LEARNING TO DRIFT: MULTIMODAL CONTINUAL LEARNING VIA DIFFERENTIABLE DELTA MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal continual learning (MCL) aims to learn from a sequence of tasks involving multiple modalities while retaining previously acquired modality-specific knowledge and cross-modal relationships. A key challenge in MCL is the drift of representations across tasks. Despite numerous representation compensation methods, most address modality-specific drift in isolation and struggle with cross-modal interactions in MCL. This paper proposes DriftMem, a novel MCL method that learns to drift through associative memory. Specifically, DriftMem integrates Differentiable Delta Memory (DDM) and Attention-guided Gradient Recovery (AGR). The former formulates associative memory as differentiable delta-rule updates, and the latter uses attention to guide restorative gradients and stabilize cross-modal interactions. To address class conflicts in memory, DriftMem further incorporates Class-structured Feature Calibration (CFC) with two auxiliary losses for reliable memory retrieval and updates. Experiments on four datasets in the general yet challenging MCL setting of class-incremental learning show remarkable performance improvements over 10 baselines, along with ablation studies supporting the significant gains from each component.

1 INTRODUCTION

Recent advances in multimodal learning (or more broadly multiview learning) have enabled neural networks to effectively integrate complementary information across heterogeneous modalities such as vision, audio and text (Radford et al., 2021; Alayrac et al., 2022; Girdhar et al., 2023). However, real-world multimodal data continuously evolves. This motivates multimodal continual learning (MCL), which aims to build a model to learn from evolving multimodal data while preserving previously acquired knowledge (Parisi et al., 2019; De Lange et al., 2021; Zhou et al., 2024; Wang et al., 2024). A central challenge is *catastrophic forgetting*, i.e., models tend to rapidly and drastically forget previously learned knowledge when learning new tasks (Kirkpatrick et al., 2017).

Preserving such knowledge is closely tied to the evolution of the underlying representation space. As a model adapts to new tasks, parameter updates inevitably alter its representations, causing the representations of previously learned inputs to shift within the feature space, a phenomenon known as *representation drift*. Specifically, representational drift refers to the gradual change across tasks in the neural activity patterns or feature representations associated with the same information or stimulus (Climer et al., 2025). Prior studies on continual learning have identified representation drift as a critical factor associated with catastrophic forgetting (Caccia et al., 2022; Davari et al., 2022; Luo et al., 2023; Kim et al., 2025). To mitigate representation drift, a typical strategy is to constrain updates to model parameters that encode previous knowledge or enforce consistency between current and historical representations (Wang et al., 2023; Pian et al., 2023; Mo et al., 2023; Yue et al., 2024). On the other hand, representation drift is non-trivial for adapting to new and future tasks. This falls into a dilemma we call the *representation drift dilemma*: restricting representation drift improves stability but limits plasticity for learning new tasks, whereas excessive drift enhances adaptation but leads to forgetting.

Representation compensation has recently emerged as an alternative approach to solving the drift dilemma. It mainly evolves feature space and then corrects mismatch between historical knowledge and current representation (Yu et al., 2020; Goswami et al., 2024; He et al., 2025). However, most existing compensation methods rely on class-level statistics or distributional shifts, which may be

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

insufficient to characterize the multimodal representation drift. The issue of representation drift is even more challenging in MCL because representation drift involves both modality-specific representations and the cross-modal interactions that jointly shape the fused representation. When cross-modal interactions evolve, the same semantic concept would be encoded differently across task stages, making the correspondence between current representations and historical states unstable. Therefore, compensation requires new mechanisms that recover historical semantics from evolving representations by modeling such a correspondence.

To address the aforementioned challenges and issues, we delve into associative memory (Hopfield, 1982; Schlag et al., 2021), which provides a new compensation mechanism by learning associations between current representations and corresponding historical knowledge. We propose a novel MCL method, called DriftMem, that learns to compensate for representation drift using associative memory. The novelties of DriftMem are at least threefold: *Differentiable Delta Memory* (DDM) for memory construction, *Attention-guided Gradient Recovery* (AGR) for cross-modal interaction stabilization, and *Class-structured Feature Calibration* (CFC) for memory consolidation. Specifically, DDM learns associations between semantic keys and drift residuals from replay examples through differentiable delta-rule updates. AGR uses attention drift to guide restorative gradient adjustments, thereby stabilizing cross-modal interactions during continual adaptation. CFC structures semantic keys according to class geometry, reducing inter-class interference during memory updates. In a nutshell, DriftMem enables multimodal representations to adapt to new tasks while preserving historical knowledge through associative drift compensation and stable cross-modal learning.

In summary, we make the following contributions:

- We propose DriftMem, a novel MCL framework that learns to compensate for representation drift through a new associative memory paradigm, enabling continual adaptation while preserving historical knowledge.
- We introduce three complementary components: DDM for learning associations between semantic keys and drift residuals, AGR for stabilizing cross-modal interactions through restorative gradient adjustments, and CFC for reducing inter-class interference during memory updates.
- Extensive experiments on four datasets in the challenging class-incremental MCL setting demonstrate that DriftMem consistently outperforms ten competitive baselines. Ablation studies further validate the contribution of each proposed component.

2 Related Work

Multimodal Continual Learning. Existing multimodal continual learning (MCL) methods can be broadly classified into four categories: *regularization-based*, *prompt-based*, *architecture-based*, and *replay-based* methods. Specifically, regularization-based methods constrain parameter updates to balance stability and plasticity. AV-CIL (Pian et al., 2023) employs visual attention distillation to preserve audio-guided visual associations. TIR (He et al., 2026) estimates task similarities to adaptively weight parameter regularization. Prompt-based methods mainly utilize predefined or learnable prompts to facilitate continual learning across multimodal tasks. For example, L2P (Wang et al., 2022c) and DualPrompt (Wang et al., 2022b) learn a set of prompts and retrieve relevant prompts conditioned on input. Architecture-based methods assign different model parameters to tackle different tasks, such as CMR-MFN (Wang et al., 2023) fixes encoders of each modality and adds a modality fusion network for each task. Replay-based methods usually retain a small number of old samples as supervisory signals. For example, SAMM (Sarfraz et al., 2024) and VQACL Zhang et al. (2023) randomly select multimodal sample to replay, while FGVIRs (He et al., 2024) and AID (Cheng et al., 2024) develop pseudo-representation and pseudo-prototype as replay strategies. However, most existing MCL approaches struggle with the issue of representation drift dilemma. In this work, we explicitly model historical correction signals through associative memory and use it to compensate for the original multimodal representation, rather than enforcing the historical representations to remain unchanged.

Representation Compensation. Representation compensation opens up an alternative pathway to address the representation drift dilemma. SDC (Yu et al., 2020) estimates the displacement of old-class representations from the feature changes observed in current-task samples across consecutive models. This method lays a foundation for subsequent drift-compensation approaches such as

(Goswami et al., 2024; Gomez-Villa et al., 2024; He et al., 2025). However, these methods primarily adapt historical class summaries to the evolving representation space. In contrast, our method explores a complementary compensation direction. Our method uses replay examples as historical reference points to learn residual corrections that move current fused representations toward their consolidated historical counterparts.

Delta Rule. The delta rule provides an error-driven mechanism for updating associative mappings (Hanson, 1990). Schlag et al. (2021) establishes a connection between linear attention and fast-weight programmers and devises a delta-rule-based alternative to additive associative updates. Yang et al. (2024) proposed DeltaNet and developed hardware-efficient parallel training for delta-rule linear Transformers. Furthermore, gated DeltaNet Hatamizadeh et al. (2026) combines delta updates with a forgetting gate, integrating targeted association updates with adaptive memory retention. Beyond the delta rule in isolation, we apply this error-driven update principle to historical representation recovery across various tasks. That is, memory values are recovery signals and their state is retained throughout the process of continual learning.

3 Problem Statement

Problem Formulation. Without loss of generality, we formulate the problem following the typical MCL setting of Class Incremental Learning (CIL) Rebuffi et al. (2017). Given a sequence of T tasks with disjoint class sets $\mathcal{C}_s \cap \mathcal{C}_t = \emptyset$ for $s \neq t$, each task t has its training data $\mathcal{D}_t = \{(x_i, y_i)\}_{i=1}^{N_t}$, where $x_i \in \mathcal{R}$ and $y_i \in \mathcal{C}_t$ denotes data features and labels, respectively. Each x_i consists of an anchor modality and a context modality, denoted as $x_i = (x_i^a, x_i^c)$. After completing the training of the t-th task, the model is evaluated on the test dataset $\mathcal{D}_t^{test}$, which is composed of all seen classes $\mathcal{C}_{\leq t} = \bigcup_{s=1}^{t} \mathcal{C}_s$ without task identity. A small set of samples from previous tasks is retained as exemplar data for training, denoted as $\mathcal{E}_t$. At the t-th training step, suppose the model can only access $\mathcal{D}_t \cup \mathcal{E}_t$. Let $\mathbf{z}_{\phi^t}(x) \in \mathbb{R}^d$ be the uncompensated fused feature and g_{ω^t} the classifier over $\mathcal{C}_{\leq t}$. Formally, the problem of MCL is formulated as follows:

$$\arg\min_{\phi^t, \omega^t} \mathbb{E}_{(x,y)\sim\mathcal{D}_{\leq t}} \mathcal{L}_{\text{CE}}(g_{\omega^t}(\mathbf{z}_{\phi^t}(x)), y), \tag{1}$$

where $\mathcal{D}_{\leq t} = \bigcup_{s\leq t} \mathcal{D}_s$. For clarity, we use superscripts for task model indices and subscripts for sample indices throughout the paper.

Representation Drift. For a historical sample x, let $\hat{\mathbf{z}}^{t-1}(x)$ denote its consolidated feature from the frozen previous model, including previous memory compensation. $\mathbf{z}^t = \mathbf{z}_{\phi^t}(x)$ denotes the current uncompensated feature. Then, representation drift is defined as a recovery-oriented residual, formulated as follows

$$\Delta\mathbf{z}^t(x) = \hat{\mathbf{z}}^{t-1}(x) - \mathbf{z}^t(x). \tag{2}$$

This residual points from the current feature toward its historical reference. Instead of requiring $\mathbf{z}^t$ to remain fixed, we learn an input-dependent correction $\mathcal{F}^t(\mathbf{k}^t(x))$, where $\mathbf{k}^t$ is a semantic key constructed from the two modality branches. The corresponding prediction objective is

$$\min_{\phi^t, \omega^t, \mathcal{F}^t} \mathbb{E}_{(x,y)\sim\mathcal{D}_{\leq t}} \mathcal{L}_{CE}\left(g_{\omega^t}\left(\mathbf{z}^t(x) + \gamma_{\text{mem}}\mathcal{F}^t(\mathbf{k}^t(x))\right), y\right). \tag{3}$$

Notably, our algorithm updates networks and delta memory separately. That is, it does not directly optimize this ideal cumulative objective.

4 Methodology

Figure 1 illustrates the overall architecture of the proposed DriftMem. As shown in Figure 1, it consists of three parts: (1) *Model Architecture*, (2) *Differentiable Delta Memory* (DDM), and (3) *Attention-guided Gradient Recovery* (AGR). In addition, DriftMem consolidates memory using *Class-structured Feature Calibration* (CFC) with two auxiliary losses to address class conflicts in memory. We next elaborate on our solutions to each component.

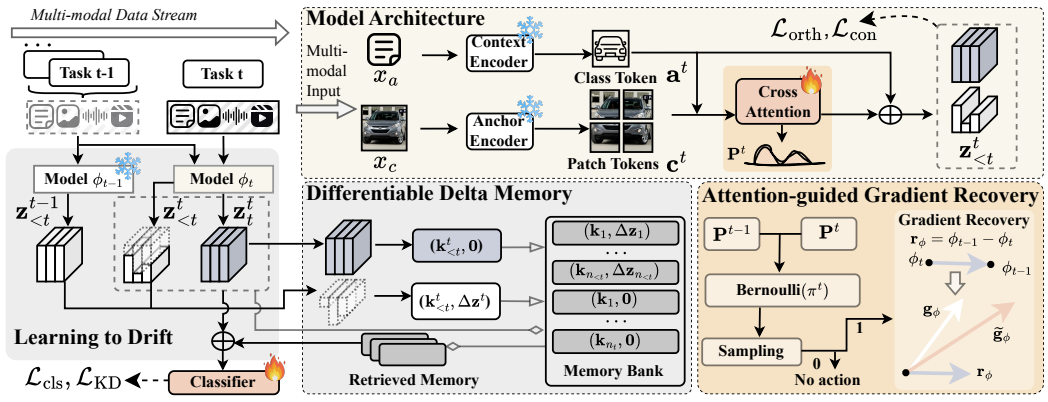


Figure 1: An overview of the proposed DriftMem. The model architecture (*yellow*) is a cross-modal representation network. Differentiable Delta Memory (*gray*) associates semantic keys with drift residuals for representation compensation, with hollow arrows denoting memory read/write operations. Attention-guided Gradient Recovery (*orange*) stabilizes cross-modal interactions based on attention drift. Class-structured Feature Calibration (CFC) implements two auxiliary losses. [Best viewed in color]

4.1 Model Architecture

We follow the cross-modal paradigm as introduced by Li et al. (2021); Pian et al. (2023). As shown in Figure 1, the inputs of multimodal data to our model are composed of anchor modality and context modality, whose representations are denoted as $x^a \in \mathbb{R}^d$ and $x^c \in \mathbb{R}^{N_1 \times \cdots \times N_k \times d}$, respectively. The anchor provides global semantic guidance using only the [CLS] token, while context tokens retain local structure. Image data has a spatial axis, and video data has both a spatial axis and a temporal axis. Each attention operation aggregates one structural axis of the context modality while retaining the feature channels. Given input data (x^a, x^c), we first project anchor and current context representations, then feed them into a trainable attention, formulated as

$$\mathbf{a} = \tanh(W_a x^a + b_a), \qquad \mathbf{c}_\ell = \tanh(W_c x^c_\ell + b_c), \tag{4}$$

where W_a, W_c, b_a, b_c are learnable parameters, ℓ indexes locations and h indexes channels. For an aggregation axis ρ, attention-based scores are

$$\mathbf{P}_{\ell,h} = \frac{\exp(S_{\ell,h})}{\sum_{j \in \rho} \exp(S_{j,h})}, \qquad S_{\ell,h} = \bar{a}_h \odot \bar{c}_{\ell,h}, \tag{5}$$

where $\odot$ represents the Hadamard product. For image context, the aggregated representation is $\tilde{\mathbf{c}}_h = \sum_\ell \mathbf{P}_{\ell,h} c_{\ell,h}$. For video context, such aggregation is employed on both spatial ℓ_s and temporal ℓ_t axes that $\tilde{\mathbf{c}} = \sum_{\ell_t} \mathbf{P}_{\ell_t,h} \sum_{\ell_s} \mathbf{P}_{\ell_t,\ell_s,h} \mathbf{c}_{\ell_t,\ell_s,h}$. Finally, our fused representation $\mathbf{z}$ is produced by the projected anchor and aggregated context representation as

$$\mathbf{u}_a = \mathrm{ReLU}(W'_a \mathbf{a} + \mathbf{b}'_a), \quad \mathbf{u}_c = \mathrm{ReLU}(W'_c \tilde{\mathbf{c}} + \mathbf{b}'_c), \quad \mathbf{z} = \mathbf{u}_a + \mathbf{u}_c, \tag{6}$$

where W'_a, W'_c, b'_a, b'_c are learnable parameters. $\mathbf{u}_a$ represents the modality-specific feature, and $\mathbf{u}_c$ denotes the result of cross-modal interaction.

4.2 Differentiable Delta Memory

As aforementioned, inevitable changes in model parameters cause both modality-specific features and cross-modal interactions to evolve, thereby inducing drift in the previously learned multimodal representations as defined in Equation 6. Rather than requiring these representations to remain fixed, we learn an correction residual that recovers historical semantics from the evolving feature space. Inspired by the DeltaNet Schlag et al. (2021), which incorporates a fast-weight programmer into linear attention as an associative memory and updates it using the delta rule, we develop a fast Differentiable Delta Memory (DDM) $\mathbf{M} \in \mathbb{R}^{d \times d}$, which associates semantic keys from replay samples with the residuals needed for representation recovery.

Semantic Keys and Recovery Targets. Given the final anchor and attended context features $\mathbf{u}_a, \mathbf{u}_c \in \mathbb{R}^d$, we construct a semantic key by normalizing the two branches separately before combining them, denoted as follows

$$\mathbf{k} = \mathrm{n}\left(\frac{\mathrm{n}(\mathbf{u}_a) + \mathrm{n}(\mathbf{u}_c)}{2}\right), \qquad \mathrm{n}(\mathbf{u}) = \frac{\mathbf{u}}{\max\{\|\mathbf{u}\|_2, \epsilon\}}. \tag{7}$$

This construction balances the contributions of the two branches without tying retrieval to their feature magnitudes. For a replay sample, the detached target is $\Delta\mathbf{z}_i^t = \mathrm{sg}\big(\hat{\mathbf{z}}_i^{t-1} - \mathbf{z}_i^t\big)$, where $\mathrm{sg}(\cdot)$ is a stop-gradient operation. We utilize semantic key to retrieve the corresponding residual from memory, denoted as $\mathbf{M}\mathbf{k}_i^t$. The previous feature $\hat{\mathbf{z}}_i^{t-1}$ includes its learned compensation. Thus each task references the representation actually available after the preceding task.

Joint Residual Learning. Fitting historical residuals alone can produce unintended responses on current-task samples. Since task identity is unavailable at inference, the memory must learn when compensation is useful from the semantic key itself. We therefore assign zero residual targets to current-task samples. Let $\mathcal{B}_e \subset \mathcal{E}_t$ and $\mathcal{B}_c \subset \mathcal{D}_t$ denote replay and current-task mini-batches. For fixed keys and targets, we formulate memory learning as a joint residual regression:

$$\mathcal{J}(\mathbf{M}) = \frac{1}{2|\mathcal{B}_e|}\sum_{i\in\mathcal{B}_e}\|\Delta\mathbf{z}_i^t - \mathbf{M}\mathbf{k}_i^t\|_2^2 + \frac{\lambda_c}{2\lambda_e|\mathcal{B}_c|}\sum_{j\in\mathcal{B}_c}\|\mathbf{M}\mathbf{k}_j^t\|_2^2. \tag{8}$$

Taking a gradient step on Equation 8, we have

$$\mathbf{M}^+ = \mathbf{M} + \frac{1}{|\mathcal{B}_e|}\sum_{i\in\mathcal{B}_e}(\Delta\mathbf{z}_i^t - \mathbf{M}\mathbf{k}_i^t)(\mathbf{k}_i^t)^\top + \frac{\lambda_c}{|\mathcal{B}_c|}\sum_{j\in\mathcal{B}_c}(-\mathbf{M}\mathbf{k}_j^t)(\mathbf{k}_j^t)^\top, \tag{9}$$

where λ_c control historical residual fitting and current-task response suppression, respectively. Both errors are evaluated using the same pre-update memory. The resulting associations are encouraged to satisfy $\mathbf{M}\mathbf{k}_i \approx \Delta\mathbf{z}_i, i \in \mathcal{B}_e$ and $\mathbf{M}\mathbf{k}_j \approx \mathbf{0}, j \in \mathcal{B}_c$. In practice, we also clip residual errors to stabilize memory updates. The derivation of Equation 9 and its conditional convergence analysis are provided in Appendix A.1.

Representation Compensation. The retrieved residual is incorporated into the current representation before classification:

$$\hat{\mathbf{z}}^t = \mathbf{z}^t + \gamma_{\mathrm{mem}}\mathbf{M}\mathbf{k}^t, \qquad \hat{y}^t = g_{\omega^t}(\hat{\mathbf{z}}^t), \tag{10}$$

where $\gamma_{\mathrm{mem}} \geq 0$ controls the compensation strength. For a perfectly fitted replay target, $\hat{\mathbf{z}}_i = (1 - \gamma_{\mathrm{mem}})\mathbf{z}_i + \gamma_{\mathrm{mem}}\hat{\mathbf{z}}^{t-1}$. Thus, choosing $0 < \gamma_{\mathrm{mem}} < 1$ provides partial recovery while retaining part of the current representation. It is worth noting that the residual target is defined in the fused feature space, which captures changes arising from modality-specific features, cross-modal aggregation, and previously accumulated memory compensation. As a result, our delta memory achieve a more fine-grained compensation compared to prototype-based or statistics-based compensation.

During training, the memory remains fixed for backpropagation, while gradients through its read operation update the representation network. After the optimizer step, we apply the delta update using detached keys and targets cached from the forward pass. At inference, the memory is read without further updates, providing sample-dependent compensation without requiring task identity.

4.3 Attention-guided Gradient Recovery

Beyond intra-modal knowledge forgetting, evolving cross-modal interactions can also induce representation drift. Since the semantic key in Equation 7 is constructed through anchor-guided context aggregation, changes in cross-modal attention directly affect the resulting key representation. In this section, we introduce Attention-guided Gradient Recovery (AGR) to promote stable cross-modal interactions in the attention network and reliable retrieval to the greatest extent possible.

Attention Drift Measure. Let $\mathcal{B}_e \subset \mathcal{E}_t$ be a replay mini-batch, and let $\mathcal{R}$ contain the attention stages: spatial attention for image context, and spatial and temporal attention for video context.

For stage $r \in \mathcal{R}$, denote the current and frozen previous-model attention distributions by $\mathbf{p}_{i,q}^{t,r}$ and $\mathbf{p}_{i,q}^{t-1,r}$, respectively. Each distribution is normalized along the aggregation axis; $q \in \mathcal{Q}_r$ indexes the remaining axes, including feature channels and, for spatial video attention, frames. We measure their discrepancy using the normalized Jensen–Shannon divergence:

$$D_r^t = \frac{1}{|\mathcal{B}_e||\mathcal{Q}_r|\log 2} \sum_{i\in\mathcal{B}_e} \sum_{q\in\mathcal{Q}_r} \mathrm{JS}\Big(\mathbf{p}_{i,q}^{t,r}, \mathbf{p}_{i,q}^{t-1,r}\Big), \tag{11}$$
$$\mathrm{JS}(\mathbf{p},\mathbf{q}) = \tfrac{1}{2}\,\mathrm{KL}(\mathbf{p}\|\mathbf{m}) + \tfrac{1}{2}\,\mathrm{KL}(\mathbf{q}\|\mathbf{m}), \qquad \mathbf{m} = \tfrac{1}{2}(\mathbf{p}+\mathbf{q}).$$

With natural logarithms, $D_r^t \in [0,1]$. We convert each stage's discrepancy into an activation probability and average across stages:

$$\pi^t = \frac{1}{|\mathcal{R}|} \sum_{r\in\mathcal{R}} \sigma\left(\frac{D_r^t - \delta_{\mathrm{att}}}{\tau_{\mathrm{att}}}\right), \quad b \sim \mathrm{Bernoulli}(\pi^t), \tag{12}$$

where $\sigma(\cdot)$ is a sigmoid function, δ_{att} controls its transition point, and $\tau_{\mathrm{att}} > 0$ denotes temperature which controls its sensitivity. In practice, attention statistics and π^t will be detached from the computation graph.

Attention Parameter Recovery. We now introduce how to recover attention parameters. Let $\phi_t \in \{W_a, b_a, W_c, b_c\}$ denote an individual attention-projection parameter tensor, and let ϕ_{t-1} be the frozen parameters from the previous task. Given ordinary gradient $\mathbf{g}_\phi = \nabla_{\phi^t}\mathcal{L}_{\mathrm{train}}$, we define

$$\mathbf{d}_\phi = \phi_t - \phi_{t-1}, \qquad \mathbf{r}_\phi = \min\left\{1, \frac{\|\mathbf{g}_\phi\|_F}{\|\mathbf{d}_\phi\|_F}\right\}\mathbf{d}_\phi, \qquad \widetilde{\mathbf{g}}_\phi = \mathbf{g}_\phi + b\mathbf{r}_\phi. \tag{13}$$

All attention-projection tensors share the same Bernoulli sample within each optimization step. The optimizer receives $\widetilde{\mathbf{g}}_\phi$ for these tensors and the ordinary gradients for the other trainable parameters. Proofs of Equation 13 and additional details are provide in Appendix A.3. The added gradient points to the displacement from the previous parameters, so its negative supplies a restoring direction. Its norm is bounded by $\|\mathbf{g}_\phi\|_F$, which prevents the correction from being arbitrarily large relative to the ordinary gradient. Intuitively, larger attention drift increases the frequency of recovery while retaining updates driven by the training objective. AGR thus regulates the attention pathway to maintain stable cross-modal interactions, while the Delta Memory estimates residual corrections for the resulting fused representations.

4.4 Class-structured Feature Calibration

Reliable retrieval also requires separating associations stored for different classes. Suppose an error-driven memory update $\mathbf{M}^+ = \mathbf{M} + \eta\mathbf{e}_i\mathbf{k}_i^\top$, where $\mathbf{e}_i$ is the residual prediction error, its effect on another key $\mathbf{k}_j$ is

$$(\mathbf{M}^+ - \mathbf{M})\mathbf{k}_j = \eta\mathbf{e}_i\langle\mathbf{k}_i, \mathbf{k}_j\rangle. \tag{14}$$

From Equation 14, we can see correlated keys couple memory updates across samples, including zero-target updates for new classes. We thus introduce Class-structured Feature Calibration (CFC) to reduce interference between semantic keys during the retrieval and update of DDM.

Inter-class Orthogonal Calibration. Let $\mathcal{B} = \mathcal{B}_c \cup \mathcal{B}_e$ combine current-task and exemplar samples, and let $\mathcal{C}_\mathcal{B}$ contain its $C_\mathcal{B}$ distinct classes. We construct a normalized prototype from the semantic keys of each class:

$$\boldsymbol{\mu}_c = \frac{\sum_{i\in\mathcal{B}_c^{\mathrm{cls}}}\mathbf{k}_i^t}{|\mathcal{B}_c^{\mathrm{cls}}|}, \qquad \mathbf{v}_c = \frac{\boldsymbol{\mu}_c}{\|\boldsymbol{\mu}_c\|_2}, \qquad \mathcal{B}_c^{\mathrm{cls}} = \{i \in \mathcal{B} : y_i = c\}, \tag{15}$$

where the keys are unit-normalized, as defined in Equation 7. The inter-class orthogonality loss is

$$\mathcal{L}_{\mathrm{orth}} = \frac{1}{C_\mathcal{B}(C_\mathcal{B}-1)} \sum_{\substack{c,c'\in\mathcal{C}_\mathcal{B}\\ c\neq c'}} (\mathbf{v}_c^\top\mathbf{v}_{c'})^2. \tag{16}$$

This objective penalizes correlations between class directions without imposing an identical key on every sample. Its connection to sample-level interference additionally depends on within-class key dispersion, see Appendix 3.

Class-wise Contrastive Calibration. Orthogonality alone does not ensure that the compensated representations remain discriminative. We therefore apply a supervised contrastive objective (Khosla et al., 2020) to $\mathbf{k}_i$. given a positive set $\mathcal{P}(i) = \{p \in \mathcal{B} : p \neq i,\ y_p = y_i\}$ and a valid-anchor set $\mathcal{V} = \{i \in \mathcal{B} : |\mathcal{P}(i)| > 0\}$, the contrastive loss is

$$\mathcal{L}_{\text{con}} = -\frac{1}{|\mathcal{V}|} \sum_{i \in \mathcal{V}} \frac{1}{|\mathcal{P}(i)|} \sum_{p \in \mathcal{P}(i)} \log \frac{\exp(\mathbf{k}_i^\top \mathbf{k}_p / \tau_{\text{con}})}{\sum_{j \in \mathcal{B} \setminus \{i\}} \exp(\mathbf{k}_i^\top \mathbf{k}_j / \tau_{\text{con}})}, \tag{17}$$

where τ_{con} denoted the temperature coefficient. We set this loss to zero when $\mathcal{V}$ is empty. Self-pairs are excluded from both the positive set and the denominator. This loss encourages class-wise compactness and inter-class separability in the semantic key space, complementing the prototype-level orthogonality constraint. In such a method, $\mathcal{L}_{\text{orth}}$ and $\mathcal{L}_{\text{con}}$ together improve key discrimination and reduce interference among memory associations.

Training and Inference. For incremental tasks $t > 1$, the final objective loss of DriftMem is

$$\mathcal{L}_{\text{train}} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{KD}} + \lambda_{\text{orth}} \mathcal{L}_{\text{orth}} + \lambda_{\text{con}} \mathcal{L}_{\text{con}}, \tag{18}$$

where $\mathcal{L}_{\text{cls}}$ combines classification on current and replay samples, and $\mathcal{L}_{\text{KD}}$ distills previous-task logits. Their precise forms are given in Appendix B.1. After backpropagation, AGR modifies the attention-projection gradients before the optimizer step. The DDM is then updated using detached keys and residual targets from the same forward pass. The memory remains fixed during backpropagation, while gradients through its read operation update the representation network. At inference, the model reads the learned memory and classifies $\hat{\mathbf{z}}^t$ over all seen classes.

5 Experiments

5.1 Experimental Setups

Datasets. We evaluated the proposed model and baselines on four benchmark datasets, including two video-audio datasets and two image-text datasets. To be specific, **AVE** (Tian et al., 2018) contains 28 audio-visual event categories, which are split into four tasks. **Kinetics-Sounds** (Arandjelovic & Zisserman, 2017) is a subset of Kinetics-400, containing 31 human action classes. We follow Tian et al. (2018) and randomly select 30 classes and divide them into five sequential tasks with six classes per task. **N24News** (Wang et al., 2022a) consists of both text and image information for each news article, with 24 categories split into four tasks of six classes each. **UMPC-Food101** (Wang et al., 2015) consists of 101 food categories, with 11 classes introduced in the first task and the remaining 90 classes divided into nine subsequent tasks of 10 classes each.

Baselines. To demonstrate the effectiveness of our proposed method, we compare it with ten state-of-the-art methods, which can be categorized into three groups: 1) **Vanilla CL:** LwF (Li & Hoiem, 2017), iCaRL (Rebuffi et al., 2017), SS-IL (Ahn et al., 2021), and AFC (Kang et al., 2022); 2) **CL with Representation Compensation:** SDC (Yu et al., 2020), ADC (Goswami et al., 2024), and DPCR (He et al., 2025); and 3) **MCL:** CMR-MFN (Wang et al., 2023), AV-CIL (Pian et al., 2023), and MMAL (Yue et al., 2024). More details of each baseline are provided in Appendix B.3.

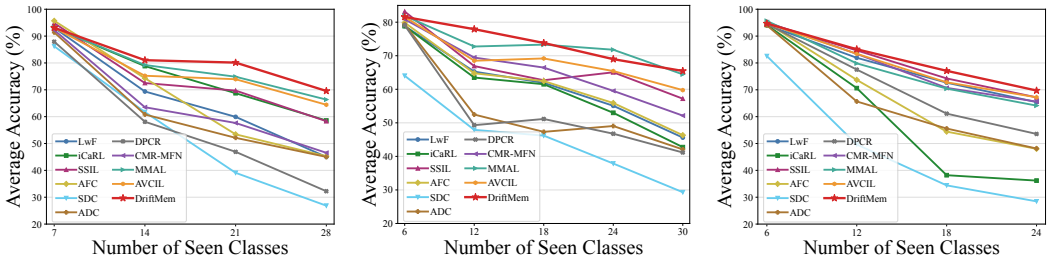
Metrics. To comprehensively evaluate our proposed DriftMem and other baselines, we employ three metrics: *Cumulative Average Accuracy* (CAA), *Final Average Accuracy* (FAA), and *Final Forgetting Measure* (FFM). We denote the test accuracy on task i with $A_{i,t}$ and the average accuracy over all classes seen with AA_t at the t-th incremental step. After finishing all T steps, we report $\text{CAA} = \frac{1}{T}\sum_t^T AA_t$, $\text{FFM} = \frac{1}{T-1}\sum_i^{T-1} \max_{t\in\{1,\cdots,T-1\}}(A_{i,t} - A_{i,T})$ and $\text{FAA} = AA_T$. The CAA and FAA reflect the overall learning performance throughout the incremental process and the final performance after learning all tasks, respectively. The FFM measures catastrophic forgetting

5.2 Experimental Results

Main Results. Evaluation results on four datasets in the setting of CIL are shown in Table 1 and Table 2. From the results, we have the following observations:

Table 1: Performance comparison on four datasets. The best results are shown in **bold** and the second-best results are underlined.

Method	AVE			Kinetics-Sounds			N24News			UMPC-Food101		
	CAA(↑)	FFM(↓)	FAA(↑)	CAA(↑)	FFM(↓)	FAA(↑)	CAA(↑)	FFM(↓)	FAA(↑)	CAA(↑)	FFM(↓)	FAA(↑)
LwF	66.85	32.85	44.98	61.34	27.62	45.82	78.53	16.53	65.70	84.87	22.74	69.75
iCaRL	69.81	35.60	59.98	59.92	51.92	42.95	59.78	73.06	35.31	89.22	17.96	81.46
SSIL	73.84	19.22	58.04	67.01	14.91	57.14	80.23	13.68	67.27	93.72	3.53	90.22
AFC	67.18	46.25	48.65	61.86	46.13	46.63	67.77	57.06	47.40	93.01	8.72	88.92
SDC	53.07	15.99	44.30	34.59	12.31	30.65	50.27	13.06	43.82	85.92	7.01	82.17
ADC	62.33	18.15	44.40	54.05	18.50	42.17	65.85	15.81	47.78	74.95	9.26	67.73
DPCR	56.30	41.90	34.07	53.49	36.95	41.33	71.68	31.78	53.82	82.85	21.78	73.21
CMR-MFN	64.98	26.09	46.41	65.69	21.32	52.31	78.72	**9.98**	65.24	92.01	4.37	88.85
MMAL	76.25	9.56	66.36	72.84	18.24	64.50	77.52	17.68	63.84	90.82	3.69	88.07
AV-CIL	74.11	10.77	64.43	68.17	14.97	60.81	75.50	16.65	66.21	92.08	5.79	89.71
DriftMem	**79.23**	**6.89**	**69.57**	**74.32**	**10.91**	**65.46**	**81.55**	10.50	**68.90**	**95.28**	**2.66**	**93.82**



(a) AVE (b) Kinetics-Sounds (c) N24News

Figure 2: Incremental curves of AVE, Kinetics-Sounds and N24News datasets. We report Average Accuracy at each incremental step.

- Representation compensation methods show inferior performance compared with vanilla CL and MCL methods, especially on video-audio datasets, suggesting that existing compensation strategies are not insufficient to characterize the multimodal representation drift. In contrast, DriftMem consistently outperforms conventional CL baselines by explicitly modeling representation drift rather than treating it as an undesirable deviation, providing an effective solution to alleviate the representation drift dilemma.
- DriftMem achieves competitive performance against existing MCL methods, while MMAL is the second best model. Although CMR-MFN achieves lower FFM in specific cases, DriftMem maintains advantages in overall performance and forgetting mitigation across datasets. These results highlight the importance of modeling multi-modal representation drift.
- We further visualize the incremental curves on AVE, Kinetics-Sounds, and N24News in Figure 2. DriftMem achieves the best performance at every incremental step on all three dataset. Although it is slightly inferior at the initial stage, DriftMem consistently surpasses other methods in later incremental steps.

Ablation Studies. Since our DriftMem consists of Differentiable Delta Memory (DDM), Attention-guided Gradient Recovery (AGR) and Class-structured Feature Calibration (CFC), we now examine each of them. We conduct an ablation study. The results are shown in Table 2. From the results, we see that the removal of any modules leads to performance degradation on the AVE dataset. In particular, DDM contribute more than the other two. Additionally, a single AGR results in a performance drop, possibly because it leads to increase the overlap between old and new task representations.

Table 2: Ablation study of three components on AVE datasets.

DDM	AGR	CFC	CAA ↑	FFM ↓	FAA ↑
✗	✗	✗	71.25	26.65	52.69
✗	✗	✓	73.22	18.91	61.89
✗	✓	✗	72.76	21.58	57.29
✗	✓	✓	73.71	16.11	60.03
✓	✗	✗	75.10	15.66	64.19
✓	✗	✓	76.94	12.89	65.22
✓	✓	✗	74.25	14.11	58.57
✓	✓	✓	**79.23**	**6.89**	**69.57**

6 Analysis of Representation Compensation

To verify the effectiveness of the proposed compensation method, we evaluate representation compensation from two perspectives: 1) *whether the retrieved corrections recover historical features*, and 2) *whether such a recovery improves predictions*?

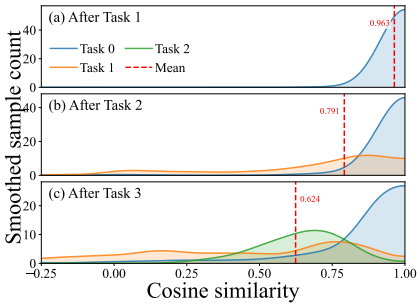


Figure 3: Cosine similarity between predicted and actual drifts.

Drift Estimation. To verify the quality of drift predicted by DDM, we statistically analyze the cosine similarity between predict drift and actual drift in each task, denoted as $\cos\left\langle \mathbf{M}\mathbf{k}^t, \mathbf{z}^{t-1} - \mathbf{z}^t \right\rangle$. The results on the AVE dataset are shown in Figure 3. We see that the predicted drift consistently exhibits strong directional alignment with the actual drift across different incremental stages, which demonstrate that DDM learns meaningful drift directions rather than arbitrary residual transformations. Although the alignment decreases with more tasks, the positive correlation remains, indicating that DDM captures meaningful drift directions for compensation.

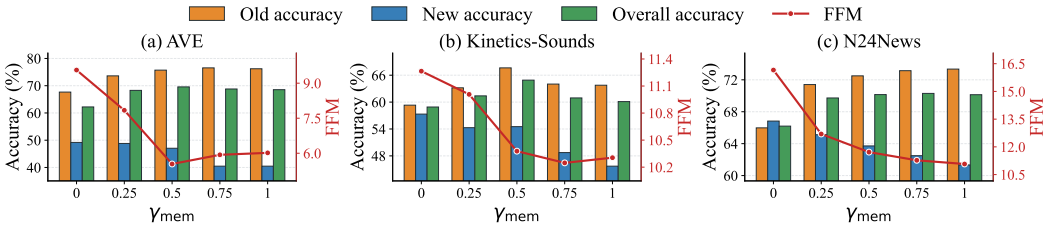


Figure 4: Performance on various compensation strength γ_{mem} across three datasets.

Performance Improvement. To investigate how the compensation predicted by DDM impacts performance, we conduct experiments under different compensation strength γ_{mem}. As shown in Figure 4, all three datasets, moderate compensation improves old-class accuracy and reduces forgetting relative to $\gamma_{\text{mem}} = 0$.However, stronger compensation can degrade new-class accuracy and offset the benefits of historical knowledge retention. Thus, we set the default value of γ_{mem} to 0.5, which provides a favorable compromise across all the datasets.

7 Conclusion

In this paper, we presented DriftMem, a novel multimodal continual learning framework that learns to compensate for representation drift through associative memory. DriftMem integrates Differentiable Delta Memory (DDM), which learns associations between semantic keys and drift residuals through differentiable delta-rule updates; Attention-guided Gradient Recovery (AGR), which stabilizes evolving cross-modal interactions; and Class-structured Feature Calibration (CFC), which reduces inter-class interference during memory updates. Extensive experiments on four datasets under the class-incremental MCL setting demonstrated that the propsoed DriftMem consistently outperforms ten competitive baselines, while ablation studies validated the contribution of each component. These results highlight the potential of explicitly modeling and compensating for representation drift as an effective alternative to constraining representation evolution in multimodal continual learning.

Limitations. Despite its effectiveness, DriftMem has several limitations. First, DDM maintains a dense memory matrix whose storage and computational costs scale with the representation dimension. Second, learning reliable drift corrections depends on the quality and coverage of replay examples. A limited or imbalanced replay buffer may provide insufficient historical references, reducing compensation accuracy for underrepresented classes.

AI USE STATEMENT

In this work, we used generative AI tools for language-related assistance, including proofreading, improving readability, and polishing the manuscript. We also used generative AI tools for limited programming assistance, such as debugging and improving code readability. We have not used generative AI tools for generating research results, conducting experiments, or making scientific claims.

All AI-assisted outputs were carefully reviewed and verified by the authors. In particular, all mathematical formulations, methodological designs, experimental settings, implementation details, and conclusions were independently developed, validated, and confirmed by the authors. AI-generated suggestions were treated only as writing or programming assistance, and the authors take full responsibility for the final content of this work, including all text, claims, and artifacts.

ETHICS STATEMENT

This work does not involve human subjects, private data collection, or any intervention on individuals. We conduct experiments on publicly available multi-modal datasets that are widely used for research purposes. The datasets are used solely for evaluating machine learning algorithms under the class-incremental learning setting, following their original usage guidelines and licenses.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we provide detailed descriptions of the proposed method, training objectives, and implementation details in the main paper and Appendix. Additional experimental settings, ablation studies, and implementation details are included in the supplementary materials. The authors will provide the source code and configuration files in supplementary material to support future reproduction.

REFERENCES

Hongjoon Ahn, Jihwan Kwak, Subin Lim, Hyeonsu Bang, Hyojun Kim, and Taesup Moon. Ss-il: Separated softmax for incremental learning. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 824–833. IEEE, 2021.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736, 2022.

Relja Arandjelovic and Andrew Zisserman. Look, listen and learn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 609–617. IEEE, 2017.

Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

Lucas Caccia, Rahaf Aljundi, Nader Asadi, Tinne Tuytelaars, Joelle Pineau, and Eugene Belilovsky. New insights on reducing abrupt representation change in online continual learning. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=N8MaByOzUfb.

Shaoxu Cheng, Chiyuan He, Kailong Chen, Linfeng Xu, Hongliang Li, Fanman Meng, and Qingbo Wu. Vision-sensor attention based continual multimodal egocentric activity recognition. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6300–6304. IEEE, 2024.

Jason R Climer, Heydar Davoudi, Jun Young Oh, and Daniel A Dombeck. Hippocampal representations drift in stable multisensory environments. *Nature*, 645(8080):457–465, 2025.

MohammadReza Davari, Nader Asadi, Sudhir Mudur, Rahaf Aljundi, and Eugene Belilovsky. Probing representation forgetting in supervised and unsupervised continual learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16691–16700. IEEE, 2022.

Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Aleš Leonardis, Gregory Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3366–3385, 2021.

Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind one embedding space to bind them all. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15180–15190. IEEE, 2023.

Alex Gomez-Villa, Dipam Goswami, Kai Wang, Andrew D Bagdanov, Bartlomiej Twardowski, and Joost Van De Weijer. Exemplar-free continual representation learning via learnable drift compensation. In *European Conference on Computer Vision*, pp. 473–490. Springer, 2024.

Dipam Goswami, Albin Soutif-Cormerais, Yuyang Liu, Sandesh Kamath, Bart Twardowski, Joost Van De Weijer, et al. Resurrecting old classes with new data for exemplar-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28525–28534, 2024.

Stephen José Hanson. A stochastic version of the delta rule. *Physica D: Nonlinear Phenomena*, 42 (1-3):265–272, 1990.

Ali Hatamizadeh, Yejin Choi, and Jan Kautz. Gated deltanet-2: Decoupling erase and write in linear attention. *arXiv preprint arXiv:2605.22791*, 2026.

Chiyuan He, Shaoxu Cheng, Zihuan Qiu, Linfeng Xu, Fanman Meng, Qingbo Wu, and Hongliang Li. Continual egocentric activity recognition with foreseeable-generalized visual–imu representations. *IEEE Sensors Journal*, 24(8):12934–12945, 2024.

Jinghan He, Haiyun Guo, Kuan Zhu, Ming Tang, and Jinqiao Wang. Continual instruction tuning for large multimodal models. *IEEE Transactions on Image Processing*, 2026.

Run He, Di Fang, Yicheng Xu, Yawen Cui, Ming Li, Cen Chen, Ziqian Zeng, and Huiping Zhuang. Semantic shift estimation via dual-projection and classifier reconstruction for exemplar-free class-incremental learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 22392–22406. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/he25d.html.

John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

Po-Yao Huang, Hu Xu, Juncheng Li, Alexei Baevski, Michael Auli, Wojciech Galuba, Florian Metze, and Christoph Feichtenhofer. Masked autoencoders that listen. *Advances in Neural Information Processing Systems*, 35:28708–28720, 2022.

Minsoo Kang, Jaeyoo Park, and Bohyung Han. Class-incremental learning by knowledge distillation with adaptive feature consolidation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16050–16059. IEEE, 2022.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020.

Joonkyu Kim, Yejin Kim, and Jy-yong Sohn. Measuring representational shifts in continual learning: A linear transformation perspective. *arXiv preprint arXiv:2505.20970*, 2025.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. doi: 10.1073/pnas.1611835114. URL https://www.pnas.org/doi/abs/10.1073/pnas.1611835114.

Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in Neural Information Processing Systems*, 34:9694–9705, 2021.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2017.

Yun Luo, Zhen Yang, Xuefeng Bai, Fandong Meng, Jie Zhou, and Yue Zhang. Investigating forgetting in pre-trained representations through continual learning. *arXiv preprint arXiv:2305.05968*, 2023.

Shentong Mo, Weiguo Pian, and Yapeng Tian. Class-incremental grouping network for continual audio-visual learning. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7754–7764. IEEE, 2023.

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural networks*, 113:54–71, 2019.

Weiguo Pian, Shentong Mo, Yunhui Guo, and Yapeng Tian. Audio-visual class-incremental learning. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 7765–7777. IEEE, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PmLR, 2021.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. Icarl: Incremental classifier and representation learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5533–5542. IEEE, 2017.

Fahad Sarfraz, Bahram Zonooz, and Elahe Arani. Beyond unimodal learning: The importance of integrating multiple modalities for lifelong learning. *arXiv preprint arXiv:2405.02766*, 2024.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. Linear transformers are secretly fast weight programmers. In *International Conference on Machine Learning*, pp. 9355–9366. PMLR, 2021.

Yapeng Tian, Jing Shi, Bochen Li, Zhiyao Duan, and Chenliang Xu. Audio-visual event localization in unconstrained videos. In *European Conference on Computer Vision*, pp. 252–268. Springer, 2018.

Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *Advances in Neural Information Processing Systems*, 35:10078–10093, 2022.

Hanxin Wang, Shuchang Zhou, Qingbo Wu, Hongliang Li, Fanman Meng, Linfeng Xu, and Heqian Qiu. Confusion mixup regularized multimodal fusion network for continual egocentric activity recognition. In *2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pp. 3552–3561. IEEE, 2023.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8):5362–5383, 2024.

Xin Wang, Devinder Kumar, Nicolas Thome, Matthieu Cord, and Frederic Precioso. Recipe recognition with large multimodal food dataset. In *2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pp. 1–6. IEEE, 2015.

Zhen Wang, Xu Shan, Xiangxie Zhang, and Jie Yang. N24news: A new dataset for multimodal news classification. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pp. 6768–6775, 2022a.

Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *European Conference on Computer Vision*, pp. 631–648. Springer, 2022b.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 139–149. IEEE, 2022c.

Songlin Yang, Bailin Wang, Yu Zhang, Yikang Shen, and Yoon Kim. Parallelizing linear transformers with the delta rule over sequence length. *Advances in Neural Information Processing Systems*, 37:115491–115522, 2024.

Lu Yu, Bartlomiej Twardowski, Xialei Liu, Luis Herranz, Kai Wang, Yongmei Cheng, Shangling Jui, and Joost van de Weijer. Semantic drift compensation for class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6982–6991, 2020.

Xianghu Yue, Xueyi Zhang, Yiming Chen, Chengwei Zhang, Mingrui Lao, Huiping Zhuang, Xinyuan Qian, and Haizhou Li. Mmal: Multi-modal analytic learning for exemplar-free audio-visual class incremental tasks. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 2428–2437, 2024.

Xi Zhang, Feifei Zhang, and Changsheng Xu. Vqacl: A novel visual question answering continual learning setting. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19102–19112. IEEE, 2023.

Da-Wei Zhou, Qi-Wei Wang, Zhi-Hong Qi, Han-Jia Ye, De-Chuan Zhan, and Ziwei Liu. Class-incremental learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12), 2024.

A Theoretical Analysis

A.1 Analysis of Delta Memory Update

For fixed keys and targets in one mini-batch, denoted as $N_e = |\mathcal{B}_e|$ and $N_c = |\mathcal{B}_c|$, we have

$$\mathcal{J}(\mathbf{M}) = \frac{1}{2N_e} \sum_{i \in \mathcal{B}_e} \|\mathbf{M}\mathbf{k}_i^t - \Delta\mathbf{z}_i^t\|_2^2 + \frac{\lambda_c}{2N_c} \sum_{j \in \mathcal{B}_c} \|\mathbf{M}\mathbf{k}_j^t\|_2^2. \tag{19}$$

Since $\nabla_{\mathbf{M}} \frac{1}{2}\|\mathbf{M}\mathbf{k} - \Delta\mathbf{z}\|_2^2 = (\mathbf{M}\mathbf{k} - \Delta\mathbf{z})\mathbf{k}^\top$, it is easy to derive

$$\nabla_{\mathbf{M}} \mathcal{J}(\mathbf{M}) = \frac{1}{N_e} \sum_{i \in \mathcal{B}_e} (\mathbf{M}\mathbf{k}_i^t - \Delta\mathbf{z}_i^t)(\mathbf{k}_i^t)^\top + \frac{\lambda_c}{N_c} \sum_{j \in \mathcal{B}_c} (\mathbf{M}\mathbf{k}_j^t)(\mathbf{k}_j^t)^\top, \tag{20}$$

By utilizing gradient descent $\mathbf{M}^+ = \mathbf{M} - \lambda_e \nabla_{\mathbf{M}} \mathcal{J}$, we prove Equation 9. Thus, the update jointly fits historical residuals and suppresses responses to current-task keys.

Theorem 1 (Conditional convergence of residual regression)**.** *Given the keys and targets in Equation 8, it has the follows*

$$\mathbf{A} = \frac{1}{N_e} \sum_i \mathbf{k}_i \mathbf{k}_i^\top + \frac{\lambda}{N_c} \sum_j \mathbf{k}_j \mathbf{k}_j^\top, \qquad \mathbf{B} = \frac{1}{N_e} \sum_i \Delta\mathbf{z}_i \mathbf{k}_i^\top. \tag{21}$$

Suppose $L = \lambda_{\max}(\mathbf{A}) > 0$ and $0 < \eta < 2/L$. Updating Equation 9 would converge to a minimizer of $\mathcal{J}$, and the objective is non-increasing. If $\mathbf{A} \succ 0$, the minimizer is $\mathbf{M}^\star = \mathbf{B}\mathbf{A}^{-1}$ and

$$\|\mathbf{M}_s - \mathbf{M}^\star\|_F \le q^s \|\mathbf{M}_0 - \mathbf{M}^\star\|_F, \quad q = \max_{\nu \in \mathrm{spec}(\mathbf{A})} |1 - \eta\nu| < 1. \tag{22}$$

For singular $\mathbf{A}$, convergence holds on the span of the active keys, while the initial component on its orthogonal complement is unchanged.

Proof. Expanding Equation 8 yields

$$\begin{aligned}
\mathcal{J}(\mathbf{M}) &= \frac{1}{2N_e} \sum_{i \in \mathcal{B}_e} (\mathbf{M}\mathbf{k}_i - \Delta\mathbf{z}_i)^\top (\mathbf{M}\mathbf{k}_i - \Delta\mathbf{z}_i) + \frac{\lambda_c}{2N_c} \sum_{j \in \mathcal{B}_c} (\mathbf{M}\mathbf{k}_j)^\top (\mathbf{M}\mathbf{k}_j) \\
&= \frac{1}{2N_e} \sum_{i \in \mathcal{B}_e} \left(\mathbf{k}_i^\top \mathbf{M}^\top \mathbf{M}\mathbf{k}_i - 2\Delta\mathbf{z}_i^\top \mathbf{M}\mathbf{k}_i + \Delta\mathbf{z}_i^\top \Delta\mathbf{z}_i \right) + \frac{\lambda_c}{2N_c} \sum_{j \in \mathcal{B}_c} \mathbf{k}_j^\top \mathbf{M}^\top \mathbf{M}\mathbf{k}_j \\
&= \frac{1}{2N_e} \sum_{i \in \mathcal{B}_e} \left(\mathrm{tr}(\mathbf{M}\mathbf{k}_i \mathbf{k}_i^\top \mathbf{M}^\top) - 2\,\mathrm{tr}(\Delta\mathbf{z}_i^\top \mathbf{M}\mathbf{k}_i) + \Delta\mathbf{z}_i^\top \Delta\mathbf{z}_i \right) + \frac{\lambda_c}{2N_c} \sum_{j \in \mathcal{B}_c} \mathrm{tr}(\mathbf{M}\mathbf{k}_j \mathbf{k}_j^\top \mathbf{M}^\top) \\
&= \frac{1}{2}\,\mathrm{tr}(\mathbf{M}\mathbf{A}\mathbf{M}^\top) - \mathrm{tr}(\mathbf{M}\mathbf{B}^\top) + \mathrm{const}, \qquad \nabla\mathcal{J}(\mathbf{M}) = \mathbf{M}\mathbf{A} - \mathbf{B}.
\end{aligned} \tag{23}$$

The matrix $\mathbf{A}$ is symmetric positive semidefinite. For any perturbation $\mathbf{U}$, $\|\mathbf{U}\mathbf{A}\|_F \le L\|\mathbf{U}\|_F$, so the gradient is L-Lipschitz. The descent inequality gives

$$\mathcal{J}(\mathbf{M} - \eta\nabla\mathcal{J}(\mathbf{M})) \le \mathcal{J}(\mathbf{M}) - \eta(1 - \eta L/2)\|\nabla\mathcal{J}(\mathbf{M})\|_F^2. \tag{24}$$

For completeness, the inequality follows by expanding the quadratic: $\mathcal{J}(\mathbf{M} + \mathbf{U}) = \mathcal{J}(\mathbf{M}) + \langle \nabla\mathcal{J}(\mathbf{M}), \mathbf{U} \rangle_F + \frac{1}{2}\,\mathrm{tr}(\mathbf{U}\mathbf{A}\mathbf{U}^\top)$, and bounding the last term by $L\|\mathbf{U}\|_F^2/2$.

If $\mathbf{A} \succ 0$, its stationary point is $\mathbf{M}^\star = \mathbf{B}\mathbf{A}^{-1}$. Subtracting its normal equation from the update gives

$$\mathbf{M}_{s+1} - \mathbf{M}^\star = (\mathbf{M}_s - \mathbf{M}^\star)(\mathbf{I} - \eta\mathbf{A}). \tag{25}$$

The eigenvalues of $\mathbf{I} - \eta\mathbf{A}$ lie strictly between -1 and 1, proving Equation 22 by iteration.

For singular $\mathbf{A}$, let $\mathbf{P} = \mathbf{A}\mathbf{A}^\dagger$ be the orthogonal projector onto its range. If $\mathbf{u} \in \ker \mathbf{A}$, then

$$0 = \mathbf{u}^\top \mathbf{A}\mathbf{u} = \frac{1}{N_e} \sum_i (\mathbf{k}_i^\top \mathbf{u})^2 + \frac{\lambda}{N_c} \sum_j (\mathbf{k}_j^\top \mathbf{u})^2. \tag{26}$$

In particular, all replay terms vanish, so $\mathbf{Bu} = 0$ and $\mathbf{B} = \mathbf{BP}$. Thus

$$\mathbf{M}_\infty = \underbrace{\mathbf{B}\mathbf{A}^\dagger}_{\text{key spans}} + \underbrace{\mathbf{M}_0(\mathbf{I}-\mathbf{P})}_{\text{unseen directions}} \tag{27}$$

is a minimizer. The same error recurrence contracts on the positive eigenspaces and leaves the nullspace component unchanged, proving convergence. □

A.2 Analysis of representation compensation

Theorem 2 (Residual compensation reduces representation drift)**.** *For a historical sample, let the representation drift be*

$$\Delta\mathbf{z}^t = \hat{\mathbf{z}}^{t-1} - \mathbf{z}^t,$$

and let the estimated drift from the delta memory be

$$\mathbf{r} = \mathbf{M}\mathbf{k}^t = \Delta\mathbf{z}^t + \mathbf{e},$$

where $\mathbf{e}$ *denotes the residual prediction error. After compensation with strength* $\gamma \geq 0$*,*

$$\hat{\mathbf{z}}^t = \mathbf{z}^t + \gamma\mathbf{r},$$

the remaining representation discrepancy satisfies, for $0 \leq \gamma \leq 1$ *and* $\|\mathbf{e}\|_2 \leq \varepsilon$*,*

$$\left\|\hat{\mathbf{z}}^t - \hat{\mathbf{z}}^{t-1}\right\|_2 \leq (1-\gamma)\left\|\Delta\mathbf{z}^t\right\|_2 + \gamma\varepsilon. \tag{28}$$

Moreover, assuming $\mathbf{r} \neq 0$ *and* $\gamma > 0$*, compensation strictly reduces the representation drift if and only if*

$$\left\langle \Delta\mathbf{z}^t, \mathbf{r}\right\rangle > 0, \qquad 0 < \gamma < \frac{2\left\langle \Delta\mathbf{z}^t, \mathbf{r}\right\rangle}{\|\mathbf{r}\|_2^2}. \tag{29}$$

Proof. Since

$$\mathbf{r} = \Delta\mathbf{z}^t + \mathbf{e},$$

the compensated representation discrepancy can be written as

$$\begin{aligned}\hat{\mathbf{z}}^t - \hat{\mathbf{z}}^{t-1} &= \mathbf{z}^t + \gamma\mathbf{r} - \hat{\mathbf{z}}^{t-1} \\ &= -\Delta\mathbf{z}^t + \gamma(\Delta\mathbf{z}^t + \mathbf{e}) \\ &= (\gamma-1)\Delta\mathbf{z}^t + \gamma\mathbf{e}.\end{aligned} \tag{30}$$

For $0 \leq \gamma \leq 1$, applying the triangle inequality gives

$$\begin{aligned}\left\|\hat{\mathbf{z}}^t - \hat{\mathbf{z}}^{t-1}\right\|_2 &\leq (1-\gamma)\left\|\Delta\mathbf{z}^t\right\|_2 + \gamma\|\mathbf{e}\|_2 \\ &\leq (1-\gamma)\left\|\Delta\mathbf{z}^t\right\|_2 + \gamma\varepsilon.\end{aligned} \tag{31}$$

To derive the strict reduction condition, compare the compensated distance with the original drift:

$$\begin{aligned}&\left\|\hat{\mathbf{z}}^t - \hat{\mathbf{z}}^{t-1}\right\|_2^2 - \left\|\Delta\mathbf{z}^t\right\|_2^2 \\ &= \left\|\gamma\mathbf{r} - \Delta\mathbf{z}^t\right\|_2^2 - \left\|\Delta\mathbf{z}^t\right\|_2^2 \\ &= \gamma^2\|\mathbf{r}\|_2^2 - 2\gamma\left\langle \Delta\mathbf{z}^t, \mathbf{r}\right\rangle.\end{aligned} \tag{32}$$

For $\gamma > 0$ and $\mathbf{r} \neq 0$, the above quantity is negative if and only if

$$\gamma\|\mathbf{r}\|_2^2 < 2\left\langle \Delta\mathbf{z}^t, \mathbf{r}\right\rangle,$$

which requires

$$\left\langle \Delta\mathbf{z}^t, \mathbf{r}\right\rangle > 0$$

and yields

$$0 < \gamma < \frac{2\left\langle \Delta\mathbf{z}^t, \mathbf{r}\right\rangle}{\|\mathbf{r}\|_2^2}.$$

Therefore, the stated condition is both necessary and sufficient. □

A.3 Analysis of Attention-guided Gradient Recovery

We here provide an analysis of Equation 13. Considering incremental step t, we have the loss function for the joint dataset $\mathcal{D}_{1:t} = \bigcup_{i=1}^{t} \mathcal{D}_i$

$$\mathcal{L}(\phi; \mathcal{D}_{1:t}) = \mathcal{L}(\phi; \mathcal{D}_{1:t-1}) + \mathcal{L}(\phi; \mathcal{D}_t). \tag{33}$$

By taking gradient of ϕ, we have

$$\nabla_\phi \mathcal{L}(\phi; \mathcal{D}_{1:t}) = \nabla_\phi \mathcal{L}(\phi; \mathcal{D}_{1:t-1}) + \nabla_\phi \mathcal{L}(\phi; \mathcal{D}_t). \tag{34}$$

However, $\mathcal{D}_{1:t-1}$ remain unknown in the setting of CIL, thereby $\nabla_\phi \mathcal{L}(\phi; \bigcup_{i=1}^{t-1} \mathcal{D}_i)$ can not be directly calculated. We approximate this term. Suppose $\phi^\star$ is the learned optimal parameters from previous tas, we have

$$\phi^\star = \arg\min_{\phi} \mathcal{L}(\phi; \mathcal{D}_{1:t-1}), \tag{35}$$

and the first-order optimality condition gives

$$\nabla_\phi \mathcal{L}(\phi^\star) = 0. \tag{36}$$

Then, we perform a first-order Taylor expansion on the gradient function

$$\mathbf{g}(\phi) \approx \mathbf{g}(\phi^\star) + \mathbf{J}_\mathbf{g}(\phi^\star)(\phi - \phi^\star), \tag{37}$$

where $\mathbf{g}(\phi) = \nabla_\phi \mathcal{L}(\phi)$ and $\mathbf{J}_\mathbf{g}(\phi^\star)$ denotes the Jacobian matrix. Since the Jacobian of the gradient is the Hessian matrix of the loss, we obtain

$$\mathbf{J}_\mathbf{g}(\phi^\star) = \nabla^2_\phi \mathcal{L}(\phi^\star) = \mathbf{H}(\phi^\star), \tag{38}$$

By the optimality condition $\nabla_\theta \mathcal{L}(\theta^\star) = 0$, the above equation reduces to

$$\nabla_\theta \mathcal{L}(\theta) \approx \mathbf{H}(\theta^\star)(\theta - \theta^\star). \tag{39}$$

However, directly estimating $\mathbf{H}$ is impractical for large neural networks. Thus, we use parameter evolution $\phi^\star - \phi$ as an efficient approximation of the historical update direction, while acknowledging the gap introduced by neglecting curvature information. In practice, we use ϕ_{t-1} as an approximate substitute for $\phi^\star$, then we derive $\mathbf{d}_\phi = \phi_{t-1} - \phi_t$ and our guidance gradient $\mathbf{r}_\phi$ in Equation 13. Furthermore, we replay the exemplar data $\mathcal{E}_t$ to obtain more accurate gradient estimates. The practical gradient expression is shown follows

$$\nabla_\phi \mathcal{L}(\phi; \mathcal{D}_{1:t}) = \underbrace{\mathbf{r}_\phi + \nabla_\phi \mathcal{L}(\phi; \mathcal{E}_t)}_{\text{historical estimation}} + \underbrace{\nabla_\phi \mathcal{L}(\phi; \mathcal{D}_t)}_{\text{current gradient}} \tag{40}$$

A.4 Analysis of Interference and Class Geometry

Proposition 3 (Prototype correlation bounds sample interference)**.** *Assume that all keys and class prototypes are nonzero and ℓ_2-normalized. For two samples i and j from distinct classes c and c', respectively, suppose*

$$\|\mathbf{k}_i - \mathbf{v}_c\|_2 \le \varepsilon_i, \qquad \|\mathbf{k}_j - \mathbf{v}_{c'}\|_2 \le \varepsilon_j.$$

Given an error-driven memory update

$$\mathbf{M}^+ = \mathbf{M} + \eta \mathbf{e}_i \mathbf{k}_i^\top,$$

the interference introduced on key $\mathbf{k}_j$ satisfies

$$\left\|(\mathbf{M}^+ - \mathbf{M})\mathbf{k}_j\right\|_2 \le \eta \|\mathbf{e}_i\|_2 \left(|\mathbf{v}_c^\top \mathbf{v}_{c'}| + \varepsilon_i + \varepsilon_j\right). \tag{41}$$

Moreover, for uniformly sampled ordered pairs of distinct classes,

$$\mathbb{E}_{c \neq c'} |\mathbf{v}_c^\top \mathbf{v}_{c'}| \le \sqrt{\mathcal{L}_{\text{orth}}},$$

where $\mathcal{L}_{\text{orth}}$ denotes the prototype orthogonality loss.

Proof. The effect of updating the memory with sample i on another key $\mathbf{k}_j$ is

$$(\mathbf{M}^+ - \mathbf{M})\mathbf{k}_j = \eta \mathbf{e}_i \mathbf{k}_i^\top \mathbf{k}_j. \tag{42}$$

Therefore,

$$\left\|(\mathbf{M}^+ - \mathbf{M})\mathbf{k}_j\right\|_2 = \eta \|\mathbf{e}_i\|_2 |\mathbf{k}_i^\top \mathbf{k}_j|. \tag{43}$$

Next, we relate sample-level correlation to prototype correlation. We have

$$\mathbf{k}_i^\top \mathbf{k}_j - \mathbf{v}_c^\top \mathbf{v}_{c'} = (\mathbf{k}_i - \mathbf{v}_c)^\top \mathbf{k}_j + \mathbf{v}_c^\top (\mathbf{k}_j - \mathbf{v}_{c'}). \tag{44}$$

Taking absolute values and applying the triangle inequality gives

$$|\mathbf{k}_i^\top \mathbf{k}_j| \le |\mathbf{v}_c^\top \mathbf{v}_{c'}| + |(\mathbf{k}_i - \mathbf{v}_c)^\top \mathbf{k}_j| + |\mathbf{v}_c^\top (\mathbf{k}_j - \mathbf{v}_{c'})|. \tag{45}$$

Since all keys and prototypes are unit-normalized, the Cauchy–Schwarz inequality yields

$$\begin{aligned} |(\mathbf{k}_i - \mathbf{v}_c)^\top \mathbf{k}_j| &\le \|\mathbf{k}_i - \mathbf{v}_c\|_2 \|\mathbf{k}_j\|_2 \le \varepsilon_i, \\ |\mathbf{v}_c^\top (\mathbf{k}_j - \mathbf{v}_{c'})| &\le \|\mathbf{v}_c\|_2 \|\mathbf{k}_j - \mathbf{v}_{c'}\|_2 \le \varepsilon_j. \end{aligned} \tag{46}$$

Combining the above inequalities gives

$$|\mathbf{k}_i^\top \mathbf{k}_j| \le |\mathbf{v}_c^\top \mathbf{v}_{c'}| + \varepsilon_i + \varepsilon_j. \tag{47}$$

Substituting this result into the interference expression completes the first claim.

Finally, by Cauchy–Schwarz,

$$\begin{aligned} \mathbb{E}_{c \neq c'} |\mathbf{v}_c^\top \mathbf{v}_{c'}| &\le \sqrt{\mathbb{E}_{c \neq c'} (\mathbf{v}_c^\top \mathbf{v}_{c'})^2} \\ &= \sqrt{\mathcal{L}_{\mathrm{orth}}}, \end{aligned} \tag{48}$$

which completes the proof. □

Discussion. The proposition shows that reducing prototype correlations can alleviate memory interference. However, prototype orthogonality alone does not guarantee interference-free retrieval, since the sample-to-prototype deviation terms ε_i and ε_j are not explicitly controlled by the orthogonality loss. Moreover, exact orthogonality is impossible when the number of nonzero class prototypes exceeds the feature dimension. Therefore, CFC aims to encourage favorable class geometry rather than enforcing perfect interference elimination.

B Implementation Details

B.1 Training Objective Details

Let $\boldsymbol{\ell}_i^t = g^t(\hat{\mathbf{z}}_i^t)$ denote the logits, and let $\mathcal{C}_{<t} = \bigcup_{s<t} \mathcal{C}_s$. Following the cross-entropy paradigm in (Caccia et al., 2022), the implementation uses

$$\mathcal{L}_{\mathrm{cls}} = \frac{1}{N_c + N_e} \left[\sum_{i \in \mathcal{B}_c} \mathcal{L}_{\mathrm{CE}}^{\mathcal{C}_t}(\boldsymbol{\ell}_i^t, y_i) + \sum_{i \in \mathcal{B}_e} \mathcal{L}_{\mathrm{CE}}^{\mathcal{C}_{<t}}(\boldsymbol{\ell}_i^t, y_i) \right]. \tag{49}$$

Classification is therefore normalized separately over new and old classes during incremental training; inference compares logits over all seen classes. For the equal-size task blocks used by the implementation, task-wise logit distillation is

$$\begin{aligned} \mathbf{p}_i^{s,t-1} &= \mathrm{softmax}(\boldsymbol{\ell}_i^{t-1}|_{\mathcal{C}_s}/T), \qquad \mathbf{p}_i^{s,t} = \mathrm{softmax}(\boldsymbol{\ell}_i^{t}|_{\mathcal{C}_s}/T), \\ \mathcal{L}_{\mathrm{KD}} &= \frac{T^2}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \sum_{s<t} \mathrm{KL}(\mathbf{p}_i^{s,t-1} \| \mathbf{p}_i^{s,t}). \end{aligned} \tag{50}$$

The frozen teacher evaluates both current and replay inputs. The initial task follows the base learner's initialization procedure with zero delta memory. Equation 9 describes subsequent tasks.

Table 3: Additional complexity of the DriftMem components. Storage excludes the shared historical model, replay buffer, and ordinary network activations. Training costs are per iteration, and inference costs are per sample.

Component	Training time	Additional storage	Inference time
DDM	$\mathcal{O}(Bd^2)$	$\mathcal{O}(d^2 + Bd)$	$\mathcal{O}(d^2)$
AGR	$\mathcal{O}(A_e + P_{\text{att}})$	None	None
CFC	$\mathcal{O}(B^2 d)$	None	None

B.2 COMPUTATIONAL AND MEMORY COMPLEXITY

We analyze the overhead introduced by DriftMem in Table 3. Let $B_c = |\mathcal{B}_c|$ and $B_e = |\mathcal{B}_e|$ denote the current and replay batch sizes, respectively, with $B = B_c + B_e$. Let d be the fused feature dimension, C_B the number of distinct classes in the combined batch, and P_{att} the number of parameters subject to AGR. We denote by A_e the total number of scalar attention entries used to measure drift on the replay batch. The analysis assumes dense matrix operations and excludes data loading and optional diagnostic computations.

Learnable Delta Memory. DDM maintains a dense matrix $\mathbf{M} \in \mathbb{R}^{d\times d}$. For a batch key matrix $K \in \mathbb{R}^{B\times d}$, memory retrieval computes $\mathbf{M}K^\top$, requiring $\mathcal{O}(Bd^2)$ time. The memory update evaluates residual prediction errors and accumulates their outer products with the keys. The replay and current-task terms together cost $\mathcal{O}((B_e + B_c)d^2) = \mathcal{O}(Bd^2)$. Thus, the overall training-time cost of DDM remains $\mathcal{O}(Bd^2)$ per iteration. Its persistent state requires $\mathcal{O}(d^2)$ memory. In our practical setting with $d = 768$, a single FP32 memory matrix requires only 2.25 MiB of storage.

Attention-guided Gradient Recovery. AGR computes normalized Jensen–Shannon divergences between current and historical replay attention maps. The discrepancy computation costs $\mathcal{O}(A_e)$ time. Constructing parameter displacements, evaluating their norms, and adding bounded restoring gradients cost $\mathcal{O}(P_{\text{att}})$ time. The total additional computation is therefore $\mathcal{O}(A_e+P_{\text{att}})$ per iteration.

Class-structured Feature Calibration. Constructing class prototypes from batch keys requires $\mathcal{O}(Bd)$ time. Computing all pairwise prototype similarities for $\mathcal{L}_{\text{orth}}$ costs $\mathcal{O}(C_B^2 d)$ time and $\mathcal{O}(C_B^2)$ similarity storage. The supervised contrastive objective computes pairwise similarities between batch keys, requiring $\mathcal{O}(B^2 d)$ time. Consequently, CFC requires $\mathcal{O}(Bd + C_B^2 d + B^2 d) = \mathcal{O}(B^2 d)$ time, since $C_B \le B$. Its temporary storage is $\mathcal{O}(Bd + B^2)$, and it introduces no additional trainable parameters.

Overall training complexity. Let $\mathcal{C}_{\text{student}}(B)$ denote the cost of the current representation network, classifier, and their backward pass, excluding memory retrieval. Let $\mathcal{C}_{\text{teacher}}(B)$ denote the forward-pass cost of the frozen historical representation network and classifier, also excluding memory retrieval. For incremental tasks, the per-iteration training cost is

$$\mathcal{O}\left(\mathcal{C}_{\text{student}}(B) + \mathcal{C}_{\text{teacher}}(B) + Bd^2 + B^2 d + A_e + P_{\text{att}}\right). \tag{51}$$

Inference complexity and storage. AGR and CFC are used only during training. For a single test sample, DDM adds $\mathcal{O}(d^2)$ computation and $\mathcal{O}(d^2)$ persistent storage to the underlying model. No historical model, replay buffer, or memory update is required at inference. As a result, the overall per-sample inference cost is

$$\mathcal{O}\left(\mathcal{C}_{\text{repr}}(1)\right), \tag{52}$$

where $\mathcal{C}_{\text{repr}}(1)$ denotes the cost of obtaining the uncompensated fused feature and key.

B.3 SUMMARY OF COMPARED BASELINES

To demonstrate the effectiveness of our method, we compare it with ten state-of-the-art approaches on the video–audio benchmarks AVE and Kinetics-Sounds (KS), and the image–text benchmarks N24News and Food101. All methods access offline-extracted multimodal features through a unified

dataset interface and are evaluated using the same data splits and class order under task-agnostic class-incremental inference. We organize the baselines into the following three categories.

Vanilla CL. This group includes LwF (Li & Hoiem, 2017), iCaRL (Rebuffi et al., 2017), SS-IL (Ahn et al., 2021), and AFC (Kang et al., 2022), which represent conventional continual-learning strategies based on knowledge distillation, exemplar replay, bias mitigation, and feature consolidation, respectively. To support multimodal inputs, we replace their unimodal representations with attention-based fused representations while preserving their original learning objectives. For replay-based methods, the memory stores paired multimodal features, and exemplar selection is performed in the fused feature space. In particular, AFC consolidates both modality-specific and attention-pooled representations, allowing its importance-weighted constraints to preserve multimodal knowledge.

CL with Representation Compensation. SDC (Yu et al., 2020), ADC (Goswami et al., 2024), and DPCR (He et al., 2025) estimate representation shifts from current-task data and compensate old-class prototypes or statistics without retaining historical samples. We apply their compensation mechanisms to the shared multimodal fused representation: SDC estimates semantic drift on normalized fused embeddings, while DPCR performs dual-projection estimation and classifier reconstruction in the fused feature space. ADC requires a more specific adaptation because its original attack is conducted in the image-pixel space; with only pre-extracted features available, we move the directed attack to the fused feature space and introduce an identity-initialized projection to provide a shared attack space for the old and new models.

Multimodal CL. CMR-MFN (Wang et al., 2023), AV-CIL (Pian et al., 2023), and MMAL (Yue et al., 2024) explicitly model cross-modal interactions during continual learning. AV-CIL directly uses our unified feature interface while retaining its cross-modal instance- and class-level contrastive learning and spatial/temporal attention distillation. For CMR-MFN, we append the second-modality feature as a token to the first-modality sequence and preserve its confusion-mixup task branches for task-agnostic inference. For MMAL, we generalize its audio and visual analytical streams to the unified two-modality interface, enabling its fusion and modality-specific compensation streams to process both video–audio and image–text features.

B.4 Deployment

We deployed the proposed model and baselines using Pytorch 2.4.1 on NVIDIA RTX 4090 GPUs with 48GB of memory. We use freezing pretrained models as our anchor encoder and context encoder. For video-audio datasets, we assign video as context modality and audio as anchor modality, which are pre-processed using VideoMAE Tong et al. (2022) and AudioMAE Huang et al. (2022), respectively. For image-text datasets, we assign image as context modality and text as anchor modality. Specifically, for N24News dataset, we employ CLIP-ViT-Patch14 Radford et al. (2021) and Longformer Beltagy et al. (2020) to encode images and text, respectively. While for UMPC-Food101 dataset, we adopt the CLIP text encoder and image encoder to obtain corresponding text and image features. For all four datasets, we set default value of γ_{mem} to 0.5, τ_{con} to 0.09 and λ_c to 0.6. We also set δ_{att} and τ_{att} to 0.0001 and 0.0005 in AGR, respectively. More hyper-parameter analysis is provided in Appendix C.1.

C Addtional Experiment Results

C.1 Parameters Study

Fixed memory size $|\mathcal{E}|$. We investigate the impact of memory size on four datasets, as shown in Table 4. In general, increasing the memory size consistently improves the performance by providing more historical information for drift compensation, leading to higher CAA and lower FFM. For example, on AVE and Kinetics-Sounds, increasing the memory size from 50 to 340 and from 150 to 1000 respectively yields consistent improvements in both metrics, indicating that sufficient memory is important for modeling representation drift in audio-visual scenarios. Similarly, N24News benefits from a larger memory budget and achieves the best performance with 1000 stored samples. However, the performance gains gradually saturate when the memory size further increases. For UPMC-Food101, the performance peaks at 1200 samples, while larger memories provide marginal improvements or even slight degradation, suggesting that excessive memory expansion is unnecessary. Considering the balance between performance and memory efficiency, we adopt memory sizes

Table 4: Effect of memory size on different datasets.

(a) AVE

Memory	Ratio	CAA ↑	FFM ↓
50	1.52%	76.78	16.00
100	3.04%	77.32	13.93
150	4.55%	78.15	13.04
200	6.07%	78.10	11.24
340	10.32%	79.23	6.89
500	15.18%	79.23	7.34

(b) N24News

Memory	Ratio	CAA ↑	FFM ↓
150	0.31%	78.74	14.47
200	0.41%	78.65	13.94
500	1.02%	79.34	16.55
750	1.53%	81.11	13.24
1000	2.04%	81.97	10.32
1200	2.44%	81.66	10.92

(c) Kinetics-Sounds

Memory	Ratio	CAA ↑	FFM ↓
150	0.78%	70.51	16.13
200	1.04%	72.34	14.42
340	1.77%	72.68	12.25
500	2.60%	73.79	10.37
750	3.91%	73.99	10.12
1000	5.21%	74.50	8.05

(d) UPMC-Food101

Memory	Ratio	CAA ↑	FFM ↓
500	0.74%	91.85	4.74
750	1.10%	94.33	3.52
1000	1.47%	94.54	3.10
1200	1.77%	95.28	2.66
1500	2.21%	94.67	2.59
2000	2.94%	94.77	2.63

of 340, 500, 1000, and 1200 for AVE, Kinetics-Sounds, N24News, and UPMC-Food101, respectively, in all experiments.

Current-task response suppression coefficient λ_c. Table 5 evaluates the weight of the current-task zero-response objective. Nonzero weights improve CAA over $\lambda_c = 0$ on both datasets, with moderate weights generally providing a favorable trade-off. However, the effect on forgetting is dataset-dependent: FFM improves at selected settings on AVE, whereas it remains higher than the zero-weight baseline on N24News. Increasing λ_c further does not yield additional accuracy gains, suggesting that this objective should be balanced against historical knowledge retention. We therefore adopt $\lambda_c = 0.6$ as the default across datasets, as it achieves competitive CAA while yielding the lowest FFM among the evaluated nonzero settings on both datasets.

Table 5: Sensitivity to λ_c. Higher CAA and lower FFM indicate better performance.

Dataset	Metric	λ_c								
		0.0	0.2	0.4	0.6	0.8	1.0	1.25	2.5	5.0
AVE	CAA ↑	75.76	77.45	78.11	**79.43**	79.17	79.32	78.77	78.24	77.62
	FFM ↓	7.89	8.51	8.80	**7.03**	8.23	8.26	7.31	8.49	9.10
N24News	CAA ↑	76.45	80.82	79.99	81.36	**81.40**	80.86	80.86	80.16	79.72
	FFM ↓	**8.51**	12.93	13.77	10.62	12.18	13.73	12.55	12.59	13.36

Contrastive temperature coefficient τ_{con} in CFC. The temperature τ scales the cosine-similarity logits in the supervised contrastive objective, thereby controlling the sharpness of the similarity distribution. As shown in Figure 5, a small τ_{con} produces overly sharp distributions and may lead to unstable optimization, whereas a large τ_{con} weakens feature discrimination. Across the three datasets, We set $\tau = 0.09$ in all experiments, as it provides a favorable trade-off between overall accuracy and resistance to catastrophic forgetting.

Weight of CFC loss $\lambda_{\text{orth}}, \lambda_{\text{con}}$. Table 6 examines the sensitivity to the orthogonality loss weight λ_{orth} and the contrastive loss weight λ_{con}. On the video-audio dataset, increasing either weight generally improves the balance between representation discriminability and knowledge retention, with the two objectives exhibiting complementary effects when jointly enabled. Accordingly, we set $(\lambda_{\text{orth}}, \lambda_{\text{con}}) = (1.0, 1.0)$ for video-audio data. In contrast, the image-text dataset benefits

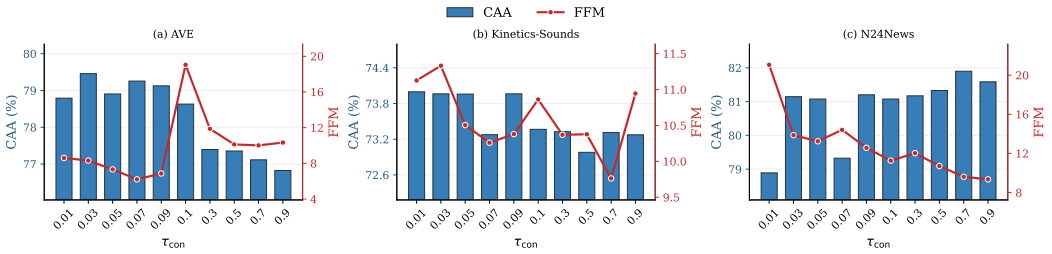


Figure 5: Sensitivity to temperature coefficient τ_{con} of contrastive learning.

Table 6: Sensitivity analysis of the orthogonality and contrastive loss weights on AVE and N24News. The best result in each column is highlighted in bold font.

Loss weight	AVE		N24News	
	CAA ↑	FFM ↓	CAA ↑	FFM ↓
λ_{orth} $(\lambda_{\text{con}} = 1.0)$				
0.0	80.13	6.12	79.17	**15.15**
0.2	81.01	**3.45**	76.16	21.25
0.4	80.76	5.64	78.70	15.79
0.6	80.88	5.88	79.19	17.26
0.8	80.98	4.99	**79.96**	17.65
1.0	**81.53**	4.36	79.03	17.06
λ_{con} $(\lambda_{\text{orth}} = 1.0)$				
0.0	76.47	10.59	78.96	16.28
0.2	80.23	7.03	**81.24**	**10.47**
0.4	80.55	6.20	80.77	16.87
0.6	79.90	5.75	79.62	17.51
0.8	80.40	**4.34**	78.76	17.88
1.0	**81.53**	4.36	79.03	17.06

from a moderate contrastive regularization strength, while an overly strong contrastive objective tends to impair the stability of previously learned representations. We therefore use $(\lambda_{\text{orth}}, \lambda_{\text{con}}) = (1.0, 0.2)$ for image-text data.

C.2 Results with Different Class Order

To investigate the robustness of our DriftMem against different class orders, we further evaluate our method and compare it with baselines under multiple class-incremental orders. Specifically, we randomly shuffle the class order according to different random seeds. We report the results under seeds 1993 and 3407 in Table 7. We observe that DriftMem consistently achieves the best CAA across different class orders on all three datasets, while maintaining the lowest FFM in most cases. The results demonstrate that the effectiveness of DriftMem does not rely on a specific task order, and explicitly modeling representation drift provides robust compensation for varying incremental learning class orders.

C.3 Experiments under Extended Task Sequences

In this section , we further conduct experiments with extende task sequences to evaluate the scalability of our method. Specifically, for the AVE dataset, we partition the classes into 7 incremental tasks, with 4 classes introduced in each task. For Kinetics-Sounds, we construct a 10-task scenario, where each task contains 3 classes. For N24News, we partition the dataset into 6 incremental tasks, with 4 classes per task. As shown in Table 8, the overall performance of all methods decreases compared with the settings of main experiment, which is expected as longer task sequences introduce more severe representation drift and accumulated forgetting. Nevertheless, DriftMem consistently maintains

Table 7: Results with two different random task orders.

Method	AVE				N24News				Food101			
	Seed 3407		Seed 1993		Seed 3407		Seed 1993		Seed 3407		Seed 1993	
	CAA ↑	FFM ↓	CAA ↑	FFM ↓	CAA ↑	FFM ↓	CAA ↑	FFM ↓	CAA ↑	FFM ↓	CAA ↑	FFM ↓
LwF	63.57	30.93	65.68	26.78	74.69	17.85	74.20	16.34	85.68	22.40	85.11	21.10
iCaRL	76.16	36.69	75.66	37.36	64.06	63.21	61.17	69.63	89.38	17.19	89.41	18.39
SSIL	73.02	11.57	71.17	11.28	78.33	13.02	77.42	**9.13**	93.80	3.43	93.85	3.32
AFC	75.19	29.95	72.52	39.99	68.33	56.74	68.35	38.62	92.46	10.61	93.27	9.18
ER-ACE	59.06	15.17	54.59	24.45	68.56	26.59	66.04	32.43	93.58	4.89	94.18	5.51
SDC-IL	55.60	64.64	56.55	67.37	48.87	65.95	46.37	54.12	59.96	55.83	61.54	63.85
ADC	52.37	40.13	55.58	18.54	67.48	11.65	66.27	19.86	70.65	13.10	77.06	10.45
DPCR	50.18	54.52	57.94	38.58	68.34	36.21	69.14	34.94	80.85	26.77	81.50	23.06
CMR-MFN	65.18	21.07	57.76	16.55	72.24	19.15	74.45	12.10	85.42	12.44	91.09	4.52
MMAL	77.30	18.10	76.55	13.35	76.48	15.38	75.38	14.01	90.20	4.24	90.70	4.29
DriftMem	**78.50**	**7.65**	**78.58**	**8.20**	**79.63**	**11.24**	**79.12**	11.92	**94.60**	**2.81**	**95.14**	**2.77**

Table 8: Comparison with existing continual learning methods under different task splits.

Method	AVE			Kinetics-Sounds			N24News		
	CAA ↑	FFM ↓	FAA ↑	CAA ↑	FFM ↓	FAA ↑	CAA ↑	FFM ↓	FAA ↑
LwF	55.78	38.63	33.85	47.80	34.85	30.39	70.32	22.53	50.09
iCaRL	74.82	33.08	58.13	57.33	62.39	35.36	58.58	74.29	30.02
SSIL	73.29	11.55	57.14	63.01	18.60	54.00	76.97	15.25	61.39
AFC	70.69	27.52	48.65	61.51	49.20	45.68	63.72	53.81	37.89
SDC-IL	56.60	24.72	42.83	40.96	20.58	33.32	58.65	13.99	46.26
ADC	54.97	33.31	36.05	48.99	29.00	34.56	41.34	36.61	34.72
DPCR	50.40	56.96	22.29	50.70	45.18	34.66	59.30	39.50	43.88
CMR-MFN	58.32	21.88	37.31	52.79	25.79	36.05	71.84	11.45	53.02
AV-CIL	72.97	16.19	59.64	70.00	15.80	56.13	78.89	15.03	63.16
DriftMem	**74.75**	**13.14**	**61.13**	**71.12**	**16.09**	**60.61**	**80.60**	**12.60**	**64.03**

the best performance across different multimodal benchmarks, achieving superior average accuracy and competitive forgetting compared with existing approaches. These results demonstrate that explicitly modeling representation drift enables DriftMem to better handle the accumulated changes in multimodal representations over long-term incremental learning, providing improved robustness under extended continual learning scenarios.

C.4 VISUALIZATION

visualization of attention scores. Figure 6 visualizes spatial and temporal attention across incremental tasks, where the heatmap and opacity indicate spatial and temporal attention, respectively. SDC exhibits substantial attention drift: its spatial focus shifts from interaction-relevant regions to irrelevant areas, while its temporal attention fluctuates across tasks. In contrast, DriftMem preserves more consistent attention on interacting subjects and informative frames, indicating more stable spatiotemporal grounding of multimodal representations. These results suggest that global representation compensation is insufficient to capture interaction-level drift in MCL. By explicitly modeling multimodal drift, DriftMem better preserves where and when complementary cues emerge, leading to more semantically consistent representations across tasks.

Visualization of semantic keys. The distribution of semantic keys is critical for both updating and retrieving DDM. To validate the effectiveness of CFC, which consists of inter-class orthogonal calibration and class-wise contrastive calibration, we visualize the semantic key space from two perspectives. For each dataset, we randomly select 10 classes for visualization. Specifically, we use the class-prototype cosine Gram matrix to evaluate inter-class separation, where lower off-diagonal

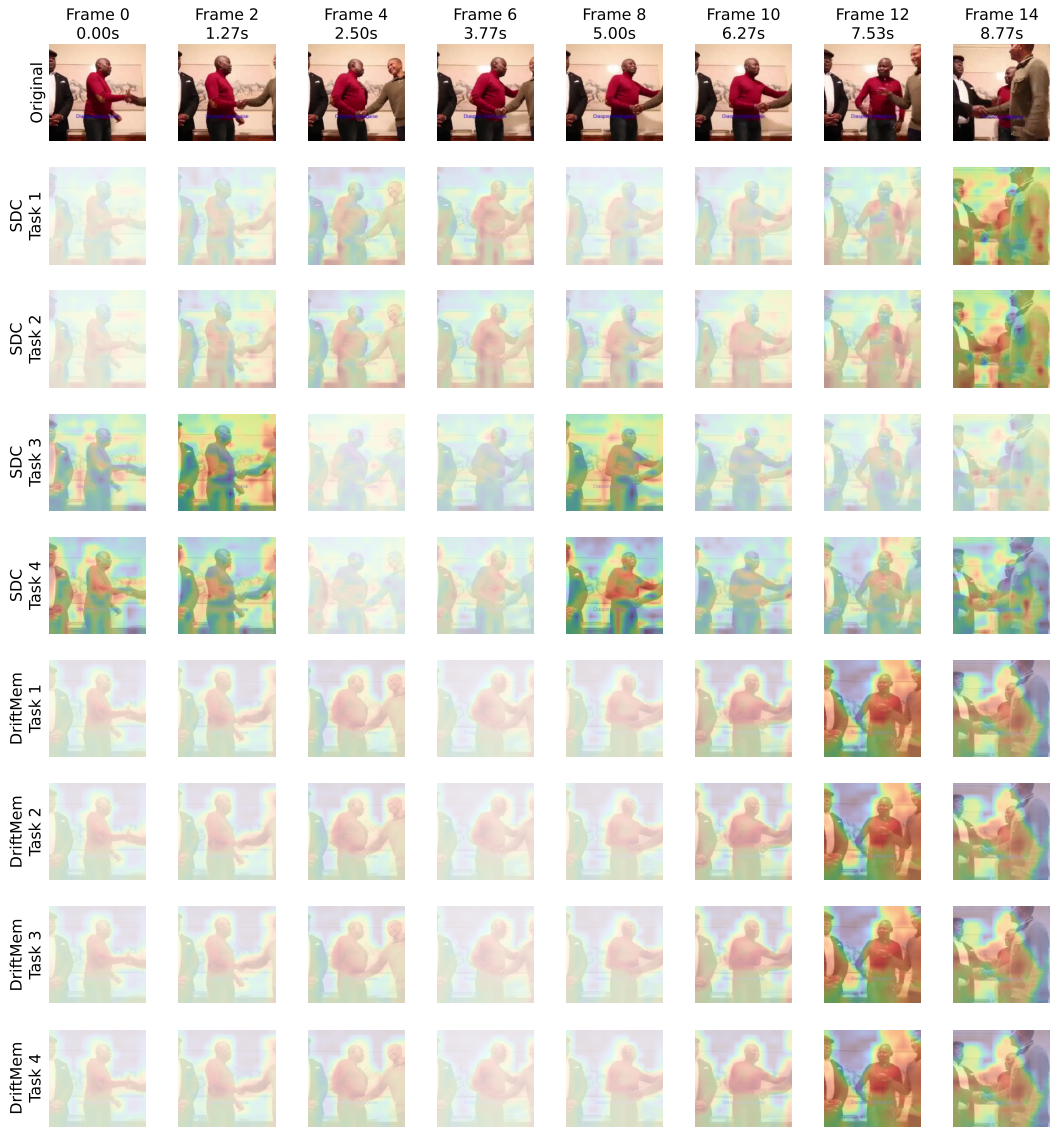
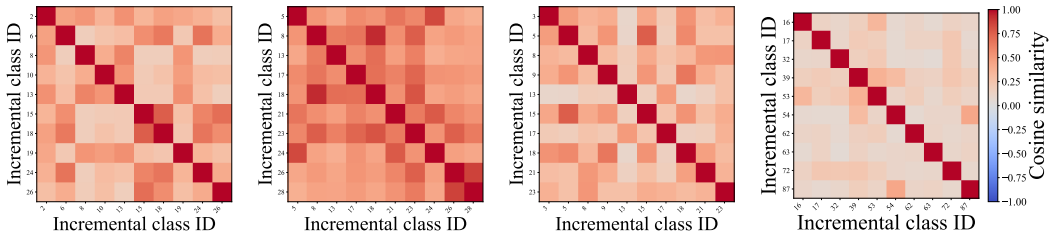


Figure 6: Visualization of attention maps.

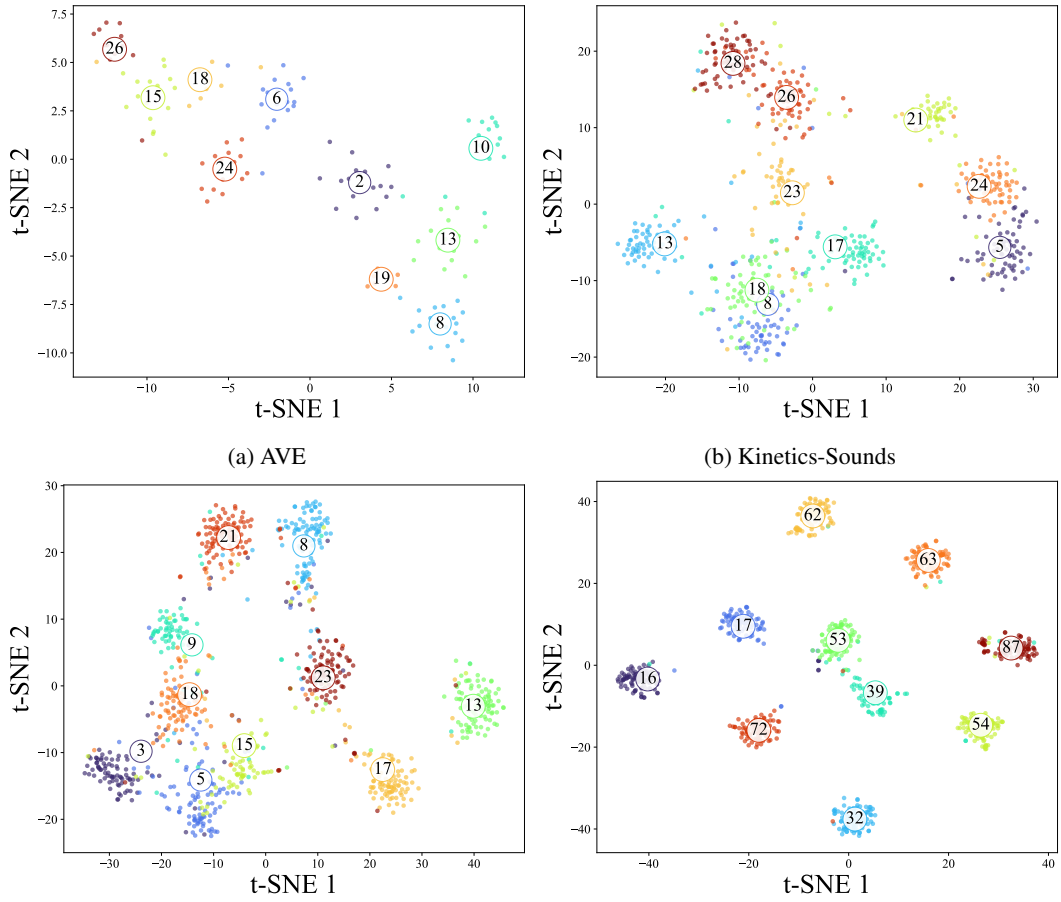
similarity indicates stronger orthogonality among different classes. Meanwhile, we employ t-SNE visualization to examine intra-class compactness and inter-class separability of semantic keys.

As shown in Figure 7, the class-prototype cosine Gram matrices exhibit relatively low off-diagonal responses, indicating that the semantic keys maintain clear separation across different classes. Furthermore, Figure 8 shows the t-SNE results, indicating that CFC produces compact intra-class clusters and clearer boundaries between different classes, as well as verifying the effectiveness of contrastive calibration in improving the discriminability of semantic keys. These observations suggest that CFC learns a more structured semantic key space, enabling more reliable drift retrieval and reducing interference during DDM updates.



(a) AVE (b) Kinetics-Sounds (c) N24News (d) UMPC-Food101

Figure 7: Class-prototype cosine Gram matrix of semantic keys.



(a) AVE (b) Kinetics-Sounds

(c) N24News (d) UMPC-Food101

Figure 8: t-SNE visualization of semantic keys.