT-4 + CoT	53.45%	0%
Ours (GPT-4)	98.10%	100%

Table 3: Comparison between GPT-4 + CoT and our method using GPT-4 on the ColorMixing task.

5 Conclusion

LLMs have shown promise as policies for complex decision-making tasks, but their effectiveness is limited by inaccuracies in their internal world models, leading to inefficient planning. To address this, we propose learning an external world model that dynamically improves multi-step reasoning by predicting future states at each decision point. Our experiments on Blocksworld and ColorMixing demonstrate significant improvements, achieving perfect success rates across all difficulty levels in Blocksworld while outperforming LLM-based world models.

Our work introduces both a novel approach to enhancing LLM-based planning and a new dataset for evaluating multi-step decision-making tasks. However, our experiments are currently focused on the Blocksworld and ColorMixing datasets. Further evaluation on more diverse and complex environments, such as VirtualHome or GSM8k, is necessary to fully assess the generalizability and effectiveness of our method. In future work, we plan to integrate our external world model with more efficient search algorithms to better handle tasks with large and complex action spaces.

Our method involves learning a dynamics model using the LLM and then utilizing the model to plan. It can be used for human-led scenario planning. Here, a human devises plausible but uncertain future scenarios, such as a shortage of gold flake paint or an increase in the cost of gold flake paint due to commodity fluctuations. This leads to a different set of initial conditions, which can be explored using our model-based planner. In this case, our method is used as a simulator for scenario planning under uncertainty. Since the world model is learned through interaction with the environment, the simulator can adapt to different initial conditions.

Limitations

Our approach, while delivering promising results, has several limitations that offers avenues for future work. Currently, the LLM is only responsible for generating and refining the State Transition Function (f_{ST}), while the State Value (f_{SV}) and Action Suggestion (f_{AS}) functions are hard coded. Extending the LLM's involvement to also learn these components of the world model would enhance autonomy and may improve performance. Moreover, the current system only plans on a predefined set of low-level actions; future work could explore enabling the LLM to learn higher-order actions consisting of several low-level actions allowing for hierarchical planning and potentially improving planning ability in complex environments.

Acknowledgments

This project is supported by research funding from the National Research Council of Canada's Artificial Intelligence for Design Program.

References

Maria Fox and Derek Long. 2003. PDDL2.1: An extension to PDDL for expressing temporal planning domains. *Journal of Artificial Intelligence Research*, 20:61–124.

Lin Guan, Karthik Valmeekam, Sarath Sreedharan, and Subbarao Kambhampati. 2023. Leveraging pretrained large language models to construct and utilize world models for model-based task planning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Shibo Hao, Yi Gu, Haotian Luo, Tianyang Liu, Xiyan Shao, Xinyuan Wang, Shuhua Xie, Haodi Ma, Adithya Samavedhi, Qiyue Gao, Zhen Wang, and Zhiting Hu. 2024. LLM reasoners: New evaluation, library, and analysis of step-by-step reasoning with large language models. In *First Conference on Language Modeling*.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. 2023. Reasoning with language model is planning with world model. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. 2022. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International conference on machine learning*, pages 9118–9147. PMLR.

Xu Huang, Weiwen Liu, Xiaolong Chen, Xingmei Wang, Hao Wang, Defu Lian, Yasheng Wang, Ruiming Tang, and Enhong Chen. 2024. Understanding the planning of LLM agents: A survey. *arXiv preprint arXiv:2402.02716*.

Subbarao Kambhampati, Karthik Valmeekam, Lin Guan, Mudit Verma, Kaya Stechly, Siddhant Bhambri, Lucas Paul Saldyt, and Anil B Murthy. 2024.

Position: LLMs can't plan, but can help planning in LLM-modulo frameworks. In *Forty-first International Conference on Machine Learning*.

Satyapriya Krishna, Jiaqi Ma, Dylan Slack, Asma Ghandeharioun, Sameer Singh, and Himabindu Lakkaraju. 2023. Post hoc explanations of language models can improve language models. *Advances in Neural Information Processing Systems*, 36:65468–65483.

Ruibo Liu, Jason Wei, Shixiang Shane Gu, Te-Yen Wu, Soroush Vosoughi, Claire Cui, Denny Zhou, and Andrew M. Dai. 2023. Mind's eye: Grounded language model reasoning through simulation. In *The Eleventh International Conference on Learning Representations*.

Matteo Merler, Katsiaryna Haitsiukevich, Nicola Dainese, and Pekka Marttinen. 2024. In-context symbolic regression: Leveraging large language models for function discovery. *arXiv preprint arXiv:2404.19094*.

Vishal Pallagani, Bharath Chandra Muppasani, Kaushik Roy, Francesco Fabiano, Andrea Loreggia, Keerthiram Murugesan, Biplav Srivastava, Francesca Rossi, Lior Horesh, and Amit Sheth. 2024. On the prospects of incorporating large language models (LLMs) in automated planning and scheduling (APS). In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 34, pages 432–444.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik R Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Haotian Sun, Yuchen Zhuang, Lingkai Kong, Bo Dai, and Chao Zhang. 2023. Adaplanner: Adaptive planning from feedback with language models. In *Advances in Neural Information Processing Systems*, volume 36, pages 58202–58245. Curran Associates, Inc.

Karthik Valmeekam, Matthew Marquez, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. 2023. Planbench: An extensible benchmark for evaluating large language models on planning and reasoning about change. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2024. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Jiannan Xiang, Tianhua Tao, Yi Gu, Tianmin Shu, Zirui Wang, Zichao Yang, and Zhiting Hu. 2023. Language models meet world models: Embodied experiences enhance language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Jian Xie, Kai Zhang, Jiangjie Chen, Tinghui Zhu, Renze Lou, Yuandong Tian, Yanghua Xiao, and Yu Su. 2024. Travelplanner: a benchmark for real-world planning with language agents. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023a. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023b. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. 2024. Expel: LLM agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19632–19642.

Zirui Zhao, Wee Sun Lee, and David Hsu. 2023. Large language models as commonsense knowledge for large-scale task planning. *Advances in Neural Information Processing Systems*, 36:31967–31987.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. 2024. Language agent tree search unifies reasoning, acting, and planning in language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

A Refined State Transition

A.1 State Transition for Blocksworld

Listing 1: Refined state transition function for the Blocksworld domain refined using GPT-4.

```
def state_transition(self, state, action
    ):
    words = action.split()
    action_type = words[0]
    params = words[1:]
    next_state = set(state)

    if action_type == "pick-up":
        block = params[0]
        if f"clear {block}" in
    next_state and f"ontable {block}" in
     next_state and "handempty" in
    next_state:
            next_state.discard(f"clear {
    block}")
            next_state.discard(f"ontable
     {block}")
            next_state.discard("
    handempty")
            next_state.add(f"holding {
    block}")

    elif action_type == "put-down":
        block = params[0]
        if f"holding {block}" in
    next_state:
            next_state.discard(f"holding
     {block}")
            next_state.add(f"ontable {
    block}")
            next_state.add(f"clear {
    block}")
            next_state.add("handempty")

    elif action_type == "stack":
        block, target = params
        if f"holding {block}" in
    next_state and f"clear {target}" in
    next_state:
            next_state.discard(f"holding
     {block}")
            next_state.discard(f"clear {
    target}")
            next_state.add(f"on {block}
    {target}")
            next_state.add(f"clear {
    block}")
            next_state.add("handempty")

    elif action_type == "unstack":
        block, base = params
        if f"on {block} {base}" in
    next_state and f"clear {block}" in
    next_state and "handempty" in
    next_state:
            next_state.discard(f"on {
    block} {base}")
            next_state.discard(f"clear {
    block}")
            next_state.discard("
    handempty")
            next_state.add(f"holding {
    block}")
            next_state.add(f"clear {base
    }")

    return next_state
```

A.2 State Transition for ColorMixing

Listing 2: State transition function for the ColorMixing environment refined using GPT-4.

```
def state_transition(self, state, action
    ):
    def find_element(my_set, condition):
        for element in my_set:
            if condition(element):
                return element
        return None

    words = action.split()
    action_type = words[0]
    params = words[1:]

    new_state = set(state)

    if action_type == "pour":
        src_idx, tgt_idx, amt = [int(x)
    for x in params]
        src_contains = find_element(
    state, lambda x: x.split()[1] == str
    (src_idx))
        tgt_contains = find_element(
    state, lambda x: x.split()[1] == str
    (tgt_idx))

        src_r, src_g, src_b, src_amt = [
    int(x) for x in src_contains.split()
    [2:]]
        tgt_r, tgt_g, tgt_b, tgt_amt = [
    int(x) for x in tgt_contains.split()
    [2:]]

        # Calculate the amount of paint
    after pouring
        new_src_amt = src_amt - amt
        new_tgt_amt = tgt_amt + amt

        # Calculate the new color in the
     target beaker
        new_tgt_r = (tgt_r * tgt_amt +
    src_r * amt) // new_tgt_amt
        new_tgt_g = (tgt_g * tgt_amt +
    src_g * amt) // new_tgt_amt
        new_tgt_b = (tgt_b * tgt_amt +
    src_b * amt) // new_tgt_amt

        # Update the state with the new
    values
        new_state.discard(src_contains)
        new_state.discard(tgt_contains)

        new_state.add(f"contains {
    src_idx} {src_r} {src_g} {src_b} {
    new_src_amt}")
        new_state.add(f"contains {
    tgt_idx} {new_tgt_r} {new_tgt_g} {
    new_tgt_b} {new_tgt_amt}")

    return new_state
```

B Training Performance on Blocksworld

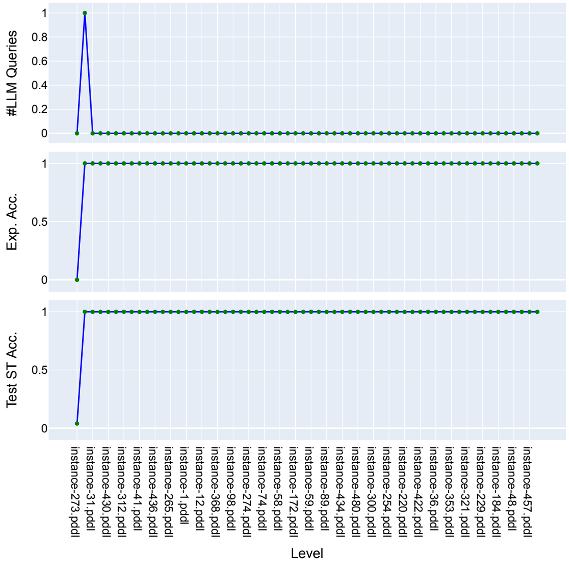


Figure 5: Training progression of the world model in our method, refined using GPT-4, on the Blocksworld domain.

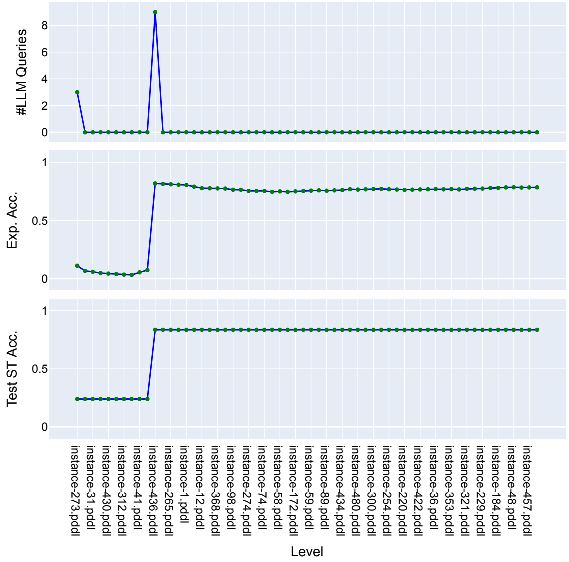


Figure 6: Training progression of the world model in the Blocksworld domain, refined using LLaMA3-70B. The plot illustrates how the model improves over training iterations.

Enhancing Naphtha Cracking Center Scheduling via Population-Based Multi-Scenario Planning

Deunsol Yoon[*], Sunghoon Hong[*], Whiyoung Jung[*], Kanghoon Lee, Woohyung Lim
LG AI Research
Seoul, South Korea
{dsyoon, sunghoon.hong, whiyoung.jung, kanghoon.lee, w.lim}@lgresearch.ai

Abstract

Naphtha Cracking Center scheduling aims to develop optimal multi-week plans under operational constraints and fluctuating demand. Our prior work (Hong et al., 2024b) introduced a multi-agent reinforcement learning (RL) system that is currently deployed in a petrochemical plant. However, standalone RL agents face several limitations: the environment is sensitive—one suboptimal action can invalidate the entire plan—and reward functions are often difficult to specify. We propose Population-Based Multi-Scenario Planning (PBMSP), a novel planning algorithm designed to complement RL agents. PBMSP maintains a diverse set of candidate schedules optimized for distinct objectives and constraints, and extends RL-based scheduling by enhancing adaptability, stability, and operational profitability.

1 Introduction

Scheduling in Naphtha Cracking Centers (NCCs) presents a fundamental challenge in petrochemical manufacturing. It involves planning a continuous, multi-stage process that converts raw naphtha into high-value products, primarily ethylene. This process includes three interdependent stages: 1) **unloading** naphtha from vessels into receipt tanks, 2) **blending** selected receipt tanks in the blending tank to achieve the target composition, and 3) **cracking** the blended feed in furnaces (LG, 2024).

The strong coupling among these stages, along with operational constraints and fluctuating demand, necessitates robust long-term scheduling. Effective plans must consider plant status, vessel arrival schedules, tank capacities, feedstock quality, and external factors such as market conditions. Based on advance shipment data, operators prepare multi-week schedules, illustrated in Figure 1, that specify which tanks receive incoming naphtha, how blending is performed, and furnace settings like feed flow rates and coil outlet temperatures. These schedules are crucial for maintaining safe, stable, and efficient operations under uncertainty.

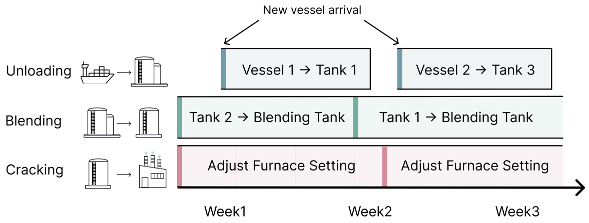


Figure 1: Simplified representation of an NCC schedule.

Previous research on NCC scheduling has primarily addressed individual process components in isolation (Lee et al., 2010; Lee, 2012; Joo et al., 2023; Kim et al., 2023). Our prior work (Hong et al., 2024b)—a demonstration paper highlighting the system architecture and web-based interface[1] of its reinforcement learning (RL)-based scheduling system deployed at a petrochemical plant—proposed a cooperative multi-agent system (MAS) framework integrating the three interdependent stages of the NCC process into a unified scheduling model. In this framework, agents are assigned to manage the unloading, blending, and cracking stages respectively, and generate production plans collaboratively.

This MAS framework inherently creates operational asynchronicity, with agent actions having varied start times and durations. For example, unloading actions commence with non-periodic vessel arrivals and their durations depend on shipment volumes, while durations of blending actions vary based on receipt tank inventories. To manage it, our prior work (Hong et al., 2024b) employed the MacDec-POMDP framework (Amato et al., 2019; Xiao et al., 2022; Hong et al., 2024a; Jung et al., 2025). This framework is designed for modeling

[*] Equal contribution.

[1] A web-based demonstration in our earlier work is at https://www.youtube.com/watch?v=TxoWG7_SLLU.

multi-agent decision-making with asynchronicity by defining macro-actions (sequences of predefined micro-actions over multiple time steps). This representation naturally accommodates the varied start times and durations of actions inherent in the NCC.

Building on our prior work, this paper proposes a complementary planning algorithm to enhance the practical usability, robustness, and adaptability of RL-based scheduling. While our deployed multi-agent RL system demonstrates promising performance, real-world implementation reveals several challenges that limit its standalone effectiveness.

First, the NCC scheduling environment is inherently sensitive. A single suboptimal action—even one that may appear minor—can invalidate an entire schedule, eventually leading to operational failure. For instance, failing to initiate blending on time may cause receipt tank overflows, while improper blending may result in off-specification feedstock and downstream disruptions. Such fragility makes it difficult for RL agents alone to consistently produce valid and safe schedules without additional safeguards.

Second, the design of a scalar reward function for RL agents is fundamentally limited in capturing the complex, often conflicting objectives inherent to petrochemical operations. Operators must frequently balance priorities such as maximizing profitability, ensuring process stability, and satisfying operational constraints—priorities that dynamically shift based on market conditions, feedstock availability, and plant status. A static reward model cannot fully reflect these evolving trade-offs, leading to policy behaviors that may diverge from operator intent or practical feasibility.

To resolve these issues, we propose Population-Based Multi-Scenario Planning (PBMSP), a novel algorithm designed to complement existing RL agents. PBMSP maintains a diverse population of candidate schedules, each optimized under different objectives and constraint levels. This diversity enables the system to handle shifting operational criteria and priorities, providing operators with a robust set of scheduling options that better align with current plant conditions and strategic goals. Furthermore, PBMSP supports efficient asynchronous planning by identifying synchronized time points across candidate schedules, allowing fair comparisons for effective local search.

In summary, our primary contribution lies in the design and integration of PBMSP, a planning algorithm that bridges the gap between the potential of RL-based scheduling and the demands of real-world NCC operations. Through PBMSP, we enhance the usability, robustness, and adaptability of multi-agent RL systems, moving closer to their deployment in actual industrial environments.

2 Population-Based Multi-Scenario Planning (PBMSP)

This section presents our algorithm for multi-scenario scheduling, built on a structured population model. Specifically, the algorithm organizes these candidate solutions by operational characteristics and details their generation and improvement under varying constraint levels.

2.1 Structured Population Design

Unlike approaches that rely on a single population (Jaderberg et al., 2017; Jung et al., 2020; Parker-Holder et al., 2020; Wu et al., 2023; Zhao et al., 2023), our framework structures the population into distinct groups.

Each group is associated with a specific operational scenario. This scenario is defined by a unique combination of an operational criterion and an operating level. The criterion is evaluated by a scalar fitness function reflecting aspects like profitability and stability. The operating level dictates the stringency of operational constraints. These levels span a spectrum from conservative (using a limited control range) to stressed (pushing equipment operation to near its critical limits). A higher level signifies more restrictive operational constraints.

The resulting hierarchical structure of operating levels presents a useful characteristic. Schedules satisfying stricter constraints (higher level) inherently meet looser ones (lower level), potentially enabling the transfer of promising solutions across different operational priorities. This design choice mirrors real-world NCC operations where constraint stringency naturally varies based on plant conditions or goals; for instance, stressed levels might suit high demand while conservative levels prioritize safety during stable periods.

By adopting this structured population, we aim to leverage the inherent benefits of population-based approaches—such as parallel exploration and local optima escape—while directly addressing the challenges posed by the multi-faceted nature of NCC scheduling. The dedication of specific groups to distinct scenarios is intended to improve the efficiency and effectiveness of the search process.

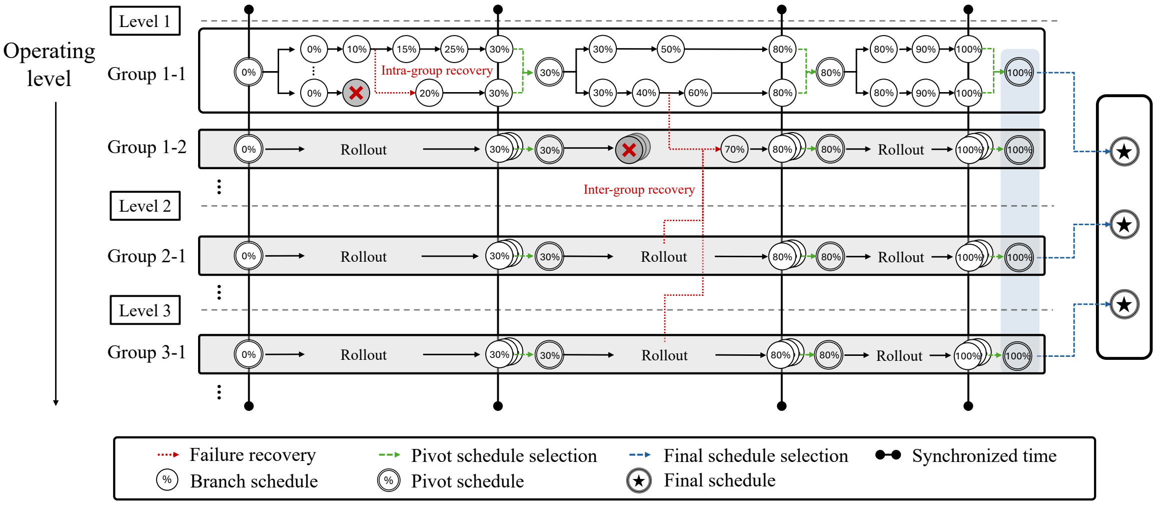


Figure 2: Multiple groups, each tied to a specific operational criterion and operating level, maintain branch schedules progressing via iterative rollouts (completion percentage tracked). At synchronized times, each group's pivot schedule is chosen from its branch schedules and those of equal or higher level groups. Failures are managed by intra/inter-group recovery. The final schedules are selected from the set of complete pivot schedules. For clarity, Group 1-1's rollout is detailed; others are brief.

This organization facilitates targeted exploration under diverse operational priorities and constraints, with the expectation of yielding a more comprehensive and robust set of high-quality schedules compared to a uniformly explored population.

2.2 Schedule Construction Process

Based on the group defined above, our framework generates a diverse set of schedules through the following iterative process as depicted in Figure 2.

Initialization At the beginning of the planning, each group's pivot schedule—which serves as the current best-known solution and the baseline for exploration for that group's scenario—is initialized as an empty sequence of macro-actions, reflecting the initial status of the NCC system.

Iterative construction The following steps are repeated iteratively until planning horizon:

(1) Pivot-based branching: The pivot schedule is replicated in parallel to create multiple branch schedules. This allows a broad exploration of various alternative decisions.

(2) Rollout based on scenario: Each branch schedule undergoes a rollout process considering its associated group's scenario. This process progressively constructs a complete schedule by sequentially applying macro-actions.

(3) Synchronized evaluation and update: At predefined *synchronized time*—specific moments where all branch schedules have reached an identical operational time (e.g., a shared event like a vessel arrival)—each group evaluates not only its own branch schedules but also those from all other groups operating at an equal or higher level, based on their respective fitness functions. This strategy facilitates the discovery of solutions that effectively balance diverse priorities and constraints.

Throughout the rollout process, the algorithm incorporates a robust failure recovery mechanism.

- *Intra-group recovery*: A failing schedule within a group is replaced by a copy of the current best-performing schedule in that group (based on its fitness), and rollout continues.

- *Inter-group recovery*: If all schedules in a group fail, the entire group is re-initialized with a copy of the best-performing schedule from all groups at an equal or higher operating level, and rollout resumes.

Furthermore, if all schedules across all groups fail, the algorithm restarts exploration from each group's pivot schedule at the last synchronized time. These layered mechanisms enhance the planning robustness by preventing premature termination.

Final schedule selection Once the iterative planning process is complete, a final selection step is performed. Instead of directly presenting all group-specific pivot schedules, a separate set of final evaluation criteria is applied to assess these complete schedules. This is because, unlike the fitness functions used during the planning process, the final criteria can consider aspects that can only be accu-

Methods	Success Rate (%)	Normalized Return	Time (min.)
PBMSP (Full resources for parallel rollout)	93.8	0.994	38.8
PBMSP (50% resources for parallel rollout)	87.5	0.988	37.8
Simple RL Rollout (10k sampling)	37.5	0.922	516.5
Simple RL Rollout (1k sampling)	12.5	0.926	57.1

Table 1: Quantitative comparison of PBMSP and Simple RL Rollout methods.

rately assessed once a complete schedule has been generated. The top-performing schedules, according to these final criteria, are then presented to the human operator for review and implementation.

3 Evaluation

Quantitative Analysis We assess the effectiveness of the proposed method through experiments based on diverse expert-designed backtest schedules. Each schedule captures the full information describing the operational status of the NCC plant at the time of scheduling, including inventory levels, equipment availability, and process constraints.

Due to confidentiality agreements with industry collaborators, we omit specific configuration details and parameter values; however, results are presented in an abstracted form that faithfully reflects the comparative performance and key insights.

We compare two methods for schedule generation based on a pre-trained RL policy from our prior work (Hong et al., 2024b): 1) **PBMSP**, our proposed method that actively explores diverse schedule groups, and 2) **Simple RL Rollout**, a baseline that samples 1,000 or 10,000 schedules using the policy and selects the highest-return one that successfully completes. All experiments were performed on two AMD EPYC 7453 28-core processors, with both methods parallelized to fully utilize available resources.

We evaluate these methods using key metrics:

- *Success Rate*: The average success rate in generating a complete schedule without failure.

- *Normalized Return*: The maximum return among successfully generated schedules. Normalized by the maximum return across all methods for each specific data point.

- *Wall-clock Time*: The average time taken to generate a schedule across all data points.

PBMSP consistently outperforms Simple RL Rollout by generating schedules from a broader range of initial operational status (higher success rate) and achieving higher returns, while also requiring significantly less time due to more efficient sampling. Although PBMSP (50% resources) has similar wall-clock time due to parallelization, its reduced group size leads to fewer schedules and thus lower performance than the full PBMSP.

Qualitative Insights from Deployment We introduced an online web service (Hong et al., 2024b) to optimize NCC operational schedules. This platform enabled users to upload the current operational status and generate schedules. The service then presented these schedules with figures and statistics in staff-friendly downloadable formats.

We have integrated PBMSP into this web service. Feedback from operators indicates this integration has significantly enhanced the service's real-world utility, delivering several key improvements:

- *Enhanced Schedule Generation and Utilization*: The frequency of generating successful schedules has dramatically increased. This allows users to rely on service-generated schedules more often in practice, leading to greater operational reliability and reduced need for manual intervention.

- *Diverse and Adaptable Schedule Offerings*: The service now provides a broader range of successful schedules. This variety gives users the flexibility to select schedules that best align with their current operational priorities.

- *Increased Profitability*: Backtesting data reveals that the PBMSP-enhanced service consistently generates more profitable schedules compared to those created by human experts.

4 Conclusion

This paper presents PBMSP, a population-based approach that enhances NCC scheduling to overcome the limitations of standalone RL. By maintaining a diverse population of candidate schedules optimized for varied objectives and constraints, it improves schedule completeness and efficiency, as confirmed by operator feedback. PBMSP also shows strong potential for broader industrial optimization problems with dynamic constraints and can contribute to the planning capabilities increasingly needed by modern large language models.

References

Christopher Amato, George Konidaris, Leslie P Kaelbling, and Jonathan P How. 2019. Modeling and planning with macro-actions in decentralized POMDPs. Journal of Artificial Intelligence Research, 64:817–859.

Sunghoon Hong, Whiyoung Jung, Deunsol Yoon, Kanghoon Lee, and Woohyung Lim. 2024a. Agent-oriented centralized critic for asynchronous multi-agent reinforcement learning. In International Conference on Autonomous Agents and Multiagent Systems Workshop on Adaptive and Learning Agents.

Sunghoon Hong, Deunsol Yoon, Whiyoung Jung, Jinsang Lee, Hyundam Yoo, Jiwon Ham, Suhyun Jung, Chanwoo Moon, Yeontae Jung, Kanghoon Lee, Woohyung Lim, Somin Jeon, Myounggu Lee, Sohui Hong, Jaesang Lee, Hangyoul Jang, Changhyun Kwak, Jeonghyeon Park, Changhoon Kang, and Jungki Kim. 2024b. Naphtha cracking center scheduling optimization using multi-agent reinforcement learning. In International Conference on Autonomous Agents and Multiagent Systems.

Max Jaderberg, Valentin Dalibard, Simon Osindero, Wojciech M. Czarnecki, Jeff Donahue, Ali Razavi, Oriol Vinyals, Tim Green, Iain Dunning, Karen Simonyan, Chrisantha Fernando, and Koray Kavukcuoglu. 2017. Population based training of neural networks. arXiv preprint arXiv:1711.09846.

Chonghyo Joo, Hyukwon Kwon, Junghwan Kim, Hyungtae Cho, and Jaewon Lee. 2023. Machine-learning-based optimization of operating conditions of naphtha cracking furnace to maximize plant profit. Computer Aided Chemical Engineering, 52:1397–1402.

Whiyoung Jung, Sunghoon Hong, Deunsol Yoon, Kanghoon Lee, and Woohyung Lim. 2025. Agent-centric actor-critic for asynchronous multi-agent reinforcement learning. In International Conference on Machine Learning.

Whiyoung Jung, Giseung Park, and Youngchul Sung. 2020. Population-guided parallel policy search for reinforcement learning. In International Conference on Learning Representations.

Jeongdong Kim, Chonghyo Joo, Minsu Kim, Nahyeon An, Hyungtae Cho, Il Moon, and Junghwan Kim. 2023. Multi-objective robust optimization of profit for a naphtha cracking furnace considering uncertainties in the feed composition. Expert Systems with Applications, 216:119464.

Ho-kyung Lee. 2012. Method for naphtha storage tank operation and system for the same. KR Patent App. KR1020100046570A.

Taeyeong Lee, Jun-Hyung Ryu, Ho-Kyung Lee, and In-Beum Lee. 2010. A study on scheduling of naphtha transportation and storage systems for naphtha cracking center. Chemical Engineering Research and Design, 88(2):189–196.

LG. 2024. What happens when a 24-hour AI system is introduced to petrochemical plant? https://www.youtube.com/watch?v=UBlgcgluIjU.

Jack Parker-Holder, Aldo Pacchiano, Krzysztof M Choromanski, and Stephen J. Roberts. 2020. Effective diversity in population based reinforcement learning. In Advances in Neural Information Processing Systems, volume 33, pages 18050–18062.

Shuang Wu, Jian Yao, Haobo Fu, Ye Tian, Chao Qian, Yaodong Yang, Qiang Fu, and Yang Wei. 2023. Quality-similar diversity via population based reinforcement learning. In International Conference on Learning Representations.

Yuchen Xiao, Weihao Tan, and Christopher Amato. 2022. Asynchronous actor-critic for multi-agent reinforcement learning. In Advances in Neural Information Processing Systems, volume 35, pages 4385–4400.

Rui Zhao, Jinming Song, Yufeng Yuan, Haifeng Hu, Yang Gao, Yi Wu, Zhongqian Sun, and Wei Yang. 2023. Maximum entropy population-based training for zero-shot human-AI coordination. In The AAAI Conference on Artificial Intelligence, pages 6145–6153.

Overview of PBIG Shared Task at AgentScen 2025: Product Business Idea Generation from Patents

Wataru Hirota,[1] Tomoko Ohkuma,[2] Tomoki Taniguchi,[2]
Chung-Chi Chen,[3] Tatsuya Ishigaki[3]

[1]Stockmark, [2]Asahi Kasei Corporation, [3]AIST
wataru.hirota@stockmark.co.jp, ishigaki.tatsuya@aist.go.jp
okuma.td, taniguchi.tcr@om.asahi-kasei.co.jp

Abstract

This paper provides an overview of the shared task *Product Business Idea Generation from Patents (PBIG)*, held as part of the AgentScen2025 workshop at IJCAI2025. The task challenges participants to generate practical and innovative product business ideas based on real patent documents, under the constraint that the proposed product must be feasible to launch within three years. Participants were required to generate four textual components for each patent: product title, product description, implementation, and differentiation. The evaluation was conducted via pairwise comparisons using both large language models (LLMs) and human annotators across multiple criteria including technical validity, innovativeness, specificity, need validity, market size, and competitive advantage. This paper outlines the task setup, dataset structure, evaluation protocols, and discusses insights derived from participant submissions.

1 Introduction

Recent advances in large language models (LLMs) have enabled impressive capabilities in ideation tasks such as scientific discovery (Wang et al., 2024; Si et al., 2025; Lu et al., 2024; Keisuke et al., 2025) and future forecast (Ishigaki et al., 2022). However, the generation of viable business ideas grounded in real-world technologies remains a challenging and underexplored area. Unlike general language generation tasks, successful product ideation requires a combination of domain expertise, identification of unmet user needs, and the creative integration of novel technologies.

To address this challenge, we organized the shared task *Product Business Idea Generation from Patents (PBIG)* at the AgentScen-2025 workshop, co-located with IJCAI-2025. This task leverages patent documents as rich sources of technical knowledge and asks participants to propose product business ideas that could realistically be implemented within a short time frame (three years). The task aims to encourage natural language processing-based systems.[1] that are not only creative but also grounded in technical feasibility and market viability.

This overview paper presents the design, data, and evaluation protocols of the PBIG shared task. We also summarize the submitted systems, present results from automatic and human evaluations, and discuss open challenges and future directions.

2 Task Definition

This section describes the task, dataset, and evaluation protocols.

2.1 Task Overview

Participants were provided with real-world patent documents in text format, including both the abstract and the full description. Given this input, the task was to generate a product business idea that leverages the patented technology.

The proposed product must be something that could realistically be implemented and brought to market within three years. For each patent, systems were required to generate the following four textual components:

- **Product Title**: A concise name for the product (up to 100 characters).
- **Product Description**: A brief explanation covering the product's function, target users, user needs, and benefits (up to 300 characters).
- **Implementation**: A description of how the patented technology will be applied to realize the product (up to 300 characters).

[1] Idea generation models are not necessarily LLM-based but all submissions this time use LLM-based approaches.

- **Differentiation**: A description of what makes the product unique and how it stands out from existing solutions (up to 300 characters).

Participants were allowed to use any external knowledge sources in addition to the input patent, including other patents, web data, or APIs. Submissions were required to follow a structured JSON format specified by the organizers.

2.2 Dataset

The shared task dataset consisted of 150 patents sampled from the USPTO,[2] categorized into three technical domains: natural language processing (NLP), computer science, and material chemistry. Each patent was provided in a structured JSONL format with metadata (title, application/publication number and date), abstract, claims, and description fields. Additional materials, including patent PDFs and figure images, were also available in per-patent directories.

2.3 Evaluation

2.3.1 Overview

The submitted product business ideas were evaluated from six perspectives:

- **Technical Validity**: Is the idea technically feasible within three years?
- **Innovativeness**: Does the idea offer a novel solution to the demand?
- **Specificity**: Is the idea concrete and clearly articulated?
- **Need Validity**: Does the idea address an actual, well-defined user need?
- **Market Size**: Is the market large enough to justify the product?
- **Competitive Advantage**: What business advantage is gained by the idea?

Two types of annotators were involved in the evaluation process: domain **human experts** and **LLMs** (LLM-as-a-Judge).

2.3.2 Human Evaluation

Annotation Groups. Human experts were divided into two groups: the **technical group** and the **marketing group**. The technical group evaluated:

- Technical Validity
- Innovativeness
- Competitive Advantage

while the marketing group evaluated:

- Need Validity
- Market Size

For the NLP and Computer Science domains, manual annotation was conducted by NLP researchers from Stockmark and AIST. In the case of the **Material Chemistry** domain, experts from Asahi Kasei participated in both roles. All human evaluators are listed in the Acknowledgements section of this paper.

Sampling Ideas for Human Evaluation Due to the large number of submissions, we selected a subset of patents for human evaluation. For each selected patent, two annotators—one from the technical group and one from the marketing group—were assigned to evaluate each idea. In rare cases where assignment conflicts occurred, some ideas were evaluated by a single annotator.

Protocol Updates and Transition to Scoring Initial rounds of human evaluation were based on **pairwise comparisons**, in which annotators were shown two ideas and asked to judge which was better. However, agreement among annotators was low in this setting. To improve consistency, we transitioned to a **scoring-based protocol**, where each idea was assigned a numerical score for each criterion. Pairwise preferences could then be reconstructed by comparing scores from the same annotator.

We attach the full annotation guidelines in the appendix.

Pipeline-Based Annotation Protocol To handle low-quality or incomplete ideas and reduce annotation burden, we adopted a **pipeline-based evaluation strategy**. In this protocol, annotators sequentially evaluated each criterion and were allowed to skip subsequent criteria if earlier conditions were not met.

Technical Group Protocol:

1. **Specificity** is first scored on a 0–4 scale. If the score is 0–2 (i.e., the idea is too vague or unreadable), annotation stops.

[2] https://www.uspto.gov

2. If Specificity ≥ 3, the annotator proceeds to **Technical Validity** (0–4). If this score is ≤ 1, annotation also stops here.

3. If Technical Validity ≥ 2, **Innovativeness** is scored on a 0–5 scale.

4. **Competitive Advantage** is scored independently using a 0–4 scale based on two criteria: (A) whether the patented technology is hard to replicate, and (B) whether the technology is core to the business idea.

Marketing Group Protocol:

1. **Specificity** is first scored (0–4). If the score is ≤ 2, the evaluation ends here.

2. If Specificity ≥ 3, **Need Validity** is evaluated separately from two perspectives:
   - **ToC (Consumer)** needs: scored 0–3 based on the severity and importance of the need.
   - **ToB (Business)** needs: scored 0–3 based on the qualitative and quantitative return expected from addressing the need.

3. If Need Validity scores are too low (e.g., ToC = 1 or ToC + ToB ≤ 2), annotation ends. Otherwise, the annotator proceeds to **Market Size**, also evaluated from both ToC and ToB perspectives on a 0–3 scale.

Other Guidelines:

- Annotators were permitted to use external resources (e.g., web search, ChatGPT) to aid in evaluating technical feasibility or market need.

- Annotations focused on idea content, not linguistic quality. Minor grammatical issues or translation artifacts were not penalized.

- If a submission was truncated due to character limits and became incomprehensible, a low Specificity score (1 or 2) was assigned.

This pipeline-based protocol allowed evaluators to efficiently filter out infeasible ideas while focusing attention on higher-quality candidates.

Statistics of Human Evaluators

- **NLP / Computer Science:**
  - Technical group: 5 annotators (task organizers)
  - Marketing group: 7 annotators (consultants from Stockmark Inc.)

- **Material Chemistry:**
  - Combined group of 4 domain experts (Asahi Kasei Corporation)

2.3.3 LLM-as-a-Judge Evaluation

Models Used. To perform automated evaluation without relying on commercial APIs, we employed three open-access LLMs:

- `google/gemma-3-27b-it`

- `Qwen/Qwen3-30B-A3B`

- `meta-llama/Llama-3.3-70B-Instruct`

Inference Protocol. Each model was run with five different random seeds to ensure robustness. Two types of instructions were used:

- **Instruction #1 (Pairwise)**: Designed for direct comparison of two ideas. For each pair, two prompts were created by reversing the order of the ideas to mitigate positional bias. Inference results across seeds and orderings were aggregated via majority voting.

- **Instruction #2 and #3 (Scoring)**: Designed to assign a numerical score to each idea for specific evaluation criteria. The final score was computed as the mean across five sampled outputs.

This combination of human and automatic evaluation offers a reliable and scalable framework for assessing both the technical soundness and the business viability of LLM-generated product ideas.

3 Results and Discussion

We report the results of both automatic and human evaluations across the three domains—**NLP**, **Computer Science**, and **Material Chemistry**—with six evaluation criteria: *Technical Validity*, *Innovativeness*, *Specificity*, *Need Validity*, *Market Size*, and *Competitive Advantage*.

Domain	Criterion	1st	2nd	3rd
NLP	Tech. Validity	MK2 (1093)	MCG_DSN_late (1053)	ditlab (1010)
	Innovativeness	MK2 (1215)	ditlab (1111)	Shiramatsulab (1108)
	Specificity	MK2 (1215)	ditlab (1150)	Shiramatsulab (1113)
	Need Validity	MK2 (1076)	ditlab (1060)	Shiramatsulab (1030)
	Market Size	ditlab (1056)	TrustAI (1025)	MK2 (1008)
	Comp. Advantage	MK2 (1150)	MCG_DSN_late (1075)	ditlab (1034)
Computer Science	Tech. Validity	MK2 (1107)	ditlab (1003)	Shiramatsulab (983)
	Innovativeness	MK2 (1169)	ditlab (1078)	Shiramatsulab (1055)
	Specificity	MK2 (1170)	ditlab (1082)	Shiramatsulab (1007)
	Need Validity	MK2 (1053)	ditlab (1031)	Shiramatsulab (998)
	Market Size	TrustAI (1035)	MK2 (999)	ditlab (965)
	Comp. Advantage	MK2 (1124)	Shiramatsulab (1019)	ditlab (1011)
Material Chemistry	Tech. Validity	MK2 (1132)	ditlab (1021)	Shiramatsulab (998)
	Innovativeness	MK2 (1207)	MCG_DSN (1185)	NS_NLP (1152)
	Specificity	MK2 (1184)	MCG_DSN (1112)	ditlab (1067)
	Need Validity	NS_NLP (1129)	MK2 (1125)	ditlab (1093)
	Market Size	MK2 (1118)	ditlab (1050)	Shiramatsulab (1024)
	Comp. Advantage	MK2 (1146)	NS_NLP (1055)	ditlab (1011)

Table 1: Top three teams in automatic evaluation for each domain and criterion. Scores in parentheses.

Domain	Criterion	1st	2nd	3rd
NLP	Tech. Validity	MK2 (1025)	TrustAI (991)	ditlab (990)
	Innovativeness	MK2 (1103)	ditlab (1025)	TrustAI (926)
	Specificity	MK2 (1044)	ditlab (1036)	TrustAI (962)
	Need Validity	MK2 (1009)	ditlab (1003)	TrustAI (993)
	Market Size	TrustAI (1048)	ditlab (1024)	MK2 (921)
	Comp. Advantage	MK2 (1035)	ditlab (1008)	TrustAI (1000)
Computer Science	Tech. Validity	MK2 (1018)	TrustAI (1008)	ditlab (973)
	Innovativeness	MK2 (1036)	ditlab (992)	TrustAI (971)
	Specificity	ditlab (1020)	MK2 (995)	TrustAI (983)
	Need Validity	MK2 (1074)	TrustAI (980)	ditlab (945)
	Market Size	TrustAI (1035)	MK2 (999)	ditlab (965)
	Comp. Advantage	MK2 (1017)	ditlab (1007)	TrustAI (974)
Material Chemistry	Tech. Validity	TrustAI (1057)	MK2 (1017)	NS_NLP (1000)
	Innovativeness	NS_NLP (1017)	MCG_DSN (1009)	ditlab (1002)
	Specificity	ditlab (1047)	NS_NLP (1017)	MK2 (1010)
	Need Validity	ditlab (1035)	MCG_DSN (1026)	NS_NLP (1007)
	Market Size	NS_NLP (1017)	MK2 (1013)	ditlab (1009)
	Comp. Advantage	ditlab (1038)	TrustAI (998)	NS_NLP (997)

Table 2: Top three teams in human evaluation for each domain and criterion. Scores in parentheses.

3.1 Automatic Evaluation Results

Table 1 shows the top three systems for each domain and criterion in automatic evaluation. Across all domains, the **MK2** team achieved the highest average scores in nearly all criteria. Notably, in the NLP and Computer Science domains, MK2 consistently outperformed other teams, indicating a strong ability to generate ideas that aligned with LLM-based evaluation.

3.2 Human Evaluation Results

Table 2 summarizes the human evaluation results. In contrast to automatic evaluation, the rankings are more varied across domains, especially in Material Chemistry, where MK2 did not dominate.

Criterion	NLP	CS	Mat. Chem.
Tech Valid	-0.500	0.780	0.191
Innov	0.103	1.000	0.459
Spec	0.185	-0.155	0.049
Market Size	*	0.000	0.281
Need Valid	*	0.000	0.099
Comp Adv	0.563	-0.800	-0.199

Table 3: Krippendorff's α coefficientss in human evaluation for each domain and criterion.

3.3 Discussion

LLM vs. Human Judgment. In the NLP and Computer Science tracks, LLM-based and human evaluations aligned well, with MK2 and ditlab dominating both. However, in Material Chemistry, the human evaluators favored TrustAI and ditlab in criteria such as *Technical Validity* and *Competitive Advantage*, revealing a domain-specific gap in LLM judgment.

Inter-annotator agreement Table 3 shows Krippendorff's α coefficients (Krippendorff, 2011) for each evaluation criterion and domain, quantifying the consistency of human judgments. Overall, the coefficients are low With the exceptions of *Technical Validity* in Computer Science ($\alpha = 0.780$) and *Innovativeness* in Computer Science ($\alpha = 1.000$). These results indicate the subjectivity and difficulty of the remaining assessments.

Domain Expertise Matters. The Material Chemistry domain required deeper domain knowledge, which human annotators brought to bear. This underscores the limitations of general-purpose LLMs in specialized fields and motivates future work in domain-adapted LLMs or hybrid evaluation pipelines.

Specificity Drives Validity. Ideas with higher specificity tended to receive better evaluations across most other criteria. This confirms that concrete, well-described ideas are easier to evaluate and more likely to be perceived as feasible and valuable.

Takeaways for System Design. The strongest submissions incorporated structured prompts, external patent knowledge, and attention to both technical and business feasibility. Future systems may benefit from iterative generation, agentic collaboration, and retrieval-augmented generation with real-world context.

4 Conclusions

This paper presented an overview and analysis of the PBIG shared task, which challenges systems to generate realistic and creative product business ideas from patent data. The task provides a novel benchmark that spans technical feasibility, market reasoning, and creative synthesis—dimensions that are critical for real-world innovation.

Our analysis of the results revealed that while current LLMs can produce promising outputs, achieving balanced performance across technical and commercial criteria remains challenging. Top-performing systems like MK2 showcased how structured prompting, external knowledge use, and attention to business context can lead to strong results.

From an evaluation standpoint, combining human expert scores with LLM-based inference enabled scalable and fine-grained assessments, though subjectivity in some criteria remains a limiting factor. Continued research into robust, interpretable, and automated evaluation strategies is needed.

Looking ahead, we suggest the following directions for future research and shared tasks:

- Incorporating interactive or iterative ideation frameworks (e.g., multi-agent discussion, critique-and-revise).
- Using retrieval-augmented generation with market or user data for grounding.
- Enhancing the reproducibility and transparency of evaluation metrics.

We hope this shared task fosters further exploration into how language models can support the journey from technical invention to product innovation.

Acknowledgements

We thank the annotators for their valuable contributions to the data annotation process, which were essential for the evaluation of this shared task. From the Asahi Kasei TENAC R&D Department, we gratefully acknowledge Yoshitaka Shigematsu, Motohiro Fukai, Tomohito Asakawa, and Shinya Akebi. From Stockmark Inc., we thank Daisuke Nakagawa, Jiro Kashiwagi, Kumiko Boshita, Shotaro Kusaura, Tomoaki Okamoto, Wataru Inoue, and Yutaka Mori for their support and insights.

References

Tatsuya Ishigaki, Suzuko Nishino, Sohei Washino, Hiroki Igarashi, Yukari Nagai, Yuichi Washida, and Akihiko Murai. 2022. Automating horizon scanning in future studies. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 319–327, Marseille, France. European Language Resources Association.

Ueda Keisuke, Hirota Wataru, Asakura Takuto, Omi Takahiro, Takahashi Kosuke, Arima Kosuke, and Ishigaki Tatsuya. 2025. Exploring design of multi-agent llm dialogues for research ideation. In *Proceedings of SIGDIAL*.

Klaus Krippendorff. 2011. Computing krippendorff's alpha-reliability.

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. 2024. The AI Scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*.

Chenglei Si, Diyi Yang, and Tatsunori Hashimoto. 2025. Can LLMs Generate Novel Research Ideas? A Large-Scale Human Study with 100+ NLP Researchers. In *ICLR*.

Qingyun Wang, Doug Downey, Heng Ji, and Tom Hope. 2024. SciMON: Scientific inspiration machines optimized for novelty. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 279–299, Bangkok, Thailand. Association for Computational Linguistics.

A Annotation Guidelines (Technical Evaluation)

Thank you for participating in the annotation for Product Business Idea Generation from patent documents (PBIG). This document explains how to evaluate ideas from a technical perspective.

1. Evaluation Flow

Ideas are to be evaluated in the following order. The scoring criteria are defined in Section 2.

1. Evaluate **Specificity**.
   - If the score is 0, 1, or 2, stop the evaluation (do not proceed to Technical Validity or Innovativeness).
   - If the score is 3 or 4, proceed to Technical Validity.
2. Evaluate **Technical Validity**.
   - If the score is 0 or 1, stop the evaluation (do not proceed to Innovativeness).
   - If the score is 2 or higher, proceed to Innovativeness.
3. Evaluate **Innovativeness**.
4. Evaluate **Competitive Advantage** independently.

2. Scoring Definitions

Specificity (0–4)

- 0: Cannot judge / Insufficient background knowledge.
- 1: Not readable as natural language.
- 2: Readable, but intention is unclear and no concrete product is imaginable.
  Example: "A platform contributing to carbon neutrality."
- 3: One or more specific product ideas can be imagined (some ambiguity remains).
  Example: "A tool for obtaining user insight from social media."
- 4: One clearly defined product is imagined.
  Example: "A washing machine operable by voice commands."

Technical Validity (0–4)

- 0: Cannot judge.
- 1: Patent technology does not appear applicable or is irrelevant.
- 2: Difficult to implement, but prototyping is feasible.
- 3: Prototyping is feasible using the patented technology.
- 4: Production-ready implementation is feasible.

Innovativeness (0–5)

- 0: Cannot judge.
- 1: Already known application; lacks novelty.
- 2: Known applications exist but underexplored.
- 3: Unusual use case, but not especially novel.
- 4: Interesting and surprising idea.
- 5: Highly innovative idea.

Competitive Advantage (0–4) Evaluate based on the following two criteria:

Criterion A: Is it difficult to replicate the business idea without the patented technology?

Example (Fails A): Extracting date mentions from text – easily replaceable by general-purpose NLP tools.

Example (Satisfies A): Making accurate recommendations with few labeled samples – difficult to replicate.

Criterion B: Is the patented technology essential to realizing the business idea?

Example (Fails B): Reducing speaker weight to improve car fuel efficiency – the component contributes minimally.

Example (Satisfies B): Reducing main body weight to improve car fuel efficiency – major impact on outcome.

Then assign a score according to the combination:

- 0: Cannot judge.

- 1: Neither A nor B is satisfied.
- 2: Only B is satisfied (the technology is core, but not strong).
- 3: Only A is satisfied (the technology is strong, but not core).
- 4: Both A and B are satisfied.

B Annotation Guidelines (Marketing Evaluation)

Thank you for participating in the annotation for PBIG. This document explains how to evaluate ideas from a market perspective.

1. Column Definitions

- `idea_id`, `patent_number`, `patent_title`, `patent_abstract`: Not used in market evaluation.
- `idea_title`, `idea_description`, `idea_implementation`, `idea_differentiations`: These are the elements being evaluated.

2. Evaluation Flow

1. Evaluate **Specificity**.
   If Specificity ≤ 2, stop here.
2. Evaluate **Need Validity** (ToC and ToB perspectives).
   If ToC = 1 or ToC + ToB ≤ 2, stop here.
3. Evaluate **Market Size** (ToC and ToB perspectives).

3. Scoring Definitions

Specificity (0–4) Same as in technical evaluation.

Need Validity (ToC)

- 0: Cannot judge / no ToC relevance.
- 1 (Low): Weak need, few seek solutions.
  Example: "Earphone cables tangle sometimes."
- 2 (Medium): Some burden, but not critical.
  Example: "Shoulder pain from computer use."
- 3 (High): Severe or essential need.
  Example: "Fall risk for elderly at home."

Need Validity (ToB)

- 0: Not a ToB idea / Cannot judge.
- 1 (Low): Minimal qualitative/quantitative benefit.
- 2 (Medium): Either qualitative or quantitative return is large.
  Examples: "Cost savings" (quantitative), "Knowledge transfer" (qualitative).
- 3 (High): Both types of return are large.

Market Size (ToC)

- 0: Cannot judge / not a ToC product.
- 1 (Small): Niche or non-essential item.
  Example: "VR goggles, road bikes"
- 2 (Medium): Popular, not essential.
  Example: "Tablets, coffee makers"
- 3 (Large): Nearly all households need it.
  Example: "Toothbrushes, smartphones"

Market Size (ToB)

- 0: Not a ToB product / Cannot judge.
- 1 (Small): Useful to a few companies.
  Example: "Fast PoC for car parts"
- 2 (Medium): Addressable need for many, but conditional.
  Example: "BI tools"
- 3 (Large): Needed by most companies.
  Example: "Procurement management tools"

4. Annotation Notes

- Annotators may use Google or ChatGPT to aid judgment.
- If technical content is unclear, market evaluation should still proceed.
- Do not penalize for minor unnatural Japanese or truncation artifacts.
- If truncation makes the idea meaningless, assign Specificity = 1 or 2.

5. Example Cases

- **Case 1:** Specificity = 2 $\Rightarrow$ stop.
- **Case 2:** Specificity = 3, ToC + ToB = 2 $\Rightarrow$ continue to Market Size.

Team NS_NLP at the AgentScen Shared Task: Structured Ideation Using Divergent and Convergent Thinking

Hayato Yoshiyasu
Nippon Shokubai Co., Ltd., 5-8 Nishi Otabi-Cho, Suita, Osaka 564-8512, Japan
hayato_yoshiyasu@shokubai.co.jp

Abstract

This paper presents our participation report for the Shared Task "Product Business Idea Generation from Patents" [1] conducted at The 2nd Work-shop on Agent AI for Scenario Planning - IJCAI 2025, as Team NS_ NLP. In this study, we explore a method that combines divergent and convergent thinking in a stepwise reasoning process, supplemented with external information, to generate business ideas based on patent data. As a result, our approach achieved first place in several criteria within the Materials Chemistry category, based on evaluation conducted by both LLMs and human experts.

1 Introduction

In today's business environment, characterized by the VUCA era—Volatility, Uncertainty, Complexity, and Ambiguity—companies are increasingly required to make rapid and flexible decisions and respond strategically to constant change. Under such conditions, the continuous generation of innovative business ideas is essential for creating new value and maintaining adaptability.

However, the process of generating business ideas still heavily relies on human experience and intuition, which presents several challenges. First, ideas are frequently shaped by individual knowledge and prior experiences, they tend to be biased and constrained by existing frameworks. Second, integrating and analyzing large and diverse information sources—such as technical data, market trends, and customer needs—is essential but challenging for humans to perform efficiently.

In addition to these challenges, the generation of high-quality business ideas demands a wide range of skills, including the ability to create novel concepts, evaluate them objectively, and continuously gather relevant information. These demands present a substantial challenge to the continuous generation of valuable and innovative ideas, thereby impeding efforts to enhance corporate value through sustained business innovation.

Considering these difficulties, growing attention has been directed toward the use of generative AI technologies, particularly large language models (LLMs). Recent studies have demonstrated the effectiveness of LLMs in complex intellectual tasks such as generating scientific hypotheses and discovering novel knowledge (Wang et al., 2024; Si et al., 2025). These findings suggest that LLMs possess strong potential for supporting idea generation through more autonomous and progressive reasoning processes in the future.

Several studies have explored the use of LLMs for generating research themes and product ideas. For example, some approaches leverage scientific literature by collecting and fine-tuning on papers to generate ideas grounded in scholarly knowledge (Wang et al., 2024; Porsdam et al., 2023), while others utilize patent data as an alternative source of domain-specific information (Zhu et al., 2022). These studies suggest that augmenting the prior knowledge of LLMs with domain-relevant data can enhance the novelty and relevance of the generated ideas. However, challenges remain in terms of evaluating the practicality and feasibility of the generated ideas, as well as in achieving sufficient technical depth.

Motivated by the challenges discussed above, we participated in the Shared Task "Product Business Idea Generation from Patents (PBIG)" [1] held at The 2nd Workshop on Agent AI for Scenario Planning (AgentScen) - IJCAI 2025, specifically in the Materials Chemistry category.

[1] https://sites.google.com/view/agentscen/shared-task

2 Task Description

2.1 Product business idea generation from patents (PBIG)[1]

The aim of the PBIG shared task is to generate viable product business ideas utilizing patent information. Generating business ideas requires diverse capabilities, including a deep understanding of relevant domains, user need identification, and creative concept integration. If LLMs can support the generation of business ideas that are both innovative and viable, they may serve as a promising means to accelerate AI-driven innovation.

In this task, participants are required to generate four explanatory texts of a business idea. A "business idea" is defined as a concept for a product or service that utilizes patented technology and is realistically implementable within a three-year timeframe. The required outputs are as follows:

1. Product Title: A concise name for the product.
2. Product Description: A brief explanation of the product's key features and functions, target users, their needs, and the benefits provided.
3. Implementation Method: A description of how the patented technology is applied to the product.
4. Differentiation Points: An explanation of how the product is unique compared to existing solutions and what makes it stand out.

2.2 Dataset

Participants were provided with a dataset containing full texts and diagrams of patents. Fifty patents were selected from the USPTO for each of three domains: Natural Language Processing, Computer Science, and Materials Chemistry.

Experts curated the patents based on technical feasibility and diversity of potential product ideas, aiming to facilitate the generation of practical and diverse ideas.

2.3 Evaluation Metrics

The generated ideas were evaluated by both human experts and LLMs. In the human evaluation, each idea was first scored based on predefined criteria, and these scores were subsequently used to perform pairwise comparisons between different generation methods. In contrast, the LLMs evaluation employed two approaches: a direct pairwise comparison and a score-based pairwise comparison analogous to the human evaluation process.

Finally, Elo ratings were computed based on the comparison results to establish a ranking of the idea generation methods according to their relative performance. Each idea was evaluated across the following six criteria.

1. Feasibility: Whether the patented techno-logy is appropriate for the product, can be implemented, and is realistically achievable within three years.
2. Novelty: Whether the patented technology offers a new solution to an existing demand.
3. Specificity: Whether the idea is concrete and clearly articulated.
4. Necessity: Whether the proposed solution addresses a genuine user need.
5. Market Potential: Whether the market is sufficiently large and has a substantial number of potential users.
6. Competitive Advantage: Whether the use of the patented technology provides a business advantage over existing solutions.

3 Methodology

This section outlines our approach to the Shared Task. Recent studies have shown that step-by-step prompting, known as Chain of Thought (CoT), is more effective than single step prompting for complex reasoning tasks (Wei et al., 2022) Based on these findings, a stepwise reasoning strategy was employed to generate business ideas. Our prompting approach leverages not only CoT reasoning but also incorporates both divergent and convergent thinking, which are known to facilitate creative ideation (Kim et al., 2013).

To address the limitations of relying solely on the model's prior knowledge—which can lead to biased outputs (Shah et al., 2024)— our approach is designed to incrementally generate content by incorporating supplementary information as needed. This process enables a stepwise development of patent-derived technologies into more well-grounded and practically viable business ideas.

An overview of our approach is illustrated in Figure 1. The process comprises seven sequential steps, each of which is described in detail in the following sections. The specific prompts used at each step are provided in the Appendix B.1~7

3.1 Step 1. Patent Analysis

In the first step, the functional properties of the materials described in each patent and their potential application markets are extracted. To ensure consistency in the granularity of the output, a few-shot prompting strategy was employed (Brown et

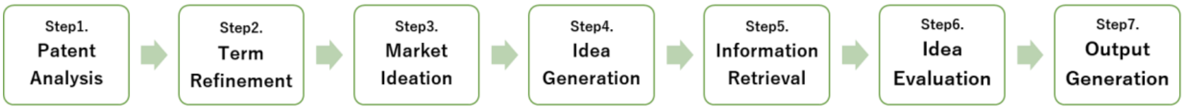


Figure 1: Workflow of our proposed approach

al., 2020). Furthermore, the prompt was formulated to differentiate between the material's inherent functions and those introduced or enhanced by the patented technology.

3.2 Step 2. Term Refinement

The extracted functional and market terms occasionally included overly broad or abstract expressions—such as "automotive industry"—that lacked sufficient specificity. To address this issue, we introduced a filtering step using an LLMs to identify and exclude such high-level concepts. Specifically, pairs of extracted terms were input into the model, which was prompted to infer whether a hypernym–hyponym (i.e., hierarchical) relationship existed between them. If such a relationship was identified, the model returned the term pair along with a confidence score ranging from 0 to 1. Terms identified as higher-level concepts with a confidence score above a predefined threshold (set to 0.9 in this study) were excluded from the final output.

3.3 Step 3. Market Ideation

Based on the refined information, we generated potential market domains. The prompt was designed to diversely generate 5 to 10 candidate market domains that satisfy the two conditions: (1) the functions described in the patent correspond to existing market needs, and (2) the functions improved or enhanced by the patented technology address known challenges. To encourage the generation of novel market ideas, the prompt also included an instruction to exclude any markets already mentioned in the original patent.

3.4 Step 4. Idea Generation

For each market identified in Step 3, the prompt was designed to generate ideas using the patented technology. It also encouraged analysis of market trends and challenges to create ideas that address real-world needs.

Furthermore, to prevent the incorporation of unrelated technologies (e.g., IoT or AI) and to ensure that the generated ideas reflect the intrinsic value of the patented technology, the prompt was constrained to utilize only the technologies explicitly described in the patent.

3.5 Step 5. Information Retrieval

To complement and enhance the business ideas generated in Step 4, we developed an agent-based information retrieval pipeline. Specifically, we integrated LLM with web search capabilities via SerpAPI[2] and implemented an agent based on the ReAct framework (Yao et al., 2023). This agent is capable of dynamically retrieving and synthesizing external information as needed, including market size and growth rate, major competitors, and the technological advantages held by those companies—factors that are essential for evaluating the feasibility and potential of the proposed business ideas.

3.6 Step 6. Idea Evaluation

In Step 6, we designed an evaluation process to identify the most promising business idea from those supplemented with external information. Rather than conducting a simultaneous comparison of all candidates, we adopted a tournament-style pairwise comparison approach. The LLM was prompted to evaluate each pair of business ideas based on the criteria defined in the "Evaluation Metrics" section.

3.7 Step 7. Output Generation

In the final step, the selected business idea was formatted according to the output specifications defined by the Shared Task. Given the character limits imposed by the submission format, the prompt explicitly instructed the model to adhere strictly to both the structural and length constraints. Additionally, a verification mechanism was implemented to ensure output length. If the generated output exceeded or fell short of the specified character limits, it was automatically regenerated.

[2] https://serpapi.com/

Table 1: Results of comparative experiment
(three-point rating)

Metrics	Baseline	Our Method
Technical validity	2.0	2.0
Innovativeness	2.0	**2.5**
Specificity	2.0	**2.9**
Need validity	2.0	**2.3**
Market size	2.0	1.8
Competitive advantage	2.0	2.0

4 Experiments

A comparative experiment was conducted within our team against the baseline method defined in this task, which generates business ideas using prompt engineering techniques. For this experiment, five patents were selected from the 50 patents in the Shared Task dataset, specifically from the Materials Chemistry category. Four researchers reviewed the content of each patent and compared the business ideas generated by both the baseline method and the proposed method.

The generated ideas were evaluated according to the criteria defined in the "Evaluation Metrics" section, using a three-point scale:

1. Inferior to the baseline
2. Equivalent to the baseline
3. Superior to the baseline

This evaluation enabled us to assess the relative advantages of the proposed method over the baseline and to verify its suitability for generating the final set of 50 business ideas.

For implementation, we constructed the processing flow using LangChain[3] and LangGraph [4] and employed GPT-4.1 [5] (gpt-4.1-2025-04-14) provided by OpenAI as the underlying LLM. To ensure structured outputs across multiple stages of the workflow, prompt engineering techniques were applied where appropriate (Marvin et al., 2023). (The specific prompt is provided in Appendix A.)

5 Results

Table 1 presents the results of the comparative experiment, showing the average scores for each evaluation criterion. These scores were assessed by researchers across five selected patents, enabling a comparison between the baseline method and the proposed method. The proposed method outperformed the baseline in terms of Innovativeness, Specificity, and Need validity. In contrast, it showed lower performance in Market size. For Technical validity and Competitive advantage, both methods performed at a comparable level.

Table 2: Elo Rating and Ranking results
※ () indicates participant ranking.

Metrics	LLMs	Human
Technical validity	971 (5)	1000 (3)
Innovativeness	1152 (3)	**1017 (1)**
Specificity	1017 (4)	1017 (2)
Need validity	**1129 (1)**	1007 (3)
Market size	997 (4)	**1017 (1)**
Competitive advantage	1055 (2)	997 (3)

Table 2 summarizes the results of the Shared Task, presenting the Elo ratings and rankings based on the submitted ideas (All results in Appendix C.). Although discrepancies were observed between ratings provided by LLMs and human evaluators, the proposed method achieved the highest scores in Innovativeness, Need Validity, and Market Size. Conversely, criteria such as Technical Validity and Competitive Advantage yielded Elo ratings below the initial baseline value of 1000.

6 Conclusion

This paper presented our method to the Shared Task: Product Business Idea Generation from Patents (PBIG). We developed a step-by-step workflow for generating business ideas by applying both divergent and convergent thinking based on the patented technology.

As a result, the proposed method achieved the highest scores in several criteria, demonstrating its effectiveness in generating business ideas. However, the evaluations for Technical Validity and Competitive Advantage were relatively low, indicating remaining challenges. Since these criteria are likely to require information beyond patent data, future work should focus on the accumulation and utilization of supplementary information sources suitable for idea generation.

[3] https://www.langchain.com/langchain

[4] https://www.langchain.com/langgraph

[5] https://platform.openai.com/docs/models/gpt-4.1

References

Wang, Q., Downey, D., Ji, H. and Hope, T. 2024. Scimon: Scientific inspiration machines optimized for novelty. *In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics* (Volume 1: Long Papers), pages 279-299.

Si, Chenglei, Diyi Yang, and Tatsunori Hashimoto. 2024. Can llms generate novel research ideas? a large-scale human study with 100+ nlp researchers. *arXiv preprint arXiv:2409.04109.*

Porsdam Mann, S., Earp, B. D., Møller, N., Vynn, S. and Savulescu, J. 2023. AUTOGEN: A personalized large language model for academic enhancement—Ethics and proof of principle. The American Journal of Bioethics 23.10: 28-41.

Zhu, Q., Luo, J. 2023. Generative Design Ideation: A Natural Language Generation Approach. In: Gero, J.S. (eds) Design Computing and Cognition'22. DCC 2022. Springer, Cham. https://doi.org/10.1007/978-3-031-20418-0_3

Elo. The Rating of Chessplayers, Past and Present. Ishi Press, 1986

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, QV. and Zhou, D. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems 35: 24824-24837.*

Kim, K.H., Pierce, R.A. 2013. Convergent Versus Divergent Thinking. *In: Carayannis, E.G. (eds) Encyclopedia of Creativity, Invention, Innovation and Entrepreneurship. Springer, New York, NY.* https://doi.org/10.1007/978-1-4614-3858-8_22

Shah, Chirag. 2024. From Prompt Engineering to Prompt Science with Humans in the Loop. *Communications of the ACM.*

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., ... and Amodei, D. (2020). Language models are few-shot learners. *Advances in neural information processing systems, 33, 1877-1901.*

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K. and Cao, Y. 2023. React: Synergizing reasoning and acting in language models. *In International Conference on Learning Representations (ICLR).*

Marvin, G., Hellen, N., Jjingo, D., & Nakatumba-Nabende, J. 2023. Prompt engineering in large language models. *In International conference on data intelligence and cognitive informatics (pp. 387-402). Singapore: Springer Nature Singapore.*

A Format Instructions

Our methodology incorporates specific format instructions at Steps 1, 2, 3, 6, and 7 of the process.

~~~
The output should be formatted as a
JSON instance that conforms to the
JSON schema below.

As an example, for the schema
{"properties": {"foo": {"title":
"Foo", "description": "a list of
strings", "type": "array", "items":
{"type": "string"}}}, "required":
["foo"]}
the object {"foo": ["bar", "baz"]} is
a well-formatted instance of the
schema. The object {"properties":
{"foo": ["bar", "baz"]}} is not well-
formatted.

Here is the output schema:
```
{Instructions for each step}
~~~

B Prompt Templates

B.1 Step 1. Patent Analysis

```
System: You are an excellent patent
analyst. Please extract information
from given patent document.

Human: {document}
```

B.2 Step 2. Term Refinement

```
System: You are a knowledgeable
assistant in linguistics and
terminology.
Determine whether the given two terms
have a hierarchical relationship:

Human:
Term A: {term_a}
Term B: {term_a}
```

B.3 Step 3. Market Ideation

```
System: Given the following two sets
of properties, identify 5 to 10
potential markets, applications, or
industries where:
- The 'General properties' are in
demand or required
- The 'Distinctive properties'
represent current challenges, unmet
needs, or innovation opportunities
```

Additionally, consider the list of 'Existing markets' provided. Exclude these from your suggestions to avoid redundancy.

Focus on real-world use cases and emerging needs. Return your results as a structured list of potential market sectors or product categories that are not already covered by the existing markets.

(e.g., 'fuel-related parts like gasoline tanks and valves', 'automotive parts exposed to cleaners', 'sliding components in AV and OA fields', 'binder resin compositions for metal powders')

Human:

General properties:

{General properties}

Distinctive properties:

{Distinctive properties}

Existing markets to exclude:

{Existing markets}

B.4 Step 4. Idea Generation

System: You are a business strategist. Based on the following patent document and the potential market, generate a new business idea that leverages the patented technology.

Analyze the current trends, critical challenge, and innovation gaps in the listed markets. Then, propose a business concept that addresses these needs using the core invention described in the patent.

Do not incorporate unrelated technologies (e.g., IoT, AI, blockchain) unless they are explicitly part of the patented invention.

Your response should include:

- A compelling business idea title
- A clear and concise description of the business model
- How the patented technology is used
- The target customer segment
- The value proposition and competitive advantage

Avoid repeating ideas that are already common in the listed markets and proposing service-based or platform-based models unless they are directly derived from the patented invention.

Human:

Patent document: {document}

Potential markets: {market}

B.5 Step 5. Information Retrieval

System: You are a market research analyst. Your task is to evaluate the commercial potential of the following business idea.

Please use external search tools to gather and summarize the following information:

1. Current market size and CAGR of the relevant industry to 2027
2. Key competitors and their offerings, competitive advantages, distinctive features, proprietary technology
3. Competitive advantages, distinctive features, and proprietary technology of each competitor
4. Major growth drivers and market trends
5. Key challenges and barriers to entry
6. Regulatory or technological considerations

Provide a concise and structured summary.

Human: {business idea}

B.6 Step 6. Idea Evaluation

System: ## Input

Read two product business ideas using the technology.

<idea id='1'>{idea_1}</idea>

<idea id='2'>{idea_2}</idea>

## Task

Your task is to choose the better idea from the perspective of criteria.

Evaluation Criteria:

- Innovation: How novel or groundbreaking is the business idea? Does it offer new solutions or improvements to existing problems?
- Feasibility: How practical and achievable is the business idea? Are the necessary resources, skills, and technologies available for implementation?
- Specificity: How clearly defined and detailed is the business plan? Does it address specific market needs, customer segments, and implementation steps?
- Market Size: How large is the potential market for the business? Does the idea target a growing or underserved market with high potential for expansion?

Table 3: Results of LLMs evaluation

Teams	Technical validity	Innovativeness	Specificity	Need validity	Market size	Competitive advantage
AiAnonymous	1038	782	883	892	944	941
ditlab	1021	1052	1067	1093	1050	1011
MK2	**1132**	**1207**	**1184**	1125	**1118**	**1146**
NS_NLP	971	1152	1017	**1129**	997	1055
Shiramatsulab	998	920	986	1006	1024	967
Team_MCG_DSN	896	1185	1112	946	939	1002
TrustAI	940	697	748	806	924	874

Table 4: Results of human evaluation (- means not evaluated)

Teams	Technical validity	Innovativeness	Specificity	Need validity	Market size	Competitive advantage
AiAnonymous	-	-	-	-	-	-
ditlab	996	1002	**1047**	**1035**	1009	**1038**
MK2	1017	990	1010	989	1013	991
NS_NLP	1000	**1017**	1017	1007	**1017**	997
Shiramatsulab	-	-	-	-	-	-
Team_MCG_DSN	928	1009	950	1026	1006	974
TrustAI	**1057**	978	973	941	952	998

```
- Competitive Advantage: How does the business plan stand out from competitors? What unique factors give it an edge in the market?

## Output
```

B.7 Step 7. Output Generation

```
System: Based on the provided information, generate a concise and information-rich business idea consisting of the following four components:
- title (80-100 characters): A short, compelling name for the product or solution.
- description (260-290 characters): A compact summary of the product’s key features, target users, their needs, and the benefits.
- implementation (260-290 characters): A brief explanation of how the patented technology will be applied in the product.
- differentiation (260-290 characters): A clear statement of what makes the product unique compared to existing solutions.
Each field must be written clearly and concisely, strictly limited to the specified character count. Prioritize clarity, specificity, and value delivery. Avoid vague or generic language. If any field is too short or too long, regenerate that field until it meets the requirement.
Use clear, specific, and value-driven language. Avoid vague or generic expressions.
Human:
# business plan
{business plan}
```

C Evaluation Results

The evaluation results of all participating teams in the Shared Task are presented. Table 3 shows the results based on evaluations conducted by LLMs, while Table 4 presents those based on human evaluators. In both cases, Elo ratings were calculated based on either pairwise comparisons or scoring-based comparative evaluations across six criteria. The detailed criteria for scoring are publicly available on GitHub[6].

[6] https://github.com/stockmarkteam/pbig-shared-task/tree/main/guidelines

Agent Ideate: A Framework for Product Idea Generation from Patents Using Agentic AI

Gopichand Kanumolu[1] **Ashok Urlana**[1,2] **Charaka Vinayak Kumar**[1]
Bala Mallikarjunarao Garlapati[1]
TCS Research, Hyderabad, India[1] IIIT Hyderabad[2]
{gopichand.kanumolu, ashok.urlana, charaka.v, balamallikarjuna.g}@tcs.com
ashok.u@research.iiit.ac.in

Abstract

Patents contain rich technical knowledge that can inspire innovative product ideas, yet accessing and interpreting this information remains a challenge. This work explores the use of Large Language Models (LLMs) and autonomous agents to mine and generate product concepts from a given patent. In this work, we design ***Agent Ideate***, a framework for automatically generating product-based business ideas from patents. We experimented with open-source LLMs and agent-based architectures across three domains: Computer Science, Natural Language Processing, and Material Chemistry. Evaluation results show that the agentic approach consistently outperformed standalone LLMs in terms of idea quality, relevance, and novelty. These findings suggest that combining LLMs with agentic workflows can significantly enhance the innovation pipeline by unlocking the untapped potential of business idea generation from patent data.

1 Introduction

With the rapid advancement of large language models (LLMs), there is growing interest in leveraging these models for tasks such as scientific discovery and innovation support. However, generating viable and actionable product ideas from patents requires not only comprehension of complex technical content but also creativity, domain knowledge, and market awareness (Urlana et al., 2024). Patents are legal documents that protect inventions and promote technological innovation (Mossoff, 2000), but their complex and technical language poses unique challenges. Despite the wealth of technical insights contained within patent documents, generating product business ideas from patents remains an underexplored area (Jiang and Goetz, 2024). To achieve this, AgentScen 2025 shared task[1] on Product Business Idea Generation from Patents (PBIG) was introduced as part of the 2^{nd} Workshop on Agent AI for Scenario Planning at IJCAI-25.

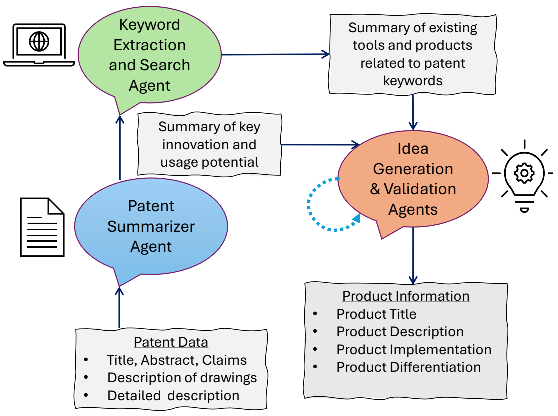

Keyword Extraction and Search Agent
Summary of existing tools and products related to patent keywords
Summary of key innovation and usage potential
Idea Generation & Validation Agents
Patent Summarizer Agent
Patent Data
• Title, Abstract, Claims
• Description of drawings
• Detailed description
Product Information
• Product Title
• Product Description
• Product Implementation
• Product Differentiation

Figure 1: Illustration of the Agent Ideate Pipeline.

1.1 Task formulation

The goal of this task is to evaluate systems that can read a patent and generate a realistic product idea that could be implemented and launched within three years. Each submission is expected to produce four concise outputs for a given patent:

1. ***Product title***: A concise name for the product.
2. ***Product description***: A brief explanation of the product outlining its essential features, target users, their needs, and the benefits provided by the product.
3. ***Implementation***: An explanation describing the implementation of patents technology into the product.
4. ***Differentiation***: An explanation highlighting what makes the product unique.

To support this task, the organizers released a curated dataset consisting of 150 U.S. patents across three categories: Computer Science (CS), Natural

[1] https://sites.google.com/view/agentscen/shared-task

Language Processing (NLP), and Material Chemistry. Participants were allowed to use external resources to enhance idea generation. System outputs were evaluated by both human experts and LLM-based evaluators based on multiple criteria, including technical feasibility, innovation, specificity, market need, and competitive advantage.

In this study, we built the **Agent Ideate** framework, which is a Multi-Agent architecture leveraging an external search tool for generating product ideas from patent text. The pipeline diagram is illustrated in Figure 1. We leverage an LLM-based judging approach to evaluate the ideas generated by the different methods and to select the most effective one. We also analyze the effectiveness of agent-based and LLM-driven architectures for transforming patent knowledge into innovative product concepts.

2 Related Work

The task of generating business ideas from patent documents (Yoshiyasu, 2025; Xu et al., 2025; Terao and Tachioka, 2025; Hoshino et al., 2025; Shimanuki et al., 2025) intersects with multiple research domains, including patent analysis (Sheremetyeva, 2003), patent summarization (Sharma et al., 2019), knowledge extraction (Tonguz et al., 2021), and large language model (LLM)-driven ideation. Prior work has explored the use of NLP and information retrieval techniques to extract technical concepts (Suzuki and Takatsuka, 2016; Tonguz et al., 2021) and commercial potential applications from patent texts (Souili et al., 2015; Jiang et al., 2025). More recently, LLMs have been applied for creative tasks such as product ideation, innovation support, showing promise in structured content generation (Girotra et al., 2023; Radensky et al., 2024; Li et al., 2024; Wen et al., 2006).

One closely related line of work is by Si et al. (2024), who investigate the research ideation capabilities of LLMs. They pose a critical question: Are current LLMs capable of generating novel ideas that rival those produced by human experts? To answer this, the authors conducted a large-scale study involving over 100 qualified NLP researchers who generated human baselines and performed blind evaluations of both human and LLM-generated ideas. Their findings reveal that LLM-generated ideas are often judged as more novel than those produced by domain experts.

Section	CS	NLP	Chemistry
Title	10	11	8
Abstract	134	138	130
Background	1058	910	6215
Claims	1499	1708	535
Description of Figures	4636	868	700
Detailed Description	1499	5068	156

Table 1: Average number of words present in each section for different datasets. CS - Computer science, NLP - Natural Language Processing.

In another study, SciMON(Wang et al., 2024) is a framework that enhances language models' ability to generate novel scientific ideas by leveraging literature-based inspirations and iterative novelty optimization. Unlike traditional link-prediction approaches, it takes contextual inputs (e.g., research problems) and produces natural language hypotheses, using retrieval from semantic, knowledge graph, and citation sources. While evaluations show improvements over GPT-4, the generated ideas still lack the depth and novelty of human-authored research. To this end, in contrast to the existing works, this study aims to generate product-based business ideas from patents by building a multi-agentic framework.

3 Dataset

The dataset provided by the shared task organizers comprises a total of 150 U.S. patents, with 50 patents each from three distinct domains: Computer Science(CS), Natural Language Processing (NLP), and Material Chemistry(MC). Each patent entry includes structured metadata such as the title, abstract, claims, description, publication number, and publication date.

Preprocessing: Among these fields, the description section is notably extensive, often exceeding the input length limitations of most large language models (LLMs). To address this challenge, we implemented a preprocessing strategy that segments the description into semantically meaningful subsections. This was achieved through regular expression-based matching, which identifies and extracts parts such as: Background information, Brief description of drawings and claims, and Detailed description of the patent technology.

This segmentation allows for more efficient and focused processing by LLMs and downstream agents. Detailed statistics about the dataset distribution and content lengths across categories are summarized in Table 1.

4 Methodology

As presented in Figure 1, we adopt three distinct methods to generate innovative business ideas from patent documents. These methods are increasingly sophisticated in terms of architecture and capability:

1. Prompt-based LLM Approach: This is the simplest baseline. We use a single-prompt approach with a large language model (LLM), wherein the entire patent (or its reduced components: title, abstract, claims, and summarized description) is passed as input to the model. The prompt is crafted to guide the model in generating business ideas, specifying the required structure in JSON format with fields such as product title, product description, implementation, and differentiation.

2. Multi-Agent LLM Architecture: The second approach builds on modularization via a multi-agent system, where different tasks are handled by different specialized agents. Specifically:

- A Patent Analyst Agent summarizes the core innovation and usage of the patent.
- A Business Idea Generator Agent uses the summarized insight to generate a structured business idea.
- A Business Validator Agent ensures the output adheres to format, character limits, and originality constraints.

Each agent uses the same LLM backend but is provided with a distinct goal and context. Tasks are executed sequentially with inter-agent context passing, allowing for better modularity, reliability, and control compared to single-shot prompting. In the rest of the paper, we refer to this method as the *Agent without Tool* approach.

3. Multi-Agent LLM with External Search Tool: The third and most comprehensive method incorporates a search tool to enrich the reasoning process with external information. It extends the second approach by introducing:

- A Keyword Extractor Agent, which identifies two core keywords from the summarized patent content.
- A Research Agent, which performs a DuckDuckGo tool-based web search using these keywords to gather information about existing tools, libraries, or products in the domain.
- The Business Idea Generator Agent utilizes both the patent summary and external market insights to create a business idea that is clearly differentiated from known solutions.
- Finally, the Business Validator Agent ensures the output is well-formed, concise, and novel.

We provide the role, goal, backstory, tool usage, task description, and expected output instructions for each agent in Appendix Table 4 and Table 5. In the rest of the paper, we refer to this method as the *Agent with Tool* approach.

5 Experiments and Evaluation

We conduct experiments with prompt-based, agent with Tool and agent without Tool based approaches. For all experiments, we used the llama-4-scout-17b-16e-instruct[2] model for response generation in both architectures: the prompt-based LLM model and each agent in the multi-agent setup. Due to resource constraints and the lack of access to proprietary APIs such as OpenAI, we opted to experiment with open-source LLMs hosted via the Groq API[3]. The LLM was configured with a temperature of 0.7 and a maximum token limit of 1000. All experiments were conducted using the free-tier access provided by Groq. For all agentic framework experiments, we used the CrewAI[4] framework to create agents and integrate with external search tools.

To assess the relative quality of business ideas generated by different methods, we employed an LLM-as-a-judge evaluation strategy. Specifically, we designed a structured prompt where the model is provided with a patent description and two product ideas generated using different approaches (e.g., baseline prompting vs. multi-agent with search). The LLM is then instructed to critically evaluate the ideas across six well-defined dimensions: technical validity, innovativeness, specificity, need validity, market size, and competitive advantage.

The evaluation setup and criteria are provided in Appendix Table 6 and Table 7. Explicitly listing the criteria reduces ambiguity and encourages the model to weigh each dimension before issuing a verdict. The output follows a strict JSON format, containing the selected better idea (idea 1 or idea 2) and a rationale for the decision.

[2] https://console.groq.com/docs/model/meta-llama/llama-4-scout-17b-16e-instruct

[3] https://console.groq.com/docs/models

[4] https://www.crewai.com/

Domain	Idea 1	Idea 2	Idea 1 Count(%)	Idea 2 Count(%)
Computer Science	Prompt-based LLM	**Agent without tool**	14	**86**
	Agent without tool	**Agent with Tool**	14	**86**
NLP	Prompt-based LLM	**Agent without tool**	02	**98**
	Agent without tool	Agent with Tool	**88**	12
Material Chemistry	Prompt-based LLM	**Agent without tool**	08	**92**
	Agent without tool	Agent with Tool	**64**	38

Table 2: Evaluation of ideas generated using various approaches. We employ the LLM-as-a-Judge method to compare the ideas and report the percentage of ideas selected by the judge.

Criteria	Chemistry	CS	NLP
Tech Validity	1	2	3
Specificity	3	3	3
Need Validation	5	2	4
Market Size	5	1	1
Innovativeness	5	3	4
Competitive Advantage	2	3	3

Table 3: Human evaluation results provided by the organizers. Each row represents the rank/position of our submission "**TrustAI**" for each domain based on the scores for each criteria.

We used a high-capacity model LLaMA 3 70B [5] hosted via Groq for inference, ensuring strong reasoning and evaluation capabilities. This method of LLM-based comparative evaluation offers a scalable and cost-effective alternative to human annotation, especially in scenarios involving nuanced technical and entrepreneurial judgments. Furthermore, by leveraging LLMs that are blind to the origin of each idea, we minimize bias and ensure that comparisons focus purely on idea quality, not model provenance.

6 Discussion

Evaluation using LLM as a judge: The automated evaluation results (using LLM as judge) in Table 2 show clear performance differences between approaches. The Agent with Tool method consistently generates highly-ranked ideas in Computer Science (86%), demonstrates moderate performance in Material Chemistry (38%), but performs poorly in NLP (12%). The standalone Agent approach without tool usage shows strong performance in NLP (98%) and Material Chemistry (64%), though it is less effective in Computer Science (14%) compared to the Agent with Tool method. The basic LLM prompt method performs poorly across all domains (Computer Science: 14%, NLP: 02%, Material Chemistry: 08%), suggesting that multi-agent frameworks provide substantial benefits even without tool access.

Based on the automatic evaluation results comparing which approach generated the best ideas for each domain, we submitted the highest-performing outputs for organizer evaluation. The results of this evaluation are discussed in the following section.

Evaluation results given by the Organizers: The human evaluation rankings in Table 3 reveal important domain-specific patterns. In Chemistry, our system achieved top rankings in Innovativeness (1^{st}) but performed poorly in Technical Validity (5^{th}), indicating highly creative but potentially less feasible ideas. For Computer Science, we see balanced performance across criteria (mostly 2^{nd}-3^{rd} place), suggesting reliable but not exceptional results. The NLP domain shows our strongest overall performance, with top-3 rankings in all criteria except Market Size (5^{th}), highlighting both the technical strength and potential niche focus of generated ideas.

7 Conclusion

This paper presented our framework Agent Ideate, for generating product ideas from patents. We have conducted experiments using prompt-based LLM, multi-agent framework, and tool-augmented agents. Automated evaluation (LLM-as-judge) showed that Agent with Tool performed best in Computer Science, while standalone Agent excelled in NLP, and Material Chemistry. Our findings highlight the potential of agentic AI for structured innovation while underscoring domain-specific challenges.

[5] https://console.groq.com/docs/model/llama-3.3-70b-versatile

8 Limitations

Our study has several key limitations. First, reliance on open-source LLMs (e.g., LLama-4-17B, and LLaMA-3-70B) may restrict performance compared to state-of-the-art proprietary models. Second, the system's effectiveness varies significantly across domains, requiring domain specific models. Finally, the tool-augmented agent's performance depends heavily on external search quality, which can introduce noise. These constraints highlight the need for more robust domain adaptation, hybrid evaluation methods, and improved tool integration in future work.

References

Karan Girotra, Lennart Meincke, Christian Terwiesch, and Karl T Ulrich. 2023. Ideas are dimes a dozen: Large language models for idea generation in innovation. *The Wharton School Research Paper Forthcoming*.

Mizuki Hoshino, Shun Shramatsu, and Fuminori Nagasawa. 2025. A business idea generation framework based on creative multi-agent discussions. In *The 2nd Workshop on Agent AI For Scenario Planning (AGENTSCEN2025)*.

Lekang Jiang and Stephan Goetz. 2024. Artificial intelligence exploring the patent field. *arXiv e-prints*, pages arXiv–2403.

Lekang Jiang, Caiqi Zhang, Pascal A. Scherz, and Stefan Goetz. 2025. Can large language models generate high-quality patent claims? In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 1272–1287, Albuquerque, New Mexico. Association for Computational Linguistics.

Long Li, Weiwen Xu, Jiayan Guo, Ruochen Zhao, Xingxuan Li, Yuqian Yuan, Boqiang Zhang, Yuming Jiang, Yifei Xin, Ronghao Dang, and 1 others. 2024. Chain of ideas: Revolutionizing research via novel idea development with llm agents. *arXiv preprint arXiv:2410.13185*.

Adam Mossoff. 2000. Rethinking the development of patents: an intellectual history, 1550-1800. *Hastings Lj*, 52:1255.

Marissa Radensky, Simra Shahid, Raymond Fok, Pao Siangliulue, Tom Hope, and Daniel S Weld. 2024. Scideator: Human-llm scientific idea generation grounded in research-paper facet recombination. *arXiv preprint arXiv:2409.14634*.

Eva Sharma, Chen Li, and Lu Wang. 2019. BIGPATENT: A large-scale dataset for abstractive and coherent summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2204–2213, Florence, Italy. Association for Computational Linguistics.

Svetlana Sheremetyeva. 2003. Natural language analysis of patent claims. In *Proceedings of the ACL-2003 Workshop on Patent Corpus Processing*, pages 66–73, ,. Association for Computational Linguistics.

Masaya Shimanuki, Naoto Shimizu, Kentaro Kinugasa, and Hiroki Sugisawa. 2025. Business idea generation from patent documents: Knowledge integration and self-improvement via llm. In *The 2nd Workshop on Agent AI For Scenario Planning (AGENTSCEN2025)*.

Chenglei Si, Diyi Yang, and Tatsunori Hashimoto. 2024. Can llms generate novel research ideas? a large-scale human study with 100+ nlp researchers. *arXiv preprint arXiv:2409.04109*.

Achille Souili, Denis Cavallucci, and François Rousselot. 2015. Natural language processing (nlp)–a solution for knowledge extraction from patent unstructured data. *Procedia engineering*, 131:635–643.

Shoko Suzuki and Hiromichi Takatsuka. 2016. Extraction of keywords of novelties from patent claims. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1192–1200, Osaka, Japan. The COLING 2016 Organizing Committee.

Yasunori Terao and Yuuki Tachioka. 2025. Collaborative invention: Refining patent-based product ideation via llm-guided selection and rewriting. In *The 2nd Workshop on Agent AI For Scenario Planning (AGENTSCEN2025)*.

Ozan Tonguz, Yiwei Qin, Yimeng Gu, and Hyun Hannah Moon. 2021. Automating claim construction in patent applications: The CMUmine dataset. In *Proceedings of the Natural Legal Language Processing Workshop 2021*, pages 205–209, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ashok Urlana, Charaka Vinayak Kumar, Ajeet Kumar Singh, Bala Mallikarjunarao Garlapati, Srinivasa Rao Chalamala, and Rahul Mishra. 2024. Llms with industrial lens: Deciphering the challenges and prospects–a survey. *arXiv preprint arXiv:2402.14558*.

Qingyun Wang, Doug Downey, Heng Ji, and Tom Hope. 2024. SciMON: Scientific inspiration machines optimized for novelty. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 279–299, Bangkok, Thailand. Association for Computational Linguistics.

Guihua Wen, Lijun Jiang, Jun Wen, and Nigel R Shadbolt. 2006. Generating creative ideas through patents. In *Proceedings of the 9th Pacific Rim international conference on Artificial intelligence*, pages 681–690.

Yuzheng Xu, Tosho Hirasawa, Seiya Kawano, Shota Kato, and Tadashi Kozuno. 2025. Mk2 at pbig competition: A prompt generation solution. In *The*

2nd Workshop on Agent AI For Scenario Planning (AGENTSCEN2025).

Hayato Yoshiyasu. 2025. Team ns_nlp at the agentscen shared task: Structured ideation using divergent and convergent thinking. In *The 2nd Workshop on Agent AI For Scenario Planning (AGENTSCEN2025)*.

A Appendix

We present the description of each agent's role, goal, backstory, and the tools they can access in Table 4 and Table 5. This also includes the task descriptions and expected outputs for each agent. Additionally, we provide the evaluation criteria used to compare the ideas generated by various methods using the LLM-as-a-judge approach in Table 6 and Table 7.

Agent Name	Role	Goal	Backstory / Tools Used
Patent Analyst	Reader Agent	Extract and summarize key features from patents	Specializes in understanding complex patent documents and identifying key technological aspects.
Keyword Extractor	Keyword Agent	Generate essential keywords from patent summary	NLP expert identifying core technologies to support product discovery.
Researcher	Search Agent	Search for relevant products/tools using keywords and synthesize results	Enthusiast in discovering tools/products relevant to keywords with clear and concise summaries. Tools Used: DuckDuckGo Tool
Idea Generator	Business Idea Agent	Generate innovative product ideas from patent content	Creative entrepreneur skilled in mapping technology to business ideas.
Business Validator	Validator Agent	Validate ideas for structure and uniqueness	Ensures business ideas are well-formatted, feasible, and differentiated from existing solutions.

Table 4: Description of each agent's role, goal, backstory, and tool usage

Task Name	Performed By	Task Description	Expected Output
Patent Analysis	Patent Analyst	Read and extract core information from patent sections	Structured summary of key patent features.
Keyword Generation	Keyword Extractor	Generate two keywords representing the patent's core technological concepts	List of keywords: `["keyword1", "keyword2"]`
Product Research	Researcher	Use keywords to search web using DuckDuckGo Tool for related products and synthesize findings	Text summary with relevant products/tools and short descriptions.
Idea Generation	Idea Generator	Based on findings and patent, generate an innovative product/business idea	JSON object with below fields: `product_title`, `product_description`, `implementation`, `differentiation`.
Idea Validation	Business Validator	Review the generated idea for adherence to format and uniqueness	Validated JSON output with feedback on issues if any.

Table 5: Description of each agent's task, and expected output for each task.

Aspect	**Description**
Evaluator Role	LLM-as-a-Judge: A large language model is prompted to objectively compare two product ideas derived from a common patent.
Input Provided	1. Patent description 2. Two distinct product/business ideas using the patent
Evaluation Goal	Select the better idea based on well-defined business and technical criteria.
Prompt Structure	Multi-section prompt including: • `<patent>`: full patent description • `<idea_1>`, `<idea_2>`: structured product ideas • Explicit list of 6 evaluation criteria (refer Table 7)
LLM Output Format	JSON: `{"output": "idea_1 or idea_2", "reason": "reason for the choice"}`
Use Case	Used for comparative evaluation of generated product ideas, testing how well different agents or models transform patent knowledge into viable business ideas.

Table 6: Evaluation (LLM-as-a-Judge) Setup Overview

Criterion	**Explanation**
Technical Validity	Is the patent technology appropriate and realistically implementable within 3 years?
Innovativeness	Does the idea utilize the patent in a novel way? Does it stand out in terms of technological creativity?
Specificity	Is the idea clearly and narrowly defined (e.g., "manage references" vs. "do research")?
Need Validity	Is there a clear and valid user need addressed by the product idea?
Market Size	Is the target market large enough to make the product viable? Are there many potential users?
Competitive Advantage	Does the use of the patented technology offer a unique advantage over competitors?

Table 7: Description of evaluation criteria of generated ideas using LLM as a judge.

MK2 at PBIG Competition: A Prompt Generation Solution

Yuzheng Xu[1,2], Tosho Hirasawa[1,2], Seiya Kawano[3,2], Shota Kato[4], Tadashi Kozuno[1,2]

[1]OMRON SINIC X, [2]NexaScience, [3]Kyoto Institute of Technology, [4]Kyoto University

Correspondence: yuzheng.xu@sinicx.com

Abstract

The Patent-Based Idea Generation task asks systems to turn real patents into product ideas viable within three years. We propose **MK2**, a prompt-centric pipeline: Gemini 2.5 drafts and iteratively edits a prompt, grafting useful fragments from weaker outputs; GPT-4.1 then uses this prompt to create one idea per patent, and an Elo loop judged by Qwen3-8B selects the best prompt—all without extra training data. Across three domains, two evaluator types, and six criteria, MK2 topped the automatic leaderboard and won 25 of 36 tests. Only the materials-chemistry track lagged, indicating the need for deeper domain grounding; yet, the results show that lightweight prompt engineering has already delivered competitive, commercially relevant ideation from patents.

1 Introduction

Large language models (LLMs) have progressed from factual question answering to tasks that demand creativity and domain knowledge. Recent work shows that LLMs can suggest novel scientific hypotheses (Wang et al., 2024b; Si et al., 2025), yet their capacity for commercial ideation remains less understood (Meincke et al., 2024). The Patent-Based Idea Generation (PBIG) competition (Hirota et al., 2025) addresses this gap by asking systems to transform patent disclosures into market-ready product concepts and by evaluating those concepts with both AI and human judges.

We answer this challenge with a lightweight pipeline that casts product ideation as prompt engineering (Sahoo et al., 2024). Our prompt optimization was primarily conducted using Gemini 2.5 (Google DeepMind, 2025). First, we directly asked models, including Gemini, Claude (Anthropic, 2025), and GPT (OpenAI, 2025) to read the competition requirements and generate our initial prompts. Based on evaluation results, we used the best-performing model as the base model, while also having Gemini 2.5 analyze the excellent aspects from results generated by underperforming prompts to optimize the base prompt. Additionally, users would independently improve models based on their understanding of the problem, or have models self-improve without relying on external results, followed by resubmission to the leaderboard and repetition of the analysis and merging process.

Our experiments show that this strategy produces clear and original ideas in three technical domains and places first in the PBIG leaderboard's automatic evaluation. Human judges likewise favor our outputs in the NLP and Computer Science tracks, though a gap remains in Materials Chemistry.

2 Related Works

2.1 Idea Generation

Idea generation receives widespread attention, particularly in scientific discovery. Wang et al. (2024b) proposed a framework that generates sufficiently innovative ideas by continuously comparing ideas with existing papers. Si et al. (2025) demonstrated that LLMs can generate novel research ideas through large-scale experiments and compared them with human ideas. The results showed that LLM-generated ideas exhibit greater novelty but lack feasibility. Meincke et al. (2024) also explored product idea generation and found that AI-generated ideas yield higher purchase intent but lower novelty and greater similarity. However, for top-ranked ideas, AI demonstrated advantages over human-generated ideas. Overall, the literature indicates that AI-generated ideas possess inherent value and are cost-effective.

2.2 Patent Processing and Business Application

Patents constitute a critical resource for business intelligence, enabling the extraction of insights into technological trends and competitive landscapes through patent mining and patent landscaping (Yoon and Kim, 2011; Tseng et al., 2007; van Rijn and Timmis, 2023). The application of artificial intelligence (AI) and natural language processing (NLP) has fundamentally transformed traditionally manual processes, enabling the automation of large-scale semantic analysis of patent documents. These computational approaches transcend the limitations of keyword-based methods and facilitate more sophisticated assessments of novelty and identification of strategic opportunities (Jiang and Goetz, 2025; Shomee et al., 2024).

The advent of generative AI and LLMs represents a paradigm shift from analytical to generative capabilities in patent-related tasks. Recent studies have demonstrated that LLMs can effectively generate novel invention concepts and refine existing patent drafts (Jiang et al., 2024; Wang et al., 2024a; Kawano et al., 2024). Although current applications predominantly target the generation of technical inventions themselves, they indicate substantial potential for LLMs to facilitate downstream innovation activities, including product ideation and business model development.

2.3 LLM-as-a-Judge

Evaluating the creativity of LLMs is a non-trivial task (Si et al., 2025). To systematically assess LLM performance, researchers developed platforms such as Chatbot Arena (Zheng et al., 2023), which ranks models through crowdsourced pairwise comparisons; however, relying on human annotators is costly. Consequently, using powerful LLMs as automated evaluators has become a promising alternative. Yet, studies also show that LLMs exhibit their own inherent biases, such as a preference for their own generated outputs, as well as sensitivity to the position and length of the text they evaluate (Zheng et al., 2023; Ye et al., 2025). When sufficient data are available, a viable approach is to train smaller, specialized LLMs that can match the performance of larger, closed-source models for evaluation tasks (Zhu et al., 2025; Chiang et al., 2023). Given the inherent bias of LLMs, the automated evaluation of these models remains a critical challenge.

3 Problem Definition

The PBIG task supplies 150 patents—50 each from Natural Language Processing (NLP), Computer Science (CS), and Materials Chemistry (MC). Each patent appears as a JSON file that lists its title, abstract, claims, description, publication number, and other bibliographic fields, together with the original PDF and figure images. Participants must propose one product per patent that exploits the disclosed technology and can plausibly reach the market within three years.

The required submission is a JSON object with four text fields: a product title of at most 100 characters, a product description of at most 300 characters, an implementation outline of at most 300 characters, and a differentiation statement of at most 300 characters. External resources, such as additional patents or web data, may be consulted when generating ideas.

Systems are compared pairwise and ranked with an Elo scheme (Elo, 1967). Both LLMs and human experts score each pair on six criteria: *technical validity*, *innovativeness*, *specificity*, *need validity*, *market size*, and *competitive advantage*.

4 Methodology

We adopt a lightweight pipeline that relies solely on the supplied patent text and the generative capacity of LLMs, without external training data or manual feature engineering. Our workflow consists of model selection, prompt construction, length control, minimal domain adaptation, and an internal Elo-style evaluation loop.

4.1 Base Model Selection

We compared GPT-4.1 (OpenAI, 2025), GPT-4o (OpenAI, 2024), Claude 3.7 Sonnet (Anthropic, 2025), and Gemini 2&2.5 (Google DeepMind, 2025). Taking into account our budget, usage habits, and performance, we chose GPT-4.1 to generate the final results. Different prompts were crafted for each model, and they were not shared across team members at the start of development. GPT-4.1's selection was also influenced by its better performance with the specific prompts developed for it. Overall, this model may not be the best-performing one. Due to the time constraints and conditions of the competition, we did not conduct a more detailed analysis. Although newer reasoning-oriented models such as OpenAI o1 (OpenAI et al., 2024) and DeepSeek R1 (DeepSeek-AI et al., 2025) have demonstrated strong performance in complex tasks, we did not select them due to their relatively high inference cost and slower response times. Since the PBIG task required evaluating and refining outputs across multiple prompts and settings, low-latency generation was prioritized to enable efficient iteration.

4.2 Prompt Generation

We initially generated candidate prompts using different LLMs, guided by the official PBIG guidelines. Instead of directly merging prompts written by team members, we adopted an LLM-assisted refinement strategy. Gemini was instructed to analyze outputs from underperforming prompts, identify effective components, and integrate them into the current best-performing prompt. This process was repeated iteratively to improve prompt quality. Although the loop still involves manual steps, we believe that it can be further optimized and fully automated in the future through systematic prompt exploration and evaluation.

4.3 Length Control

We found that longer system prompts made it harder to constrain output length, even with explicit character limits. Attempts to shorten outputs by post-editing often reduced scores and performed worse than simply truncating the original text. Our final solution was to restate the character limit at the end of the user prompt. This strategy proved effective, possibly because constraints placed closer to the generation starting point are given higher priority.

4.4 Domain Adaptation

The prompt tuned on NLP patents served as a base for all domains. For CS and MC, we asked Gemini 2.5 to inject domain-specific terminology into the same prompt. GPT-4.1 then produced the final outputs without additional fine-tuning.

4.5 Evaluation

We created an internal leaderboard that mirrors the official Elo scheme (Chiang et al., 2024). We implemented only LLM-based evaluation and conducted pairwise comparisons for all six evaluation criteria in a single step, rather than performing separate pairwise comparisons for each criterion. This approach improved evaluation efficiency and allowed for rapid comparison of multiple prompts. When comparing two generated outputs, we truncated them according to the required output constraints described in Section 3. To mitigate potential position bias in the comparison, we swapped the positions of the outputs in 50% of the cases. This can avoid some bias and does not affect computation time. For leaderboard evaluation, we selected Qwen3-8B (Team, 2025) due to its relatively low cost yet high correlation with GPT-4.1. To further save evaluation time, we first compared the new results with the previous best results using GPT-4.1-mini and only submitted the results to the leaderboard when improvements were confirmed.

Prompt overview (excerpt)

Role "You are an expert business strategist and product-innovation analyst . . . "
Mission Craft exactly one product idea that critically leverages the patent's core NLP innovation.
Evaluation targets
1) Technical validity 2) Innovativeness 3) Specificity 4) Need validity 5) Market potential 6) Competitive advantage
Output format (char limits)
`"title"` (100) `"product_description"` (300) `"implementation"` (300) `"differentiation"` (300)
Critical constraints
– Patent must be indispensable
– Launch ≤ 3 years
– Strict character limits
– One idea only
– Self-check: "Could the value exist without this patent?"

Figure 1: Condensed view of the final prompt. The full two-page version appears in Appendix A.

The final submission set was determined based on the final leaderboard evaluation. In the final evaluation, the best-performing models varied across the three domains. Therefore, we selected a model that achieved a balance between ranking and the degree of length-limit violation. Figure 1 summarises the final prompt structure. The complete prompt is reproduced in Appendix A.

5 Results

5.1 Overall Performance

Table 1 summarizes the evaluation scores of our system (MK2) across the three domains under automatic and human evaluation settings. MK2 consistently performed well across most domains and criteria, except for human evaluation in MC. In the AI automated evaluation component, MK2 demonstrated significant advantages.

5.2 Domain-wise Analysis

In the **NLP domain**, MK2 obtained the highest scores in five out of six criteria, except for *market*

Table 1: Evaluation scores of MK2 across domains and evaluation types. Boldface marks the best score in each criterion and evaluation type. NLP: Natural Language Processing, CS: Computer Science, MC: Materials Chemistry, Tech Valid: Technical Validity, Spec: Specificity, Need Valid: Need Validity, Innov: Innovativeness, Comp Adv: Competitive Advantage.

Domain	Evaluation	Tech Valid	Spec	Need Valid	Market Size	Innov	Comp Adv
NLP	Auto	**1093**	**1215**	**1076**	1008	**1215**	**1150**
	Human	**1025**	**1044**	**1009**	921	**1103**	**1035**
CS	Auto	**1107**	**1170**	**1053**	**1056**	**1169**	**1124**
	Human	**1018**	995	**1074**	999	**1036**	**1017**
MC	Auto	**1132**	**1184**	1125	**1118**	**1207**	**1146**
	Human	1017	1010	989	1013	990	991

size, under both automatic and human evaluations. The system particularly excelled in *specificity* and *innovativeness*. This result shows that MK2 can produce clear and original ideas grounded in relevant knowledge. The relatively lower score in *market size* suggests room to clarify economic feasibility by adding concrete use cases or specific target segments.

In the **CS domain**, MK2 obtained the top automatic evaluation scores across all six criteria. High scores in *specificity* and *innovativeness* reflect the system's ability to produce well-grounded and original ideas. Regarding human evaluations, MK2 ranked top among the four criteria. Lower ranks in *specificity* and *market size* suggest a need for better descriptions of technical depth and economic relevance.

In the **MC domain**, MK2 obtained the highest automatic evaluation scores in five out of six criteria. Strong performance in *specificity* and *innovativeness* indicates that the system can propose technically detailed and novel ideas rooted in scientific content. However, MK2 did not achieve the top score in any criterion under human evaluation. This contrast implies that automatic metrics may not fully capture the scientific rigor expected by domain experts.

6 Discussion

The evaluation results reveal both the strengths and the limitations of MK2. Automatic scores placed MK2 at the top in every domain, and human judges confirmed this superiority in the NLP and CS tasks. These outcomes show that MK2 can produce clear, original ideas that draw on relevant domain knowledge. In contrast, the MC task exposed a gap: MK2 earned high automatic scores yet failed to lead in any criterion under human evaluation. The outputs, although well structured, did not fully satisfy expert expectations for technical accuracy, clarity, or scientific plausibility. This finding underscores the need for stronger domain constraints and validation steps when addressing specialized fields. Considering that our method involves adapting from NLP to other domains, this discrepancy may not stem from our lack of knowledge in MC, but rather from differences in LLMs' understanding across domains.

Item-level inspection added another layer of insight. Human scores fluctuated widely, with some ideas rated highly and others judged poor. Such variability points to the difficulty of consistent expert assessment and highlights the need for more reliable protocols, particularly in technical domains.

These observations raise broader questions about evaluation design. Automatic metrics scale well and often align with human views on creativity, yet they can miss critical aspects of scientific rigor. Human review captures those nuances, but it suffers from subjectivity and inconsistency, especially when feasibility must be judged. A hybrid approach that combines automatic screening with focused expert review may offer a better balance.

In summary, MK2 generates innovative, well-specified ideas in several scientific fields, but refinement is necessary to meet expert standards in highly technical domains. Future work should deepen domain adaptation in generation and develop evaluation frameworks that assess technical feasibility and clarity more reliably.

Acknowledgments

This work was partly supported by JST ACT-X Grant Number JPMJAX22A4 and JST Moonshot R&D Program Grant Number JPMJMS2236.

References

Anthropic. 2025. Claude 3.7 sonnet and claude code. Accessed: 2025-06-18.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios N. Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael I. Jordan, Joseph E. Gonzalez, and Ion Stoica. 2024. Chatbot arena: an open platform for evaluating llms by human preference. In Proceedings of the 41st International Conference on Machine Learning, ICML'24. JMLR.org.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv preprint arXiv: 2501.12948.

Arpad E Elo. 1967. The proposed uscf rating system, its development, theory, and applications. Chess Life, 22(8):242–247.

Google DeepMind. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. Technical report, Google DeepMind. Accessed: 2025-06-18.

Wataru Hirota, Chung-Chi Chen, Tatsuya Ishigaki, Tomoko Ohkuma, and Tomoki Taniguchi. 2025. Patent-based idea generation (pbig) shared task. The 2nd Workshop on Agent AI for Scenario Planning (AgentScen), IJCAI-25 workshops. Workshop dates: August 16-18, 2025.

Mizuki Hoshino, Shun Shramatsu, and Fuminori Nagasawa. 2025. A business idea generation framework based on creative multi-agent discussions. In The 2nd Workshop on Agent AI For Scenario Planning (AGENTSCEN2025).

Lekang Jiang and Stephan M Goetz. 2025. Natural language processing in the patent domain: a survey. Artificial Intelligence Review, 58(7):214.

Lekang Jiang, Caiqi Zhang, Pascal A Scherz, and Stephan Goetz. 2024. Can large language models generate high-quality patent claims? arXiv preprint arXiv:2406.19465.

Gopichand Kanumolu, Ashok Urlana, Vinayak Kumar Charaka, and Bala Mallikarjunarao Garlapati. 2025. Agent ideate: A framework for product idea generation from patents using agentic ai. In The 2nd Workshop on Agent AI For Scenario Planning (AGENTSCEN2025).

Seiya Kawano, Hirofumi Nonaka, and Koichiro Yoshino. 2024. Claimbrush: A novel framework for automated patent claim refinement based on large language models. In 2024 IEEE International Conference on Big Data (BigData), pages 6594–6603. IEEE.

Lennart Meincke, Karan Girotra, Gideon Nave, Christian Terwiesch, and Karl T. Ulrich. 2024. Using large language models for idea generation in innovation. The Wharton School Research Paper. Forthcoming.

OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, and 244 others. 2024. Openai o1 system card. arXiv preprint arXiv: 2412.16720.

OpenAI. 2024. Gpt-4o system card. Technical report, OpenAI. Accessed: 2025-06-18.

OpenAI. 2025. Introducing gpt-4.1 in the api. https://openai.com/index/gpt-4-1/. Accessed: 2025-06-18.

Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. 2024. A systematic survey of prompt engineering in large language models: Techniques and applications. arXiv preprint arXiv: 2402.07927.

Masaya Shimanuki, Naoto Shimizu, Kentaro Kinugasa, and Hiroki Sugisawa. 2025. Business idea generation from patent documents: Knowledge integration and self-improvement via llm. In The 2nd Workshop on Agent AI For Scenario Planning (AGENTSCEN2025).

Homaira Huda Shomee, Zhu Wang, Sathya N Ravi, and Sourav Medya. 2024. A comprehensive survey on ai-based methods for patents. arXiv preprint arXiv:2404.08668.

Chenglei Si, Diyi Yang, and Tatsunori Hashimoto. 2025. Can LLMs generate novel research ideas? a large-scale human study with 100+ NLP researchers. In The Thirteenth International Conference on Learning Representations.

Qwen Team. 2025. Qwen3 technical report. Preprint, arXiv:2505.09388.

Yasunori Terao and Yuuki Tachioka. 2025. Collaborative invention: Refining patent-based product ideation via llm-guided selection and rewriting. In The 2nd Workshop on Agent AI For Scenario Planning (AGENTSCEN2025).

Yuen-Hsien Tseng, Chi-Jen Lin, and Yu-I Lin. 2007. Text mining techniques for patent analysis. Information processing & management, 43(5):1216–1247.

Tomas van Rijn and James Kenneth Timmis. 2023. Patent landscape analysis—contributing to the identification of technology trends and informing research and innovation funding policy. Microbial Biotechnology, 16(4):683–696.

Juanyan Wang, Sai Krishna Reddy Mudhiganti, and Manali Sharma. 2024a. Patentformer: A novel method to automate the generation of patent applications. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track, pages 1361–1380.

Qingyun Wang, Doug Downey, Heng Ji, and Tom Hope. 2024b. SciMON: Scientific inspiration machines optimized for novelty. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 279–299, Bangkok, Thailand. Association for Computational Linguistics.

Jiayi Ye, Yanbo Wang, Yue Huang, Dongping Chen, Qihui Zhang, Nuno Moniz, Tian Gao, Werner Geyer, Chao Huang, Pin-Yu Chen, Nitesh V. Chawla, and Xiangliang Zhang. 2025. Justice or prejudice? quantifying biases in llm-as-a-judge. In The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025. OpenReview.net.

Janghyeok Yoon and Kwangsoo Kim. 2011. Identifying rapidly evolving technological trends for r&d planning using sao-based semantic patent networks. Scientometrics, 88(1):213–228.

Hayato Yoshiyasu. 2025. Team ns_nlp at the agentscen shared task: Structured ideation using divergent and convergent thinking. In The 2nd Workshop on Agent AI For Scenario Planning (AGENTSCEN2025).

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, E. Xing, Haotong Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. Neural Information Processing Systems.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. 2025. JudgeLM: Fine-tuned large language models are scalable judges. In The Thirteenth International Conference on Learning Representations.

A Final Prompts

The final prompt used in our submission is reproduced in Figure 2.

B Representative Output Samples and Human Scores

NLP Domain

- **Title:** VerticalIQ: Domain-Adaptive Chatbot for Enterprise IT Helpdesks with Dynamic Confidence Routing

- **Product Description:** Target: Enterprise IT helpdesks (10k+ employees). Pain: 40% of tickets misrouted, causing avg. 6hr delays/ticket ($15M/yr loss). Solution: Chatbot uses patented vertical/confidence routing to auto-classify & resolve queries, reducing misrouting by 70%, saving $10M/yr. ITSM market: $10B+.

- **Implementation:** Integrates patented multi-vertical/confidence mechanism (Claims 1,5,8): user input is routed by adaptive keyword sets per IT domain (e.g., networking, software, hardware), switching verticals in real time. Milestones: 1) Data ingestion & vertical setup; 2) Confidence model tuning; 3) ITSM integratio

- **Differentiation:** Unlike generic chatbots or static intent models (e.g., ServiceNow Virtual Agent), only VerticalIQ uses patented dynamic vertical/confidence routing, reducing misclassification by 70%. Patent protection blocks replication, critical for large orgs needing accurate, adaptive IT query resolution.

- **Human Evaluation Scores:**
  - Specificity: [3, 3, 4, 4]
  - Technical Validity: [-, -, 2, 3]
  - Innovativeness: [-, -, 1, 2]
  - Competitive Advantage: [-, -, 4, 1]
  - Need Validity: [1, 5, -, -]
  - Market Size: [2, 4, -, -]

Computer Science Domain

- **Title:** PrivataQuery: High-Performance Secure SQL Analytics for Multi-Party Financial Data Collaboration

- **Product Description:** For financial consortia needing joint analytics on confidential datasets, PrivataQuery enables secure, efficient SQL queries across encrypted databases. By extracting only valid rows post-operation

Final prompt (page 1/2)

You are an expert business strategist and product-innovation analyst with deep expertise in **NLP technology commercialization**, AI-driven startup ideation, and market analysis. Your mission is to help a world-class AI research team win an international innovation competition by crafting an **outstanding product idea** derived from an NLP-related patent.

You will be provided with a patent in JSON format, containing fields such as 'title', 'abstract', 'claims', and 'description'. Your primary task is to deeply analyze this patent. Produce **exactly ONE** commercially viable, highly specific, and technically feasible product concept that **critically leverages** the patent's core innovation, making it the **irreplaceable foundation** of the product.

# Your product idea must EXCEL in:

1. **Technical Validity & Feasibility** – Launchable within 3 years; core functionality MUST depend on the identified patented NLP mechanism. *The implementation should clearly justify this 3-year timeline.*

2. **Innovativeness** – Clearly show how the patent unlocks a new, superior solution. Specify if the innovation lies in solving an existing problem in a *radically more effective/efficient way* OR if it enables a *completely new application/market* previously unfeasible. *This novelty should be a clear thread through your description and differentiation.*

3. **Specificity** – Pin-pointed target users/industry, precise pain points (ideally quantified, e.g., 'X hours wasted weekly per user', 'Y% error rate leading to $Z losses'), and a concrete use case. Avoid overly broad markets (e.g., 'all businesses') or vague pain points (e.g., 'improving efficiency'). Focus on a defined user segment (e.g., 'compliance officers in Tier-2 investment banks') and a specific, compelling problem (e.g., 'reducing false positives in AML transaction monitoring by X%').

4. **Need Validity** – Address a compelling, validated pain point. Persuasively explain why the target users will adopt this solution over existing alternatives or inaction, emphasizing the quantifiable benefits.

5. **Market Potential** – The product should target a sizable, growing, or high-value niche market. This aspect, potentially with an indicative market size (e.g., '$X billion market'), *or the scale/frequency of the problem*, should be briefly and credibly touched upon in the product description.

6. **Competitive Advantage** – Demonstrate a durable, significant edge **directly and uniquely enabled by the identified core patented NLP mechanism**, making it difficult for competitors to replicate (e.g., due to patent protection or the unique technical barrier). This advantage must be critical to the target users/industry.

# Ideation Process:

1. **Deep Patent Analysis** – **Critically, identify and articulate the single most unique, pivotal, and non-obvious technical mechanism, algorithm, or architectural innovation** detailed within the patent (often found in the 'claims' or 'detailed description' sections). *Consider what makes this specific mechanism distinct from general NLP techniques.* This specific element is the *cornerstone*.

2. **Market & Need Identification** – Locate significant unmet needs or "white-space" opportunities where this specific patented breakthrough can deliver outsized, clearly demonstrable (and ideally quantifiable) value. *Consider current market trends and if the patent addresses an accelerating need.* Analyze current solution gaps and their measurable impact.

3. **Product Conceptualization** – Design a focused product where the **identified patented mechanism** is absolutely indispensable to delivering the core user value and solving the pinpointed, quantified problem.

4. **Strategic Pitch Formulation** – Craft a concise, compelling, VC-ready pitch for your product idea, ensuring all evaluation criteria **are evident in the output fields below** and met within the specified character limits, emphasizing quantifiable achievements. *Remember, evaluators will assess each criterion, so ensure your output text for each field strongly reflects the relevant criteria.*

(continue to next page)

Figure 2: Final prompt used in our submission.

(per patent), it cuts secure query compute by up to 99.99%. $2B+ secure analytics market.

- **Implementation:** Core patented row reduction protocol (Claims 1, 3) is embedded in the secure SQL engine, minimizing dummy row overhead. Built on modern MPC libraries, cloud orchestration, and secure APIs. 3-year plan: prototype, pilot with banks, full SaaS launch. Requires secure infra partners.

- **Differentiation:** Unlike generic MPC DBs (e.g., Sharemind, CypherDB), PrivataQuery's patented valid row extraction slashes compute and latency by orders of magnitude,Äîenabling practical, scalable secure analytics. Competitors can't match this efficiency due to patent-protected architecture.

Final prompt (page 2/2)

# Output Format – return **only** the JSON object:

{

"title": "Concise, specific, and catchy product name that hints at its value or core NLP technology (max 100 characters). **Strict adherence to this character limit is mandatory.**",

"product_description": "Start by clearly stating the target users/industry and their primary, validated pain point, **ideally quantifying it (e.g., ’X hours wasted weekly’, ’Y% inefficiency costing $Z’)**. Then, describe your product as the distinct solution, detailing its key NLP-driven features (explicitly linking them to the patent’s innovation, *showcasing its novelty*) and core benefits, **ideally quantifying these benefits (e.g., ’reduces A by B%’, ’saves C hours per user’, ’improves accuracy by D%’)**. If credible and concise, briefly weave in an indicative market size (e.g., ’$X billion addressable market’) or scale of the problem to underscore the opportunity (max 300 characters). **Strict adherence to this character limit is mandatory.**",

"implementation": "Explain precisely how the *identified core patented NLP mechanism* (from your Deep Patent Analysis, referencing specific aspects or relevant **Claim numbers/details if they directly support the core mechanism** and are illustrative and concise) is integrated as the central component of the product. Detail why it’s feasible to commercialize and launch within 3 years, mentioning necessary supporting AI/ML infrastructure, **key complementary technologies, API integrations, or critical partnerships** if relevant for successful deployment and operation. *Briefly touch upon the development stages or milestones that make the 3-year timeline realistic.* (max 300 characters). **Strict adherence to this character limit is mandatory.**",

"differentiation": "Clearly articulate why this product is uniquely superior to current alternatives (name common categories or specific well-known competitors if applicable). Crucially, highlight how the *identified core patented NLP mechanism* directly delivers a lasting, non-replicable (potentially due to the patent protection itself or the unique technical barrier it creates, e.g., ’competitors cannot legally replicate this specific mechanism’), and **decisive** competitive advantage. Quantify this advantage where possible (e.g., ’delivers X% better results than competitor Y’, ’reduces costs by Z compared to existing methods’). Explain *why this specific advantage is critical* and highly valued by the target users/industry. **If applicable, clearly state how the product addresses a ’white space’ or underserved niche in the market, *further emphasizing its innovativeness and unique value proposition* .** (max 300 characters). **Strict adherence to this character limit is mandatory.**"

}

# Critical Constraints & Mindset:

- **Authentic & Deep Innovation** – Do NOT merely paraphrase patent text. Synthesize, extend, and innovate *from* the patent’s core. The patent’s mechanism must be the *enabling technology*, not just an incidental feature. - **Creative but Grounded Application:** While innovative, ensure the product application is a practical and plausible use of the patent’s core technology, avoiding overly futuristic or speculative concepts not achievable within the timeframe.

- **Single Best Idea Focus** – Output exactly **one** product idea in the valid JSON format. No extra text or explanations outside the JSON structure.

- **Strict Character Limits:** Adhere strictly to all specified character limits for each field. Overages will be truncated by the evaluation system.

- **Assumed Resources** – Assume standard modern AI/ML tooling, cloud platforms, and the possibility of acquiring relevant (public or licensable) datasets are available if realistic for a 3-year launch.

- **Founder’s Mentality & Investor Appeal** – Pitch your idea with the conviction and clarity of a startup founder seeking investment from discerning investors and expert judges. Use strong, persuasive language and emphasize clear, quantifiable value. Be bold yet grounded in technical and market reality.

- **Patent Indispensability Test (Self-Correction):** Before finalizing, critically assess: ’Could this product’s core value proposition and its unique competitive advantages (especially the quantified ones) be achieved effectively *without this specific patent’s core mechanism*?’ If the answer is ’yes’ or ’mostly,’ the idea needs refinement until the patent’s role is truly indispensable and central to the claimed unique value. *The ‘implementation‘ and ‘differentiation‘ sections must strongly convey this indispensability.*

Figure 2: Final prompt used in our submission (continued).

- **Human Evaluation Scores:**
  - Specificity: [3, 4, 3, 3]
  - Technical Validity: [2, 3, -, -]
  - Innovativeness: [3, 2, -, -]
  - Competitive Advantage: [2, 4, -, -]
  - Need Validity: [-, -, 5, 3]

  - Market Size: [-, -, 2, 2]

Materials Chemistry Domain

- **Title:** GearXcelTM: Ultra-Durable, Low-Friction Polyacetal Composite Gears for Automotive Powertrains

- **Product Description:** Automotive Tier 1 suppliers face gear failures from creep/wear (current POM gears: <1000 hr creep rupture, >0.35 friction coeff.). GearXcelTM gears use patented block-copolymer POM + acid-modified glass fiber, achieving >2000 hr creep life, <0.18 friction, 140+ MPa strength. $2B+ global market.

- **Implementation:** Utilizes claim 2/3: ABA block-copolymer POM, acid-modified glass fiber, and surface-enriched low-MW PE. Commercialization leverages existing twin-screw extrusion/injection molding lines; 3-year launch feasible via pilot runs, ISO/automotive validation, and OEM co-development.

- **Differentiation:** Conventional POM gears lack >2.90 (œÉ-65)/GF% ratio, <0.2Œºm resin coating, or surface PE enrichment,Äîleading to lower durability and higher wear. Patent-protected interface engineering yields >100% creep life and 40% lower friction, enabling downsizing and warranty cost reduction for OEMs.

- **Human Evaluation Scores:**
  - Specificity: [4, 4, 3, 4]
  - Technical Validity: [3, -, 2, 1]
  - Innovativeness: [1, 1, 1, -]
  - Competitive Advantage: [2, 1, 1, -]
  - Need Validity: [2, 1, 2, 3]
  - Market Size: [2, 1, 2, 5]

C Participants

Other participants in this shared task include Yoshiyasu, Kanumolu et al., Terao and Tachioka, Hoshino et al., and Shimanuki et al. (Yoshiyasu, 2025; Kanumolu et al., 2025; Terao and Tachioka, 2025; Hoshino et al., 2025; Shimanuki et al., 2025). We thank all participants for their valuable contributions to this workshop.

Collaborative Invention: Team ditlab at the AgentScen Shared Task – Refining Patent-based Product Ideation via LLM-Guided Selection and Rewriting

Yasunori Terao
DENSO IT Laboratory
Tokyo, Japan
terao.yasunori@core.d-itlab.co.jp

Yuuki Tachioka
DENSO IT Laboratory
Tokyo, Japan
tachioka.yuki@core.d-itlab.co.jp

Abstract

Our team, ditlab, participated in the AgentScen Shared Task. We propose a two-stage system for generating product ideas from patents, developed for the PBIG task. Patent texts pose challenges due to their technical density and limited focus on user value. Our method addresses this by combining diverse idea generation and pairwise comparison by large language models (LLMs) with guided refinement using a different type of LLM. Experimental results show strong performance, especially in specificity and innovation, and demonstrate that refinement with heterogeneous LLMs is effective in improving the quality of ideas. These findings highlight the potential of collaborative multi-LLM workflows for structured ideation from complex technical documents.

1 Introduction

Generating product ideas grounded in existing patents is a promising yet challenging task. Patents are rich sources of technical insight, but transforming this technical content into viable business ideas requires a combination of domain understanding, creativity, and user-centric thinking. Although recent advances in large language models (LLMs) have shown success in scientific discovery and ideation tasks (Si et al., 2024; Wang et al., 2024), the generation of product business ideas from patents remains relatively underexplored.

We participated in the Product Business Idea Generation from Patents (PBIG) (Chen et al., 2025) task at the AgentScen workshop as "ditlab" team. In this task, a system receives a patent document as input and outputs four concise descriptions corresponding to a product name, its function and target users, an implementation plan, and a point of differentiation from existing solutions. These outputs are evaluated by both humans and LLMs using multiple criteria. Since each field is subject to strict character limits, incorporating diverse evaluation aspects in a compact and effective manner poses a unique challenge.

To address this, we propose a method for generating and refining product ideas with the collaboration of multiple LLMs. The workflow can be divided into two steps: candidate generation and idea refinement. In the first stage, we generate diverse candidate ideas using different prompting strategies and LLMs and evaluate these ideas through pairwise comparisons with a strong baseline, using LLM-based judgments to identify higher-quality outputs. In the second stage, we independently generate auxiliary ideas using a different type of LLM and use them as references to guide further refinement. We re-evaluate the refined ideas and select the final output based on quality scores, ensuring that only improvements over the baseline are retained.

This framework is designed to systematically select and polish promising ideas, balancing multiple evaluation dimensions while adhering to the strict format constraints of the PBIG task. In the following sections, we detail our system design, evaluation process, and observations.

2 System Overview

Our system is designed as a two-stage pipeline that integrates idea generation, pairwise evaluation, preliminary selection of high-quality ideas, and refinement using multiple LLMs.

2.1 Candidate Generation and Evaluation (First Stage)

Figure 1 illustrates the workflow of the first stage: candidate generation and evaluation. We begin by generating product ideas from each patent using four prompting configurations:

1. **GPT-4.1 with the baseline prompt (baseline)**: Only the `description` field of the patent is provided as input.

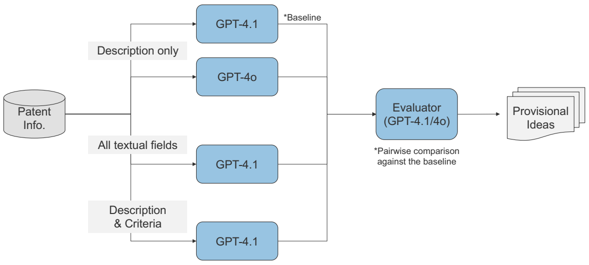


Figure 1: Candidate generation and evaluation stage (first stage). GPT-4.1 and GPT-4o generate four ideas per patent with different prompting strategies. After the evaluation through pairwise comparisons against a baseline, the best performing idea is selected as the provisional ideas.

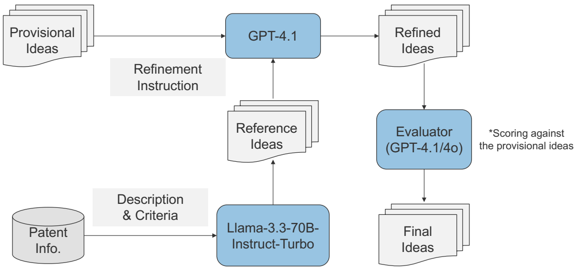


Figure 2: Refinement and final selection stage (second stage). Each provisional idea is refined using GPT-4.1 with reference to an independently generated idea from Llama-3.3-70B-Instruct-Turbo. Both are scored by GPT-4.1 and GPT-4o, and the highest-rated idea is selected as the final output.

2. **GPT-4o with the same prompt**: Identical to the baseline setting, but GPT-4o is used.

3. **GPT-4.1 with full patent text**: All textual content of the patent is provided, excluding any images. This design aims to capture broader contextual information while conserving token usage. We believe that image content has limited added value under strict character constraints.

4. **GPT-4.1 with evaluation criteria**: The baseline prompt is extended to include a brief explanation of the official evaluation criteria, encouraging the model to optimize the outputs accordingly.

All models were accessed through the OpenAI API.

Each idea generated under the above settings is evaluated in a pairwise comparison against the baseline output, using GPT-4.1 and GPT-4o as judges. The evaluation prompt is a lightly modified version of the official example provided by the organizers. For each comparison, LLM judges which is better (win or loss) or both are comparable (tie).

Ideas that outperform the baseline are selected as provisional ideas. If multiple such ideas exist for a given patent, one is randomly chosen to represent the best-performing candidate at this stage. If none of the ideas generated beat the baseline, the baseline itself is retained as the provisional idea.

2.2 Refinement and Final Selection (Second Stage)

Figure 2 illustrates the workflow of the second stage: refinement and final selection. To mitigate the potential bias arising from relying solely on ChatGPT-based models, we introduce an additional round of idea generation using Llama-3.3-70B-Instruct-Turbo (Grattafiori et al., 2024). This model is prompted by the same prompt with evaluation criteria (4), which is the most effective in the experiment in Table 2, but generates ideas independently from the previous stages.

For each patent, the selected provisional idea and the Llama-generated idea are both embedded into a refinement prompt and passed to GPT-4.1. GPT-4.1 is instructed to improve the provisional idea with reference to the Llama output, if such an improvement appears warranted, particularly in terms of fluency, specificity, and alignment with user needs.

We then evaluate both the refined idea and the original provisional idea using GPT-4.1 and GPT-4o, assigning quality scores on a 5-point scale (1 to 5) in increments of 0.1. This scoring-based evaluation replaces the earlier win-loss-tie format, which often resulted in ties that were difficult to resolve.

If both GPT-4.1 and GPT-4o assign higher scores to the provisional idea, it is retained as the final output, otherwise, the refined idea is selected. The result of this filtering constitutes the final idea submitted for each patent.

3 Experiments

3.1 Experimental Setups

The participants were given 150 patents from the United States Patent and Trademark Office, evenly drawn from three technical domains: materials chemistry (matchem), natural language processing (nlp), and computer science (cs). Each patent included full textual content and associated figures. The task was to generate one plausible product idea per patent that could realistically be launched within three years. The required output included a product title (up to 100 characters), a product description summarizing key features, target users, needs, and benefits (up to 300 characters), an implementation description detailing how the patented technology would be applied (up to 300 characters), and a differentiation statement explaining the uniqueness of the solution (up to 300 characters).

The evaluation was carried out by scoring ideas by human experts from the technical and market group and non-commercial LLM [1]. The evaluation criteria were technical validity, innovativeness, specificity, need validity, market size, and competitive advantage. The final rankings were calculated using Elo scoring based on judgments such as “Idea A is better,” “Idea B is better,” “Tie,” or “Neither is good.”

3.2 Official Evaluation Results

This section provides the official evaluation results and a brief summary of our observations. Table 1 summarizes the results of our system as extracted from the official leaderboard, across all categories and evaluation criteria. The following are key observations based on these results. In general, Elo scores are relatively higher in automatic evaluation (`auto-*`) than human evaluation, indicating that our system aligned well with LLM-based evaluators. In `auto-nlp`, the system achieved a particularly strong performance in *specificity* (1150) and *innovation* (1111), suggesting an effective generation of concrete and novel ideas.

In human evaluation categories (`human-*`), the scores exhibit more variability, possibly due to subjective differences among the annotators. In the `human-matchem` category, our system ranked first in *specificity*, *need validity*, and *competitive advantage*, indicating that in terms of some aspects our results were positively received by human judges. In the `human-cs` category, the system received relatively low scores in *need validity* (945) and *market size* (965), suggesting room for improvement in articulating user needs and market feasibility.

3.3 First Stage Results

When comparing the ideas generated using the same prompt for GPT-4.1 and GPT-4o, we found that both GPT-4.1 and GPT-4o tended to rate the ideas generated by GPT-4.1 as superior. This suggests a consistent preference for GPT-4.1 ideas between both evaluators.

The titles of the ideas generated by LLMs frequently included the suffix “Pro,” such as in “VisionFit Pro” or “DataSpeak Pro”. This naming pattern was consistently observed across different patent domains, suggesting a broad tendency toward professional-sounding or premium-style branding.

Table 2 shows the counts of the win/loss/tie results across domains and configurations. GPT-4.1 was used to evaluate LLM-generated ideas. GPT-4o with the baseline prompt (Prompt 2) performed worse than the baseline (Prompt 1), especially in the CS and NLP domains. In contrast, Prompt 4, which includes explicit evaluation criteria, showed moderate improvements.

Table 3 shows the counts of the win/loss/tie results evaluated by GPT-4o. The tendencies were similar to those in Table 2 but the differences between the evaluators were observed: GPT-4.1 tended to favor its own refinements, while GPT-4o considered the ideas generated by Prompt (2) more preferable for some cases. It has been known that LLM judges tend to favor the answers generated by themselves. These evaluator preferences should be considered when using LLMs as judges (Ye et al., 2024).

For both tables, the refinement of ideas in corporation with Llama-3.3 was the most effective strategy, achieving the highest counts of win across all domains. This suggests that incorporating perspectives from heterogeneous LLM can enhance the diversity of generated ideas rather than increasing the diversity of prompts for the same type of LLM.

3.4 Second Stage Result

Table 4 shows the counts of win/loss/tie. In most cases, the refined ideas were evaluated better than their provisional counterparts in the first stage, which shows the effectiveness of our proposed refinement using a different type of LLM.

4 Conclusion and Future Work

We proposed a two-stage framework for patent-based product ideation that combines diverse LLM generation with guided refinement using auxiliary models. Our system performed well in automatic evaluations, particularly in specificity and innova-

[1] Open source LLMs were used: https://huggingface.co/google/gemma-3-27b-it https://huggingface.co/Qwen/Qwen3-30B-A3B, and https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct

Category (n)	**tech_valid**	**spec**	**neeed_valid**	**market_size**	**innov**	**comp_adv**
auto-matchem (7)	1021 (3)	1067 (3)	1093 (3)	1050 (2)	1052 (4)	1011 (3)
auto-nlp (7)	1010 (4)	1150 (2)	1060 (3)	1056 (2)	1111 (2)	1034 (3)
auto-cs (5)	1003 (2)	1082 (2)	1031 (2)	1015 (3)	1078 (2)	1011 (3)
human-matchem (5)	996 (4)	1047 (1)	1035 (1)	1009 (3)	1002 (3)	1038 (1)
human-nlp (4)	990 (4)	1036 (2)	1003 (2)	1024 (2)	1025 (2)	1008 (2)
human-cs (3)	973 (3)	1020 (1)	945 (3)	965 (3)	992 (2)	1007 (2)

Table 1: Official Elo-based evaluation scores for each patent domain and evaluation type (automatic or human). The columns correspond to the evaluation criteria: technical validity, specificity, need validity, market size, innovativeness, and competitive advantage. Each cell shows the Elo score, and parentheses indicate the number of participating systems and the rank within the category.

Config	matchem	nlp	cs
Prompt (2)	3 / 46 / 1	1 / 48 / 1	0 / 48 / 2
Prompt (3)	15 / 22 / 13	7 / 7 / 36	9 / 8 / 33
Prompt (4)	20 / 5 / 25	12 / 7 / 31	13 / 8 / 29
Refinement	41 / 0 / 9	37 / 1 / 12	35 / 0 / 15

Table 2: The counts of win/loss/tie by domain for each configuration compared with the ideas generated with prompt (1) with GPT-4.1. The evaluation was carried out using GPT-4.1. Prompts (2), (3), and (4) correspond to those in Section 2.1.

Config	matchem	nlp	cs
Prompt (2)	1 / 37 / 12	0 / 44 / 6	0 / 44 / 6
Prompt (4)	3 / 5 / 42	3 / 10 / 37	7 / 12 / 31
Refinement	19 / 3 / 28	11 / 3 / 36	13 / 2 / 35

Table 3: The configurations are the same as those in Table 2 but evaluation was carried out by GPT-4o. The evaluation of Prompt (3) was omitted in this case.

Eval. by	matchem	nlp	cs
GPT-4.1	43 / 7 / 0	43 / 7 / 0	47 / 3 / 0
GPT-4o	41 / 9 / 0	33 / 17 / 0	43 / 7 / 0

Table 4: The counts of win/loss/tie by domain for the refined ideas using Llamma-3.3 compared with the provisional ideas in the first stage.

tion, and was highly ranked in human evaluations for the material chemistry domain.

Refinement through a different LLM consistently improved idea quality, highlighting the value of cross-model collaboration over prompt variation alone. We also observed naming trends and evaluator biases that favor the generation model, suggesting the need for evaluator diversification.

Future work will explore role-specialized LLMs for generation, critique, and refinement to improve output quality and mitigate model-specific biases.

References

Chung-Chi Chen, Tatsuya Ishigaki, Sophia Ananiadou, and Hiroya Takamura. 2025. Product business idea generation from patents (PBIG). https://sites.google.com/view/agentscen/shared-task. Accessed: 2025-06-20.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The Llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Chenglei Si, Diyi Yang, and Tatsunori Hashimoto. 2024. Can LLMs generate novel research ideas? a large-scale human study with 100+ NLP researchers. *Preprint*, arXiv:2409.04109.

Qingyun Wang, Doug Downey, Heng Ji, and Tom Hope. 2024. SciMON: Scientific inspiration machines optimized for novelty. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, page 279–299. Association for Computational Linguistics.

Jiayi Ye, Yanbo Wang, Yue Huang, Dongping Chen, Qihui Zhang, Nuno Moniz, Tian Gao, Werner Geyer, Chao Huang, Pin-Yu Chen, Nitesh V Chawla, and Xiangliang Zhang. 2024. Justice or prejudice? quantifying biases in llm-as-a-judge. *Preprint*, arXiv:2410.02736.

A Score Results in Second Stage

Fig. 3 shows the rated scores for all patents in each category. The scores of almost all ideas ranged between 4.3 and 4.8, although we requested the LLM to evaluate the ideas on a scale of 1-5. If this score is precise, the improvements were generally modest.

Table 5 shows the average and standard deviation of scores in all domains. In all cases, the scores for the refined ideas are better than those for the baseline ideas. In general, the scores evaluated by GPT-4o are consistently lower than those of GPT-4.1, and the differences between the baseline and refined versions are also smaller under GPT-4o. Scores are similar in the NLP and CS domains. In contrast, the matchem domain shows distinctly lower scores, possibly due to its higher level of domain expertise required. However, refined ideas in the matchem domain still received relatively high scores under GPT-4o, indicating the effectiveness of refinement even in specialized fields.

B Examples of Generated Ideas

Figures 4, 5, and 6 show examples of idea refinement with the greatest score improvements in matchem, nlp, and cs, respectively. The scores of these ideas, evaluated by GPT-4.1, improved by 0.4 to 0.5 points through the refinement process. Across the three themes analyzed, it is evident that the Refined Ideas (C) consistently build upon the Provisional Ideas (A), enhancing their practicality and scalability. Although the ideas generated by Llama-3.3 (B) do not appear to directly influence the refined versions, they do play a meaningful role in stimulating broader thinking and offering alternative perspectives for extending or generalizing the initial concepts.

One clear pattern across all Refined Ideas is the use of the suffix “Pro” in their product names. This consistency is unlikely to be coincidental. Rather, it reflects GPT-4.1’s implicit preference for positioning the refined outputs as high-performance, commercially viable versions of the original ideas.

Idea	matchem	nlp	cs
	Eval. by GPT-4.1		
Baseline	4.48 ± 0.16	4.58 ± 0.10	4.57 ± 0.12
Refined	4.67 ± 0.11	4.73 ± 0.08	4.74 ± 0.08
	Eval. by GPT-4o		
Baseline	4.39 ± 0.14	4.43 ± 0.15	4.42 ± 0.14
Refined	4.55 ± 0.13	4.52 ± 0.10	4.57 ± 0.12

Table 5: The average and standard deviation of scores in terms of domain and evaluators.

Examining the content of the Refined Ideas reveals a distinctive structural shift. While the Provisional Ideas tend to focus on the technology and its immediate use case, the Refined versions expand on this by explicitly addressing who the users are, in what contexts the products are deployed, and what real-world problems they solve. For example, in the NLP case, although the core classification and explanation functions remain the same, the refined version targets the enterprise segment and incorporates terms such as audit readiness, compliance, and enterprise-scale deployment, defining the tool within the context of organizational trust and accountability. Similarly, in the case of matchem, the refined idea incorporates terms such as Industry 4.0 to align the product with broader industrial trends and visions.

In contrast, the B ideas tend to be more abstract and less grounded in specific commercial or operational use cases. For example, the target of the NLP B idea, “scientists and researchers”, is a narrower and less commercially attractive market. In the cs example, Idea B refers broadly to “electronic devices” without specifying which industries or applications would benefit the most. Although this vagueness may be a limitation in terms of market clarity, it also provides conceptual flexibility, allowing new use cases and variations to emerge. Notably, elements such as the template-based NLG from the NLP B idea or the variable-depth compressive stress layer from the glass cover B idea may not appear explicitly in the Refined Ideas, but likely serve as conceptual input that enriches the refinement process.

Comparing the three case studies highlights how the Refined Ideas evolve the Provisional ones. In the cs example, A is focused narrowly on wearables, while C extends the scope to “critical medical sensors”. C addresses failure modes such as breakage, leakage, and warping with a material engineering solution optimized for biomedical environments. The result is a concept that is both technically robust and clearly aligned with the unmet needs in the target domain.

Taken together, these findings suggest that Refined Ideas are polished versions of the Provisional Ideas and combine technical validity with deployment readiness and market relevance. The B ideas, while not directly mirrored in the refined ideas,

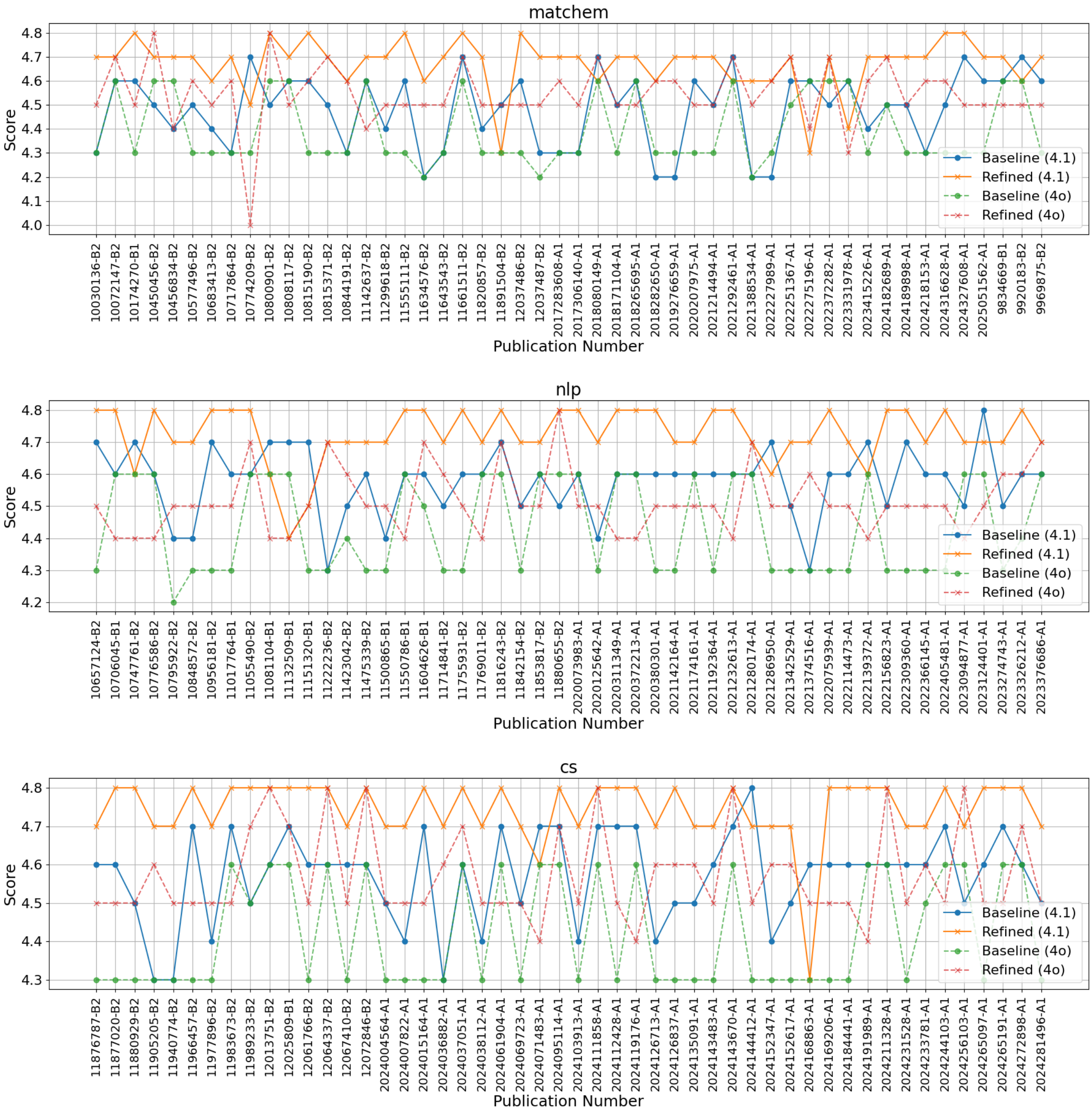


Figure 3: Scores of second stage for each category.

contribute by offering abstract concepts and broadening the thinking space.

C Prompts Used for Generation and Evaluation

The prompts for generation are shown in Figures 7–10, and those for evaluation in Figures 11 and 12.

(A) Provisional idea

Product Title: SnapGear: High-Performance Modular Gears for Collaborative Robotics

Product Description: Precision-molded polyacetal gears with superior impact resistance, thermal stability, and easy moldability for collaborative robot joints in factories. Targeted at robotics OEMs needing reliable, long-life, low-maintenance drive components.

Implementation: Utilize the patented polyacetal resin composition to injection mold gears with complex tooth geometries and integrated mounting features, enabling direct use in robot joints that require high impact strength, heat resistance, and smooth operation.

Differentiation: Unlike standard gears, SnapGear delivers higher durability, reduced yellowing, and superior mold release, enabling cost-effective mass production of complex, lightweight gears optimized for demanding collaborative robotics environments.

(B) Idea generated by Llama-3.3

Product Title: SmartGear

Product Description: A line of high-performance, impact-resistant gears for industrial machinery and automotive applications, utilizing the patented polyacetal resin composition for enhanced thermal stability and releasability.

Implementation: The patented polyacetal resin composition will be used to manufacture the gears, providing improved toughness and resistance to deformation. The unique blend of antioxidants, nitrogen-containing compounds, and fatty acid amides will ensure optimal performance in high-temperature environments.

Differentiation: SmartGear stands out from existing solutions due to its unparalleled combination of impact resistance, thermal stability, and moldability, making it an ideal choice for demanding industrial and automotive applications where reliability and efficiency are crucial.

(C) Refined idea

Product Title: SnapGear Pro: Self-Lubricating Precision Gears for Collaborative Robots & Smart Automation

Product Description: Injection-molded polyacetal gears with built-in lubrication and high impact/thermal resistance, designed for next-gen collaborative robots and automated machinery. Delivers quieter, longer-lasting, maintenance-light performance for OEMs and factories.

Implementation: Leverage the patented polyacetal resin composition, including fatty acid amide for internal lubrication, to mold gears with complex features. Optimize process for superior release, impact strength, and low friction—ideal for robot joints and conveyor drives.

Differentiation: Unlike standard or even advanced gears, SnapGear Pro combines intrinsic lubrication, high durability, and stable performance under heat, reducing downtime and maintenance—key for robotics and Industry 4.0 automation where reliability is paramount.

Figure 4: Idea refinement for 2019276659-A1 in matchem.

(A) Provisional idea

Product Title: ExplainAI: Transparent Document Classifier for Regulated Industries
Product Description: ExplainAI is an NLP-powered text classification tool for legal, healthcare, and finance professionals, providing automated document categorization with clear, natural language explanations and user feedback loops for improved trust and compliance.
Implementation: By integrating the patented micro/classifier system, ExplainAI assigns document labels and generates human-readable explanations for each decision. User feedback helps refine explanations and prunes ineffective micro classifiers, enhancing accuracy over time.
Differentiation: Unlike black-box NLP tools, ExplainAI uniquely delivers transparent, user-validated rationale for every classification, supporting regulatory compliance, audit trails, and user trust in high-stakes industries where explainability is critical.

(B) Idea generated by Llama-3.3

Product Title: Explainify
Product Description: Explainify is an AI-powered tool that provides natural language explanations for statistical classifier predictions, helping data scientists and researchers understand and trust their models' decisions.
Implementation: Explainify will utilize the patented technology to train a statistical classifier and a set of micro classifiers on labeled and unlabeled data, generating explanations for predictions using a template-based natural language generation technique.
Differentiation: Explainify stands out from existing solutions by offering transparent and interpretable AI decisions, enabling users to identify biases and errors, and improving model performance over time through user feedback and continuous learning.

(C) Refined idea

Product Title: ExplainAI Pro: Trustworthy Document Intelligence for Compliance-Driven Enterprises
Product Description: ExplainAI Pro empowers legal, healthcare, and financial teams to auto-categorize documents with AI-generated, user-validated natural language explanations, ensuring regulatory compliance, audit readiness, and user trust in high-stakes workflows.
Implementation: Utilizing the patent's dual-classifier and micro-classifier approach, ExplainAI Pro delivers document labels alongside clear, tailored explanations. Integrated user feedback refines explanations and model accuracy, supporting large-scale enterprise deployments.
Differentiation: Unlike generic NLP tools, ExplainAI Pro uniquely combines explainable AI, user-driven feedback, and audit-ready trails, giving compliance-critical industries transparent, continuously improving classification for both regulatory and operational advantage.

Figure 5: Idea refinement for 2021374561-A1 in nlp.

(A) Provisional idea

Product Title: SealSure: Ultra-Flat, Chemically Strengthened Glass Covers for Medical Wearables

Product Description: SealSure offers custom-shaped, ultra-flat glass covers for medical wearables (e.g., glucose monitors, ECG patches) that ensure hermetic sealing, high impact resistance, and reliable sensor function for patients and clinicians.

Implementation: We apply the dual ion-exchange and local material removal process to manufacture glass covers with precisely flat mounting surfaces, tailored for medical wearables that require robust sealing and durability over extended wear.

Differentiation: Unlike generic covers, SealSure guarantees micron-level flatness for leak-proof seals crucial in medical devices, combining advanced impact resistance with biocompatibility and custom geometries, solving both breakage and sealing failures.

(B) Idea generated by Llama-3.3

Product Title: SmartCover

Product Description: A chemically strengthened glass cover for electronic devices, providing enhanced durability and resistance to warping, with a unique mounting surface design for seamless integration.

Implementation: Utilizing the patented ion-exchange technology to create a compressive stress layer, ensuring the cover's strength and flatness, while incorporating a specialized polishing process to maintain the mounting surface's planarity.

Differentiation: SmartCover stands out with its innovative material removal process, enabling the creation of a variable-depth compressive stress layer, resulting in unparalleled strength and durability, setting it apart from conventional glass covers.

(C) Refined idea

Product Title: SealSure Pro: Precision-Engineered Glass Covers for Critical Medical Sensors

Product Description: SealSure Pro delivers ultra-flat, chemically strengthened glass covers tailored for advanced medical sensors (e.g., continuous glucose monitors, cardiac patches), ensuring leak-proof seals, durability, and reliable sensor readings for healthcare providers and patients.

Implementation: We leverage the dual ion-exchange with region-specific material removal to achieve micron-level flatness and custom compressive stress profiles, optimizing sealing surfaces and mechanical strength for each device geometry and medical application.

Differentiation: SealSure Pro uniquely combines patent-driven variable-depth compressive stress layers with biocompatible designs to guarantee both impact resistance and hermetic, distortion-free sealing—solving failures that generic covers or coatings can't address in medical wearables.

Figure 6: Idea refinement for 11905205-B2 in cs.

```
I give you the description of a patent. Read it.

<patent>
description
</patent>

## Task

Generate one business idea for a product using this patent.

Output the idea in the following format:

{
"product_title": "...",
"product_description": "...",
"implementation": "...",
"differentiation": "..."
}

## Rules

- product_title: A concise name for your product (up to 100 characters).
- product_description:  A brief explanation of the product outlining its
essential features and functions, the target users, their needs, and the
benefits provided by the product (up to 300 characters).
- implementation:  An explanation describing how you will implement the
patent's technology into your product (up to 300 characters).
- differentiation:  An explanation highlighting what makes your product
unique and the reason why it stands out from existing solutions (up to 300
characters).
```

Figure 7: Baseline prompt for idea generation.

```
I will give you structured information about a patent. Please read it
carefully and use it to generate one product idea.

## Patent
<patent_title>
{{ title }}
</patent_title>


<abstract> {{ abstract }}
</abstract>

<claims>
{{ claims }}
</claims>

<description>
{{ description }}
</description>

## Task
Use the patent information to propose a **new product idea** that applies
the patented technology.

- Use the **title and abstract** to understand the general scope of
the invention.
- Use the **claims** to understand what makes the invention unique or legally
protected.
- Use the **description** to understand possible implementations and
technical details.

## Output Format
Please output your idea in the following JSON format:

“‘json
{ ”product_title”: ”...”,
”product_description”: ”...”,
”implementation”: ”...”,
”differentiation”: ”...”
} “‘

## Rules
- product_title: A concise name for your product (up to 100 characters).
- product_description: A brief explanation of the product outlining its
essential features and functions, the target users, their needs, and the
benefits provided by the product (up to 300 characters).
- implementation: An explanation describing how you will implement the
patent’s technology into your product (up to 300 characters).
- differentiation: An explanation highlighting what makes your product
unique and the reason why it stands out from existing solutions (up to 300
characters).
```

Figure 8: Prompt with full patent text.

```
I give you the description of a patent. Read it.

<patent>
{{ description }} </patent>

## Task Generate one business idea for a product using this patent.
Your idea should not only be relevant to the technology described in the
patent, but also designed with the following evaluation criteria in mind.

### Evaluation Criteria
1. **Technical validity** — Is the patent suitable for the product? Is the
implementation feasible? Can it be done within three years?
2. **Innovativeness** — Does the patented technology offer a novel solution
to the demand?
3. **Specificity** — Is the idea specific? For example, “help researchers
manage references” is more specific than “help researchers do research.”
4. **Need validity** — Do the described users really need this solution?
5. **Market size** — Is the market large enough? Are there many potential
users?
6. **Competitive advantage** — What business advantage does the product
gain by using this patented technology?

## Output format
Present your idea in the following format:

“‘json
{
"product_title": "...",
"product_description": "...",
"implementation": "...",
"differentiation": "..."
}
“‘

### Rules
- product_title: A concise name for your product (up to 100 characters).
- product_description: A brief explanation of the product outlining its
essential features and functions, the target users, their needs, and the
benefits provided by the product (up to 300 characters).
- implementation: An explanation describing how you will implement the
patent’s technology into your product (up to 300 characters).
- differentiation: An explanation highlighting what makes your product
unique and the reason why it stands out from existing solutions (up to 300
characters).
```

Figure 9: Prompt with evaluation criteria.

```
I give you the description of a patent. Read it carefully.

<patent>
{{ description }}
</patent>

You have already proposed the following business idea based on this
patent:

<idea1>
{{ idea1 }}
</idea1>

Another person, based on the same patent, proposed this alternative
idea:

<idea2>
{{ idea2 }}
</idea2>

## Task
Your task is to **refine your own idea (idea1)** using the same patent and
evaluation criteria.
If the alternative idea (idea2) contains good elements that improve upon
your original idea — such as greater specificity, a stronger competitive
advantage, or better technical feasibility — you are encouraged to
incorporate them.
However, do not simply copy idea2.  Instead, **use it as inspiration to
enhance your own idea**, while maintaining originality and grounding your
solution in the patent.

### Evaluation Criteria

...
```

Figure 10: Prompt for idea refinement.

```
## Inputs

Read (1) a patent and (2) two product business ideas using the technology
in the patent.

<patent>
{{ patent.description }}
</patent>

<idea id="1">
{{ idea1 }}
</idea>

<idea id="2">
{{ idea2 }}
</idea>

## Task

Your task is to evaluate **both ideas across multiple criteria** and
determine which one is better overall.

Please carefully consider the following evaluation criteria:

...


## Judgment Instructions

After reviewing all criteria, select **one overall judgment** based
on the idea that performs better **across the board**.

Use the following judgment codes:

- ‘1‘: Idea 1 is better
- ‘2‘: Idea 2 is better
- ‘3‘: Tie (both are equally strong)
- ‘4‘: Neither is good (both ideas are weak)

## Output Format

Output your judgment in the following strict JSON format:

{
"judgement": <1 or 2 or 3 or 4>,
"reason": "<reason explaining why you selected this judgment, ideally
referencing multiple criteria>"
}
```

Figure 11: Prompt for evaluation (win/loss/tie).

```
## Inputs

Read (1) a patent and (2) two product business ideas using the technology
in the patent.

<patent>
{{ patent.description }}
</patent>

<idea id="1">
{{ idea1 }}
</idea>

<idea id="2">
{{ idea2 }}
</idea>

## Task

Your task is to evaluate **both ideas across multiple criteria** and
assign a total score to each idea using a 5.0-point scale (in increments of
0.1).

Please carefully consider the following evaluation criteria:

...


## Judgment Instructions

Assign a score between 0.0 and 5.0 (inclusive) to each idea, reflecting its
overall quality across all six criteria.

- Use increments of 0.1 only (e.g., 4.5, 3.2, 0.7).
- Base your judgment on how well the idea satisfies the criteria as a whole.
- Then explain your reasoning for both scores, referencing specific criteria.

## Output Format

Output your judgment in the following strict JSON format:

{
"score_idea1": <float between 0.0 and 5.0>,
"score_idea2": <float between 0.0 and 5.0>,
"reason": "<reason explaining how each score was derived, referencing
multiple criteria>"
}
```

Figure 12: Prompt for evaluation (score).

A Business Idea Generation Framework Based on Creative Multi-Agent Discussions

Mizuki HOSHINO[1]**, Fuminori NAGASAWA**[1]**, Shun SHIRAMATSU**[1]
[1]Nagoya Institute of Technology

Abstract

Recent advances in large language model (LLM) have enabled various applications in idea generation. However, generating business ideas from patent information remains underexplored. We participated in the PBIG 2025 shared task, which required generating business ideas from patents in three fields.

In this study, we propose a multi-agent framework in which five types of agents cooperate in stages to support the generation of business ideas that include diverse perspectives. Each phase involves one or more agents with different roles. This framework begins with a discussion between agents who are given personas, leading to the generation, selection by ranking, and refinement of ideas. Compared to conventional method, we show that the proposed method can promote the creation of more diverse and in-depth ideas.

In comparison with the output of other teams by the organizers, our system performed well in terms of specificity and Innovativeness. Also compared with the baseline, idea refinement phase is effective to improve quality of idea. However, generating ideas solely with a single agent may restrict the diversity of idea.

1 Introduction

In recent years, the accuracy of the large language model (LLM) has improved dramatically, and research on applying LLM to idea generation has been actively conducted. In general, it has been shown that the quality of idea generation for a theme can be improved by acquiring and utilizing domain information through RAG (Retrieval-Augmented Generation) . However, there has not yet been sufficient research on a framework for using patent information to think of business ideas.

In this paper, we report the results of participating in the Shared Task: Product business idea generation from patents (PBIG) and working on the task of generating business ideas from patent information. Participants will be given 150 USPTO patents extracted from three fields: NLP, Computer_Science, and Material_Chemistry, and will develop a system that outputs JSON for each patent with four items (Product Title, Product Description, Implementation, and Differentiation) that make up a "product business idea that can be realized within three years." We participated in the Shared Task as Team Shiramatsulab.

We applied a multi-agent system as an approach to this shared task. In a multi-agent system, multiple agents interact with each other to solve problems. Multi-agent system is used as a group discussion to generate business ideas from patent information. We developed the system based on the hypothesis that better ideas can be generated by multiple agents who express their opinions from their own perspectives and broaden the range of ideas.

2 Related Works

2.1 LLM-based Multiagent System for Ideation

Su et al. (2025) introduce *Virtual Scientists (VirSci)*, an LLM-driven multi-agent framework that forms a virtual research team to *generate, evaluate, and iteratively refine* scientific ideas. Their five-stage workflow—collaborator selection, topic discussion, idea generation, novelty assessment, and abstract drafting—outperforms single-agent baselines in both novelty and impact metrics.

Nomura et al. (2024) implement a brainstorming support system in which multiple LLM-based agents each assume an ISSUE–IDEA–PROS–CONS (IBIS) role. By mimicking human group dynamics while a single user interacts with the agents, the system boosts the quantity and diversity of ideas without the production-blocking effects often seen in conventional group brainstorming.

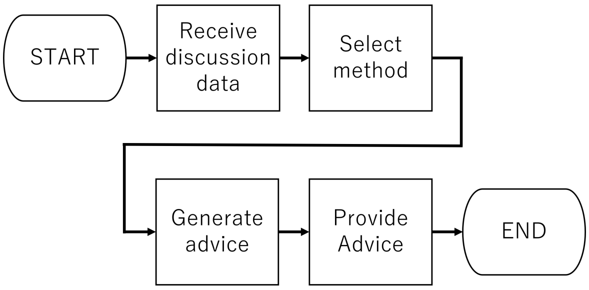


Figure 1: The process flow of ICS Agent

While the above systems each rely on a single ideation paradigm, our research integrates multiple, well-established idea-generation technique within a unified multiagent discussion environment. By orchestrating agents that embody these complementary heuristics and mediating their dialogue, we aim to provide broader creative coverage and finer-grained support for idea-centric debates.

2.2 Idea Creation Support Agent: ICS Agent

We developed an Idea Creation Support Agent (ICS Agent) that participates in group discussions to generate ideas, with the goal of enabling participants to consider ideas from a various perspectives (Hoshino et al., 2025). ICS Agent provides advice based on the state of the discussion. The advice is generated using an idea creation support method. One of three idea generation support methods (synectics method, search lighting method, and checklist method) is selected based on the state of the discussion. We included this idea generation support agent in a multi-agent discussion to enable discussions from more diverse perspectives and knowledge. The process flow of ICS Agent is shown in Figure 1.

3 System Overview

The overview of our system is shown in Figure 2. This system is divided into five phases: a persona generation phase, a multi-agent discussion phase, an idea generation phase, an idea evaluation phase, and idea refinement phase. The system uses gpt-4.1-mini.

3.1 Persona Generation Phase

In this phase, patent information is read and customer personas for discussion are created. Personas are created using LLM, and the following elements are determined (Table 1). In our system, three personas are generated and they join discussion as Discussion Participating Agents (DP Agents). The Personas are used only DP Agents.

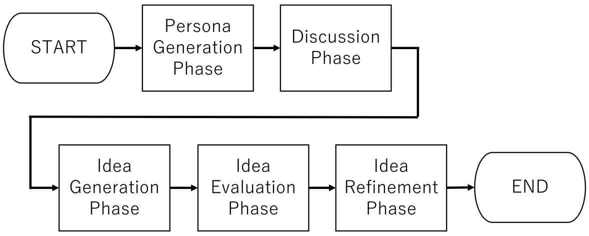


Figure 2: The process flow of our system

Table 1: Persona's attributes

Element	description
id	Identification number
name	name in discussion
age	Persona's age
gender	Persona's gender
occupation	Persona's occupation
lifestyle	Persona's lifestyle
value	Important points
needs	Persona's needs
pain_points	Issues persona thinks
purchasing_behavior	Buying patterns

3.2 Discussion Phase

In this phase, a discussion is held using ICS Agent and DP Agents that are provided personas determined in the Persona Generation Phase. DP Agents think of business ideas from patent information and express their idea. The discussion proceed for a total of five turns, with each turn defined by all agents express idea. Between each turn, ICS Agent provides advice to DP Agents to encourage ideas from the discussion history. In turns 2–5, each Discussion Participating Agents thinks their ideas considering the discussion history and advice of the ICS Agent.

3.3 Idea Generation Phase

In this phase, Generation Agent generates multiple business ideas based on the discussion history. In our system, three ideas are generated from discussion history.

3.4 Idea Evaluation Phase

In this phase, Evaluation Agent evaluate the multiple ideas generated in Idea Generation Phase and select best idea. The evaluation was performed using LLM-as-a-Judge on six criteria, the same as the evaluation criteria for the shared task (technical validity, innovativeness, specificity, need validity, market size, competitive advantage). Each idea is compared pairwise against the others and the final ranking is determined using Elo-based scoring.

3.5 Idea Refinement Phase

In this phase, DP Agent and Refinement Agent improves the idea selected in the Idea Evaluation Phase to make them more viable. DP Agents give their opinions on the selected idea from the persona's perspective. DP Agents review idea from two perspectives: technical and business issues. Refinement Agent then uses these opinions to further update the idea. At the end of this phase, the final idea is output.

4 Evaluation / Results

We evaluated our system output and compared it with the baseline method provided in the Shared Task.

First, we compare the final output of our system against two baselines: (1) the output of baseline method provided by the organizers, and (2) the output before the Idea Refinement Phase.

Second, the final output submitted by our system were evaluated by the task organizers via pairwise comparisons against outputs of other participating teams, using Elo scores derived from LLM-as-a-judge.

4.1 Comparison of System Output with Baseline

To assess the effectiveness of our system, we evaluated the ideas generated by the following three methods: (1) Baseline: the baseline method uses baseline prompt provided by the task organizers, (2) Intermediate: our system's intermediate output generated before Idea Refinement Phase, and (3) Final: our system's final output submitted to the shared task.

We selected randomly 10 patents selected from each of the three fields and generated ideas for each patents. And compared with ideas generated by same patents each other. Ideas were evaluated using pairwise evaluation by LLM. To prevent bias in the LLM output, we conduct the evaluation of the ideas twice, with the order of the ideas swapped. In other words, the total number of comparisons is 20. The LLM model used was gpt-4.1-mini.

Tables 2 - 4 show the number of ideas that were judged to be superior when the ideas generated by each method were evaluated in pairs.

Table 2: Comparison result Baseline with Final

	Baseline	Final
cs	1	19
mc	12	8
nlp	2	18
total	15	45

Table 3: Comparison result Intermediate with Final

	Intermediate	Final
cs	15	5
mc	15	5
nlp	14	6
total	44	16

As shown in Table 2 and Table 4, Final output of our system may be more effective than Baseline method and Intermediate output. However, limited to the material_chemistry field, the output of the baseline method output is superior than Final output. Also, Table 4 shows that Intermediate output of our system was not more effective than Baseline method output. Therefore, it was suggested that the Idea Refinement Phase may contribute to improving the quality of ideas.

4.2 System Leader Board Results

The leaderboard results for the Share Task is shown in Table 5. These score were evaluated by the organizer. Table 5 displays the highest scores, our team's scores, and the average scores for only five teams that submitted ideas in three fields. Each team's evaluation score is calculated by averaging the scores in the three criteria for each patent field. In this comparison, we did not include teams that submitted ideas in only one or two fields for calculation. These results are based on evaluations by only LLM-as-a-Judge.

As shown in Table 5, generated ideas of our system is superior in Specificity and Innovativeness. On the other hand, evaluation scores in Market size and Technical validity are lower evaluation than

Table 4: Comparison result Baseline with Intermediate

	Baseline	Intermediate
cs	8	12
mc	13	7
nlp	7	13
total	28	32

Table 5: Evaluation score using LLM for each criteria

	Top	Our	Average
tech_valid	1110.7	993.3	1012.3
spec	1189.7	1035.3	1008.0
need_valid	1084.7	1011.3	1008.3
market_size	1060.7	985.7	1014.2
innov	1086.7	1027.7	1000.6
comp_adv	1140.0	1004.7	1001.7

the average.

5 Discussion

5.1 Idea generation ability against baseline

As shown in Table 2 - Table 4, while Idea Refinement Phase is effective to make quality of idea higher, intermediate idea is as same quality as baseline idea. This result likely reflects the design of Idea Generation Phase. In this phase, the ideas are generated by agent who is not participating in the discussion. As mentioned in Section 3, Idea Generation Agent does not have a specific persona. In addition, ideas are generated by a single agent. Because the Idea Generation Agent is not provided a persona, three similar ideas are generated, which may have prevented the idea from reflecting discussion as collective knowledge. Therefore, it is considered to have reduced diversity of the generated ideas. Similarly, a single agent that no persona has is used in the Idea Refine Phase. However, Refinement Agent refines idea by considering feedback from the DP Agents. It might be that by gaining insights into idea challenges and improvement suggestions from the diverse perspectives of DP Agents, Refinement Agent was able to improve quality of idea.

5.2 Evaluation in Shared Task Leaderboard

As mentioned in Section 4.2, ideas generated by our system is high scores in Specificity and Innovativeness. One contributing factor for Specificity may be DP Agents. Each DP Agents express their thoughts on business idea and discuss them. Through this discussion, idea details may become more specific.

In contrast, the scores in Market size and Technical validity are low. It may be considered that Market size evaluation is related to Specificity. As the discussions become more specific, the target market may become more limited and the market size may become smaller. Also regarding Technical validity, because our system did not conduct technical analysis, it is possible that ideas difficult to realize within three years remained.

6 Conclusion

We proposed a multi-agent framework using LLM for generating business ideas from patents as part of the PBIG: Shared Task. Our system outputs were high score in Specificity and Innovativeness. In addition, compared with baseline method, it may have been suggested that our system output is superior than baseline. We found that views from agents with different personas are effective for quality of idea. However, it may be that generating idea by a single agent caused loss of diversity of idea. For future work, we plan to : (1) verify the effects of generating idea by each multi agents to its quality. (2) investigate the effects of the discussion structure and dialogue format between agents on the quality and diversity of ideas generated.

Acknowledgments

This study was supported by JST CREST (JPMJCR20D1).

References

Mizuki Hoshino, Sora Matsumoto, and Shun Shiramatsu. 2025. Development and evaluation of llm-based ideation support agent for idea generation discussions. In *Proceedings of the 39th Annual Conference of the Japanese Society for Artificial Intelligence (JSAI 2025)*, Osaka, Japan. Japanese Society for Artificial Intelligence. Paper ID: 3J4-GS-5-02 (in Japanese).

Moeka Nomura, Takayuki Ito, and Shiyao Ding. 2024. Towards collaborative brainstorming among humans and AI agents: An implementation of the IBIS-based brainstorming support system with multiple AI agents. In *Proceedings of the ACM Collective Intelligence Conference (CI 2024)*, Boston, MA, USA.

Haoyang Su, Renqi Chen, Shixiang Tang, Zhenfei Yin, Xinzhe Zheng, Jinzhe Li, Biqing Qi, Qi Wu, Hui Li, Wanli Ouyang, Philip Torr, Bowen Zhou, and Nanqing Dong. 2025. Many heads are better than one: Improved scientific idea generation by a LLM-based multi-agent system. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL 2025)*.

Team MCG DSN at the AgentScen Shared Task: Knowledge Integration and Self-Improvement via LLMs for Generating Business Ideas from Patent Documents

Masaya Shimanuki, Naoto Shimizu, Kentaro Kinugasa, Hiroki Sugisawa[*]

Mitsubishi Chemical Corporation

Abstract

Patents represent valuable sources of commercial potential; however, generating viable business idea from such information requires advanced expertise. This paper proposes a prompt-based framework that integrates patent element, inventor profile, market potential, and business model grounded in TRIZ theory to generate high-quality business ideas. Furthermore, we introduce a self-improvement mechanism that extracts AI judges' personas from results of the PBIG competition-based evaluation and incorporates these insights to refine subsequent generations of ideas. While our output demonstrated strong performance under AI-based evaluation, notable discrepancies with human judgment were observed, highlighting the need for further alignment with human evaluation.

1 Introduction

Generating business ideas from patent documents poses a critical challenge in contemporary industrial innovation. While patents serve as rich and accurate repositories of technical knowledge, transforming this information into concrete product concepts or viable business models that achieve market acceptance demands advanced domain expertise and multifaceted evaluation [1]. Although recent studies have introduced automated generation methods using large language models (LLMs) [1,2], a unified framework that concurrently integrates technical feasibility and commercial viability remains lacking.

In this study, we propose a prompt-based model that emulates the domain knowledge of business expertise to generate higher-quality ideas [3]. Our approach consists of a five-module prompt flow: (1) extraction of core technical elements via patent component analysis, (2) assessment of the inventor's strengths, (3) ranking of market potential and applicability, (4) construction of business models guided by TRIZ principles, and (5) iterative refinement through AI-driven pairwise evaluations, during which judge personas are extracted and additional constraints are applied for self-improvement.

2 Methodology

2.1 PBIG Task

The Patent Business Idea Generation (PBIG) [4] shared task is a competition aimed at generating business ideas from patents using generative AI, which are feasible for market launch within three years. A dataset of 150 USPTO patents (50 each from the domains of natural language processing, computer science, and materials chemistry) is provided, including JSONL metadata (title, application number, publication number, publication date, abstract, claims, and description), PDF documents, and figure images. Participating systems must output four JSONL formatted fields with strict length limits: Product Title (≤ 100 characters), Product Description, Implementation, and Differentiation (each ≤ 300 characters) and may leverage external resources to enhance idea

[*] Email: hiroki.sugisawa.ma@mcgc.com
The name, MCG DSN indicates the Data Scientist Network (DSN) in Mitsubishi Chemical Group (MCG).

diversity and practical relevance. Evaluation combines human expert judgments with three AI based automated judges in a pairwise comparison of ideas originating from the same patent, and final rankings are computed using the Elo rating method.

2.2 Moduled Prompt Flow

We propose a domain knowledge-based prompt flow to guide generative AI in producing higher-quality business ideas from a patent [3]. This flow consists of five sequential Modules; all prompt templates are provided in the Appendix A.

(1) **Patent Element Analysis**: Prompt **1** provide an analytical framework to deconstruct the patent and organize its technical contents. From a JSONL - formatted patent record, we extracted (i) the inventor's name, (ii) the intrinsic value of the patent, (iii) its applicability domains, and (iv) a concise summary. These outputs were then forwarded to Modules 2–5.

(2) **Inventor Profiling**: Prompt **2** employs a structured inference template to identify the inventor's (company, organization, or individual) core strengths, based on web search results. The investigation was conducted by the Gemini API to retrieve and synthesize publicly available information.

(3) **Market and Application Analysis**: Using Prompt **3**, we enumerated potential markets and applications for the patented technology and rank them according to projected profitability and societal impact. The top candidates were presented in tabular form, and the highest-priority market / application was selected for further modules.

(4) **Business Model Construction**: Leveraging TRIZ theory via Prompt **4**, we constructed candidate business models that maximize revenue by capitalizing on the patent's identified strengths (from Module 1) within the selected market / application (from Module 3).

(5) **Business Idea Proposal**: Finally, integrating outputs from Modules 1–4 via Prompt **5**, we generated a concrete product and business model proposal. Each proposal was evaluated against six criteria: technical validity, innovativeness, specificity, need validity, market size, and competitive advantage.

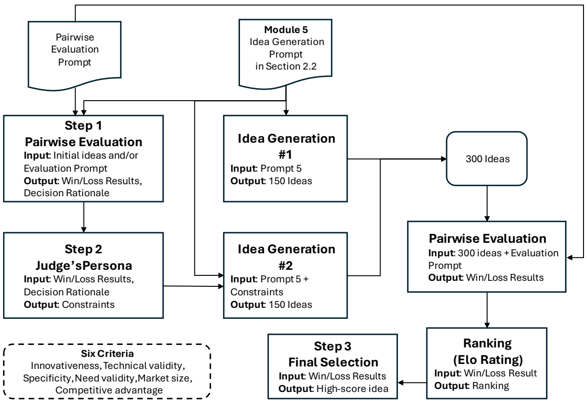
2.3 Self-Improvement via AI Judge

To further refine the business ideas generated in Section 2.2, we implemented the following self-improvement steps within an AI judge (Scheme 1):

Step (1) Initial Pairwise Evaluation (Prompt 6): We employed the o3-mini model to perform pairwise comparisons of business ideas, using six criteria. To reduce computational costs, comparison pairs were selected via random sampling. For each comparison, the judge outputs a justification, including the winning idea's identifier and a brief rationale. All ideas were subsequently ranked using Elo ratings [5,6].

Step (2) Judges' Persona Extraction: In Prompt **7**, we extracted "judge personas" from the justifications generated in Step 1, capturing judge's thinking patterns and prioritization criteria. Building upon these personas, Prompt **8** generated generalized supplementary constraints reflecting the judges' perspectives. These constraints were incorporated into Prompt **5** to guide the model in producing outputs that avoid impractical ideas and better fulfill all evaluation criteria.

Step (3) Final Selection: A total of 300 business ideas were generated—150 each from Module 5 in Section 2.2 and from Steps 1–3 in this section. Elo scores were computed for all outputs, and the highest-scoring idea in each category was selected as the final submission. Outputs that violated predefined guidelines, such as character limits, were manually excluded from consideration.


Pairwise Evaluation Prompt
Module 5 Idea Generation Prompt in Section 2.2
Step 1 Pairwise Evaluation
Input: Initial ideas and/or Evaluation Prompt
Output: Win/Loss Results, Decision Rationale
Idea Generation #1
Input: Prompt 5
Output: 150 Ideas
300 Ideas
Step 2 Judge'sPersona
Input: Win/Loss Results, Decision Rationale
Output: Constraints
Idea Generation #2
Input: Prompt 5 + Constraints
Output: 150 Ideas
Pairwise Evaluation
Input: 300 ideas + Evaluation Prompt
Output: Win/Loss Results
Six Criteria
Innovativeness, Technical validity, Specificity, Need validity, Market size, Competitive advantage
Step 3 Final Selection
Input: Win/Loss Results
Output: High-score idea
Ranking (Elo Rating)
Input: Win/Loss Result
Output: Ranking

Scheme 1. Self-improvement via AI judge

3 Results & Discussion

3.1 Self-Improvement via AI Judge

In this study, we focused on the category of materials chemistry and conducted Step (1) using approximately 20 randomly selected patents. An example output of Step (1) is presented in Listing 1. Through pairwise comparisons of business ideas, Prompt **6** succeeded in generating pseudo-rationale to why one idea (Idea A) was considered superior to another (Idea B).

In Step (2), we applied Prompt **7** to the list of pseudo-rationales to extract AI judges' personas, which are described in detail in Appendix B. As a result, seven key evaluation criteria were identified as influential in the assessment of AI-generated business ideas: (1) technical rigor, (2) feasibility, (3) market applicability, (4) regulatory compliance, (5) specificity, (6) innovativeness, and (7) intellectual property defensibility. Specifically, technically oriented judges tended to favor proposals that adhered closely to patent content and included quantitative specifications (e.g., Young's modulus ≥ 2700 MPa or additive concentrations of 0.1‑2 wt%), while abstract or loosely related ideas were consistently disfavored. Judges emphasizing feasibility preferred ideas that were compatible with existing infrastructure and realistically implementable within a two- to three-year timeframe; they rated proposals requiring large-scale investment or long development cycles lower. Market- and regulation-oriented personas prioritized ideas targeting large-scale, regulation-heavy markets (e.g., automotive, healthcare, home appliances) and those addressing pressing regulatory frameworks such as Euro 7 and VDA 278. Furthermore, ideas with detailed technical specifications (such as chemical structures, process conditions, catalysts, and operational temperature ranges) were positively evaluated, as were inventions exhibiting originality and resistance to imitation. Conversely, vague or overly generic ideas were consistently rated unfavorably.

Idea A leverages the patent's chemistry to achieve ultra-low (<1 ppm) formaldehyde VOC release, a performance level not explicitly disclosed in the patent claims or typical commercial POM offerings. **Idea B**'s focus on mechanical strength and acid resistance closely matches multiple embodiments already taught in the patent, offering less novelty. Thus Idea A demonstrates the higher innovative leap.

Listing 1: Example of justification.

"**product_description**":"Drop-in molded quick-connects for gasoline & diesel rails that use patented low-formaldehyde POM (0.4–0.9 mol % oxy-alkylene; 0.1–2 wt % branched POM) plus 0.8 wt % MgO scavenger. Targets Tier-1 fuel-system makers battling Euro-7 aldehyde caps and permeation limits; delivers ≥2700 MPa modulus and <0.3 mg m^{2} formaldehyde, extending service life and cutting cabin odor (370 charactors)"

Listing 2: Example of product description.

Subsequently, in Prompt **8** we elicited the PBIG task's evaluation criteria from the pseudo-judge personas and incorporated the insights thus obtained as constraints into Prompt **5**. Although this approach succeeded in securing favorable evaluations from the AI reviewers on the PBIG task (Section 3.2), the imposed constraints sometimes proved counterproductive, leading to instances in which prescribed character limits were ignored or unrealistic numerical values were adopted as shown in Listing 2. This tendency was especially pronounced in the chemistry domain, where it is difficult to idealize the materials properties, thereby emphasizing the necessity of grounding the task's parameters in practical realities.

3.2 Elo Score

Table 1 summarizes the evaluation results for the PBIG task in the materials chemistry category, reporting three score types: the "auto-score" assigned by an AI judge, the "human-score" given by expert reviewers, and "our score," computed according to the six criteria defined in Section 2.3. Notably, our method achieved an auto-score of 1185 for the "Innovation (innov)" criterion (second highest among all PBIG participants), while human scores were lower, indicating a disparity between AI and human judgments. It should be noted, however, that our approach did not incorporate an analysis of human judges' personas. Consequently, applying our framework to analyze real-world customer feedback may offer a promising avenue for generating proposals that resonate more effectively with human evaluators. For a detailed discussion of the LLM-as-a-Judge concept, see References [7–12].

Table 1: Elo Scores in Materials Chemistry category.

criteria	auto-score	human-score	Our-score
tech_valid	896	928	1484
spec	1112	950	1464
need_valid	946	1026	1477
market_size	939	1006	1196
innov	1185	1009	1164
comp_adv	1002	974	1487

Conclusion

We suggested a prompt-based model for generating business ideas from patent documents and achieved the generation of high-quality ideas in the PBIG task [7]. Our approach systematically integrates processes such as extracting patent elements, profiling inventors, conducting market analysis, and constructing business models based on TRIZ principles, leveraging domain knowledge from business strategy experts. Furthermore, AI self-improvement mechanism enhanced idea quality according to predefined competition criteria. While demonstrating strong performance in AI-based evaluations, the study also highlighted challenges in aligning with human judges' preferences. Future work should focus on refining our approach to better accommodate human judgment, for example by incorporating feedback analysis from real-world business settings.

References

[1] A. Subramanian, K.P. Greenman, A. Gervaix, T. Yang, R. G.-Bombarelli, arXiv:2303.08272 (2023).

[2] O. Plätke, R.C. Geibel (2024) The use of artificial intelligence for idea generation in the innovation process (Springer Proceedings in Business and Economics) Cham, Switzerland: Springer. https://doi.org/10.1007/978-3-031-66517-2_14

[3] H. Sugisawa, K. Kinugasa, Patent application in preparation (2025).

[4] https://sites.google.com/view/agentscen/shared-task

[5] Elo. The Rating of Chessplayers, Past and Present. Ishi Press, (1986)

[6] Chatbot Arena: Elo Rating Calculation (July 17, 2023)

[7]https://sites.google.com/view/agentscen/shared-task/evaluation

[8] H. Yoshiyasu, "Team NS\NLP at the AgentScen Shared Task: Structured Ideation Using Divergent and Convergent Thinking," The 2nd Workshop on Agent AI For Scenario Planning (AGENTSCEN2025)}, in press (2025).

[9] G. Kanumolu, A. Urlana, V.K. Charaka, B.M. Garlapati, "Agent Ideate: A Framework for Product Idea Generation from Patents Using Agentic AI," The 2nd Workshop on Agent AI For Scenario Planning (AGENTSCEN2025), in press (2025)

[10] Y. Xu, T. Hirasawa, S. Kawano, S. Kato, and T. Kozuno, "MK2 at PBIG Competition: A Prompt Generation Solution," The 2nd Workshop on Agent AI For Scenario Planning (AGENTSCEN2025), in press (2025)

[11] Y. Terao and Y. Tachioka, "Collaborative Invention: Refining Patent-based Product Ideation via LLM-Guided Selection and Rewriting," The 2nd Workshop on Agent AI For Scenario Planning (AGENTSCEN2025), in press(2025).

[12] M. Hoshino, S. Shramatsu, and F. Nagasawa, "A Business Idea Generation Framework Based on Creative Multi-Agent Discussions", The 2nd Workshop on Agent AI For Scenario Planning (AGENTSCEN2025), in press (2025).

Appendix A. Prompts

Prompt 1: Patent Element Analysis

Task

- Propose a novel product and/or business model based on the patent.
- Organize and present the content of the patent clearly.

Structure

(1). Who developed it? Output: Name of the organization or company.

(2). What is the value of this patent? Output: A detailed explanation. Note: Describe the unique features and differentiators from existing inventions as thoroughly as possible.

(3). What are the potential applications? Output: A list in table format. Note: Include both market sectors and use cases in the columns. Propose as many as possible.

(4). Summary of the patent Output: A detailed explanation. Note: Describe it as clearly and accessibly as possible, highlighting the core-essence of the patent.

Prompt 2: Inventor Profiling

Request

Based on the company/organization name, infer and list its potential strengths using the structure below.

Company/organization

[Output(s) suggested by Prompt **1 (1)**]

Structure

1. **Strength**: **Description**
2. **Strength**: **Description**

Prompt 3: Market and Application Analysis

Request

We are exploring potential markets and applications for the given patent. Summarize your research following the output format below.

Patent Details

[Output(s) suggested by by Prompt **1 (2)**]

List of Potential Applications

[Output(s) suggested by Prompt **1 (3)**]

Output Format

* Present the information in a table only (do not include any additional text).
* The table should include the following columns:
** Market
** Market Growth Rate
** Application
** Estimated Profit from Application
** Social Significance
* Sort the entries in descending order of estimated profit.

Prompt 4: Business Model Construction

Request
Based on the strengths of the following patent, investigate which business model would likely generate the highest revenue when entering the specified industry.

Patent Strengths
[Output(s) suggested by Prompt **1 (2)**.]

Target market
[Max-profit output suggested by from Prompt **3**]

Output Format
* Present the results in a table.
* The columns should be:
* Business Model
* Estimated Revenue
* Combined TRIZ Principles Used
* Description of the Business Model
* Each business model should be proposed by applying two or three TRIZ problem-solving principles.

Prompt 5: Business Idea Proposal

Request
Using the information collected so far, propose a product and corresponding business model.

Output Requirements
- product_title: A concise name for your product (up to 100 characters).
- product_description: A brief explanation of the product outlining its essential features and functions, the target users, their needs, and the benefits provided by the product (up to 300 characters).
- implementation: An explanation describing how you will implement the patent's technology into your product (up to 300 characters).
- differentiation: An explanation highlighting what makes your product unique and the reason why it stands out from existing solutions (up to 300 characters).

Company Strengths
[Output(s) suggested by Prompt **3**]

Target Market
[Output(s) suggested by Prompt **2**]

Business Model Concept
[Output(s) suggested by Prompt **4**]

Patent Summary
[Output(s) suggested by Prompt **1 (4)**]

Patent Advantages
[Output(s) suggested by Prompt **1 (2)**]

Constraints
* The proposal must clearly address all of the following elements:
Technical validity: Is the patent suitable for the product? Is the implementation feasible? Can it be done within three years?
* Innovativeness: Does the patented technology offer a novel solution to the demand?
* Specificity: Is the idea specific? For example, "help researchers manage references" is more specific than "help researchers do research."
* Need validity: Do the described users really need this solution?
* Market size: Is the market large enough? Are there many potential users?
Competitive advantage: What business advantage does the product gain by using this patented technology?
* Only output the content (no additional text).
* Count the word total and strictly stay within the word limit.

Prompt 6: Initial Pairwise Evaluation

Task
- Your task is to choose the better idea from the perspective of **Technical validity**.
- Is the patent suitable for the product? Is the implementation feasible? Can it be done within three years?
- The idea must be capable of being made or used in some industry, which can include manufacturing, agriculture, or other practical applications. It should not be a purely theoretical concept.

Output format
Return a JSON object with exactly these keys:
- idea_id: either "A" or "B"
- reason: brief justification.

Remember: output ONLY the JSON object.

Prompt 7: Judges' personas extraction

Background
- You are participating in a competition to generate inventions.
- Below are the reasons for victory or defeat when pitting your inventions against a baseline method.
- To improve the prompt, we would like to infer the judges' characteristics from these win/loss explanations.

Request
- Please extract the judges' personas based on the win/loss explanations.

Evaluation Criteria
- Technical validity, innovativeness, concreteness, alignment with needs, market size, competitive advantage.
- This is a competition for generating inventions from patents.

Judging Reasons
["reason" list suggested by Prompt **6**]

Prompt 8: Factors that Judges' take into account

Background
- You are a generative AI participating in a competition to create inventions.
- The competition results - including score, title, summary, and points of differentiation—have been compiled.

- You wish to impose constraints on the AI prompt in order to improve these results.

Request
To achieve a score above 1200 points, generate a list of additional constraints to include in the prompt, using the judges' personas. These constraints should be universally applicable to any patent.

Evaluation Criteria
Technical validity, innovativeness, specificity, alignment with needs, market size, competitive advantage.
This is a competition to derive inventions from patents.

Desired Output
A bullet-point list of constraints.

Score Results
[Elo score]

Judges' Persona (extracted from reasons for wins and losses)
[Output(s) suggested by Prompt **7**]

Appendix B. Persona and Constraints

Judges' Persona (specialized in chemistry)

1. Technically Rigorous & Patent-Faithful
Key Traits: Engineering-first mindset, conservative with feasibility, and require direct use of patented materials, processes, or compositions.
What Wins: Inventions that specify detailed formulation ranges, processing routes, and quantifiable outputs (e.g., <0.5 mg/m^2 formaldehyde).
What Fails: Ideas that stray into abstract software, AI, or vaguely connected uses, regardless of market appeal.

2. Feasibility-Driven Realists
Key Traits: Expect commercialization within \~2–3 years using existing equipment and infrastructure.
What Wins: Drop-in solutions with low switching cost, validated manufacturing paths, and minimal retooling.
What Fails: High-risk or speculative technologies requiring new plants, new chemistry platforms, or novel infrastructure.

3. Regulation-Responsive Evaluators
Key Traits: Highly sensitive to global regulatory frameworks—especially emissions (e.g., Euro 7, VDA 278, GB/T 27630).
What Wins: Inventions that enable immediate compliance or preempt near-future regulatory mandates.
What Fails: Concepts not linked to urgent regulatory pain points—even if innovative or sustainable.

4. Market-Oriented Strategists
Key Traits: Prioritize large, urgent, and high-growth markets with tangible demand and broad applicability.
What Wins: Ideas addressing mass markets like automotive interiors or appliances with regulatory pressure and OEM interest.
What Fails: Niche, narrow, or speculative markets with limited customer base.

5. Specificity- and Detail-Seeking Engineers
Key Traits: Demand clear articulation of how the invention works, what it contains, and what it improves.
What Wins: Highly detailed ideas including exact ppm levels, mol % comonomers, phr ranges, and processing methods.
What Fails: Generic, broad-stroke proposals lacking technical, chemical, or process depth.

6. Innovation-Oriented but Execution-Focused
Key Traits: Favor originality only when technically validated and commercially actionable.
What Wins: Combinations that extend the patent's scope in a novel, IP-defensible, and manufacturable way (e.g., resin + acoustic damping).
What Fails: "Buzzword" novelty (e.g., AI, platforms) that lacks tangible linkage to the patent or physical deliverables.

7. Competitive Advantage and IP Defensibility Seekers
Key Traits: View IP as a moat; prefer solutions difficult to replicate or bypass.
What Wins: Inventions with unique formulations, protected performance features, or platform-level lock-in.
What Fails: Me-too products or ideas that offer little legal or performance protection from competition.

Constraints (specialized in chemistry)

Technical Rigor and Patent Fidelity
- Specify exact compositional ranges (e.g., 0.1–2 wt% additive, 0.4–0.9 mol% comonomer).
- Define quantifiable performance targets (e.g., Young's modulus ≥ 2700 MPa, formaldehyde < 0.5 mg/m^2).
- Directly integrate patented methods, materials, or formulations into the invention concept.

Feasibility and Manufacturing Readiness
- Require drop-in compatibility with existing infrastructure (no equipment swaps or new machinery).
- Limit implementation time to ≤ 3 years by leveraging current supply chains and industry-standard processes.

Regulatory Relevance
- Align the invention with imminent or forthcoming global regulations (Euro 7, GB/T 27630, VDA 278, etc.).
- Include language demonstrating proactive compliance, certification readiness, or the benefits of regulatory exemptions.

Market Scope and Applicability
- Target large, regulated markets (automotive, appliances, medical, etc.).
- Quantify the total addressable market (TAM) or cite specific OEM/Tier-1 applications.

Detail, Specificity, and Process Clarity
- Provide precise chemical structures, blend ratios, processing temperatures, catalysts, and quench agents.
- Define processing parameters (e.g., melt index, temperature range, MI = 0.5–1.5).

Innovative Yet Practical

- Restrict novelty to physically realizable combinations (e.g., a patented resin plus an acoustic layer), avoiding vague software abstractions.
- Ensure any AI or digital component directly controls or monitors a physical/material process.

IP Strength and Competitive Advantage
- Emphasize IP-backed differentiation (e.g., patented compositions, protected process steps).
- Include commercialization strategies such as OEM licensing models, territorial exclusivity, or supply contracts tied to compliance.