Lyapunov-Stable Adaptive Control for Multimodal Concept Drift

Tianyu Bell Pan, Mengdi Zhu, Alexa Cole, Ronald Wilson, Damon Woodard
Department of Electrical and Computer Engineering
Florida Institute of National Security
Applied Artificial Intelligence Group
University of Florida
Gainesville, FL 32611
{tpan1, zhum, alexa.cole, ronaldwilson}@ufl.edu
dwoodard@ece.ufl.edu

Abstract

Multimodal learning systems often struggle in non-stationary environments due to concept drift, where changing data distributions can degrade performance. Modality-specific drifts and the lack of mechanisms for continuous, stable adaptation compound this challenge. This paper introduces LS-OGD, a novel adaptive control framework for robust multimodal learning in the presence of concept drift. LS-OGD uses an online controller that dynamically adjusts the model's learning rate and the fusion weights between different data modalities in response to detected drift and evolving prediction errors. We prove that under bounded drift conditions, the LS-OGD system's prediction error is uniformly ultimately bounded and converges to zero if the drift ceases. Additionally, we demonstrate that the adaptive fusion strategy effectively isolates and mitigates the impact of severe modality-specific drift, thereby ensuring system resilience and fault tolerance. These theoretical guarantees establish a principled foundation for developing reliable and continuously adapting multimodal learning systems.

1 Introduction

Multimodal learning combines information from various data modalities, such as text and images, to enhance understanding and predictions. Real-world applications are naturally multimodal, such as autonomous systems leveraging cameras, LiDAR, and IMUs [22]; driver monitoring systems integrating visual and physiological data [10]; human-machine interaction using EOG and speech [17]; and online platforms handling diverse media [21]. A significant challenge arises when these systems operate in non-stationary environments, where the underlying data distribution changes over time, known as concept drift. By definition, concept drift is a change in the statistical properties of the target data distribution over time [18]. Practically, a model's past experiences may no longer be valid as new patterns, topics, or styles emerge.

Concept drift is prevalent in real-world streaming data. For example, seasonal changes require weather prediction models to be adjusted periodically; shifting user preferences impact recommender systems; and social media or news topics can "drift" over weeks or months [18]. In multimodal misinformation detection, concept drift might appear as new vocabulary or slang in text, or new image memes and deepfakes in visuals that were not present in the training data. These shifts in distribution can degrade a model's performance if it fails to adapt. Traditional machine learning models typically assume a stationary data distribution; therefore, the model's error rates increase when drift occurs, and previously learned decision boundaries may become misaligned.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

Despite the prevalence of concept drift, current multimodal learning systems largely lack mechanisms to adapt to data distribution changes continuously. Most research on adapting to concept drift has focused on unimodal data streams (such as purely numeric or textual data) [38]. Multimodal scenarios introduce new challenges that have been underexplored: (1) **Modality-specific drifts:** One modality may experience drift while others remain stable, for example, the content of images may change while the relevance of the text stays constant. This situation requires adaptive fusion strategies that are unnecessary in single-modality settings. Traditional fusion techniques (like early or late fusion) rely on fixed combination rules, which become suboptimal under drift [3]. (2) **Detection complexity:** Changes can be subtle and vary across modalities, complicating drift detection. Standard drift detectors often monitor a single error or a distribution metric [18]. In the context of multimodal data, the question arises: Which signal indicates drift? How can we determine whether the drift originates from the textual or visual channel? (3) **Lack of theoretical guarantees:** Adaptation mechanisms in machine learning are often heuristic. There is a need for Lyapunov stability guarantees, which are essential in control theory for maintaining stable adaptive systems; however, these principles have not been fully applied to multimodal machine learning dealing with concept drift. Traditional adaptive control theory provides tools to ensure an adaptive system remains stable [7], but these tools need further exploration in multimodal machine learning. (4) **Computational constraints:** In real-world applications like content filtering systems, models must adapt in real-time without extensive retraining. Many existing concept drift solutions retrain a new model on recent data or reset parts of the model, which can be inefficient or impractical in live systems. There is a pressing need for mechanisms that can continually, safely, and efficiently adjust models as new data becomes available.

This paper addresses the above challenges by proposing a novel framework for Lyapunov-stable adaptive multimodal learning under concept drift (LS-OGD). Our key contributions are:

Formulation of Multimodal Drift Adaptation: We present, to the best of our knowledge, the first formal integration of concept drift theory into a multimodal learning setting that combines text and image data. We define the multimodal concept drift problem and discuss the challenges that arise when the informational value of one modality changes over time. Additionally, we identify shortcomings in current fusion methods and emphasize the necessity for an adaptive fusion mechanism capable of dynamically adjusting the contributions of each modality.

Adaptive Control Strategy for Learning: We have developed an online adaptation controller that monitors the model's performance and dynamically adjusts two critical elements of the learning process: (1) the model's learning rate, which governs how quickly the model learns from new data compared to how well it retains information from past data, and (2) the fusion weighting between different modalities, which prioritizes the more reliable modality when drift is detected in the other. This controller is rule-based and lightweight. It detects drift through statistical changes in the model's error over time and triggers adjustments to the learning rate (η) and the fusion coefficient (α). Our approach treats the prediction error as a feedback signal and establish adaptation rules similar to a PID controller to minimize the error.

Lyapunov Stability Analysis: A key theoretical contribution of our work is developing a Lyapunov function for the adaptive learning system. We present two new theorems that establish conditions for the stability of the closed-loop learning process (comprising the model and the controller) in the sense of Lyapunov. Specifically, we demonstrate that, under mild assumptions, the system's error will remain bounded during periods of continuous drift and will converge to zero (or to an arbitrarily small value) once the drift ceases.

2 Related Work

Concept Drift Detection and Adaptation: Drift detection techniques fall into three categories: (1) Statistical Input Monitoring: testing distribution differences between recent and earlier data using methods like Kullback-Leibler (KL) divergence [8] or adaptive windowing (ADWIN) [19]; (2) Performance Monitoring: tracking error rates against statistical thresholds [12, 11, 2, 36], effective in supervised settings; and (3) Parameter Tracking: adaptation strategies include model retraining, prioritizing recent data, or using ensembles where new learners handle emerging patterns [1, 4, 20]. Most methods are designed for single-modality streams and may not suit multimodal systems where drift can affect individual modalities.

Recent advancements in deep learning have spurred research into concept drift in neural networks and large models. While continual learning and concept drift address learning from non-stationary data, continual learning typically prevents catastrophic forgetting of previous tasks. In contrast, concept drift emphasizes rapid adaptation to evolving data. Some studies, like [38], propose extending concept drift theory to multimodal language models, recognizing their vulnerability to gradual shifts and sudden out-of-distribution events. During pre-training, they introduced a drift adapter module to address drift, highlighting a growing focus on this issue in complex AI systems. However, their approach primarily targets pre-training bias rather than real-time adaptation. Our work stands apart by addressing online drift during deployment while ensuring stability.

Multimodal Fusion and Learning Techniques: Combining modalities improves prediction but presents fusion challenges [3]. Methods include early fusion (feature-level merging), late fusion (decision-level integration), bilinear pooling, attention mechanisms, and cross-modal transformers [25]. In fake news detection, specialized architectures have evolved from EANN's [33] adversarial training for topic-invariant representations to SpotFake [27] and MMDFND's [30] attention-based approaches. Recent bidirectional cross-modal fusion (BCMF) [39] enables information flow between text and image encoders, representing significant progress in multimodal integration techniques.

Despite recent advancements, most multimodal models operate under a fixed fusion strategy once trained. During training, the model learns to weigh and utilize different modalities, but these weights remain static afterward. If a particular modality's relevance changes over time, the model cannot easily reconfigure itself due to drift. For example, a fake news detection model might initially rely heavily on textual patterns to identify deception. However, if bad actors use innocuous text paired with deceptive images over time, the model's original fusion scheme could misallocate attention and risk missing the visual cues. Some recent works [25, 37, 15] have explored dynamic modulation of importance; for instance, a model might use an attention mechanism to emphasize the modality that provides a more significant "signal" for a given sample. However, this approach usually focuses on the content of the sample itself rather than detecting distributional shifts over time. In contrast, our proposed method explicitly adjusts fusion weights in response to drift over time. We conceptualize this as an adaptive late fusion approach, introducing a controllable parameter $\alpha(t)$ to interpolate between modalities. This idea of adaptive fusion has limited precedents; related concepts include gating networks or conditional fusion, in which an auxiliary network predicts fusion weights for each sample. We go further by making α a function of time and past errors, rather than relying on the current sample, thus directly addressing the temporal drift in modality utility.

3 Preliminary and Problem Formulation

Concept Drift Definition: We consider a supervised learning problem in an online setting. At each time step $t = 1, 2, 3, \dots$, the system receives a new sample $(x_t^{(1)}, x_t^{(2)}, \dots, x_t^{(M)}, , y_t)$, where $x_t^{(M)}$ denotes the input from the M-th modality (for instance, $M = 2$, $x^{(1)}$ is text and $x^{(2)}$ is image), and y_t is the ground-truth label or target. The crucial aspect is that the data distribution is time-dependent. Let $\mathcal{D}_t$ denote the distribution generating $(x_t^{(1)}, \dots, x_t^{(M)}, y_t)$ at time t. Concept drift means $\mathcal{D}_t$ changes over time: there exists some t and $t + \Delta$ for which $\mathcal{D}_t \neq \mathcal{D}_{t+\Delta}$. Following standard definitions, a concept drift occurs at time t if the joint probability $P_t(X^{(1)}, \dots, X^{(M)}, Y)$ is statistically different from $P_{t+1}(X^{(1)}, \dots, X^{(M)}, Y)$. These changes can be abrupt or gradual. This paper assumes a piecewise-constant drift for theoretical analysis (drift occurs at discrete points, dividing the timeline into relatively stable distribution segments) and later simulates the drifts.

Multimodal Learning Model: Define a model $f_\theta(x^{(1)}, x^{(2)}, \dots, x^{(M)})$ with parameters θ that produces a prediction $\hat{y}_t$ given the multimodal input, and the model can be decomposed by modality. For instance, for $M = 2$, we have two sub-networks: a text encoder $f^{(1)}(x^{(1)}; \theta_1)$ and an image encoder $f^{(2)}(x^{(2)}; \theta_2)$, producing modality-specific feature representations. A fusion function then combines these. In our design, the fusion is a weighted combination: let $z_t^{(1)}$ and $z_t^{(2)}$ be scalar logit outputs (final hidden features) from the text and image pipelines, respectively. We define the fused logit

$$z_t = \alpha_t \cdot z_t^{(1)} + (1 - \alpha_t) \cdot z_t^{(2)}, \tag{1}$$

where $\alpha_t \in [0, 1]$ is a fusion weight at time t. The prediction $\hat{y}_t$ is then obtained by applying a sigmoid or softmax to z_t. The parameter α_t effectively controls the influence of each modality:

$\alpha_t = 0.5$ would weight them equally, $\alpha_t \to 1$ relies mostly on modality 1, and $\alpha_t \to 0$ relies on modality 2. The rest of θ encompasses the weights of $f^{(1)}$, $f^{(2)}$, and possibly any additional layers (e.g., if the features are concatenated and passed to an MLP, those MLP weights are in θ).

Loss and Error Measures: Let $L(\hat{y}_t, y_t)$ be the loss at time t (e.g. cross-entropy loss for classification). We define the instantaneous prediction error e_t as the difference between the predicted output and the true output in an appropriate sense. For analysis, assume e_t is a real-valued error signal (for instance, e_t could be the difference in probability assigned to the true class vs. the ideal, or some bounded transformation of the loss). We assume $e_t = 0$ corresponds to perfect prediction, and larger $|e_t|$ means worse performance. The exact definition of e_t will be specified in proofs, but intuitively, one can think of e_t as proportional to the classification error or loss at time t. We also define a desired equilibrium for this error: we desire $e_t \to 0$. However, under drift, the minimum achievable error might change over time as the optimal model changes.

Adaptation Controller: We augment the learning process with an adaptive controller that adjusts two sets of parameters over time: the model weights θ are updated via stochastic gradient descent (or a similar optimizer) with a learning rate η_t, and the fusion weight α_t is also updated on a slower timescale. The controller has two primary responsibilities: drift detection and parameter adaptation.

Drift detection detects when a significant increase in error e_t indicates a possible concept drift. This could be done by checking if e_t exceeds some threshold or deviates significantly from a long-term error trend. In our implementation, we use a sliding window over the recent W samples to compute

$$\bar{e}_t = \frac{1}{W} \sum_{i=t-W+1}^{t} e_i. \tag{2}$$

The controller flags a drift onset if $\bar{e}_t$ exceeds a dynamic threshold (for example, mean error + k standard deviations in a reference period). Furthermore, parameter adaptation works when drift is detected, the controller computes adjustments $\Delta\eta$ and $\Delta\alpha$. Specifically, we define update rules:

$$\eta_{t+1} = \eta_t + \Delta\eta_t, \tag{3}$$

$$\alpha_{t+1} = \alpha_t + \Delta\alpha_t, \tag{4}$$

to reduce future errors.

The adaptation logic is as follows: (1) Learning rate adaptation: If the error is increasing, this suggests the current model is not keeping up with changes. The controller increases the learning rate: $\Delta\eta_t$ is set to a positive value proportional to the error increase and possibly the error magnitude. A simple rule could be $\Delta\eta_t = k_\eta(e_t - e_{t-1})$ when $e_t > e_{t-1}$, for some gain $k_\eta > 0$. To prevent divergence, we also cap η_t not to exceed a safe maximum. Conversely, if error starts decreasing (model catching up), the controller may slowly decrease η to a normal level to avoid overshooting (much like lowering gain as we approach a setpoint). This mechanism ensures fast learning during drift and stable convergence afterward. (2) Fusion weight adaptation: The controller adjusts α_t to put more weight on the modality that currently appears more reliable. We can estimate modality-specific error contributions. For instance, we can compute an "imagined" prediction using only modality 1, $\hat{y}_t^{(1)} = \text{sigmoid}(z_t^{(1)})$, and similarly $\hat{y}_t^{(2)}$ using only modality 2. By comparing these to y_t, the controller can judge which modality alone would perform better on recent data. Suppose modality 1's predictions are significantly more accurate than modality 2's. In that case, it implies modality 1 carries the useful signal and modality 2 might be drifting or noisy, so we increase α to rely more on modality 1. In the opposite case, we decrease α. Formally, one could maintain separate error metrics $e_t^{(1)}$ and $e_t^{(2)}$ for the two modalities. Then, $\Delta\alpha_t$ can be set by a rule like $\Delta\alpha_t = k\alpha\big(e_t^{(2)} - e_t^{(1)}\big)$: if modality 2's error is higher, $\Delta\alpha_t$ is positive (increase weight on modality 1); if modality 1 error is higher, $\Delta\alpha_t$ is negative (shift weight to modality 2). We clamp α_t within [0,1]. This update might be done conservatively (lower frequency or smaller step size) than the learning rate, since fusion changes too often could destabilize training. This slow actuator nudges the model's attention towards the currently best modality.

State-Space Representation: We can conceptualize the system's evolution with a state vector that includes at least the error and possibly the parameter deviations. For analysis, consider the state as $x_t = e_t$ (the error at time t). The error evolves depending on the changing data distribution and our adaptive updates. We update model weights online using mini-batch stochastic gradients. A simplified dynamics can be written as:

$$e_{t+1} = F(e_t, \alpha_t, \eta_t, ; \delta_t), \tag{5}$$

where δ_t represents the disturbance due to concept drift (i.e., how the optimal model's output would change from t to $t+1$ due to distribution shift). The exact form of F is complex since it involves the data and the model's optimization landscape. However, we can reason qualitatively: if there were no drift ($\delta_t = 0$) and the model perfectly matched the current concept, then ideally e_t would be zero for all t. If the model is imperfect but we fix α and a small η, gradient descent will tend to reduce e_t over time toward some residual error (training convergence). Under drift ($\delta_t \neq 0$), the error will increase unless η_t is large enough to track the change. Our controller effectively modulates F by changing η_t and α_t.

Furthermore, we introduce a candidate Lyapunov function to analyze stability:

$$V(t) = \frac{1}{2} e_t^2. \tag{6}$$

This V is always nonnegative and $V = 0$ only when the error is zero, meeting the criteria of a Lyapunov function candidate. We aim to design the adaptation laws ($\Delta\eta, \Delta\alpha$ as functions of e and possibly Δe) such that $V(t)$ does not increase significantly and preferably decreases after some transient, despite drift.

Proposed Algorithm: Algorithms 1 and 2 in Appendix B outline the main online learning process and the controller logic. Figure 1 also presents the proposed system architecture.

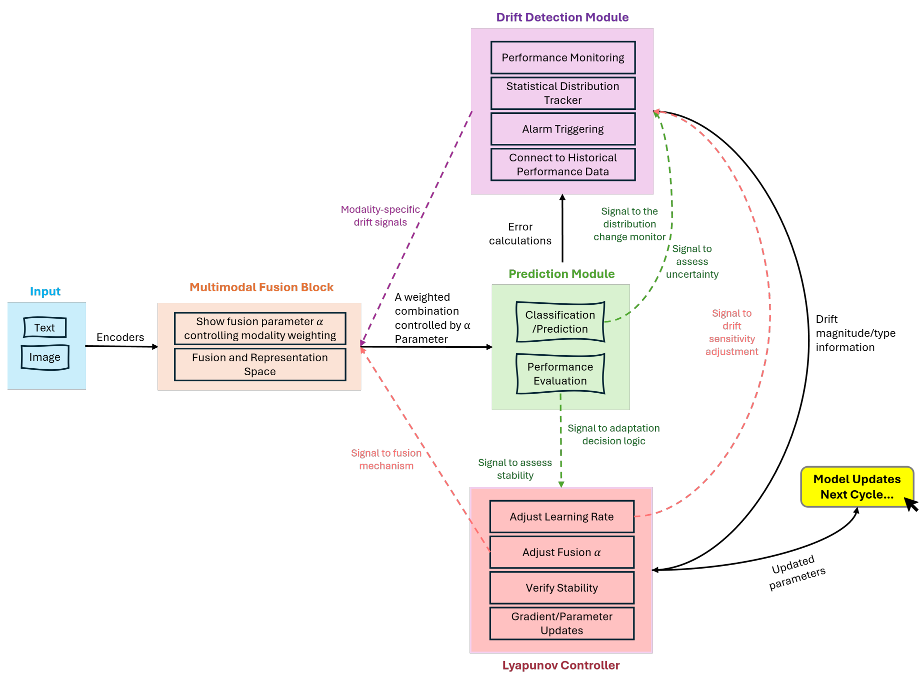

Figure 1: Illustrations of the proposed system architecture. Dashed and solid arrows represent the feedback signals and main data flow, respectively.

4 Main Results

This section presents the theoretical analyses of the proposed LS-ODG framework. Our analyses establish the learning system's stability and adaptability when operating in non-stationary multimodal environments.

4.1 Foundational Axioms and System Properties

To analyze the main framework, three axioms are defined:

Axiom 1 (Bounded Drift Rate) Concept drift, represented by the perturbation δ_t in the error dynamics or by the rate of change of optimal model parameters $||\theta^*_{t+1} - \theta^*_t||$, is assumed to be bounded. That is, there exists a constant $\delta_{max} > 0$ such that $||\delta_t|| \leq \delta_{max}$ for all t.

This axiom is fundamental because an arbitrarily fast or large drift would make it impossible for any adaptive system with finite learning capacity to maintain low error. It ensures that the "target" the system is trying to track does not move so erratically as to preclude effective adaptation.

Axiom 2 (Sufficient Model Capacity) The multimodal learning model f_θ possesses sufficient capacity such that for any stationary concept $\mathcal{D}$ encountered (i.e., a period where $\mathcal{D}_t$ is fixed), there exist optimal parameters θ^* and a fusion weight α^* that can achieve the minimum error $e_t(\theta^*, \alpha^*)$.

This axiom ensures that performance limitations are primarily due to drift or adaptation transients, not due to the model's inherent inability to learn the underlying concepts. It allows us to focus the analysis on the dynamics of adaptation rather than on model approximation errors for a fixed concept.

Axiom 3 (Lipschitz-like Error Dynamics) The error dynamics function $F(e_t, \alpha_t, \eta_t; \delta_t)$ and the loss landscape exhibit local Lipschitz continuity with respect to e_t and the adaptable parameters θ (implicitly via η_t) and α_t. This means that small changes in parameters or error lead to proportionally bounded changes in the subsequent error or its gradient.

This axiom ensures that the gradient-based updates and the adaptive adjustments lead to predictable changes in the error, preventing chaotic or overly sensitive responses. This helps analyze the system using local approximations and ensures that the Lyapunov function's evolution is well-behaved.

4.2 Characterizing the Evolution of the Lyapunov Function

Before presenting the main stability theorem, we introduce Lemma 1. This Lemma characterizes the change in the Lyapunov function $\Delta V(t) = V(t+1) - V(t)$ over one time step, under the influence of the learning process, the adaptive controller, and the external drift.

Lemma 1 (Bounded Increment of the Lyapunov Function with Adaptation) Given the system dynamics $e_{t+1} = F(e_t, \alpha_t, \eta_t; \delta_t)$ and the adaptive update rules for η_t and α_t, as described in Section 3 and Algorithm 2. Assuming Axiom 1-3 hold, and the learning rate η_t is positive and the adaption gains k_η and k_α are chosen within stable regions, then the change in the Lyapunov function, $\Delta V(t) = V(t+1) - V(t)$, can be bounded as:

$$\Delta V(t) \leq -\gamma_1 \eta_t e_t^2 + \gamma_2 \eta_t \left| e_t \right| \left\| \delta_t \right\| + \gamma_3 \eta_t^2 \left(\mathcal{G}\left(e_t, \alpha_t \right) + \left\| \delta_t \right\|^2 \right) + \mathcal{O}\left(\left\| \Delta \alpha_t \right\| \right), \tag{7}$$

where γ_1, γ_2 and γ_3 are positive constants.

In Lemma 1, the first term $-\tilde{\gamma_1} \eta_t e_t^2$ is referred to as the stabilizing term and reflects the error reduction due to the learning algorithm attempting to minimize the current loss. γ_1 is related to the learning effectiveness, i.e., a local convexity or gradient dominance property from Axiom 3. The second term $\gamma_2 \eta_t |e_t| ||\delta_t||$ is referred to as the disturbance term, which measures the impact of concept drift on the error. The third term $\gamma_3 \eta_t^2 \left(\mathcal{G}\left(e_t, \alpha_t \right) + \left\| \delta_t \right\|^2 \right)$ is the noise term, evaluating the nonlinearities or gradient stochasticity in SGD. The final term $\mathcal{O}\left(\left\| \Delta \alpha_t \right\| \right)$ is referred to as the fusion adaptation term, representing the effect of changing fusion weight on the error.

For the candidate Lyapunov function $V(t) = \frac{1}{2} e_t^2$, Lemma 1 establishes that the stabilizing effect of learning must overcome all three other terms to decrease error. This balance is mediated by the adaptive learning rate η_t. For instance, if $|e_t|$ is large, the controller tends to increase η_t, therefore increasing the stabilizing term. Additionally, since $\Delta V(t)$ remains negative when $|e_t|$ is sufficiently large, even when $\delta_t \neq 0$, Lemma 1 forms the basis for proving the uniform ultimate boundedness (UUB). The asymptotic stability is achieved without concept drift (i.e., $\delta_t = 0$).

Remark 1 (On the Choice of Lyapunov Function and Adaptation Gains) The choice of $V(t) = \frac{1}{2} e_t^2$ is an intuitive selection for tracking error, relating the "energy" of the system to the squared prediction error. The condition in Lemma 1 and Theorem 1 regarding "appropriately small" adaptation gains (k_η, k_α) is common in adaptive control. If gains are too large, the system might overreact to errors or drift, leading to oscillations or instability, violating the conditions needed for $\Delta V(t)$ to be reliably negative or bounded as desired. The theoretical existence of such stable gain regions is key, and practical selection often involves tuning guided by these theoretical constraints.

4.3 System Stability and Error Convergence under Bounded Concept Drift

The first main theoretical result concerning the overall stability of the adaptive learning system is presented with the Axioms and Lemma 1.

Theorem 1 (Lyapunov Stability under Bounded Drift) Consider an adaptive multimodal learning system with error dynamics characterized by $e_{t+1} = F(e_t, \alpha_t, \eta_t; \delta_t)$, operating under Axioms 1-3.

Assume that the adaptive controller adjusts η_t and α_t such that Lemma 1 holds, and the adaptation gains k_η and k_α are appropriately small, then the closed-loop learning system is Lyapunov stable (i.e., the prediction error e_t is UUB). This implies that the ultimate bound on the error ε_B and the bound on the initial error B_0 exist, and are positive constants, such that the error trajectory would remain bounded, and eventually stays within the ultimate bound ε_B if $|e_{t_0}| < B_0$, where e_{t_0} denotes the initial error.

Theorem 1 establishes a formal Lyapunov stability guarantee in adaptive multimodal learning under concept drift for the first time. Unlike heuristic approaches, this provides a mathematical foundation for the system's behavior. The UUB acknowledges that the error may not always converge to zero in persistent drift. Instead, UUB guarantees that the error will eventually enter and remain within a small, predictable region around zero. The size of this region, ε_B, will depend on the magnitude of the drift (Axiom 1) and the system's capacity to adapt. Also, this theorem guarantees that if the environment stabilizes (i.e., $\delta_t \to 0$), the system will fully recover, with the prediction error converging to zero. This demonstrates the system's ability to not only manage ongoing drift but also achieve optimal performance once stability is restored.

Corollary 1 (Asymptotic Stability in Stationary Environments) If the environment is stationary from the outset (i.e., $\delta_t = 0$ for all $t \geq t_0$), and Axioms 2-3 hold, then the LS-OGD system with an appropriately chosen, potentially constant, learning rate η and fusion weight α ensures that the prediction error e_t converges asymptotically to zero (i.e., $e_t \to 0$ as $t \to \infty$).

This corollary simplifies Theorem 1 for the non-drifting case. It confirms that the learning mechanism within LS-OGD is sound and can achieve optimal performance when the environment is stable. The UUB property of Theorem 1 then extends this, showing robustness by guaranteeing bounded error even when the stationarity assumption is violated by bounded drift.

4.4 Targeted Adaptation of the Fusion Weight

The second main theorem concerns the system's ability to adapt its fusion strategy. This relies on the controller's ability to adjust α_t correctly. Hence, Lemma 2 is presented as a key for Theorem 2.

Lemma 2 (Convergent Dynamics of Fusion Weight α_t under Unilateral Modality Drift) Given the fusion weight α_t on modality 1, the fusion weight on modality 2 is defined as $1 - \alpha_t$.

Consider the fusion weight update rule:

$$\alpha_{t+1} = \mathrm{clip}_{[0,1]}\left(\alpha_t + \Delta\alpha_t\right), \quad \Delta\alpha_t = k_\alpha\left(e_t^{(2),\mathrm{est}} - e_t^{(1),\mathrm{est}}\right) \tag{8}$$

where $e_t^{(m),\mathrm{est}}$ being the estimated error from modality m, and $k_\alpha > 0$ is the adaptation gain.

Following the update rule for fusion weight, if modality 1 becomes less reliable than modality 2, leading to $e_t^{(1),\mathrm{est}} > e_t^{(2),\mathrm{est}} + \epsilon_\alpha$ for $t > t_0$ for some positive margin ϵ_α, then:

(i) α_t will tend to decrease.

(ii) Provided k_α is sufficiently small to ensure smooth adaptation and prevent overshoot. The error estimation difference is persistent, α_t will converge towards 0 or a small value close to 0 if the error difference is not consistently large or noise is present in the error estimates.

Lemma 2 describes how the controller adjusts α_t as one of the modalities is less reliable. The update rule is a form of gradient descent on an implicit objective that seeks to minimize reliance on the modality with higher perceived error. It confirms that the fusion adaptation mechanism reduces the weight of a consistently underperforming modality, even being able to "switch of" a compromised modality by setting α_t to 0 or 1 depending on which modality is less reliable.

Proposition 1 (Convergence Rate of Fusion Weight α_t) Under the conditions of Lemma 2, assume that modality 1 is consistently with low reliability (i.e., $e_t^{(1),\text{est}} > e_t^{(2),\text{est}} + \epsilon_\alpha$ for all $t > t_0$), the fusion weight is expected to be 0. Given that α_t is not yet converged to its boundary, then the expected decreasing rate of α_t is approximately linear:

$$\mathbb{E}\left[\alpha_{t+1} - \alpha_t\right] \approx -k_\alpha \left(\mathbb{E}\left[e_t^{(1),\text{est}}\right] - \mathbb{E}\left[e_t^{(2),\text{est}}\right]\right) \tag{9}$$

The steps to boundary convergence is therefore inversely proportional to k_α and the persistent magnitude of the estimated error difference, barring noise in error estimation.

Proposition 1 provides insight into the dynamics of the fusion adaptation. While Lemma 2 states the tendency and convergence, this suggests that the speed of this adaptation is tunable via k_α and directly influenced by how clearly the system can distinguish the performance of the modalities. A larger error difference allows for faster isolation of the faulty modality. This has a practical implication for setting k_α: a larger k_α leads to faster fusion adaptation but must be balanced against the risk of instability or overreaction to noisy error estimates (as per Remark 1). Building upon Lemma 2 and Proposition 1, Theorem 2 establishes the system's robust response in the case of severe drift affecting a single modality.

Theorem 2 (Modality Adaptation and Error Reduction under Severe Unilateral Drift) Consider the same scenario defined previously, where modality 1 is consistently less reliable. If the adaptive controller adjusts α_t such that Lemma 2 holds, then:

(i) The fusion weight α_t will converge towards 0, decreasing and eventually nullifying the influence of the compromised modality 1.

(ii) Consequently, the system's overall prediction error e_t will, after this reconfiguration transient, asymptotically approach the error level achievable by an optimal unimodal model utilizing only the reliable modality 2.

(iii) The Lyapunov stability of the system is preserved throughout this fusion reconfiguration process and thereafter, with the ultimate error bound being determined by the performance achievable with modality 2 and any residual drift or noise in its stream.

Theorem 2 indicates that the internal structure of the model is actively changing, making this model different from traditional multimodal systems, as traditional systems often suffer from a persistent drop in performance [42, 34]. Theorem 2 also shows that the adaptive fusion mechanism can correctly identify and isolate a "fault" modality by driving its weight (α_t) towards zero. This theorem quantifies the system's performance post-adaptation as it converges to the best possible performance using the remaining valid information sources. Additionally, system behavior with partial modality degradation and practical estimation of modality-specific errors are presented in Appendix C. The full proofs for our main results are in Appendix D.

5 Experiment

To empirically validate LS-OGD's theoretical contributions and its capacity to address key challenges in multimodal concept drift, we conducted experiments on the M3A [35] classification dataset. After initial training in a stable environment (Phase 1), the multimodal system was subjected to significant concept drift (Phase 2). Full experimental setup [1] and additional results are detailed in Appendix E.

First, Figure 2 demonstrates LS-OGD's ability to maintain stability. The controller's estimated drift signal, an empirical proxy for the bounded drift δ_t (Axiom 1), frequently activates during Phase 2. Correspondingly, while fluctuating with drift, the system's prediction error remains contained and is effectively driven down by the adaptive responses. This behavior empirically corroborates the UUB of the error, guaranteed by Theorem 1, and the error-reducing system dynamics characterized by Lemma 1, indicating stable learning in a non-stationary environment.

Second, Figure 3 reveals how LS-OGD addresses modality-specific challenges. Dynamic learning rate adjustments enhanced plasticity during instability, aligning with Lemma 1 and stable gain principles (Remark 1). More importantly, when confronted with severe unilateral modality drift,

[1] The code is available at https://github.com/Bellleuang1022/Lyapunov-Stable-Adaptive-Control-for-Multimodal-Concept-Drift.git.

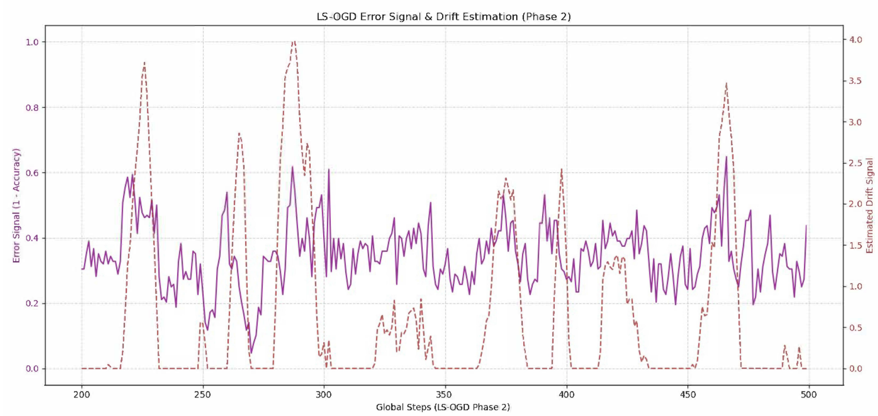

Figure 2: LS-OGD Error Signal & Drift Estimation in Phase 2

the fusion parameter α_t underwent a rapid and significant shift. This sharp re-weighting, isolating the compromised modality and preserving performance by relying on the healthier one, empirically confirms the adaptive fusion dynamics (Lemma 2) and the fault-tolerant capabilities (Theorem 2). This tackles the limitations of fixed fusion strategies and demonstrates robust, stable adaptation to modality-specific drift, a core challenge highlighted in the introduction.

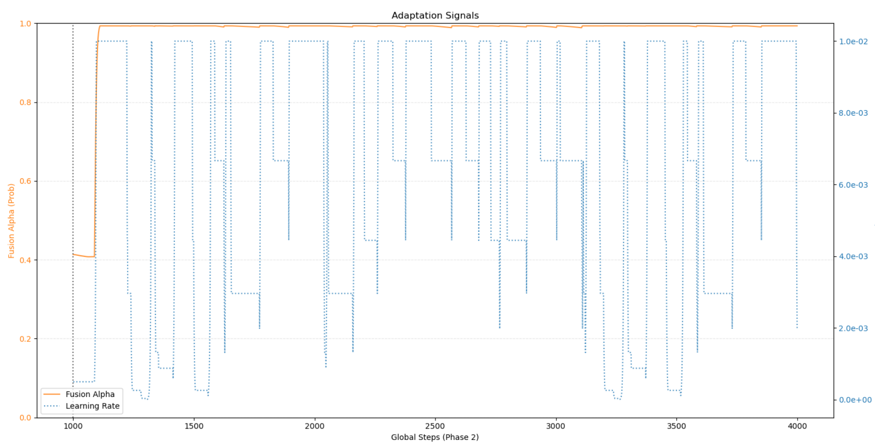

Figure 3: LS-OGD Controller Adaptation Signals in Phase 2

6 Conclusion

The LS-OGD framework represents a significant advancement in adaptive AI by combining Lyapunov-based control theory with online gradient learning and dynamic fusion weights. This integration allows for stability and resilience in the face of non-stationary, multimodal inputs. The framework's core assurance is that prediction errors will remain consistently bounded under limited drift and converge to zero as disturbances diminish. This ensures predictable performance, even when data distributions change. Moreover, LS-OGD features a modality-aware adaptation mechanism that automatically down-weights unreliable sensors or data streams, allowing the system to concentrate learning on trustworthy information sources. This lightweight, online approach is suited for real-world applications, such as autonomous vehicles that need to adjust the importance of inputs from vision, LiDAR, and radar under varying conditions; intelligence systems that integrate imagery, signals, and human reports while facing adversarial pressures; and misinformation detectors that combat evolving text and image manipulations. By establishing a robust framework for dynamic sensor re-weighting and bounded-error learning, LS-OGD lays a solid foundation for trustworthy AI. Its modular design also allows for extensions to nonlinear models, deep architectures, and context-aware adaptation strategies. Limitation and broader impact are discussed in Appendix G.

7 Acknowledgment

The authors sincerely thank the program and area chairs, along with the anonymous reviewers, for their constructive comments on this work. Additionally, Bell expresses her gratitude to her beloved pets, Kiwi (an Australian Shepherd) and Meemeow Pan (a Siamese Tabby), for their emotional support.

References

[1] Stephen H Bach and Marcus A Maloof. Paired learners for concept drift. In *2008 Eighth IEEE International Conference on Data Mining*, pages 23–32. IEEE, 2008.

[2] Manuel Baena-Garcıa, José del Campo-Ávila, Raul Fidalgo, Albert Bifet, Ricard Gavalda, and Rafael Morales-Bueno. Early drift detection method. In *Fourth international workshop on knowledge discovery from data streams*, volume 6, pages 77–86. Citeseer, 2006.

[3] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence*, 41(2):423–443, 2018.

[4] Albert Bifet and Ricard Gavalda. Learning from time-changing data with adaptive windowing. In *Proceedings of the 2007 SIAM international conference on data mining*, pages 443–448. SIAM, 2007.

[5] Giuseppe Canonaco, Alex Bergamasco, Alessio Mongelluzzo, and Manuel Roveri. Adaptive federated learning in presence of concept drift. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2021.

[6] Haoyu Chu, Shikui Wei, Ting Liu, Yao Zhao, and Yuto Miyatake. Lyapunov-stable deep equilibrium models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 11615–11623, 2024.

[7] Hongkai Dai, Benoit Landry, Lujie Yang, Marco Pavone, and Russ Tedrake. Lyapunov-stable neural-network control. *arXiv preprint arXiv:2109.14152*, 2021.

[8] Tamraparni Dasu, Shankar Krishnan, Suresh Venkatasubramanian, and Ke Yi. An information-theoretic approach to detecting changes in multi-dimensional data streams. In *Proc. Symposium on the Interface of Statistics, Computing Science, and Applications (Interface)*, 2006.

[9] Bogdan Doicin, Marian Popescu, and Cristian Patrascioiu. Pid controller optimal tuning. In *2016 8th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, pages 1–4. IEEE, 2016.

[10] Anthony Dontoh, Stephanie Ivey, Logan Sirbaugh, Andrews Danyo, and Armstrong Aboah. Visual dominance and emerging multimodal approaches in distracted driving detection: A review of machine learning techniques. *arXiv preprint arXiv:2505.01973*, 2025.

[11] Joao Gama and Gladys Castillo. Learning with local drift detection. In *International conference on advanced data mining and applications*, pages 42–55. Springer, 2006.

[12] Joao Gama, Pedro Medas, Gladys Castillo, and Pedro Rodrigues. Learning with drift detection. In *Advances in Artificial Intelligence–SBIA 2004: 17th Brazilian Symposium on Artificial Intelligence, Sao Luis, Maranhao, Brazil, September 29-Ocotber 1, 2004. Proceedings 17*, pages 286–295. Springer, 2004.

[13] Taesik Gong, Jongheon Jeong, Taewon Kim, Yewon Kim, Jinwoo Shin, and Sung-Ju Lee. Note: Robust continual test-time adaptation against temporal correlation. *Advances in Neural Information Processing Systems*, 35:27253–27266, 2022.

[14] Taesik Gong, Yewon Kim, Taeckyung Lee, Sorn Chottananurak, and Sung-Ju Lee. Sotta: Robust test-time adaptation on noisy data streams. *Advances in Neural Information Processing Systems*, 36:14070–14093, 2023.

[15] Zongbo Han, Fan Yang, Junzhou Huang, Changqing Zhang, and Jianhua Yao. Multimodal dynamics: Dynamical fusion for trustworthy multimodal classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20707–20717, 2022.

[16] Vijay Manikandan Janakiraman, XuanLong Nguyen, and Dennis Assanis. A lyapunov based stable online learning algorithm for nonlinear dynamical systems using extreme learning machines. In *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2013.

[17] Kendi Li, Di Chen, Zuguang Rao, Zijing Guan, Ya Jiang, and Yuanqing Li. A multimodal asynchronous human-machine interaction method based on electrooculography and speech recognition for wheelchair control. *IEEE Sensors Journal*, 2024.

[18] Anjin Liu, Yiliao Song, Guangquan Zhang, and Jie Lu. Regional concept drift detection and density synchronized drift adaptation. In *IJCAI international joint conference on artificial intelligence*, 2017.

[19] Anjin Liu, Guangquan Zhang, and Jie Lu. Fuzzy time windowing for gradual concept drift adaptation. In *2017 IEEE international conference on fuzzy systems (FUZZ-IEEE)*, pages 1–6. IEEE, 2017.

[20] Dong Liu, YouXi Wu, and He Jiang. Fp-elm: An online sequential learning algorithm for dealing with concept drift. *Neurocomputing*, 207:322–334, 2016.

[21] Moyang Liu, Kaiying Yan, Yukun Liu, Ruibo Fu, Zhengqi Wen, Xuefei Liu, and Chenxing Li. Exploring modality disruption in multimodal fake news detection. *arXiv preprint arXiv:2504.09154*, 2025.

[22] Shuang Ma, Sai Vemprala, Wenshan Wang, Jayesh K Gupta, Yale Song, Daniel McDufft, and Ashish Kapoor. Compass: Contrastive multimodal pretraining for autonomous systems. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1000–1007. IEEE, 2022.

[23] Kohei Miyaguchi and Hiroshi Kajino. Cogra: Concept-drift-aware stochastic gradient descent for time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4594–4601, 2019.

[24] Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. *arXiv preprint arXiv:2302.12400*, 2023.

[25] Stefanos-Iordanis Papadopoulos, Christos Koutlis, Symeon Papadopoulos, and Panagiotis C Petrantonakis. Red-dot: Multimodal fact-checking via relevant evidence detection. *IEEE Transactions on Computational Social Systems*, 2025.

[26] Omkar Sudhir Patil, Duc M Le, Emily J Griffis, and Warren E Dixon. Lyapunov-based deep residual neural network (resnet) adaptive control. *arXiv preprint arXiv:2404.07385*, 2024.

[27] Shivangi Singhal, Rajiv Ratn Shah, Tanmoy Chakraborty, Ponnurangam Kumaraguru, and Shin'ichi Satoh. Spotfake: A multi-modal framework for fake news detection. In *2019 IEEE fifth international conference on multimedia big data (BigMM)*, pages 39–47. IEEE, 2019.

[28] Junha Song, Jungsoo Lee, In So Kweon, and Sungha Choi. Ecotta: Memory-efficient continual test-time adaptation via self-distilled regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11920–11929, 2023.

[29] Runhan Sun, Max L Greene, Duc M Le, Zachary I Bell, Girish Chowdhary, and Warren E Dixon. Lyapunov-based real-time and iterative adjustment of deep neural networks. *IEEE Control Systems Letters*, 6:193–198, 2021.

[30] Yu Tong, Weihai Lu, Zhe Zhao, Song Lai, and Tong Shi. Mmdfnd: Multi-modal multi-domain fake news detection. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 1178–1186, 2024.

[31] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

[32] Qin Wang, Olga Fink, Luc Van Gool, and Dengxin Dai. Continual test-time domain adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7201–7211, 2022.

[33] Yaqing Wang, Fenglong Ma, Zhiwei Jin, Ye Yuan, Guangxu Xun, Kishlay Jha, Lu Su, and Jing Gao. Eann: Event adversarial neural networks for multi-modal fake news detection. In *Proceedings of the 24th acm sigkdd international conference on knowledge discovery & data mining*, pages 849–857, 2018.

[34] Renjie Wu, Hu Wang, Hsiang-Ting Chen, and Gustavo Carneiro. Deep multimodal learning with missing modality: A survey. *arXiv preprint arXiv:2409.07825*, 2024.

[35] Qingzheng Xu, Huiqiang Chen, Heming Du, Hu Zhang, Szymon Łukasik, Tianqing Zhu, and Xin Yu. M3a: A multimodal misinformation dataset for media authenticity analysis. *Computer Vision and Image Understanding*, 249:104205, 2024.

[36] Shuliang Xu and Junhong Wang. Dynamic extreme learning machine for data stream classification. *Neurocomputing*, 238:433–449, 2017.

[37] Zihui Xue and Radu Marculescu. Dynamic multimodal fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2575–2584, 2023.

[38] Xiaoyu Yang, Jie Lu, and En Yu. Adapting multi-modal large language model to concept drift from pre-training onwards. *arXiv preprint arXiv:2405.13459*, 2024.

[39] Chuanming Yu, Yinxue Ma, Lu An, and Gang Li. Bcmf: A bidirectional cross-modal fusion model for fake news detection. *Information Processing & Management*, 59(5):103063, 2022.

[40] Dan Zhang and Bin Wei. A review on model reference adaptive control of robotic manipulators. *Annual Reviews in Control*, 43:188–198, 2017.

[41] Wenyu Zhang. Pola: Online time series prediction by adaptive learning rates. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3375–3379. IEEE, 2021.

[42] Tianyi Zhao, Liangliang Zhang, Yao Ma, and Lu Cheng. A survey on safe multi-modal learning systems. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6655–6665, 2024.

[43] Fangyu Zou, Li Shen, Zequn Jie, Weizhong Zhang, and Wei Liu. A sufficient condition for convergences of adam and rmsprop. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 11127–11135, 2019.

Appendices

A More Related Work

A.1 Concept Drift Detection and Adaptation

In machine learning, a "no drift" scenario signifies a stable environment where the joint probability distribution of input features and the target variable, $P(X, Y)$, remains constant over time. Conversely, concept drift occurs when this joint distribution changes, $P_t(X, Y) \neq P_{t'}(X, Y)$ for different time points t and t', and can manifest in sudden, gradual, or incremental patterns. This broader concept can be broken down into two types: real concept drift, which directly impacts predictive modeling by altering the conditional distribution of the target given the input, $P_t(X|Y)$, meaning the fundamental relationship the model learns ($h : X \to Y$) changes; and virtual drift, where only the input data distribution $P_t(X)$ shifts while the conditional relationship $P_t(Y|X)$ remains unchanged. Figure 4 presents the different types of concept drifts.

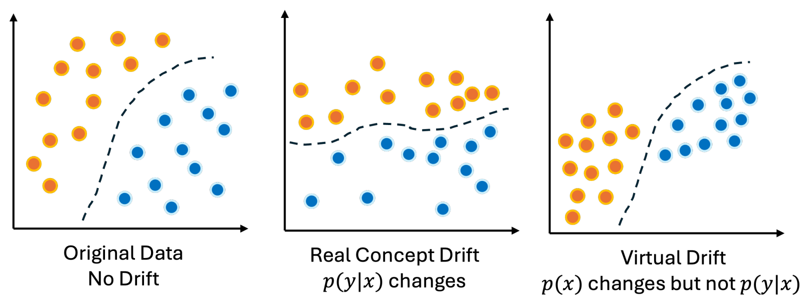

Figure 4: Types of concept drifts: The circles in this figure represent instances, and the different colors represent different classes.

A.2 Adaptive Control Strategies and Learning Rate Adjustment

Adaptive control provides a framework for real-time adjustment of system parameters to maintain stability and performance. In machine learning, the learning rate is a critical parameter often used with fixed or scheduled values, which can be problematic in concept drift. A low learning rate can hinder the model's ability to learn new concepts, while a high rate may lead to overfitting or instability. Research [41, 43] indicates that dynamically adapting the learning rate can enhance a model's ability to respond to distribution shifts. For instance, RMSProp and Adam adjust step sizes based on gradient statistics but are not specifically designed for concept drift. Some approaches, like COGRA [23], dynamically tune the learning rate during time-series forecasting by increasing it when prediction error variance rises and decreasing it once performance stabilizes. Similarly, [5] adapts learning rates in federated learning to handle drifting data better. These methods demonstrate the benefits of a higher learning rate post-drift for quicker adaptation. Inspired by these principles, our controller increases the learning rate upon detecting drift to adapt to new data swiftly, then gradually reduces it as the error decreases. This mirrors gain scheduling in control, where one variable (error rate) influences another (learning rate) to optimize performance.

In addition to learning rates, the field of adaptive control presents concepts such as model reference adaptive control (MRAC) [40] and Lyapunov-based adaptation laws [29], which ensure stability for dynamic systems with unknown parameters. In recent research that combines control theory and deep learning, [26] developed Lyapunov stable weight update rules for deep neural network controllers. Although their research focuses on a robotic control loop, the key insight is that parameter update equations can be designed to reduce a Lyapunov function gradually. Our paper treat the learning process itself as a system that needs to be controlled. Additionally, there's a relevant concept in which some researchers have utilized PID control for optimizer tuning [9]; they treat the training error like a process variable and employ PID controllers to adjust hyperparameters or even the magnitudes of gradients. Our method can be viewed as a specialized version of this approach: when the error increases, the rate of change of the error prompts a proportional rise in the learning rate, similar to a P/PI controller aiming to minimize the error.

A.3 Lyapunov Stability in Learning Systems

A Lyapunov function decreases over time concerning the system's state, guaranteeing that it will not diverge from its equilibrium position. In control theory, this provides strong assurances about both convergence and robustness. However, applying Lyapunov stability analysis to machine learning systems is relatively new [16]. Traditional training methods often lack formal stability guarantees, and neural network training can face issues such as exploding gradients or oscillations when hyperparameters are not chosen carefully. Recent efforts have begun to integrate stability theory into learning algorithms. For example, [16] proposed a Lyapunov-based online learning algorithm for nonlinear systems, ensuring the model's prediction error remains bounded. Researchers have incorporated Lyapunov-based constraints in deep reinforcement learning to maintain stability in an agent's value function or dynamics during the learning process [6].

In adaptive neural controllers, where a neural network controls a physical system, attempts have been made to enforce Lyapunov stability of the closed-loop system. [7] developed a neural network policy alongside a neural Lyapunov function that certifies stability within a certain region. This was a verification problem where a mixed-integer program checks the Lyapunov conditions for the neural policy. Such research confirms that it is feasible to train or constrain neural networks while considering Lyapunov criteria. However, these studies mainly focus on controlling external processes, whereas we emphasize the stability of the learning algorithm and its fusion mechanism.

The key innovation in our work is the application of Lyapunov stability analysis to adapt to concept drift in multimodal learning. We construct a Lyapunov function based on the dynamics of the model's error. Intuitively, the error signal parallels the state of a dynamical system. Our adaptive update rules for the learning rate and fusion weight are designed to ensure this error diminishes over time, even in the face of external disturbances caused by drift. This approach is reminiscent of MRAC, where adaptive updates drive the difference between a system and a reference model to zero [26]. In this context, the "reference" is the new concept (or zero error), and we adapt parameters to track it. To the best of our knowledge, no prior work in multimodal learning has incorporated a Lyapunov stability perspective to manage streaming non-stationary data.

B Proposed Algorithm

B.1 LS-OGD Main Adaptation Loop

The algorithm of the main adaptation loop for multimodal learning is

Algorithm 1 LS-OGD Main Adaptation Loop for Multimodal Learning

Input:
 Data stream for Phase 2: $\mathcal{D}_{drift} = \{(X_{text}^{(i)}, X_{img}^{(i)}, Y_{true}^{(i)})\}_{i=1}^{N_{P2}}$
 Model M_θ (Text Encoder M_T, Image Encoder M_I, Fusion M_F) initialized from Phase 1
 Initial learning rate $\eta_{current}$
 Initial raw fusion parameter $\hat{\alpha}_{current}$ (for M_F)
 Optimizer Opt (e.g., AdamW) initialized with parameters of M_θ
 Accuracy history deque H_{acc} (max length L_{hist}), potentially primed
 Drift detection parameters: $W_{drift}, R_{comp}, \tau_{drop}$ (passed to Algorithm 2)
 LR adaptation parameters: $k_{lr}, [\eta_{min}, \eta_{max}], \theta_{high\_drift}, \theta_{mod\_drift}$ (passed to Algorithm 2)
 Alpha adaptation parameters: k_α, drift type indicators (passed to Algorithm 2)
 Loss function $\mathcal{L}_{BCE}$ (Binary Cross-Entropy)
Initialize: Apply $\eta_{current}$ to Opt.
for each batch $(X_{text}, X_{img}, Y_{true})$ from $\mathcal{D}_{drift}$ **do**
 // Perform forward pass
 $p_{text} \leftarrow M_T(X_{text})$ ▷ Text modality probability
 $p_{img} \leftarrow M_I(X_{img})$ ▷ Image modality probability
 $\alpha_{display} \leftarrow \sigma(\hat{\alpha}_{current})$ ▷ Current sigmoid-squashed fusion weight
 $p_{fused} \leftarrow (1 - \alpha_{display}) \cdot p_{text} + \alpha_{display} \cdot p_{img}$ ▷ Fused probability
 // Evaluate performance
 $L_t \leftarrow$ mean$(\mathcal{L}_{BCE}(p_{fused}, Y_{true}))$
 $Acc_t \leftarrow$ ComputeAccuracy(p_{fused}, Y_{true})
 Add Acc_t to H_{acc}
 // Obtain controller actions and new parameters from LS-OGD Controller Logic (Algorithm 2)
 $(S_{drift}, \eta_{new}, \hat{\alpha}_{new}) \leftarrow$ LSOGDControllerLogic$(H_{acc}, \eta_{current}, \hat{\alpha}_{current},$ params$)$ ▷ params include all detection and adaptation hyperparameters
 // Apply adaptations and optimize
 Update learning rate in Opt to η_{new}
 Opt.zero_grad()
 L_t.backward()
 Opt.step() ▷ Updates M_θ using η_{new}
 // Update state for next iteration
 $\eta_{current} \leftarrow \eta_{new}$
 $\hat{\alpha}_{current} \leftarrow \hat{\alpha}_{new}$ ▷ $\hat{\alpha}$ in M_F is now updated to $\hat{\alpha}_{new}$
 LogMetrics$(L_t, Acc_t, \eta_{current}, \sigma(\hat{\alpha}_{current}), S_{drift},$ etc.$)$
end for

For each batch, it performs a forward pass using the current fusion weight, evaluates performance, calls Algorithm 2 to get controller actions (drift signal, new learning rate, new fusion parameter), applies these adaptations, performs backpropagation and optimization, and updates the state for the next iteration. Algorithm 2 details the controller's decision-making. It takes the accuracy history and current parameters as input. It consists of three main procedures: calculating a drift signal based on the drop in accuracy between recent and past windows, adjusting the learning rate based on the drift signal strength, with clamping to predefined min/max values, and modifying the raw fusion parameter $\hat{\alpha}$ based on the drift signal and indicators of which modality might be drifting.

B.2 LS-OGD Controller Logic

The proposed LS-OGD controller is

Algorithm 2 LS-OGD Controller Logic

Input for Controller Logic function **LSOGDCONTROLLERLOGIC:**
  Accuracy history deque H_{acc}
  Current learning rate η_{in}
  Current raw fusion parameter $\hat{\alpha}_{in}$
  Drift detection parameters: $W_{drift}, R_{comp}, \tau_{drop}$
  LR adaptation parameters: $k_{lr}, [\eta_{min}, \eta_{max}], \theta_{high\_drift}, \theta_{mod\_drift}$
  Alpha adaptation parameters: k_α, drift type indicators

// Detect concept drift
$S_{drift} \leftarrow$ EstimateDriftSignal$(H_{acc}, W_{drift}, R_{comp}, \tau_{drop})$
// Determine new learning rate
$\eta_{out} \leftarrow$ AdaptLearningRate$(\eta_{in}, S_{drift}, k_{lr}, [\eta_{min}, \eta_{max}], \theta_{high_drift}, \theta_{mod_drift})$
// Determine new raw fusion parameter
$\hat{\alpha}_{out} \leftarrow$ AdaptFusionParameter$(\hat{\alpha}_{in}, S_{drift}, k_\alpha,$ drift type indicators)

return $(S_{drift}, \eta_{out}, \hat{\alpha}_{out})$

procedure ESTIMATEDRIFTSIGNAL$(H_{acc}, W_{drift}, R_{comp}, \tau_{drop})$
  if $|H_{acc}| < W_{drift}$ **then return** 0
  end if
  $Acc_{recent} \leftarrow$ mean accuracy over last $\lfloor W_{drift} \cdot R_{comp} \rfloor$ entries of H_{acc}
  $Acc_{past} \leftarrow$ mean accuracy over $W_{drift} - \lfloor W_{drift} \cdot R_{comp} \rfloor$ entries of H_{acc} before recent window
  if $Acc_{past} - Acc_{recent} > \tau_{drop}$ **then**
    return $(Acc_{past} - Acc_{recent})/Acc_{past}$ ▷ Normalized drop as drift signal
  else
    return 0
  end if
end procedure

procedure ADAPTLEARNINGRATE$(\eta_{prev}, S_{drift}, k_{lr}, [\eta_{min}, \eta_{max}], \theta_{high}, \theta_{mod})$
  if $S_{drift} > \theta_{high}$ **then**
    $\eta_{cand} \leftarrow \eta_{prev} \cdot k_{lr}$
  else if $S_{drift} > \theta_{mod}$ **then**
    $\eta_{cand} \leftarrow \eta_{prev}/k_{lr}$
  else
    $\eta_{cand} \leftarrow \eta_{prev}$
  end if
  return $\max(\eta_{min}, \min(\eta_{cand}, \eta_{max}))$
end procedure

procedure ADAPTFUSIONPARAMETER$(\hat{\alpha}_{prev}, S_{drift}, k_\alpha,$ drift indicators)
  $\Delta\hat{\alpha} \leftarrow 0$
  if $S_{drift} > \theta_{mod\_drift}$ **and** (image modality suspected based on indicators) **then**
    $\Delta\hat{\alpha} \leftarrow -k_\alpha$ ▷ Adjust to reduce reliance on suspected drifting image modality
  else if $S_{drift} > \theta_{mod\_drift}$ **and** (text modality suspected based on indicators) **then**
    $\Delta\hat{\alpha} \leftarrow +k_\alpha$ ▷ Adjust to reduce reliance on suspected drifting text modality
  end if
  return $\hat{\alpha}_{prev} + \Delta\hat{\alpha}$ ▷ Apply clamping if $\hat{\alpha}$ has bounds
end procedure

C Additional Results

C.1 Example 1. Scalar Error Dynamics

Suppose at time t, the error is e_t. An adaptive learning step aims to reduce this error. Let the change due to learning be $-\eta_t \lambda e_t$ (where $\lambda > 0$ reflects learning efficacy and η_t is the learning rate). Let concept drift introduce an additive error δ_t. So, $e_{t+1} = e_t - \eta_t \lambda e_t + \delta_t = (1- \eta_t\lambda)\, e_t + \delta_t$.

Consider $V(t) = \frac{1}{2} e_t^2$. Then,

$$\Delta V(t) = \frac{1}{2}\left(e_{t+1}^2 - e_t^2\right) = \frac{1}{2}\left(\left((1-\eta_t\lambda)\, e_t + \delta_t\right)^2 - e_t^2\right) \tag{10}$$

$$= \frac{1}{2}\left((1-\eta_t\lambda)^2 e_t^2 + 2\,(1-\eta_t\lambda)\, e_t\delta_t + \delta_t^2 - e_t^2\right) \tag{11}$$

$$= \frac{1}{2}\left(-2\eta_t\lambda + \eta_t^2\lambda^2\right) e_t^2 + (1-\eta_t\lambda)\, e_t\delta_t + \frac{1}{2}\delta_t^2 \tag{12}$$

If $\eta_t\lambda$ is small (e.g., < 1), and we approximate $1 - \eta_t\lambda \approx 1$:

$$\Delta V(t) \approx -\eta_t\lambda e_t^2 + e_t\delta_t + \frac{1}{2}\delta_t^2. \tag{13}$$

Our controller (Algorithm 2) adapts η_t. Suppose if $|e_t|$ is persistently large, η_t is increased. If $|\delta_t| \leq \delta_{\max}$ (Axiom A1). For $\Delta V(t)$ to be negative when $|e_t|$ is large, we need $\eta_t\lambda e_t^2$ to dominate $\left|e_t\delta_t + \frac{1}{2}\delta_t^2\right|$. This can be ensured if $|e_t| > \frac{\delta_{\max}}{\eta_k\lambda} + \frac{\delta_{\max}^2}{2\eta_k\lambda|e_t|}$.

This implies that e_t will be driven towards a bound related to $\delta_{\max}/\left(\eta_t\lambda\right)$. If $\delta_t \to 0$, then $\Delta V(t) \approx -\eta_t\lambda e_t^2 < 0$ for $e_t \neq 0$, leading to $e_t \to 0$. This simplified example illustrates the core mechanism behind UUB and asymptotic convergence. The adaptive controller's role in adjusting η_t is crucial for maintaining the dominance of the negative term when e_t is large.

C.2 Corollary 2. System Behavior with Partial Modality Degradation

If, under the conditions of Theorem 2, modality 1 does not become entirely uninformative but rather its signal-to-noise ratio significantly degrades (making $e_t^{(1),\text{est}}$ consistently higher than $e_t^{(2),\text{ est}}$ but not necessarily reflective of pure noise), the fusion weight α_t will still decrease, reducing the influence of the degraded modality 1. The system's error e_t will then converge to a UUB state primarily determined by the more reliable modality 2, potentially augmented by any residual (but down-weighted) useful information from modality 1, or slightly inflated by its downweighted noise.

Theorem 2 discusses the case of a modality becoming essentially uninformative. This corollary extends the intuition to a “soft" failure. It highlights that adaptive fusion is not just a switch; it can find an intermediate balance if a modality is partially compromised. It still attempts to optimize the fusion based on relative estimated reliabilities. The system reduces reliance rather than completely discarding a modality unless its error contribution becomes overwhelmingly negative.

C.3 Remark 2. Practical Estimation of Modality-Specific Errors

The efficacy of Theorem 2 and its supporting Lemma 2 hinges on the ability to obtain reliable estimates of modality-specific errors $\left(e_t^{(m),\text{est}}\right)$. As outlined in Section 3 and Algorithm 2, this often involves techniques like evaluating hypothetical unimodal predictions. The accuracy of these estimates can be affected by:

(i) Shared representations: If early parts of the model are shared before modality-specific processing, disentangling error contributions can be indirect.

(ii) Complexity of drift: Observing the relative degradation might be challenging if drift affects both modalities correlatedly.

(iii) Noise in labels or inputs: This can affect the stability of individual error estimates.

C.4 Remark 3. Continuous-Time Analysis vs. Discrete-Time Implementation

For conceptual clarity and leveraging established mathematical tools, our Lyapunov analysis often refers to continuous-time derivatives or small discrete differences $\Delta V(t)$. The proposed LS-OGD system (Algorithms 1 and 2) operates in discrete time steps (batches). The theoretical stability results (UUB) derived from a continuous-time perspective generally translate to discrete-time systems provided that the discrete time steps (effectively related to how frequently parameters η_t, α_t, and model weights θ are updated) are sufficiently small. If updates are too large or infrequent relative to the system's or drift's dynamics, discrete-time effects could lead to behaviors not captured by the continuous approximation, including instability even if the continuous analog is stable. This highlights the importance of the learning rate and adaptation gain selection in the practical discrete-time implementation.

C.5 Example 2. Fusion Weight Dynamics

Assume α_t is the weight for modality 1 (text), so the fused output is $z_t = \alpha_t z_t^{(1)} + (1 - \alpha_t)\, z_t^{(2)}$. Suppose at t_0, modality 1 starts providing misleading information, leading to a consistently higher estimated error $e_t^{(1),\text{ est}}$ compared to modality 2's error $e_t^{(2),\text{ est}}$. Let $e_t^{(1),\text{ est}} = 0.8$ and $e_t^{(2),\text{ est}} = 0.2$ for $t > t_0$. The adaptation logic for α_t is designed to shift weight away from the modality with higher error. An effective update for α_t might be $\alpha_{t+1} = \text{clip}_{[0,1]}\,(\alpha_t + k_\alpha \cdot$ (signal favoring modality 2 over 1)).

If the signal is proportional to $\left(e_t^{(2),\text{ est}} - e_t^{(1),\text{ est}}\right)$, then $\Delta\alpha_t = k_\alpha(0.2 - 0.8) = -0.6k_\alpha$. If $k_\alpha = 0.05$, then $\Delta\alpha_t = -0.03$. Starting with $\alpha_{t_0} = 0.5$ (equal weighting):

$$\alpha_{t_0+1} = 0.5 - 0.03 = 0.47 \tag{14}$$

$$\alpha_{t_0+2} = 0.47 - 0.03 = 0.44 \tag{15}$$

This trend will continue, reducing α_t until it reaches 0 (due to clipping). At this point, the system relies entirely on modality 2 $\left(z_t \approx z_t^{(2)}\right)$, having effectively isolated the "faulty" modality 1. This illustrates how Lemma 2's assertion of α_t convergence leads to the outcome in Theorem 2. The implementation in Algorithm 2 uses $\hat{\alpha}$ and a sigmoid, but the principle of shifting away from the higher-error modality is the same.

D Proofs

D.1 Proof for Lemma 1

By the joint learning-control update (Eq. (5) in the main paper), the prediction error evolves as

$$e_{t+1} = e_t - \eta_t g(e_t, \alpha_t) + \delta_t + r_t + \rho_t, \tag{16}$$

where $g(e_t, \alpha_t)$ is the effective gradient-like term from the controller and learner, $\delta_t \in \mathbb{R}^n$ is the bounded drift disturbance (Axiom 1: $||\delta_t|| \leq \delta_{max}$), r_t collects higher-order Taylor-remainder terms in η_t (hence, $|r_t| = \mathcal{O}(\eta_t^2)$), and ρ_t collects the effect of the fusion-weight update (hence, $|\rho_t| = \mathcal{O}(||\Delta\alpha_t||)$).

We defined the Lyapunov function in Eq. (6) and get

$$\Delta V(t) = \frac{1}{2}(e_{t+1}^2 - e_t^2) = \frac{1}{2}\Big[(e_t - \eta_t g + \delta_t + \rho_t)^2 - e_t^2\Big]. \tag{17}$$

Expand e_{t+1}^2:

$$e_{t+1}^2 = e_t^2 - 2\eta_t e_t g(e_t, \alpha_t) + 2e_t\delta_t + 2e_t(r_t + \rho_t) \tag{18}$$

$$+ (\eta_t g(e_t, \alpha_t))^2 + ||\delta_t||^2 + (r_t + \rho_t)^2 - 2\eta_t g(e_t, \alpha_t)^T(\delta_t + r_t + \rho_t). \tag{19}$$

Subtracting e_t^2 and dividing by 2 gives

$$\Delta V(t) = -\eta_t e_t g\left(e_t, \alpha_t\right) + e_t\delta_t + e_t\left(r_t + \rho_t\right) + \frac{1}{2}\eta_t^2 \left\|g\left(e_t, \alpha_t\right)\right\|^2 \tag{20}$$

$$+ \frac{1}{2}\left\|\delta_t\right\|^2 + \frac{1}{2}\left(r_t + \rho_t\right)^2 - \eta_t g\left(e_t, \alpha_t\right)^\top\left(\delta_t + r_t + \rho_t\right). \tag{21}$$

By Axiom 2, there is $l_1 > 0$ so that

$$e_t g(e_t, \alpha_t) \geq l_1 e_t^2, \tag{22}$$

and $||g(e_t, \alpha_t)|| \leq l_2|e_t|$ for some $l_2 > 0$. The remainder $r_t = \mathcal{O}(\eta_t^2)$ and fusion-weight effect $\rho_t = \mathcal{O}(||\Delta\alpha_t||)$, so their contributions can be gathered into

$$\mathcal{G}(e_t, \alpha_t) = ||g(e_t, \alpha_t)||^2 + \frac{(r_t + \rho_t)^2}{\eta_t^2} + \frac{g(e_t, \alpha_t)^T(r_t + \rho_t)}{\eta_t}, \tag{23}$$

which is bounded by a polynomial in $|e_t|$, $||\Delta\alpha_t||$, and η_t. By choosing k_{eta}, k_α small enough, all higher-order terms and cross-terms involving δ_t, r_t, ρ_t can be upper-bounded by constants times $\eta_t^2\mathcal{G}(e_t, \alpha_t)$ and $||\Delta\alpha_t||$.

Group the dominant negative term and the rest into the form

$$\Delta V(t) \leq -\underbrace{\ell_1}_{\gamma_1}\eta_t e_t^2 + \underbrace{1}_{\gamma_2}\eta_t\left|e_t\right|\left\|\delta_t\right\| + \underbrace{\frac{1}{2}\left(1+\ell_2^2\right)}_{\gamma_3}\eta_t^2\left(\mathcal{G}\left(e_t, \alpha_t\right) + \left\|\delta_t\right\|^2\right) + \mathcal{O}\left(\left\|\Delta\alpha_t\right\|\right), \tag{24}$$

Hence, the stated bound holds with $\gamma_1 = l_1$, $\gamma_2 = 1$, $\gamma_3 = \frac{1}{2}(1 + l_2^2)$, and with the $\mathcal{O}(||\Delta\alpha_t||)$ term capturing the fusion-weight adaptation effect. This completes the proof of Lemma 1.

D.2 Proof for Lemma 2

By design (Eq. (8) in the paper), the fusion weight is adjusted each step according to the signed error gap:

$$\alpha_{t+1} = \mathrm{clip}_{[0,1]}\left(\alpha_t - k_\alpha\left(\hat{e}_t^{(1)} - \hat{e}_t^{(2)}\right)\right) = \mathrm{clip}_{[0,1]}\left(\alpha_t - k_\alpha\Delta e_t\right), \tag{25}$$

where $\mathrm{clip}_{[0,1]}(\cdot)$ denotes clipping into [0,1]. Since by hypothesis $\Delta e_t > 0$ and $k_\alpha\Delta e_t < 1$, we have

$$\alpha_t - k_\alpha\Delta e_t < \alpha_t, \tag{26}$$

and furthermore $\alpha_t - k_\alpha \Delta e_t \geq 0$ as long as $\alpha_t \geq k_\alpha \Delta e_t$. Because k_α is chosen small enough, for any interior $\alpha_t \in (0, 1]$ we remain in [0,1] after subtracting $k_\alpha \Delta e_t$. Thus, the projection does nothing and

$$\alpha_{t+1} = \alpha_t - k_\alpha \Delta e_t < \alpha_t. \tag{27}$$

Hence, whenever $\alpha_t > 0$, the update yields $\alpha_{t+1} < \alpha_t$. By induction, the sequence $\{\alpha_t\}$ is strictly decreasing until it first hits a point $\alpha_T \leq k_\alpha \Delta e_T$. At that step, the unclipped value $\alpha_T - k_\alpha \Delta e_T$ would be negative, so the projection forces $\alpha_{T+1} = 0$. Thereafter, $\alpha_t \equiv 0$ for all $t \geq T + 1$.

Under a persistent positive error gap $\Delta e_t > 0$, the fusion weight moves monotonically downward at rate $k_\alpha \Delta e_t$ and converges in finitely many steps to the boundary $\alpha = 0$, fully down-weighting the degraded modality. This completes the proof of Lemma 2.

D.3 Proof for Proposition 1

In the interior $\alpha_t \in (0, 1)$ and under the persistent gap assumption $\Delta e_t > 0$ with $k_\alpha \Delta e_t < 1$, the projection $\text{clip}_{[0,1]}$ is inactive. Hence, the update rule

$$\alpha_{t+1} = \alpha_t - k_\alpha \Delta e_t \tag{28}$$

holds exactly. Taking expectations conditioned on the history up to time t gives

$$\mathbb{E}\left[\alpha_{t+1} \mid \mathcal{F}_t\right] = \alpha_t - k_\alpha \mathbb{E}\left[\Delta e_t \mid \mathcal{F}_t\right]. \tag{29}$$

Since α_t is $\mathcal{F}_t$-measurable,

$$\mathbb{E}\left[\alpha_{t+1} - \alpha_t \mid \mathcal{F}_t\right] = -k_\alpha \mathbb{E}\left[\Delta e_t \mid \mathcal{F}_t\right]. \tag{30}$$

Removing the conditioning yields

$$\mathbb{E}\left[\alpha_{t+1} - \alpha_t\right] = -k_\alpha \mathbb{E}\left[\Delta e_t\right]. \tag{31}$$

Thus, on average, the fusion weight decreases by a constant amount $k_\alpha \mathbb{E}\left[\Delta e_t\right]$ each step. Denote this mean decrement by

$$d = k_\alpha \mathbb{E}\left[\Delta e_t\right] > 0. \tag{32}$$

Then $\mathbb{E}\left[\alpha_{t+1}\right] = \mathbb{E}\left[\alpha_t\right] - d$, so by unrolling the expectation,

$$\mathbb{E}\left[\alpha_t\right] = \alpha_0 - td. \tag{33}$$

To reach a small threshold $\varepsilon \in [0, \alpha_t)$, solve

$$\alpha_0 - Td \leq \varepsilon \tag{34}$$

$$T \geq \frac{\alpha_0 - \varepsilon}{d} = \frac{\alpha_0 - \varepsilon}{k_\alpha \mathbb{E}\left[\Delta e_t\right]}. \tag{35}$$

Hence, the expected number of steps to drive the fusion weight down to ε scales inversely with the product $k_\alpha \mathbb{E}[\Delta e_t]$. Because the fusion-weight update is a simple decrement by $k_\alpha \Delta e_t$ in expectation, the convergence toward the boundary is linear in expectation, with rate constant $k_\alpha \mathbb{E}[\Delta e_t]$. This completes the proof of Proposition 1.

D.4 Proof for Theorem 1

We defined the Lyapunov function in Eq. (6). By Lemma 1, there exist constants $\gamma_1, \gamma_2, \gamma_3 > 0$ and functions $\mathcal{G}(e_t, \alpha_t)$, $\Delta\alpha_t$ such that

$$\Delta V(t) = V(t+1) - V(t) \leq -\gamma_1 \eta_t e_t^2 + \gamma_2 \eta_t \left|e_t\right| \left\|\delta_t\right\| + \gamma_3 \eta_t^2 \left(\mathcal{G}\left(e_t, \alpha_t\right) + \left\|\delta_t\right\|^2\right) + \mathcal{O}\left(\left\|\Delta\alpha_t\right\|\right), \tag{36}$$

By Axiom 1,

$$||\delta_t|| \le \delta_{max}. \tag{37}$$

Also, by design, the controller ensures

$$0 < \eta_t \le \eta_{\max}, \quad \|\Delta\alpha_t\| \le \Delta\alpha_{\max}, \tag{38}$$

with both η_{max}, $\Delta\alpha_{max}$ arbitrarily small when k_η, k_α are small. Hence, each of the positive terms on the right-hand side of $\Delta V(t) = V(t+1) - V(t)$ can be bounded as

$$\gamma_2\eta_t\,|e_t|\,\|\delta_t\| \le \gamma_2\eta_{\max}\delta_{\max}\,|e_t|\,, \tag{39}$$

$$\gamma_3\eta_t^2\,\|\delta_t\|^2 \le \gamma_3\eta_{\max}^2\delta_{\max}^2 \tag{40}$$

$$\mathcal{O}\left(\|\Delta\alpha_t\|\right) \le c_\alpha\Delta\alpha_{\max}, \tag{41}$$

and similarly $\gamma_3\eta_t^2\mathcal{G}(e_t, \alpha_t)$ grows at most quadratically in η_{max} and $|e_t|$. Thus, there exists a constant $c_1 > 0$ such that

$$\Delta V(t) \le -\gamma_1\eta_t e_t^2 + c_1\left(\eta_{\max}\,|e_t| + \eta_{\max}^2 + \Delta\alpha_{\max}\right). \tag{42}$$

Choose $\varepsilon_B > 0$ and sufficiently small η_{max}, $\Delta\alpha_{max}$ so that

$$\gamma_1\eta_{\max}\varepsilon_B^2 > c_1\left(\eta_{\max}\varepsilon_B + \eta_{\max}^2 + \Delta\alpha_{\max}\right). \tag{43}$$

Then, whenever $|e_t| > \varepsilon_B$, the negative quadratic term dominates and

$$\Delta V(t) < 0, \tag{44}$$

so $V(t)$ and thus $|e_t|$ decreases until $|e_t| \le \varepsilon_B$, after which it can never exceed ε_B. This is the UUB property.

Furthermore, if additionally $\delta_t \to 0$, then for any $\epsilon > 0$, there exists T such that for all $t \ge T$, $||\delta_t|| \le \epsilon$. Likewise, $\mathcal{G}(e_t, \alpha_t)$ and $||\Delta\alpha_t||$ can be made arbitrarily small by design of the adaptation gains. Hence, for t large enough, the bounded Eq. (36) reduces to

$$\Delta V(t) \le -\frac{\gamma_1}{2}\eta_t e_t^2 < 0, \quad \text{whenever } e_t \ne 0. \tag{45}$$

Thus, $V(t)$ decreases strictly unless $e_t = 0$, forcing $e_t \to 0$ as $t \to \infty$. The above establishes the UUB of the error and asymptotic convergence to zero when the drift vanishes, completing the proof of Theorem 1.

D.5 Proof for Theorem 2

Theorem 2 is supposing that one modality (say modality 1) experiences a persistent, large drift so that for all $t \ge t_0$

$$\Delta e_t = \hat{e}_t^{(1)} - \hat{e}_t^{(2)} > \delta_{\text{gap}} > 0, \tag{46}$$

where δ_{gap} is a fixed positive constant. Under the fusion-weight update of Lemma 2 with step-size k_α chosen so that $k_\alpha\delta_{\text{gap}} < 1$, the following hold:

1. $\alpha_t \to 0$ in at most $T \le \left\lceil \frac{\alpha_{t_0}}{k_\alpha\delta_{\text{gap}}} \right\rceil$ steps (finite-time convergence to the boundary).
2. Thereafter, $t \ge t_0 + T$, the closed-loop error e_t evolves as a single-modality system using only modality 2 and thus satisfies the UUB and asymptotic convergence properties of Theorem 1 with “drift" replaced by the residual error of modality 2 alone.

Since $\Delta e_t \ge \delta_{gap} > 0$ for all $t \ge t_0$, Lemma 2 guarantees

$$\alpha_{t+1} = \alpha_t - k_\alpha\Delta e_t \le \alpha_t - k_\alpha\delta_{gap}, \tag{47}$$

until α first reaches the boundary at 0. By induction, after m steps

$$\alpha_{t_0+m} \le \alpha_{t_0} - mk_\alpha\delta_{\text{gap}}. \tag{48}$$

Hence, $\alpha_{t_0+m} \leq 0$ as soon as

$$m \geq \frac{\alpha_{t_0}}{k_\alpha \delta_{\text{gap}}}. \tag{49}$$

Because m must be an integer, the first time α hits zero is

$$T = \min \{m : \alpha_{t_0+m} \leq 0\} \leq \left\lceil \frac{\alpha_{t_0}}{k_\alpha \delta_{\text{gap}}} \right\rceil. \tag{50}$$

At that step, the projection in the update forces $\alpha_{t_0+m} = 0$, and thereafter $\alpha_t \equiv 0$.

For $t \geq t_0 + T$, with $\alpha_t = 0$, the fusion component relying on modality 1 is entirely disabled. The joint update (Eq. (1) in the paper) then reduces to the single-modality-2 learning-control system:

$$e_{t+1} = e_t - \eta_t g^{(2)}(e_t) + \delta_t^{(2)} + r_t^{(2)} + \rho_t^{(2)}, \tag{51}$$

where each term now refers only to modality 2 (its drift $\delta_t^{(2)}$, its gradient effect $g^{(2)}$, etc.), and the small remainder $\rho_t^{(2)}$ captures any residual fusion-weight logic (now inactive).

By the same Lyapunov argument used in Theorem 1, invoking Lemma 1 with the single-modality terms in place of the joint ones, we conclude that:

UUB: There is an $\varepsilon_B^{(2)}$ (determined by the bound on $||\delta_t^{(2)}||$) such that $|e_t|$ enters and remains within $|e_t| \leq \varepsilon_B^{(2)}$.

Asymptotic convergence: If $\delta_t^{(2)} \to 0$, then $e_t \to 0$ as $t \to \infty$.

Thus, once the faulty modality is fully down-weighted, the system behaves like a stable, single-modality learner and inherits all the stability and convergence guarantees of Theorem 1, but now concerning modality 2 alone. This completes the proof of Theorem 2.

E Additional Experiment Details

We conducted a series of experiments to empirically validate the theoretical contributions of LS-OGD and its capacity to address key challenges in multimodal concept drift. Our setup is designed to simulate real-world scenarios where data distributions change over time, impacting model performance. We compare LS-OGD against a static baseline to demonstrate its adaptive capabilities. The experiments are built upon the M3A misinformation dataset, chosen for its relevance to dynamic, real-world information streams.

E.1 Dataset and Preprocessing

We utilize the M3A dataset, which contains multimodal samples (text and images) relevant to misinformation detection. Table 1 and Figure 5 present the descriptive statistics of the dataset used in this study.

Table 1: Dataset statistics for text and image modalities (the length of each text sample and the size of each image sample are calculated at the token-level and pixel-level, respectively)

	Text			Images		
	# Samples	Max Length	Avg. Length	Percentage	Max Size	Avg. Size
Real	708,425	77	60.21	67.56%	2,523,960	1,005,724.74
Fake	340,150	77	38.34	32.44%	2,523,960	1,019,480.79

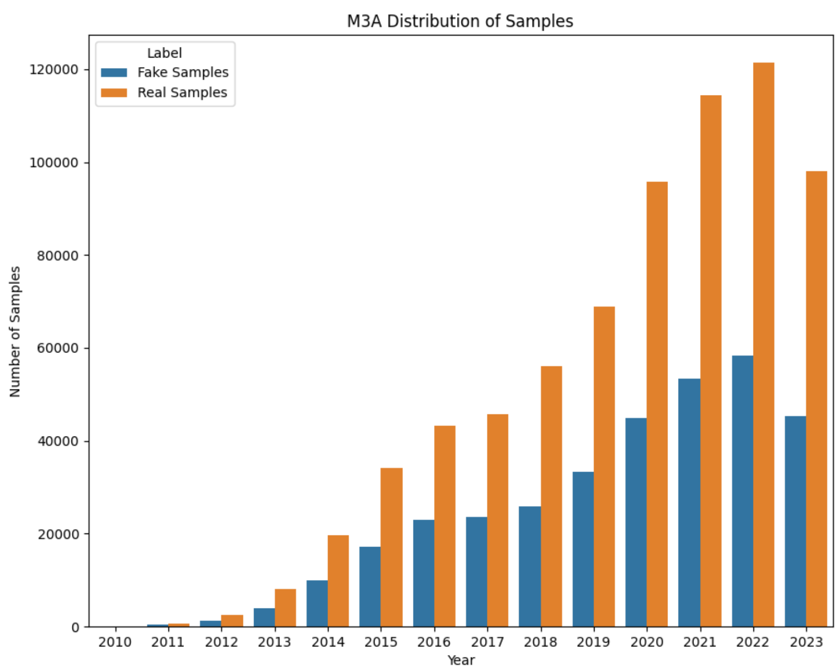

Figure 5: The distribution of samples in the M3A dataset we used in experiments (per year per label)

Following the methodology in this paper, we structure the experiment into two main phases:

Phase 1 (Stable Environment): Data samples are chronologically ordered based on the year column. Samples from years up to and including 2014 are used for initial model training. This phase represents a period of relatively stable data distribution.

Phase 2 (Drifting Environment): Samples from years after 2014 are used to simulate the operational deployment phase where concept drift is introduced.

E.2 Experimental Setup

E.2.1 Multimodal Model Architecture

Our multimodal classification model consists of three main components: text encoder, image encoder, and fusion module. We use a pre-trained CLIP-ViT model as the text encoder. The model processes input text to extract semantic features, and a linear head projects these features to a single logit for binary classification. The same ViT model is also employed as the image encoder. Similarly, it processes images to extract visual features, followed by a linear head for logit generation. Regarding the fusion module, the logits from the text and image encoders are combined using a weighted average fusion mechanism, as described in Section 3 (Eq. (1) in the paper). This module has a learnable parameter, α_t, that balances the contribution of each modality: $z_t = (1 - \alpha_t) \cdot z_t^{(\text{text})} + \alpha_t \cdot z_t^{(\text{image})}$.

E.2.2 Concept Drift Simulation

In Phase 2, we introduce concept drift into the data stream to test the adaptability of LS-OGD. Based on the configuration, we simulate both image and text modality drifts. For the image modality drift, visual data is degraded by JPEG compression with quality set to 50, addition of Gaussian noise with a standard deviation of 0.05, and application of Gaussian blur with a radius of 3.0. Regarding the text modality drift, textual data undergoes semantic shifts through keyword replacement and appending a fixed phrase to text samples. The configuration allows drift to be applied to all samples or targeted to a specific class.

E.2.3 Baseline

We compare LS-OGD against a static baseline. This baseline model is identical to the LS-OGD model architecture but is trained only during Phase 1. In Phase 2, its parameters remain frozen. This baseline represents typical multimodal models that do not adapt to concept drift post-deployment.

E.2.4 LS-OGD Implementation

The LS-OGD framework is implemented as an adaptive controller that interacts with the learning process during Phase 2. First, a drift signal is estimated by monitoring the model's accuracy over a sliding window (size of 100 steps). A significant drop in accuracy indicates potential drift. This aligns with the drift detection mechanism described in Section 3 of the paper. Second, regarding the learning rate adaptation, if the drift signal exceeds a high threshold (0.7), η_t is increased by a factor of 1.5. If it exceeds a moderate threshold (0.2), η_t is decreased by the same factor. The learning rate is bounded between 1e-7 and 1e-2. Last, for the fusion weight adaptation, if moderate drift is detected, and the type of simulated drift suggests a particular modality is compromised, the controller adjusts the internal raw logit for α_t by a step of 0.25 to shift reliance towards the more stable modality.

E.2.5 Training and Evaluation Procedure

The experiment follows a two-phase procedure:

Phase 1 (Initial Training): The multimodal model (both for what will become the static baseline and the initial state of LS-OGD) is trained on the stable dataset partition for 1000 steps.

Phase 2 (Adaptation and Evaluation under Drift): For the static baseline evaluation, the model parameters from Phase 1 training are loaded and frozen, and they are used to evaluate on the drifted data from the Phase 2 partition. Regarding LS-OGD adaptation, the model parameters from Phase 1 training are loaded, and the controller is activated. The model continues to train on the drifted data from the Phase 2 partition for 3000 steps, with the controller dynamically adjusting η_t and α_t. In addition, models are trained using the AdamW optimizer with an initial learning rate of 5e-4 and a weight decay of 0.001. All experiments are seeded for reproducibility.

E.3 Evaluation Metrics

To evaluate the performance and characteristics of LS-OGD and the baseline, we track several metrics throughout Phase 2:

F1-Score: The harmonic mean of precision and recall, providing a balanced measure for binary classification.

Loss: The binary cross-entropy loss.

Learning Rate (η_t): The dynamic learning rate applied by LS-OGD.

Fusion Alpha (α_t): The dynamic fusion weight used by LS-OGD, indicating reliance on text vs. image modality.

Estimated Drift Signal (δ_t proxy): Our drift detection mechanism's output quantifies the perceived change in data distribution based on performance degradation. This is an empirical proxy for δ_t (Axiom 1 in the paper).

Controller Action: The specific adaptation action taken by the LS-OGD controller at each step.

Controller Cost: An assigned cost for each controller action, allowing for analysis of the cumulative cost of adaptation.

Error Signal (e_t): Defined as 1-Accuracy, this is the error signal monitored by the system.

Delta Error Signal (Δe_t): The change in the error signal from the previous step, $e_t - e_{t-1}$. This metric serves as an empirical proxy to observe the error dynamics and the behavior of the Lyapunov function candidate $V(t) = \frac{1}{2} \cdot e_t^2$, as discussed in Lemma 1 and Theorem 1 of the paper.

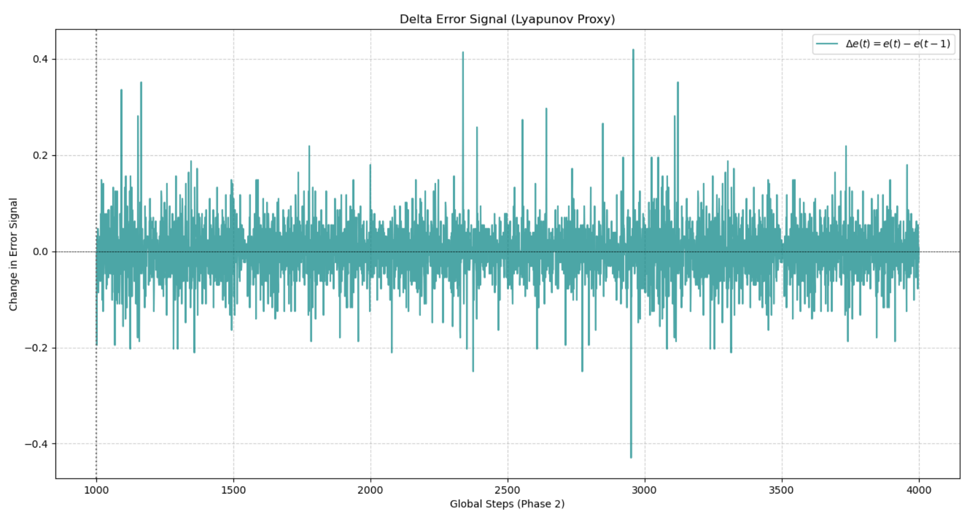

Figure 6: Delta Error Signal for the LS-OGD Model during Phase 2

E.4 Additional Results

Our experiments used 4 NVIDIA A100 GPUs with 110 GB of CPU memory per job, which averaged 20 hours of execution.

Figure 6 is an empirical proxy for observing the behavior related to the Lyapunov stability analysis presented in Lemma 1 and Theorem 1. The Lyapunov function $V(t) = \frac{1}{2}e_t^2$ is stable if its change $\Delta V(t)$ is generally bounded. While this plot shows Δe_t rather than $\Delta V(t)$, Δe_t frequently becomes negative (error is decreasing), supporting the idea that the adaptive controller effectively drives the error down after perturbations. The bounded nature of these fluctuations suggests that the error is not

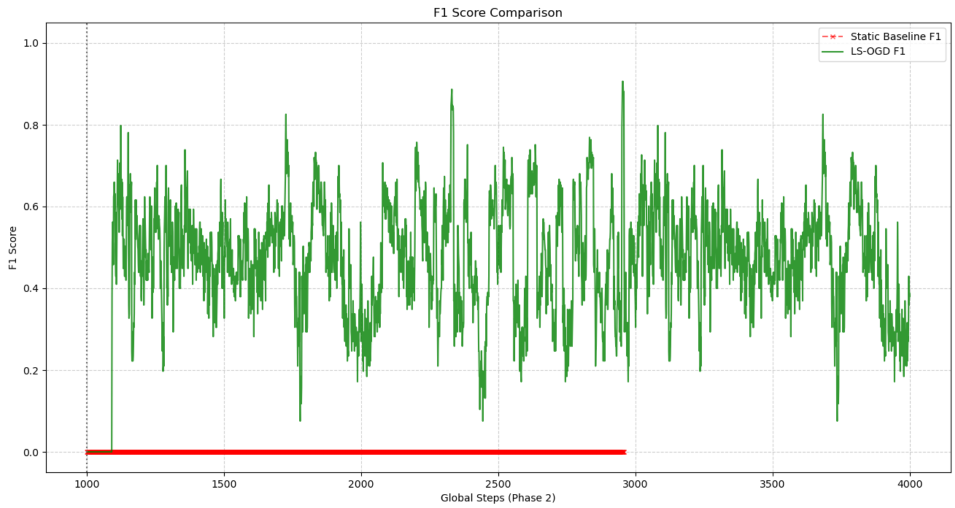

Figure 7: The Comparison between the Static Baseline F1 and LS-OGD F1 over Phase 2

growing uncontrollably, which aligns with the concept of UUB of the error e_t. UUB implies that the error will remain within a certain bound even under persistent drift. Positive spikes in Δe_t are expected when concept drift occurs, as the existing model becomes misaligned with the new data distribution, causing an increase in error. The LS-OGD controller's role is to take corrective actions to make subsequent Δe_t values negative, thereby reducing the overall error e_t.

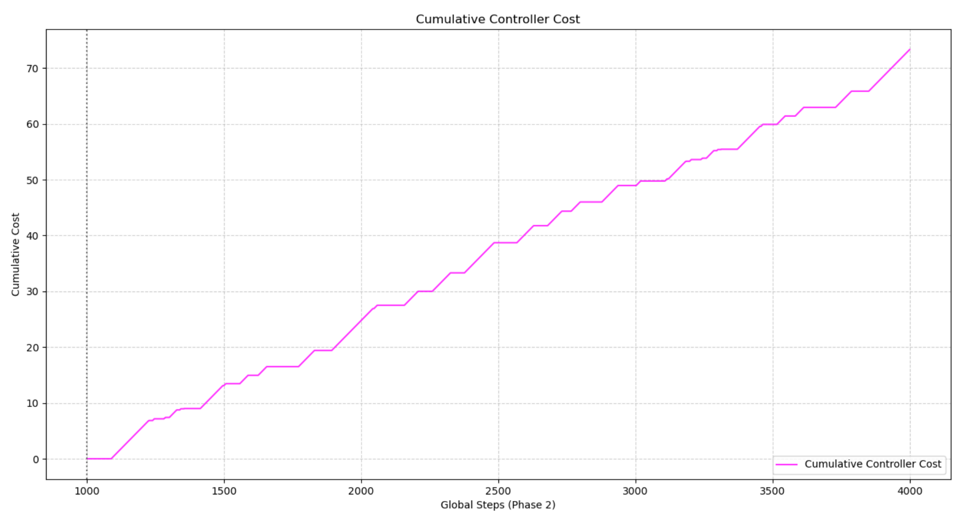

Figure 8: Cumulative Controller Cost for the LS-OGD Model Throughout Phase 2

Figure 7 supports the effectiveness of the LS-OGD framework. The near-zero F1 score of the static baseline highlights the detrimental impact of concept drift on non-adaptive models. LS-OGD maintains a much better F1 score, indicating that it can adapt its decision boundary and modality reliance to changing data. The variability in LS-OGD's F1 score indicates a continuous learning and adaptation process, with the system constantly trying to catch up with the drift. This result aligns with the paper's goal of developing a robust multimodal learning system under concept drift.

Figure 8 provides insight into the “operational overhead" of the LS-OGD adaptive controller. While adaptation is crucial for maintaining performance under drift, it does not come for free. Each adjustment might represent a change in system state that could have other minor costs. The shape of this curve can be correlated with other plots. For instance, periods of rapid increase in controller cost would likely correspond to significant activity in Figure 3, such as changes in learning rate or fusion alpha. Analyzing this cost is important for practical deployments. A controller that adapts effectively but at a high cumulative cost might be less desirable than one that achieves a good balance. The current costs are proxies in our experiments. In a real system, these could represent actual computational overhead, risk associated with change, or resource consumption. This figure helps quantify how “active" the controller needs to be to manage the specific drift scenario in the experiments.

F Test-Time Adaptation (TTA) vs. LS-OGD

Most TTA methods predominantly utilize a fixed, single-modal model for inference, operating without access to labels or source data. These approaches typically focus on minimizing prediction entropy and, in some cases, adjusting normalization layers. The primary objective is to achieve rapid and lightweight adaptation to a stationary or slowly drifting target domain, as exemplified by approaches like Tent [31]. In contrast, the LS-OGD framework addresses the challenges posed by streaming, potentially non-stationary multimodal inputs that exhibit modality-specific drift. It introduces a controller capable of jointly adjusting both the learning rate and the fusion weight in real time. Additionally, LS-OGD offers Lyapunov-style stability guarantees, a feature notably absent in traditional TTA methodologies.

Canonical TTA techniques [31] enhance model confidence through entropy minimization, typically employing only test batches to update batch normalization (BN) affine parameters and/or running statistics. Subsequent variants aimed at stabilizing this objective include CoTTA [32], which incorporates model and augmentation averaging alongside stochastic weight restoration to mitigate forgetting induced by continual data drift. Additionally, SAR [24] employs filtering of high-gradient and noisy samples while executing sharpness-aware entropy minimization, thereby avoiding performance collapse in dynamic and varied data streams. In contrast, LS-OGD introduces a different adaptive approach by focusing on the adaptation of the global learning rate and a time-varying fusion coefficient, $\alpha(t)$. This method conceptualizes prediction error as a feedback signal within a control loop, utilizing error trends to strategically schedule plasticity and adjust modality weights in response to drift, an adaptation mechanism not addressed by pure entropy objectives.

Furthermore, a significant portion of TTA research focuses on the evaluation of static corruptions or gradually varying sequences. In contrast, when evaluation shifts to continual TTA, methodologies must address challenges such as error accumulation and catastrophic forgetting. The framework of CoTTA [32] has been established to formalize continual TTA, demonstrating that naive updates based solely on entropy tend to degrade performance over time. CoTTA proposes enhancements such as weight and augmentation averaging, along with stochastic restoration, to mitigate these issues. Subsequent studies, including NOTE [13], EcoTTA [28], and SoTTA [14], have contributed to additional robustness under non-independent and identically distributed (non-IID) streaming data and have also addressed constraints related to memory and noise that deviate from the main interest. LS-OGD presents a framework explicitly designed to tackle bounded yet recurring drift, providing theoretical guarantees concerning error bounds during the presence of drift and ensuring convergence when the drift subsides. This formulation offers a principled alternative to approaches that rely solely on heuristic stabilization methods.

Moreover, recent advancements in contemporary TTA emphasize empirical stability through various methods, such as sharpness-aware updates for achieving flat minima, entropy ranking/filtering techniques, and protected online self-training combined with change-point detection. However, these approaches still fall short in providing guarantees regarding closed-loop behavior in the presence of drift. The LS-OGD framework introduces a Lyapunov-based analysis that ensures bounded error under drift conditions and demonstrates convergence following instances of drift. This framework allows for the tuning of adaptation laws within stable gain regions, thereby incorporating control-theoretic rigor into the process of online model updates.

G Limitation and Broader Impact

Limitation: While LS-OGD offers robust theoretical guarantees and adaptive capabilities, several limitations must be considered. The framework's practical effectiveness, particularly regarding the dynamic adjustments of fusion weights (as supported by Theorem 2), heavily relies on the accuracy of the underlying drift detection mechanism and the reliable estimation of modality-specific error contributions. In highly entangled multimodal drift scenarios, where changes in different modalities are strongly correlated or the individual impacts of errors are difficult to separate, the current controller's error-based attribution logic may face significant challenges.

Furthermore, LS-OGD introduces several control-specific hyperparameters, such as adaptation gains (k_η, k_α) and various drift detection thresholds (e.g., $W_{\text{drift}}, R_{\text{comp}}, \tau_{\text{drop}}$). Although our theoretical results indicate stability for "appropriately small" gains, optimal tuning to achieve both rapid adaptation and maximum stability in practice may require careful empirical validation across different datasets and drift characteristics.

Moreover, while Axiom 1 supports our stability proofs, the system's performance and guaranteed stability margins during exceptionally rapid or severe drift events, those that significantly violate this boundedness assumption, need further empirical investigation. Finally, the current adaptation focuses primarily on learning dynamics (via η_t) and inter-modality reliance (via α_t). Future research could explore extending adaptive control principles to the internal architectural components or feature representations within the modality-specific encoders, which may enhance resilience to evolving data characteristics.

Additionally, another limitation of the current framework is its dependence on accurate supervised feedback during the test phase to effectively drive the controller. This approach diverges from optimizing unsupervised criteria, such as prediction entropy, or from stabilizing continual data streams without supervision. Such reliance raises a practical challenge, creating a disconnect between the theoretical guarantees presented in the paper and real-world applications characterized by limited labels. To address this gap, two future research directions are proposed. First, the analysis could be extended to accommodate noisy or weak feedback, thereby allowing the controller to leverage imperfect signals that are prevalent in uncontrolled environments, such as clicks or upvotes. Second, it is advisable to eliminate the dependency on supervisory uncertainty as feedback for the controller. In this context, deep ensembles emerge as a well-validated, label-free proxy for monitoring epistemic uncertainty online, facilitating the detection of shifts and enabling appropriate adaptation strategies.

Broader Impact: The development of the LS-OGD framework, which combines control-theoretic stability guarantees with multimodal artificial intelligence (AI), represents a significant step toward creating trustworthy adaptive AI systems that can be more safely deployed in dynamic, real-world environments. Beyond its immediate theoretical contributions, our approach can serve as a blueprint for enhancing continual learning and robust adaptation in domains with inherent non-stationarity and multimodal data.

In autonomous vehicles (AV) design, ensuring operational reliability under constantly changing conditions is critical. AVs rely on a combination of sensors, and their data characteristics can shift due to weather, lighting, sensor degradation, or novel road scenarios. The LS-OGD's principles of adaptive fusion could enable an AV's perception system to dynamically re-weight sensor inputs, reducing reliance on a modality compromised by fog or glare while prioritizing others. Importantly, Theorem 1 offers a pathway to more predictable and verifiable adaptation behavior. The ability to adapt the learning rate would allow AVs to quickly learn from new, unexpected encounters on the road while maintaining overall system stability.

For national security applications, intelligence analysis frequently involves fusing information from disparate multimodal sources, including satellite imagery, textual reports, signals intelligence, and open-source media. The operational environment is characterized by adversaries actively changing tactics, the emergence of novel threat signatures, and evolving geopolitical landscapes. They all are different forms of concept drift. An LS-OGD-like system could provide robust decision support by adaptively integrating these diverse data streams, automatically down-weighting sources that become outdated, compromised, or intentionally deceptive. The stability guarantees ensure that analytical tools remain reliable even as threat patterns evolve, allowing for more consistent situational awareness and the potential for early detection of novel anomalous activities by adapting to new "normal" baselines.

The challenge of misinformation detection also benefits from stable multimodal adaptation. Misinformation campaigns are increasingly sophisticated, blending text, images, and video, and rapidly evolving their narratives, visual styles, and distribution tactics to evade detection. The ability of LS-OGD to adapt its fusion strategy is critical. For example, it could learn to prioritize visual cues if textual components become more anodyne to bypass filters, or vice-versa, as discussed in our problem formulation. By continuously adapting to these shifting tactics with bounded error (Theorem 1), LS-OGD offers a more resilient defense mechanism than static models that quickly become obsolete, thereby contributing to a more informed public discourse.

H NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the claims made, including the contributions made in the paper and important assumptions and limitations.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of this work have been discussed in our Appendix F.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: All the theorems, formulas, and proofs in this paper have been numbered and cross-referenced. All assumptions are clearly stated and referenced in the statement of any theorems. The proofs are in our Appendix D.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Although this paper is submitted to the theory track, we still provide empirical analysis and experiments in both the main paper and Appendix E.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Anyone who is interested in reproducing our experiments using M3A data must apply for access from their official GitHub repo: https://github.com/FinalYou/M3A?tab=readme-ov-file. Our code is publicly available online for peer review only at this link: https://github.com/IDontKnowWhoYouAre1022/LS-OGD_Review_ONLY

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All experimental setting and details have been provided in our Appendix E.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The experiment statistics and evaluation metrics have been provided in Appendix E.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The experiments compute resources are reported in Appendix E.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The authors have reviewed the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We have discussed the broader impacts in the conclusion section and Appendix F.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: NA.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cited all resources we used in our reference.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: NA.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: NA.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: NA.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [No]

    Justification: LLMs were only used for editing purposes and do not impact the core methodology, scientific rigor, and originality of this research.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.