Not Every Image Teaches Vision: Visual-Necessity-Gated Continual Learning for Multimodal Large Language Models

Jiamu Xue[1] **Haidong Kang**[1]* **Wen Luo**[2] **Jubo Chen**[1] **Xuhao Guo**[3]
[1]Northeastern University, China
[2]Hebei University of Technology [3]South China University of Technology
jiamu0905@gmail.com kanghaidong@qhd.neu.edu.cn wl1984@yahoo.com
jbchen@stumail.neu.edu.cn sexuhao.guo@mail.scut.edu.cn

Abstract

Continual tuning is essential for adapting multimodal large language models to evolving tasks and domains, yet existing methods often assume that every image-present example provides valid supervision for the visual pathway. We study how to learn from all incoming multimodal instructions while protecting the visual interface from visually unnecessary supervision. We propose Visual-Necessity-Gated Continual Tuning (VNG-CT), a path-aware framework that estimates sample-level dependence on visual evidence by comparing target likelihoods under the original image and a counterfactual null image. The resulting gate routes gradients to different parameter paths: all examples update the language path, visually necessary examples update the visual path, and low-necessity examples update a lightweight calibration path that promotes invariance to irrelevant visual evidence. Across CoIN, MLLM-CL, and UCIT-style continual instruction streams, VNG-CT improves final and average performance while reducing visual forgetting. These results suggest that visual necessity offers a practical principle for stabilizing continual multimodal adaptation without discarding useful language-side supervision.

1 Introduction

Multimodal large language models (MLLMs) and vision–language models (VLMs) have become a general interface for visual question answering, image-grounded instruction following, retrieval, captioning, and embodied assistance [Zhang et al., 2024a, AlSaad et al., 2024, Li et al., 2023]. In realistic deployments, robust generalization under distribution shifts remains a challenge [Yu et al., 2026, Wei et al., 2026, Wang et al., 2025b, Zheng et al., 2026a, Huang et al., 2026], and MLLMs must be updated continuously as new tasks, domains, user instructions, and visual distributions emerge. This makes multimodal continual learning a central problem: a model should acquire new multimodal capabilities without erasing previously learned visual grounding, language understanding, or cross-modal alignment [Huo and Tang, 2025].

Recent progress has substantially advanced this goal. Early multimodal continual learning benchmarks formalize sequential vision–language adaptation across heterogeneous tasks [Srinivasan et al., 2022, Zhang et al., 2023b, Yu et al., 2026], while CLIP-based continual learning methods aim to preserve pretrained cross-modal geometry, zero-shot transfer, and open-vocabulary recognition under task streams [Zheng et al., 2023, Garg et al., 2024, Jha et al., 2024, Xu et al., 2024, Tang et al., 2024, Yu et al., 2024, Liu et al., 2025, Zhang et al., 2026e]. More recently, continual instruction tuning for generative MLLMs has revealed broader forms of forgetting, including instruction-following

*Corresponding author.

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

degradation, visual knowledge forgetting, and interference between pretrained and newly acquired capabilities [Chen et al., 2024a, 2025, Guo et al., 2025a, Zhang et al., 2025, Wang et al., 2025a, Zhao et al., 2025]. Grounding can also deteriorate even when answer accuracy is retained [Chen et al., 2026b]. Recent approaches preserve compositional structure through reversible primitive–composition alignment [Xiao et al., 2026a]. GeCoin uses null-space projection to incorporate external knowledge while preserving core semantic structure in VLMs [Wu et al., 2026b].

Despite these advances, existing methods largely inherit a common assumption: if an example contains an image, its supervised loss provides valid multimodal supervision for the visual pathway. This assumption is often fragile in instruction-tuning data. Many image-present examples are answerable from language priors, domain conventions, or the instruction alone; the image may be redundant, weakly informative, or even irrelevant to the target answer. Such examples remain useful for language adaptation and instruction following, while their gradients can unnecessarily move the visual projector or cross-modal adapters. Existing anti-forgetting objectives mainly regulate the amount of model change; the validity of each sample's supervision for a particular modality pathway remains less explored.

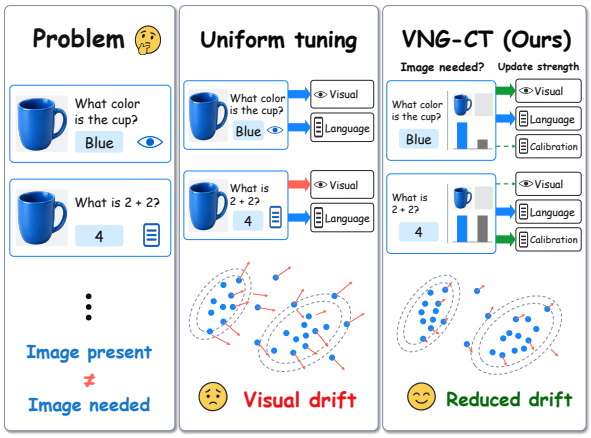

Figure 1: **Not every image-present sample should update vision.** In continual multimodal learning, many samples include an image but do not truly require visual grounding. Prior methods nevertheless apply uniform visual updates, causing conflicting gradients and cumulative visual drift. Our method estimates sample-level visual necessity and routes updates accordingly, so that vision is updated only when needed, preserving prior grounding while maintaining stable continual adaptation.

We study how to retain all incoming examples for learning while protecting the visual pathway from visually unqualified supervision. We propose **Visual-Necessity-Gated Continual Tuning** (VNG-CT), a path-aware framework that distinguishes visual evidence dependence from image presence. VNG-CT estimates each example's *visual necessity*[1] and assigns the learning signal to the parameter pathway it can validly supervise. All examples contribute to language adaptation, while visual-path updates concentrate on examples whose answers depend on visual evidence.

Our contributions are threefold:

- We identify *visual-path contamination* as an overlooked failure mode in multimodal continual tuning: image-present but visually insensitive examples can still induce non-negligible and misaligned visual-path updates, leading to accumulated visual-grounding drift.
- We propose VNG-CT, a visual-necessity-guided training paradigm that separates language adaptation, visual grounding, and calibration within a single continual update rule. VNG-CT retains low-necessity examples for language adaptation while restricting their updates to the main visual interface.
- Across CoIN, MLLM-CL, and UCIT-style continual instruction streams, VNG-CT consistently improves final and trajectory-level continual performance, reduces visual forgetting and visual-path drift on visually grounded old-task subsets, and preserves new-task adaptation by keeping low-necessity examples active for language learning.

2 Related Work

Continual learning for vision–language models. Early multimodal continual learning studied sequential adaptation in captioning, VQA, and mixed vision–language tasks, with benchmarks such

[1]We define visual necessity as the extent to which a model's prediction depends on visual evidence, quantified by the performance degradation caused by removing or perturbing the original visual input.

as CLiMB and VQACL exposing that standard replay or regularization can reduce forgetting but often fails to promote robust cross-task transfer and compositional generalization [Del Chiaro et al., 2020, Srinivasan et al., 2022, Zhang et al., 2023b, Yu et al., 2026]. With the rise of CLIP-like foundation models, the field shifted from preserving task-specific classifiers to preserving pretrained cross-modal geometry and zero-shot transfer. Representative methods include feature/parameter distillation for zero-shot retention [Zheng et al., 2023], time-continual CLIP training [Garg et al., 2024], probabilistic or analytic adaptation [Jha et al., 2024, Xu et al., 2024], residual or adapter-based interference control [Tang et al., 2024, Yu et al., 2024], and recent methods that preserve modality gap or restrict updates in protected subspaces [Huang et al., 2025, Kang et al., 2025, Jo and Kim, 2026]. CoMem supports compositional retention through concept-graph memory and subgraph-conditioned feature-space rehearsal [Zhou et al., 2026]. These methods address the preservation of pretrained multimodal capabilities during continual learning, while mostly treating each image-text pair as uniformly valid multimodal supervision. Related work on selective unlearning constrains updates to protect retained concepts in open-vocabulary detectors [Wu et al., 2026a]. Our method uses counterfactual visual necessity to determine how strongly each example updates the visual pathway.

Continual instruction tuning for MLLMs. Recent work extends continual learning from discriminative VLMs to generative MLLMs. Analyses of LLM fine-tuning dynamics also support downstream performance prediction and model selection [Zeng et al., 2025b]. CoIN reveals that sequential instruction tuning can severely degrade instruction following and general knowledge, while SEFE further separates superficial answer-style drift from essential knowledge forgetting [Chen et al., 2024a, 2025]. Recent methods such as CL-MoE, HiDe-LLaVA, BranchLoRA, SMoLoRA, and instruction-grounded visual projectors reduce task interference through expert routing, hierarchical decoupling, asymmetric LoRA branches, or instruction-conditioned projector adaptation [Huai et al., 2025, Guo et al., 2025a, Zhang et al., 2025, Wang et al., 2025a, Jin et al., 2025]. Instance-level interference-aware routing selects experts using a sample's projected update conflict [Zhang et al., 2026a]. InduceKV maintains a fixed memory budget by retrieving layerwise key–value payloads for a frozen backbone [Chen et al., 2026a]. Benchmarks and libraries such as MLLM-CL and MCITlib further standardize evaluations under domain and ability shifts [Zhao et al., 2025, Guo et al., 2025b]. Our method assigns supervision at the sample and path levels: a single example can update the language path, be suppressed on the visual path, and train a calibration path when its answer is not visually dependent.

Visual grounding, unimodal bias, and visual-path contamination. Language bias in VQA motivates adversarial regularization and question-only debiasing [Ramakrishnan et al., 2018, Cadene et al., 2019]. Uni-Med aligns medical VQA explanations with image patches and reduces non-core instruction interference through intention-guided attention [Wu et al., 2024]. TOIST grounds actions in object masks [Li et al., 2022]; Hint-AD aligns language with perception, prediction, and planning outputs [Ding et al., 2025]. LLM errors include training mismatches, inference instabilities [Zeng et al., 2026], knowledge-application failures [Chen and Liu, 2026, Liu et al., 2026], and unreliable actions under missing or conflicting evidence [Zheng et al., 2026b]. Multimodal models exhibit hallucination, spurious cross-modal correlations, and instruction-tuning-induced visual forgetting [Liu et al., 2024, Leng et al., 2024, Wu et al., 2025, Meng et al., 2026a]. To suppress harmful cross-modal messages under incomplete observations, Meng et al. [2026b] propose MRCF, which uses sample-specific modality reliability to regulate interaction and fusion. Attention-spectrum regularization preserves skill-conditioned cross-modal attention during continual adaptation [Zhao et al., 2026]. VNG-CT uses image ablation to qualify visual-path updates and curb drift, retaining visually insensitive examples for language learning.

Evaluating capability contributions. Xu et al. [2026] isolate planning-capability contributions through controlled comparisons tracking presence, activation, executed-plan changes, and operational outcomes, assessing benefits across operating regimes with equivalence and uncertainty criteria. This perspective connects to visual necessity: true/null-image comparisons test visual evidence's effect on target-answer likelihood, while routing ablations and independent grounding evaluations assess the downstream contribution of necessity-guided updates to adaptation and retention. Varying visual-dependence levels further examines the conditions for benefit.

Structured and efficient representation learning. Applications span video super-resolution [Zhang et al., 2023a], facial-expression recognition [Zhang et al., 2024b], and

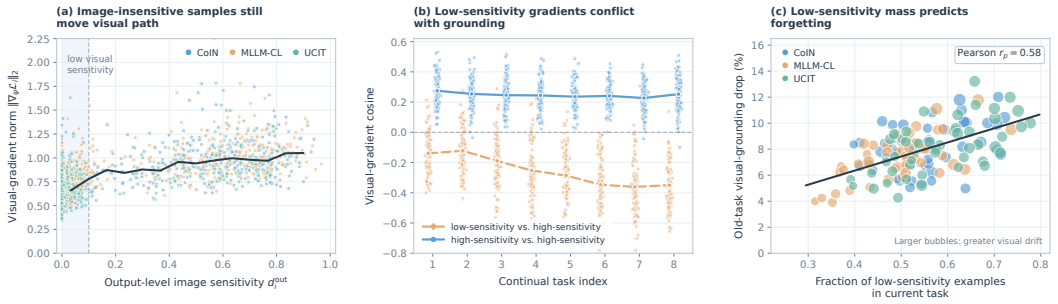

Figure 2: **Preliminary evidence of visual-path contamination. (a)** Many image-present examples have low output-level image sensitivity yet induce non-negligible visual-path gradients. **(b)** Their gradients are often weakly or negatively aligned with visually grounded updates. **(c)** Across continual streams, a larger fraction of low-sensitivity examples is associated with stronger visual-path drift and greater old-task visual-grounding forgetting under standard tuning. These patterns suggest that visually unqualified supervision can contaminate the visual pathway.

medical segmentation [Zhang et al., 2026d]. Instance-aware alignment preserves super-resolution semantics [Guo et al., 2026]. TimeSAF separates unimodal learning from cross-modal fusion for LLM-guided forecasting [Zhang et al., 2026c]; RACER enables explainable multi-agent graph reasoning [Lou et al., 2026]. Replay supports continual threat attribution [Wang et al., 2026] and multi-scale self-supervision supports provenance-graph anomaly detection [Ma et al., 2026]. Efficient designs use recoverable pruning for modulation recognition [Lu et al., 2026], forecasting activation functions [Zhang et al., 2026b], efficient attention [Wen et al., 2026, 2025], and training-free long-context prompt compression [Gao et al., 2026]. Related application settings include agentic benchmark auditing [Yang et al., 2026b], privacy-aware LLM alignment [Yang et al., 2026a], optical-network automation [Xiao et al., 2026b,c], communication-limited UAV coordination [Fan et al., 2026], swarm-based GPS spoofing detection [Fan et al., 2023], semantic-signal-guided reverse logistics [He et al., 2026], and stability-aware retail forecasting [Li et al., 2026]. VNG-CT addresses supervision validity by using visual necessity to route each sample's gradients across language, visual, and calibration paths.

3 Preliminary Study

Motivation. Many image-present instruction-following examples have answers that are insensitive to visual evidence. We examine whether these examples can still move the visual path during standard continual tuning and impair subsequent visual grounding.

Diagnostic protocol. We use standard continual tuning and post-hoc measurements, with routing, filtering, and path-specific update rules disabled. Given an incoming example (x_i, q_i, a_i) and a pretrained multimodal model f_{θ_0}, we measure output-level image sensitivity by comparing the model response under the original image and perturbed visual inputs:

$$d_i^{\text{out}} = 1 - \max_{\tilde{x} \in \{\varnothing,\, \text{shuffle}(x_i),\, x_j\}} \text{Sim}\left(f_{\theta_0}(x_i, q_i), f_{\theta_0}(\tilde{x}, q_i)\right), \tag{1}$$

where $\varnothing$ denotes a null image, $\text{shuffle}(x_i)$ denotes a patch-shuffled image, x_j is a randomly mismatched image from the same task, and $\text{Sim}(\cdot,\cdot)$ is token-level answer similarity. A small d_i^{out} indicates that the model output is almost unchanged after removing or corrupting the image, suggesting that the example is visually insensitive at the output level.

We then compute the visual-path gradient norm that the same example would induce under ordinary supervised tuning:

$$m_i^{\text{vis}} = \left\|\nabla_\psi \mathcal{L}_{\text{sft}}\left(f_{\theta_0}(x_i, q_i), a_i\right)\right\|_2, \tag{2}$$

where ψ denotes the visual projector and cross-modal adaptation layers. Finally, for each mini-batch, we compare the average visual gradient of low-sensitivity examples with that of high-sensitivity

examples:

$$\kappa_b = \cos\left(\frac{1}{|\mathcal{B}_b^{\text{low}}|}\sum_{i\in\mathcal{B}_b^{\text{low}}}\nabla_\psi\mathcal{L}_i, \frac{1}{|\mathcal{B}_b^{\text{high}}|}\sum_{i\in\mathcal{B}_b^{\text{high}}}\nabla_\psi\mathcal{L}_i\right). \tag{3}$$

Here, $\mathcal{B}_b^{\text{low}}$ and $\mathcal{B}_b^{\text{high}}$ are the bottom and top sensitivity quantiles within mini-batch b. This measures whether visually insensitive examples push the visual path in directions compatible with visually grounded examples. Refer to §B.1 for more details.

Observations. Figure 2 reveals three consistent patterns. First, many image-present examples are nearly invariant to visual perturbations but still induce substantial visual-path gradients, showing that visually insensitive samples can affect the visual path under standard tuning. Second, their gradients are often poorly aligned with those from genuinely image-sensitive examples: low–high gradient cosine is frequently near zero or negative, while high–high compatibility remains positive. This indicates that visually insensitive examples can interfere with visual-grounding updates. Third, tasks with a larger fraction of low-sensitivity examples exhibit stronger subsequent visual forgetting and visual-path drift, suggesting that repeated updates from language-dominant supervision accumulate and degrade old-task visual grounding.

Implication. The preliminary study motivates path-specific supervision assignment. Low-sensitivity examples can contribute to language style, domain wording, and instruction following. We therefore retain all examples for language adaptation, concentrate visual-path updates on visually dependent examples, and route low-sensitivity examples to calibration.

4 Method

We propose **Visual-Necessity-Gated Continual Tuning** (VNG-CT), a path-specific adaptation framework for multimodal continual learning. At each continual step, the model estimates whether the target answer depends on the image by comparing the supervised answer loss under the true image and under an image-ablated input. This counterfactual dependence score is then used to route gradients: all examples update the language path, visually necessary examples update the visual path, and low-necessity examples update a small calibration path that learns invariance to irrelevant visual evidence. All examples are retained, with their training signals assigned to the parameter paths they can validly supervise. The pipeline is shown in Figure 3.

4.1 Continual Multimodal Tuning with Parameter Paths

Multimodal continual tuning is usually performed with a single supervised objective applied to all trainable parameters. This update rule ignores whether a sample's target answer actually depends on visual evidence. As a result, language-dominant examples can update visual projectors and cross-modal adapters even when their answer can be predicted from the instruction alone. We make this failure mode explicit by separating trainable parameters according to the pathway they control.

We consider a stream of multimodal instruction datasets

$$\mathcal{D}_{1:T} = \{\mathcal{D}_1, \ldots, \mathcal{D}_T\}, \qquad \mathcal{D}_t = \{(x_i^t, q_i^t, a_i^t)\}_{i=1}^{n_t}, \qquad \theta = (\psi, \omega, \chi). \tag{4}$$

Here, x_i^t is the image, q_i^t is the instruction, and $a_i^t = (a_{i,1}^t, \ldots, a_{i,L_i}^t)$ is the target answer. The trainable parameter groups are: the visual path ψ, containing the visual projector and cross-modal LoRA layers; the language path ω, containing text-side LoRA layers; and the calibration path χ, containing a low-rank residual calibrator after the visual projector. The vision encoder and the backbone language model weights are frozen.

For each example, the supervised tuning loss is

$$\mathcal{L}_i(\theta) = -\frac{1}{L_i}\sum_{\ell=1}^{L_i}\log p_\theta\left(a_{i,\ell}^t \mid a_{i,<\ell}^t, x_i^t, q_i^t\right). \tag{5}$$

The normalization by L_i makes the loss comparable across answers of different lengths. Standard continual tuning applies $\nabla_\theta\mathcal{L}_i$ uniformly to all trainable paths. VNG-CT weights each path according to the example's visual necessity.

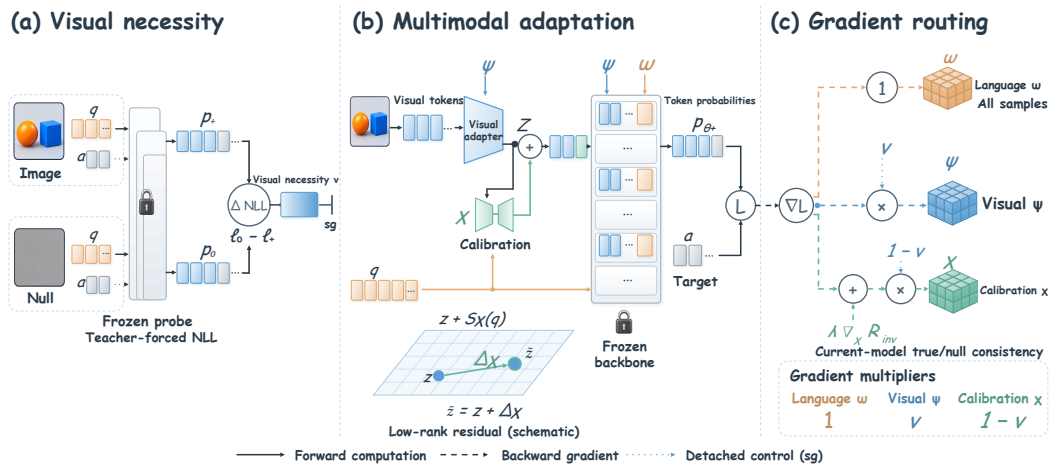

Figure 3: **Overview of VNG-CT. (a)** A frozen probe maps the length-normalized answer-NLL gap $\ell^0 - \ell^+$ between null-image and true-image views to a detached visual-necessity gate ν. **(b)** Calibration adds an instruction-conditioned low-rank residual to projected visual tokens before prediction with a frozen backbone and trainable adapters. The plane schematically depicts residual addition. **(c)** Each example's gradients update the language (ω), visual (ψ), and calibration (χ) paths with weights 1, ν, and $1 - \nu$, respectively. The current-model true/null consistency loss updates only χ, using detached null-image predictions as the reference. The gate scales backward updates; all examples retain full language supervision. Sample and task indices are omitted for clarity.

4.2 Counterfactual Visual-Necessity Estimation

Target-answer loss alone is insufficient to determine visual dependence. We measure visual necessity through the change in target-answer likelihood when the image is replaced with a neutral visual input.

At the start of task t, we freeze the scoring snapshot $\bar{\theta}_t = \mathrm{sg}(\theta_{t-1})$ and keep it fixed throughout the task to compute visual-necessity scores. Let $\varnothing$ denote a fixed null image. We define

$$\begin{aligned} \ell_i^+ &= \ell_{\bar{\theta}_t}(a_i^t \mid x_i^t, q_i^t), \qquad \ell_i^0 = \ell_{\bar{\theta}_t}(a_i^t \mid \varnothing, q_i^t), \\ s_i^t &= \ell_i^0 - \ell_i^+, \qquad \nu_i^t = \mathrm{sg}\left[\mathrm{clip}_{[0,1]}\left(\frac{s_i^t - \rho}{\tau_\nu}\right)\right]. \end{aligned} \tag{6}$$

The loss $\ell_{\bar{\theta}_t}$ uses the same token-level negative log-likelihood as Eq. (5). The score s_i^t measures how much worse the model predicts the target answer after image ablation. The margin $\rho \geq 0$ removes weak loss gaps, and $\tau_\nu > 0$ controls the softness of the transition from suppressed to active visual-path updates. The stop-gradient prevents the model from changing its own gate during the current update. Appendix A.1 gives the exact scoring procedure, snapshot schedule, and counterfactual variants used for robustness analysis.

4.3 Calibration for Low-Necessity Examples

Low-necessity examples carry useful supervision for response style, domain wording, instruction following, and robustness to irrelevant images. We route them to a small calibration path that reduces sensitivity to irrelevant visual inputs while keeping the visual projector stable.

The calibration path χ is a low-rank residual module applied after the visual projector. Its architecture is given in Appendix A.2. During training, we encourage low-necessity examples to produce consistent next-token distributions with the true image and with the null image. Let $p_{i,\ell}^+ = p_\theta(\cdot \mid a_{i,<\ell}^t, x_i^t, q_i^t)$ and $p_{i,\ell}^0 = p_\theta(\cdot \mid a_{i,<\ell}^t, \varnothing, q_i^t)$. The invariance regularizer is

$$\mathcal{R}_i^{\mathrm{inv}} = \frac{1}{L_i} \sum_{\ell=1}^{L_i} D_{\mathrm{KL}}\left(\mathrm{sg}[p_{i,\ell}^0] \,\middle\|\, p_{i,\ell}^+\right). \tag{7}$$

This regularizer is applied only to the calibration path in Eq. (9). The stop-gradient on $p^0_{i,\ell}$ makes the null-image prediction a stable reference distribution.

4.4 Path-Specific Gradient Routing

A scalar sample weight cannot capture this path-specific behavior: global downweighting would discard useful language supervision, while full weighting would still update the visual path. VNG-CT therefore uses path-specific weights, allowing the same example to train language behavior while being suppressed for visual-path adaptation. For each example, the path weights are

$$(w_i^{\text{lang}}, w_i^{\text{vis}}, w_i^{\text{cal}}) = (1, \nu_i^t, 1 - \nu_i^t). \tag{8}$$

Given a mini-batch $\mathcal{B}_t \subset \mathcal{D}_t$, the routed gradients are

$$\begin{aligned} G_\omega &= \frac{1}{|\mathcal{B}_t|} \sum_{i \in \mathcal{B}_t} \nabla_\omega \mathcal{L}_i(\theta), & G_\psi &= \frac{1}{|\mathcal{B}_t|} \sum_{i \in \mathcal{B}_t} \nu_i^t \nabla_\psi \mathcal{L}_i(\theta), \\ G_\chi &= \frac{1}{|\mathcal{B}_t|} \sum_{i \in \mathcal{B}_t} (1 - \nu_i^t) \nabla_\chi \left[\mathcal{L}_i(\theta) + \lambda_{\text{inv}} \mathcal{R}_i^{\text{inv}} \right]. \end{aligned} \tag{9}$$

The language path receives all examples. The visual path receives gradients in proportion to visual necessity. The calibration path receives stronger updates from low-necessity examples and learns invariance through $\mathcal{R}_i^{\text{inv}}$. This vector-valued routing is the key distinction from data filtering and scalar sample weighting; a mathematical comparison is provided in Appendix A.4.

4.5 Continual Update and Inference

The routed gradients in Eq. (9) preserve visual plasticity for visually necessary examples while restricting drift from language-dominant supervision directly at the optimizer level. For each mini-batch, we update the three trainable paths as

$$\omega \leftarrow \omega - \eta_\omega G_\omega, \qquad \psi \leftarrow \psi - \eta_\psi G_\psi, \qquad \chi \leftarrow \chi - \eta_\chi G_\chi, \tag{10}$$

where η_ω, η_ψ, and η_χ are path-specific learning rates. After all mini-batches of task t are processed, the updated parameter tuple defines $\theta_t = (\psi_t, \omega_t, \chi_t)$. The full training algorithm is shown in Appendix A.5.

At inference time, the visual-necessity probe is not used. The final model predicts with the full multimodal input using $p_{\theta_T}(a \mid x, q)$. Thus, VNG-CT changes only the training dynamics and introduces no inference-time routing cost.

5 Experiments

5.1 Experimental Setup

Benchmarks and metrics. We evaluate VNG-CT on three continual multimodal instruction-tuning settings: CoIN [Chen et al., 2024a], MLLM-CL [Zhao et al., 2025], and UCIT-style mixed instruction streams used by recent MCIT methods [Guo et al., 2025a]. These benchmarks cover visual question answering, OCR-oriented reasoning, classification, grounding, captioning, multimodal science reasoning, domain-incremental adaptation, and ability-incremental adaptation. We use the official task-specific metrics for each benchmark and report standard continual learning metrics, including final average accuracy, mean average accuracy, backward transfer, and forgetting. To evaluate whether VNG-CT mitigates visual-path contamination, we additionally report visual forgetting, counterfactual visual sensitivity, and visual-necessity ranking quality in §B.4.

Comparison protocol. We compare VNG-CT with standard continual learning baselines, parameter-efficient tuning baselines, and recent MLLM continual instruction-tuning methods, including Fwd-Prompt [Zheng et al., 2024], ModalPrompt [Zeng et al., 2025a], CL-MoE [Huai et al., 2025], ProgLoRA [Yu et al., 2025], HiDe-LLaVA [Guo et al., 2025a], BranchLoRA [Zhang et al., 2025], SEFE [Chen et al., 2025], SMoLoRA [Wang et al., 2025a], Instruction-Grounded Visual Projectors [Jin et al., 2025], LiLoRA [Che et al., 2026], and SAME [Xie et al., 2026]. See §B.5 for more details.

Table 1: **Main comparison on CoIN and UCIT.** Best and second-best values in each metric column are shown in **bold** and underlined, respectively.

Method	MLLM Backbone	CoIN ACC ↑	CoIN MAA ↑	CoIN BWT ↑	UCIT FAA ↑	UCIT CAA ↑	UCIT FM ↓
General continual learning and PEFT baselines							
LoRA [Hu et al., 2022]	LLaVA-v1.5-7B	28.74	32.97	-32.62	60.86	67.61	17.44
O-LoRA [Wang et al., 2023]	LLaVA-v1.5-7B	36.42	40.19	-28.36	64.63	75.98	14.87
LwF [Li and Hoiem, 2017]	LLaVA-v1.5-7B	35.75	39.43	-29.21	58.66	66.19	18.63
EWC [Kirkpatrick et al., 2017]	LLaVA-v1.5-7B	36.62	40.05	-28.58	59.45	66.82	17.85
Recent multimodal continual instruction tuning methods							
Continual LLaVA [Cao et al., 2024]	LLaVA-v1.5-7B	53.33	45.56	-23.94	61.54	74.14	16.02
Fwd-Prompt [Zheng et al., 2024]	InstructBLIP-FlanT5XL	54.34	47.24	-21.88	62.27	75.12	16.47
ModalPrompt [Zeng et al., 2025a]	LLaVA-v1.5-7B	55.06	54.28	-3.59	52.53	57.67	**6.92**
CL-MoE [Huai et al., 2025]	LLaVA-v1.5-7B	59.23	50.49	-18.72	60.07	74.89	18.25
ProgLoRA [Yu et al., 2025]	LLaVA-v1.5-7B	59.09	62.38	-6.59	63.94	77.15	16.95
SEFE [Chen et al., 2025]	LLaVA-v1.5-7B	58.57	52.14	-12.33	65.17	76.64	14.82
HiDe-LLaVA [Guo et al., 2025a]	LLaVA-v1.5-7B	63.95	56.64	-8.47	65.93	79.03	16.76
BranchLoRA [Zhang et al., 2025]	LLaVA-v1.5-7B	44.20	49.94	-20.98	66.35	79.34	16.71
SMoLoRA [Wang et al., 2025a]	LLaVA-v1.5-7B	64.97	57.28	-6.02	67.60	80.42	16.39
Instr.-Grounded VP [Jin et al., 2025]	Vicuna-7B	65.11	57.86	-5.78	68.01	80.75	16.44
LiLoRA [Che et al., 2026]	LLaVA-v1.5-7B	65.82	58.16	-4.77	68.92	81.52	15.98
SAME [Xie et al., 2026]	LLaVA-v1.5-7B	66.82	59.32	-3.97	72.53	81.93	11.91
Ours							
VNG-CT	LLaVA-v1.5-7B	68.61 ± 0.14	62.69 ± 0.08	-1.68 ± 0.12	**76.47 ± 0.35**	**85.09 ± 0.33**	10.79 ± 0.01
VNG-CT	InternVL-Chat-7B	**69.60 ± 0.23**	**63.34 ± 0.31**	**-1.21 ± 0.10**	75.35 ± 0.33	84.69 ± 0.38	11.68 ± 0.02
VNG-CT	Qwen2-VL-7B	66.35 ± 0.16	60.06 ± 0.27	-2.42 ± 0.15	74.26 ± 0.29	83.62 ± 0.39	11.86 ± 0.02

Table 2: **Comparison on MLLM-CL and visual-path diagnostics.** DCL and ACL denote domain continual learning and ability continual learning. Reported DCL/ACL values are averaged from the task-wise Last/Average scores. VFR and VCS are defined in Appendix B.4. Best and second-best values in each metric column are shown in **bold** and underlined, respectively.

Method	MLLM Backbone	DCL Last ↑	DCL Avg. ↑	ACL Last ↑	ACL Avg. ↑	VFR ↓	VCS ↑
LoRA [Hu et al., 2022]	LLaVA-v1.5-7B	28.49	45.61	23.80	27.38	18.58	21.24
O-LoRA [Wang et al., 2023]	LLaVA-v1.5-7B	48.28	58.37	32.25	35.54	12.69	28.62
L2P [Wang et al., 2022]	LLaVA-v1.5-7B	50.59	59.23	30.49	33.78	13.35	27.72
ModalPrompt [Zeng et al., 2025a]	LLaVA-v1.5-7B	50.36	55.99	37.06	34.52	11.78	30.96
HiDe-LLaVA [Guo et al., 2025a]	LLaVA-v1.5-7B	43.90	52.60	35.19	34.16	9.34	34.58
MR-LoRA [Zhao et al., 2025]	LLaVA-v1.5-7B	67.33	69.65	39.87	37.83	8.16	36.27
SAME [Xie et al., 2026]	LLaVA-v1.5-7B	66.69	68.31	40.36	37.45	5.48	41.66
VNG-CT	LLaVA-v1.5-7B	69.96	71.38	42.46	40.11	3.36	45.89
VNG-CT	InternVL-Chat-7B	**71.21**	**72.59**	**43.73**	**41.23**	**3.09**	**47.09**
VNG-CT	Qwen2-VL-7B	66.84	68.46	39.32	37.97	4.02	43.85

Backbones and implementation. Our main backbone is LLaVA-1.5-7B [Liu et al., 2023], with additional robustness evaluation on InternVL-Chat-7B [Chen et al., 2024b] and Qwen2-VL [Wang et al., 2024]. The vision encoder and base language model are frozen, while LoRA modules and the proposed calibration path are trained continually. Unless otherwise specified, we use LoRA rank 16, AdamW, cosine learning-rate decay, bfloat16 precision, and three random seeds. Full implementation details, hyperparameters, ablations, and evaluation procedures are provided in §B.

5.2 Main Results

Table 1 shows strong continual-learning performance of VNG-CT on CoIN and UCIT. On CoIN, SAME has the best reported prior result, and VNG-CT further improves final average accuracy under the target protocol. On UCIT, VNG-CT achieves the highest FAA and CAA, indicating strong final retention and new-task adaptation. ModalPrompt has the lowest FM (6.92), followed by VNG-CT on LLaVA (10.79). These results suggest that explicitly preventing visually unnecessary examples from updating the visual path provides a complementary advantage over expert stabilization, prompt routing, and post-hoc parameter merging.

Table 2 further evaluates domain- and ability-level continual adaptation. The reported MLLM-CL results show that parameter isolation with routing is a strong baseline, especially under domain shifts. VNG-CT improves over these methods under the target protocol, and the diagnostic metrics show

substantially lower visual forgetting and higher counterfactual visual sensitivity. This supports the contribution of assigning supervision to the parameter path it can validly supervise, beyond the benefit of additional parameters.

5.3 Ablation Study and Analysis

Single-factor ablation. Table 3 studies the contribution of each design component in VNG-CT. All ablations use LLaVA-v1.5-7B as the backbone. Table 3 shows that each component contributes to VNG-CT, with the visual-necessity gate being the most critical. Removing or randomizing the gate causes the largest drops, reducing UCIT FAA by 3.89 and 4.51 points and substantially weakening visual diagnostics. Hard gating and global weighting are better but still inferior to the full model, indicating that soft, path-specific routing is more effective than binary filtering or scalar reweighting. Removing the calibration path mainly harms visual stability, increasing VFR by 1.18 points, which suggests that calibration helps absorb low-necessity supervision without drifting the visual interface.

Table 3: **Ablation study of VNG-CT.** All results use LLaVA-v1.5-7B. $\mathrm{VNR}_{\mathrm{route}}$ evaluates the actual visual routing weights; LNVL uses independent reference necessity scores (Appendix B.4). Blue numbers in parentheses denote the change relative to the full VNG-CT model. For ↑ metrics, negative changes indicate degradation; for ↓ metrics, positive changes indicate degradation.

Variant	CoIN ACC ↑	CoIN MAA ↑	UCIT FAA ↑	DCL Avg. ↑	VFR ↓	VCS ↑	$\mathrm{VNR}_{\mathrm{route}}$ ↑	LNVL ↓
No gate	65.74 (-2.87)	59.38 (-3.31)	72.58 (-3.89)	67.43 (-3.95)	6.42 (+3.06)	40.12 (-5.77)	50.00 (-34.12)	0.88 (+0.73)
Hard gate	67.42 (-1.19)	61.21 (-1.48)	74.93 (-1.54)	70.12 (-1.26)	4.21 (+0.85)	43.72 (-2.17)	82.35 (-1.77)	0.23 (+0.08)
Random gate	64.98 (-3.63)	58.71 (-3.98)	71.96 (-4.51)	66.88 (-4.50)	7.08 (+3.72)	38.94 (-6.95)	51.36 (-32.76)	0.62 (+0.47)
Global weighting	66.83 (-1.78)	60.46 (-2.23)	73.81 (-2.66)	68.92 (-2.46)	4.12 (+0.76)	43.16 (-2.73)	83.08 (-1.04)	0.31 (+0.16)
No calibration	67.56 (-1.05)	61.53 (-1.16)	74.84 (-1.63)	70.25 (-1.13)	4.54 (+1.18)	42.89 (-3.00)	83.61 (-0.51)	0.18 (+0.03)
VNG-CT	**68.61**	**62.69**	**76.47**	**71.38**	**3.36**	**45.89**	**84.12**	**0.15**

Gate validity and visual-path contamination. We examine whether the gate tracks image dependence and identifies potentially contaminating visual gradients. Figure 4(a) shows that the gate tracks the image-ablation score drop. Because both quantities use image removal, this is an internal consistency check. Panels (b,c) show that low-ν examples still produce non-negligible visual gradients with weak or negative alignment to grounding-preserving anchors. Thus, gradient magnitude alone does not determine whether a sample should update the visual path.

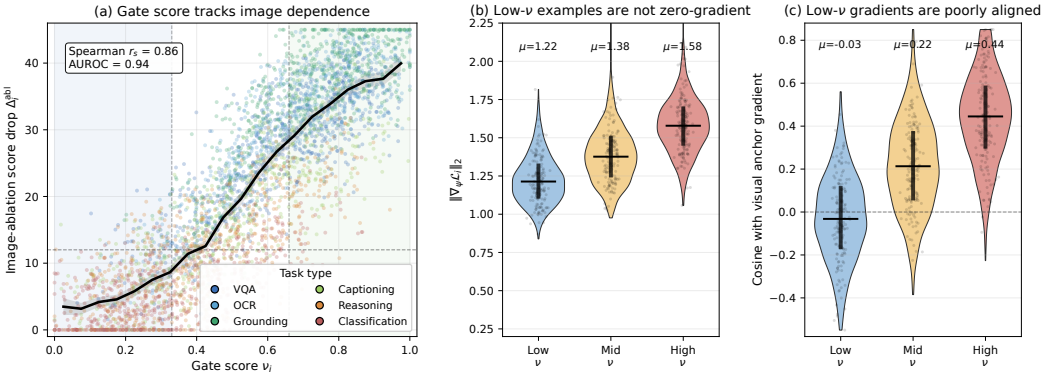

Figure 4: **Gate validity and visual-path contamination.** The gate score correlates with image-ablation drop; low-ν examples still induce visual-path gradients, but these gradients are poorly aligned with visually grounded anchors. See Appendix B.9, *Gate-validity protocol* for implementation details.

Independent human validation. Three authors independently annotate 900 held-out examples (300 per benchmark), blinded to gate scores, null-image outputs, and attention maps. Table 4 reports human AUROC 0.891 on LLaVA versus 0.758 for image-token attribution, with FNR reduced from 22.6% to 7.8%. AUROC spans 0.873–0.902 across three backbones, supporting agreement between the model-relative gate and human visual dependence. See Appendix C.2 for validation details.

Table 4: **Independent human validation of the gate.** Results on 900 author-annotated examples. AUROC and AUPRC measure visual-dependence ranking; Spearman measures correlation with human image benefit. FNR uses human visual-dependence labels at threshold 0.35; ECE is expected calibration error. Best and second-best values are bold and underlined.

Estimator	Human agreement			Gate reliability	
	AUROC ↑	AUPRC ↑	Spearman ↑	FNR (%) ↓	ECE ↓
Raw target NLL	0.642	0.671	0.281	34.7	0.171
Visual-gradient norm	0.704	0.729	0.394	28.9	0.143
Image-token attribution	0.758	0.781	0.486	22.6	0.112
VNG, LLaVA-1.5-7B	<u>0.891</u>	<u>0.912</u>	<u>0.742</u>	<u>7.8</u>	<u>0.055</u>
VNG, InternVL-Chat-7B	**0.902**	**0.921**	**0.758**	**7.1**	**0.051**
VNG, Qwen2-VL-7B	0.873	0.896	0.711	9.4	0.063

Path-specific routing versus global weighting. We compare path-specific routing and global sample weighting in terms of visual forgetting and new-task adaptation, seeking low VFR and high NTA. Figure 5 shows that global weighting reduces visual forgetting relative to standard tuning, but also weakens new-task adaptation because it suppresses low-necessity examples for all trainable paths, including the language path. VNG-CT lies on the Pareto frontier by restricting low-necessity gradients on the visual path while keeping these examples active for language learning. The calibration residual further absorbs low-necessity supervision under an explicit image-invariance constraint. This confirms the benefit of path-specific supervision assignment over global downweighting of visually weak samples.

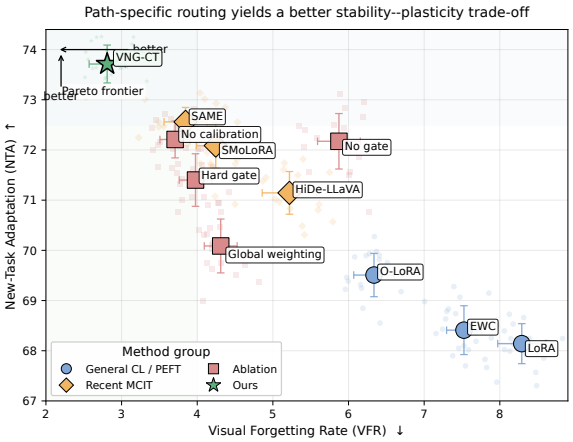

Figure 5: **Visual forgetting–adaptation trade-off.** VNG-CT achieves a better Pareto trade-off than global weighting by preserving language-side plasticity while selectively restricting visual-path updates.

6 Conclusion

This paper identifies visual-path contamination as a key challenge in multimodal continual tuning, where image-present but visually unnecessary examples can drift the visual interface. We propose VNG-CT, which estimates visual necessity and routes supervision to language, visual, and calibration paths according to what each example can validly teach. Experiments show that VNG-CT better preserves old-task visual grounding while maintaining strong new-task adaptation. These results suggest that continual multimodal learning should distinguish modality presence from supervision validity. A promising direction is to extend visual necessity from sample-level routing to more fine-grained token-, region-, or relation-level supervision assignment in long-horizon multimodal adaptation.

Acknowledgments and Disclosure of Funding

This work is supported by Natural Science Foundation of China under Grant 62606079, the Fundamental Research Funds for the Central Universities (N2623009).

References

Rawan AlSaad, Alaa Abd-Alrazaq, Sabri Boughorbel, Arfan Ahmed, Max-Antoine Renault, Rafat Damseh, and Javaid Sheikh. Multimodal large language models in health care: applications, challenges, and future outlook. *Journal of Medical Internet Research*, 26:e59505, 2024.

Remi Cadene, Corentin Dancette, Matthieu Cord, Devi Parikh, et al. Rubi: Reducing unimodal biases for visual question answering. *Advances in Neural Information Processing Systems*, 32, 2019.

Meng Cao, Yuyang Liu, Yingfei Liu, Tiancai Wang, Jiahua Dong, Henghui Ding, Xiangyu Zhang, Ian Reid, and Xiaodan Liang. Continual llava: Continual instruction tuning in large vision-language models. *arXiv preprint arXiv:2411.02564*, 2024.

Chang Che, Ziqi Wang, Pengwan Yang, Cheems Wang, Hui Ma, and Zenglin Shi. Lora in lora: Towards parameter-efficient architecture expansion for continual visual instruction tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40(24), pages 19978–19986, 2026.

Cheng Chen, Junchen Zhu, Xu Luo, Heng T Shen, Jingkuan Song, and Lianli Gao. Coin: A benchmark of continual instruction tuning for multimodel large language models. *Advances in Neural Information Processing Systems*, 37:57817–57840, 2024a.

Jinpeng Chen, Runmin Cong, Yuzhi Zhao, Hongzheng Yang, Guangneng Hu, Horace Ho Shing Ip, and Sam Kwong. Sefe: Superficial and essential forgetting eliminator for multimodal continual instruction tuning. In *Proceedings of the 42nd International Conference on Machine Learning*, pages 7982–8001. ML Research Press, 2025.

Qianyu Chen, Ziteng Feng, Canran Xiao, and Runxuan Tang. InduceKV: Fixed-footprint continual adaptation of multimodal LLMs via inducing KV memories. In *Advances in Neural Information Processing Systems*, 2026a.

Qianyu Chen, Canran Xiao, and Runxuan Tang. Hidden forgetting in continual multimodal learning: When accuracy survives but grounding fails. In *Advances in Neural Information Processing Systems*, 2026b.

Qibai Chen and Zeming Liu. SemVerBench: Benchmarking LLM comprehension of version-constraint resolution semantics. arXiv preprint arXiv:2609.11180, 2026.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198, 2024b.

Riccardo Del Chiaro, Bartłomiej Twardowski, Andrew Bagdanov, and Joost Van de Weijer. Ratt: Recurrent attention to transient tasks for continual image captioning. *Advances in Neural Information Processing Systems*, 33:16736–16748, 2020.

Kairui Ding, Boyuan Chen, Yuchen Su, Huan-ang Gao, Bu Jin, Chonghao Sima, Xiaohui Li, Wuqiang Zhang, Paul Barsch, Hongyang Li, and Hao Zhao. Hint-AD: Holistically aligned interpretability in end-to-end autonomous driving. In *Proceedings of the 8th Conference on Robot Learning*, volume 270 of *Proceedings of Machine Learning Research*, pages 3742–3765. PMLR, 2025.

Enguang Fan, Anfeng Peng, Matthew Caesar, Jae Kim, Josh Eckhardt, Greg Kimberly, and Denis Osipychev. Towards effective swarm-based GPS spoofing detection in disadvantaged platforms. In *Proceedings of the 2023 IEEE Military Communications Conference*, pages 722–728, 2023. doi: 10.1109/MILCOM58377.2023.10356314.

Enguang Fan, Yifan Chen, Zihan Shan, Klara Nahrstedt, Matthew Caesar, and Jae Kim. Communication-aware multi-agent reinforcement learning for decentralized cooperative UAV deployment. *arXiv preprint arXiv:2603.16141*, 2026.

Yaxin Gao, Yao Lu, Zongfei Zhang, Jiaqi Nie, Shanqing Yu, and Qi Xuan. DSPC: Dual-stage progressive compression framework for efficient long-context reasoning. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 19387–19391. IEEE, 2026.

Saurabh Garg, Mehrdad Farajtabar, Hadi Pouransari, Raviteja Vemulapalli, Sachin Mehta, Oncel Tuzel, Vaishaal Shankar, and Fartash Faghri. Tic-clip: Continual training of clip models. In *Proceedings of the 12th International Conference on Learning Representations*, 2024.

Haiyang Guo, Fanhu Zeng, Ziwei Xiang, Fei Zhu, Da-Han Wang, Xu-Yao Zhang, and Cheng-Lin Liu. Hide-llava: Hierarchical decoupling for continual instruction tuning of multimodal large language model. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13572–13586, 2025a.

Haiyang Guo, Fei Zhu, Hongbo Zhao, Fanhu Zeng, Wenzhuo Liu, Shijie Ma, Da-Han Wang, and Xu-Yao Zhang. Mcitlib: Multimodal continual instruction tuning library and benchmark. *arXiv preprint arXiv:2508.07307*, 2025b.

Zixin Guo, Kai Zhao, and Luyan Zhang. InstanceRSR: Real-world super-resolution via instance-aware representation alignment. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 10577–10581. IEEE, 2026.

Jiani He, Dingyan Shang, Yihua Xu, Shiqi Huang, Yan Lyu, Jize Li, and Shangjing Tang. Semantic signal-assisted inspection and recovery allocation in reverse logistics. *arXiv preprint arXiv:2609.02116*, 2026.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *Proceedings of the 10th International Conference on Learning Representations*, 2022.

Saisai Hu. Research on security enhancement methods for adversarial robust large language model intelligent agents for medical decision-making tasks. In *Proceedings of the 7th International Seminar on Artificial Intelligence, Networking and Information Technology*, pages 599–603. IEEE, 2026.

Tianyu Huai, Jie Zhou, Xingjiao Wu, Qin Chen, Qingchun Bai, Ze Zhou, and Liang He. Cl-moe: Enhancing multimodal large language model with dual momentum mixture-of-experts for continual visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19608–19617, 2025.

Linlan Huang, Xusheng Cao, Haori Lu, Yifan Meng, Fei Yang, and Xialei Liu. Mind the gap: Preserving and compensating for the modality gap in clip-based continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3777–3786, 2025.

Shiqi Huang, Jiani He, Dingyan Shang, Yihua Xu, Jize Li, Yan Lyu, and Lakshmi Muraleedharan Nair. DACRI: Decision-aware causal intervention ranking for critical supply chains. *arXiv preprint arXiv:2608.11154*, 2026.

Yukang Huo and Hao Tang. When continue learning meets multimodal large language model: A survey. *arXiv preprint arXiv:2503.01887*, 2025.

Saurav Jha, Dong Gong, and Lina Yao. Clap4clip: Continual learning with probabilistic finetuning for vision-language models. *Advances in Neural Information Processing Systems*, 37:129146–129186, 2024.

Hyundong Jin, Hyung Jin Chang, and Eunwoo Kim. Instruction-grounded visual projectors for continual learning of generative vision-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3466–3476, 2025.

Yujin Jo and Taesup Kim. Memory-free continual learning with null space adaptation for zero-shot vision-language models. In *Proceedings of the 14th International Conference on Learning Representations*, 2026.

Borui Kang, Lei Wang, Zhiping Wu, Tao Feng, Yawen Li, Yang Gao, and Wenbin Li. Dynamic multi-layer null space projection for vision-language continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2077–2086, 2025.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114 (13):3521–3526, 2017.

Sicong Leng, Yun Xing, Zesen Cheng, Yang Zhou, Hang Zhang, Xin Li, Deli Zhao, Shijian Lu, Chunyan Miao, and Lidong Bing. The curse of multi-modalities: Evaluating hallucinations of large multimodal models across language, visual, and audio. *arXiv preprint arXiv:2410.12787*, 2024.

Jize Li, Jiani He, Dishu Yang, Dingyan Shang, Jingjing Liu, and Shiqi Huang. Accuracy-preserving stability regularization for large-scale retail demand forecasting. *arXiv preprint arXiv:2607.13331*, 2026.

Pengfei Li, Beiwen Tian, Yongliang Shi, Xiaoxue Chen, Hao Zhao, Guyue Zhou, and Ya-Qin Zhang. TOIST: Task oriented instance segmentation transformer with noun-pronoun distillation. *Advances in Neural Information Processing Systems*, 35:17597–17611, 2022.

Yang Li, Xiaoxue Chen, Hao Zhao, Jiangtao Gong, Guyue Zhou, Federico Rossano, and Yixin Zhu. Understanding embodied reference with Touch-Line Transformer. In *Proceedings of the 11th International Conference on Learning Representations*, 2023.

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

Fuxiao Liu, Kevin Lin, Linjie Li, Jianfeng Wang, Yaser Yacoob, and Lijuan Wang. Mitigating hallucination in large multi-modal models via robust instruction tuning. In *Proceedings of the 12th International Conference on Learning Representations*, 2024.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in Neural Information Processing Systems*, 36:34892–34916, 2023.

Jingjing Liu, Dishu Yang, Hongbo Qin, and Hansong Zhang. Passing tests, missing semantics: Evidence of interpretive lock-in in plan-based DataFrame code generation. *arXiv preprint arXiv:2610.15797*, 2026.

Wenzhuo Liu, Fei Zhu, Longhui Wei, and Qi Tian. C-clip: Multimodal continual learning for vision-language model. In *Proceedings of the 13th International Conference on Learning Representations*, 2025.

Yuwei Lou, Hao Hu, Yuzhou Jiang, Zongfei Zhang, Liang Wang, Jincai Liu, Jidong Ge, and Xianping Tao. RACER: Reinforced agent collaboration for explainable reasoning on knowledge graphs. *arXiv preprint arXiv:2608.29263*, 2026.

Yao Lu, Tengfei Ma, Zeyu Wang, Zhuangzhi Chen, Dongwei Xu, Yun Lin, Qi Xuan, and Guan Gui. FCOS: A two-stage recoverable model pruning framework for automatic modulation recognition. *IEEE Transactions on Cognitive Communications and Networking*, 2026.

Mingjun Ma, Saisai Hu, Tiantian Zhu, Yitian Lin, Jiachen Sun, Zhengqiu Weng, Dehao Li, and Yajie Zhang. MuSK: Multi-scale knowledge learning for provenance-graph anomaly detection. *Computer Networks*, page 112728, 2026.

Chunlei Meng, Pengbin Feng, Rong Fu, Hoi Leong Lee, Xiaojing Du, Zhaolu Kang, Zeyu Zhang, Weilin Zhou, Chun Ouyang, and Zhongxue Gan. Group cognition learning: Making everything better through governed two-stage agents collaboration. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, pages 87822–87834. PMLR, 2026a.

Chunlei Meng, Jacqueline J. Pang, Pengbin Feng, Zhenyu Yu, Chun Ouyang, and Zhongxue Gan. Rethinking modality reliability in multimodal sentiment analysis with incomplete observations. *arXiv preprint arXiv:2608.03611*, 2026b.

Sainandan Ramakrishnan, Aishwarya Agrawal, and Stefan Lee. Overcoming language priors in visual question answering with adversarial regularization. *Advances in Neural Information Processing Systems*, 31, 2018.

Wanru Shao. Interpretable ensemble learning for network traffic anomaly detection: A SHAP-based explainable AI framework for embedded systems security. In *Proceedings of the 14th International Symposium on Digital Forensics and Security*, pages 1–4. IEEE, 2026a.

Wanru Shao. Policy-driven vulnerability risk quantification framework for large-scale cloud infrastructure data security. In *Proceedings of the 5th International Conference on Cyber Security, Artificial Intelligence and Digital Economy*, pages 40–45, New York, NY, USA, 2026b. Association for Computing Machinery.

Tejas Srinivasan, Ting-Yun Chang, Leticia Pinto Alva, Georgios Chochlakis, Mohammad Rostami, and Jesse Thomason. Climb: A continual learning benchmark for vision-and-language tasks. *Advances in Neural Information Processing Systems*, 35:29440–29453, 2022.

Longxiang Tang, Zhuotao Tian, Kai Li, Chunming He, Hantao Zhou, Hengshuang Zhao, Xiu Li, and Jiaya Jia. Mind the interference: Retaining pre-trained knowledge in parameter efficient continual learning of vision-language models. In *Proceedings of the European Conference on Computer Vision*, pages 346–365. Springer, 2024.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuan-Jing Huang. Orthogonal subspace learning for language model continual learning. In *Findings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10658–10671, 2023.

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

Zijun Wang, Saisai Hu, Tiantian Zhu, Wenrui Cheng, Zhengqiu Weng, Haiting Chen, Xiangyang Zheng, and Suyu Zhang. Trident: Dual-stream APT attribution over heterogeneous threat knowledge graphs. *Computers & Security*, 171:105094, 2026.

Ziqi Wang, Chang Che, Qi Wang, Yangyang Li, Zenglin Shi, and Meng Wang. Smolora: Exploring and defying dual catastrophic forgetting in continual visual instruction tuning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 177–186, 2025a.

Zixuan Wang, Yu Sun, Hongwei Wang, Baoyu Jing, Xiang Shen, Xin Dong, Zhuolin Hao, Hongyu Xiong, and Yang Song. Reasoning-enhanced domain-adaptive pretraining of multimodal large language models for short video content governance. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1104–1112. Association for Computational Linguistics, 2025b.

Yilan Wei, Kumiko Long, Arielle Granston, and Adrian Rodriguez-Contreras. USVexplorer: Robust detection of ultrasonic vocalizations with cross species generalization. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 15232–15236. IEEE, 2026.

Qishuai Wen, Zhiyuan Huang, and Chun-Guang Li. Towards interpretable and efficient attention: Compressing all by contracting a few. In *Advances in Neural Information Processing Systems*, volume 38, pages 160116–160139, 2025.

Qishuai Wen, Zhiyuan Huang, Xianghan Meng, Wei He, and Chun-Guang Li. Mixture-of-Top-k Attention: Efficient attention via scalable fast weights. In *Advances in Neural Information Processing Systems*, 2026.

Junda Wu, Yuxin Xiong, Xintong Li, Yu Xia, Ruoyu Wang, Yu Wang, Tong Yu, Sungchul Kim, Ryan A Rossi, Lina Yao, et al. Mitigating visual knowledge forgetting in mllm instruction-tuning via modality-decoupled gradient descent. *arXiv preprint arXiv:2502.11740*, 8, 2025.

Zhongze Wu, Hongyan Xu, Yitian Long, Shan You, Xiu Su, Jun Long, Yueyi Luo, and Chang Xu. Detecting any instruction-to-answer interaction relationship: Universal instruction-to-answer navigator for Med-VQA. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 53909–53927. PMLR, 2024.

Zhongze Wu, Xiu Su, Feng Yang, Dan Niu, Shan You, Yueyi Luo, and Jun Long. Unlearning without forgetting: Securely removing targeted concepts from large-scale vision-language open-vocabulary detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6271–6281, 2026a.

Zhongze Wu, Xiu Su, Feng Yang, Shan You, Yueyi Luo, and Jun Long. Injection without distortion: Geometrically constrained knowledge enhancement for vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 10897–10905, 2026b.

Canran Xiao, Tianxiang Xu, Siyuan Ma, Yiyang Jiang, Haoyu Gao, and Yuhan Wu. Reversible primitive–composition alignment for continual vision–language learning. In *Proceedings of the 14th International Conference on Learning Representations*, 2026a.

Linqi Xiao, Aparaajitha Gomathinayakam Latha, Venkateswarlu Gudepu, and Andrea Fumagalli. A YANG-grounded LLM agent supporting multi-vendor OpenROADM optical transport networks. In *Proceedings of the ACM SIGCOMM 2026 Conference*, pages 2109–2115, 2026b. doi: 10.1145/3789240.3822606.

Linqi Xiao, Venkateswarlu Gudepu, and Andrea Fumagalli. CORA: A conversational AI-agent for intelligent automation of OpenROADM multi-vendor optical transport networks. In *Applications of Machine Learning 2026*, volume 14218 of *Proceedings of SPIE*, page 142180Y, 2026c. doi: 10.1117/12.3113861.

Zhen-Hao Xie, Jun-Tao Tang, Yu-Cheng Shi, Han-Jia Ye, De-Chuan Zhan, and Da-Wei Zhou. SAME: Stabilized mixture-of-experts for multimodal continual instruction tuning. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, pages 139659–139683. PMLR, 2026.

Yicheng Xu, Yuxin Chen, Jiahao Nie, Yusong Wang, Huiping Zhuang, and Manabu Okumura. Advancing cross-domain discriminability in continual learning of vision-language models. *Advances in Neural Information Processing Systems*, 37:51552–51576, 2024.

Yihua Xu, Yan Lyu, Jize Li, Jiani He, You Lin, Yidan Hu, and Rubing Qiang. When does an added planning capability create operational value? an observable evidence standard. *arXiv preprint arXiv:2610.27305*, 2026. doi: 10.48550/arXiv.2610.27305.

Dishu Yang, Jingjing Liu, and Jize Li. Balancing privacy, utility, and safety in LLM alignment through preference optimization. *arXiv preprint arXiv:2608.30141*, 2026a.

Dishu Yang, Qi Su, Hongbo Qin, and Hansong Zhang. Measurement-first auditing of agentic leaderboards: Contamination susceptibility, matched-control re-evaluation, and scorer validation. *arXiv preprint arXiv:2610.05830*, 2026b.

Dianzhi Yu, Xinni Zhang, Yankai Chen, Aiwei Liu, Yifei Zhang, Philip S Yu, and Irwin King. Recent advances of multimodal continual learning: A comprehensive survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2026.

Jiazuo Yu, Yunzhi Zhuge, Lu Zhang, Ping Hu, Dong Wang, Huchuan Lu, and You He. Boosting continual learning of vision-language models via mixture-of-experts adapters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23219–23230, 2024.

Yahan Yu, Duzhen Zhang, Yong Ren, Xuanle Zhao, Xiuyi Chen, and Chenhui Chu. Progressive lora for multimodal continual instruction tuning. In *Findings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pages 2779–2796, 2025.

Fanhu Zeng, Fei Zhu, Haiyang Guo, Xu-Yao Zhang, and Cheng-Lin Liu. Modalprompt: Towards efficient multimodal continual instruction tuning with dual-modality guided prompt. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 12137–12152, 2025a.

Xinyue Zeng, Haohui Wang, Junhong Lin, Jun Wu, Tyler Cody, and Dawei Zhou. LensLLM: Unveiling fine-tuning dynamics for LLM selection. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 74175–74196. PMLR, 2025b.

Xinyue Zeng, Junhong Lin, Yujun Yan, Feng Guo, Liang Shi, Jun Wu, and Dawei Zhou. HalluGuard: Demystifying data-driven and reasoning-driven hallucinations in LLMs. In *Proceedings of the 14th International Conference on Learning Representations*, 2026.

Changyuan Zhang, Tianxiang Xu, Fei Shen, and Canran Xiao. Interference-aware continual vision–language learning via instance-level expert routing. In *Proceedings of the European Conference on Computer Vision*, volume 17034 of *Lecture Notes in Computer Science*, pages 691–711. Springer Nature Switzerland, 2026a.

Duzhen Zhang, Yahan Yu, Jiahua Dong, Chenxing Li, Dan Su, Chenhui Chu, and Dong Yu. Mm-llms: Recent advances in multimodal large language models. *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12401–12430, 2024a.

Duzhen Zhang, Yong Ren, Zhong-Zhi Li, Yahan Yu, Jiahua Dong, Chenxing Li, Zhilong Ji, and Jinfeng Bai. Enhancing multimodal continual instruction tuning with branchlora. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5743–5756, 2025.

Fan Zhang, Gongguan Chen, Hua Wang, Jinjiang Li, and Caiming Zhang. Multi-scale video super-resolution transformer with polynomial approximation. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(9):4496–4506, 2023a.

Fan Zhang, Gongguan Chen, Hua Wang, and Caiming Zhang. CF-DAN: Facial-expression recognition based on cross-fusion dual-attention network. *Computational Visual Media*, 10(3):593–608, 2024b.

Fan Zhang, Shiming Fan, and Hua Wang. Rethinking activation function: A simple path to efficient and accurate time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 2026b.

Fan Zhang, Shiming Fan, and Hua Wang. TimeSAF: Towards LLM-guided semantic asynchronous fusion for time series forecasting. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 26267–26284, 2026c.

Fan Zhang, Zhiwei Gu, and Hua Wang. Decoding with structured awareness: integrating directional, frequency-spatial, and structural attention for medical image segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 12421–12429, 2026d.

Jiayu Zhang, Chuangxin Zhao, Canran Xiao, Ruibo Duan, Wenyi Mo, Haoyu Gao, and Wenshuo Wang. Pi-CCA: Prompt-invariant CCA certificates for replay-free continual multimodal learning. In *Proceedings of the 14th International Conference on Learning Representations*, 2026e.

Xi Zhang, Feifei Zhang, and Changsheng Xu. Vqacl: A novel visual question answering continual learning setting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19102–19112, 2023b.

Chuangxin Zhao, Canran Xiao, Siyuan Ma, Mengyao Lyu, Yanbiao Ma, Jun Xia, Guiguang Ding, and Yang Liu. Attention-spectrum regularization for replay-free continual multimodal LLMs. *arXiv preprint arXiv:2606.23063*, 2026.

Hongbo Zhao, Fei Zhu, Haiyang Guo, Meng Wang, Rundong Wang, Gaofeng Meng, and Zhaoxiang Zhang. Mllm-cl: Continual learning for multimodal large language models. *arXiv preprint arXiv:2506.05453*, 2025.

Junhao Zheng, Qianli Ma, Zhen Liu, Binquan Wu, and Huawen Feng. Beyond anti-forgetting: Multimodal continual instruction tuning with positive forward transfer. *arXiv preprint arXiv:2401.09181*, 2024.

Zangwei Zheng, Mingyuan Ma, Kai Wang, Ziheng Qin, Xiangyu Yue, and Yang You. Preventing zero-shot transfer degradation in continual learning of vision-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19125–19136, 2023.

Zihao Zheng, Baichuan Li, Junyi Yao, and Jiayu Long. Reliable financial named entity recognition under domain shift: Confidence estimation and selective prediction. arXiv preprint arXiv:2608.19558, 2026a.

Zihao Zheng, Junyi Yao, Baichuan Li, and Jiayu Long. Auditing escalation reliability in healthcare LLM agents: A perturbation-controlled benchmark of workflow-state failures. arXiv preprint arXiv:2610.21270, 2026b.

Heng Zhou, Jing Tang, Juheng Zhang, Yanshu Li, Canran Xiao, Liwei Hou, Zong Ke, and Jiawei Yao. CoMem: Compositional concept-graph memory for vision–language adaptation. In *Proceedings of the 14th International Conference on Learning Representations*, 2026.

A Additional Method Details

A.1 Visual-Necessity Scoring Details

Token-level scoring. For an answer sequence $a_i^t = (a_{i,1}^t, \ldots, a_{i,L_i}^t)$, the target-conditioned loss used by the scoring model is

$$\ell_\theta(a_i^t \mid x, q) = -\frac{1}{L_i} \sum_{\ell=1}^{L_i} \log p_\theta \left(a_{i,\ell}^t \mid a_{i,<\ell}^t, x, q\right). \tag{11}$$

This loss is computed by teacher forcing on the supervised target tokens, so the visual-necessity score measures the image's effect on the likelihood of the provided answer. Free-form response matching is excluded from this score.

Null-image construction. The main experiments use a fixed mean-color blank image as the null input $\varnothing$. It is passed through the same visual encoder and visual projector as a real image. This keeps the sequence shape unchanged and removes visual evidence without changing the model interface.

Scoring snapshot and gate construction. At the start of each task t, we freeze the scoring snapshot:

$$\bar{\theta}_t = \mathrm{sg}(\theta_{t-1}). \tag{12}$$

The snapshot stays fixed across all mini-batches and epochs of the task (Algorithm 1). Following Eq. (6), we use the raw token-averaged loss gap $s_i^t = \ell_i^0 - \ell_i^+$ directly: subtract ρ, divide by τ_ν, clip to $[0, 1]$, and detach. The snapshot can retain biases from earlier tasks.

Evolution of gate assignments. A fixed example's gate stays unchanged within each task, including an erroneous zero gate. Nonzero gates and related high-necessity examples can still update the shared visual path. At the next task boundary, the snapshot is rebuilt from θ_t: improved grounding can increase the loss gap, while stronger language-only prediction can reduce it. Appendix C.3 examines recovery from biased initialization and the visual warm-up and gate-floor safeguards.

Mismatched-image robustness variant. For robustness analysis, we replace the null-image loss in Eq. (6) with a mixture of null-image and mismatched-image losses:

$$\ell_i^{\mathrm{cf}} = (1-\beta)\ell_{\bar{\theta}_t}(a_i^t \mid \varnothing, q_i^t) + \frac{\beta}{K} \sum_{k=1}^{K} \ell_{\bar{\theta}_t}(a_i^t \mid \tilde{x}_{i,k}^t, q_i^t), \tag{13}$$

where $\tilde{x}_{i,k}^t$ is sampled from other examples in the same mini-batch and $\beta \in [0, 1]$. The corresponding score is

$$s_i^{t,\mathrm{cf}} = \ell_i^{\mathrm{cf}} - \ell_{\bar{\theta}_t}(a_i^t \mid x_i^t, q_i^t), \qquad \nu_i^{t,\mathrm{cf}} = \mathrm{sg}\left[\mathrm{clip}_{[0,1]}\left(\frac{s_i^{t,\mathrm{cf}} - \rho}{\tau_\nu}\right)\right]. \tag{14}$$

The main results use the null-image score. The mismatched-image variant is reported as a robustness check.

A.2 Calibration Module

Architecture. Let $V_i = E_{\mathrm{v}}(x_i^t)$ be the frozen visual-encoder output and $Z_i = B_\psi(V_i) \in \mathbb{R}^{N_i \times d}$ be the projected visual tokens. Let $\bar{u}_i \in \mathbb{R}^d$ be the mean-pooled instruction representation from the frozen language embedding layer. The calibration module produces a low-rank residual

$$\Delta_\chi(Z_i, \bar{u}_i) = \alpha_\chi \left(Z_i A_c B_c\right) \odot \sigma\left(\bar{u}_i A_g B_g\right), \tag{15}$$

where $A_c, A_g \in \mathbb{R}^{d \times r_\chi}$, $B_c, B_g \in \mathbb{R}^{r_\chi \times d}$, $r_\chi \ll d$, α_χ is a residual scale, and $\odot$ denotes channel-wise multiplication with broadcasting over visual tokens. The calibrated visual tokens are

$$\widetilde{Z}_i = Z_i + \Delta_\chi(Z_i, \bar{u}_i). \tag{16}$$

The same calibrator is used for true-image and null-image forward passes. The invariance loss in Eq. (7) updates only χ through the routed gradient G_χ. The four trainable matrices contain $N_\chi = 4dr_\chi$ parameters, giving $131{,}072 \approx 0.131$M for $d = 4096$ and $r_\chi = 8$ in LLaVA-v1.5-7B.

Role of the calibration path. The calibration path is smaller than the visual path. It adjusts token scale and instruction-conditioned sensitivity, while the main visual projector and cross-modal LoRA layers remain protected by the visual-necessity gate. This design gives low-necessity examples a controlled training destination without allowing them to reshape the visual interface.

The additive form preserves the original token representation at zero residual initialization and keeps the visual-token topology unchanged. The low-rank, instruction-conditioned correction provides an architectural inductive bias. Table 15 compares its contribution with equal-capacity alternatives.

A.3 Visual-Path Contamination Diagnostics

Contaminating gradient mass. To measure how much visual-path update comes from low-necessity examples under standard tuning, we compute

$$\mathcal{M}_{\mathrm{contam}}(\mathcal{B}) = \frac{\sum_{i\in\mathcal{B}}(1-\nu_i^t)\,\|\nabla_\psi \mathcal{L}_i(\theta)\|_2^2}{\sum_{i\in\mathcal{B}}\|\nabla_\psi \mathcal{L}_i(\theta)\|_2^2+\epsilon}. \tag{17}$$

A large value indicates that examples judged visually unnecessary occupy a large fraction of the visual-path gradient energy.

Alignment with visual grounding anchors. We also measure whether an example's visual-path gradient agrees with gradients from visually grounded anchor examples. Let $\mathcal{A}$ be a small held-out anchor set whose answers require visual evidence, and define

$$\bar{g}_{\mathcal{A}}^{\psi} = \frac{1}{|\mathcal{A}|}\sum_{j\in\mathcal{A}}\nabla_\psi \mathcal{L}_j(\theta). \tag{18}$$

For a training example i, its visual-grounding alignment is

$$\alpha_i^{\psi} = \frac{\left\langle \nabla_\psi \mathcal{L}_i(\theta), \bar{g}_{\mathcal{A}}^{\psi}\right\rangle}{\|\nabla_\psi \mathcal{L}_i(\theta)\|_2\left\|\bar{g}_{\mathcal{A}}^{\psi}\right\|_2+\epsilon}. \tag{19}$$

This diagnostic tests whether low-necessity examples provide visual-path gradients aligned with grounding-preserving updates.

A.4 Comparison with Data Filtering and Scalar Weighting

Data filtering. A filtering method uses a binary mask $m_i \in \{0,1\}$ and updates all paths with

$$\Delta\theta_i^{\mathrm{filter}} = -\eta m_i \nabla_\theta \mathcal{L}_i(\theta). \tag{20}$$

When $m_i = 0$, filtering removes the example's language supervision, including information about instruction style, answer format, and domain language.

Scalar sample weighting. A scalar weighting method uses a single coefficient w_i:

$$\Delta\theta_i^{\mathrm{scalar}} = -\eta w_i \nabla_\theta \mathcal{L}_i(\theta). \tag{21}$$

This suppresses the visual and language paths simultaneously. VNG-CT applies a vector-valued path weight,

$$\Delta\omega_i = -\eta_\omega \nabla_\omega \mathcal{L}_i, \qquad \Delta\psi_i = -\eta_\psi \nu_i^t \nabla_\psi \mathcal{L}_i, \qquad \Delta\chi_i = -\eta_\chi (1-\nu_i^t)\nabla_\chi\left[\mathcal{L}_i + \lambda_{\mathrm{inv}}\mathcal{R}_i^{\mathrm{inv}}\right]. \tag{22}$$

Thus, low-necessity examples remain active for language learning while being suppressed for visual-path adaptation.

A.5 Full Training Algorithm

Algorithm 1 summarizes training. At the start of each task, we freeze a scoring snapshot from the previous task. For each mini-batch, this detached snapshot computes the true-image and null-image target losses and maps their gap to ν_i^t. The current model computes the supervised loss and calibration invariance regularizer. All examples update the language path; the visual and calibration paths receive gradients weighted by ν_i^t and $1-\nu_i^t$, respectively. The parameters learned after task t initialize the next task.

Algorithm 1 Visual-Necessity-Gated Continual Tuning

Require: Continual datasets $\mathcal{D}_{1:T}$; initial parameters $\theta_0 = (\psi_0, \omega_0, \chi_0)$; margin ρ; scale τ_ν; regularization weight λ_{inv}.

for $t = 1, \dots, T$ **do**
  Set scoring snapshot $\bar{\theta}_t \leftarrow \text{sg}(\theta_{t-1})$; keep it fixed throughout task t.
  for mini-batch $\mathcal{B}_t \subset \mathcal{D}_t$ **do**
    for each $(x_i^t, q_i^t, a_i^t) \in \mathcal{B}_t$ **do**
      Compute true-image loss $\ell_i^+ = \ell_{\bar{\theta}_t}(a_i^t \mid x_i^t, q_i^t)$.
      Compute null-image loss $\ell_i^0 = \ell_{\bar{\theta}_t}(a_i^t \mid \varnothing, q_i^t)$.
      Compute $s_i^t = \ell_i^0 - \ell_i^+$.
      Compute $\nu_i^t = \text{sg}[\text{clip}_{[0,1]}((s_i^t - \rho)/\tau_\nu)]$.
      Compute supervised loss $\mathcal{L}_i(\theta)$ using the trainable model.
      Compute invariance regularizer $\mathcal{R}_i^{\text{inv}}$ by Eq. (7).
    end for
    Compute routed gradients G_ω, G_ψ, G_χ by Eq. (9).
    Update ω, ψ, χ by Eq. (10).
  end for
  Set $\theta_t \leftarrow (\psi, \omega, \chi)$.
end for
return Final model θ_T.

A.6 Implementation Notes

Parameter groups. The visual path ψ contains the visual projector and the cross-modal LoRA layers. The language path ω contains text-side LoRA layers. The calibration path χ contains the low-rank residual module in Eq. (15). These disjoint parameter sets are registered as separate optimizer parameter groups to specify their learning rates and other optimizer options. Sample-dependent routing is implemented explicitly during gradient computation.

Routed gradient computation. Let `L`, `R_inv`, and `nu` be vectors of length $|\mathcal{B}_t|$ containing the per-example supervised losses, invariance penalties, and necessity scores, respectively. The losses and penalties are token-normalized as in Eqs. (5) and (7); the null-image reference distribution in `R_inv` is detached. The reusable parameter tuples `params_lang`, `params_vis`, and `params_cal` correspond to ω, ψ, and χ. We construct three scalar losses and differentiate each with respect to its designated parameter tuple:

```
optimizer.zero_grad(set_to_none=True)
L_lang = L.mean()
L_vis = (nu.detach() * L).mean()
L_cal = ((1 - nu.detach()) * (L + lambda_inv * R_inv)).mean()

g_lang = torch.autograd.grad(
    L_lang, params_lang, retain_graph=True)
g_vis = torch.autograd.grad(
    L_vis, params_vis, retain_graph=True)
g_cal = torch.autograd.grad(L_cal, params_cal)

for params, grads in ((params_lang, g_lang),
                      (params_vis, g_vis),
                      (params_cal, g_cal)):
    for param, grad in zip(params, grads):
        param.grad = grad.detach()
optimizer.step()
```

Detaching `nu` makes these gradients exactly G_ω, G_ψ, and G_χ in Eq. (9), with every mean normalized by the mini-batch size. The invariance term contributes only to G_χ. The first two calls retain the shared computation graph for the remaining gradient evaluations. We assign the returned gradients to the corresponding `.grad` fields before the optimizer updates the parameters. The code shows one mini-batch update; with gradient accumulation, each micro-batch gradient is scaled by its fraction of the effective batch and accumulated before a single optimizer step.

Training cost. The necessity score requires one additional null-image forward pass through the frozen scoring snapshot. Since the score is detached and uses the same target tokens as supervised tuning, it does not introduce extra annotation. The null-image forward pass is batched with the true-image pass in implementation.

Inference cost. At inference time, the model uses the full image and instruction. The scoring snapshot, null-image pass, and gradient gates are not used. The inference cost is therefore the same as the adapted MLLM with the trained visual, language, and calibration paths.

B Additional Experimental Details

B.1 Details of the Preliminary Diagnostic Study

The preliminary study in Section 3 measures visual-path updates from examples whose answers are largely insensitive to visual evidence. It uses ordinary multimodal continual tuning without a visual-necessity gate, gradient routing, data filtering, or calibration module.

Diagnostic setting. For each benchmark stream, we use the same task order, training split, and evaluation split as in the main experiments. Let $\mathcal{D}_{1:T} = \{\mathcal{D}_1, \ldots, \mathcal{D}_T\}$ denote the continual stream, where each example is an image–instruction–answer triplet (x_i^t, q_i^t, a_i^t). Before training on task t, we keep a frozen pre-task snapshot $\theta_{t-1}^{\mathrm{std}}$ of the model trained by standard continual tuning on previous tasks. All sensitivity scores and diagnostic gradients for task t are computed using this frozen snapshot before any update on $\mathcal{D}_t$. This prevents the diagnostic scores from being affected by the same examples that they are used to analyze.

The baseline update used in the preliminary study is ordinary supervised continual tuning:

$$\theta_t^{\mathrm{std}} = \mathrm{Update}\left(\theta_{t-1}^{\mathrm{std}}, \mathcal{D}_t, \nabla_{\psi,\omega} \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathcal{L}_{\mathrm{sft}}(x_i^t, q_i^t, a_i^t)\right), \tag{23}$$

where ψ denotes the visual projector and cross-modal adaptation layers, and ω denotes the language-side adaptation layers. All image-present examples update both paths in Eq. (23). The calibration path is absent from this diagnostic experiment.

Image perturbations. To estimate whether an example's answer depends on the image, we compare the model output under the original image with its output under several visual counterfactuals. For each image x_i^t, we construct

$$\mathcal{A}(x_i^t) = \left\{\varnothing,\ \mathrm{Shuffle}(x_i^t),\ \mathrm{Mismatch}(x_i^t)\right\}. \tag{24}$$

Here, $\varnothing$ is a neutral null image after the same image normalization pipeline as real inputs. $\mathrm{Shuffle}(x_i^t)$ randomly permutes non-overlapping patches while preserving low-level color and texture statistics. Unless otherwise specified, the patch size is set to the patch size of the vision encoder. $\mathrm{Mismatch}(x_i^t)$ replaces x_i^t with an image sampled from another example in the same current task. When answer categories are available, we avoid sampling an image from an example with the same answer to reduce accidental semantic overlap. In practice, we average the mismatched-image result over three random mismatches and keep the same random seed across model variants.

Output-level image sensitivity. For each example, we decode the model response with greedy decoding under the true image and under each perturbed image:

$$\hat{a}_i^+ = \mathrm{Decode}_{\theta_{t-1}^{\mathrm{std}}}(x_i^t, q_i^t), \qquad \hat{a}_{i,\tilde{x}} = \mathrm{Decode}_{\theta_{t-1}^{\mathrm{std}}}(\tilde{x}, q_i^t), \quad \tilde{x} \in \mathcal{A}(x_i^t). \tag{25}$$

We then compute the output-level image sensitivity score

$$d_i^{\mathrm{out}} = 1 - \max_{\tilde{x} \in \mathcal{A}(x_i^t)} \mathrm{Sim}\left(\hat{a}_i^+, \hat{a}_{i,\tilde{x}}\right). \tag{26}$$

A small value of d_i^{out} means that at least one strong visual perturbation leaves the model response nearly unchanged, suggesting low output-level visual sensitivity. Taking the maximum similarity

over perturbations gives a conservative estimate of visual irrelevance based on response preservation after image removal or corruption.

The similarity function combines lexical and semantic agreement:

$$\mathrm{Sim}(u, v) = \frac{1}{2}\,\mathrm{F1}_{\mathrm{tok}}(u, v) + \frac{1}{2}\max\{0,\, \cos\left(e(u), e(v)\right)\}\,, \tag{27}$$

where $\mathrm{F1}_{\mathrm{tok}}$ is normalized token-level F1 and $e(\cdot)$ is a frozen sentence encoder. For short-answer benchmarks, we additionally normalize answers by lowercasing and removing punctuation before computing token overlap. We do not use ground-truth visual-relevance labels.

Visual-path gradient magnitude. We measure the visual-path gradient norm induced by each example to quantify the effect of visually insensitive supervision under standard tuning. For a frozen pre-task snapshot $\theta_{t-1}^{\mathrm{std}}$, we compute

$$g_{i,\psi}^{t} = \nabla_{\psi}\mathcal{L}_{\mathrm{sft}}\left(\theta_{t-1}^{\mathrm{std}}; x_i^t, q_i^t, a_i^t\right), \qquad m_i^{\mathrm{vis}} = \|g_{i,\psi}^{t}\|_2. \tag{28}$$

These gradients are used only for measurement; no optimizer step is applied when computing m_i^{vis}. In Figure 2(a), each point corresponds to one training example, the x-axis is d_i^{out}, and the y-axis is m_i^{vis}. The shaded region marks examples with very low output-level image sensitivity.

Gradient compatibility analysis. We further examine whether gradients from visually insensitive examples are aligned with gradients from visually sensitive examples. For each mini-batch $\mathcal{B}_b$, we split examples into low- and high-sensitivity groups using within-batch quantiles:

$$\begin{aligned} \mathcal{B}_b^{\mathrm{low}} &= \left\{ i \in \mathcal{B}_b : d_i^{\mathrm{out}} \le Q_{\alpha_q}\left(\{d_j^{\mathrm{out}} : j \in \mathcal{B}_b\}\right)\right\}, \\ \mathcal{B}_b^{\mathrm{high}} &= \left\{ i \in \mathcal{B}_b : d_i^{\mathrm{out}} \ge Q_{1-\alpha_q}\left(\{d_j^{\mathrm{out}} : j \in \mathcal{B}_b\}\right)\right\}, \end{aligned} \tag{29}$$

where $\alpha_q = 0.3$ in our experiments. We then compute the mean visual-path gradient of each group:

$$\bar{g}_{b,\psi}^{\mathrm{low}} = \frac{1}{|\mathcal{B}_b^{\mathrm{low}}|}\sum_{i \in \mathcal{B}_b^{\mathrm{low}}} g_{i,\psi}^{t}, \qquad \bar{g}_{b,\psi}^{\mathrm{high}} = \frac{1}{|\mathcal{B}_b^{\mathrm{high}}|}\sum_{i \in \mathcal{B}_b^{\mathrm{high}}} g_{i,\psi}^{t}. \tag{30}$$

The low–high gradient compatibility is

$$\kappa_b^{\mathrm{low,high}} = \cos\left(\bar{g}_{b,\psi}^{\mathrm{low}}, \bar{g}_{b,\psi}^{\mathrm{high}}\right). \tag{31}$$

As a reference, we also compute high–high compatibility by randomly splitting $\mathcal{B}_b^{\mathrm{high}}$ into two equally sized subsets and measuring the cosine between their mean gradients:

$$\kappa_b^{\mathrm{high,high}} = \cos\left(\bar{g}_{b,\psi}^{\mathrm{high},1}, \bar{g}_{b,\psi}^{\mathrm{high},2}\right). \tag{32}$$

We repeat this random split three times and report the average. This reference controls for the fact that gradient cosine can be noisy in high-dimensional parameter spaces. In Figure 2(b), each point denotes one mini-batch, and the curves report the mean compatibility at each continual task index.

Stream-level drift and visual forgetting. Finally, we test whether the prevalence of visually insensitive examples in a task predicts visual-path drift and old-task visual forgetting under standard continual tuning. For task t, the low-sensitivity mass is

$$r_t^{\mathrm{low}} = \frac{1}{|\mathcal{D}_t|}\sum_{i=1}^{|\mathcal{D}_t|} \mathbb{1}\left[d_i^{\mathrm{out}} \le \tau_{\mathrm{out}}\right], \tag{33}$$

where $\tau_{\mathrm{out}} = 0.1$ unless otherwise specified. We measure the visual path drift after training on task t as

$$\Delta_t^{\psi} = \frac{\|\psi_t^{\mathrm{std}} - \psi_{t-1}^{\mathrm{std}}\|_2}{\|\psi_{t-1}^{\mathrm{std}}\|_2 + \epsilon}, \tag{34}$$

with $\epsilon = 10^{-12}$ for numerical stability.

Table 5: Summary of benchmark usage. "Official" means that we follow the metric and evaluation script released by the corresponding benchmark.

Benchmark	Continual setting	Main coverage	Primary metrics
CoIN [Chen et al., 2024a]	Task-incremental MCIT	VQA, OCR-VQA, grounding, classification, reasoning	Official task scores, MAA, BWT
MLLM-CL [Zhao et al., 2025]	Domain / ability continual learning	Domain shift and ability shift	Last, Average, forgetting
UCIT-style streams [Guo et al., 2025a]	Mixed instruction continual learning	Captioning, VQA, reasoning, classification, math	Official task scores, FAA, CAA, FM

To isolate forgetting of visual grounding, we evaluate old tasks on their visually sensitive validation subsets. Specifically, for each previous task $s < t$, we define $\mathcal{V}_s^{\mathrm{vis}}$ as the subset of validation examples whose output-level sensitivity under the pre-task model is in the top quantile. Let

$$G_s(\theta) = \mathrm{Score}\left(\theta; \mathcal{V}_s^{\mathrm{vis}}\right) \tag{35}$$

be the visual-grounding score, implemented as task-specific accuracy or normalized answer score depending on the benchmark. The old-task visual forgetting after task t is

$$F_t^{\mathrm{vis}} = \frac{100}{t-1} \sum_{s=1}^{t-1} \left[G_s(\theta_s^{\mathrm{std}}) - G_s(\theta_t^{\mathrm{std}}) \right]. \tag{36}$$

Thus, F_t^{vis} measures the percentage-point drop in visual-grounding performance on previous tasks after standard tuning proceeds to later tasks. In Figure 2(c), each point corresponds to a dataset–seed–task triple, the x-axis is r_t^{low}, the y-axis is F_t^{vis}, and the marker size represents Δ_t^{ψ}. The regression line is fit by ordinary least squares, and the reported Pearson correlation r_p is computed over all plotted triples.

B.2 Benchmark Details

CoIN. CoIN [Chen et al., 2024a] is our primary benchmark for continual instruction tuning of MLLMs. It contains a heterogeneous stream of multimodal instruction-following tasks, including visual question answering, OCR-oriented VQA, image classification, grounding, and multimodal reasoning. We follow the official task order, train split, evaluation split, and task-specific scoring scripts. This benchmark is particularly suitable for our study because its tasks contain different proportions of visually grounded, partially grounded, and language-dominant image-present examples.

MLLM-CL. MLLM-CL [Zhao et al., 2025] is used to evaluate whether VNG-CT generalizes across broader continual learning regimes. We use both domain continual learning and ability continual learning. The domain-incremental setting evaluates adaptation across visually and semantically distinct domains, while the ability-incremental setting evaluates sequential acquisition of capabilities such as OCR, visual perception, multimodal reasoning, and GUI-style understanding. This benchmark tests whether visual-necessity gating remains effective when the source of distribution shift changes from task identity to domain or ability identity.

UCIT-style mixed instruction streams. We additionally evaluate on UCIT-style mixed instruction streams adopted by recent MCIT studies [Guo et al., 2025a]. This setting mixes visually diverse tasks such as image classification, captioning, visual reasoning, multimodal math, and open-ended VQA. Compared with CoIN, this protocol contains stronger heterogeneity in answer format and visual-dependence level. We use it to test whether VNG-CT remains stable when task boundaries are less aligned with modality usage.

B.3 Continual Learning Metrics

Let T be the number of tasks and $A_{t,j}$ denote the performance on task j after training on task t. For metrics where higher is better, the final average accuracy is

$$\mathrm{FAA} = \frac{1}{T} \sum_{j=1}^{T} A_{T,j}. \tag{37}$$

Following CoIN [Chen et al., 2024a], mean average accuracy over the continual learning trajectory is

$$\mathrm{MAA} = \frac{1}{T} \sum_{t=1}^{T} \left(\frac{1}{t} \sum_{j=1}^{t} A_{t,j} \right). \tag{38}$$

Backward transfer is

$$\mathrm{BWT} = \frac{1}{T}\sum_{j=1}^{T}\left(A_{T,j} - A_{j,j}\right). \tag{39}$$

The forgetting measure is

$$\mathrm{FM} = \frac{1}{T-1}\sum_{j=1}^{T-1}\left[\max_{t\in\{j,\ldots,T-1\}} A_{t,j} - A_{T,j}\right]. \tag{40}$$

We additionally report current average accuracy (CAA) on UCIT-style streams:

$$\mathrm{CAA} = \frac{1}{T}\sum_{t=1}^{T} A_{t,t}. \tag{41}$$

CAA measures the average performance on each task immediately after that task is learned, and therefore reflects new-task adaptation or plasticity. It complements FAA and FM: FAA measures the final retained performance after the whole stream, while FM measures the amount of forgetting.

B.4 Visual-Path Metrics

Standard continual learning metrics measure task-level forgetting. We use the following additional metrics to assess visual-path preservation.

Visual forgetting rate. For each task j, we construct a visually grounded evaluation subset $\mathcal{V}_j$. An example belongs to $\mathcal{V}_j$ if replacing or removing the image substantially changes the reference model's answer score. Let $A^{\mathrm{vis}}_{t,j}$ denote the score on this subset after training task t. The visual forgetting rate is

$$\mathrm{VFR} = \frac{1}{T-1}\sum_{j=1}^{T-1}\left[\max_{t\in\{j,\ldots,T-1\}} A^{\mathrm{vis}}_{t,j} - A^{\mathrm{vis}}_{T,j}\right]. \tag{42}$$

Lower VFR indicates better preservation of old visual grounding.

Counterfactual visual sensitivity. For an evaluation example (x_i, q_i, a_i), let $m(\hat{a}, a_i)$ be the task-specific answer score. We define

$$\mathrm{VCS}_i = m(f_\theta(x_i, q_i), a_i) - \max_{\tilde{x}\in\{\varnothing,\mathrm{shuffle}(x_i),x_{r(i)}\}} m(f_\theta(\tilde{x}, q_i), a_i), \tag{43}$$

where $\varnothing$ is the null image, $\mathrm{shuffle}(x_i)$ is a patch-shuffled image, and $x_{r(i)}$ is a randomly mismatched image from the same task. Higher VCS indicates that the model relies more appropriately on the correct visual evidence. We report average VCS on visually grounded subsets and also provide per-task VCS in the diagnostic tables.

Visual-necessity ranking quality. We distinguish the ranking induced by visual-path routing from the final model's counterfactual separability. A fixed reference checkpoint θ_{ref}, independent of each evaluated method's routing and trained checkpoints, provides labels shared across methods under the same benchmark and backbone setting. For each held-out example, we define

$$y_i^{\mathrm{vis}} = \mathbb{1}\left[m(f_{\theta_{\mathrm{ref}}}(x_i, q_i), a_i) - m(f_{\theta_{\mathrm{ref}}}(\varnothing, q_i), a_i) > \delta\right], \tag{44}$$

where δ is a small validation threshold. Let $w_i^{\mathrm{vis,M}}$ denote the actual scalar weight applied to example i's visual-path gradient by method M. It equals ν_i for the default VNG-CT, global weighting, and no-calibration variants; the binary or permuted gate for hard or random gating; and one for methods without sample-wise visual reweighting.

Routing and post-hoc VNR. For a trained method M, let θ_T^M denote its final checkpoint after the continual stream. We freeze this checkpoint and compute a method-specific counterfactual separability score

$$r_i^M = \ell_{\mathrm{sg}(\theta_T^M)}(a_i \mid \varnothing, q_i) - \ell_{\mathrm{sg}(\theta_T^M)}(a_i \mid x_i, q_i), \tag{45}$$

where ℓ is the token-averaged target-answer NLL. On the held-out diagnostic set $\mathcal{E}_{\text{diag}}$, the two ranking metrics are

$$\text{VNR}_{\text{route}}(M) = \text{AUROC}\left(\left\{(w_i^{\text{vis,M}}, y_i^{\text{vis}})\right\}_{i\in\mathcal{E}_{\text{diag}}}\right), \tag{46a}$$

$$\text{VNR}_{\text{post}}(M) = \text{AUROC}\left(\left\{(r_i^M, y_i^{\text{vis}})\right\}_{i\in\mathcal{E}_{\text{diag}}}\right). \tag{46b}$$

Table 3 reports $\text{VNR}_{\text{route}}$, and Table 7 reports VNR_{post}, both on a percentage scale. The no-gate variant has constant routing weights and hence $\text{VNR}_{\text{route}} = 50\%$. AUPRC uses the corresponding score–label pairs when reported. For all methods, including VNG-CT, VNR_{post} evaluates the frozen final checkpoint. Higher values indicate greater separation between visually dependent and language-dominant image-present examples under image removal.

Low-necessity visual leakage. We assess low necessity with a reference score $\hat{\nu}_i^{\text{ref}}$ computed once from θ_{ref} and shared across methods. Applying the mapping in Eq. (6) gives

$$s_i^{\text{ref}} = \ell_{\theta_{\text{ref}}}(a_i \mid \varnothing, q_i) - \ell_{\theta_{\text{ref}}}(a_i \mid x_i, q_i),$$
$$\hat{\nu}_i^{\text{ref}} = \text{clip}\left(\frac{s_i^{\text{ref}} - \rho}{\tau_\nu}, 0, 1\right).$$

Let $g_i^{\psi,M} = \nabla_{\psi^M}\mathcal{L}_i^M$ be the per-example visual-path gradient before applying the routing weight. Averaging over diagnostic mini-batches $\mathcal{B}$, we define

$$\text{LNVL}(M) = \mathbb{E}_{\mathcal{B}}\left[\frac{\left\|\sum_{i\in\mathcal{B}}(1-\hat{\nu}_i^{\text{ref}})\, w_i^{\text{vis,M}}\, g_i^{\psi,M}\right\|_2}{\left\|\sum_{i\in\mathcal{B}} w_i^{\text{vis,M}}\, g_i^{\psi,M}\right\|_2 + \epsilon}\right]. \tag{47}$$

Lower LNVL indicates a smaller low-necessity gradient contribution relative to the actual routed visual-path update. The fixed reference score identifies low-necessity examples independently of the evaluated method's routing weights.

Visual-path drift. Let $h_{\psi_t}(x)$ denote the visual representation after task t, and let $\mathcal{A}$ be a fixed held-out anchor set. We define

$$\text{VPD}_t = \frac{1}{|\mathcal{A}|}\sum_{x\in\mathcal{A}}\frac{\|h_{\psi_t}(x) - h_{\psi_0}(x)\|_2}{\|h_{\psi_0}(x)\|_2 + \epsilon}. \tag{48}$$

This metric measures accumulated drift of the visual interface from the initial pretrained model; at initialization, $\text{VPD}_0 = 0$.

Scope of gate-related diagnostics. LNVL combines independent reference necessity scores with actual visual routing weights. $\text{VNR}_{\text{route}}$ evaluates these weights, while VNR_{post} evaluates final-checkpoint counterfactual scores. Both use fixed reference labels shared across methods; VFR uses reference-defined visual subsets. Figure 4(a) checks internal consistency between two image-removal-based quantities. Appendix C.2 complements these diagnostics with human-label, pixel-versus-prior, and real-image measurements independent of the gate's null-image labels.

B.5 Comparison Protocol

Baseline groups. We compare the following groups of methods in Tables 1 and 2.

- **Standard continual learning baselines:** EWC [Kirkpatrick et al., 2017] and LwF [Li and Hoiem, 2017].
- **Parameter-efficient tuning baselines:** sequential LoRA [Hu et al., 2022], O-LoRA [Wang et al., 2023], and L2P [Wang et al., 2022].
- **MLLM continual instruction-tuning methods:** Continual LLaVA [Cao et al., 2024], Fwd-Prompt [Zheng et al., 2024], ModalPrompt [Zeng et al., 2025a], CL-MoE [Huai et al., 2025], ProgLoRA [Yu et al., 2025], HiDe-LLaVA [Guo et al., 2025a], BranchLoRA [Zhang et al., 2025], SEFE [Chen et al., 2025], SMoLoRA [Wang et al., 2025a], Instruction-Grounded Visual Projectors [Jin et al., 2025], MR-LoRA [Zhao et al., 2025], LiLoRA [Che et al., 2026], and SAME [Xie et al., 2026].

B.6 Backbones and Training Details

Backbones. Our main backbone is LLaVA-1.5-7B [Liu et al., 2023]. We choose it because it is widely used in recent MLLM continual instruction-tuning studies and supports efficient LoRA-based adaptation. To test architecture robustness, we additionally evaluate InternVL-Chat-7B [Chen et al., 2024b] and Qwen2-VL [Wang et al., 2024].

Trainable parameters. For VNG-CT, we split trainable parameters into three paths: the language path ω, the visual path ψ, and the calibration path χ. The language path contains LoRA modules in text-side attention and MLP projections. The visual path contains the multimodal projector and cross-modal LoRA modules that directly affect visual-token integration. The calibration path is a small residual adapter placed after the visual projector. The vision encoder and base LLM weights are frozen in all main experiments.

Default hyperparameters. Unless otherwise specified, we use LoRA rank $r_{\mathrm{LoRA}} = 16$, LoRA scaling factor $\alpha_{\mathrm{LoRA}} = 32$, LoRA dropout 0.05, and no weight decay. The default learning rates are

$$\eta_\omega = 2 \times 10^{-4}, \qquad \eta_\psi = 1 \times 10^{-4}, \qquad \eta_\chi = 2 \times 10^{-4}. \tag{49}$$

We use AdamW with cosine learning-rate decay and a warmup ratio of 0.03. Training is performed with bfloat16 precision. The global batch size is 16 unless otherwise specified, with gradient accumulation used when necessary. All main results are averaged over three random seeds.

Visual-necessity scoring. For each training example in task t, the frozen snapshot $\bar{\theta}_t = \mathrm{sg}(\theta_{t-1})$ computes the token-averaged target losses under the original image and a fixed null image, both using the same image preprocessing. The snapshot is held fixed for the entire task. Following Eq. (6), the visual-necessity gate is

$$\nu_i = \mathrm{sg}\left[\mathrm{clip}_{[0,1]}\left(\frac{\ell_i^0 - \ell_i^+ - \rho}{\tau_\nu}\right)\right], \tag{50}$$

where ℓ_i^+ is the loss with the original image and ℓ_i^0 is the loss with the null image; task indices are omitted for brevity. We use the raw loss gap directly: subtract ρ, divide by τ_ν, clip to $[0, 1]$, and detach. We set $\rho = 0.02$ and $\tau_\nu = 0.10$ by default.

Calibration path. The calibration path follows the instruction-conditioned low-rank residual module defined in Appendix A.2. Let $V_i = E_v(x_i^t)$ denote the frozen visual-encoder output and $Z_i = B_\psi(V_i) \in \mathbb{R}^{N_i \times d}$ denote the projected visual tokens. Let $\bar{u}_i \in \mathbb{R}^d$ be the mean-pooled instruction representation from the frozen language embedding layer. The calibration module produces

$$\Delta_\chi(Z_i, \bar{u}_i) = \alpha_\chi \left(Z_i A_c B_c\right) \odot \sigma(\bar{u}_i A_g B_g), \tag{51}$$

where $A_c, A_g \in \mathbb{R}^{d \times r_\chi}$, $B_c, B_g \in \mathbb{R}^{r_\chi \times d}$, $r_\chi \ll d$, and $\odot$ denotes channel-wise multiplication with broadcasting over visual tokens. The calibrated visual tokens are

$$\widetilde{Z}_i = Z_i + \Delta_\chi(Z_i, \bar{u}_i). \tag{52}$$

The same calibration module is used for both the true-image and null-image forward passes. For the null-image view, Z_i is replaced by the projected null-image tokens, while the instruction representation $\bar{u}_i$ is kept unchanged. This design allows the calibration path to learn instruction-conditioned invariance to visually irrelevant evidence without modifying the main visual projector or the frozen vision encoder.

The learnable parameters of the calibration path are $\chi = \{A_c, B_c, A_g, B_g\}$. Unless otherwise specified, we set the calibration rank to $r_\chi = 8$ and use $\lambda_{\mathrm{inv}} = 0.1$ for the invariance regularizer. We tune $r_\chi \in \{4, 8, 16\}$ and $\lambda_{\mathrm{inv}} \in \{0.03, 0.1, 0.3\}$ in sensitivity analysis.

Hardware. Experiments are run on NVIDIA A100-80GB GPUs. We use DeepSpeed ZeRO optimization for memory efficiency. Table 6 reports peak memory per GPU, full-stream training time, throughput, training overhead, and inference latency change for sequential LoRA, SAME, and VNG-CT on LLaVA/UCIT.

B.7 Ablation Studies

We conduct a single-factor ablation study to isolate the contribution of the main design choices in VNG-CT. All ablations use LLaVA-v1.5-7B as the backbone and keep the same task order, training budget, optimizer, LoRA rank, and evaluation protocol as the full model. The results are reported in Table 3. For each variant, we modify only one component while keeping the remaining training pipeline unchanged.

Gate design. We first examine whether the visual-necessity gate is necessary and whether its value needs to be sample-specific. We compare the full soft gate with three variants:

- **No gate**: sets $\nu_i = 1$ for all image-present examples, so every example updates the visual path as in standard continual tuning.
- **Hard gate**: replaces the soft gate with a binary decision, $\nu_i = \mathbb{1}[\ell_i^0 - \ell_i^+ > \rho]$. This tests whether discretizing visual necessity is sufficient.
- **Random gate**: randomly permutes ν_i within each mini-batch while preserving the same gate-value distribution. This isolates the contribution of sample-specific visual-necessity estimates at a fixed distribution of visual-path update weights.

Path-specific routing. We compare VNG-CT with a scalar reweighting baseline:

- **Global weighting**: multiplies the entire supervised loss by the visual-necessity score ν_i, applying the same weight to the language, visual, and calibration paths.

This ablation compares scalar sample weighting with path-specific routing. Global weighting suppresses low-necessity examples for all trainable parameters, including the language path, and may discard useful language supervision.

Calibration path. We further test whether the calibration path is necessary:

- **No calibration**: removes the calibration path χ and its invariance regularization. Low-necessity examples continue to train the language path, with the dedicated invariance-learning path removed.

This ablation evaluates the contribution of a separate calibration path to visual stability when low-necessity visual updates are suppressed.

Summary. These ablations are designed to answer three questions: whether visual-necessity estimation is useful, whether soft and sample-specific gating is better than hard or random gating, and whether path-specific routing with calibration is more effective than scalar sample reweighting. The empirical results in Table 3 show that removing or randomizing the gate causes the largest degradation, global weighting underperforms path-specific routing, and removing calibration mainly harms visual-path stability.

B.8 Evaluation and Reporting

Main tables. For each benchmark, the main table reports FAA, MAA, BWT, and FM. For MLLM-CL, we additionally report the benchmark's official Last and Average metrics.

Diagnostic tables. The diagnostic tables report VFR, VCS, LNVL, and VPD, together with $\mathrm{VNR}_{\mathrm{route}}$ in Table 3 and $\mathrm{VNR}_{\mathrm{post}}$ in Table 7. These metrics assess the effect of VNG-CT on the visual pathway.

Statistical reporting. Our main evaluations use three random seeds; where the benchmark protocol allows multiple task orders, we additionally average over two task orders. Appendix C.6 specifies the controlled rerun subset of the published baselines, reports mean and standard deviation across three seeds, and gives paired 95% confidence intervals for differences from SAME. This subset uses an equal 12-configuration search budget.

Model selection. Hyperparameters are selected on the validation split of the first task sequence and fixed across all tasks and other sequences. Visual-path diagnostics are used only for analysis, independently of model selection.

B.9 Other experimental details

Gate-validity protocol. For each training example, we compute the visual-necessity gate ν_i using the frozen scoring snapshot $\bar{\theta}$. We then evaluate image dependence by replacing the original image with the null image and measuring the task-specific score drop:

$$\Delta_i^{\mathrm{abl}} = m(f_{\bar{\theta}}(x_i, q_i), a_i) - m(f_{\bar{\theta}}(\varnothing, q_i), a_i), \tag{53}$$

where $m(\cdot, \cdot)$ denotes the benchmark-specific answer score. A larger Δ_i^{abl} indicates stronger dependence on the correct image. To diagnose visual-path contamination, we group examples into Low-, Mid-, and High-ν bins according to their gate scores and measure both gradient magnitude and gradient direction. The visual-path gradient norm and anchor alignment are computed as

$$m_i^{\mathrm{vis}} = \left\|\nabla_\psi \mathcal{L}_i\right\|_2, \qquad c_i^{\mathrm{anchor}} = \cos\left(\nabla_\psi \mathcal{L}_i, \bar{g}_{\mathcal{A}}^{\psi}\right), \tag{54}$$

where $\bar{g}_{\mathcal{A}}^{\psi}$ is the average visual-path gradient of a held-out visually grounded anchor set $\mathcal{A}$. This protocol tests whether low-gate examples produce visual gradients that conflict with grounding-preserving updates.

Routing-vs.-weighting protocol. To compare path-specific routing with global sample weighting, we evaluate methods in a two-dimensional stability–plasticity plane. The stability axis is the visual forgetting rate (VFR), defined in Appendix B.4, where lower is better. The plasticity axis is new-task adaptation (NTA), defined as the average performance on each task immediately after learning it:

$$\mathrm{NTA} = \frac{1}{T}\sum_{t=1}^{T} A_{t,t}, \tag{55}$$

where $A_{t,t}$ denotes the score on task t right after training on task t. A higher NTA indicates stronger adaptation to incoming tasks. The global-weighting baseline applies the visual-necessity score as a scalar sample weight to the whole supervised loss:

$$\mathcal{L}_i^{\mathrm{global}} = \nu_i \mathcal{L}_i. \tag{56}$$

This suppresses low-necessity examples for all trainable paths, including the language path. VNG-CT uses path-specific routing: low-necessity examples continue to update the language path while their direct updates to the main visual path are restricted. In Figure 5, large markers denote averages over benchmark streams and random seeds, faint markers denote individual runs, and the dashed curve indicates the non-dominated Pareto frontier.

Controlled-stream protocol. To evaluate robustness under different levels of visual dependence, we construct controlled continual streams by varying the proportion of low-visual-necessity examples. For a task dataset $\mathcal{D}_t$, we define the low-necessity ratio as

$$r_{\mathrm{low}} = \frac{1}{|\mathcal{D}_t|}\sum_{i=1}^{|\mathcal{D}_t|} \mathbb{1}[\nu_i \le \tau_{\mathrm{gate}}], \tag{57}$$

where $\tau_{\mathrm{gate}} = 0.33$ by default. We create five stream variants with

$$r_{\mathrm{low}} \in \{0.15, 0.30, 0.45, 0.60, 0.75\}.$$

For each target ratio, we keep the total number of examples and the task order fixed, but resample examples within each task to match the desired low-necessity proportion. This yields high-, medium-, and low-visual-dependence regimes while controlling for stream length and task sequence. We evaluate final average accuracy (FAA), visual forgetting rate (VFR), and counterfactual visual sensitivity (VCS). All curves in Figure 13 report the mean over random seeds, and shaded regions denote one standard deviation.

B.10 Computational Resources

All experiments are conducted on NVIDIA A100-80GB GPUs using bfloat16 precision. For VNG-CT, the vision encoder and the base language model are frozen, and only LoRA modules, the visual projector, and the lightweight calibration path are updated. Each main continual-learning run uses 4 GPUs with DeepSpeed ZeRO optimization and gradient checkpointing. Depending on the benchmark and backbone, one full continual stream takes approximately 6–14 wall-clock hours, corresponding to 24–56 A100 GPU-hours.

The additional cost of VNG-CT mainly comes from visual-necessity scoring, which requires a counterfactual null-image forward pass during training. Since this pass is computed with a detached scoring snapshot and batched with the normal forward pass, the wall-clock overhead is about 25% compared with standard LoRA continual tuning. The gate, null-image pass, and scoring snapshot are not used at inference time; therefore, VNG-CT introduces no additional inference-time routing cost beyond the trained lightweight calibration module.

Measured training and inference cost. Table 6 compares sequential LoRA, SAME, and VNG-CT under the same LLaVA/UCIT setup on four A100-80GB GPUs. Relative to LoRA, VNG-CT uses an additional 4.6GB of peak memory per GPU and has 24.9% measured training-time overhead. Its inference routing overhead is zero, with a measured total latency increase of 0.6%. SAME has 7.9% inference routing overhead and an 8.4% total latency increase in this setup.

Table 6: **Measured resource cost on LLaVA/UCIT.** Training overhead and inference latency change are relative to sequential LoRA. Full-stream times are rounded to one decimal place; relative overheads are reported from the timing measurements.

Method	Peak memory (GB/GPU)	Stream time (hours)	Samples/s ↑	Training overhead	Inference latency change
Sequential LoRA	51.2	7.3	5.80	0%	0%
SAME	61.7	9.7	4.25	+31.6%	+8.4%
VNG-CT	55.8	9.1	4.65	+24.9%	+0.6%

The additional VNG-CT memory/time costs are 1.7GB/7.1% for the detached scoring snapshot, 2.4GB/15.9% for the null-image forward pass, and 0.5GB/1.9% for calibration. Counterfactual scoring uses forward passes only.

C Additional Experimental Results

C.1 Extended Visual-Path Stability Diagnostics

Visual-path diagnostic results. Table 7 provides a detailed visual-path diagnostic analysis on the held-out visually grounded subsets. These metrics examine preservation of the visual interface during continual tuning. We report visual forgetting rate (VFR), counterfactual visual sensitivity (VCS), post-hoc visual-necessity ranking quality ($\mathrm{VNR}_{\mathrm{post}}$), low-necessity visual leakage (LNVL), and visual-path drift (VPD). All methods use the same diagnostic protocol. Lower VFR, LNVL, and VPD indicate better visual-path stability, while higher VCS and $\mathrm{VNR}_{\mathrm{post}}$ indicate stronger visual grounding and greater counterfactual visual-dependence separability.

The diagnostic results in Table 7 show that VNG-CT substantially reduces visual forgetting and visual-path drift compared with prior parameter-isolation and expert-routing methods. The lower LNVL indicates that low-necessity examples contribute much less gradient leakage to the visual path, which directly supports the motivation of path-specific routing. The higher VCS suggests that VNG-CT preserves sensitivity to the correct image.

Finally, the higher $\mathrm{VNR}_{\mathrm{post}}$ indicates that the final trained model better separates visually dependent and language-dominant image-present examples through its counterfactual scores. Table 3 evaluates routing-weight ranking separately with $\mathrm{VNR}_{\mathrm{route}}$.

Visual representation drift across continual tasks. We further examine whether VNG-CT stabilizes the visual representation space during continual tuning. To measure changes in the visual

Table 7: **Visual-path diagnostic results.** VFR measures old-task visual forgetting; VCS measures sensitivity to the correct image; $\mathrm{VNR}_{\mathrm{post}}$ is the AUROC of final-checkpoint counterfactual scores; LNVL measures routed visual-gradient leakage using independent reference necessity scores; VPD measures visual-path representation drift.

Method	Backbone	VFR ↓	VCS ↑	$\mathrm{VNR}_{\mathrm{post}}$ ↑	LNVL ↓	VPD ↓
Same-backbone diagnostic comparison						
LoRA [Hu et al., 2022]	LLaVA-v1.5-7B	18.58	21.24	61.51	0.69	0.176
CL-MoE [Huai et al., 2025]	LLaVA-v1.5-7B	17.71	22.54	62.45	0.72	0.163
O-LoRA [Wang et al., 2023]	LLaVA-v1.5-7B	12.69	28.62	66.98	0.55	0.134
L2P [Wang et al., 2022]	LLaVA-v1.5-7B	13.35	27.72	65.77	0.63	0.129
ModalPrompt [Zeng et al., 2025a]	LLaVA-v1.5-7B	11.78	30.96	68.59	0.50	0.121
HiDe-LLaVA [Guo et al., 2025a]	LLaVA-v1.5-7B	9.34	34.58	71.09	0.44	0.085
MR-LoRA [Zhao et al., 2025]	LLaVA-v1.5-7B	**8.16**	**36.27**	**73.62**	**0.37**	**0.072**
SAME [Xie et al., 2026]	LLaVA-v1.5-7B	**5.48**	**41.66**	**78.15**	**0.31**	**0.058**
Ours						
VNG-CT	LLaVA-v1.5-7B	**3.36**	**45.89**	**84.27**	**0.15**	**0.021**
VNG-CT	InternVL-Chat-7B	3.09	47.09	84.98	0.09	0.019
VNG-CT	Qwen2-VL-7B	4.02	43.85	80.55	0.22	0.038

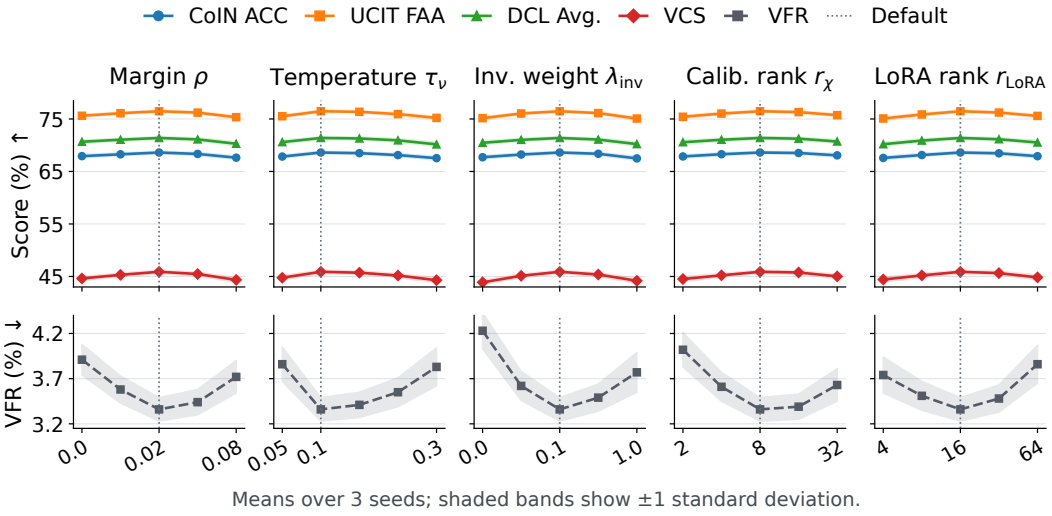

Figure 6: **Parameter sensitivity analysis of VNG-CT.** We vary five key hyperparameters, including the visual-necessity margin ρ, the gate temperature τ_ν, the invariance weight λ_{inv}, the calibration rank r_χ, and the LoRA rank r_{LoRA}. The vertical dotted line marks the default setting. Each column corresponds to one hyperparameter. The upper row reports CoIN ACC, UCIT FAA, DCL Avg., and VCS; the lower row reports VFR on a separate, shared scale. Overall, VNG-CT remains stable across a broad range of hyperparameter values, indicating strong robustness.

interface after each task, we track a fixed held-out visual anchor set $\mathcal{A}$ throughout the continual stream. For method m after task t, we measure visual representation drift as

$$\mathrm{VPD}_{m,t} = \frac{1}{|\mathcal{A}|} \sum_{x \in \mathcal{A}} \frac{\left\| h_{\psi_{m,t}}(x) - h_{\psi_0}(x) \right\|_2}{\left\| h_{\psi_0}(x) \right\|_2 + \epsilon}, \tag{58}$$

where $h_{\psi_{m,t}}(x)$ denotes the projected visual representation after training task t.

Figure 7 shows that standard LoRA rapidly moves the visual representation away from its initial state as continual tasks accumulate. SAME reduces this movement, with clear drift remaining over the task stream. VNG-CT maintains a much shorter and more compact trajectory, with consistently lower VPD at each task. This indicates that visual-necessity routing directly stabilizes the visual interface by preventing visually unnecessary supervision from accumulating in the visual path.

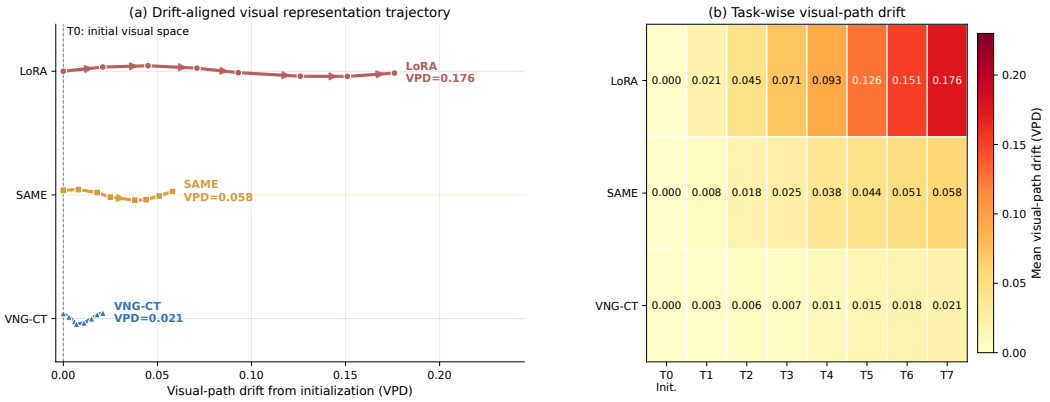

Figure 7: **Visual representation drift across continual tasks. (a)** Drift-aligned trajectory of visual representations, where the horizontal coordinate directly corresponds to the visual-path drift from initialization. LoRA accumulates the largest drift, SAME reduces the movement, and VNG-CT keeps the trajectory compact across the entire stream. **(b)** Task-wise visual-path drift heatmap measured in the original representation space. T0 denotes the initial pretrained model, for which VPD is exactly zero; T1–T7 denote checkpoints after the corresponding tasks. VNG-CT consistently achieves the lowest drift, showing that visual-necessity routing stabilizes the visual interface during continual tuning.

C.2 Independent Validation of Visual Dependence and Retention

Human evaluation protocol. We evaluate the gate on 900 held-out examples, with 300 from each of CoIN, MLLM-CL, and UCIT. Three authors independently annotate each example for visual dependence, blinded to gate scores, null-image outputs, and attention maps. We compare the gate with their visual-dependence labels and assessments of image benefit. All manual evaluation in this appendix is conducted by the authors.

Table 8 reports ranking quality (AUROC and AUPRC), Spearman correlation with human image benefit, the false-negative rate at gate threshold 0.35, and expected calibration error (ECE). The VNG score agrees more closely with human judgments than raw target NLL, visual-gradient norm, and image-token attribution. Across the three backbones, human AUROC ranges from 0.873 to 0.902. On benchmark-defined OCR, grounding, and counting/spatial categories, AUROC is 0.936, 0.948, and 0.918, respectively. These results support the model-relative score as a routing signal in the evaluated settings.

Table 8: **Gate validation against human judgments.** The held-out set contains 900 examples with three independent author annotations per example. FNR is in percent and uses human visual-dependence labels; it differs from the image-ablation-based FNR in Table 11.

Estimator	Human AUROC ↑	Human AUPRC ↑	Spearman ↑	FNR@0.35 ↓	ECE ↓
Raw target NLL	0.642	0.671	0.281	34.7	0.171
Visual-gradient norm	0.704	0.729	0.394	28.9	0.143
Image-token attribution	0.758	0.781	0.486	22.6	0.112
VNG, LLaVA-1.5-7B	0.891	0.912	0.742	7.8	0.055
VNG, InternVL-Chat-7B	0.902	0.921	0.758	7.1	0.051
VNG, Qwen2-VL-7B	0.873	0.896	0.711	9.4	0.063

Visual evidence versus language priors. We evaluate cases in which image evidence conflicts with language priors, independently of null-image gate labels. Table 9 reports visual-conflict accuracy, prior-following rate, and consistency under image interventions. Relative to SAME, VNG-CT improves conflict accuracy by 6.1 points, reduces prior-following by 6.6 points, and improves intervention consistency by 6.8 points. It also improves on the pretrained checkpoint on all three

measures. This audit provides evidence of reduced prior-following in the evaluated conflict setting. Appendix C.3 examines recovery from deliberately biased initialization.

Table 9: **Pixel-versus-prior conflict audit.** The audit uses no null-image gate labels. All entries are percentages.

Method	Visual-conflict Acc. ↑	Prior-following rate ↓	Image-intervention consistency ↑
Pretrained checkpoint	57.9	34.8	63.1
Sequential LoRA	49.2	44.8	56.1
Global weighting	57.8	35.6	64.7
SAME	61.1	30.9	69.3
VNG-CT	**67.2**	**24.3**	**76.1**

Retention beyond gate-defined diagnostics. Table 10 reports human old-visual scores, Human-VFR, OCR and grounding retention, and centered kernel alignment (CKA) on real images. Their labels and metrics are independent of ν and null images. Human-VFR uses human visual-dependence labels, while VFR uses image-ablation-based subsets. VNG-CT improves over SAME on each measure, including a reduction in Human-VFR from 5.7 to 3.8 and an increase in real-image CKA from 0.921 to 0.968. Agreement across these measurements supports visual retention independently of diagnostics defined by the routing rule.

Table 10: **Independent visual-retention measurements.** The first four metrics use percentage-scale scores; CKA is unitless. Human-VFR is reported separately from the original VFR.

Method	Human old-visual score ↑	Human-VFR ↓	OCR retention ↑	Grounding retention ↑	Real-image CKA ↑
Sequential LoRA	52.4	17.2	58.1	54.7	0.781
Global weighting	61.8	7.0	70.4	66.3	0.891
SAME	65.7	5.7	73.2	69.1	0.921
VNG-CT	**70.9**	**3.8**	**77.8**	**74.6**	**0.968**

C.3 Gate Behavior and Interpretability

Gate–anchor gradient alignment. We test whether the gate score ν_i reflects both output-level image dependence and the direction of the visual-path update. For visually grounded high-ν_i examples, we expect stronger gradient alignment with held-out visual-grounding anchors.

Let $\mathcal{A}$ denote a small anchor set whose answers strongly depend on visual evidence. We compute the average visual-path anchor gradient as

$$\bar{g}_{\mathcal{A}}^{\psi} = \frac{1}{|\mathcal{A}|} \sum_{j \in \mathcal{A}} \nabla_{\psi} \mathcal{L}_j. \tag{59}$$

For each training example i, we measure its visual-grounding alignment by

$$a_i^{\psi} = \cos\left(\nabla_{\psi} \mathcal{L}_i, \bar{g}_{\mathcal{A}}^{\psi}\right). \tag{60}$$

A larger a_i^{ψ} means that the example pushes the visual path in a direction more consistent with visually grounded updates.

Figure 8 shows a clear monotonic relation between ν_i and visual-grounding gradient alignment. Low-ν examples frequently have near-zero or negative cosine similarity, indicating that their visual gradients are weakly related or even conflicting with grounding-preserving updates. High-ν examples concentrate in the positive-alignment region across task types. This supports routing visually necessary examples to the main visual path and low-necessity examples to the other paths.

Backbone-level consistency of visual necessity. We examine the consistency of sample-level visual-necessity scores across backbones. For each training example, we compute the gate score with

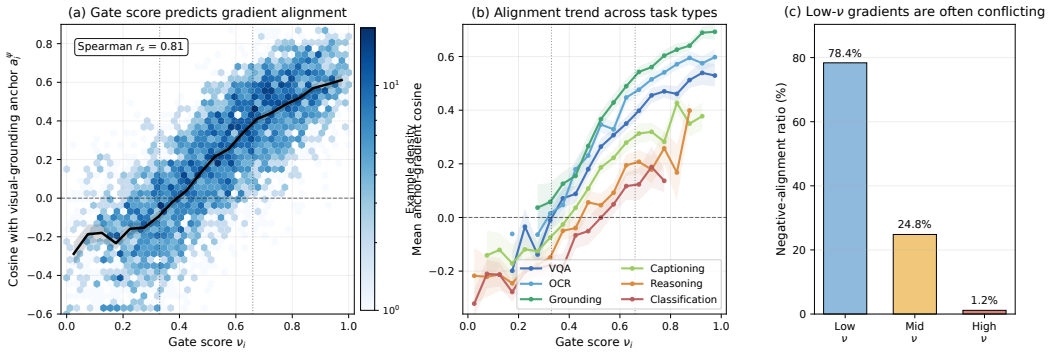

Figure 8: **Gate score and visual-grounding gradient alignment. (a)** Density scatter between the gate score ν_i and the cosine similarity a_i^{ψ} to the visual-grounding anchor gradient. Darker hexagons indicate denser regions. The black curve shows the binned mean with a confidence band. **(b)** Task-wise alignment trends show that the positive relation between ν_i and gradient alignment is consistent across task types. **(c)** Low-ν examples contain a larger fraction of negative-alignment gradients, while high-ν examples are mostly aligned with the visual-grounding anchor. These results indicate that the gate score is predictive of whether an example provides a qualified visual-path update.

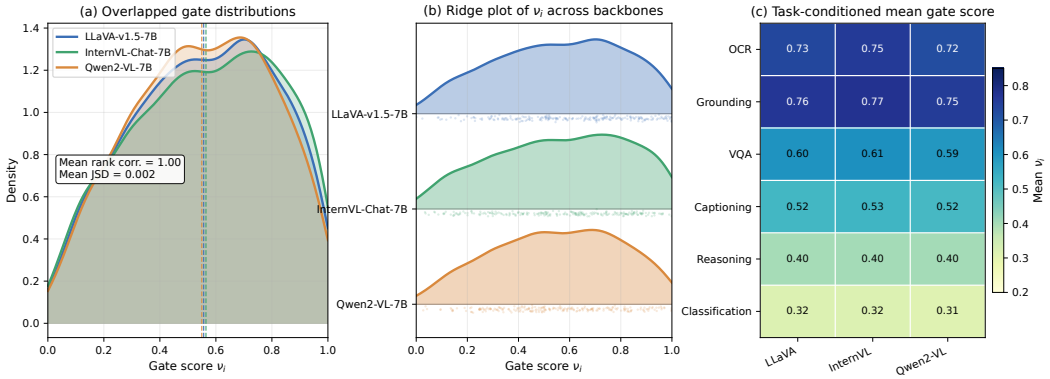

Figure 9: **Backbone-level consistency of the visual-necessity gate. (a)** The marginal distributions of ν_i largely overlap across LLaVA-v1.5-7B, InternVL-Chat-7B, and Qwen2-VL-7B. **(b)** The ridge plot shows similar gate-score profiles across backbones, with only mild shifts in distribution sharpness. **(c)** Task-conditioned mean gate scores preserve the same semantic ordering across backbones: visually grounded tasks such as OCR and grounding receive higher gate scores, while language-dominant or weakly visual tasks receive lower scores. These results indicate that example and task structure primarily determine visual necessity across the evaluated backbones.

three different multimodal backbones: LLaVA-v1.5-7B, InternVL-Chat-7B, and Qwen2-VL-7B. All backbones use the same counterfactual scoring protocol and the same null-image construction. Let $\nu_i^{(b)}$ denote the visual-necessity score of example i under backbone b. We compare both the marginal gate distributions and the task-conditioned mean gate scores.

As shown in Figure 9, the gate distributions are highly consistent across backbones. Although different backbones induce small calibration shifts, their overall distribution shapes and task-level rankings remain similar. This suggests that the visual-necessity score captures a stable property of image dependence across the evaluated backbones. The result also explains why VNG-CT transfers well to InternVL-Chat and Qwen2-VL in the main experiments.

Semantic structure of the visual-necessity gate. Beyond example-level validity, we ask whether the visual-necessity gate captures structured semantic patterns across abilities and datasets. We expect ν_i to be higher for abilities requiring visual evidence, such as OCR, localization, counting, and spatial

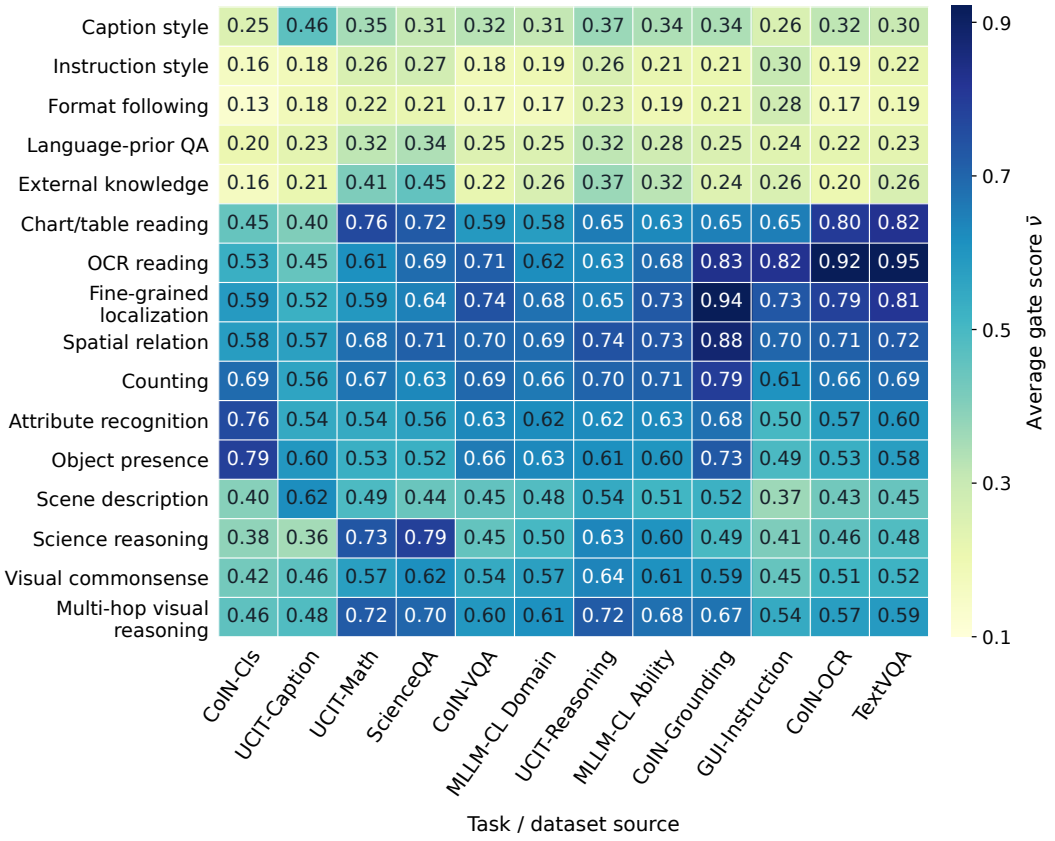

Figure 10: **Semantic structure of the visual-necessity gate.** Each cell reports the average gate score $\bar{\nu}_{a,d}$ for a semantic ability and a task/dataset source. Rows and columns are reordered by hierarchical clustering. High visual-dependence abilities, such as OCR, localization, counting, and spatial relation reasoning, form a clear high-ν cluster, while language-dominant abilities, such as format following, instruction style, and language-prior QA, form a low-ν cluster. This indicates that the gate captures semantically meaningful visual-dependence patterns.

reasoning, and lower for language-dominant abilities, such as format following or commonsense-only questions.

We group training examples by semantic ability and task source, and compute the average gate score:

$$\bar{\nu}_{a,d} = \frac{1}{|\mathcal{S}_{a,d}|} \sum_{i \in \mathcal{S}_{a,d}} \nu_i, \tag{61}$$

where $\mathcal{S}_{a,d}$ denotes examples belonging to semantic ability a and dataset/task source d. We visualize the resulting ability–task matrix with hierarchical clustering.

Figure 10 shows that the gate has a clear semantic organization. Abilities that require direct visual evidence consistently receive higher gate scores across datasets, whereas language-dominant abilities remain low even when an image is present. The clustered structure also reveals task-level similarity: OCR- and grounding-heavy datasets are grouped together, while instruction-style and language-prior tasks form a separate low-necessity region. This supports the interpretation that VNG-CT learns a semantically structured routing signal for deciding when visual-path updates are justified.

Gate false-negative analysis. Examples from new visual domains or fine-grained tasks may initially have small true-image versus null-image loss gaps. Their low gate values can delay useful visual updates. We analyze how often visually necessary examples receive insufficient routing weights.

To quantify this failure mode, we construct a visually necessary subset $\mathcal{V}^\star$ from held-out examples. An example is included in $\mathcal{V}^\star$ if it belongs to a visually demanding task type, such as OCR, grounding,

Table 11: **Gate false-negative analysis and conservative safeguards.** FNR measures the fraction of visually necessary examples incorrectly assigned a low effective visual-path gate. SGM measures the suppressed visually qualified gradient mass. Visual-required Acc. is evaluated on $\mathcal{V}^\star$. UCIT FAA and VFR are reported under the same LLaVA-v1.5-7B setting as the main ablation table.

Method	FNR ↓	SGM ↓	Visual-required Acc. ↑	UCIT FAA ↑	VFR ↓
VNG-CT	4.9	10.7	75.6	76.47	3.36
VNG-CT + gate floor	3.8	9.2	75.8	76.42	3.40
VNG-CT + visual warm-up	3.5	8.7	75.9	76.36	3.44
VNG-CT + warm-up + floor	2.9	7.6	76.0	76.21	3.52

counting, spatial relation reasoning, or fine-grained attribute recognition, and if its image-ablation score drop exceeds a threshold δ_{vis}:

$$y_i^\star = \mathbf{1}\left[\Delta_i^{\text{abl}} > \delta_{\text{vis}}\right] \cdot \mathbf{1}\left[\text{type}(i) \in \mathcal{T}_{\text{vis}}\right], \tag{62}$$

where Δ_i^{abl} follows the same null-image ablation protocol as Eq. (53), and $\mathcal{T}_{\text{vis}}$ denotes visually demanding task types. We define a gate false negative as a visually necessary example whose effective visual-path routing weight is below a threshold τ_g:

$$\text{FN}_i(\tau_g) = \mathbf{1}[y_i^\star = 1] \cdot \mathbf{1}[\tilde{\nu}_i < \tau_g], \tag{63}$$

where $\tilde{\nu}_i$ denotes the effective gate used for visual-path routing. The false-negative rate is

$$\text{FNR}(\tau_g) = \frac{\sum_i \mathbf{1}[y_i^\star = 1]\mathbf{1}[\tilde{\nu}_i < \tau_g]}{\sum_i \mathbf{1}[y_i^\star = 1] + \epsilon}. \tag{64}$$

We also measure the suppressed visually qualified gradient mass:

$$\text{SGM} = \frac{\sum_i y_i^\star (1 - \tilde{\nu}_i)\|\nabla_\psi \mathcal{L}_i\|_2}{\sum_i y_i^\star \|\nabla_\psi \mathcal{L}_i\|_2 + \epsilon}. \tag{65}$$

While FNR measures how often visually necessary examples are assigned insufficient visual-path weight, SGM measures how much useful visual-gradient mass is suppressed.

We evaluate two conservative safeguards. The first is a gate floor,

$$\tilde{\nu}_i = \max(\nu_i, \nu_{\min}), \tag{66}$$

which prevents the visual path from being fully shut off when the counterfactual score is uncertain. The second is a task-level visual warm-up strategy,

$$\tilde{\nu}_i = \begin{cases} 1, & \text{if the current task has been trained for fewer than } K_{\text{warm}} \text{ steps,} \\ \nu_i, & \text{otherwise,} \end{cases} \tag{67}$$

which allows the model to acquire an initial visual signal before relying on counterfactual gating. We set $\tau_g = 0.35$, $\nu_{\min} = 0.20$, and K_{warm} to the first 5% of updates in each task.

Table 11 shows a low false-negative rate under the standard initialization and image-ablation-based subset used here. The safeguards reduce FNR and suppressed gradient mass, with unchanged or lower aggregate UCIT accuracy and slightly higher VFR. The default gate therefore provides a favorable trade-off in this setting. Vision-blind initialization and FNR against independent human labels are evaluated separately below and in Table 8.

Recovery from biased initialization. We additionally stress-test the gate under two deliberately biased initializations. For *moderate text bias*, preconditioning adds a weak answer-correlated text cue while retaining the real image; the cue is removed for recovery. For *severe visual blindness*, preconditioning replaces real images with the mean-color null image; real images are restored for recovery. Both settings then use a clean OCR, grounding, and counting stream.

Table 12 tracks effective routing FNR during recovery. With the default gate, moderate-bias FNR decreases from 18.6% to 7.9%, whereas severe-bias FNR remains at 18.9% at the end. Under severe bias, visual warm-up for the first 5% of task updates followed by a 0.20 gate floor reduces final FNR

Table 12: **Recovery under deliberately biased initialization.** FNR is measured using the effective visual-path routing weight. The three FNR columns and final accuracy are percentages. These stress-test results are distinct from the standard-initialization analysis in Table 11.

Recovery setting	Initial effective FNR ↓	FNR after 10% updates ↓	Final FNR ↓	Final visual-required Acc. ↑
Moderate, default gate	18.6	11.2	7.9	74.8
Severe, default gate	41.8	28.5	18.9	70.6
Severe, 5% warm-up + 0.20 floor	0.0	8.7	6.8	74.8

Table 13: **Ablation over counterfactual views for visual-necessity scoring.** The default null-image counterfactual achieves the strongest overall trade-off, while patch-shuffled, mismatched-image, and learned-null-token variants obtain similar gate-ranking quality and continual performance. This indicates that the visual-necessity signal is consistent across these counterfactual constructions.

Counterfactual view for scoring	Gate–drop Spearman ↑	Visual-dep. AUROC ↑	UCIT FAA ↑	VFR ↓	VCS ↑
Null image (default)	0.86	0.94	76.47	3.36	45.89
Patch-shuffled image	0.82	0.90	75.92	3.58	44.96
Mismatched image	0.84	0.92	76.16	3.44	45.38
Learned null token	0.85	0.93	76.32	3.40	45.63

to 6.8% and increases visual-required accuracy from 70.6 to 74.8. The initial 0.0% FNR reflects the warm-up routing weight of one; this effective routing rate is distinct from the raw necessity estimate. The floor remains below the 0.35 FNR threshold, so false negatives can persist after applying it.

Recovery is possible because low but nonzero gates pass visual gradients, and related visually clear examples can improve the shared visual path. At the next task boundary, a scoring snapshot initialized from the updated model can yield a larger true-image versus null-image loss gap. Within the current task, the frozen snapshot keeps the gate of a fixed example unchanged. Remaining false negatives concentrate in visually ambiguous or shifted examples, including language-prior classification, reasoning, captioning, and unseen fine-grained domains. Explicit grounding, OCR, and counting/spatial tasks are less affected.

Counterfactual-view ablation. We evaluate sensitivity to counterfactual input construction using four views: a fixed mean-color null image, a patch-shuffled image, a randomly mismatched image, and an input-independent learned null token. The default mean-color image preserves the multimodal input shape while removing image-specific evidence.

Let $Z_i = B_\psi(E_v(x_i))$ denote the projected visual tokens of the true image. For each counterfactual type c, we construct counterfactual visual tokens Z_i^c, compute the corresponding visual necessity score s_i^c, and derive a normalized gating weight v_i^c:

$$s_i^c = \ell_{\bar{\theta}_t}(a_i^t \mid Z_i^c, q_i^t) - \ell_{\bar{\theta}_t}(a_i^t \mid Z_i, q_i^t), \qquad \nu_i^c = \mathrm{sg}\left[\mathrm{clip}_{[0,1]}\left(\frac{s_i^c - \rho}{\tau_\nu}\right)\right]. \tag{68}$$

The four variants differ only in how Z_i^c is obtained. The null-image variant forwards a fixed mean-color image through the same vision encoder and projector. The patch-shuffle variant randomly permutes non-overlapping image patches before visual encoding. The mismatched-image variant replaces x_i with an image sampled from another example in the same task, avoiding the same answer category when labels are available. The learned-null-token variant replaces the visual tokens with an input-independent token sequence $Z_\varnothing^{\mathrm{learn}}$, initialized from the running mean of projected visual tokens and fitted with a label-free moment-matching objective so that it preserves in-distribution feature statistics without carrying image-specific semantics.

For each variant, we train VNG-CT with the corresponding gate ν_i^c and evaluate both aggregate continual-learning performance and gate-ranking quality. Gate ranking is measured by the Spearman correlation between ν_i^c and the image-ablation score drop Δ_i^{abl}, as well as the AUROC for identifying visually dependent examples using the binary label in Eq. (44).

Table 13 shows that VNG-CT is robust to the choice of counterfactual view. The learned-null-token and mismatched-image variants achieve gate-ranking quality close to the default null-image variant,

suggesting that removal of image-specific evidence primarily drives the gate. Patch shuffling performs slightly worse, likely because local texture and color remain informative for some recognition and scene-understanding examples. The default null image provides the best aggregate stability–plasticity trade-off, and the alternatives yield consistent conclusions. These results support the null-image score as a simple and reliable visual-necessity estimator.

Independent validation and cost of counterfactual views. Table 14 complements Table 13 with author-label validation, CoIN accuracy, and measured training overhead. Human AUROC uses author visual-dependence labels; the earlier table uses image-ablation-based labels. Across the four views, human AUROC ranges from 0.852 to 0.891 and CoIN accuracy from 68.04 to 68.61. The results are consistent across these constructions, with some variation in quality. The mean-color null achieves the highest human AUROC and CoIN accuracy at the lowest training overhead among the tested choices.

Table 14: **Counterfactual views: external gate validation and training cost.** Human AUROC uses the independent 900-example annotations. Overhead is relative to sequential LoRA in the LLaVA/UCIT timing setup of Table 6.

Counterfactual view	Human AUROC ↑	CoIN ACC ↑	Training overhead
Mean-color null	0.891	68.61	+24.9%
Patch shuffled	0.852	68.04	+27.6%
Mismatched image	0.879	68.29	+29.3%
Learned null token	0.884	68.45	+25.9%

C.4 Calibration and Hyperparameter Analysis

Calibration-path absorption. We analyze how the calibration path χ handles visually weak supervision. VNG-CT routes low-necessity examples to this path with weight $1-\nu_i$ while suppressing their direct updates to the main visual path.

For each task and each gate interval, we measure two quantities. First, we compute the normalized calibration residual norm:

$$C_{t,b}^{\mathrm{res}} = \frac{1}{|\mathcal{D}_{t,b}|} \sum_{i \in \mathcal{D}_{t,b}} \frac{\|\Delta_\chi(Z_i, \bar{u}_i)\|_F}{\|Z_i\|_F + \epsilon}, \tag{69}$$

where $\mathcal{D}_{t,b}$ denotes examples from task t whose gate scores fall into bin b, and $\Delta_\chi(Z_i, \bar{u}_i)$ is the residual produced by the calibration module. Second, we measure the relative reduction of the invariance loss:

$$C_{t,b}^{\mathrm{inv}} = \frac{100}{|\mathcal{D}_{t,b}|} \sum_{i \in \mathcal{D}_{t,b}} \frac{\mathcal{R}_{i,\mathrm{before}}^{\mathrm{inv}} - \mathcal{R}_{i,\mathrm{after}}^{\mathrm{inv}}}{\mathcal{R}_{i,\mathrm{before}}^{\mathrm{inv}} + \epsilon}. \tag{70}$$

A large value of $C_{t,b}^{\mathrm{res}}$ indicates that χ actively modifies the projected visual tokens, while a large value of $C_{t,b}^{\mathrm{inv}}$ indicates that this modification reduces sensitivity to irrelevant visual evidence.

Figure 11 shows that calibration activity varies across samples. Its residual norm and invariance loss reduction are concentrated in low- and mid-ν bins, where image-present examples are more likely to contain visually weak or language-dominant supervision. This pattern supports the intended role of χ: it absorbs visually weak supervision and learns invariance to irrelevant visual inputs, while leaving high-necessity visual examples to update the main visual path.

Capacity-matched calibration controls. Table 15 compares five variants with 0.131M added trainable parameters against a no-calibration reference. The full additive and low-ν residual variants share the four-matrix calibrator ($N_\chi = 4dr_\chi = 131{,}072$). Extra LoRA and the generic residual use all samples without routing or invariance supervision. The low-ν residual uses routing alone; matched replacement and full additive combine routing and invariance supervision.

The full additive residual exceeds the extra-LoRA control by 1.53 UCIT FAA points and reduces Human-VFR by 1.9 points. It also improves over the low-ν residual without invariance and the matched replacement control. These capacity-matched comparisons empirically support combining routing, invariance supervision, and an identity-preserving residual.

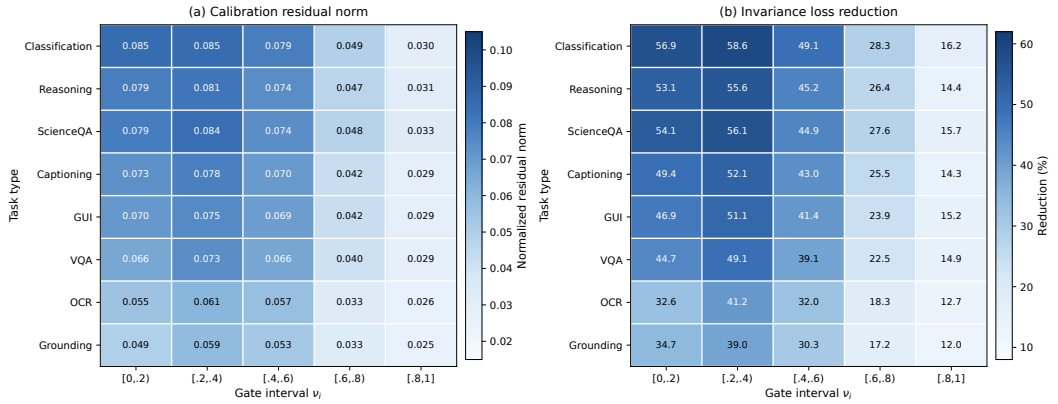

Figure 11: **What does the calibration path absorb?** We group examples by task type and visual-necessity gate interval. **(a)** Normalized calibration residual norm. **(b)** Invariance loss reduction after calibration. The calibration path is most active on low- and mid-ν examples, especially in language-dominant or weakly grounded tasks such as classification, reasoning, captioning, and science QA. Strongly visual tasks such as grounding and OCR show weaker calibration activity in high-ν bins, because these examples are mainly routed to the visual path.

Table 15: **Calibration design under matched added capacity.** Each augmented variant adds 0.131M trainable parameters. An asterisk marks controls rerun at this matched capacity. Human-VFR is the independent human-based measure, while VCS follows the original counterfactual diagnostic.

Variant	Added params.	CoIN ACC ↑	UCIT FAA ↑	Human-VFR ↓	VCS ↑
No calibration	0	67.56	74.84	5.1	42.89
Extra LoRA*	0.131M	67.66	74.94	5.7	42.65
Generic residual*	0.131M	67.85	75.11	5.5	43.01
Low-ν, no invariance	0.131M	68.07	75.56	4.6	44.30
Matched replacement*	0.131M	68.15	75.74	4.5	44.70
Full additive	0.131M	**68.61**	**76.47**	**3.8**	**45.89**

Parameter sensitivity analysis. We analyze the sensitivity of VNG-CT to five key hyperparameters: the visual-necessity margin ρ, gate temperature τ_ν, invariance weight λ_{inv}, calibration rank r_χ, and LoRA rank r_{LoRA}. For each study, we vary one hyperparameter while keeping the others fixed to their default values. Figure 6 reports the mean and standard deviation over three random seeds.

Overall, VNG-CT remains stable across a broad range of hyperparameter values. The performance curves of CoIN ACC, UCIT FAA, DCL Avg., and VCS change smoothly, while VFR stays consistently low. This indicates that the path-specific routing mechanism provides gains across a broad range of settings.

For the visual-necessity gate, moderate values of ρ and τ_ν work best. A small ρ or overly sharp gate may allow noisy low-necessity examples to update the visual path, whereas overly conservative gating can suppress useful visual supervision. For the calibration path, a moderate λ_{inv} and rank r_χ provide enough capacity to absorb low-necessity supervision without over-enforcing image invariance. Finally, the LoRA rank $r_{\text{LoRA}} = 16$ gives a good stability–plasticity trade-off: smaller ranks limit adaptation, while larger ranks slightly increase visual-path drift. These results confirm that VNG-CT is robust to practical hyperparameter choices.

C.5 Robustness to Continual-Learning Protocols

Matched controls across backbones. Table 16 compares no gating, global sample weighting, and VNG-CT within each of the three backbones. No gating allows all image-present examples to update the visual path; global weighting applies ν_i to the whole sample loss. VNG-CT outperforms

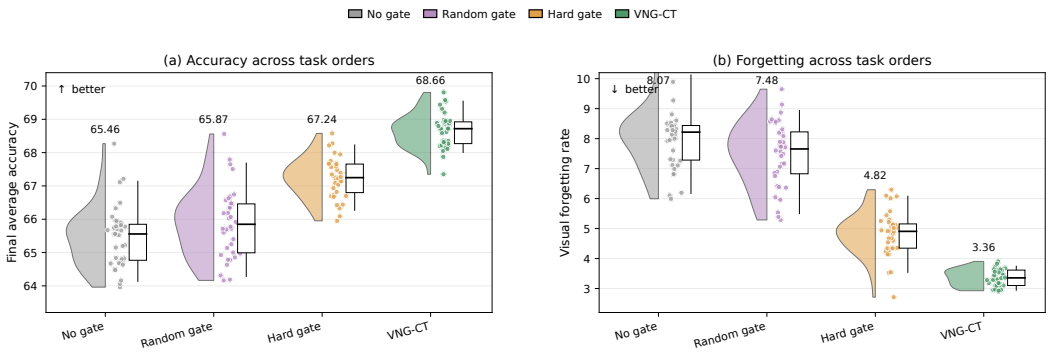

Figure 12: **Robustness to task-order variation.** We evaluate different gate variants under multiple task-order permutations and three random seeds per order. Each raincloud plot shows the distribution over all order–seed runs. **(a)** VNG-CT consistently achieves higher final average accuracy across task orders. **(b)** VNG-CT also yields lower and less variable visual forgetting. The result suggests that the visual-necessity gate provides stable path-specific routing across diverse continual streams.

both controls on all four reported metrics for every backbone. Averaged over backbones, its gains over global weighting are 1.70 CoIN ACC points, 2.22 UCIT FAA points, and 2.32 DCL Avg. points. These within-backbone controls support the routing mechanism beyond a single projector architecture.

Table 16: **Matched routing controls across backbones.** Comparisons are made within each backbone. VFR uses the image-ablation-based visual-forgetting protocol.

Backbone	Method	CoIN ACC ↑	UCIT FAA ↑	DCL Avg. ↑	VFR ↓
LLaVA-1.5-7B	No gate	65.74	72.58	67.43	6.42
	Global weighting	66.83	73.81	68.92	4.12
	VNG-CT	68.61	76.47	71.38	3.36
InternVL-Chat-7B	No gate	66.88	72.21	68.96	5.94
	Global weighting	67.71	73.47	70.14	3.88
	VNG-CT	69.60	75.35	72.59	3.09
Qwen2-VL-7B	No gate	63.84	70.71	64.95	6.88
	Global weighting	64.93	72.14	66.42	4.79
	VNG-CT	66.35	74.26	68.46	4.02

Task-order robustness. We evaluate robustness to the order of visual, language-dominant, and reasoning-oriented tasks in the continual stream.

We construct multiple task-order permutations from the same benchmark tasks. For each order, we train the model with three random seeds and report the final average accuracy and visual forgetting rate. We compare VNG-CT with three gate-related variants: *No gate*, which updates the visual path using all image-present examples; *Random gate*, which preserves the same gate-value distribution but randomly permutes gate scores within each mini-batch; and *Hard gate*, which replaces the soft score with a binary visual-path decision.

As shown in Figure 12, VNG-CT achieves the highest median final accuracy and the lowest median forgetting across different task orders. The random-gate variant performs substantially worse, confirming the importance of sample-specific visual-path weights. Hard gating improves over no gating but exhibits larger variance, suggesting that binary routing is more sensitive to task-order-induced score noise. VNG-CT maintains a compact performance distribution, indicating that soft visual-necessity routing provides a more robust stability–plasticity trade-off across continual streams.

Robustness to low-visual-dependence streams. Finally, we evaluate when visual-necessity routing is most beneficial by varying the proportion of low-visual-necessity examples in the continual stream. We construct controlled streams with different r_{low} values while keeping the task order and total

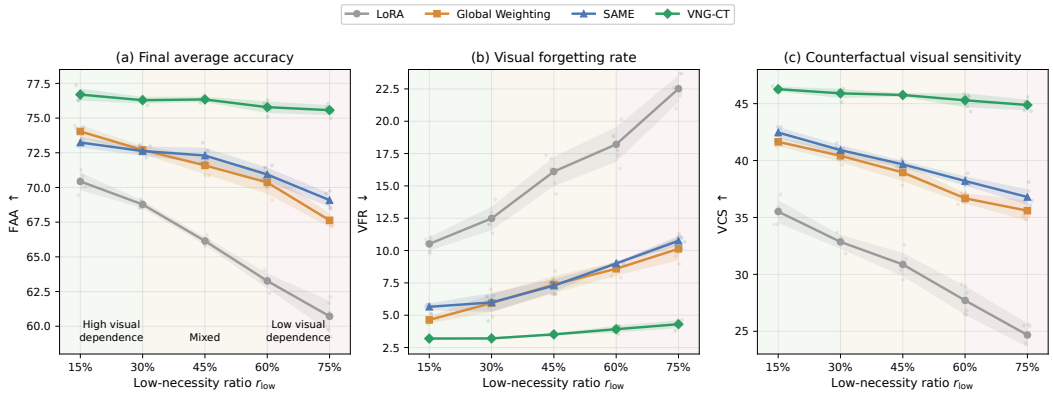

Figure 13: **Controlled visual-dependence streams.** As the fraction of low-visual-necessity examples increases, VNG-CT maintains higher FAA and VCS with lower VFR than LoRA and global weighting. See Appendix B.9, *Controlled-stream protocol* for construction details.

number of examples fixed, and evaluate FAA, VFR, and VCS. As shown in Figure 13, standard LoRA degrades rapidly as r_{low} increases, showing lower final accuracy, weaker counterfactual visual sensitivity, and higher visual forgetting. Global weighting is more stable but sacrifices adaptation because it globally suppresses useful low-necessity language supervision. VNG-CT degrades smoothly and keeps visual forgetting low, indicating that path-specific routing is especially useful when image-present data contain many visually weak examples.

C.6 Unified Baseline Comparisons

We rerun a subset of code-available baselines and VNG-CT with identical splits, task order, training steps, and validation protocol. Each method uses a search budget of 12 configurations and three random seeds; the baselines use their official code and configurations. Table 17 reports mean and standard deviation across seeds for this controlled subset of the main comparisons.

Table 17: **Unified reruns on LLaVA-1.5-7B.** All methods use the same evaluation and tuning budget: 12 configurations and three seeds. Entries are mean ± standard deviation across seeds.

Method	CoIN ACC ↑	UCIT FAA ↑	DCL Avg. ↑
Sequential LoRA	28.74 ± 0.31	60.86 ± 0.42	45.61 ± 0.37
HiDe-LLaVA	63.95 ± 0.22	65.93 ± 0.35	52.60 ± 0.48
SMoLoRA	64.97 ± 0.25	67.60 ± 0.31	64.72 ± 0.43
SAME	66.82 ± 0.18	72.53 ± 0.29	68.31 ± 0.35
VNG-CT	$\mathbf{68.61 \pm 0.14}$	$\mathbf{76.47 \pm 0.35}$	$\mathbf{71.38 \pm 0.26}$

SAME is the strongest baseline among the unified reruns in Table 17. Table 18 reports paired differences between VNG-CT and SAME with 95% confidence intervals. The intervals exclude zero for the three benchmark scores and Human-VFR.

Table 18: **Paired comparison with SAME.** Differences are VNG-CT minus SAME, in points. Negative values are favorable for Human-VFR. SAME is the strongest baseline within the reported unified reruns.

Metric	SAME	VNG-CT	Difference	Paired 95% CI
CoIN ACC ↑	66.82	68.61	$+1.79$	$[1.54, 2.05]$
UCIT FAA ↑	72.53	76.47	$+3.94$	$[3.38, 4.49]$
DCL Avg. ↑	68.31	71.38	$+3.07$	$[2.52, 3.63]$
Human-VFR ↓	5.7	3.8	-1.9	$[-2.31, -1.46]$

C.7 Stratified Case Audit and Failure Analysis

We predefine three strata—old-task visual-grounding retention, new-task visual acquisition, and low-necessity invariance—and randomly sample 40 cases from each. Two blinded author-evaluators judge correctness and invariance. Table 19 reports all 120 cases, including 18 VNG-CT failures. VNG-CT succeeds on 102 cases, compared with 85 for SAME and 72 for sequential LoRA, with gains in all three strata.

Table 19: **Stratified audit of 120 randomly sampled cases.** Entries count correct responses for the retention/acquisition strata and successful invariance judgments for the invariance stratum. The final column counts VNG-CT failures.

Predefined stratum	Cases	LoRA	SAME	VNG-CT	VNG-CT failures
Old-task visual-grounding retention	40	21	29	34	6
New-task visual acquisition	40	29	30	33	7
Low-necessity invariance	40	22	26	35	5
Total	120	72	85	**102**	18

The observed behaviors include preserving OCR characters and grounding regions after later tasks, acquiring new visual attributes and relations, and keeping answers stable when an irrelevant image is replaced. These observations complement the aggregate retention and acquisition scores. One failure involves an unseen fine-grained visual domain: the example receives a default gate of $\nu = 0.29$, below the 0.35 threshold, despite requiring visual evidence. Visual warm-up corrects this low-score assignment.

D Limitations

Model-relative necessity and inherited bias. The gate measures the current scoring model's benefit from an image for a given target, so it can assign a low score to a visually necessary example when the model underuses vision. The snapshot is fixed throughout each task and its scores are detached; biases inherited from pretraining or earlier tasks can persist. Human-label agreement and the pixel-versus-prior audit provide complementary validation within the tested settings.

Recovery and the stability–plasticity trade-off. The default soft gate recovers from moderate text bias and shows incomplete recovery under deliberately severe visual blindness (Appendix C.3). Visual warm-up and a gate floor improve recovery in this stress test, which still requires relevant visual signals. Broader visual-path updates can increase forgetting, as in Table 11. Unseen fine-grained domains and ambiguous visual evidence remain difficult. The sample-level score assigns one weight to all tokens, regions, and relations, limiting separation of useful and spurious supervision within an example.

Scope of diagnostics and empirical coverage. LNVL measures routed-gradient leakage using independent reference necessity scores. $\mathrm{VNR}_{\mathrm{route}}$ ranks routing weights, while $\mathrm{VNR}_{\mathrm{post}}$ ranks final-checkpoint counterfactual scores. Both use shared reference labels, and VFR uses reference-defined visual subsets; the gate–ablation correlation in Figure 4(a) checks internal consistency. The 900-example author-annotated set and 120-case audit complement these diagnostics with human and real-image evaluations. Empirical coverage comprises three benchmark suites and three backbones, with three seeds for the unified reruns.

Computational cost. Counterfactual scoring adds training time and memory: the measured LLaVA/UCIT overhead is 24.9% in time and 4.6GB of peak memory per GPU over sequential LoRA. The scoring snapshot and null-image pass are removed at inference, and the retained calibration module adds 0.6% measured latency. These costs constrain use under tighter resource budgets; overhead also depends on counterfactual construction.

E Broader Impacts

This work aims to improve the reliability of continual multimodal adaptation by reducing unintended visual-path drift caused by image-present but visually unnecessary supervision. The proposed visual-necessity-guided routing may benefit deployed multimodal assistants, educational tools, accessibility systems, and domain-specific visual reasoning applications where models must adapt to new instructions and data while preserving previously acquired visual grounding. Complementary work on deployment security addresses cloud-infrastructure vulnerability risk quantification [Shao, 2026b] and interpretable network-traffic anomaly detection [Shao, 2026a]. Security-focused work on medical LLM agents also studies adversarial robustness and evidence-constrained decision making [Hu, 2026].

F LLM Usage

Large language model tools were used for language editing, grammar correction, LaTeX formatting, code-snippet drafting, and script debugging.

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 describe visual-necessity estimation, path-specific gradient routing, and calibration. The method and experiments evaluate these components on CoIN, MLLM-CL, and UCIT-style continual instruction streams.

2. **Limitations**

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

   Answer: [Yes]

   Justification: Appendix D discusses model-relative necessity, inherited bias, recovery under severe visual blindness, the stability–plasticity trade-off, diagnostic scope, empirical coverage, and computational cost.

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 study is empirical. Its mathematical content defines the visual-necessity score, calibration regularizer, routed gradients, continual update rule, and diagnostic metrics.

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 paper describes the benchmarks, task streams, metrics, comparison protocol, backbones, parameter groups, scoring and calibration procedures, optimizer settings, hyperparameters, seeds, ablations, and diagnostics. Algorithm 1 specifies training; Appendix B provides experimental details.

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: Code and reproduction scripts are not included with the paper. The experiments use existing benchmarks and pretrained backbones, with implementation, optimization, and evaluation protocols described in the paper.

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: The main text and appendix specify the benchmark and evaluation settings, backbone models, frozen and trainable parameters, LoRA and calibration hyperparameters, optimizer, learning-rate schedule, batch size, precision, and validation-based model selection.

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: Appendix C.6 reports mean and standard deviation across three random seeds and paired 95% confidence intervals for differences from SAME. Appendix B.8 distinguishes the controlled rerun subset from published baseline results.

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: Appendix B.10 describes the computational setup.

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The work uses existing benchmarks and pretrained multimodal backbones. The authors conduct manual evaluation on benchmark examples and model outputs, as described in Appendix C.2. External participants are not recruited, and no new scraped dataset or high-risk pretrained model is released.

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: Appendix E discusses potential benefits for multimodal assistants, educational tools, accessibility systems, and domain-specific visual reasoning applications.

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 work develops a training framework on existing benchmarks and backbones, without releasing a new pretrained model, image generator, scraped dataset, or other standalone high-risk asset.

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: [No]

    Justification: The paper credits the datasets, benchmarks, baselines, and pretrained backbones. The inventory of licenses, access conditions, and terms of use is incomplete.

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 contribution is an algorithmic framework and diagnostic protocol built on existing benchmarks and models. No new dataset, benchmark, pretrained model, or standalone software asset is released.

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 authors annotate existing benchmark examples and model outputs (Appendices C.2 and C.7); external participants are not recruited.

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 evaluation uses existing benchmark examples and model outputs and is conducted by the authors (Appendix C.2). It does not recruit external participants or study their behavior or personal data.

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: [Yes]

    Justification: Appendix F discloses LLM assistance with language editing, grammar correction, LaTeX formatting, code-snippet drafting, and script debugging.