CHEC: COUNTERFACTUAL LOW-RANK ADAPTATION WITH HISTORICAL EVIDENCE CORRECTION FOR CONTINUAL MISSING MODALITY LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Continual Missing Modality Learning (CMML) is important for multimodal systems that must continually acquire new knowledge while remaining reliable under intermittent modality absence. However, existing methods still face two limitations that require further improvement in performance: (1) they mainly compensate for missing-side information, without explicitly adapting the surviving modality after its cross-modal context changes; and (2) preserving historical knowledge does not ensure that its contribution to the current prediction is supported by the current input. To address these problems, this paper proposes CHEC, a novel CMML framework combining Counterfactual Low-Rank Adaptation with Historical Evidence Correction. During training, CHEC adds a controller conditioned on the current sample and modality availability to LoRA, allowing the transformation applied to the surviving modality to adapt to each input. It further introduces Utility-Gated Counterfactual Functional Correction (UG-CFC), which projects the functional difference between paired Complete and Missing views into a counterfactual correction attainable in the Missing-view rank space, and retains it only when the correction is predicted to benefit classification. This resolves the mismatch between the desired functional change and what the Missing representation can realize, while filtering harmful corrections without increasing LoRA rank. During inference, Historical Evidence Correction (HEC) replaces task-level hard routing with class-wise evidence correction. It retains all learned classes and calibrates historical evidence according to support from the current input, avoiding both premature class exclusion and unsupported contributions from historical knowledge. Across two existing CMML benchmarks, CHEC achieves the state-of-the-art performance among six baselines in all 30 settings. Further analyses show relative reductions of 14.0–40.4% in the Complete–Missing functional gap and 27.4–32.2% in the final historical-class error rate.

1 INTRODUCTION

Continual Missing Modality Learning (CMML) is important for multimodal systems that must continually acquire new knowledge while remaining reliable when individual modalities become unavailable. Existing research mainly follows two lines: handling missing modalities and preserving or accessing knowledge across sequential tasks. The first line mainly recovers unavailable information or adapts representations to incomplete inputs. Representative methods include MMIN (Zhao et al., 2021), which predicts missing-modality representations from available modalities, and SMIL (Ma et al., 2021) , which learns under severe modality missingness. Parameter-efficient approaches further adapt prompts to different modality-availability states: MAP (Lee et al., 2023) learns prompts for different missing cases, MSP (Jang et al., 2024) uses modality-specific prompts, Deep Correlated Prompting (Hu et al., 2024) models interactions between prompts and input features, SCP (Pipoli et al., 2025) conditions prompts on available modalities, and Synergistic Prompting (Zhang et al., 2025) combines complementary prompting mechanisms. More recently, MoRA (Zhao et al., 2026) applies low-rank adaptation to missing-modality recognition. These methods mainly compensate for information on the missing side. However, modality absence also changes the surviving modality itself: even when its raw content is unchanged, losing the other modality alters its cross-modal con-

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.

text and hidden representation. Existing methods do not explicitly learn how the adaptation applied to the surviving modality should change with the current sample and modality availability.

A second line of work focuses on preserving and accessing knowledge across sequential tasks. L2P (Wang et al., 2022b) and DualPrompt (Wang et al., 2022a) use prompts to store and retrieve task knowledge, while InfLoRA (Liang & Li, 2024) constrains low-rank updates to reduce interference with previous tasks. In CMML, RebQ (Zhao et al., 2024) decomposes prompts into modality-related components and reconstructs queries for accessing them; Guo et al. (Guo et al., 2025) represent modality and task information with prompts and encourage interaction across tasks; DeLo (Liu et al., 2026) stores task knowledge in partitioned low-rank components and uses Task-Key Memory for test-time routing; and PAL (Yue et al., 2026) combines prompting with analytic class-incremental learning. These methods improve how historical knowledge is preserved or accessed, but preserving historical knowledge does not ensure that its contribution to the current prediction is reliable. Hard routing can exclude candidate classes from other tasks before class-wise comparison, whereas allowing knowledge from multiple previous tasks to participate can introduce prediction evidence unsupported by the current input. Task-agnostic inference therefore requires not only access to historical knowledge, but also control over how much historical evidence should contribute to the current prediction.

To address these two problems, this paper proposes CHEC, a framework for continual missing modality learning that adapts the surviving modality under missing inputs while controlling unreliable historical evidence during task-agnostic inference. The main contributions are summarized as follows:

This paper proposes Counterfactual Low-Rank Adaptation with Historical Evidence Correction for Continual Missing Modality Learning, which jointly addresses the adaptation of the surviving modality under missing inputs and the unreliable contribution of historical knowledge during task-agnostic inference.

On the training side, CHEC uses a conditional controller together with Utility-Gated Counterfactual Functional Correction (UG-CFC). UG-CFC projects the functional difference between paired Complete and Missing views into a correction realizable in the current Missing-view rank space and retains it only when it benefits classification, enabling instance-specific adaptation of the surviving modality without increasing LoRA rank.

On the inference side, Historical Evidence Correction (HEC) replaces task-level hard routing with class-wise evidence correction derived from historical statistics. It retains all learned classes while selectively suppressing unsupported historical evidence, avoiding both candidate-class exclusion and indiscriminate contributions from historical knowledge.

Across two CMML benchmarks and six baselines, CHEC achieves the state-of-the-art performance in all 30 settings. Targeted analyses further show relative reductions of 14.0–40.4% in the Complete–Missing functional gap and 27.4–32.2% in the final historical-class error rate.

2 Related Work

Recognition with missing modalities. Existing methods for missing modalities mainly follow three directions. First, reconstruction-based methods recover or predict unavailable information from observed modalities; MMIN (Zhao et al., 2021), for example, predicts a missing-modality representation from the available modalities. Second, representation-learning methods seek features that remain robust under incomplete inputs; SMIL (Ma et al., 2021) uses Bayesian meta-learning to handle severe modality missingness. Third, parameter-efficient adaptation methods adjust learnable parameters according to modality availability. MAP (Lee et al., 2023), MSP (Jang et al., 2024), Deep Correlated Prompting (Hu et al., 2024), SCP (Pipoli et al., 2025), and Synergistic Prompting (Zhang et al., 2025) develop different prompting mechanisms for missing-modality conditions, while MoRA (Zhao et al., 2026) introduces shared and modality-specific low-rank adaptation. However, these methods do not explicitly address how the surviving modality should adapt when its cross-modal context changes after modality loss; CHEC addresses this problem by conditioning the surviving-modality adaptation on the current sample and availability state without reconstructing the missing modality.

Continual learning. Continual learning (CL) aims to acquire knowledge from sequential tasks while retaining previously learned information (Wang et al., 2022a;b; Liang & Li, 2024). Parameter-efficient CL methods preserve task knowledge without updating the entire model: L2P (Wang et al., 2022b) and DualPrompt (Wang et al., 2022a) store and retrieve task information through prompts, while InfLoRA (Liang & Li, 2024) constrains low-rank updates to reduce interference with previous tasks. In CMML (Zhao et al., 2024), continual learning is further complicated by changing modality availability. RebQ (Zhao et al., 2024) decomposes prompts into modality-related components and reconstructs queries to access them; Guo et al. (Guo et al., 2025) encode modality and task information with prompts and promote interaction across tasks; and DeLo (Liu et al., 2026) organizes modality-related low-rank factors into task-specific partitions to reduce cross-task interference. However, these methods do not determine whether the evidence contributed by retained historical knowledge is supported by the current input; CHEC addresses this problem with input-dependent historical evidence correction, retaining access to historical knowledge while suppressing unsupported contributions to the current prediction.

3 Problem formulation

Following the CMML protocol introduced by RebQ (Zhao et al., 2024), this paper considers a sequence of T tasks, where T is the total number of tasks, t indexes the current task, and $\mathcal{D}_t$ denotes the training set of task t. Each sample contains an image, text, target y, and an availability state $s \in \{\mathrm{C}, \mathrm{TM}, \mathrm{IM}\}$, denoting Complete, text-missing (image-only), and image-missing (text-only) observations, respectively. When learning task t, only $\mathcal{D}_t$ is accessible; the current task boundary and label set $\mathcal{Y}_t$ are known, while data from previous tasks cannot be revisited. At test time, task identity is unavailable and prediction is performed over all labels observed up to stage t, $\mathcal{Y}_{1:t} = \bigcup_{k=1}^{t} \mathcal{Y}_k$, where k is the task index in the accumulated label set.

Let $\eta \in [0, 1]$ denote the fraction of modality-incomplete samples. Under Mixed, text-missing, and image-missing settings, the $(\mathrm{C}, \mathrm{TM}, \mathrm{IM})$ proportions are $(1-\eta, \eta/2, \eta/2)$, $(1-\eta, \eta, 0)$, and $(1-\eta, 0, \eta)$, respectively. Training and testing use the same missingness pattern and rate.

4 The CHEC method

CHEC addresses the two problems at the stages where they arise. During training, CHEC uses two coupled components: a conditional controller makes the surviving-modality LoRA action depend on the current sample and modality availability, while Utility-Gated Counterfactual Functional Correction (UG-CFC) determines which functional correction is both realizable under the Missing representation and beneficial to classification. During inference, Historical Evidence Correction (HEC) retains all learned classes and corrects historical evidence according to support from the current input. Figures 1 and 2 summarize the training and inference mechanisms, respectively.

4.1 Conditional low-rank adaptation

For fixed model parameters, a standard task LoRA applies the same low-rank transformation to different samples in that task, although Complete, TM, and IM inputs produce different surviving representations. The first training-side component of CHEC is therefore a conditional controller that makes this transformation depend on the current sample and modality availability within the existing LoRA rank space.

For task t, projection site ℓ, and modality m, let $\boldsymbol{A}_{t\ell m} \in \mathbb{R}^{r \times d_{\mathrm{in}}}$ and $\boldsymbol{B}_{t\ell m} \in \mathbb{R}^{d_{\mathrm{out}} \times r}$ be the LoRA factor matrices, where r is the LoRA rank and d_{in} and d_{out} are the input and output dimensions of the projection. The backbone projection matrix $\boldsymbol{W}_\ell$ remains frozen. For readability, task, site, and modality subscripts are omitted below when unambiguous. Following DeLo (Liu et al., 2026), text and image use separate factors, reducing interference and keeping each correction in its modality branch. For a token-representation vector $\boldsymbol{h}$ from an available modality in sample x, we have $\boldsymbol{h}' = \boldsymbol{W}\boldsymbol{h} + \boldsymbol{B}\boldsymbol{A}\boldsymbol{h} + \mathrm{sg}(\boldsymbol{B})\boldsymbol{R}_x\,\mathrm{sg}(\boldsymbol{A}\boldsymbol{h})$, with $\boldsymbol{R}_x = G_{\boldsymbol{\phi}_t}(\boldsymbol{q}_x) \in \mathbb{R}^{r \times r}$.

Here, $\boldsymbol{R}_x$ is the sample-conditioned rank-space transformation matrix, $G_{\boldsymbol{\phi}_t}$ is the task-t controller with parameter vector $\boldsymbol{\phi}_t$, and $\boldsymbol{q}_x$ is its input vector. The operator $\mathrm{sg}(\cdot)$ denotes stop-gradient: it preserves the forward value but blocks gradients through the enclosed quantity. The controller

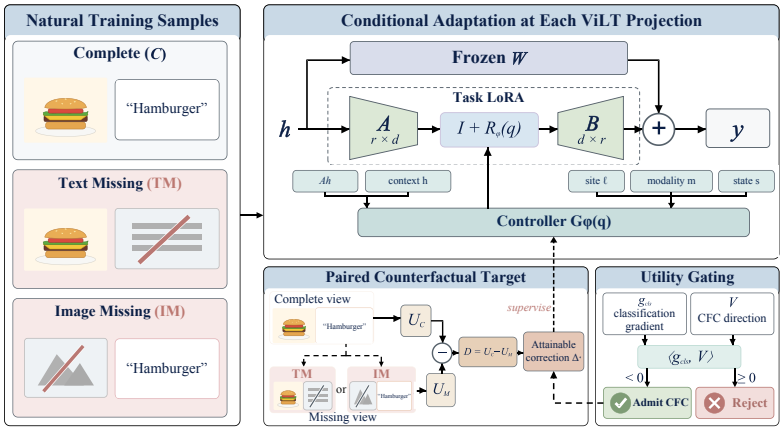

Figure 1: Training mechanism of the proposed CHEC: conditional controller and UG-CFC.

input $\boldsymbol{q}_x$ contains rank coordinates, same-modality context, and embeddings for site, modality, and availability state; tokens from the same sample, site, and modality share $\boldsymbol{R}_x$.

Using the $r \times r$ identity matrix $\boldsymbol{I}_r$, the forward transformation can be written as $\boldsymbol{W}\boldsymbol{h} + \boldsymbol{B}(\boldsymbol{I}_r + \boldsymbol{R}_x)\boldsymbol{A}\boldsymbol{h}$. Thus, $\boldsymbol{B}\boldsymbol{A}$ preserves the native task-LoRA action, whereas $\boldsymbol{B}\boldsymbol{R}_x\boldsymbol{A}$ provides the sample- and availability-dependent deviation. Because $\boldsymbol{R}_x$ only recombines coordinates within the existing r-dimensional LoRA space, this decomposition preserves the native task adaptation while adding sample- and availability-dependent flexibility without increasing the rank of an individual LoRA update or introducing a separate missing-modality LoRA branch. Zero initialization makes start as standard LoRA; controller details are in Appendix B.

Feature-derived parts of $\boldsymbol{q}_x$ are also detached. Naturally occurring Complete, TM, and IM examples train the native LoRA, classifier, and conditional controller through the classification objective, whereas the auxiliary correction path defined below updates the controller without directly modifying $\boldsymbol{A}$, $\boldsymbol{B}$, or the classifier. Historical-task adapters, controllers, and classifiers remain frozen.

4.2 Utility-Gated Counterfactual Functional Correction (UG-CFC)

The controller can change the LoRA action under Missing inputs but does not specify a useful direction. UG-CFC derives this direction from Complete and Missing views of the same instance while keeping the raw surviving-modality content fixed.

For one surviving-modality branch at an adapted site, let $\boldsymbol{H}_{\mathrm{C}}, \boldsymbol{H}_{\mathrm{M}} \in \mathbb{R}^{d_{\mathrm{in}} \times n}$ denote the hidden-state matrices of the same n aligned valid surviving-modality tokens in the Complete and Missing views, where n is the number of aligned tokens. Their LoRA rank-coordinate matrices are $\boldsymbol{Z}_{\mathrm{C}} = \boldsymbol{A}\boldsymbol{H}_{\mathrm{C}}$ and $\boldsymbol{Z}_{\mathrm{M}} = \boldsymbol{A}\boldsymbol{H}_{\mathrm{M}}$, with $\boldsymbol{Z}_{\mathrm{C}}, \boldsymbol{Z}_{\mathrm{M}} \in \mathbb{R}^{r \times n}$. Let $\boldsymbol{R}_{\mathrm{C}}$ and $\boldsymbol{R}_{\mathrm{M}}$ denote the corresponding controller-output matrices. The conditioned rank-space actions are $\boldsymbol{U}_{\mathrm{C}} = (\boldsymbol{I}_r + \boldsymbol{R}_{\mathrm{C}})\boldsymbol{Z}_{\mathrm{C}}$ and $\boldsymbol{U}_{\mathrm{M}} = (\boldsymbol{I}_r + \boldsymbol{R}_{\mathrm{M}})\boldsymbol{Z}_{\mathrm{M}}$, and their desired functional displacement is $\boldsymbol{D} = \boldsymbol{U}_{\mathrm{C}} - \boldsymbol{U}_{\mathrm{M}}$. Because the Missing branch can act only through its current coordinates, an arbitrary displacement is not necessarily realizable. CHEC therefore seeks a rank-space correction matrix $\boldsymbol{\Delta} \in \mathbb{R}^{r \times r}$ such that $\boldsymbol{\Delta}\boldsymbol{Z}_{\mathrm{M}}$ approximates $\boldsymbol{D}$. Let $\|\cdot\|_F$ denote the Frobenius norm. With ridge coefficient $\rho > 0$, CHEC solves $\boldsymbol{\Delta}^* = \arg\min_{\boldsymbol{\Delta} \in \mathbb{R}^{r \times r}} \left\|\boldsymbol{\Delta}\boldsymbol{Z}_{\mathrm{M}} - \boldsymbol{D}\right\|_F^2 + \rho\left\|\boldsymbol{\Delta}\right\|_F^2$.

The ridge problem has closed-form solution: $\boldsymbol{\Delta}^* = \boldsymbol{D}\boldsymbol{Z}_{\mathrm{M}}^{\mathsf{T}}\left(\boldsymbol{Z}_{\mathrm{M}}\boldsymbol{Z}_{\mathrm{M}}^{\mathsf{T}} + \rho\boldsymbol{I}_r\right)^{-1}$.

projects the desired displacement onto directions supported by the current Missing-view rank coordinates. The detached target matrix $\boldsymbol{R}_{\mathrm{T}} = \mathrm{sg}(\boldsymbol{R}_{\mathrm{M}} + \boldsymbol{\Delta}^*)$ therefore contains only changes supported by the current Missing-view rank coordinates. The Complete view is used only to construct this training-time reference.

Counterfactual Functional Correction (CFC) measures the functional error induced at the current projection site's LoRA output rather than a raw operator distance. For the current batch, let $\mathcal{P}$ denote

the set of surviving-modality branches indexed by a Complete instance i, a constructed missing state $s \in \{\mathrm{TM}, \mathrm{IM}\}$, and an adapted site u. For a branch $p = (i, s, u)$, let n_p be its number of valid tokens and $d_{\mathrm{out},p}$ its output dimension. We define $\mathcal{L}_p^{\mathrm{CFC}} = \frac{\left\| \mathrm{sg}(\boldsymbol{B}_p)\left(\boldsymbol{R}_{\mathrm{M},p} - \boldsymbol{R}_{\mathrm{T},p}\right) \mathrm{sg}(\boldsymbol{Z}_{\mathrm{M},p}) \right\|_F^2}{n_p d_{\mathrm{out},p}}$ and $\mathcal{L}_{\mathrm{CFC}} = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \mathcal{L}_p^{\mathrm{CFC}}$.

therefore supervises only operator errors activated by the current Missing-view rank coordinates and mapped through $\boldsymbol{B}$ into the LoRA output space.

A correction supported by the current Missing-view coordinates may still hurt classification. For Complete instance i, missing state $s \in \{\mathrm{TM}, \mathrm{IM}\}$, and adapted site u among the S adapted sites, define the CFC descent direction $\boldsymbol{V}_{i,s,u} = -\nabla_{\boldsymbol{R}_{\mathrm{M},i,s,u}} \mathcal{L}_{i,s,u}^{\mathrm{CFC}}$ and classification-gradient probe $\boldsymbol{G}_{i,s,u} = \nabla_{\boldsymbol{R}_{\mathrm{M},i,s,u}} \ell_{\mathrm{probe}}(x_{i,s}, y_i)$. Here $x_{i,s}$ is the constructed Missing view of instance i, y_i is its target, and ℓ_{probe} uses the same classification criterion as $\mathcal{L}_{\mathrm{cls}}$. It is evaluated only to obtain the gradient with respect to $\boldsymbol{R}_{\mathrm{M}}$. Let $\mathbb{I}[\cdot]$ denote the indicator function and $a_{i,s} \in \{0, 1\}$ the binary utility gate. Let $\langle \cdot, \cdot \rangle_F$ denote the Frobenius inner product. Using the directions in Eqs. (4.2)–(4.2), CHEC admits CFC for a Missing view only if $a_{i,s} = \mathbb{I}\left[\sum_{u=1}^{S} \left\langle \boldsymbol{G}_{i,s,u}, \boldsymbol{V}_{i,s,u} \right\rangle_F < 0\right]$.

The gate accepts the correction only when the joint direction is predicted, to first order, to reduce classification loss. TM and IM are gated separately because they retain different modalities. Appendix A provides the local first-order view.

Complete, TM, and IM samples from the benchmark protocol use the standard classification objective $\mathcal{L}_{\mathrm{cls}}$; constructed Missing views are used only for the CFC target and utility probe. Let N_C denote the number of Complete instances in the current batch. The UG-CFC loss reads $\mathcal{L}_{\text{UG-CFC}} = \frac{1}{2N_C S} \sum_{i=1}^{N_C} \sum_{s \in \{\mathrm{TM},\mathrm{IM}\}} a_{i,s} \sum_{u=1}^{S} \mathcal{L}_{i,s,u}^{\mathrm{CFC}}$, set to zero when $N_C = 0$. Let λ denote the UG-CFC loss weight and $\boldsymbol{\theta}_{\mathrm{head}}$ the classifier-parameter vector. The total objective and separated update rules are $\mathcal{L} = \mathcal{L}_{\mathrm{cls}} + \lambda \mathcal{L}_{\text{UG-CFC}}$, with gradients $\nabla_{\boldsymbol{A},\boldsymbol{B},\boldsymbol{\theta}_{\mathrm{head}}} \mathcal{L} = \nabla_{\boldsymbol{A},\boldsymbol{B},\boldsymbol{\theta}_{\mathrm{head}}} \mathcal{L}_{\mathrm{cls}}$ and $\nabla_{\boldsymbol{\phi}_t} \mathcal{L} = \nabla_{\boldsymbol{\phi}_t} \left(\mathcal{L}_{\mathrm{cls}} + \lambda \mathcal{L}_{\text{UG-CFC}}\right)$.

keeps the auxiliary gradient confined to the controller. Together, ridge projection enforces realizability, the utility gate enforces classification usefulness, and gradient separation prevents the auxiliary correction objective from directly altering the native LoRA or classifier. Thus, UG-CFC shapes only the controller's Missing-state conditional action through its auxiliary path, and the Complete view is required only to construct training-time supervision. After each task, its adapter, controller, and classifier are frozen.

4.3 Historical evidence correction (HEC)

Task isolation protects historical parameters, not prediction reliability. HEC therefore keeps all learned classes instead of selecting one task and measures how well the current input matches each historical class. Figure 2 illustrates this support-based evidence-correction process.

At stage k, where k denotes the current continual-learning stage, let x denote the current test input. All learned task experts evaluate x. For task expert t, let $\boldsymbol{f}_t(x) \in \mathbb{R}^d$ be its classification-token feature vector, where d is the feature dimension, and let $z_{t,c}(x)$ be the scalar logit for class index c. After task t, CHEC estimates the class-center vector $\boldsymbol{\mu}_{t,c} \in \mathbb{R}^d$ from positive examples and one task-shared precision matrix $\boldsymbol{\Omega}_t \in \mathbb{R}^{d \times d}$ from pooled residuals after subtracting class centers, using Ledoit–Wolf shrinkage (Ledoit & Wolf, 2004) to regularize covariance estimation. The scalar class-conditional distance is $d_{t,c}(x) = (\boldsymbol{f}_t(x) - \boldsymbol{\mu}_{t,c})^{\top} \boldsymbol{\Omega}_t (\boldsymbol{f}_t(x) - \boldsymbol{\mu}_{t,c})$.

For class c, let $\{b_{t,c,i}\}_{i=1}^{N_{t,c}}$ denote the positive validation distances, where $b_{t,c,i}$ is the ith distance and $N_{t,c}$ is their count. HEC calibrates $d_{t,c}(x)$ into the empirical class support: $p_{t,c}(x) = \frac{1 + \sum_{i=1}^{N_{t,c}} \mathbb{I}\left[b_{t,c,i} \geq d_{t,c}(x)\right]}{N_{t,c} + 1}$.

where $\mathbb{I}[\cdot]$ is the indicator function. The support decreases as the current feature becomes less typical of the positive distribution of class c; thus, smaller $p_{t,c}(x)$ indicates weaker class support. After task completion, CHEC stores only class centers, the task-shared precision matrix, and per-class validation-distance distributions.

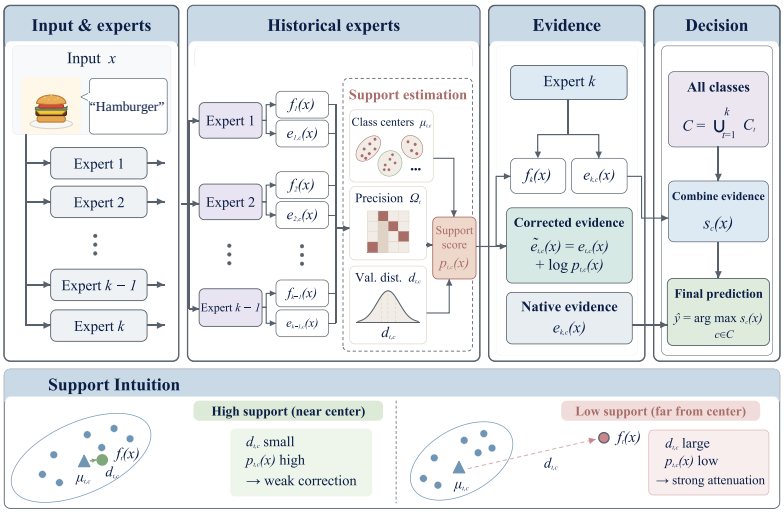

Figure 2: Inference mechanism of the proposed CHEC: HEC at stage k.

To place single-label and multi-label predictions on a common evidence scale, define the scalar native evidence: $e_{t,c}(x) = z_{t,c}(x)$ for multi-label settings; $e_{t,c}(x) = z_{t,c}(x) - \log \sum_{c' \neq c} \exp z_{t,c'}(x)$ for single-label settings. where c' indexes the other classes of task expert t. Using the common evidence scale, HEC defines the scalar corrected evidence $\widetilde{e}_{t,c}(x)$ by $\widetilde{e}_{t,c}(x) = e_{t,c}(x) + \mathbb{I}[t < k] \log p_{t,c}(x)$.

The correction is one-sided because $p_{t,c}(x) \in (0, 1]$: it can attenuate unsupported historical evidence but cannot create additional positive evidence, while the current-task evidence remains unchanged. Unlike hard routing, no learned class is removed before prediction. This conservative correction therefore preserves candidate-class coverage while reducing unsupported contributions from historical knowledge, and the same rule applies to both single-label and multi-label decisions. Food101 selects the largest evidence over all learned classes, whereas MM-IMDb applies the common zero threshold independently to each class.

5 Experiments

5.1 Experimental setup

Benchmarks and protocol. Following the CMML protocol introduced by RebQ (Zhao et al., 2024), experiments use two established CMML benchmarks. UPMC-Food101-CMML is derived from UPMC-Food101 (Wang et al., 2015); after removing the least frequent category, its 100 classes form ten sequential single-label tasks. MM-IMDb-CMML is derived from MM-IMDb (Arevalo Ovalle et al., 2017); after removing seven infrequent genres, its 20 labels form five multi-label tasks with four new labels each. Experiments evaluate text-missing (TM), image-missing (IM), and Mixed patterns at $\eta \in \{10, 30, 50, 70, 90\}\%$. Modality absence occurs during both training and testing, and test-time task identity is unavailable.

Evaluation metrics. Evaluation uses the standard CMML metrics Average Performance (AP) and peak-based Average Forgetting (FG). Let $a_{i,j}$ denote test performance on task i after training through task j:

$$\mathrm{AP} = \frac{1}{T} \sum_{i=1}^{T} a_{i,T}, \qquad \mathrm{FG} = \frac{1}{T-1} \sum_{i=1}^{T-1} \left(\max_{i \leq j < T} a_{i,j} - a_{i,T} \right). \tag{1}$$

On Food101, $a_{i,j}$ is accuracy; on MM-IMDb, it is macro-F1.

Implementation details. CHEC uses a frozen ViLT-B/32 backbone (Kim et al., 2021) with rank-8 LoRA adapters in all Transformer query, key, and value projections. Training uses AdamW with

Table 1: Performance comparison on UPMC-Food101-CMML across five missingness rates under text-missing (TM), image-missing (IM), and Mixed conditions. Each entry reports AP ↑/FG ↓ (%), with AP computed from accuracy. Bold indicates the highest AP in each setting.

η	Pattern	MAP	MSP	L2P	DualPrompt	RebQ	DeLo	**CHEC**
		AP / FG	AP / FG	AP / FG	AP / FG	AP / FG	AP / FG	AP / FG
10%	TM	20.66 / 82.50	21.45 / 80.03	34.09 / 5.80	59.56 / 3.38	68.67 / 9.50	77.19 / 10.21	**78.37** / 10.88
10%	IM	21.53 / 82.20	23.29 / 78.98	35.21 / 5.51	59.90 / 1.72	72.46 / 7.64	80.23 / 7.88	**81.17** / 7.63
10%	Mixed	22.84 / 80.13	22.12 / 79.35	34.00 / 6.55	58.18 / 6.32	71.06 / 8.22	78.12 / 10.32	**79.43** / 10.91
30%	TM	16.57 / 83.77	21.37 / 79.95	30.40 / 6.53	51.86 / 6.46	62.06 / 10.76	74.08 / 9.69	**75.71** / 10.54
30%	IM	24.00 / 77.12	22.21 / 76.02	34.75 / 4.66	52.66 / 5.11	71.62 / 6.44	78.20 / 6.34	**79.46** / 6.12
30%	Mixed	20.66 / 79.81	21.89 / 79.01	29.76 / 6.87	48.56 / 8.73	66.37 / 8.07	73.63 / 8.89	**75.12** / 9.74
50%	TM	18.18 / 79.77	17.88 / 77.31	28.95 / 6.33	47.70 / 6.09	55.87 / 12.07	71.41 / 14.77	**73.32** / 16.03
50%	IM	23.85 / 75.59	21.57 / 77.99	32.18 / 4.55	50.22 / 5.03	69.23 / 5.84	76.38 / 7.39	**77.70** / 7.26
50%	Mixed	18.66 / 79.80	18.10 / 78.04	25.30 / 6.17	43.67 / 8.52	62.40 / 8.24	70.59 / 8.27	**72.62** / 9.91
70%	TM	17.68 / 77.82	19.29 / 78.17	26.57 / 5.80	43.09 / 8.96	50.00 / 12.47	70.04 / 14.91	**72.25** / 17.22
70%	IM	22.48 / 75.21	21.22 / 75.21	30.43 / 3.99	50.28 / 5.19	69.41 / 3.73	74.80 / 5.32	**76.27** / 5.19
70%	Mixed	20.00 / 75.95	19.76 / 76.54	24.62 / 5.66	40.69 / 7.26	59.92 / 8.56	67.42 / 9.41	**69.88** / 12.67
90%	TM	16.92 / 76.60	18.62 / 75.14	24.94 / 5.60	37.26 / 11.46	48.15 / 12.76	67.30 / 15.68	**69.93** / 18.07
90%	IM	24.89 / 70.84	21.69 / 71.23	29.77 / 4.54	51.16 / 4.80	67.71 / 4.71	74.21 / 4.83	**75.82** / 4.66
90%	Mixed	18.41 / 75.19	20.45 / 74.31	24.62 / 5.66	40.69 / 7.29	54.67 / 8.78	63.37 / 10.97	**65.91** / 14.19

Table 2: Performance comparison on MM-IMDb-CMML across five missingness rates under text-missing (TM), image-missing (IM), and Mixed conditions. Each entry reports AP ↑/FG ↓ (%), with AP computed from macro-F1. Bold indicates the highest AP in each setting.

η	Pattern	MAP	MSP	L2P	DualPrompt	RebQ	DeLo	**CHEC**
		AP / FG	AP / FG	AP / FG	AP / FG	AP / FG	AP / FG	AP / FG
10%	TM	15.77 / 34.67	19.49 / 35.11	13.67 / 8.96	22.79 / 20.42	24.19 / 32.46	29.14 / 32.11	**29.68** / 31.22
10%	IM	16.63 / 33.30	19.01 / 34.89	13.54 / 8.26	24.82 / 24.09	27.35 / 28.88	34.77 / 30.36	**35.16** / 29.58
10%	Mixed	17.24 / 31.28	15.69 / 32.17	13.90 / 8.42	24.19 / 24.04	26.90 / 28.26	31.84 / 30.72	**32.30** / 29.96
30%	TM	19.85 / 32.14	20.01 / 34.87	10.68 / 8.03	22.96 / 23.96	24.75 / 25.86	29.28 / 32.79	**29.85** / 31.74
30%	IM	17.19 / 31.09	18.31 / 32.45	10.98 / 7.27	20.98 / 19.67	26.68 / 27.89	34.56 / 32.38	**35.00** / 31.25
30%	Mixed	14.44 / 33.29	14.20 / 35.57	11.58 / 7.90	22.18 / 20.72	25.80 / 26.95	30.62 / 29.23	**31.13** / 28.35
50%	TM	19.19 / 36.18	15.93 / 35.29	9.54 / 5.54	20.04 / 21.48	23.89 / 25.27	28.88 / 33.18	**29.60** / 31.91
50%	IM	18.83 / 32.02	16.78 / 34.14	9.77 / 4.88	19.97 / 20.24	26.45 / 27.53	32.83 / 31.09	**33.36** / 30.05
50%	Mixed	17.36 / 29.67	16.24 / 31.81	10.08 / 5.23	20.32 / 19.09	25.31 / 25.88	30.15 / 33.72	**30.82** / 32.46
70%	TM	17.45 / 27.14	18.01 / 27.32	7.26 / 5.37	16.80 / 13.82	22.41 / 24.39	27.21 / 33.39	**28.02** / 31.86
70%	IM	16.09 / 34.59	14.30 / 35.72	7.49 / 5.20	18.42 / 15.32	26.27 / 27.56	31.03 / 32.76	**31.61** / 31.44
70%	Mixed	17.20 / 29.39	18.27 / 29.34	6.93 / 6.23	16.19 / 15.76	23.77 / 25.26	28.97 / 31.84	**29.84** / 29.93
90%	TM	18.85 / 25.03	18.01 / 29.14	5.98 / 5.82	15.50 / 11.27	21.37 / 23.66	26.11 / 35.11	**27.04** / 33.53
90%	IM	17.79 / 31.65	17.88 / 32.19	6.29 / 5.09	21.22 / 18.38	26.22 / 27.50	30.23 / 33.85	**30.85** / 32.29
90%	Mixed	16.28 / 27.89	16.20 / 31.47	6.84 / 6.39	16.82 / 8.44	25.42 / 25.82	28.59 / 32.79	**29.35** / 31.17

batch size 4, learning rate 10^{-4}, and $\lambda = 1$. After each task, HEC uses only that task's training and validation data to construct support statistics and calibration distributions, without revisiting historical raw samples. Additional implementation details are provided in Appendix B.

Baselines. Baselines include MAP (Lee et al., 2023) and MSP (Jang et al., 2024) for missing-modality learning, L2P (Wang et al., 2022b) and DualPrompt (Wang et al., 2022a) for continual learning, and RebQ (Zhao et al., 2024) and DeLo (Liu et al., 2026) as direct CMML methods.

5.2 Performance across missingness conditions

As shown in Tables 1–2, CHEC achieves the best AP among the compared methods in all 30 missingness settings. Relative to DeLo, mean AP across the 15 settings improves by 1.73 percentage points on Food101 and 0.63 percentage points on MM-IMDb, with gains across all missingness rates and TM, IM, and Mixed patterns.

Table 3: Training controls at 70% Mixed missingness, evaluated with native all-head inference and HEC disabled. AP and FG are reported in percentage points for Food101 and MM-IMDb.

Configuration	Food AP ↑	Food FG ↓	MM AP ↑	MM FG ↓
Compute-matched control	61.73	10.57	26.05	**29.41**
Same-label CFC	63.41	10.14	26.37	29.76
Paired CFC, no gate	66.79	8.91	27.26	30.44
UG-CFC	**68.50**	**8.02**	**27.87**	31.18

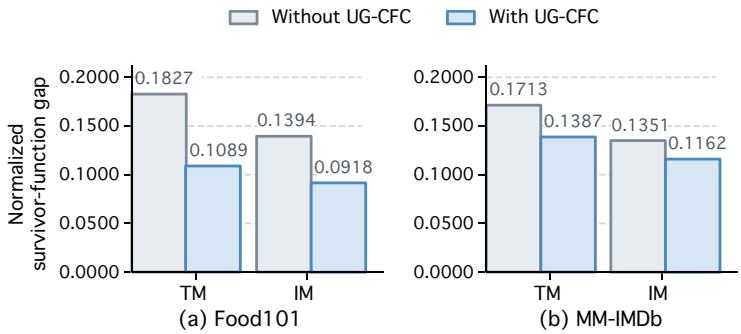

Figure 3: Complete-to-Missing surviving-modality functional gap with and without UG-CFC, where lower is better. All protocol-Complete validation inputs are used. TM and IM denote image and text survivors, respectively. Appendix D gives the exact definition and aggregation.

Relative to DeLo, CHEC lowers FG by 1.24 percentage points on average on MM-IMDb. Section 5.4 further analyzes FG behavior.

To examine the stability of the smaller AP margin on MM-IMDb, we repeat the 70% Mixed setting with paired seeds 17, 42, and 73. Relative to DeLo, CHEC improves AP and reduces FG for all three seeds, with paired differences of $\Delta\mathrm{AP} = +0.89 \pm 0.15$ and $\Delta\mathrm{FG} = -1.88 \pm 0.25$ percentage points (mean $\pm$ sample standard deviation). Full per-seed results are reported in Appendix E.

5.3 Counterfactual low-rank adaptation: pairing and utility gating

Targeted analyses use 70% Mixed missingness.

Under native all-head inference with HEC disabled, Table 3 isolates the two design choices of UG-CFC: Complete-reference pairing and utility gating. Relative to the compute-matched control, same-label CFC improves AP by 1.68 percentage points on Food101 and 0.32 points on MM-IMDb. Replacing same-label pairing with the same-instance Complete/Missing pair adds 3.38 points on Food101 and 0.89 points on MM-IMDb, showing that instance correspondence provides supervision beyond label agreement. Utility gating adds a further 1.71 points on Food101 and 0.61 points on MM-IMDb, indicating that not every realizable correction benefits classification. On MM-IMDb, FG rises from 29.41 to 31.18 across these controls. Section 5.4 separates this increase from final old-task performance.

Figure 3 directly tests the functional discrepancy targeted by UG-CFC. On Food101, the TM gap decreases from 0.1827 to 0.1089 and the IM gap from 0.1394 to 0.0918. On MM-IMDb, the TM gap decreases from 0.1713 to 0.1387 and the IM gap from 0.1351 to 0.1162. Thus, UG-CFC reduces the gap in all four dataset–missing-state combinations, bringing surviving-modality behavior closer to the paired Complete reference.

5.4 Historical evidence correction

Table 4 compares native all-head inference, DeLo's Task-Key single-task hard routing (Liu et al., 2026), and HEC using the same UG-CFC-trained experts. Native all-head inference yields AP 68.50 on Food101 and 27.87 on MM-IMDb. Hard routing raises Food101 AP to 69.03 but reduces MM-

Table 4: Inference controls after UG-CFC training at 70% Mixed missingness. AP and FG are reported in percentage points for Food101 and MM-IMDb.

Configuration	Food AP ↑	Food FG ↓	MM AP ↑	MM FG ↓
Native all-head	68.50	**8.02**	27.87	31.18
DeLo Task-Key hard routing	69.03	10.11	26.91	32.24
HEC	**69.88**	12.67	**29.84**	**29.93**

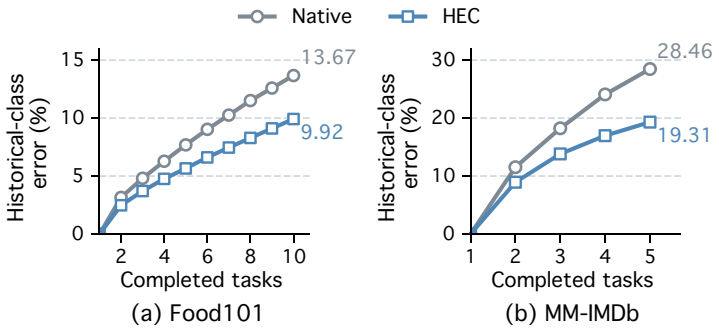

Figure 4: Historical-Class Error Rate (HCER) at 70% Mixed missingness using the same UG-CFC-trained experts. Native denotes inference without HEC. HCER counts incorrect historical top-1 predictions on Food101 and samples with at least one historical false positive on MM-IMDb.

IMDb AP to 26.91, whereas HEC raises AP to 69.88 and 29.84, respectively. This comparison supports correcting historical evidence while retaining all learned classes rather than restricting prediction to one task partition before class comparison. Food101 FG rises from 8.02 to 12.67, whereas MM-IMDb FG falls from 31.18 to 29.93. The analysis below explains this difference.

Figure 4 reports the Historical-Class Error Rate (HCER): a wrong historical top-1 on Food101 or at least one historical false positive on MM-IMDb. At the final stage, HCER falls from 13.67% to 9.92% on Food101 and from 28.46% to 19.31% on MM-IMDb. The reduction on both datasets directly shows that HEC suppresses historical predictions that are not supported by the current input.

The FG values require a separate interpretation because FG measures the gap between an earlier peak and final performance, rather than final old-task performance alone. On MM-IMDb, with HEC disabled, UG-CFC raises the average historical peak of old tasks by 3.59 points and their final performance by 1.82 points. The 1.77-point difference accounts for the increase in FG from 29.41 to 31.18 even though final old-task performance improves. HEC then reduces FG to 29.93 while increasing AP to 29.84.

Food101 shows the complementary case. Relative to native all-head inference, HEC raises the average historical peak of old tasks by 5.06 points and their final performance by 0.41 points, yielding the 4.65-point FG increase from 8.02 to 12.67. The larger FG therefore comes from a larger rise in the historical peak, not from a decline in final old-task performance. This behavior is possible because HEC begins correcting an expert only after its task becomes historical; although expert parameters are frozen, the global evidence used for prediction can still change across stages.

AI USE STATEMENT

Generative AI tools were used for language polishing.

REFERENCES

John Edison Arevalo Ovalle, Thamar Solorio, Manuel Montes-y Gómez, and Fabio A. González. Gated multimodal units for information fusion. In *International Conference on Learning Representations (ICLR) Workshop*, 2017.

Zirun Guo, Shulei Wang, Wang Lin, Weicai Yan, Yangyang Wu, and Tao Jin. Efficient prompting for continual adaptation to missing modalities. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4317–4327, 2025.

Lianyu Hu, Tongkai Shi, Wei Feng, Fanhua Shang, and Liang Wan. Deep correlated prompting for visual recognition with missing modalities. In *Advances in Neural Information Processing Systems*, pp. 67446–67466, 2024.

Jaehyuk Jang, Yooseung Wang, and Changick Kim. Towards robust multimodal prompting with missing modalities. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8070–8074, 2024.

Wonjae Kim, Bokyung Son, and Ildoo Kim. ViLT: Vision-and-language transformer without convolution or region supervision. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 5583–5594, 2021.

Olivier Ledoit and Michael Wolf. A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, 88(2):365–411, 2004.

Yi-Lun Lee, Yi-Hsuan Tsai, Wei-Chen Chiu, and Chen-Yu Lee. Multimodal prompting with missing modalities for visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14943–14952, 2023.

Yan-Shuo Liang and Wu-Jun Li. InfLoRA: Interference-free low-rank adaptation for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 23638–23647, 2024.

Xiwei Liu, Yulong Li, Feilong Tang, and Imran Razzak. DeLo: Dual decomposed low-rank experts collaboration for continual missing modality learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 23855–23863, 2026.

Mengmeng Ma, Jian Ren, Long Zhao, Sergey Tulyakov, Cathy Wu, and Xi Peng. SMIL: Multimodal learning with severely missing modality. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 2302–2310, 2021.

Vittorio Pipoli, Federico Bolelli, Sara Sarto, Marcella Cornia, Lorenzo Baraldi, Costantino Grana, Rita Cucchiara, and Elisa Ficarra. Semantically conditioned prompts for visual recognition under missing modality scenarios. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 4968–4977, 2025.

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

Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. DualPrompt: Complementary prompting for rehearsal-free continual learning. In *European Conference on Computer Vision (ECCV)*, pp. 631–648, 2022a.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 139–149, 2022b.

Xianghu Yue, Yiming Chen, Xueyi Zhang, Xiaoxue Gao, Mengling Feng, Mingrui Lao, Huiping Zhuang, and Haizhou Li. PAL: Prompting analytic learning with missing modality for multimodal class-incremental learning. *Pattern Recognition*, 179:113467, 2026.

Zhihui Zhang, Luanyuan Dai, Qika Lin, Yunfeng Diao, Guangyin Jin, Yufei Guo, Jing Zhang, and Xiaoshuai Hao. Synergistic prompting for robust visual recognition with missing modalities. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1881–1890, 2025.

Jinming Zhao, Ruichen Li, and Qin Jin. Missing modality imagination network for emotion recognition with uncertain missing modalities. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 2608–2618, 2021.

Shu Zhao, Xiaohan Zou, Tan Yu, and Huijuan Xu. Reconstruct before query: Continual missing modality learning with decomposed prompt collaboration. *arXiv preprint arXiv:2403.11373*, 2024.

Shu Zhao, Nilesh Ahuja, Tan Yu, Tianyi Shen, and Vijaykrishnan Narayanan. MoRA: Missing modality low-rank adaptation for visual recognition. In *International Conference on Learning Representations*, 2026.

A LOCAL ALGEBRA OF THE CORRECTION TARGET

For $\rho > 0$, the matrix $\boldsymbol{Z}_{\mathrm{M}}\boldsymbol{Z}_{\mathrm{M}}^{\mathsf{T}} + \rho\boldsymbol{I}_r$ is positive definite, so the ridge problem in Equation 4.2 has a unique solution. Differentiating the objective with respect to $\boldsymbol{\Delta}$ gives

$$2(\boldsymbol{\Delta}\boldsymbol{Z}_{\mathrm{M}} - \boldsymbol{D})\boldsymbol{Z}_{\mathrm{M}}^{\mathsf{T}} + 2\rho\boldsymbol{\Delta} = \boldsymbol{0}. \tag{2}$$

Rearranging yields Equation 4.2, and right multiplication by $\boldsymbol{Z}_{\mathrm{M}}$ yields Equation 3. Let $\boldsymbol{Z}_{\mathrm{M}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathsf{T}}$ be its singular-value decomposition. On the subspace supported by $\boldsymbol{Z}_{\mathrm{M}}$, the eigenvalues of $\boldsymbol{P}_\rho$ are $\sigma_i^2/(\sigma_i^2 + \rho)$. The ridge target therefore spectrally shrinks the desired change according to the support provided by the Missing-view coordinates: strongly supported directions are largely retained, whereas weakly supported directions are attenuated more heavily. Consequently, the correction can reproduce only the component supported by the current Missing-view coordinates.

Right-multiplying Equation 4.2 by $\boldsymbol{Z}_{\mathrm{M}}$ gives

$$\boldsymbol{\Delta}^*\boldsymbol{Z}_{\mathrm{M}} = \boldsymbol{D}\boldsymbol{P}_\rho, \qquad \boldsymbol{P}_\rho = \boldsymbol{Z}_{\mathrm{M}}^{\mathsf{T}}(\boldsymbol{Z}_{\mathrm{M}}\boldsymbol{Z}_{\mathrm{M}}^{\mathsf{T}} + \rho\boldsymbol{I}_r)^{-1}\boldsymbol{Z}_{\mathrm{M}}. \tag{3}$$

Thus, the ridge solution does not match the Complete-view change $\boldsymbol{D}$ directly; it matches a spectrally shrunk component of $\boldsymbol{D}$ supported by the current Missing-view coordinates. Equation 4.2 then measures the effect of the operator error $\boldsymbol{R}_{\mathrm{M}} - \boldsymbol{R}_{\mathrm{T}}$ through both the native up-projection $\boldsymbol{B}$ and the actual Missing-view coordinates $\boldsymbol{Z}_{\mathrm{M}}$, rather than penalizing the raw operator distance alone. Consequently, two operator errors with the same Frobenius norm can receive different penalties if they induce different changes on the current Missing representation. In this sense, the ridge target determines which part of the correction can be represented from the Missing-view coordinates, while Equation 4.2 determines which operator errors affect the current projection site's LoRA output.

With the LoRA factors, intermediate features, and detached target held fixed, the CFC loss for one Missing branch has gradient

$$\frac{\partial\mathcal{L}_p}{\partial\boldsymbol{R}_{\mathrm{M}}} = \frac{2}{nd_{\mathrm{out}}}\boldsymbol{B}^{\mathsf{T}}\boldsymbol{B}(\boldsymbol{R}_{\mathrm{M}} - \boldsymbol{R}_{\mathrm{T}})\boldsymbol{Z}_{\mathrm{M}}\boldsymbol{Z}_{\mathrm{M}}^{\mathsf{T}}. \tag{4}$$

This gradient shows that the auxiliary supervision acts only on operator errors that are both activated by the current Missing-view coordinates $\boldsymbol{Z}_{\mathrm{M}}$ and capable of changing the current projection site's LoRA output through the up-projection $\boldsymbol{B}$. Errors lying in directions unsupported by $\boldsymbol{Z}_{\mathrm{M}}$ or in the output null space of $\boldsymbol{B}$ therefore receive no corresponding auxiliary gradient. CFC therefore optimizes operator differences that have functional consequences under the current missing representation, rather than the raw distance over the entire $r \times r$ operator space. Because the LoRA up-projection $\boldsymbol{B}$ is initialized at zero, this auxiliary gradient is also zero at the beginning of training; as the classification objective learns a nonzero native up-projection, the functional supervision from CFC becomes active.

Scope of the utility test. Let $\mathcal{R} = \{\boldsymbol{R}_u\}_u$ denote the actions for the Missing view at the surviving-modality sites and $\mathcal{V} = \{\boldsymbol{V}_u\}_u$ the corresponding negative CFC gradients. For fixed intermediate representations and differentiable downstream computation, an infinitesimal perturbation along $\mathcal{V}$ gives

$$\ell_{\mathrm{probe}}(\mathcal{R} + \alpha\mathcal{V}) = \ell_{\mathrm{probe}}(\mathcal{R}) + \alpha\sum_u \langle\boldsymbol{G}_u, \boldsymbol{V}_u\rangle_F + o(\alpha). \tag{5}$$

Therefore, when the inner-product sum in Equation 4.2 is negative, $\mathcal{V}$ is a first-order descent direction for the classification probe loss of the current Missing-view sample, and the utility gate admits the corresponding CFC supervision. This criterion is defined in action space, whereas optimization updates the shared controller parameters. Because the controller Jacobian couples multiple actions and AdamW applies a finite parameter step, the criterion characterizes the local classification utility of the auxiliary direction at the current actions rather than guaranteeing a loss decrease for every sample after the realized parameter update. The probe gradients, CFC directions, and admission decisions are treated as detached quantities, so no gradients are propagated through the admission test itself and no second-order optimization is required.

B IMPLEMENTATION AND PROTOCOL DETAILS

The frozen backbone is ViLT with 12 Transformer blocks and hidden dimension 768. Each block contributes query, key, and value projection sites, yielding 36 adapted projection sites in total; each

site contains separate rank-8 LoRA adapters for the text and image branches. Each task is associated with one conditional controller implemented as an MLP with two 256-dimensional GELU hidden layers. For a given sample and projection site, the controller input concatenates eight pooled rank coordinates, 768 normalized pooled hidden features, a 32-dimensional site embedding, an eight-dimensional modality embedding, and a three-dimensional availability-state indicator. The controller outputs 64 values that are reshaped into the 8×8 rank-space operator $\boldsymbol{R}_x$. Its final output layer is initialized to zero, so the conditional residual operator starts from $\boldsymbol{R}_x = \mathbf{0}$. The LoRA down-projections use Kaiming initialization, whereas the up-projections are initialized to zero. No additional LoRA scaling factor is applied.

Both datasets are trained for ten epochs per task. The batch size is set to four for training samples from the benchmark protocol and for evaluation. All main results use random seed 42. Optimization uses AdamW with an initial learning rate of 10^{-4}, weight decay of 0.01, $\beta_1 = 0.9$, and $\beta_2 = 0.98$, together with cosine learning-rate decay and linear warmup over the first 10% of training steps. Automatic mixed precision is enabled for standard forward and backward computation, whereas ridge-target construction and probe-gradient accumulation are performed in float32, and HEC covariance estimation and calibration use float64. For each Complete/Missing pair and adapted site, the ridge coefficient in Equation 4.2 is scaled to the current Missing representation as $\rho = 10^{-3}\,\mathrm{tr}(\boldsymbol{Z}_{\mathrm{M}}\boldsymbol{Z}_{\mathrm{M}}^{\mathsf{T}})/r + 10^{-8}$. Text sequences are truncated to a maximum length of 512 tokens. The configured positional capacities are 512 for Food101 and 1024 for MM-IMDb; when a larger positional capacity is required, the frozen positional embeddings are linearly interpolated. The 1024-position capacity used for MM-IMDb affects only the positional representation range and does not change the 512-token text truncation limit.

The benchmark manifest provides each record's task identifier, split, image, text, and label. Food101 partitions the classes into successive blocks of ten, whereas MM-IMDb introduces four labels per task. Missingness states for training, validation, and testing follow the assignment tables from the RebQ protocol (Zhao et al., 2024) for the corresponding splits. Missing text is represented by an empty string, and a missing image is replaced by the protocol-defined constant image; both are passed through the standard processor so that the expected input structure is preserved. Naturally incomplete examples never provide an unobserved counterpart and therefore do not supply hidden complete views for paired supervision. For every training example that is Complete under the benchmark protocol, one text-missing view and one image-missing view are additionally constructed. CFC uses only aligned valid tokens from the surviving modality in each complete–missing pair.

Training and inference controls. The CFC-weight-zero control retains the conditional adapter and the same Complete/Missing view construction but assigns zero optimization weight to CFC. Paired CFC without the gate optimizes all paired terms. UG-CFC further applies the utility gate and admits a correction only when the aggregated first-order directional derivative is strictly negative. Inference using all task experts evaluates every seen task expert and expresses class outputs using Equation 4.3. HEC uses the same candidate classes and evidence definition and changes only historical evidence, with no parameter update at inference. Task-Key hard routing follows DeLo (Liu et al., 2026): an exponential moving average of training queries forms each task key with decay 0.99, and the task with the largest cosine similarity is selected at test time. Same-label CFC replaces the complete counterpart of the current instance with a different complete instance carrying the same supervised label; for MM-IMDb, the entire multi-label vector must match. The utility gate is disabled for both Same-label CFC and paired CFC without the gate. Their comparison therefore isolates instance correspondence while retaining label correspondence and all other training settings.

C Support estimation and storage

Let $\boldsymbol{X}$ contain the class-centered residuals of all positive training examples for one task, with a multi-label example contributing one residual for each of its positive labels. After the pooled residuals are globally centered, the empirical covariance is

$$\boldsymbol{C} = \boldsymbol{X}^{\mathsf{T}}\boldsymbol{X}/n, \qquad \tau = \mathrm{tr}(\boldsymbol{C})/d. \tag{6}$$

Ledoit–Wolf shrinkage gives

$$\widehat{\boldsymbol{\Sigma}} = (1-\gamma)\boldsymbol{C} + \gamma\tau\boldsymbol{I}_d, \qquad \gamma \in [0,1]. \tag{7}$$

The implementation adds $100\epsilon_{64}\max(\mathrm{tr}(\widehat{\boldsymbol{\Sigma}})/d, 1)\boldsymbol{I}_d$ before computing the symmetric pseudoinverse. The shrinkage coefficient γ is estimated from the training residuals and is not tuned using test results. The current fitting procedure requires at least one positive training example and one positive validation example for every class. Calibration distances are computed from positive validation examples using the views specified by the benchmark protocol, and neither support fitting nor calibration uses test labels.

After each task, CHEC stores its class indices, class means, task-shared precision matrix, sorted per-class validation distances, and class counts. These support statistics are fixed after task completion and are not refitted at later stages. At inference, the Mahalanobis distance for class c of expert t is evaluated as

$$d_{t,c}(x) = \boldsymbol{f}_t(x)^\top \boldsymbol{\Omega}_t \boldsymbol{f}_t(x) - 2\boldsymbol{f}_t(x)^\top \boldsymbol{\Omega}_t \boldsymbol{\mu}_{t,c} + \boldsymbol{\mu}_{t,c}^\top \boldsymbol{\Omega}_t \boldsymbol{\mu}_{t,c}. \tag{8}$$

The final quadratic term is cached. The empirical upper-tail count uses the left insertion position in the sorted calibration list, so calibration distances equal to the query distance are included in the tail. Complete and incomplete observations share the same task covariance and per-class calibration distributions; HEC does not fit separate support statistics for different availability states. Each expert produces its own ViLT representation under its corresponding adapter and controller, and representations are not shared across experts during native all-expert inference.

D Diagnostic definitions and evaluation units

All diagnostics use the 70% Mixed missingness condition. HCER uses the formal test split and the same test examples for native and corrected evidence, whereas the functional-gap diagnostic uses validation examples only.

Historical-class error rate (HCER). After task k (indexed from zero), let $\mathcal{C}_{\le k}$ be the seen classes and $\mathcal{C}_{<k}$ the historical classes. Write $e_{i,c}^{(q)}$ for evidence under rule $q \in \{\mathrm{Native}, \mathrm{HEC}\}$. On Food101,

$$I_{i,k}^{(q)} = \mathbb{I}\left[\arg\max_{c \in \mathcal{C}_{\le k}} e_{i,c}^{(q)} \in \mathcal{C}_{<k} \setminus \{y_i\}\right], \qquad \mathrm{HCER}_k^{(q)} = \frac{1}{k+1}\sum_{j=0}^{k} \frac{\sum_{i \in \mathcal{D}_j^{\mathrm{test}}} I_{i,k}^{(q)}}{|\mathcal{D}_j^{\mathrm{test}}|}. \tag{9}$$

The event is a final top-1 prediction assigned to an incorrect historical class.

On MM-IMDb, let $\mathcal{H}_{i,k}^- = \{c \in \mathcal{C}_{<k} : y_{i,c} = 0\}$. Then

$$I_{i,k}^{(q)} = \mathbb{I}[\exists c \in \mathcal{H}_{i,k}^- : e_{i,c}^{(q)} > 0], \qquad \mathcal{E}_{k,j} = \{i \in \mathcal{D}_j^{\mathrm{test}} : |\mathcal{H}_{i,k}^-| > 0\},$$
$$\mathrm{HCER}_k^{(q)} = \frac{1}{|\mathcal{J}_k|}\sum_{j \in \mathcal{J}_k} \frac{\sum_{i \in \mathcal{E}_{k,j}} I_{i,k}^{(q)}}{|\mathcal{E}_{k,j}|}, \tag{10}$$

where $\mathcal{J}_k$ contains tasks with nonempty $\mathcal{E}_{k,j}$. The error rate is computed within each task and then macro-averaged across tasks. Both datasets have $\mathrm{HCER}_0 = 0$ because no historical class exists at the first stage.

Functional gap between Complete and Missing views. Every validation instance that is Complete under the benchmark protocol is included without random or class-balanced subsampling. Let N_C denote their total number. Each instance produces a text-missing view (TM, image survivor) and an image-missing view (IM, text survivor). For the valid tokens of the surviving modality aligned between the two views at site ℓ, define

$$\boldsymbol{F}_{i,\ell}^{\mathrm{C}} = \boldsymbol{B}_\ell(\boldsymbol{I} + \boldsymbol{R}_{i,\ell}^{\mathrm{C}})\boldsymbol{Z}_{i,\ell}^{\mathrm{C}}, \qquad \boldsymbol{F}_{i,\ell}^{M} = \boldsymbol{B}_\ell(\boldsymbol{I} + \boldsymbol{R}_{i,\ell}^{M})\boldsymbol{Z}_{i,\ell}^{M}, \tag{11}$$

where $M \in \{\mathrm{TM}, \mathrm{IM}\}$. Figure 3 reports

$$g_{i,\ell}^{M} = \frac{\|\boldsymbol{F}_{i,\ell}^{\mathrm{C}} - \boldsymbol{F}_{i,\ell}^{M}\|_F^2}{\|\boldsymbol{F}_{i,\ell}^{\mathrm{C}}\|_F^2 + \epsilon}, \qquad G_M = \frac{1}{36N_C}\sum_{i=1}^{N_C}\sum_{\ell=1}^{36} g_{i,\ell}^{M}, \tag{12}$$

where $\epsilon > 0$ stabilizes zero denominators. Each missing state contributes $36N_C$ sample–site observations. Normalization is performed before averaging. Complete and missing views use the same instance and its corresponding task expert.

Table 5: MM-IMDb at 70% Mixed missingness across three seeds. AP ↑ is final-stage task-mean macro-F1; FG ↓ is peak-based forgetting, both in percentage points. Bold marks the better result within each seed/metric. The final row gives mean ± sample standard deviation. Differences are CHEC minus DeLo, paired by seed.

	DeLo		**CHEC**		Paired difference	
Seed	AP	FG	AP	FG	ΔAP	ΔFG
17	28.88	31.71	**29.64**	**30.09**	+0.76	−1.62
42	28.97	31.84	**29.84**	**29.93**	+0.87	−1.91
73	29.11	32.03	**30.16**	**29.91**	+1.05	−2.12
Mean ± SD	28.99 ± 0.12	31.86 ± 0.16	$\mathbf{29.88 \pm 0.26}$	$\mathbf{29.98 \pm 0.10}$	$+0.89 \pm 0.15$	-1.88 ± 0.25

Table 6: Overall component ablation at 70% Mixed missingness. Food/MM denote Food101/MM-IMDb. AP uses task-mean accuracy on Food101 and task-mean macro-F1 on MM-IMDb; FG is peak-based forgetting. All rows retain the conditional adapter, and all values are percentage points. Bold marks the best result within each column.

Configuration	Food AP ↑	Food FG ↓	MM AP ↑	MM FG ↓
Neither component	61.73	10.57	26.05	29.41
UG-CFC only	68.50	**8.02**	27.87	31.18
HEC only	62.64	13.76	27.72	**28.83**
CHEC: both components	**69.88**	12.67	**29.84**	29.93

E Random-seed stability

Table 5 reports the paired three-seed MM-IMDb robustness check used in Section 5. Under 70% Mixed missingness, CHEC improves AP and reduces FG relative to DeLo for seeds 17, 42, and 73. The final row reports the mean and sample standard deviation across the paired runs.

F Overall component ablation

Table 6 provides an overall decomposition of the two top-level CHEC components under 70% Mixed missingness. Relative to retaining neither component, UG-CFC increases AP on both datasets, and HEC also increases AP when applied independently. Combining the training-side and inference-side components yields the highest AP on both Food101 and MM-IMDb. Their effects on peak-based FG differ across the two datasets, consistent with the targeted analyses in the main text.