Journal of Computer Technology and Software

ISSN:2998-2383

Vol. 4, No. 5, 2025

High-Order Dependency Graph Learning for Container Anomaly Detection in Microservice Environments

Chun-Yao Hsieh[1*]
[1]Columbia University, New York, USA
*Corresponding author: Chun-Yao Hsieh; ch3884@columbia.edu

Abstract: This paper addresses the structural modeling challenge of anomaly detection in modern containerized microservice systems. It proposes a graph attention network-based method for modeling container dependencies and detecting anomalies. The method builds a high-order dependency graph that represents dynamic invocation relationships between containers using runtime logs and monitoring metrics. It initializes node embeddings and employs a multi-head graph attention mechanism to adaptively model the interaction strength between upstream and downstream components. A multi-task optimization framework is introduced, including structural reconstruction loss, attention entropy regularization, and a contrastive objective. This enhances the model's expressiveness in terms of structural consistency and anomaly discriminability. The model can accurately aggregate key behavioral paths between containers and effectively model and score weak anomaly patterns within complex dependencies. To evaluate the method, multiple sensitivity experiments were designed to analyze performance trends concerning attention heads, embedding dimensions, and regularization parameters. The results confirm the adaptability and stability of the graph attention mechanism in container-level structural modeling and anomaly detection. The proposed method outperforms existing approaches across several key metrics and achieves accurate and robust anomaly identification in highly dynamic and complex system scenarios.

Keywords: Container anomaly detection; graph attention mechanism; structure-aware modeling; multi-task optimization

1. Introduction

In modern large-scale distributed systems, container technology has become the mainstream paradigm for deploying and managing cloud-native applications, serving as the core support for microservice architectures. Containers offer advantages such as lightweight design, portability, and elastic scalability, which enable their widespread use in key domains like finance, e-commerce, transportation, and industrial IoT. However, as container clusters scale up, their runtime states become increasingly dynamic and complex. The dependency relationships among containers exhibit high concurrency, multi-layered hierarchies, and strong heterogeneity, which significantly reduce system observability and interpretability. Especially in environments characterized by frequent service interactions and dynamic resource scheduling, traditional anomaly detection methods based on static rules or isolated metrics fail to capture potential risks effectively. There is an urgent need for an intelligent detection mechanism capable of structural modeling and semantic awareness, enabling unified representation and efficient modeling of both micro-level behaviors and macro-level dependencies [1].

Anomalous behaviors at the container level rarely occur in isolation. They often intertwine with issues such as abnormal communication links between components, drift in call paths, and imbalanced resource competition. Therefore, accurate anomaly detection relies on effective modeling of container dependencies. During actual operations, the dependency structures of containers are not only temporal and context-dependent but also exhibit structural evolution and non-stationarity. Traditional flat or statistical modeling approaches cannot capture such dynamic and complex relationships. Moreover, anomalous behaviors are often triggered by subtle perturbations in the dependency paths. For instance, a container suddenly receiving a large volume of unexpected requests may not be misconfigured itself, but affected by failures in upstream components. Thus, constructing a model that captures upstream and downstream interactions and dynamically tracks behavioral evolution is key to improving detection accuracy and robustness [2].

Graph-based modeling methods have demonstrated strong expressive capabilities in representing complex systems. The interactions, dependencies, and scheduling processes among containers naturally form a multi-node, multi-edge directed graph, which provides a solid foundation for introducing graph neural networks. In particular, graph attention networks assign adaptive weights to nodes, allowing the model to learn non-uniform importance relationships without relying on global structural priors. Compared to traditional graph convolution methods, the attention mechanism enables finer differentiation between critical and redundant paths during local dependency modeling. This enhances structural perception flexibility and generalization. In container anomaly detection scenarios, such mechanisms are expected to dynamically aggregate key dependency information and effectively mitigate information dilution and overfitting. This allows precise modeling of local anomaly propagation patterns.

In highly dynamic environments, container behavior often exhibits strong non-linearity and sparsity. Anomalous signals can be easily masked by normal behaviors, increasing detection difficulty. Additionally, container-level anomalies often manifest as a combination of latency and bursts, which cannot be reliably detected using isolated logs or metrics. Therefore, it is essential to build a modeling framework centered on structure and integrating semantic and behavioral multimodal

features. Combining graph-based modeling with time series analysis can improve sensitivity to subtle perturbations in dependency chains and enhance detection capabilities across hierarchical and multi-granular anomalies [3]. In this context, graph attention networks show unique advantages. Their flexible attention mechanism and topology-preserving features support multi-scale modeling of complex container relationships, thereby improving detection stability, accuracy, and interpretability.

In summary, container-level dependency modeling and anomaly detection are core technical challenges for ensuring the stability and security of distributed systems. They are also directly related to the sustainable evolution and operational efficiency of microservice architectures. Introducing graph attention mechanisms to dynamically model container dependencies can break the limitations of traditional methods that assume static structure and uniform data. This enables agile responses and accurate identification of abnormal behaviors. This research direction not only provides theoretical support for cloud-native infrastructure but also lays a solid foundation for building integrated intelligent operations systems with structural awareness, behavioral understanding, and risk identification capabilities. It holds significant engineering value and theoretical importance [4].

Beyond container-specific stream and graph attention studies, recent anomaly-detection work has explored self-attention log modeling, interpretable graph-level detection, multi-scale temporal graphs, and multivariate graph signals [5], [6], [7], [8]. Backend-oriented studies further connect multi-granularity representations with causal localization and extend anomaly modeling through Mamba-LSTM collaboration and compressed representations in ETL and edge-intelligence scenarios [9], [10], [11]. These works motivate robust representation learning, but they still leave room for explicit high-order container dependency modeling that captures dynamic invocation paths and sparse cross-container propagation.

Scheduling and intelligent decision studies offer another reliability perspective. Task-graph modeling, multi-tenant inference scheduling, and reinforcement-learning-based cloud placement frame service execution as coordinated decisions over nodes, dependencies, and resources [12], [13], [14]. Broader work on supply-chain optimization, multi-agent financial decision making, retrieval-augmented model trust, and stable fine-tuning stresses uncertainty control, selective rejection, and representation stability across heterogeneous systems [15], [16], [17], [18]. Accordingly, this study couples graph attention, structural reconstruction, and contrastive learning rather than treating each container as an isolated metric stream.

2. Proposed Approach

The network architecture illustrates a graph attention-based framework for container-level dependency modeling. The overall process includes log input, graph construction, and node embedding initialization. The core module employs multi-head graph attention layers to dynamically model dependencies and adaptively aggregate critical upstream and downstream behavioral features. The optimization is guided by a combination of reconstruction loss, attention entropy regularization, and a contrastive objective, enabling robust modeling and fine-grained discrimination of abnormal propagation paths. The model architecture is shown in Figure 1.

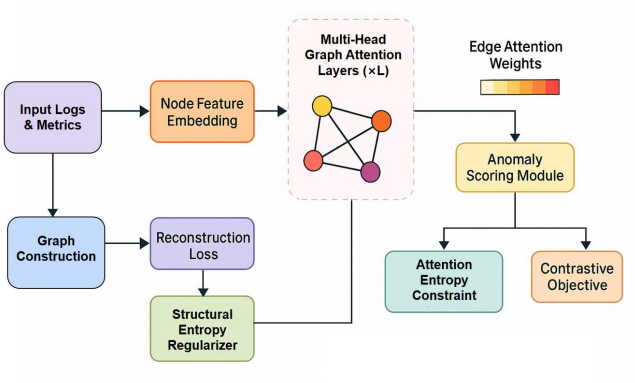

Figure 1. Container Dependency Modeling

In this study, we propose a graph attention network-based approach for container-level behavior modeling and anomaly detection. The goal is to capture complex dependency structures among containers and trace abnormal propagation paths. The method takes invocation relationships and system metrics collected during the container runtime as input. It constructs a dynamic graph to model upstream and downstream interactions between components. To achieve structural awareness and semantic fusion, we first apply nonlinear projection to the initial features of container nodes to obtain embedding representations. These embeddings are then propagated and aggregated based on the topological adjacency in the dependency graph, enabling unified modeling of micro-level behaviors and macro-level dependencies.

The core of the model is to introduce a graph attention mechanism to adaptively adjust the influence of different neighbor nodes on the representation of the target node. Specifically, for the target node, its embedding update process at layer A can be formalized as:

$$h_i^{(l)} = \sigma\Big(\sum_{j \in N(i)} \alpha_{ij}^{(l)} W^{(l)} h_j^{(l-1)} \Big)$$

Among them, $\alpha_{ij}^{(l)}$ represents the normalized adjacency weight learned based on the attention mechanism, $W^{(l)}$ is a learnable linear transformation matrix, and $\sigma(\cdot)$ represents a nonlinear activation function. The calculation of the attention weight is based on the feature correlation between node pairs, using the following scoring function:

$$e_{ij} = \mathrm{LeakyReLU}\big(a^T |\mathrm{W}h_i || \mathrm{W}h_j|\big)$$

Then the scoring results of all adjacent nodes are normalized to obtain the attention coefficient:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in N(i)} \exp(e_{ik})}$$

In the multi-head attention structure, the model calculates the attention weights of multiple subspaces in parallel to improve the diversity and stability of feature expression. Its output is represented by the concatenation or averaging of multiple head outputs:

$$h_i = ||_{m=1}^{M} \sigma(\sum_{j \in N(i)} \alpha_{ij}^{(m)} W^{(m)} h_j)$$

To further improve the expressiveness of the model in anomaly detection, we introduce structure preservation constraints and retain the high-order relationship information in the graph structure through graph reconstruction loss. The loss function is defined as:

$$L_{struct} = \sum_{(i,j)} ||\hat{A}_{ij} - A_{ij}||_2^2$$

Among them, A_{ij} represents the adjacency relationship of the original dependency graph, and $\hat{A}_{ij}$ is the adjacency probability matrix predicted by the model. Through this constraint, the model can retain global structural consistency during the node embedding process, providing more stable structural support for subsequent anomaly detection.

To adapt to the non-stationarity and heterogeneity of container states in real-world environments, we introduce a contrastive learning strategy to enhance the discriminability of node representations. We construct positive and negative sample pairs and minimize the distance between embeddings of similar samples while maximizing the separation between dissimilar ones. This improves the model's sensitivity and robustness to anomalous states. The proposed method demonstrates strong adaptability in multi-scale modeling, structure preservation, and behavior differentiation, providing an interpretable and structurally consistent mechanism for container-level anomaly detection.

3. Performance Evaluation

3.1 Dataset

This study uses the Misuse Detection in Containers Dataset, released on Kaggle in 2024. The dataset originates from real Kubernetes clusters and contains network traffic and system call logs between microservice containers. It meets the research requirements for container-level dependency modeling. The dataset captures information exchanges under both normal and abnormal conditions, which supports the construction of container dependency graphs and the detection of anomaly propagation signals.

The dataset includes multidimensional features for each container node, such as CPU, memory, and I/O usage metrics. It also contains network traffic interactions between containers, including direction, protocol, and volume. Labels for various types of anomalies are also provided. This structured and sequential data aligns well with the modeling mechanisms of graph attention networks, especially for node embedding and dynamic dependency aggregation. It directly supports model training and loss function design.

Overall, the dataset offers several advantages. First, it is based on real data and covers both system behavior and abnormal interactions. Second, it has a rich structure with clear container dependencies. Third, it includes complete labels suitable for supervised or semi-supervised anomaly detection tasks. As the data foundation of the study, it fully supports the experimental validation and methodological design of the proposed graph attention framework for container dependency modeling and anomaly detection.

3.2 Experimental Results

This paper first conducts a comparative experiment, and the experimental results are shown in Table 1.

Table1. Comparative experimental results

Method	Precision	Recall	F1-score	AUROC	AUPR
GACAD(Ours)	94.6	91.8	93.1	97.8	96.4
SAAD [5]	90.2	87	88.5	95.4	93.1
IGAD [6]	88.9	85.3	86.7	93.8	91.7
MT-GNN [7]	91.4	88	89.6	95.9	94.2
GDN [8]	87.1	83.4	85.2	91.7	89.5

The experimental results show that the proposed GACAD method achieves superior performance across all evaluation metrics, with notable advantages in F1-score, AUROC, and AUPR. This performance gain is attributed to the introduction of a graph attention mechanism during container dependency modeling. It enables the model to effectively identify key paths in the dynamic upstream and downstream relationships among multiple containers. As a result, the model improves both the accuracy and sensitivity of anomaly propagation chain modeling. Compared with traditional methods, GACAD achieves a better balance between precision and recall, indicating strong anomaly coverage while maintaining low false positive rates.

When compared with methods such as SAAD and MT-GNN, although they employ self-attention mechanisms and multi-scale graph modeling strategies, respectively, they still suffer from insufficient structural information utilization and unclear detection boundaries. GACAD explicitly models the asymmetric interaction strength between container nodes using multi-head graph attention layers. Combined with node embedding and context aggregation mechanisms, it significantly enhances the ability to detect weak anomaly signals in complex dependency chains. This is especially evident in the results of AUROC and AUPR.

IGAD and GDN exhibit solid capabilities in graph structure modeling. However, their performance is limited when handling non-static container dependencies and high-dimensional system state data. In particular, their recall rates drop when dealing with compound anomalies that propagate across multiple containers. In contrast, GACAD leverages attention-guided information aggregation to focus on behavioral deviation paths within the graph. This improves the recognition of cross-node anomaly patterns.

Overall, the experimental results validate the effectiveness and adaptability of the GACAD method in container-level anomaly detection scenarios. Its strengths go beyond metric improvements and lie in the deep modeling of complex container dependency structures. By incorporating structure-aware graph attention mechanisms and multiple regularization objectives, the method offers a more robust and interpretable solution for anomaly perception and risk monitoring in distributed container systems.

This paper also analyzes the sensitivity of the number of attention heads to the performance of container-dependent

anomaly detection. The experimental results are shown in Figure 2.

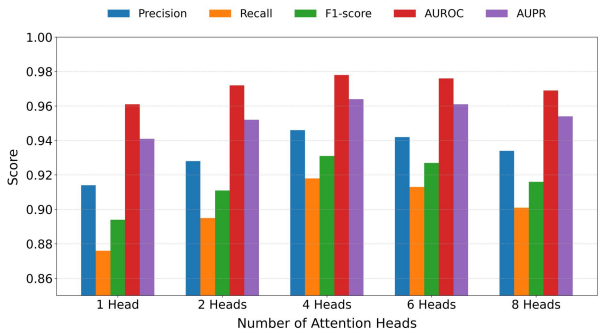

Figure 2. Sensitivity analysis of the number of attention heads on container-dependent anomaly detection performance

As the number of attention heads increases from 1 to 4, the model shows a steady improvement across all five metrics. The gains in F1-score and AUPR are particularly notable. This indicates that the multi-head attention mechanism enhances the model's fine-grained representation of container dependency structures. Multiple attention channels allow the model to aggregate upstream and downstream information from different perspectives. This improves the ability to distinguish abnormal propagation paths from normal interaction patterns. Such multi-perspective perception is especially effective in high-complexity dependency graphs and helps capture sparse but high-impact anomaly chains.

When the number of attention heads exceeds 4, model performance exhibits slight fluctuations. There is a mild imbalance between Precision and Recall. This may be due to excessive attention heads introducing redundant contextual information. As a result, the model's focus on critical dependency paths becomes diluted, which reduces its accuracy in locating local anomalies. In container dependency modeling, anomalies often propagate through complex paths. Too many attention branches can weaken path consistency and reduce the model's sensitivity to structural anomalies.

The best performance is achieved with 4 attention heads. The model shows the most stable results in AUROC and F1-score under this setting. This suggests a good balance between overall anomaly discrimination and local false positive control. The results confirm the role of the attention mechanism in modeling asymmetric dependency strengths between containers. In systems with frequent service calls and highly dynamic dependency graphs, a moderate number of attention heads helps build more robust behavioral decision boundaries.

Moreover, the upward trend in AUPR reflects the model's strong detection capability under sparse anomaly conditions. The attention mechanism enables selective aggregation of high-identifiability nodes along anomaly paths. This increases the expressiveness of anomalies in the embedding space. It plays a key role in detecting subtle but important disruptions in structural anomalies and further enhances the model's practicality and generalization in complex container environments.

This paper also conducted a sensitivity experiment on the node embedding dimension on the anomaly recognition effect. The experimental results are shown in Figure 3.

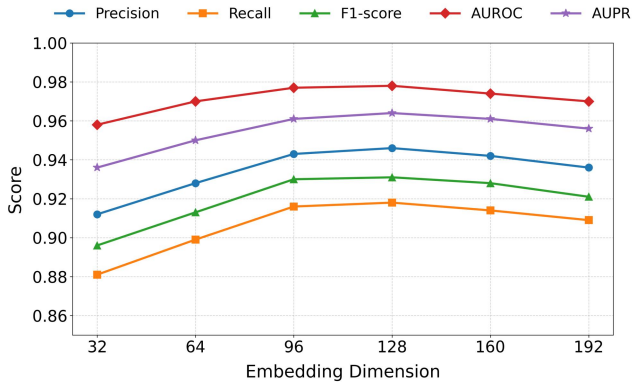

Figure 3. Comparison of the sensitivity of node embedding dimension to the anomaly recognition effect

The experimental results show that as the node embedding dimension increases from 32 to 128, the model exhibits a consistent improvement across all metrics. The gains in F1-score and AUROC are especially pronounced. This trend indicates that a moderate increase in embedding dimensionality enhances the model's ability to capture complex dependencies among containers. Behavioral signals along key paths can be more accurately represented in the high-dimensional space, improving the separability of anomaly propagation chains. Under the guidance of the graph attention mechanism, expanding the embedding space strengthens the expressive power of container nodes within the graph, leading to improved performance in multi-scale behavior modeling.

When the dimension exceeds 128, the model's performance shows diminishing returns and slight declines, particularly in Precision and AUPR. This may be caused by the introduction of redundant features at higher dimensions. Such redundancy can lead to dimensional dilution and semantic interference during attention-based aggregation. As a result, the model's stability in structural anomaly detection may degrade. In complex container dependency graphs, these redundant features can obscure critical contextual signals along anomaly paths. This weakens the structural orientation of abnormal behaviors and reduces the reliability of final anomaly scores.

The best results are achieved when the embedding dimension is set to 128. At this configuration, the model achieves a good balance between Recall and AUROC. This reflects the direct impact of embedding dimensionality on structural anomaly recognition. In container-level dependency modeling, node embeddings serve not only to encode features but also to reconstruct the spatial trajectories of anomaly propagation. Low-dimensional settings limit expressive capacity, while high-dimensional ones increase interference. Therefore, selecting an optimal dimension is essential for designing a model that integrates structural awareness with behavioral discrimination.

In addition, the trend in AUPR reveals that embedding dimensionality is highly sensitive to sparse anomaly detection. When the dimension is properly set, the model can better represent the nonlinear dependencies among a small number of high-risk nodes. This enhances the model's ability to respond to potential anomaly diffusion patterns in the graph. Such a capability is valuable for real-world container scheduling

environments, where high-frequency behavior monitoring is required. In scenarios with chain-like anomaly propagation or cross-node triggering, fine-tuning the embedding dimension becomes a key factor in improving the model's generalization performance.

4. Conclusion

This paper addresses the core challenge of anomaly detection in containerized microservice systems. It proposes a graph attention network-based method for modeling container dependencies and detecting anomalies. The method constructs high-order dependency graphs and integrates multi-head attention mechanisms with structure-aware embedding strategies. It effectively captures key path features within dynamic interaction patterns between containers. This enhances the model's ability to respond to chained anomalies and structural perturbations. In addition, the approach introduces multi-task optimization objectives, including reconstruction loss, attention entropy regularization, and contrastive learning. These improve the precision and stability of anomaly detection from both structural consistency and behavioral discriminability perspectives.

Empirical results across multiple evaluation metrics demonstrate that the proposed method performs well in highly complex container environments. It shows particular strength in detecting anomalies characterized by sparse propagation chains and hidden dependency structures. The graph attention mechanism improves the model's capacity to represent structural heterogeneity and asymmetric upstream-downstream relationships. It also provides an interpretable feature space for reasoning across modules and nodes. This structure-driven and semantics-integrated modeling paradigm offers effective technical support for enhancing service availability, security, and observability in modern cloud-native architectures.

From a practical standpoint, the proposed method has broad application value. It is especially suitable for industries such as finance, energy, manufacturing, and edge computing, where service continuity and anomaly risk control are critical. As the basic execution unit in microservice ecosystems, container-level anomaly detection directly affects scheduling, resource reclamation, and security response mechanisms. The method can be efficiently integrated with existing log analysis, behavior auditing, and resource management systems. It can serve as a core module for structure-aware and behavior-based reasoning in intelligent operations platforms, supporting the construction of full-lifecycle intelligent monitoring capabilities.

Future work will focus on improving the adaptability and generalization of the model in larger and more heterogeneous container platforms. Special attention will be given to modeling anomaly behavior migration in multi-tenant and multi-version service environments. In addition, incorporating spatiotemporal modeling, self-supervised pretraining, and federated learning will enhance the model's learning capacity and security boundary control. Building a unified structure–semantics–behavior modeling system is expected to drive advances in intelligent security for containers. It will enable more intelligent decision-making, real-time responses, and autonomous system management in complex environments, offering strong support for the development of future intelligent infrastructure.

References

[1] R. Wang, H. Qiu, X. Cheng, et al., "Anomaly detection with a container-based stream processing framework for industrial internet of things," Journal of Industrial Information Integration, vol. 35, p. 100507, 2023.

[2] H. Latif-Martinez, J. Suarez-Varela, A. Cabellos-Aparicio, et al., "GAT-AD: Graph Attention Networks for contextual anomaly detection in network monitoring," Computers & Industrial Engineering, vol. 200, p. 110830, 2025.

[3] L. Lin, A. Gu, F. Min, et al., "Multi-Level Graph Attention Network-Based Anomaly Detection in Industrial Control System," Actuators, vol. 14, no. 5, p. 210, 2025.

[4] H. Zhao, Y. Wang, J. Duan, et al., "Multivariate time-series anomaly detection via graph attention network," in Proceedings of the 2020 IEEE International Conference on Data Mining (ICDM), IEEE, 2020, pp. 841-850.

[5] M. Fält, S. Forsström, and T. Zhang, "Machine learning based anomaly detection of log files using ensemble learning and self-attention," in Proceedings of the 2021 5th International Conference on System Reliability and Safety (ICSRS), IEEE, 2021, pp. 209-215.

[6] Y. Liu, K. Ding, Q. Lu, et al., "Towards self-interpretable graph-level anomaly detection," Advances in Neural Information Processing Systems, vol. 36, pp. 8975-8987, 2023.

[7] Z. Ning, Z. Jiang, H. Miao, et al., "MST-GNN: A multi-scale temporal-enhanced graph neural network for anomaly detection in multivariate time series," in Proceedings of the Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data, Cham: Springer Nature Switzerland, 2022, pp. 382-390.

[8] A. Deng and B. Hooi, "Graph neural network-based anomaly detection in multivariate time series," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 5, 2021, pp. 4027-4035.

[9] Y. Yang, C. Wen, H. Cui, Y. Sun, and Y. Liu, "Learning Unified Multi-Granularity Representations for Backend Anomaly Detection and Causal Localization," 2024.

[10] Y. Zhang and E. Cheng, "Anomaly Detection in E-Commerce Multi-Table ETL Processes through Mamba-LSTM Collaborative Modeling," 2025.

[11] X. Zhang and A. He, "Lightweight Anomaly Detection for Edge Intelligence through Compressed Representation Learning," 2025.

[12] H. Cui, "Task Graph Modeling and Constraint-Aware Scheduling for Reliable Large Language Model Agent Execution," 2024.

[13] Z. Huang, J. Luo, and C. Chen, "Learning-Driven Collaborative Scheduling for Multi-Tenant Distributed Inference Services," 2025.

[14] R. Wei and D. Qiao, "Task Scheduling Research for Cloud Computing Infrastructures: A Reinforcement Learning Method Integrating Resource Occupancy Prediction, Dynamic Node State Encoding, and Placement Profit Maximization," 2025.

[15] J. Kou, W. Wang, and Y. Xu, "Collaborative Decision Optimization for Timely Order Fulfillment and Service Quality Enhancement in E-Commerce Supply Chains," 2025.

[16] S. Sun, "Adaptive Stock Trading Algorithms Jointly Driven by Multi-Agent Collaborative Decision Making and Large Language Models in Complex Financial Market Environments," 2025.

[17] Y. Xue, "Enhancing Trustworthiness of Retrieval-Augmented Large Language Models via Confidence Calibration and Selective Rejection," 2024.

[18] H. Zhuang and A. Shen, "Semantic Energy-Guided Stable Fine-Tuning for Distribution-Controlled Generation in Large Language Models," 2025.