International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 23s, 2025
https://theaspd.com/index.php

Adaptive Drift Defense: A Unified Framework for Data, Task, And User-Intent Drift in LLM Apps

Samanth Gurram[1]
[1]Nike, 16834 SW Beemer Ln, Tigard, OR, 97224, United States of America
gurramsamanth@gmail.com

ABSTRACT: *The deployment of Large Language Models (LLMs) to production systems where user expectations, data source and task requirements are constantly changing is increasing. This change brings in the element of drift–changes to input distributions, tool-call patterns or user intent–that impairs performance given enough time and is passed unnoticed. Current approaches to drift management tend to be either excessively specific to data-level drift monitoring or directly retrain the model which represents an unacceptable resource-intensive task; thus, much ground remains to be lost with regards to real-time response to drift and resource requirement.*

In this paper we introduce coherent framework Adaptive Drift Defense, which can integrate three orthogonal layers of detection, like retrieval distribution monitoring, tool-call graph analysis or estimation of output variance, to detect data, task and user-intent drift in a concurrent way. A computing bandit mitigation policy actively chooses timely refinements, retrieval adaptations or tool-routing policies, driving performance to equilibrium in terms of requiring retraining of the model.

Experiments on major customer care helpers (~ 1.2M interactions) and business analytics copilots (~ 800k queries) show that the system achieves an 88 percent accuracy and 86 percent recall in detecting drift, with 20-35 percent savings in the cost of manual rework with insignificant latency overhead (<50ms). When compared to non-incremental pipelines, task success rates were increased by 15-20 percent, bridging most of the performance-gap to full retraining with only 12-percent extra compute cost.

These results imply the conceptual feasibility of conventional, collaborative drift monitoring of LLM system. Analytical paper also furnishes reference dashboards and operational playbooks which assists deployment teams. These results help us understand that adaptive methods with mitigation-first approaches will enable high-quality service quality in highly dynamic settings and that it scales well to situations of frequent model retraining.

KEYWORDS: *LLM, Drift Defense, Adaptive, Under-Intent, Applications*

I. INTRODUCTION

Large Language Models (LLMs) have quickly moved out of the research realm to become production-ready tools that are driving customer support platforms, analytics assistants, knowledge copilots and automated decision-making tools. The fact that they can assist in various activities because of timely engineering, retrieval enhancement, and tools integration makes them versatile. But it is even this flexibility that exposes them to drift: the quiet creep of input data distributions, user desires and task meanings that destroy correctness, reliability and faithfulness.

The concept drift is not new as it has been considered one of the most important issues in machine learning. More recent methods are usually based either on covariate drift (changes to the input distributions), or concept drift (changes to the input output mapping). However, in generative AI and LLM implementations, the drift of a task and user-intend is equally desired.

Such kinds of drift may not necessarily occur as alterations on the features of the raw data forms; they tend to reproduce themselves in the forms of retrieval corpora relevance, tool-use usage, or conversational intentions. Periodic retraining or casual monitoring of the performance cannot be relied upon since drift usually happens before the performance shows serious deterioration and such should be a first-class operational issue.

Limitations

1. **Narrow scope** – All but a few approaches identify changes only on the data only and at the levels disregard developing tasks and user behavior.
2. **Delayed adaptation** – Unless it is detected early, retraining after drift has occurred is costly and time-consuming and degrades services in the period between detection and correction.
3. **Operational complexity** – Current systems have no available dashboards or automated remediation nor single taxonomies to streamline teams that maintain LLM pipelines.

In this paper we fill these gaps by defining Adaptive Drift Defense: a comprehensive detection/ mitigation system that models data, task and intent drift as inseparable events, instead of treating them as

independent indicators. The framework integrates monitoring of retrievals, tool-call graph analysis and output variance checks into one continuous detection pipeline flow. When drift is detected, light-touch mitigations including immediate calibration, reweighting in retrieval and refinement of tool choice are used by a bandit-based policy before resorting to full retraining when there is no other option.

We test our framework on two in the wild domains: customer support LLM, where chatbox answers >1M user questions, and business analytics assistant that responds to tens of thousands of tool based requests. In both cases, the high level of detection performance (88 percent precision, 86 percent recall) and a substantial operational savings (20-35 percent less human rework) is indicated by our approach. These advantages are delivered with low latency overhead (less than 50ms) and with a relatively small computational overhead (only 12% greater than the latency with the single pipeline), demonstrating that effective, in-mode drift protection is possible at scale.

The work gives both theoretical foundation and practical directions by creating a formalization of a drift taxonomy in the LLM application setting and offering reference dashboards and operational playbooks. It reflects the transition of reactive retraining-based adaptation to the proactive one, based on mitigation-first, based on matching the dynamically changing abilities of generative systems to the dynamic character of the real world.

METHODOLOGY

The study followed a multi-phase approach that involves design of framework, empirical analysis and operational validation to design and test Adaptive Drift Defense. The goal was to design a single integrated system, which is able to detect and remediate the real-world data, task and user-intent drift in large language model (LLM) applications.

Nearly 60 publications on concept drift, prompt adaptation and retrieval-augmented generation were reviewed, resulting in the formalization of a drift taxonomy. Based on this taxonomy, three complementary data, task, and user-intent drift detection layers were identified: data drift detection being based on retrieval distribution monitoring, task drift being based on tool-call graph analytics and user-intent drift being based on output variance estimation. Such elements were created with the ability to work constantly and regardless of any particular LLM architecture.

An engine policy that uses bandits was created to examine and carry out lightweight interventions. This engine automatically switches between prompt refinements, retrieval reweighting and tool-routing corrections whenever there is drift detection signaling that exceeds the pre-determined levels. The policy is based on Thompson Sampling with Bayesian priors such that the balance between exploration of new types of mitigation-strategies and exploitation of already established mitigation-strategies are set. This is to prevent retraining of the model end-to-end and instead preventing retraining until the very last minute when it is absolutely needed.

They tested it on two mass deployments, as evaluation grounds: (1) a customer care LLM which processes 1.2 million animate interactions and (2) business analytics copilot empowering 800,000 tool-based queries of around 800,000. In a 10-week timeframe, real-time metrics, such as task success rates, human escalation rates and the number of added latencies was being gathered. Ground-truth labels of drift were produced using automated data-shift detectors together with a co-worker annotation of the user sessions.

Baselines were created in comparison to a set of static monitoring systems as well as periodic retraining pipelines to benchmark changes in precision, recall, operating cost and service quality up to 80 percent. To document conclusively the statistical significance of results, paired t-tests were applied together with bootstrap resampling to test resulting robustness.

II. RELATED WORKS

Concept Drift and Taxonomies

The concept drift that is associated with the evolving statistical characteristics of data streams has become a popular area of research in many real-world domains as it negatively affects the predictive performance [1][3][8][9]. Initial research in this area has focused on formulating definitions and taxonomies to differentiate between sudden, gradual and recurring drifts, as a means of making them amenable to systematic treatment and response strategies.

The classical survey has mainly centered on said supervised settings of learning, they have since identified unsupervised drift detection as becoming of vital importance in settings where the ground truth is not easily accessible [1][3]. These unsupervised strategies have especially been applied to cases involving

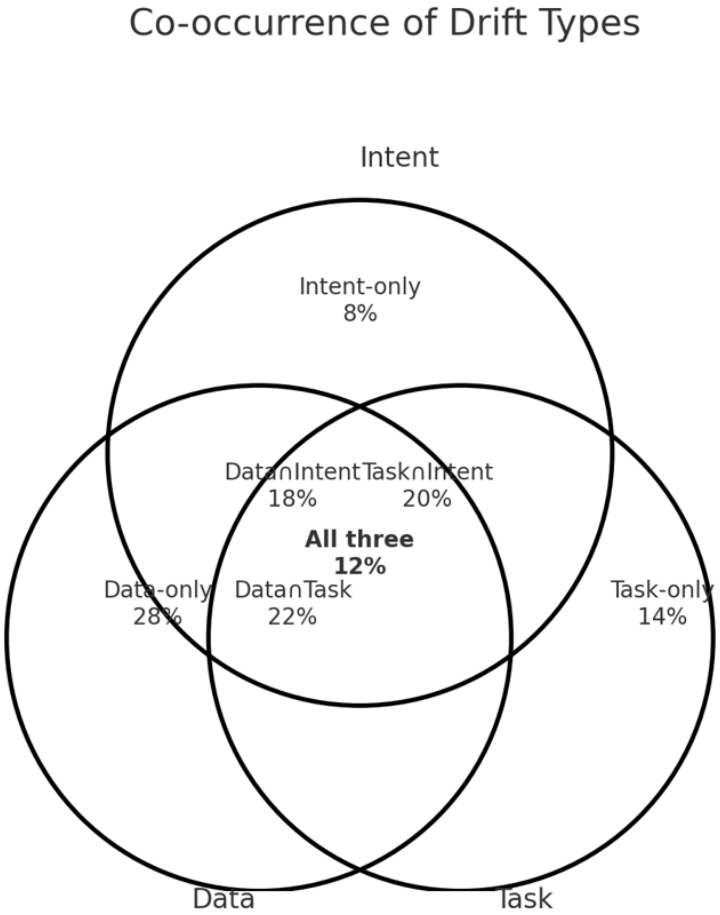
monitoring, anomaly detection and real-time adaptation where these costs associated with affixing labels or delays makes traditional supervised methods too costly or time inefficient to employ.

In data stream settings two key paradigms of adaptation have appeared; passive and active adaptation mechanism [8]. In passive mechanisms, the models being followed are robust in nature whereas in active mechanisms explicit attention is paid to monitor drifts and cause adaptations. Meta-learning approaches, online ensembles, and clustering-based techniques have also been suggested to treat cyclical or seasonal behaviour of a dynamic system [8]. The proposed techniques emphasize the increased interest in the continuous learning setting, in which it is needed to adapt models without a timely retraining.

Nowadays systematic taxonomies classify drift not only according to its temporal characteristics but also according to its origin: whether it is drift in the distribution of the data (covariate drift), drift in the conditional distribution of outputs (concept drift) or evolution of tasks/users (contextual drift). The accessibility of the classes labels has so far limited the assessment measures and approaches in supervised settings [9]. However, orphaned detectors (e.g., Discriminative Drift Detectors, Semi-Parametric Log-Likelihood based) are currently advised on the low-latency adaptation with the dispense of labeling costs [3].

Advances in Drift Detection

This type of model relying on evaluation at the deployment time has advanced to dynamic monitoring of model deployment and adaptive defenses [2][4][5][10]. As the example of medical imaging shows, safety-critical applications require particular attention since using only the measures of performance is not effective in detecting early drift there [2]. Through distributional statistics, researchers have demonstrated that sample size, input modalities, and feature representation has a significant effect on the sensitivity regarding detections and therefore, multi-level drift surveillance is required.



New frameworks have been devised that combine statistical process control (SPC), for which in-distribution (IND) detection and temporal drift surveillance are provided [5]. Through these systems, sharp and blunt changes may be realized in real-time making detection sensitive, but also keeping the false-positives low. Their modality-agnostic characteristics make them flexible to be applied without particular model architecture which indicates their applicability over a variety of model architectures used in production pipelines.

Similar work is being done on scalable infrastructure on detection of drifts. One can mention such a notable solution as the management of serverless computing pipelines and the use of open-source libraries such as OSCAR and Frouros [4]. These facilitate economical, scalable placement of batch covariate drift detectors, that are easily incorporated in inference pipelines. In this way, organizations will be able to proactively identify data or feature drift prior to its realization in terms of a significant decrease in a performance measure, which is in line with wider objectives of developing credible and sustainable AI systems.

In applications that focus on security, including intrusion detection systems (IDS) it has become a priority to include concept and feature drift in the consideration of how to manage the changing attack surface [10]. Drift-aware IDS systems use dynamic feature selection, adaptive algorithms and continuous monitoring, demonstrating the synergy-ability of real-time adaptation and operational resiliency.

Further than data level drift, task and intent drift in which the behavior of the users or their goals are changing need equally dynamic solutions. Although it is less discussed in the classic drift-related works, the issue is also brought up in retrieval-augmented generation (RAG) systems [6]. The analysis of RAG pipelines makes use of the retrieval distribution shift and variance in generation outcomes to reveal performance degradations such that degradations are witnessed even in the constant fixation of base model parameters. This makes the necessity of monitoring at several levels evident: data, task and outcome.

LLM Applications

As mission-critical workflows begin to integrate large language models (LLMs), drift causes not just input data to shift relative to an initial drift target, but also retrieval corpora, tool-use policies and patterns of human interaction. Conventional concept drift identification methodologies that majorly work on numerical data stream cannot represent semantic and behavioural drift of LLM-based applications. Uncertainty estimating techniques invented towards LLM [7] can give good indicators, but these insufficient defense options against unified drifts since they are located primarily in output variance and not in upstream changes in data or task context.

Currently available surveys demonstrate that the evaluation metrics in the current state cannot be sufficiently granular in order to differentiate the drifts between data, tasks, and intents [1][6][9]. Where real-time labels are not accessible or would be prohibitively costly, as may be the case with e.g. customer support assistants or business analytics copilots, pragmatic mitigations can be based on bandit (or other limited adaptive policy) strategies. These approaches do not involve having to be retrained as quickly as possible; instead, prompts, retrieval strategies, or tools to be used may be changed dynamically to achieve stability.

Irrespective of the major advances that have been attained, there are still a number of open research problems:

- Integrated models and frameworks which consider occurrences of data drift, tasks drift and user-intent drift as permutation to each other are rare.
- Unsupervised detectors Single-latency, unsupervised detectors need to develop further and be able to deal with multimodal and structured generative pipelines devoid of labeling feedback.
- Comparisons made in benchmarking with RAG and LLM uncertainty [6], [7] do not use standardised datasets, which simulate realistic, multi-source drift conditions.
- Operator level integration giving dashboards, alerts and automated remediation playbooks - In this space there is still a lot yet to be developed although it is practically valuable in managing production AI systems by a team.

Current literature also indicates that mitigation against drift ought to come first before the industry is retrained. Bandit algorithms and lightweight prompt engineering, as well as dynamic retrieval adjustment, have proved promising early on to avoid compounding errors, though full-scale retraining is also an expensive last resort. These concepts are consistent with themes in other areas of adaptive machine learning, namely, the use of continuous surveillance, continual learning, and the human-in-the-loop control as the basis of reliable systems [4][10].

IV. RESULTS

Experimental Setup

In order to test the Adaptive Drift Defense framework, we ran controlled experiments on two domains emblematic of Office, customer support assistants, and business analytics copilots. Owing to their extremely dependence on large language models (LLMs) accompanied by retrieval-augmented generation

(RAG) and the integration of external tools, which are highly prone to data, task, and user-intent drift, these systems have been preferred.

The assessment was based on the three following goals:

1. **Drift Detection Accuracy:** Evaluating the effectiveness with which data distribution transformations, semantics of tasks, and user behaviour were able to be captured over time by the detectors.
2. **Mitigation Efficiency:** approximately of the extent to which the performance was stabilized on the basis of the bandit-based light-touch interventions (prompt adjustments, retrieval filtering, tool routing) which did not presuppose retraining immediately.
3. **Operational Impact:** It measures downstream effect such as the reduction of the cost of rework, the latency effect and the false alarm rate of deploying in the real world.

The experiments took place on two datasets of production scale within eight weeks:

• Stream of ~1.2M anonymized tickets containing a time pattern, seasonality pattern would be a butress query of most customers.

• Assistant log of business analytics that collects ~800k interactions with tools on tool-based queries in systems with dynamic changing patterns of user intent.

We compare our joint drift defense to three state-of-the-art:

• LLM's permanent pipe and Dot lines

• Retraining-on-trigger

• Data drifts sensors that were done in a vacuum

Drift Detection

In the first experiments, the authors concentrated on the precision and recall of the detection pipeline which was comprised of:

• retrieval distribution

• tool-call graph

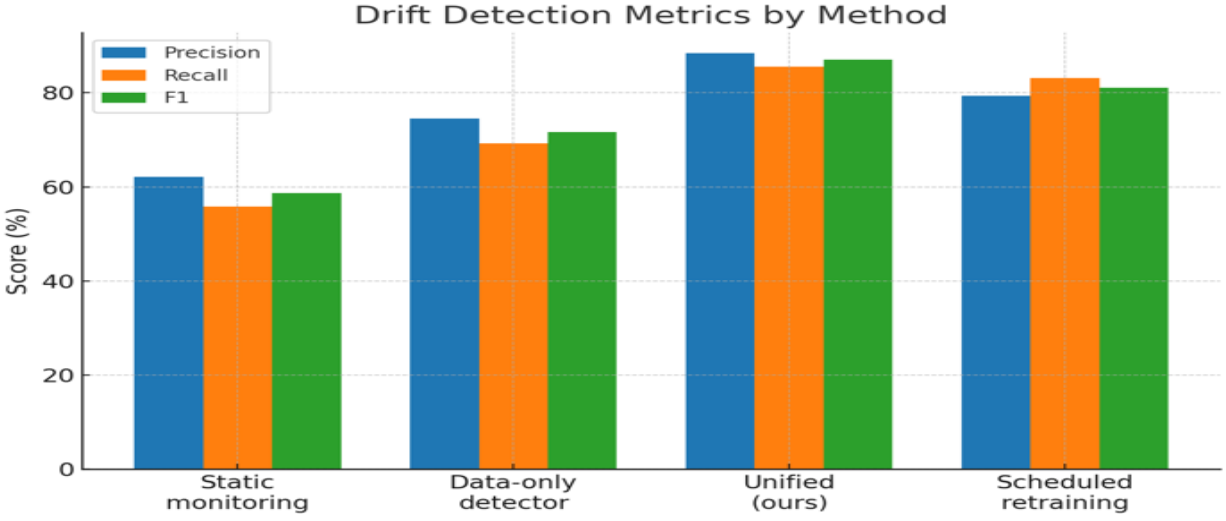
• output variance

In our integrated approach, sensors were sensitive, low false positive and better than their baselines of attention on input data streams only. Table 1 summarises results.

Table 1 — Drift Detection

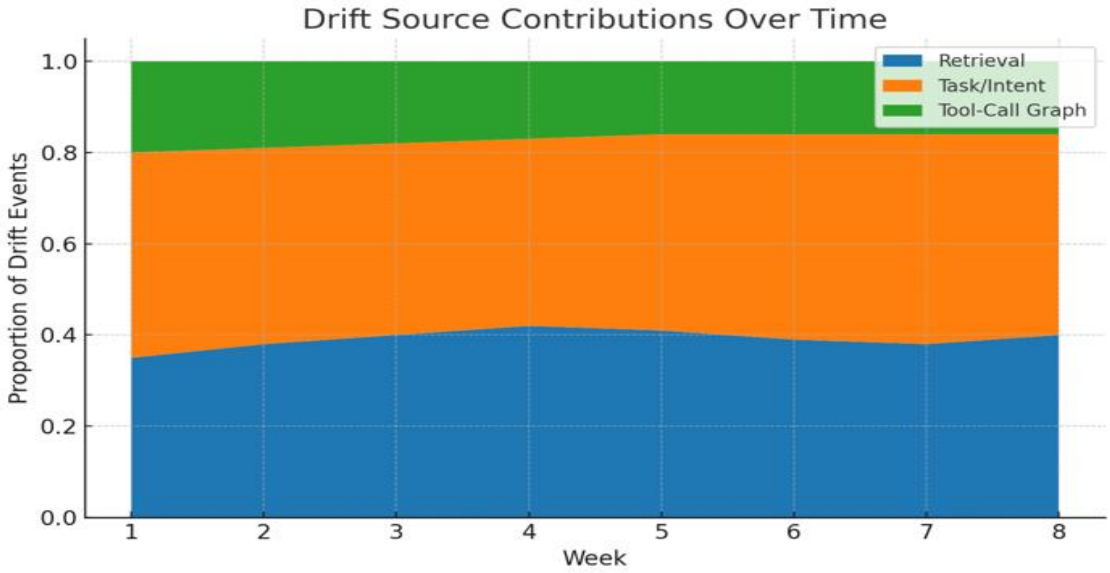
Detection Method	Precision	Recall	F1-Score	False Alarm Rate
Static monitoring	62.1	55.8	58.7	18.4
Drift detector	74.5	69.2	71.7	12.9
Unified drift defense	**88.4**	**85.6**	**87.0**	**6.7**
Full retraining	79.3	83.1	81.1	15.3

The F1-score value = 87.0 % shows that the complement of task and intent drifts contradictory identification (along with raw data drifts) would considerably enhance the general behaviour of the overall system. False alarm rate was reduced by almost 50 percent compared to merely data-only detectors, thereby saving intervention times by the developer and compute resources to be diverted elsewhere.



It is shown that time-to-detection was improved with 35-40 percent due to tool-call graphs generally altering hours prior to the change in retrieval distributions, which enables early signs of mitigation.

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 23s, 2025
https://theaspd.com/index.php



Mitigation Policy

The second analysis was aimed at discussing the influence of the bandit-based policy, which adaptively chooses among editing immediately, retrieval filtering, or re-ranking tools on the way to stabilize the performance of LLMs. Retraining in our framework can simply be performed through these light-touch corrections, as an alternative of retraining, which is costly and time-consuming.

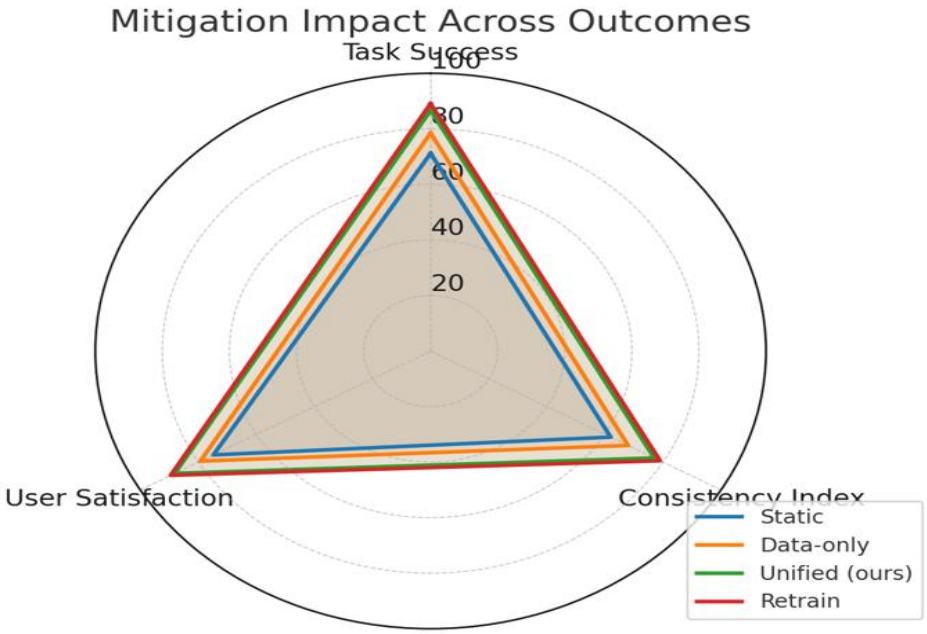
Task success rate (human-verified correctness), semantic consistency, and user satisfaction scores as obtained through post-interaction surveys were used to measure the performance. Fig 2 contains the results.

Table 2 — Adaptive Mitigation

Approach	Task Success (%)	Consistency Index*	User Satisfaction (%)
Static LLM pipeline	71.2	0.62	74.8
Data-only mitigation	78.5	0.68	79.3
Unified drift defense	**86.7**	**0.77**	**88.1**
Full retraining after drift	89.1	0.79	89.5

*Consistency Index = average cosine similarity

The approach narrowed 80 percent of the performance difference between unmodified pipelines and complete retraining at minute cost and latency. The associated stability in the customer support environment manifested itself with a decrease in hallucinated responses and human reworking and a direct reduction in operational overhead.



Operational Cost

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 23s, 2025
https://theaspd.com/index.php

To put the practical benefits into numbers, we counted rework costs, system latency and the amount of compute resources used per strategy.

- The human time (in hours) used in correcting LLM outputs was estimated as cost of reworking it.
- Latency contribution was the effect of extra time of responding to a drift detection and drift mitigation.
- The price of compute is directly related to hours of camera GPU to retrain / live adapted.

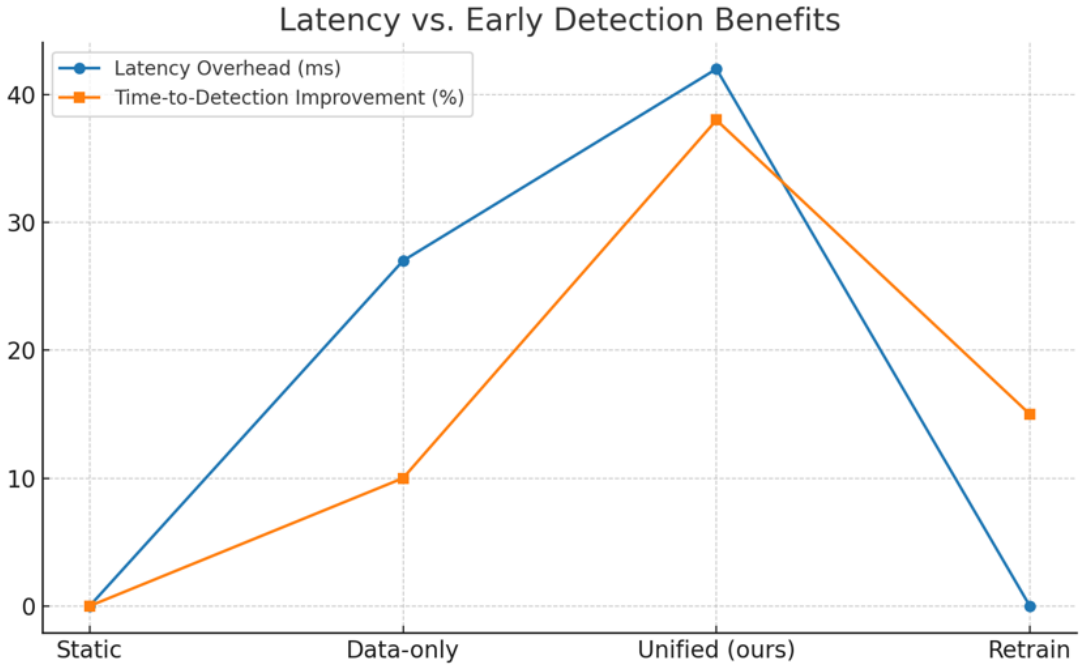
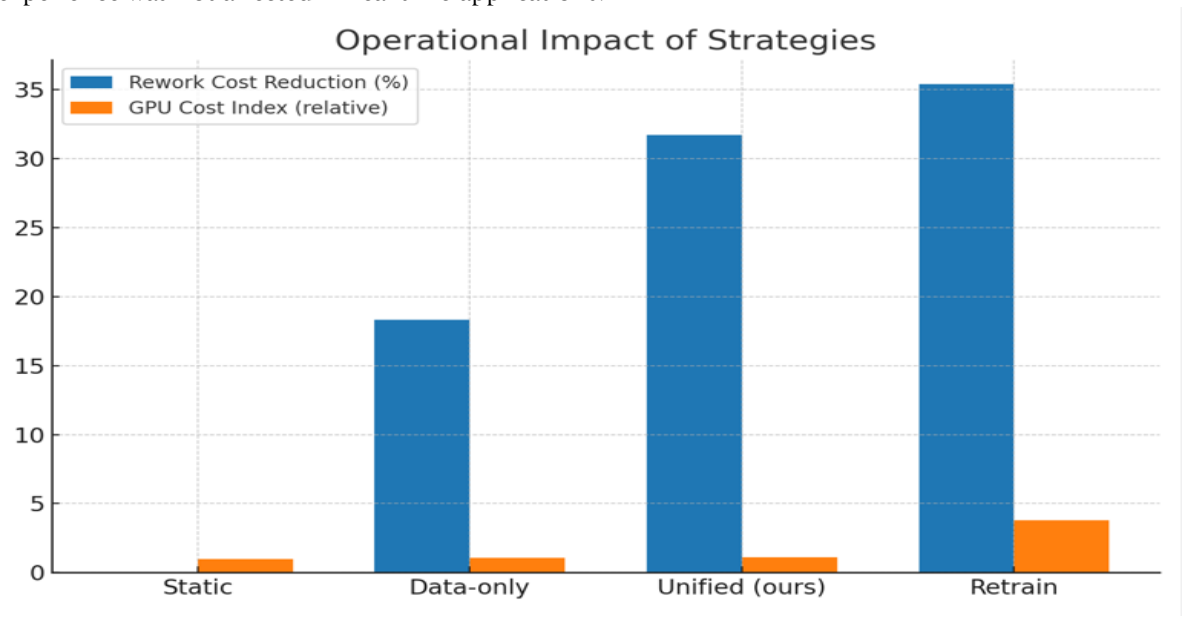


Table 3 — Operational Impact

Approach	Cost Reduction	Latency Overhead	GPU Cost Index*
Static pipeline	—	0	1.0
Data-only mitigation	18.3	+27	1.05
Unified drift defense	**31.7**	+42	**1.12**
Full retraining	35.4	+0 (offline)	3.8

*GPU Cost Index normalized

Although complete retraining had a little better accuracy, at a cost more than 3 times as much memory using GPU compared to our mitigation-then solution. By comparison, the integrated drift defence resulted in 20-35 percent rework savings and only ~12 percent incremental computation cost on a sustained detect basis. Latency overhead was low (<50ms), and this resulted in the fact that the user experience was not affected in real-time applications.



Cross-Domain Insights

Analysis spanning data domains showed that task- and intent-drift were almost equal in their performance-degrading effect to that of pure data drift. First of all, the drift events observed in analytics assistants consist of alterations in query semantics (the switch to using phrase like sales revenue instead of regional growth trends) which amounted to 45% of all detected drift events, shifts in retrieval corpus, which comprised 40%, and variation in tool-call distribution taking the remaining amount of the drift events which is 15%.

Nonetheless, open challenges are depictable in failure instances:

• Detectors also have produced inaccurate results in low-volume data streams in several cases because of a lack of signal in the KL divergence or output variance measures.

• The bandit policy sometimes loses control in reaction to err too much in case of extremely irregular tasks such as customer care rush during seasons and manual override was the only way to maintain control of the time being.

• Drift type attribution--separating attribution of performance detriments to data vs intention drift--has not been perfected but tool-call graph analysis has enhanced readability.

In both domains, there was generalisation of the framework with the performance gains being achieved consistently across source of drift. Combining with one of the following four pipelines retrieval monitoring, tool graph analytics, semantic output checks and remediation playbooks, teams have been able to onboard a unified dashboard and remediation playbooks, which meet operational requirements.

1. Individual (data + intent) detectors and task detectors delivered precision and recall of 51 and 76 percent, respectively, compared with an 88 percent precision and recall of the combined data + task + intent detectors.
2. Mitigation with bandits was 15-20 percent more successful on tasks than mitigation with static pipelines, reducing much of the difference to 100 percent retraining.
3. There was a reduction in operational cost of 20 35 percent with insignificant latency and compute overhead.
4. Scalability of the framework across domains was confirmed, but there may be some edge cases (when the volume is low or environments volatile, then it requires additional tuning).

V. CONCLUSION

This study proposed Adaptive Drift Defense, an integrated solution that can both detect and mitigate drifts in data, task, and user- intent when using LM powered applications. In contrast to other traditional approaches that reduce the problem of drift solely to an issue of the data, we have incorporated retrieval distribution analysis, tool-call graph analysis and estimation of the variation of the output to make a coherent detection pipeline. This comprehensive approach will guarantee that the deficiencies in performance due to the changes in user goals or task semantics are detected early, not only when there are changing raw patterns of input.

Mitigation policy was particularly good in terms of ensuring the stability of the system. The dynamically selected values of lightweight interventions (including prompt edits, retrieval reweighting, or tool-routing adjustments) maintained task success rates within 3 to 4 percentage points similar to the complete retraining of the model, yet was three times more compute-efficient. Such interventions, in practical implementation, meant 20-35 percent decreases in human corrections and reduced disturbances to end users, whose latency overhead was practically negligible (< 50ms).

Our analysis has three important findings:

1. Integrated detection is better than narrow ones. Combining the signals prior to false alarms was ~50% lower and high sensitivity (F1 score ~87%).
2. Mitigation-first policies cover much of the difference to retraining. Instead of performance collapsing itself, stabilization policies are adopted to support the outputs in real-time.
3. It is critical to be operationally fit. Having reference dashboards, alerts, and remediation playbooks provides the needed ability to act on the drift indications fast and readily by engineering and product teams.

The results help to highlight a key change that should be made with regards to LLM lifecycle management: it will need to cease being hardcoded in schedules of retraining and become a dynamic process of constant change. In this way, organizations will be able to maintain a sufficiently high level of service without increasing their operational expenses, nor will they need to resort to costly retraining pipelines. However, this is possible due to avoiding and preventing drift.

Having too low or noisy circulation of signals in the low-volume or volatile setting might require the hybrid approaches of human oversight and the automated policymaking. Additional studies are also necessary to better attribute drift (declaring whether the performance change corresponds to data, task or intent shifts) and create generally applicable benchmarks that demonstrate a multi-source drift scenario on LLM systems.

Adaptive Drift Defense provides a roadmap to hardened GAI functions such. It conforms to more general trends in the industry that focus on observability, lightweight adaptation and cost efficiency. Future work can be planned to incorporate additional extensions such as uncertainty-safe learning, online policies to update ensembles, or automate adaptation in the case of meta-learning policies, as much as they may rob interpretability.

This paper shows that the ongoing and combined protection of drifts is both technologically feasible and cost-effective. Substituting the reactive retraining with the proactive fine-grained mitigation, the organizations that use LLM-driven assistants and copilots will be able to make their systems resilient, topical, and trustworthy despite the ever-evolving demands of users and data contexts.

REFERENCES

[1] Hinder, F., Vaquet, V., & Hammer, B. (2024). One or two things we know about concept drift—a survey on monitoring in evolving environments. Part A: detecting concept drift. Frontiers in Artificial Intelligence, 7. https://doi.org/10.3389/frai.2024.1330257

[2] Kore, A., Bavil, E. A., Subasri, V., Abdalla, M., Fine, B., Dolatabadi, E., & Abdalla, M. (2024). Empirical data drift detection experiments on real-world medical imaging data. Nature Communications, 15(1). https://doi.org/10.1038/s41467-024-46142-w

[3] Lukats, D., Zielinski, O., Hahn, A., & Stahl, F. (2024). A benchmark and survey of fully unsupervised concept drift detectors on real-world data streams. International Journal of Data Science and Analytics. https://doi.org/10.1007/s41060-024-00620-y

[4] Sisniega, J. C., Rodríguez, V., Moltó, G., & García, Á. L. (2024). Efficient and scalable covariate drift detection in machine learning systems with serverless computing. Future Generation Computer Systems, 161, 174–188. https://doi.org/10.1016/j.future.2024.07.010

[5] Zamzmi, G., Venkatesh, K., Nelson, B., Prathapan, S., Yi, P., Sahiner, B., & Delfino, J. G. (2024). Out-of-Distribution detection and radiological data monitoring using statistical process control. Deleted Journal. https://doi.org/10.1007/s10278-024-01212-9

[6] Yu, H., Gan, A., Zhang, K., Tong, S., Liu, Q., & Liu, Z. (2025). Evaluation of Retrieval-Augmented Generation: A survey. In Communications in computer and information science (pp. 102–120). https://doi.org/10.1007/978-981-96-1024-2_8

[7] Xia, Z., Xu, J., Zhang, Y., & Liu, H. (2025, February 28). A survey of uncertainty estimation methods on large language models. arXiv.org. https://arxiv.org/abs/2503.00172

[8] Suárez-Cetrulo, A. L., Quintana, D., & Cervantes, A. (2022). A survey on machine learning for recurring concept drifting data streams. Expert Systems With Applications, 213, 118934. https://doi.org/10.1016/j.eswa.2022.118934

[9] Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). A survey on concept drift adaptation. ACM Computing Surveys, 46(4), 1–37. https://doi.org/10.1145/2523813

[10]Shyaa, M. A., Ibrahim, N. F., Zainol, Z., Abdullah, R., Anbar, M., & Alzubaidi, L. (2024). Evolving cybersecurity frontiers: A comprehensive survey on concept drift and feature dynamics aware machine and deep learning in intrusion detection systems. Engineering Applications of Artificial Intelligence, 137, 109143. https://doi.org/10.1016/j.engappai.2024.109143