Disentangling Channel Semantics in Vision Transformers via Token Decorrelation and Composition-Aware Modulation

Daeun Kim[1], **Hyesong Choi**[2], **Hyejin Park**[1,†], **Dongbo Min**[1,†]
[1]Division of Artificial Intelligence & Software, Ewha Womans University
[2]Soongsil University
{daeunkim00,clrara,dbmin}@ewha.ac.kr
hyesong@ssu.ac.kr

Abstract

Multi-channel images encode heterogeneous channel semantics aligned over a shared spatial structure, which requires models to jointly capture globally shared structure and channel-specific variations. Recent multi-channel Vision Transformers attempt to address this challenge by augmenting the [CLS] token with memory tokens to organize multi-channel representations under varying channel compositions. However, we find that these context tokens collapse onto a few dominant channels under high inter-channel redundancy, producing redundant representations and a biased global summary that fails to reliably support composition-aware channel interpretation. We propose **MuCa-ViT (Multi-Channel Composition-Aware Vision Transformer)**, a unified framework that addresses these limitations through two coupled mechanisms. **Factorial Token Learning (FTL)** enforces token-level decorrelation among context tokens, encouraging the [CLS] token to capture globally shared semantics while memory tokens preserve complementary channel-specific cues. **Channel Composition-Aware Modulation (CAM)** uses the FTL-refined [CLS] as a semantic anchor to generate sample-wise modulation signals, which enable composition-aware interpretation of channels within a unified backbone rather than relying on static channel embeddings. Experiments on JUMP-CP, CHAMMI, and So2Sat show that MuCa-ViT outperforms prior multi-channel ViTs across all three benchmarks, and the magnitude of improvement aligns with the inter-channel redundancy of each dataset ($\rho \in [0.19, 0.77]$), validating the diagnostic analysis that motivates our design. Our code is available on GitHub.

1 Introduction

While Vision Transformers (ViTs) [9, 15] have advanced natural image analysis through their ability to capture long-range dependencies, their extension to multi-channel imaging introduces fundamental challenges. Unlike RGB images, whose channels exhibit strong spectral correlation and jointly encode a single visual object, multi-channel images encode heterogeneous channel semantics aligned over a shared spatial structure. Effectively modeling such data requires jointly learning global representations and fine-grained, channel-specific semantic factors. In real-world biomedical imaging and remote sensing applications, the number and type of available channels vary across samples due to acquisition protocols, sensor heterogeneity, or missing modalities, further increasing the complexity of representation learning. Consequently, multi-channel ViTs must handle inputs in which heterogeneous channel-specific information coexists on top of shared structural information, while organizing such partially observed inputs into a coherent internal representation.

[†]Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Existing multi-channel ViTs typically encode channel identity using static channel embeddings while serializing channel-wise patches into a unified token sequence [18, 4, 19]. To further aggregate multi-channel information, recent approaches introduce additional memory tokens alongside the `[CLS]` token [18]. We collectively refer to these tokens as **Context Semantic Tokens (CSTs)**. However, whether globally shared and channel-specific semantics are separately captured across these tokens, and whether channel representations adapt to the observed channel composition, remain underexplored.

To examine these questions, we identify two co-occurring limitations in existing multi-channel ViTs. First, in the absence of any structural constraint on CSTs, both the [CLS] token and memory tokens collapse onto a small subset of dominant channels, producing highly redundant representations. This pattern arises because multi-channel inputs contain both globally shared structure and channel-specific information. Under high inter-channel redundancy, the model learns a shortcut by relying on a few channels with the strongest signal rather than attending to channel-specific cues. As a result, the global representation becomes biased toward dominant channels and no longer reliably reflects the full channel composition of the input. Second, existing models lack a mechanism to interpret each channel relative to the observed channel composition. The semantic role of a channel depends on which other channels are present, yet static channel embeddings remain fixed regardless of the input composition. Combined with the collapse of CSTs onto dominant channels, this absence of composition-aware interpretation leaves channel features entangled around a few dominant channels rather than separated into channel-specific factors.

To address these two limitations, we propose **MuCa-ViT** (Multi-Channel Composition-Aware Vision Transformer), a unified framework built on two design choices that have not been jointly considered in prior multi-channel ViTs. First, we introduce **Factorial Token Learning (FTL)**, which enforces explicit token-level decorrelation among context tokens to encourage auxiliary memory tokens to preserve complementary information rather than inheriting the channel preferences of the [CLS] token. This suppresses redundancy among CSTs and yields a reliable global summary. Second, we introduce Channel **Composition-Aware Modulation (CAM)**, which complements static channel identity encoding with conditioning derived from the input channel composition, allowing the same channel to be interpreted differently depending on the observed channel subset. By using the refined [CLS] token as a semantic anchor, CAM enables composition-aware interpretation of channel features. When combined, FTL refines the conditioning signal that CAM relies on, yielding larger gains than either component alone, particularly under high inter-channel redundancy.

Our contributions are summarized as follows:

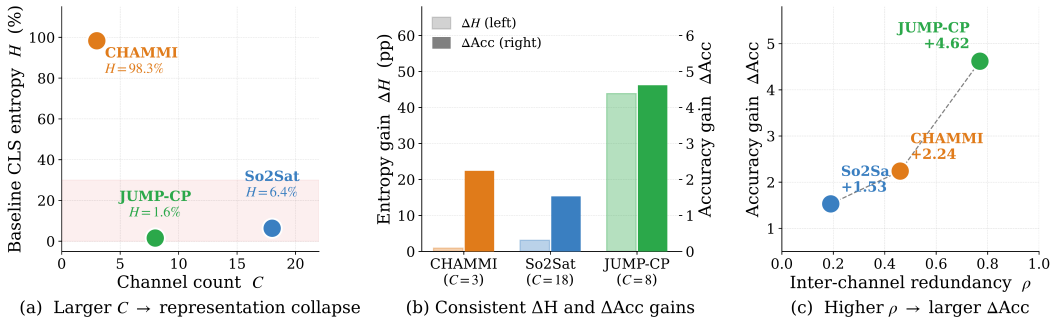
(i) Diagnostic analysis of multi-channel ViTs. We analyze how inter-channel redundancy promotes context token collapse onto dominant channels in multi-channel ViTs, and how this collapse undermines composition-aware channel interpretation.

(ii) Principled complementary design of FTL and CAM. Motivated by our analysis, we propose MuCa-ViT, a framework with a novel complementary design that addresses collapse at two coupled levels.

(iii) Generalizable Improvements. We demonstrate consistent improvements across three multi-channel benchmarks and two backbone architectures, with the magnitude of each gain quantitatively tracking inter-channel redundancy as predicted by our analysis.

2 Related Works

Vision Transformers for Multi-Channel Imaging. Modeling multi-channel biomedical imaging requires handling heterogeneous channel configurations and varying channel availability. Early approaches to channel adaptation included template-mixing methods that construct channel-specific input projections from shared parameter templates [21, 23]. With Vision Transformers (ViTs), subsequent works serialized channel-wise patches into a unified token sequence and introduced learnable channel embeddings to preserve channel identity [2]. To improve robustness under irregular channel compositions, DiChaViT [19] incorporated diverse channel sampling and orthogonality constraints to promote channel-wise diversity. More recently, pretraining-based strategies have been explored. ChA-MAEViT [18] framed channel dropping as a masked reconstruction problem within a joint MAE-classification framework [14, 1] and introduced memory tokens linked to the `[CLS]` token.



(a) Larger $C \rightarrow$ representation collapse (b) Consistent ΔH and ΔAcc gains (c) Higher $\rho \rightarrow$ larger ΔAcc

Figure 1: **Representation collapse and its mitigation across datasets.** Δ values are computed against our reproduced ChA-MAEViT [18]. (a) **CLS channel coverage across datasets** ($H_{\texttt{CLS}}\%$, y-axis). Near-max on CHAMMI but drops sharply on JUMP-CP and So2Sat, indicating that CLS coverage concentrates on few channels as C grows. (b) **Consistent recovery.** Our method yields positive $\Delta H_{\texttt{CLS}}\%$ and ΔAcc on all three datasets. (c) **Larger gains in higher-redundancy regimes.** ΔAcc is larger on benchmarks with higher inter-channel redundancy ρ, consistent with redundancy being a major structural factor that promotes CST collapse.

IC-ViT [13] pretrains on single-channel inputs and integrates cross-channel dependencies during fine-tuning. Despite these advances, existing approaches primarily operate at the input, embedding, or supervision level. The internal token space remains implicitly structured, and the interaction between global context and channel-specific semantics within token-based architectures therefore remains underexplored.

Conditional Feature Modulation. Feature-wise conditioning adapts representations via affine transformations driven either by external signals such as language embeddings, timesteps, or class labels [17, 8, 16, 30, 11], or by the model's own intermediate features [3, 5]. In both cases, the conditioning signal is used as given. Extending this to multi-channel ViTs requires care with the conditioning signal itself: the `[CLS]` token is a natural internal candidate, but already reflects dominant-channel bias (Sec. 3), and an effective signal should be interpreted jointly with the input channel configuration. This motivates calibrating the conditioning signal with respect to the channel configuration before modulation, rather than applying it on uncorrected context tokens.

3 Problem Formulation and Analysis

Preliminary. Multi-channel images consist of C channels that share a common spatial structure, where each channel encodes heterogeneous semantic information. Existing ViT-based architectures typically adopt an early-fusion strategy, in which each channel $c \in \{1, \ldots, C\}$ is independently patchified into P tokens and all tokens are concatenated into a single sequence of length $C \times P$ [2, 4, 19]. To preserve channel identity, a learnable channel embedding is added to each token. To summarize channel-wise information, ViT-based architectures employ a global `[CLS]` token. In practical training protocols, channel composition is often varied by sampling different subsets of channels during training, resulting in dynamically changing channel configurations across iterations.

To improve robustness under such varying channel compositions, some recent multi-channel extensions introduce additional memory tokens that interact with patch tokens within the encoder to aggregate information across channels [18]. In this work, we collectively refer to the `[CLS]` token and memory tokens as **Context Semantic Tokens (CSTs)**, which serve as compact representations of global and channel-specific semantics.

Channel Dominance in CSTs. CSTs are intended to organize global and channel-specific information across the `[CLS]` token and auxiliary memory tokens, going beyond the limited capacity of a single global representation. Under this design, the `[CLS]` token is expected to capture global semantics that support varying channel compositions and are forwarded to the prediction head, while memory tokens preserve complementary channel-specific cues.

To examine whether this design is actually realized in practice, or whether CSTs instead concentrate on a narrow subset of channels and undermine their intended role, we measure how broadly channels

are covered within the CSTs at two complementary units, namely the [CLS] token and the CST set as a whole. We analyze token interactions through similarity in the final feature space, since this representation is what is directly forwarded to the prediction head and therefore most relevant to downstream behavior. Let $\mathbf{Z} \in \mathbb{R}^{(C\times P)\times D}$ denote the final representation of patch tokens at the encoder output, where $\mathbf{Z}_{c,p}$ is the p-th token of channel c. For a query token $\mathbf{q}$ (either the [CLS] token or a memory token), we form a softmax distribution over the full set of $C \times P$ patch tokens based on dot-product alignment, marginalize over patches within each channel, and compute the resulting channel coverage entropy:

$$\alpha_{c,p}(\mathbf{q}) = \frac{\exp\langle \mathbf{q}, \mathbf{Z}_{c,p}\rangle}{\sum_{c',p'} \exp\langle \mathbf{q}, \mathbf{Z}_{c',p'}\rangle}, \qquad \tilde{S}_c(\mathbf{q}) = \sum_{p=1}^{P} \alpha_{c,p}(\mathbf{q}), \qquad H(\mathbf{q}) = -\frac{1}{\log C} \sum_{c=1}^{C} \tilde{S}_c(\mathbf{q}) \log \tilde{S}_c(\mathbf{q}). \tag{1}$$

The first term $\alpha_{c,p}(\mathbf{q})$ is a distribution over patches, and $\tilde{S}_c(\mathbf{q})$ aggregates patch-level attention into a channel-wise marginal representing the total attention mass that $\mathbf{q}$ places on channel c, where dominant channels correspond to large $\tilde{S}_c(\mathbf{q})$. The entropy $H(\mathbf{q}) \in [0, 1]$ approaches 1 when channels contribute evenly and 0 when a few channels dominate. We instantiate $H(\mathbf{q})$ at two complementary scopes.

(i) Token-level coverage of [CLS] : $H_{\texttt{CLS}} = H(q = \texttt{[CLS]})$ which captures channel dominance from the perspective of the global summary token forwarded to the prediction head.

(ii) Set-level coverage of CSTs : Let $\mathcal{T} = \{q_0, q_1, \ldots, q_K\}$, where $q_0 = \texttt{[CLS]}$ and $q_1, \ldots, q_K$ are memory tokens. We aggregate the per-CST channel marginals and compute the entropy of their average:

$$\bar{S}_c = \frac{1}{|\mathcal{T}|} \sum_{q\in\mathcal{T}} \tilde{S}_c(q), \qquad H_{\text{CST}} = -\frac{1}{\log C} \sum_{c=1}^{C} \bar{S}_c \log \bar{S}_c. \tag{2}$$

A high H_{CST} indicates that the CSTs *as a group* attend broadly across channels, while a low value indicates that they concentrate on the same dominant channels as [CLS]. We use H_{CST} to assess whether memory tokens fulfill their intended role under missing-channel scenarios, where preserving information requires the CSTs to collectively cover the channel set. Since H_{CST} measures collective coverage rather than redundancy among individual tokens, we complement it with Gram-matrix redundancy and per-CST attention analyses in Sec. F, which directly diagnose whether memory tokens replicate the channel preferences of [CLS]. We report $H_{\texttt{CLS}}$ and H_{CST} as percentages of the maximum entropy $\log C$, denoted $H_{\texttt{CLS}}\%$ and $H_{\text{CST}}\%$.

Channel Dominance and Inter-Channel Redundancy.

Our analysis reveals that channel dominance emerges broadly in existing multi-channel ViTs (Fig. 1(a)). On JUMP-CP and So2Sat, $H_{\texttt{CLS}}$ drops to 1.6% and 6.4%, respectively, indicating that the global summary token is heavily concentrated on a small subset of channels rather than capturing diverse channel-wise information across the input. In contrast, CHAMMI ($C \in \{3, 4\}$) exhibits substantially weaker collapse with $H_{\texttt{CLS}} = 98.3\%$, suggesting that a smaller channel set imposes a weaker structural bottleneck on the global token. This indicates that $H_{\texttt{CLS}}$ alone is insufficient to characterize multi-channel representation quality, since channel-specific information may remain entangled with shared structure even when token-level coverage appears intact.

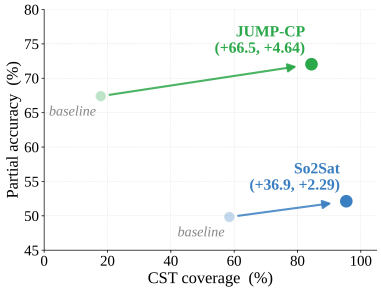


Figure 2: Restoring CST coverage ($\mathcal{H}_{\text{CST}}\%$, x-axis) drives partial-channel performance.

Channel dominance is not merely a representational phenomenon but is also associated with downstream performance degradation. As shown in Fig. 1(b), mitigating collapse improves both entropy and classification accuracy on all three datasets, establishing a direct link between representation quality and downstream task performance. While $H_{\texttt{CLS}}$ diagnoses channel dominance at the representational level, whether the model truly captures distinct channel-specific information requires examining the underlying channel structure.

We next examine when channel dominance arises and what governs the extent of recoverable performance. Multi-channel data inherently couples globally shared structure and channel-specific information over the same spatial layout, which makes their separation a defining challenge of

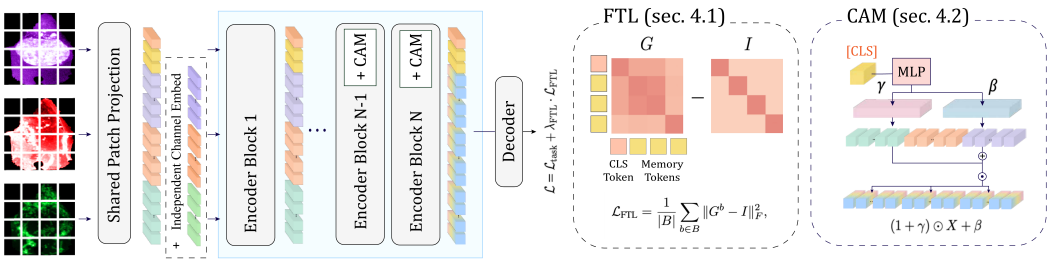


Figure 3: **Overall architecture of MuCa-ViT.** Multi-channel images are patchified using a shared projection and serialized with independent channel embeddings. Factorial Token Learning (FTL) enforces structured decorrelation among Context Semantic Tokens (CSTs), consisting of the global `[CLS]` token and memory tokens, via a token-wise Gram matrix constraint in (3). Channel Composition-Aware Modulation (CAM) uses the FTL-refined `[CLS]` token as a semantic anchor to generate sample-wise modulation parameters (γ, β), adaptively aligning patch representations under varying channel configurations.

multi-channel representation learning. When inter-channel redundancy is high, channels carry largely overlapping content, so the model can rely on a small set of dominant channels as a shortcut, while fine-grained channel-specific cues are easily suppressed. Recovering these suppressed cues therefore requires not only mitigating channel dominance, but also a mechanism that interprets each channel as a distinct factor at the representation level, so that channel-specific information can be separated rather than absorbed into the shared structure. We quantify inter-channel redundancy ρ as the off-diagonal mean of the per-image channel-wise cosine similarity matrix (formal definition in Appendix E). As shown in Fig. 1(c), the magnitude of the performance gain scales with ρ, while remaining positive across the full redundancy spectrum spanned by three benchmarks ($\rho \in [0.19, 0.77]$).

We further examine whether memory tokens fulfill the role assumed in our earlier discussion, namely distributing channel-wise information across multiple tokens, as quantified by the collective coverage $\mathcal{H}_{\text{CST}}$ in Eq. (2). As shown in Fig. 2, baseline CSTs exhibit $\mathcal{H}_{\text{CST}}$ substantially below the maximum on both JUMP-CP (17.87%) and So2Sat (58.48%), indicating that the CST set as a whole covers only a limited subset of the channel composition. This collective coverage deficit suggests that memory tokens do not specialize on distinct channels but instead inherit the dominance pattern of the `[CLS]` token. Sec. F further supports this view at the individual-token level on JUMP-CP, where both the `[CLS]` and memory tokens align with the same subset of dominant channels.

Taken together, these findings reveal a structural limitation of existing CSTs. The intended decomposition, where the [CLS] token captures globally shared semantics and memory tokens preserve diverse channel-wise cues, is not realized in practice. Instead, both token types collapse onto the same dominant channels, leaving channel-specific information entangled with the shared structure rather than disentangled into distinct channel factors. Effective multi-channel modeling therefore requires both explicit separation of channel information across CSTs and an adaptive mechanism that interprets each channel under the observed channel composition.

4 MuCa-ViT: Multi-Channel Composition-Aware Vision Transformer

As analyzed in Sec. 3, multi-channel ViTs exhibit two coupled limitations: (i) context tokens collapse onto dominant channels, producing redundant CSTs that fail to capture distinct channel-specific information, and (ii) the model lacks a mechanism to interpret channel semantics under varying channel compositions, leaving channel-specific information entangled with the shared structure. These limitations prevent the model from maintaining diverse channel information and adapting representations to dynamic input configurations. To address them, we propose **MuCa-ViT**, a unified framework consisting of two complementary components: **Factorial Token Learning (FTL)** and **Channel Composition-Aware Modulation (CAM)**. FTL suppresses redundancy among CSTs to recover distinct channel-specific information across tokens, while CAM leverages this refined representation to disentangle channel semantics from the shared structure under the observed channel composition.

4.1 Factorial Token Learning (FTL)

As discussed in Sec. 3, existing multi-channel ViTs suffer from channel dominance in which the [CLS] token and memory tokens align to a small subset of dominant channels, resulting in highly redundant Context Semantic Tokens (CSTs). This collapse undermines the role that CSTs would ideally play as a structured representation space, where global shared semantics and channel-specific information are distributed across multiple tokens. To address this limitation, we propose **Factorial Token Learning (FTL)**, which explicitly enforces token-level decorrelation by suppressing redundancy across the CST set. The key idea is to discourage different context tokens from encoding overlapping information, thereby encouraging them to organize complementary channel-specific semantic factors across CSTs. In this way, FTL restores the decomposition that CSTs would ideally support: the [CLS] token captures globally shared semantics, while memory tokens preserve complementary channel-specific cues.

To quantify redundancy among CSTs, we consider the set of context tokens $\mathrm{CST} \in \mathbb{R}^{(K+1)\times D}$, where K is the number of memory tokens. We compute a token-wise Gram matrix $G_{ij} = \frac{1}{D}\langle \mathrm{CST}_i, \mathrm{CST}_j \rangle$, where large off-diagonal values indicate overlapping semantic information across tokens, and define a decorrelation objective:

$$\mathcal{L}_{\text{FTL}} = \frac{1}{|B|} \sum_{b \in B} \|W \odot (G^b - I)\|_F^2, \qquad W = \begin{bmatrix} 1 & \mathbf{1}^\top \\ \mathbf{1} & W' \end{bmatrix}, \qquad W'_{mn} = \begin{cases} 1, & m = n \\ \alpha, & m \neq n \end{cases} \tag{3}$$

where I is the identity matrix, G^b is the Gram matrix of sample b (batch-normalized beforehand), and $\alpha = 0.1$ unless otherwise specified. This asymmetric weighting reflects the asymmetry in how task supervision propagates through the context tokens. The [CLS] token directly feeds into the prediction head and therefore receives the strongest task-loss gradient, which makes it most susceptible to absorbing dominant-channel signals as a shortcut. By enforcing decorrelation between [CLS] and memory tokens, FTL pushes channel-specific information away from [CLS] into the memory tokens, allowing [CLS] to attend more evenly across channels rather than collapsing onto the dominant ones. We thus prioritize [CLS]–memory decorrelation as the primary target of factorization, while treating memory–memory decorrelation as a secondary constraint to reduce local redundancy without suppressing useful cross-channel correlations, as validated in Appendix B.2.1. FTL is applied to the final encoder output and combined with the task loss $\mathcal{L} = \mathcal{L}_{\text{task}} + \lambda_{\text{FTL}}\, \mathcal{L}_{\text{FTL}}$.

By suppressing redundancy across the CST set, FTL mitigates the channel dominance identified in Sec. 3 and recovers distinct channel-specific information at the token level. However, distributing information across CSTs alone does not disentangle channel-specific information from the shared structure at the feature representation level. Achieving this further requires adapting the patch representations themselves to the observed channel composition, which we address with CAM in Sec. 4.2.

4.2 Channel Composition-Aware Modulation (CAM)

While FTL recovers channel-specific information at the token level by suppressing redundancy across the CST set, it does not transform the patch representations themselves. The semantic role of each channel is inherently composition-dependent, and disentangling channel-specific information from the shared structure requires adapting the patch representations to the observed channel composition (Sec. 3).

Existing approaches rely on fixed channel embeddings, which treat channels independently of the input composition and fail to capture such contextual variations. In contrast, the [CLS] token aggregates globally shared semantics across varying channel configurations. However, its direct exposure to the task-loss gradient biases it toward dominant channels (Sec. 3), and only after FTL suppresses this bias does it serve as a reliable conditioning source. To this end, we propose **Channel Composition-Aware Modulation (CAM)**, which uses the FTL-refined [CLS] token as a global semantic anchor to interpret the input channel configuration and generate composition-aware, sample-adaptive modulation parameters for the patch representations.

For a multi-channel image, we denote the patch tokens of channel $c \in \{1, \ldots, C\}$ as $\mathbf{X}_c \in \mathbb{R}^{P\times D}$, and stack them into $\mathbf{X} \in \mathbb{R}^{C\times P\times D}$. A common strategy [2, 19, 18] injects channel identity via channel embeddings, $\mathbf{Z}^{\text{in}}_{patch} = \mathbf{X} + \mathbf{e}$, where $\mathbf{e} \in \mathbb{R}^{C\times D}$ is broadcast across patches. While this

preserves channel identity, it does not capture how each channel should be interpreted under the observed channel composition.

To address this limitation, CAM generates modulation parameters from the [CLS] token at the kth encoder block:

$$(\gamma^k, \beta^k) = f_{\mathrm{CAM}}(\texttt{[CLS]}^k), \tag{4}$$

which are applied to all patch tokens. $f_{\mathrm{CAM}}(\cdot)$ denotes a lightweight conditioning network implemented as a two-layer MLP with a SiLU activation [22] in between. Given the patch token set $\mathbf{Z}^k$, we perform feature-wise affine transformation:

$$\mathbf{Z}^k_{patch} \leftarrow (1 + \gamma^k) \odot \mathbf{Z}^k_{patch} + \beta^k, \quad \mathbf{Z}^k_{patch} \in \mathbb{R}^{(C\times P)\times D}, \tag{5}$$

where $\odot$ denotes element-wise multiplication. In CAM, the modulation parameters (γ, β) are generated in a sample-specific manner from the global context token [CLS]. This self-conditioned formulation allows the model to adapt feature transformations for each input based on its global semantics. Importantly, since the [CLS] token reflects the active channel composition, the resulting modulation parameters are not only instance-specific but also composition-aware. We empirically validate this design choice in Appendix B.3.2. By conditioning on this global representation, CAM adapts feature transformations under each input configuration, enabling composition-aware channel interpretation that disentangles channel-specific information from the shared structure (Sec. 5.3). As a result, the same channel can be interpreted differently under varying channel subsets.

Coupled Design of FTL and CAM. FTL and CAM are inherently coupled, operating at complementary levels. FTL recovers channel-specific information at the token level by mitigating channel dominance, while CAM transforms the patch representations themselves under the observed channel composition. Without FTL, the [CLS] token remains biased toward dominant channels, and CAM inherits this bias when generating its modulation parameters. Conversely, without CAM, recovering information across CSTs alone does not transform the patch representations to disentangle channel-specific information from the shared structure. Together, FTL and CAM address the two coupled limitations identified in Sec. 3, jointly enabling channel-specific information to be both distributed across CSTs and disentangled from the shared structure (Sec. 5.3).

5 Experiments

Datasets. We evaluated our method on three publicly available multi-channel benchmarks: **JUMP-CP** [6], an 8-channel microscopy dataset comprising five fluorescence and three brightfield channels, with 161 classes (160 perturbations and one control); **CHAMMI** [7, 27, 24, 28], an integration of three microscopy datasets (3–5 channels) for domain generalization evaluated by the CHAMMI Score; and **So2Sat** [32], an 18-channel SAR and multispectral remote sensing dataset with 17 climate zone classes.

Implementation Details. We employ a ViT-Small backbone [9] trained for 100 epochs on top of ChA-MAEViT [18] and IC-ViT [13], with $\lambda_{\mathrm{FTL}} = 0.1\mathrm{e}{-5}$ and CAM applied to the final Transformer block (last two blocks for JUMP-CP). Full configurations are provided in Appendix A.

5.1 Main Results on Multi-Channel Benchmarks

A clear performance pattern emerges across datasets when compared to our reproduced ChA-MAEViT*. Our analysis in Sec. 3 identifies inter-channel redundancy ρ (see Appendix E) as the factor governing how much channel-specific information has been suppressed and is therefore available for recovery. Consistent with this prediction, gains scale with ρ across all three benchmarks (Fig. 1(c)).

The three datasets each probe a distinct regime of our analysis. JUMP-CP ($\rho = 0.77$, $H_{\texttt{CLS}} = 1.6\%$) exhibits the most severe channel dominance and yields the largest gain, where both FTL and CAM operate on heavily collapsed representations. CHAMMI ($\rho = 0.46$) presents a complementary case. Although its $H_{\texttt{CLS}}$ is already saturated, MuCa-ViT still yields a $+2.24$ pp improvement, demonstrating that channel-specific information remains entangled with the shared structure at the representation level even when token-level coverage is fully achieved. This regime, where token-level diagnostics alone cannot reveal the underlying entanglement, is precisely where representation-level

Table 1: Comparison of test accuracy (%) of channel-adaptive models. **Full** denotes inference using all channels, while **Partial** evaluates on a subset of channels. All results are reported as the mean ± standard deviation of three independent runs ($n = 3$). Values marked with * were reproduced using the official implementation.

Model	**CHAMMI** [7]	**JUMP-CP** [6]		**So2Sat** [32]	
	Avg score	**Full**	**Partial**	**Full**	**Partial**
HyperNetViT [10]	56.08 ± 0.41	47.07 ± 0.47	42.43 ± 0.65	60.73 ± 0.24	41.88 ± 0.85
DepthwiseViT [7]	61.80 ± 0.43	49.86 ± 0.45	44.98 ± 0.71	60.41 ± 0.22	43.41 ± 1.10
TemplateMixingViT [20, 21, 23]	58.16 ± 0.42	52.48 ± 0.27	43.85 ± 0.73	55.86 ± 0.10	37.28 ± 0.34
ChAda-ViT [4]	63.93 ± 0.42	65.03 ± 0.98	42.15 ± 2.33	56.98 ± 0.46	12.38 ± 2.03
ChannelViT [2]	64.97 ± 0.58	67.51 ± 0.35	56.49 ± 0.53	61.03 ± 0.17	46.16 ± 0.40
DiChaViT [19]	69.77 ± 0.44	69.19 ± 0.47	57.98 ± 0.41	63.36 ± 0.11	47.76 ± 0.23
ChA-MAEViT [18]	74.63 ± 0.54	90.73 ± 0.14	68.05 ± 0.21	67.44 ± 0.38	52.11 ± 0.49
ChA-MAEViT* [18]	73.95 ± 0.54	90.52 ± 0.40	67.40 ± 0.65	66.44 ± 0.74	49.83 ± 1.25
MuCa-ViT	$\mathbf{76.19 \pm 0.37}$	$\mathbf{95.14 \pm 0.32}$	$\mathbf{72.04 \pm 0.14}$	$\mathbf{67.97 \pm 0.68}$	$\mathbf{52.12 \pm 0.95}$

disentanglement (Sec. 5.3) becomes essential. So2Sat ($\rho = 0.19$) shows a smaller yet positive gain (+1.53 pp), confirming that the mechanism remains effective at the low-redundancy end. The same recovery is visible at the token level. MuCa-ViT raises H_{CST} from 17.87% to 84.41% on JUMP-CP and from 58.48% to 95.41% on So2Sat (Fig. 2), and the partial-channel gains in Table 1 reflect the joint effect of this restored coverage and CAM's composition-aware adaptation.

5.2 Ablation Study

Ablation on FTL and CAM. Table 2 evaluates the individual contributions of FTL and CAM on JUMP-CP. FTL alone provides moderate improvements through token-level decorrelation, addressing channel dominance across CSTs. CAM alone improves performance by introducing composition-aware modulation at the representation level. However, the combination of FTL and CAM yields substantially higher performance than the sum of their individual gains, indicating that CAM benefits substantially from the conditioning signal refined by FTL.

Table 2: Component ablation on JUMP-CP. CAM is applied to the final encoder block only.

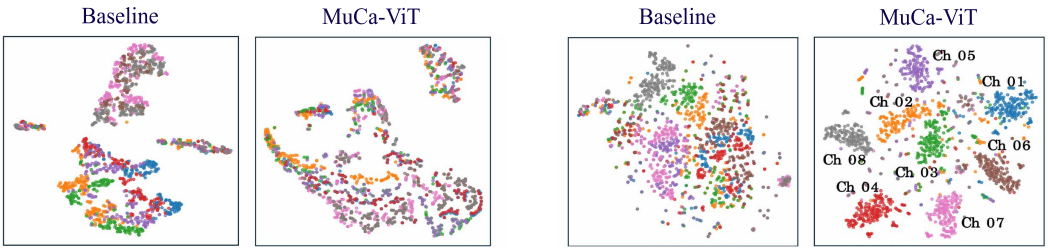
Model	**Full**	**Partial**
ChA-MAEViT	90.35	68.10
+ FTL	91.58	68.23
+ CAM	92.46	70.96
+ FTL + CAM	**94.35**	**71.81**

This confirms that the two modules address the two coupled limitations identified in Sec. 3 at complementary levels: FTL recovers channel-specific information at the token level, while CAM leverages this refined representation to disentangle channel-specific information from the shared structure at the feature representation level. Following the observation in CaiT [25] that the [CLS] token reaches semantic maturity in deeper blocks, we apply CAM to the later encoder blocks, where adaptation benefits more from composition-aware interpretation. We provide a detailed analysis of CAM's optimal insertion depth in Appendix B.3.3 and Appendix B.3.4.

Validation of CAM. While CAM improves performance, we further verify that it performs composition-aware modulation rather than generic feature scaling. As shown in Appendix Fig. 7, the modulation parameters cluster according to channel configurations rather than sample identity, indicating that CAM adapts feature transformations based on channel composition. This behavior provides direct evidence that CAM adapts feature transformations based on the underlying channel configuration. Together with the ablation results, this confirms that CAM functions as a composition-aware conditioning mechanism, complementing FTL in addressing the coupled limitations identified in Sec. 3.

5.3 Spectral Analysis on Channel Disentanglement

While Sec. 3 analyzes channel dominance through token-level coverage, here we examine whether channel-specific information is also disentangled from the shared structure at the feature representation



(a) Top Spectral Reconstruction (b) Middle Spectral Reconstruction

Figure 4: SVD-based subspace analysis of multi-channel representations. The 384-dimensional singular spectrum is partitioned into 55 groups of 7 consecutive singular vectors each. **(a)** t-SNE of $\mathbf{Z}^N_{\text{top}}$ (Group 1, d_{top}=7), capturing globally shared channel semantics. **(b)** t-SNE of $\mathbf{Z}^N_{\text{mid}}$ (Groups 3–35, d_{mid1}=15, d_{mid2}=245), corresponding to residual semantic components. In the middle spectral space, the baseline collapses into an entangled representation, whereas MuCa-ViT preserves clear channel-wise clustering.

Table 3: Plug-in compatibility of FTL and CAM on IC-ViT [13] and DiChaViT [19]. JUMP-CP is reported under full and partial channel settings. Models with superscript *mem* denote architectures augmented with memory tokens using the same initialization as MuCa-ViT.

Model	**Full**	**Partial**	**Model**	**Full**	**Partial**
IC-ViT^{mem}	86.68	66.41	DiChaViT^{mem}	64.83	54.69
+ FTL + CAM	**87.83**	**68.50**	+ FTL + CAM	**68.09**	**57.37**

level. Multi-channel inputs couple globally shared structure and channel-specific semantics over the same spatial layout, and an effective multi-channel representation should separate these two factors rather than absorbing channel-specific information into the shared component. To examine how these factors are organized in the learned representation space, we perform a spectral analysis based on singular value decomposition (SVD), with detailed formulation provided in Appendix D.

As shown in Fig. 4, the top spectral component (a) exhibits substantial overlap across channels for both the baseline [18] and our method, indicating that it primarily captures globally shared cross-channel structure. In contrast, the middle spectral component (b) reveals a clear difference: the baseline remains entangled across channels, whereas MuCa-ViT produces well-separated channel-wise clusters. This provides direct evidence that MuCa-ViT addresses the second limitation identified in Sec. 3, namely the entanglement of channel-specific information with the shared structure, while the top spectral component remains shared as expected. This shared-private organization aligns with the factorization observed in multimodal representation learning [26, 12], where shared and modality-specific factors are explicitly disentangled. Our results indicate that the coupled design of FTL and CAM induces this structure: FTL recovers channel-specific information across CSTs and refines the conditioning signal, which CAM then leverages to disentangle channel-specific information from the shared structure at the representation level.

5.4 Plug-in Compatibility Across Architectures

We evaluate the plug-in compatibility of FTL and CAM by integrating them into IC-ViT [13] and DiChaViT [19]. As shown in Table 3, our modules consistently improve performance across both architectures. On JUMP-CP, incorporating FTL and CAM yields $+1.15$pp (full) and $+2.09$pp (partial) on IC-ViT^{mem}, and $+3.26$pp (full) and $+2.68$pp (partial) on DiChaViT^{mem}. Here, IC-ViT^{mem} and DiChaViT^{mem} denote memory-token-augmented variants constructed to enable a controlled evaluation of FTL and CAM under matched architectural settings, as both modules operate on memory-token slots. Within this matched setting, these results demonstrate that FTL and CAM are architecture-agnostic and can be seamlessly integrated into diverse multi-channel ViT models.

6 Conclusion

We presented **MuCa-ViT**, a structured framework for multi-channel representation learning that addresses representation collapse and the lack of composition-aware channel interpretation in existing multi-channel ViTs. Our approach combines **Factorial Token Learning (FTL)**, which suppresses token-level redundancy to produce a reliable global representation, with **Channel Composition-Aware Modulation (CAM)**, which leverages this representation to adaptively interpret channel semantics under varying channel compositions. By conditioning feature transformations on the global context, CAM enables the same channel to be interpreted differently depending on the input configuration. Extensive experiments show that the gains are larger in higher-redundancy regimes, consistent with our analysis that inter-channel redundancy is a major structural factor that promotes CST collapse. Plug-in evaluations further demonstrate the compatibility of FTL and CAM with other multi-channel ViT architectures. Overall, our results suggest that effective multi-channel modeling benefits from both structured representation and composition-aware interpretation.

Acknowledgments and Disclosure of Funding

This research was supported by the G-LAMP Program of the National Research Foundation of Korea (NRF), funded by the Ministry of Education (RS-2025-25442252); the Bio & Medical Technology Development Program of the NRF, funded by the Korean government (MSIT) (RS-2022-NR068424); the BK21 FOUR program of the NRF; and the AI Seoul Tech Research Support Program of the Seoul Future Foundation. This work was supported by the Korea Bio Data Station (K-BDS) with computing resources including technical support.

References

[1] Mahmoud Assran, Mathilde Caron, Ishan Misra, Piotr Bojanowski, Florian Bordes, Pascal Vincent, Armand Joulin, Mike Rabbat, and Nicolas Ballas. Masked siamese networks for label-efficient learning. In *Computer Vision – ECCV 2022*, pages 456–473. Springer, 2022.

[2] Yujia Bao, Srinivasan Sivanandan, and Theofanis Karaletsos. Channel vision transformers: An image is worth $1 \times 16 \times 16$ words. In *International Conference on Learning Representations*, 2024.

[3] Sawyer Birnbaum, Volodymyr Kuleshov, S. Zayd Enam, Pang Wei Koh, and Stefano Ermon. Temporal FiLM: Capturing long-range sequence dependencies with feature-wise modulations. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[4] Nicolas Bourriez, Ihab Bendidi, Ethan Cohen, Gabriel Watkinson, Maxime Sanchez, Guillaume Bollot, and Auguste Genovesio. ChAda-ViT: Channel adaptive attention for joint representation learning of heterogeneous microscopy images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11556–11565, 2024.

[5] Marc Brockschmidt. GNN-FiLM: Graph neural networks with feature-wise linear modulation. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1144–1152. PMLR, 2020.

[6] Srinivas Niranj Chandrasekaran, Beth A Cimini, Amy Goodale, Lisa Miller, Maria Kost-Alimova, Nasim Jamali, John G Doench, Briana Fritchman, Adam Skepner, Michelle Melanson, et al. Three million images and morphological profiles of cells treated with matched chemical and genetic perturbations. *Nature Methods*, 21(6):1114–1121, 2024.

[7] Zitong Sam Chen, Chau Pham, Siqi Wang, Michael Doron, Nikita Moshkov, Bryan Plummer, and Juan C Caicedo. CHAMMI: A benchmark for channel-adaptive models in microscopy imaging. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, volume 36, pages 19700–19713, 2023.

[8] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, pages 8780–8794, 2021.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

[10] David Ha, Andrew M. Dai, and Quoc V. Le. HyperNetworks. In *International Conference on Learning Representations*, 2017.

[11] Seokhyeon Hong, Chaelin Kim, Serin Yoon, Junghyun Nam, Sihun Cha, and Junyong Noh. SALAD: Skeleton-aware latent diffusion for text-driven motion generation and editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7158–7168, 2025.

[12] Songning Lai, Jiakang Li, Guinan Guo, Xifeng Hu, Yulong Li, Yuan Tan, Zichen Song, Yutong Liu, Zhaoxia Ren, Chun Wan, Danmin Miao, and Zhi Liu. Shared and private information learning in multimodal sentiment analysis with deep modal alignment and self-supervised multi-task learning. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2024.

[13] Wenyi Lian, Patrick Micke, Joakim Lindblad, and Nataša Sladoje. Isolated channel vision transformers: From single-channel pretraining to multi-channel finetuning. In *Proceedings of the 36th British Machine Vision Conference*. BMVA, 2025.

[14] Feng Liang, Yangguang Li, and Diana Marculescu. SupMAE: Supervised masked autoencoders are efficient vision learners. *arXiv preprint arXiv:2205.14540*, 2022.

[15] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[16] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.

[17] Ethan Perez, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. FiLM: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[18] Chau Pham, Juan C Caicedo, and Bryan A Plummer. ChA-MAEViT: Unifying channel-aware masked autoencoders and multi-channel vision transformers for improved cross-channel learning. In *Advances in Neural Information Processing Systems*, volume 38, pages 13342–13370, 2025.

[19] Chau Pham and Bryan A. Plummer. Enhancing feature diversity boosts channel-adaptive vision transformers. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 89782–89805. Curran Associates, Inc., 2024.

[20] Chau Pham, Piotr Teterwak, Soren Nelson, and Bryan A. Plummer. MixtureGrowth: Growing neural networks by recombining learned parameters. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2024.

[21] Bryan A Plummer, Nikoli Dryden, Julius Frost, Torsten Hoefler, and Kate Saenko. Neural parameter allocation search. In *International Conference on Learning Representations*, 2022.

[22] Prajit Ramachandran, Barret Zoph, and Quoc V Le. Searching for activation functions. *arXiv preprint arXiv:1710.05941*, 2017.

[23] Pedro Savarese and Michael Maire. Learning implicitly recurrent CNNs through parameter sharing. In *International Conference on Learning Representations*, 2019.

[24] Peter J Thul, Lovisa Åkesson, Mikaela Wiking, Diana Mahdessian, Aikaterini Geladaki, Hammou Ait Blal, Tove Alm, Anna Asplund, Lars Björk, Lisa M Breckels, et al. A subcellular map of the human proteome. *Science*, 356(6340):eaal3321, 2017.

[25] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 32–42, 2021.

[26] Yao-Hung Hubert Tsai, Shaojie Bai, Paul Pu Liang, J. Zico Kolter, Louis-Philippe Morency, and Ruslan Salakhutdinov. Multimodal transformer for unaligned multimodal language sequences. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6558–6569. Association for Computational Linguistics, 2019.

[27] Matheus P Viana, Jianxu Chen, Theo A Knijnenburg, Ritvik Vasan, Calysta Yan, Joy E Arakaki, Matte Bailey, Ben Berry, Antoine Borensztejn, Eva M Brown, et al. Integrated intracellular organization and its variations in human iPS cells. *Nature*, 613(7943):345–354, 2023.

[28] Gregory P Way, Ted Natoli, Adeniyi Adeboye, Lev Litichevskiy, Andrew Yang, Xiaodong Lu, Juan C Caicedo, Beth A Cimini, Kyle Karhohs, David J Logan, et al. Morphology and gene expression profiling provide complementary information for mapping cell state. *Cell Systems*, 13(11):911–923.e9, 2022.

[29] Seokju Yun, Seunghye Chae, Dongheon Lee, and Youngmin Ro. SoMA: Singular value decomposed minor components adaptation for domain generalizable representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 25602–25612, 2025.

[30] Jisi Zhang, Pablo Peso Parada, Md Asif Jalal, and Karthikeyan Saravanan. Diffusion based text-to-music generation with global and local text based conditioning. In *ICASSP 2025 – 2025 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1–5. IEEE, 2025.

[31] Tengxue Zhang, Yang Shu, Xinyang Chen, Yifei Long, Chenjuan Guo, and Bin Yang. Assessing pre-trained models for transfer learning through distribution of spectral components. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 22560–22568, 2025.

[32] Xiao Xiang Zhu, Jingliang Hu, Chunping Qiu, Yilei Shi, Jian Kang, Lichao Mou, Hossein Bagheri, Matthias Häberle, Yuansheng Hua, Rong Huang, et al. So2Sat LCZ42: A benchmark data set for the classification of global local climate zones [software and data sets]. *IEEE Geoscience and Remote Sensing Magazine*, 8(3):76–89, 2020.

Appendix Overview

This appendix complements the main paper as follows.

Sec. A details the experimental setup and reproduction protocol.

Sec. B reports design analyses and ablations on FTL and CAM, covering memory-token count, asymmetric weighting, λ_{FTL} sensitivity, modulation parameters, conditioning source, and insertion depth.

Sec. C provides additional results, including task-level CHAMMI scores, robustness across channel configurations, ViT-Base scaling, computational cost, and token-budget analysis.

Sec. D formalizes the SVD-based spectral analysis. It also evaluates missing-channel reconstruction and reinjection.

Sec. E quantifies inter-channel redundancy and its correlation with performance gains.

Sec. F presents token-level evidence of representation collapse via Gram-matrix and attention analyses. It also tests the robustness of channel coverage to similarity normalization and temperature, and evaluates prediction under final `[CLS]` corruption.

Sec. G describes the CAM algorithm.

Sec. H discusses limitations and future work.

A Experimental Details

Table 4: Training and evaluation configurations used to reproduce ChA-MAEViT [18] (denoted ChA-MAEViT* in Table 1). MuCa-ViT is trained under identical configurations except for the memory token count and the additional FTL/CAM hyperparameters specified in Appendix B.3.3, B.2.2.

Configuration	**JUMP-CP**	**CHAMMI**	**So2Sat**
Hardware and batch			
GPUs (NVIDIA A6000)	2	1	1
Train batch size	32	80	128
Eval batch size	300	256	512
Num workers	1	3	3
Optimization			
Optimizer	AdamW	AdamW	AdamW
Learning rate	4×10^{-4}	4×10^{-4}	4×10^{-4}
LR warmup epochs	10	0	0
Max gradient norm	3	1	1
Training epochs	100	100	100
Architecture			
Patch size	16	16	8
Encoder depth	11	11	11
Decoder depth	1	1	1
Memory tokens (baseline)	4	4	8
Memory tokens (MuCa-ViT)	3	3	7

Datasets. For JUMP-CP, the eight channels consist of five fluorescence and three brightfield channels. For So2Sat, we follow the city-split protocol with 352K training and 24K test samples.

Memory Tokens Configuration. Following the setup described in Fig. 5, we used 3, 3, and 7 memory tokens for JUMP-CP, CHAMMI, and So2Sat, respectively. We further applied orthogonal initialization to the memory tokens to reduce token correlation during optimization. For ChA-MAEViT [18], we follow the default memory token settings that yield the best performance, using 4, 4, and 8 tokens for JUMP-CP, CHAMMI, and So2Sat, respectively. For IC-ViT [13], we use 4 memory tokens on the JUMP-CP dataset. Notably, MuCa-ViT used fewer memory tokens than ChA-MAEViT's original settings; thus, the gains reported in Table 1 are not attributed to larger token budgets (further analyzed in Appendix B.1).

CAM Insertion Depth and FTL Weighting. Across datasets, applying CAM to the final Transformer block yields the best overall performance. For JUMP-CP, however, applying CAM to the last two blocks (10–11) provides consistent additional gains; we therefore adopt this setting when reporting its results in Table 1. For architectures without memory tokens, we insert memory tokens with identical initialization and apply FTL and CAM in a plug-and-play manner, preserving the original training protocol.

Downstream Tasks and Evaluation Protocol. Our method follows the training and evaluation settings of ChA-MAEViT [18] and is implemented on top of the baseline. Across all datasets, we jointly optimize the reconstruction and classification objectives for 100 epochs; the same training length is used for the fine-tuning stage in the IC-ViT [13] plug-in experiments. Performance is measured by top-1 classification accuracy on JUMP-CP and So2Sat, and by macro-averaged F1 over the WTC-11 and HPA tasks for CHAMMI. We trained the models using the AdamW optimizer, minimizing cross-entropy loss for JUMP-CP and So2Sat, and proxy loss for CHAMMI, following the protocol of ChA-MAEViT [18].

Hardware Setup and Reproduction Protocol. Experiments were conducted on a single NVIDIA A6000 GPU (CHAMMI, So2Sat) and dual A6000 GPUs (JUMP-CP). To ensure consistency, all Δ values reported in this paper are computed against our reproduced ChA-MAEViT [18] (marked $*$ in Table 1) under the same training protocol. For completeness, Table 1 also includes the published results, and MuCa-ViT outperforms ChA-MAEViT under both references on JUMP-CP and CHAMMI.

B Design Analysis

B.1 Effect of the Number of Memory Tokens

As shown in Fig. 5, we analyze the effect of the number of memory tokens on the JUMP-CP, So2Sat, and CHAMMI datasets. In ChA-MAEViT [18], the default numbers of memory tokens are set to 4, 8, and 4 for JUMP-CP, So2Sat, and CHAMMI, respectively. In contrast, our model achieves the best performance with 3, 7, and 3 memory tokens on the corresponding datasets. This observation suggests that our method requires fewer memory tokens to achieve optimal performance. We attribute this to the reduced redundancy among CSTs enabled by FTL, which allows information to be more effectively separated and stored within each memory token. Since the main comparison in Table 1 uses each method's best-performing memory-token count, the comparison is favorable to the baseline: MuCa-ViT outperforms ChA-MAEViT despite using fewer memory tokens, so the reported gains cannot be attributed to a larger token budget.

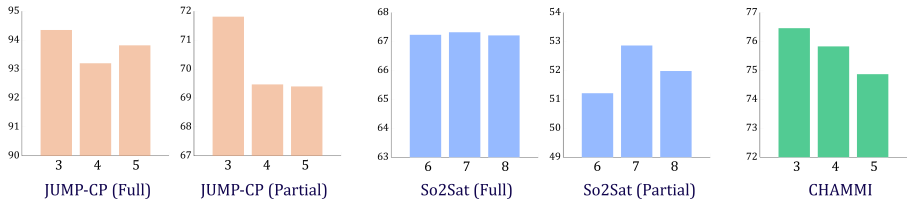


Figure 5: Effect of the number of memory tokens on JUMP-CP, So2Sat, and CHAMMI. Our model consistently achieves optimal performance with fewer memory tokens compared to the default settings of ChA-MAEViT.

B.2 Ablation Study on FTL

B.2.1 Ablation on Asymmetric Weight α

Analyzing the structure of inter-channel collapse reveals that [CLS]–memory decorrelation and memory–memory decorrelation serve distinct roles: [CLS]–memory decorrelation corrects channel bias in the global representation, while memory–memory decorrelation eliminates local redundancy between individual channel tokens. Treating W (Eq. (3)) as a uniform matrix with equal weights ignores this asymmetry. As shown in Table 5, the asymmetric weighting yields +1.32 pp / +0.45 pp

over the equal-weight variant, validating that this design is not a cosmetic choice but a necessary consequence of the structure of inter-channel collapse.

Table 5: Ablation on asymmetric weighting in FTL. JUMP-CP accuracy.

Variant	Full	Partial
Equal-weight ablation	94.09	71.81
MuCa-ViT (asymmetric)	**95.41**	**72.26**

B.2.2 Sensitivity to Regularization Weight λ_{FTL}

We analyze the sensitivity of FTL to the decorrelation weight λ_{FTL} in Fig. 6. As shown in Fig. 6, performance peaks at a moderate regularization strength and gradually decreases as λ_{FTL} increases in both full- and partial-channel settings. This trend suggests that while structured decorrelation is beneficial for alleviating token redundancy, overly strong orthogonality constraints suppress useful task-relevant correlations. Importantly, the performance variation remains smooth and stable across a reasonable range of λ_{FTL}, indicating that FTL does not rely on fragile hyperparameter tuning and achieves a balanced trade-off between semantic factorization and discriminative capacity.

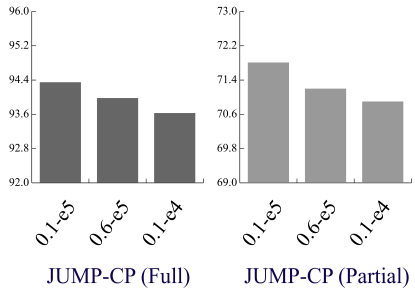


Figure 6: Sensitivity analysis of the FTL decorrelation weight λ_{FTL} on JUMP-CP under full and partial channel settings. Performance peaks at a moderate regularization strength and remains stable across a broad range of λ_{FTL}.

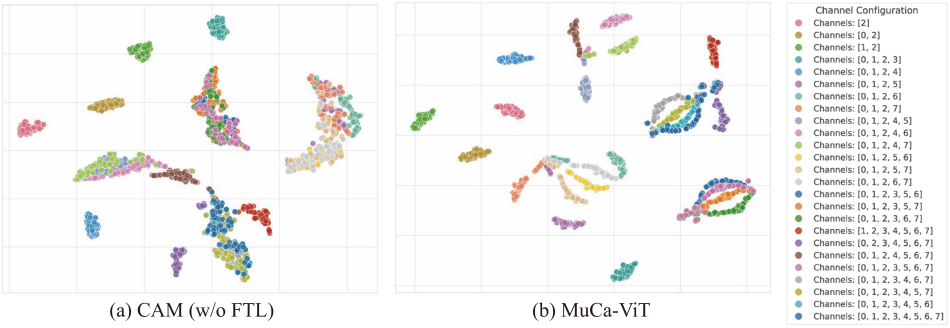
B.3 Ablation Study on CAM

B.3.1 Analysis of CAM Modulation Parameters

To verify that CAM performs composition-dependent modulation, we visualize the learned affine parameters (γ, β) using t-SNE in Fig. 7. Each point corresponds to the (γ, β) generated by CAM when the *same underlying samples* are fed with a *different channel subset*; we vary the subset across all configurations shown in the legend and collect the resulting modulation parameters. As shown in Fig. 7, different channel subsets form clearly separable clusters, organized by channel configuration rather than by sample identity, indicating that CAM produces configuration-aware modulation signals rather than static adjustments. Configurations with overlapping channel sets exhibit related but distinct embeddings, suggesting that modulation parameters smoothly adapt to the global semantic context encoded in the `[CLS]` token. Comparing (a) CAM without FTL and (b) MuCa-ViT (CAM with FTL), the clusters become substantially sharper under FTL, confirming that the FTL-refined `[CLS]` provides a cleaner composition signal for modulation. These results confirm that CAM functions as an intrinsic self-conditioning mechanism, where the global semantic anchor dynamically governs feature recalibration under heterogeneous channel compositions.

B.3.2 Ablation on Conditioning Source in CAM

To isolate the contribution of FTL-refined self-conditioning, we compare three conditioning sources in Table 6, organized along two axes. The first axis distinguishes *external* conditioning, which uses signals provided alongside the input as explicit priors, from *intrinsic* conditioning, which uses representations the model itself produces from the input, specifically the `[CLS]` token.



(a) CAM (w/o FTL) (b) MuCa-ViT

Figure 7: t-SNE visualization of the learned CAM modulation parameters (γ, β) across different channel configurations. Each point is obtained by feeding the same underlying samples with a different channel subset. Distinct clusters emerge for different channel subsets, organized by configuration rather than sample identity, indicating that CAM generates configuration-aware modulation signals rather than static embeddings.

External vs. intrinsic conditioning. External conditioning approaches [17, 8, 16, 3, 30] typically rely on auxiliary inputs such as language embeddings, class labels, or timestep encodings as explicit priors. In our setup, we directly inject the channel composition as a multi-hot vector indicating the active channels, providing the model with explicit information about which channels are present. Intrinsic conditioning instead uses the model's own `[CLS]` token, which not only reflects the channel composition but also captures sample-specific semantics, allowing CAM to adapt its modulation per sample rather than relying on the channel composition alone. As shown in Table 6, intrinsic conditioning with the FTL-refined `[CLS]` outperforms external conditioning by $+2.84$ pp, indicating that the global semantic representation built up within the model carries richer information about the channel configuration than an external categorical prior.

Effect of FTL refinement within intrinsic conditioning. Within intrinsic conditioning, we further compare two variants. The first uses the raw `[CLS]` token, which remains biased toward dominant channels, and the second uses the FTL-refined `[CLS]` token. The FTL-refined variant outperforms the raw `[CLS]` variant by $+2.23$ pp. This isolates the contribution of collapse correction within intrinsic conditioning and indicates that the effectiveness of CAM depends on the quality of the internal signal, not merely on whether the signal is internal.

Table 6: Ablation on the conditioning source in CAM. JUMP-CP accuracy.

Conditioning Source	Full	Partial
External conditioning	92.57	70.47
Intrinsic conditioning (w/o FTL refinement)	93.18	71.12
FTL-refined intrinsic conditioning (MuCa-ViT)	**95.41**	**72.26**

B.3.3 Effect of CAM Insertion Depth in the Encoder

Fig. 8 analyzes the impact of inserting CAM at different transformer blocks. As shown in Fig. 8, applying CAM to the final encoder block provides a strong default across datasets, while extending CAM to the last two blocks yields additional gains on JUMP-CP. Placing CAM too early limits its access to sufficiently abstract global semantics. These results support our design choice of inserting CAM near the final transformer stage, where the global `[CLS]` token encodes stable semantic context while still allowing effective feature recalibration.

We further observe that the optimal CAM *placement depth* depends on inter-channel redundancy. Datasets with higher redundancy benefit from applying CAM to more blocks, while datasets with lower redundancy reach their best performance with CAM at the final block alone. Specifically, JUMP-CP ($\rho = 0.77$, Table 14) benefits from extending CAM to the last two blocks, whereas CHAMMI ($\rho = 0.46$) and So2Sat ($\rho = 0.19$) achieve their best performance with CAM applied only to the final block.

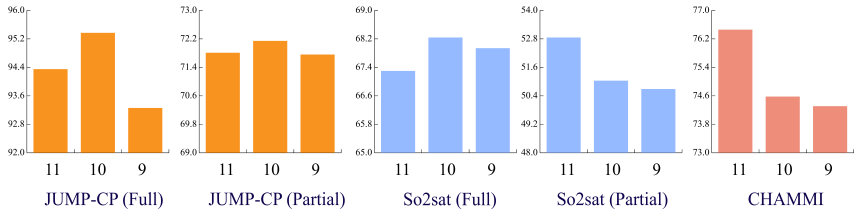


Figure 8: Effect of CAM insertion depth across different transformer blocks. Each x-axis label denotes the starting block from which CAM is applied to all subsequent blocks (e.g., Block 9 indicates CAM applied to blocks 9–11).

B.3.4 Ablation on CAM Placement across Network Depth

Following the observation in CaiT [25] that the [CLS] token reaches its full semantic maturity in deeper blocks, we justify placing CAM in the last two blocks. Applying CAM in early layers can interfere with low-level feature extraction before the [CLS] token has accumulated sufficient semantic information to serve as a reliable conditioning signal. In contrast, placing CAM in deeper layers allows the model to leverage matured high-level abstractions for precise channel configuration. As shown in Table 7, bottom placement (last two blocks) yields a +3.23 pp / +1.60 pp improvement over top placement.

Table 7: Ablation on CAM placement across network depth. JUMP-CP accuracy.

Variant	Full	Partial
ChA-MAEViT (Baseline)	90.35	68.10
+ FTL (Ours w/o CAM)	91.58	68.23
+ FTL + CAM (Top, first 2 blocks)	92.18	70.66
+ FTL + CAM (Bottom, last 2 blocks)	**95.41**	**72.26**

C Additional Results

C.1 Detailed CHAMMI Results

In Table 8, we provide task-level F1 scores on the CHAMMI benchmark to complement the aggregated results reported in the main text. CHAMMI consists of multiple sub-datasets with varying channel compositions and domain shifts. Compared with our reproduced ChA-MAEViT*, MuCa-ViT achieves higher scores on 6 of the 9 tasks, including all WTC and HPA tasks, and obtains the best result among all models on four of them.

Table 8: F1 scores of multi-channel ViT models on the CHAMMI benchmark. ChA-MAEViT* denotes our reproduction under the same training environment as MuCa-ViT; both are single-run results. **Bold** indicates the best result within each group, and underline indicates the best result overall.

	WTC [27]		HPA [24]			CP [28]			
Model	Task1	Task2	Task1	Task2	Task3	Task1	Task2	Task3	Task4
HyperNetViT [10]	58.83	45.78	88.78	82.70	52.52	82.13	53.74	23.16	7.42
DepthwiseViT [7]	69.81	52.19	91.65	**88.04**	54.78	81.24	54.08	**23.21**	5.87
TemplateMixingViT [20, 21, 23]	61.66	51.52	85.01	79.91	49.69	77.45	48.83	22.56	5.60
ChAda-ViT [4]	77.58	67.18	87.49	75.94	45.41	83.92	45.58	21.94	6.28
ChannelViT [2]	78.36	67.58	83.93	76.73	47.97	77.70	55.16	21.89	6.38
DiChaViT [19]	80.87	75.18	88.08	79.26	49.45	84.08	53.03	20.95	5.60
ChA-MAEViT [18]	**84.52**	**77.15**	**94.14**	87.47	**56.75**	**90.89**	**56.68**	10.25	**7.50**
ChA-MAEViT* [18]	83.42	76.00	96.12	86.94	57.20	**96.89**	**48.19**	10.12	**8.90**
MuCa-ViT (Ours)	**85.98**	**80.29**	**96.27**	**87.95**	**57.34**	96.59	45.30	**11.15**	8.51

C.2 Evaluation under Various Channel Configurations

To evaluate robustness under missing-channel scenarios, we evaluated the model on all channel configurations derived from the full set of eight channels on JUMP-CP. We vary the number of available channels and consider all corresponding channel combinations, reporting the mean and standard deviation across configurations in Table 9. As shown in Table 9, MuCa-ViT consistently outperforms the baseline across nearly all channel budgets, with the largest gains observed in intermediate regimes (5–7 channels) where channel composition variability is most challenging.

Table 9: Performance comparison across different numbers and combinations of channels at inference. Mean ± std over all channel combinations at each budget.

	Number of channels at inference							
Method	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**
DiChaViT [19]	69.19	61.91 ±9.3	54.49 ±12.4	46.35 ±13.4	38.00 ±12.4	30.09 ±9.3	23.97 ±4.9	20.90 ±1.6
ChA-MAEViT [18]	90.73	83.36 ±8.3	74.55 ±11.7	63.46 ±13.8	50.85 ±13.9	38.13 ±11.2	27.62 ±6.4	**21.59** ±2.1
MuCa-ViT	**95.41**	**89.46** ±6.28	**80.06** ±11.32	**69.75** ±12.84	**55.84** ±13.80	**41.18** ±11.65	**28.79** ±6.84	21.54 ±2.25

C.3 Scaling to ViT-Base

We further evaluate our method at the ViT-Base scale, with results summarized in Table 10. ChA-MAEViT* results in this section are reproduced by us under the same training environment as MuCa-ViT. MuCa-ViT consistently improves over ChA-MAEViT* at the ViT-Base scale on both JUMP-CP and CHAMMI, indicating that the gains from FTL and CAM are not specific to a particular backbone capacity.

Table 10: Performance at ViT-Base scale.

Model	JUMP-CP (Full)	JUMP-CP (Partial)	CHAMMI
ChA-MAEViT*	92.22	70.24	76.20
MuCa-ViT (Ours)	**94.28**	**74.31**	**76.72**

C.4 Computational Complexity and Inference Latency

We evaluate the computational overhead and inference latency at batch size 32 on ViT-Small, with results reported in Table 11. The CAM modules introduce an additional latency of approximately 0.15s per batch, while the overall FLOPs remain effectively unchanged. We provide a detailed token-budget analysis in Appendix C.5, which shows that the gains are not attributable to additional capacity.

Table 11: Computational complexity and inference latency at ViT-Small (batch size 32).

Model	Inference Latency	FLOPs
ChA-MAEViT	1.084s	249.48
MuCa-ViT (Ours)	1.235s	**249.33**

C.5 Computational Cost and Token Budget Analysis

Information density, not capacity, drives the gains. ChA-MAEViT* requires 4 memory tokens to reach its best configuration, and reducing the budget from 4 to 3 produces a consistent drop (-0.11/-0.75 pp on JUMP-CP Full/Partial, and -1.38 pp on CHAMMI), confirming that its representation quality depends on additional token slots. MuCa-ViT exhibits the opposite behavior. It reaches its best performance with only 3 memory tokens and still outperforms the best-configured ChA-MAEViT* by +5.06/+4.16 pp on JUMP-CP and +2.42 pp on CHAMMI. Since MuCa-ViT improves under a reduced token budget where the baseline degrades, the observed gains cannot be

Table 12: Performance comparison under matched and best-performing memory-token budgets. MuCa-ViT outperforms ChA-MAEViT* at the same token budget and exceeds its best-performing configuration with fewer memory tokens. ChA-MAEViT* denotes our reproduced baseline. Results are from single runs.

Model	mem	JUMP-CP		CHAMMI
		Full	Partial	
ChA-MAEViT*	3	90.24	67.35	72.66
ChA-MAEViT* (best)	4	90.35	68.10	74.04
MuCa-ViT (Ours)	3	**95.41**	**72.26**	**76.46**

attributed to additional capacity. They reflect a higher information density per token, induced by FTL's suppression of redundancy among Context Semantic Tokens (Sec. 3). Once each memory token is encouraged to encode information complementary to `[CLS]` rather than remaining redundant with it, fewer tokens suffice to cover the channel set.

Computational cost. The overall FLOPs remain effectively unchanged from the baseline (249.33 vs. 249.48 G, a 0.06% difference), even though MuCa-ViT introduces the CAM modules. This is because the reduction by one memory token slightly shortens the self-attention sequence and offsets the additional cost of CAM. Latency increases by approximately 0.15s per batch at batch size 32, reflecting the sequential overhead of inserting CAM at the late transformer blocks.

D Representation Analysis

This section provides the formal details of the spectral analysis introduced in Sec. 5.3, including the SVD-based decomposition formulation and the quantitative diversity metric used in Fig. 4.

Multi-channel visual representations contain two fundamentally different sources of variation: globally shared structural information common across channels, and heterogeneous channel-specific semantics. To examine how these factors are encoded and separated in the learned representation space, we conduct a spectral analysis based on singular value decomposition (SVD), which has been widely adopted to probe semantic structure in representation learning [31, 29].

Spectral decomposition. Let the final patch output at the Nth encoder block be $\mathbf{Z} \in \mathbb{R}^{(C\times P)\times D}$, where patch tokens from all channels are concatenated along the token dimension. We compute the SVD as

$$\mathbf{Z} = U\Sigma V^\top, \tag{6}$$

where the right singular vectors $V \in \mathbb{R}^{D\times D}$ span the principal semantic directions of the representation space. Following prior spectral grouping strategies, we partition this basis into dominant and residual components,

$$V_{\text{top}} = V[:, 1\!:\!d_{\text{top}}], \quad V_{\text{mid}} = V[:, d_{\text{mid1}}\!:\!d_{\text{mid2}}], \tag{7}$$

where d_{top} denotes the number of dominant singular directions capturing globally shared structure, and $[d_{\text{mid1}}, d_{\text{mid2}}]$ specifies the residual spectral band used to analyze channel-specific semantic variation. The reconstructed spectral component for a group $g \in \{\text{top}, \text{mid}\}$ is given by

$$\widetilde{\mathbf{Z}}_g = (\mathbf{Z} - \mu)\, V_g V_g^\top + \mu, \tag{8}$$

where μ is the token-wise mean and $V_g \in \mathbb{R}^{D\times d_g}$ contains the selected right singular vectors. We use these reconstructions to probe shared and channel-specific structure both qualitatively (Sec. D.1) and quantitatively (Sec. D.2).

D.1 Visual Separation of Shared and Channel-Specific Semantics

We first visualize the reconstructed components $\widetilde{\mathbf{Z}}_{\text{top}}$ and $\widetilde{\mathbf{Z}}_{\text{mid}}$ via t-SNE in Fig. 4. As shown in Fig. 4 (a), the top spectral component exhibits substantial overlap across channels for both the baseline [18] and our method, indicating that it primarily encodes globally shared cross-channel

structure. In contrast, Fig. 4 (b) shows that the middle spectral component yields clear channel-wise clustering under our method, whereas the baseline exhibits substantially weaker separation. This contrast indicates that while shared structure is consistently captured by the dominant spectrum, our method isolates heterogeneous channel-specific semantics within the residual spectral subspace.

D.2 Quantitative Spectral Diversity

To quantify this separation across spectral groups, we measure the inter-channel diversity of each reconstructed component. Letting $\widetilde{\mathbf{Z}}_{g,c} \in \mathbb{R}^{P\times D}$ denote the tokens belonging to channel c within $\widetilde{\mathbf{Z}}_g$, we compute the mean patch token

$$m_{g,c} = \frac{1}{P}\sum \widetilde{\mathbf{Z}}_{g,c}, \tag{9}$$

and define the inter-channel similarity and diversity as

$$\mathrm{Sim}(g) = \frac{1}{C(C-1)}\sum_{c\neq c'} \cos(m_{g,c}, m_{g,c'}), \quad \mathrm{Div}(g) = 1 - \mathrm{Sim}(g). \tag{10}$$

As shown in Fig. 9, the diversity curve reveals a structural transition in the latent subspace. In the **early spectral bands**, the reconstructed components exhibit high similarity across channels, primarily encoding shared spatial structure. As we move toward the **middle spectral bands**, a **distinct increase** in diversity emerges for MuCa-ViT, while the baseline saturates at a substantially lower level. This confirms quantitatively what the t-SNE visualization shows the per-channel reconstructions become increasingly distinct under our method, as MuCa-ViT captures discriminative channel-specific semantics within these residual bands.

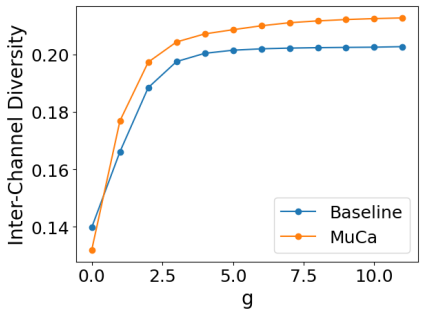


Figure 9: Inter-channel diversity $\mathrm{Div}(g)$ of reconstructed spectral components across spectral groups g on JUMP-CP. MuCa-ViT (orange) maintains consistently higher diversity than the baseline (blue), especially in the middle spectral bands.

Connection to multimodal factorization. This behavior aligns with the shared–private factorization commonly observed in multimodal representation learning [12]. In our setting, the dominant spectrum corresponds to semantics shared across channels, whereas the middle spectrum captures fine-grained, channel-specific variation. Notably, such spectral disentanglement emerges only under our method and correlates with improved downstream performance, suggesting that FTL and CAM jointly promote semantically meaningful factorization in the representation space.

D.3 Missing-Channel Reconstruction and Reinjection

The spectral analysis above examines how shared and channel-specific information is organized in the representation space. We further evaluate whether the learned memory representations preserve complementary information that can recover missing channels and support downstream prediction. Following the reconstruction protocol of ChA-MAEViT [18], we provide only the three brightfield channels of JUMP-CP and reconstruct the five missing fluorescence channels using the learned memory tokens. At test time, the reconstructed fluorescence channels are concatenated with the observed brightfield channels and reinjected into the frozen classifier.

As shown in Table 13, MuCa-ViT obtains a larger reinjection gain (+13.85 pp) than ChA-MAEViT (+11.35 pp), while reducing both spatial reconstruction error and Fourier error. This improvement

Table 13: Missing-channel reconstruction and reinjection on JUMP-CP. Five fluorescence channels are reconstructed from three observed brightfield channels and reinjected into the frozen classifier. Reinjection gain is measured relative to brightfield-only accuracy.

Method	Brightfield-only Acc. (%)	After reinjection Acc. (%)	Gain (pp)	MSE ↓	Fourier error ↓
ChA-MAEViT	22.77	34.12	+11.35	0.028	0.343
MuCa-ViT (Ours)	**23.54**	**37.39**	**+13.85**	**0.027**	**0.321**

indicates that the memory representations preserve complementary information useful for recovering the missing fluorescence channels. Together with the spectral organization in Sec. D.1, these results support the functional value of preserving channel-specific information alongside shared structural cues.

E Channel Redundancy Analysis

This section provides the concrete measurement procedure and per-dataset values for the inter-channel redundancy ρ used in Sec. 3.

A central observation of this work is that, when the input channels carry substantial overlapping information, training a multi-channel ViT without an explicit decorrelation mechanism is associated with `[CLS]` token collapse, as quantified by our entropy-based proxies in Sec. 3. To support this observation, we need a way to quantify how much redundancy each dataset actually contains. To this end, we measure channel-channel cosine similarity directly on the input images, using 16×16 patches.

For each dataset, we take normalized images $x \in \mathbb{R}^{C\times H\times W}$ and partition each image into N non-overlapping 16×16 patches. For each image, we compute the per-image channel-channel similarity matrix $S \in \mathbb{R}^{C\times C}$ as

$$S_{ij} = \frac{1}{N}\sum_{p=1}^{N} \cos\big(x_i^p, x_j^p\big), \qquad i, j \in \{1, \dots, C\}, \tag{11}$$

where $x_i^p \in \mathbb{R}^{256}$ is the flattened 16×16 patch of channel i at location p, and $\cos(u, v) = \langle u, v\rangle/(\|u\|\,\|v\|)$. Intuitively, at each patch location we compare the channel-i and channel-j patches, and then average over all N patch locations. We then average S over $2{,}048$ randomly sampled images per dataset to obtain $\bar{S} \in \mathbb{R}^{C\times C}$, and summarize $\bar{S}$ by its off-diagonal mean

$$\text{Redundancy} = \frac{1}{C(C-1)}\sum_{i\neq j} \bar{S}_{ij}, \tag{12}$$

which captures the average similarity between distinct channels.

Per-dataset cosine similarity matrices. Figure 10 shows the resulting matrices for the four evaluation groups, with CHAMMI's WTC and HPA chunks shown separately and JUMP-CP and So2Sat shown at the dataset level. Darker blue indicates higher cosine similarity between two channels. The structural differences across datasets are striking. JUMP-CP appears almost uniformly dark, indicating that its eight channels share substantial information. CHAMMI/WTC also exhibits strong redundancy, CHAMMI/HPA is mixed, and So2Sat is dominated by near-zero off-diagonal entries, with only a small number of strongly similar pairs (e.g., within the dual-polarization Sentinel-1 channels).

Quantitative summary. Table 14 reports the off-diagonal mean similarity together with the number of channels and the absolute performance gain of MuCa-ViT over our reproduced ChA-MAEViT* baseline on the Full evaluation setting from Table 1. Across the three benchmarks, the average cosine similarity ranges from 0.19 (So2Sat) to 0.77 (JUMP-CP). For CHAMMI, we report the average over its WTC and HPA chunks and visualized separately in Fig. 10.

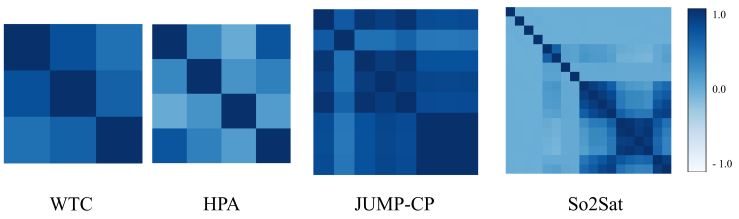


Figure 10: Per-image channel-channel cosine similarity averaged over 2,048 samples for each evaluation group. Darker blue indicates higher similarity. JUMP-CP and CHAMMI/WTC exhibit strong redundancy across channels, whereas So2Sat is largely uncorrelated except for a few pairs

Table 14: Channel redundancy and corresponding performance gain. **Redundancy** is the off-diagonal mean of the channel-channel cosine similarity matrix (Fig. 10). **ΔFull Acc.** is computed against our reproduced ChA-MAEViT* under the same training environment (cf. Table 1).

Dataset / chunk	C	**Redundancy**	**ChA-MAEViT***	**ΔFull Acc.**
So2Sat [32]	18	0.19	66.44	+1.53
CHAMMI (avg) [7]	3-4	0.46	73.95	+2.24
JUMP-CP [6]	8	0.77	90.52	+4.62

Redundancy aligns with performance gain. Table 14 reports the relationship between channel redundancy and the improvement ΔFull Acc. across the three benchmarks. Datasets with higher channel redundancy show substantially larger gains. JUMP-CP, with the highest redundancy (0.77), benefits the most (+4.62). CHAMMI lies in the middle (+2.24 at 0.46), and So2Sat, whose 18 channels carry largely orthogonal information, shows a positive improvement (+1.53) at the low-redundancy end of the range. The three benchmarks span a wide redundancy range from 0.19 to 0.77, and the gain ordering follows the redundancy ordering.

F Information Bottleneck in Context Semantic Tokens

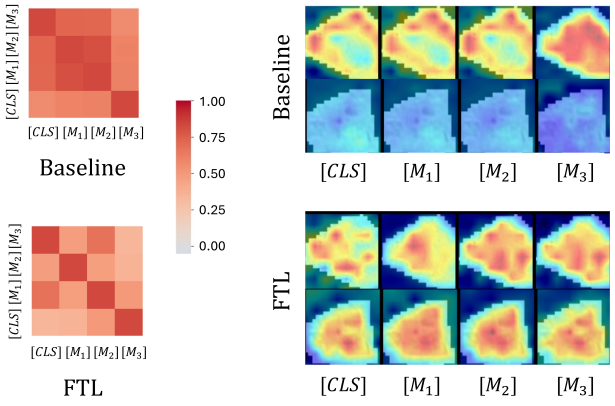
This section provides supplementary evidence supporting the CST collapse phenomenon analyzed in Sec. 3.

Multi-channel images substantially increase the length of the patch sequence as the number of channels grows, forcing increasingly rich semantic information to be summarized by a limited set of context tokens. In ViT-based architectures [9], the `[CLS]` token serves as a global semantic abstraction for downstream tasks. However, when shared spatial semantics and channel-specific semantics are entangled within a long token sequence, a single `[CLS]` token inevitably becomes an information bottleneck and is insufficient to summarize heterogeneous multi-channel information.

Prior work introduces memory tokens to improve robustness under missing-channel conditions [18]. Here, we reinterpret memory tokens as semantic storage slots intended to alleviate the token bottleneck by distributing multi-channel information across multiple context tokens. Nevertheless, in the absence of explicit structural constraints, these memory tokens become highly redundant with the `[CLS]` token, as shown in the baseline of Fig. 11 (a), resulting in low information density and failing to relieve the information bottleneck.

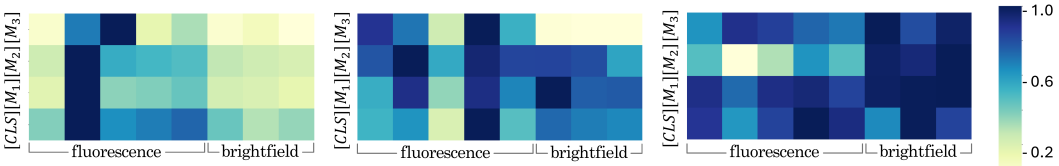
We analyze this redundancy from two complementary perspectives. First, channel-wise attention heatmap analysis reveals that Context Semantic Tokens consistently attend to the same dominant spatial regions rather than capturing distinct semantic factors across channels in the baseline of Fig. 11 (b), indicating a lack of functional specialization and semantic role separation. Second, from an information storage perspective, we measure the dot-product similarity between CSTs and channel-wise aggregated patch representations, and observe that both the `[CLS]` token and memory tokens collapse toward a small set of dominant channels in the baseline (Fig. 12). Only a subset of channel tokens contributes to semantic understanding, exhibiting severe semantic collapse.

Taken together, naive expansion of CSTs fails to distribute multi-channel semantics and instead biases the representation toward dominant channels. These observations suggest that effective multi-



(a) (b)

Figure 11: Comparison of Context Semantic Token (CST) behaviors between baseline ChA-MAEViT [18] and our method. **(a)** Gram-matrix-based redundancy analysis among CSTs. **(b)** Attention heatmaps showing spatial focus for each token.



(a) Baseline (b) FTL (c) FTL + CAM (Ours)

Figure 12: Dot-product similarity between channel-wise aggregated patch representations (x-axis) and the Context Semantic Tokens, including the [CLS] token and memory tokens M_i $(i = 1, 2, 3)$. In the baseline, both [CLS] and memory tokens exhibit strong alignment toward dominant channels, indicating severe semantic collapse and redundancy, whereas the proposed method yields well-disentangled similarity distributions.

channel representation learning requires explicit structural constraints that suppress redundancy and promote semantic factorization, with redundancy suppressed across CSTs so that memory tokens are encouraged to encode information complementary to [CLS].

F.1 Robustness of Channel Coverage to Similarity Normalization and Temperature

The coverage measures in Eqs. (1)–(2) use dot-product similarities, which depend on both directional alignment and representation norms. To examine whether the observed coverage advantage persists after removing norm effects, we recompute $H_{\texttt{CLS}}$ and H_{CST} using cosine-normalized similarities at multiple softmax temperatures. Specifically, we replace the patch-level distribution in Eq. (1) with

$$\alpha_{c,p}^{\cos,\tau}(\mathbf{q}) = \frac{\exp(\cos(\mathbf{q}, \mathbf{Z}_{c,p})/\tau)}{\sum_{c',p'} \exp(\cos(\mathbf{q}, \mathbf{Z}_{c',p'})/\tau)}, \qquad \tau \in \{0.01, 0.02, 0.04, 0.06\}, \tag{13}$$

and retain the channel aggregation and entropy definitions in Eqs. (1)–(2).

Table 15 shows that MuCa-ViT achieves higher $H_{\texttt{CLS}}$ and H_{CST} on both datasets at every tested temperature. Although the absolute entropy increases as the softmax distribution becomes less concentrated, the ordering between the two models remains unchanged. The cosine-based analysis therefore complements the dot-product diagnostic in Sec. 3, showing that the broader channel coverage is not explained solely by representation magnitude.

F.2 Prediction under Final [CLS] Corruption

The token-level analyses above examine how information is distributed across CSTs. We next test whether the intact memory-token pathway supports prediction when the final global representation

Table 15: Cosine-based channel coverage across softmax temperatures. Coverage is reported as a percentage of maximum entropy, and Δ denotes the improvement of MuCa-ViT over ChA-MAEViT in percentage points. MuCa-ViT maintains higher coverage at every tested temperature.

	$H_{\texttt{CLS}}\%$			$H_{\text{CST}}\%$		
τ	ChA-MAEViT	MuCa-ViT	Δ	ChA-MAEViT	MuCa-ViT	Δ
JUMP-CP						
0.01	10.23	**42.31**	+32.08	39.74	**76.65**	+36.91
0.02	28.79	**63.96**	+35.17	57.07	**89.10**	+32.03
0.04	60.66	**88.74**	+28.08	78.64	**98.54**	+19.90
0.06	81.93	**96.61**	+14.68	90.17	**99.63**	+9.46
So2Sat						
0.01	40.30	**42.67**	+2.37	70.83	**82.92**	+12.09
0.02	69.05	**71.56**	+2.51	82.26	**93.42**	+11.16
0.04	88.43	**90.58**	+2.15	91.41	**98.33**	+6.92
0.06	94.09	**95.56**	+1.47	94.97	**99.34**	+4.37

becomes unreliable. After the encoder's final LayerNorm, we replace only the `[CLS]` output with

$$\widetilde{\mathbf{z}}_{\text{CLS}} = \sqrt{1-\eta}\,\mathbf{z}_{\text{CLS}} + \sqrt{\eta}\,\boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, I), \quad \eta \in \{0.5, 0.7\}. \tag{14}$$

All final memory-token outputs are kept intact. The corrupted token is then used as the query in the prediction-time pooling module,

$$\mathbf{z}_{\text{pool}} = \text{CrossAttn}(q = \widetilde{\mathbf{z}}_{\text{CLS}}, K = [\mathbf{M}_1, \ldots, \mathbf{M}_K], V = [\mathbf{M}_1, \ldots, \mathbf{M}_K]), \tag{15}$$

and the pooled representation is passed to the classifier. We evaluate both Full and Partial settings on JUMP-CP.

Table 16: Prediction under final `[CLS]` corruption on JUMP-CP. Accuracy is reported in percent; the gap and parenthesized changes from clean accuracy are reported in percentage points.

Condition	**Setting**	**ChA-MAEViT**	**MuCa-ViT**	**Gap**
Clean	Full	90.35	**95.41**	+5.06
	Partial	68.10	**72.26**	+4.16
$\eta = 0.5$	Full	83.24 (-7.11)	**89.69** (-5.72)	+6.45
	Partial	58.67 (-9.43)	**67.78** (-4.48)	+9.11
$\eta = 0.7$	Full	71.64 (-18.71)	**80.14** (-15.27)	+8.50
	Partial	48.68 (-19.42)	**58.06** (-14.20)	+9.38

As shown in Table 16, MuCa-ViT exhibits a smaller accuracy drop at both corruption levels. The difference is particularly pronounced under Partial evaluation, where the performance gap increases from 4.16 pp in the clean setting to 9.11 pp at $\eta = 0.5$ and 9.38 pp at $\eta = 0.7$. These results provide functional evidence that the memory-token pathway preserves information complementary to the final `[CLS]` representation, allowing prediction to remain more reliable when the global token is perturbed.

G Algorithmic Details of CAM

We implement CAM as a lightweight feature-wise modulation block inserted after the self-attention layer in selected transformer blocks; the full procedure is summarized in Algorithm 1. Given the semantic context tokens x, we extract the global `[CLS]` token to generate modulation parameters (γ, β), which are then applied to patch tokens through affine transformation.

CAM first projects the `[CLS]` token through a lightweight network to generate feature-wise modulation parameters:

$$(\gamma, \beta) = f_{\text{CAM}}(\texttt{[CLS]}), \quad \gamma, \beta \in \mathbb{R}^D. \tag{16}$$

f_{CAM} consists of a linear projection to dimension D, followed by a SiLU activation [22], and a final linear layer outputting scaling and bias terms (γ, β). These parameters are then broadcast and applied to each patch token via an affine transformation (see lines 4–5 in Algorithm 1).

Unlike external conditioning approaches [17, 8, 16] that require auxiliary modules such as text encoders or label embedding networks to provide the conditioning signal, CAM derives (γ, β) directly from the `[CLS]` token, enabling instance-adaptive channel interpretation without any additional encoder. The effectiveness of this internal signal depends on the quality of the `[CLS]` representation. FTL refines `[CLS]` into a reliable summary of cross-channel shared semantics by suppressing dominant-channel bias under high inter-channel redundancy (Sec. 3). The empirical importance of this refinement is verified in Appendix B.3.2.

As discussed in Sec. 3, the raw `[CLS]` token is biased toward dominant channels under high inter-channel redundancy. FTL is therefore introduced to refine `[CLS]` into a reliable summary of cross-channel shared semantics, which then serves as the conditioning signal for CAM. The empirical importance of this refinement is verified in Appendix B.3.2, where FTL-refined conditioning outperforms both raw `[CLS]` conditioning and external multi-hot conditioning.

Algorithm 1 Channel Composition-Aware Modulation (CAM)

Require: Token set $\mathbf{Z} \in \mathbb{R}^{(1+K+CP)\times D}$
Ensure: Modulated token set $\mathbf{Z}' \in \mathbb{R}^{(1+K+CP)\times D}$
Notation:
$\odot$: element-wise multiplication
$(\gamma, \beta) \in \mathbb{R}^D$ are broadcast over all $(C \times P)$ patch tokens along the feature dimension.
Linear: a linear layer.
C, P, D: the number of channels, the number of patches per channel, the embedding dimension
K: the number of memory tokens.

1: **function** $\mathrm{CAM}(\mathbf{Z})$
2: $\quad$ `[CLS]` $\leftarrow \mathbf{Z}[0]$
3: $\quad \mathbf{X} \leftarrow \mathbf{Z}[1 + K : 1 + CP + K]$
4: $\quad (\gamma, \beta) \leftarrow \mathrm{Linear}(\mathrm{SiLU}(\mathrm{Linear}(\texttt{[CLS]})))$
5: $\quad \mathbf{X} \leftarrow (1 + \gamma) \odot \mathbf{X} + \beta$
6: $\quad$ **return** $\mathbf{Z}' \leftarrow \mathrm{Concat}(\texttt{[CLS]}, \mathbf{Z}[1\!:\!1 + K], \mathbf{X})$
7: **end function**
8: $\mathbf{Y} \leftarrow \mathrm{Attn}(\mathrm{Norm}_1(\mathbf{Z}), \mathrm{mask})$
9: $\mathbf{Z} \leftarrow \mathbf{Z} + \mathrm{DropPath}(\mathrm{LS}_1(\mathbf{Y}))$
10: $\mathbf{Z} \leftarrow \mathrm{CAM}(\mathbf{Z})$
11: $\mathbf{Z} \leftarrow \mathrm{Norm}_2(\mathbf{Z})$
12: $\mathbf{Z} \leftarrow \mathbf{Z} + \mathrm{DropPath}(\mathrm{LS}_2(\mathrm{Linear}(\mathbf{Z})))$

H Limitations and Future Work

Our framework opens several directions for extension.

First, the underlying mechanism of token-level redundancy under high inter-channel redundancy and composition-dependent channel interpretation is not specific to fluorescence microscopy or remote sensing. Domains such as hyperspectral imaging, multi-parametric MRI, and multiplexed tissue imaging exhibit similar structural properties and represent natural targets for MuCa-ViT-style constraints, particularly as channel counts grow beyond what a single `[CLS]` token can summarize.

Second, the FTL hyperparameters are shared across JUMP-CP, CHAMMI, and So2Sat. A common final-block CAM configuration provides consistent gains across datasets, while extending CAM to the last two blocks is an additional refinement for JUMP-CP. Making the CAM insertion depth learnable, for example through block-wise gating, would enable the model to adapt its modulation structure without manually selecting the insertion depth.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction state three contributions: (i) a diagnostic analysis of context token collapse in multi-channel ViTs, (ii) the principled complementary design of FTL and CAM addressing collapse at two coupled levels, and (iii) consistent improvements that scale with inter-channel redundancy. These claims are supported by the entropy-based collapse analysis in Section 3, the method design in Section 4, and the experimental results in Section 5.1 (Table 1).

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: A dedicated Limitations and Future Work section is provided in Appendix H, which discusses extension to other multi-channel domains and learnable CAM insertion depth. It also clarifies that the FTL hyperparameters are shared across datasets.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [N/A]

   Justification: The paper does not include formal theoretical results; contributions are supported by diagnostic analyses in Sections 3 and 5.3.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The method is described in Section 4 and Appendix G (CAM pseudocode). Full training configurations across all datasets are provided in Appendix A (Table 4), and all datasets used are publicly available.

   Guidelines:

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

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [No]

   Justification:

   Justification: Code will be available at `https://github.com/kimdanni/MuCa-ViT`. All datasets used (JUMP-CP, CHAMMI, So2Sat) are publicly available, and Appendices A and G provide implementation details sufficient to reproduce our method on top of the official ChA-MAEViT and IC-ViT codebases.

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Core experimental settings are described in Section 5.1, with full details provided in Appendix A (Table 4). Hyperparameter selection is supported by sensitivity analyses for λ_{FTL} (Appendix B.2.2), CAM insertion depth (Appendix B.3.3), and memory token count (Appendix B.1).

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

Justification: All main results in Table 1 report mean $\pm$ standard deviation over three independent runs ($n = 3$) with different random seeds, capturing variability from random initialization and training stochasticity.

Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Hardware configurations are reported in Appendix A: experiments were conducted on a single NVIDIA A6000 GPU (48GB) for CHAMMI and So2Sat, and dual A6000 GPUs for JUMP-CP. Inference latency and FLOPs comparisons against the baseline are provided in Appendix C.4 (Table 11).

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: This work uses publicly available datasets (JUMP-CP, CHAMMI, So2Sat) for representation learning research, does not involve human subjects or personally identifiable information, and conforms with the NeurIPS Code of Ethics in all respects.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: This work proposes a foundational representation learning framework for multi-channel images, with primary applications in scientific domains such as biomedical imaging and remote sensing. The work is not tied to specific deployments and does not have a direct path to negative societal applications beyond those generally associated with representation learning research.

    Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper does not release pretrained generative models, scraped datasets, or other assets carrying a high risk of misuse. The proposed method is a backbone modification for multi-channel classification tasks.

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: All datasets used (JUMP-CP [6], CHAMMI [7], So2Sat [32]) and baseline architectures (ChA-MAEViT [18], IC-ViT [13], DiChaViT [19], ChannelViT [2]) are properly cited. We follow the original licenses and terms of use for each asset.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [N/A]

    Justification: The paper does not release new datasets or pretrained models. Code for the proposed method will be available at `https://github.com/kimdanni/MuCa-ViT` with documentation.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

    Answer: [N/A]

    Justification: The paper does not involve human subjects research, so IRB approval is not applicable.

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [N/A]

    Justification: LLMs are not used as a component of the core methodology. Any use of LLMs was limited to writing, editing, or formatting assistance, which does not require declaration per the NeurIPS LLM policy.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.