ELGAR: EXECUTION-LINKED GATING FOR ADAPTER ROUTING IN CONTINUAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Selecting a visual expert changes the representation that language experts must process. Execution-Linked Gating for Adapter Routing (ELGAR) turns this dependency into an allocation rule: an instruction-conditioned gate selects a visual projector, and a language gate observes that same projector's output to combine frozen low-rank adaptation (LoRA) updates, with one projector and one autoregressive path at deployment. Our finite-candidate analysis characterizes when an observation supports an optimal allocation and shows how a pooled bank mean can hide distinctions preserved by the selected output. Three paired source-order repeats on six UCIT tasks with 3,000 examples each give 75.53 mean final average score (FAA) for selected-output routing, versus 74.85 for bank-mean routing and 73.75 for static allocation. The additional 0.68-point gain is positive in every repeat with the visual route and task-origin supervision fixed. Separate matched pairs favor output content over projector identity in every repeat, with a mean gain of 0.62 points. Complete ELGAR reaches 75.63 FAA on UCIT–LLaVA and leads reproduced continual baselines in FAA and backward transfer across all four UCIT/DCL–backbone settings. Thus observing the executed representation adds measurable allocation value beyond knowing which branch ran.

1 INTRODUCTION

Retaining task experts preserves capabilities but leaves a crucial execution decision: which language adaptations should process the selected visual representation? In continual multimodal instruction tuning (Wang et al., 2024; Guo et al., 2025c), a visual projector creates the tokens seen by the decoder, and language experts process those tokens. The first decision therefore changes the input to the second. We study what the language allocator should observe about that executed computation.

Multimodal expert allocation, projector routing and dynamic adapter composition supply the necessary components (Zhu et al., 2025; Jin et al., 2025; Cheng et al., 2024; Wang et al., 2025). ELGAR addresses the case where the projector is itself a routed choice: the language allocator observes the outcome of that upstream decision. A raw-image summary describes the input; a mean over projectors describes the bank. Neither necessarily preserves the allocation-relevant distinctions in the visual output selected for a request. ELGAR supplies that output to both the decoder and the language gate, aligning the gate's observation with the representation it must allocate experts to process.

Figure 1 links this distinction to a matched test. With the visual route fixed, three paired source-order repeats give 75.53 mean final average score (FAA) for selected-output routing, 74.85 for bank-mean routing and 73.75 for static allocation. The 1.10-point benefit of dynamic allocation and the additional 0.68-point benefit of the executed observation are distinct, and both recur in every repeat. A separate projector-identity comparison gains 0.62 points on average, with the visual route, request summaries and training budget matched. Together these controls test the value of the output representation beyond a bank summary or branch label.

ELGAR implements this dependency with two lightweight gates and signed module coefficients for layer-specific LoRA allocation. During training, a differentiable projector-output mixture feeds both the decoder and the language gate. Answer and task-origin supervision fit the allocation parameters while all experts remain frozen. Deployment uses one selected projector and one autoregressive trajectory, without task labels.

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

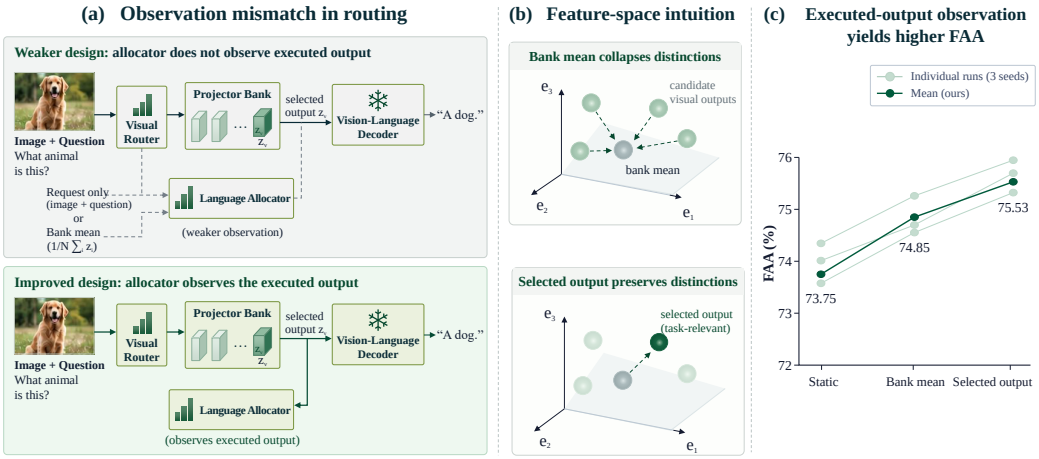


Figure 1: **The language allocator benefits from observing the visual output that the decoder actually receives.** **(a)** ELGAR supplies the selected projector output to both the decoder and the language allocator, rather than conditioning allocation on the request or projector-bank mean alone. The snowflake denotes frozen backbone weights. **(b)** The feature-space schematic illustrates how averaging candidate outputs can blur distinctions preserved by the selected output. **(c)** Three paired source-order repeats compare a fitted constant allocation (Static), a bank-mean allocator, and a selected-output allocator, with the visual route and expert bank fixed. Each receives 4,608 updates and is evaluated on six UCIT tasks with 3,000 examples each using LLaVA-1.5-7B. Pale lines show individual repeats; the dark line shows their means. Final average score (FAA), the mean of the six task scores, rises from 73.75 to 74.85 to 75.53. Higher is better.

Our finite-candidate analysis shows that a summary supports optimal allocation exactly when settings it leaves indistinguishable share an optimal choice. A fixed-projector construction gives identical raw-image and bank-mean summaries while the selected output separates the required allocations. Matched training then tests the value realized by a fitted gate.

These controls retain task-origin supervision, absent at inference; their 75.53 mean is distinct from complete ELGAR's 75.63 endpoint. The complete method improves FAA over the strongest reproduced continual baseline in all four evaluated settings, with gains of 3.48 and 3.43 points on UCIT with LLaVA and InternVL. Module removals and crossed deployments separately examine how the system uses its two gates. Because the selected output is already computed for the decoder, the execution link also retains single-path deployment without evaluating the full projector bank. Our contributions can be summarized as follows:

- ELGAR connects instruction-conditioned projector selection to language LoRA routing through pooled selected-projector features, with a joint training objective and an explicit single-path deployment rule.
- We characterize observation sufficiency for restricted allocation and give a fixed-projector construction separating selected-projector and projector-ensemble mean features. This motivates the selected-output observation and the matched controls that test its practical value.
- Complete trajectories establish final-score and retention gains in four benchmark–backbone settings. Separate paired studies show repeatable 0.68- and 0.62-point advantages over bank-mean and projector-identity observations with the visual route fixed, testing the value of executed-output content.

2 Related Work

Continual learning and conditional expert composition. Continual learning retains earlier capabilities through regularization, replay, and modular adaptation (Wang et al., 2024; Kirkpatrick et al., 2017; Buzzega et al., 2020; Wang et al., 2022). LoRA provides compact task-specific updates that can be stored alongside a common backbone (Hu et al., 2021); CoIN and MCITlib establish multimodal instruction sequences, while CL-MoE and HiDe-LLaVA investigate expert-based and modality-specific retention (Chen et al., 2024a; Guo et al., 2025c; Huai et al., 2025; Guo et al., 2025a). Once specialist parameters are available, their use ranges from static consolidation to input-dependent composition. Model soups and task arithmetic combine weights or task vectors, Fisher-weighted merging and TIES account for importance or interference, and RegMean reconstructs expert responses (Wortsman et al., 2022; Ilharco et al., 2022; Matena & Raffel, 2022; Yadav et al., 2023; Jin et al., 2022). KnOTS aligns low-rank updates and AdaMerging learns combination coefficients (Stoica et al., 2025; Yang et al., 2024). Conditional approaches retain a role for the current request: Twin-Merging combines shared and task-specific information, while DAM dynamically merges adapters for continual video question answering (Lu et al., 2024; Cheng et al., 2024). This progression from storage to consolidation and allocation motivates treating retained expertise and its joint execution as separate design problems. ELGAR addresses the allocation setting with frozen experts and query-dependent language updates inside one backbone.

Coordinating visual and language experts. Multimodal expert systems must coordinate the representation entering the language model with the adaptations that process it. MVP learns instruction-grounded projector selection, PMA combines projector expansion with multimodal routing, and MAny mixes prototype-weighted projector outputs alongside static language consolidation (Jin et al., 2025; Zhang et al., 2026; Gao et al., 2026). REMEDY extends input-conditioned composition across projector and shallow-language recipes, predicting their weights from shared multimodal features (Zhu et al., 2025). LEO-MINI uses projected visual tokens, textual instructions, and current hidden states to route language experts within language layers (Wang et al., 2025). These systems differ in the components they compose, the granularity of allocation, and the representations available to each gate. ELGAR focuses on the dependency between two query-level decisions. An instruction-conditioned gate selects a task projector; that projector's output supplies both the decoder's visual tokens and the pooled observation used by the language allocator. Learned module coefficients then combine the retained language updates under this observation. The resulting execution graph makes the selected upstream computation explicit to the downstream decision, connecting visual selection to language allocation through the representation actually consumed by the decoder.

Routing objectives and decision information. A routing rule also depends on what evidence is available when the decision is made. RouteLLM learns between-model selection from quality preferences, whereas R2-T2 uses reference neighborhoods to revise multimodal expert allocation (Ong et al., 2025; Li et al., 2025). Decision-focused learning connects prediction objectives to downstream choices, and counterfactual routing analyses evaluate alternative experts under a fixed model (Elmachtoub & Grigas, 2022; Yoon et al., 2026). Together, these perspectives distinguish predicting a task label, estimating an expert's utility, and representing the information required to make a useful allocation. They also separate access to an informative observation from the ability of a chosen policy class and training procedure to exploit it. In ELGAR, the observation is available before answer generation: the instruction and pooled selected-projector output determine language allocation, which remains fixed through prefill and decoding. Its finite allocation analysis asks when a restricted observation admits an optimal action for every setting it leaves indistinguishable. This provides a formal basis for comparing executed-projector observations with fixed bank summaries, while matched feature controls assess the benefit realized by trained gates. The decision is made within one vision–language model, rather than by selecting among completed candidate answers.

3 Methodology

ELGAR links each allocation to the computation it controls. The visual gate uses the image and instruction to select a projector; the language gate uses that projector's actual output to decide which

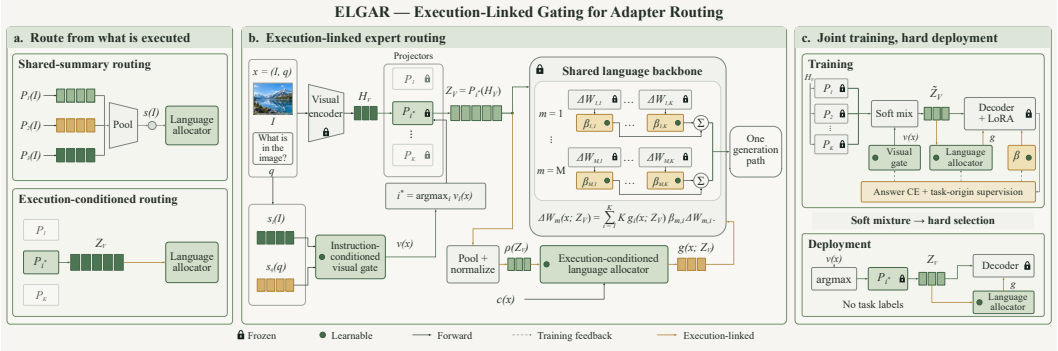


Figure 2: **ELGAR routes language experts using the visual output that the decoder receives. (a)** A shared image summary ignores which projector is selected, whereas the lower path routes from that projector's output. **(b)** For an image and instruction, the visual gate selects one frozen projector. Its tokens enter the decoder, and a pooled summary of those same tokens joins summaries of the raw image-patch and instruction embeddings in the language allocator. The allocator mixes frozen language-expert updates within one backbone generation path. **(c)** Training uses a soft projector mixture so both gates can learn from answers and task-origin supervision. Deployment selects one projector without reference answers or task labels. Locks mark frozen components, green dots mark learned parameters, and orange arrows mark the connection from selected visual output to language routing.

adaptations should process it. Figure 2 shows the architecture and distinguishes joint calibration from deployment.

3.1 Retained experts and routing inputs

Let $x = (I, q)$ be an image and its textual instruction. We retain K projectors P_i and K LoRA experts around frozen vision encoder E_V and language backbone W_0. At module m, expert i supplies $\Delta W_{m,i} = s_{m,i} B_{m,i} A_{m,i}$. Compatible experts share backbone, tokenizer and module shapes but may differ in rank. Keeping these experts frozen lets calibration focus on when and how to combine their responses. The gates need compact summaries of the image and the requested task. For vision patches H_V, classification token h_{CLS}, and instruction embeddings H_q, let n denote length normalization and define

$$c(x) = [n(\mathrm{mean}\, H_V); n(\mathrm{mean}\, H_q)], \qquad u(x) = [n(h_{\mathrm{CLS}}); n(\mathrm{mean}\, H_q)].$$

Neither summary contains answers or labels. These inputs describe the request, but not what its selected projector produces. The language gate therefore also observes that output, as defined below. Normalization and feature-layer details are in Appendix B.

3.2 Visual module: instruction-conditioned gate

Different instructions can require different uses of the same image, so the visual choice observes both modalities. A single affine routing head $r_\phi(u) = A_V u + b_V$, with $\phi = (A_V, b_V)$, predicts

$$v(x) = \mathrm{softmax}(r_\phi(u(x))), \qquad \widetilde{Z}_V(x) = \sum_{i=1}^{K} v_i(x) P_i(H_V).$$

Here $v_i(x)$ is the routing probability assigned to projector i. Training combines matching output tensors, not nonlinear projector parameters. At deployment, deterministic tie-breaking gives

$$i^*(x) = \arg\max_i v_i(x), \qquad Z_V(x) = P_{i^*(x)}(H_V).$$

The soft mixture lets answer supervision train the visual choice. Hard deployment then sends the same selected representation to both the decoder and language router, keeping allocation tied to the features actually processed.

3.3 LANGUAGE MODULE: PROJECTOR-CONDITIONED ROUTER

The language gate observes the visual tokens produced for the current request. To allocate language experts to those features, let Z denote the visual-token output tensor: $\widetilde{Z}_V$ during training and Z_V at deployment. Its normalized token mean, $\rho(Z) = n(\mathrm{mean}_{\mathrm{token}}\, Z)$, is the pooled feature vector observed by the language gate. The learned maps U and W_o, with bias b, act on the input and executed-output summaries:

$$a^L = Uc(x) + W_o\rho(Z) + b, \qquad g(x; Z) = \mathrm{softmax}(a^L).$$

This head shares query-level weights across modules, while learned query-independent coefficients $\beta_{m,i}$ specialize each module:

$$\Delta W_m(x; Z) = \sum_{i=1}^{K} K g_i(x; Z)\beta_{m,i}\Delta W_{m,i}. \tag{1}$$

Coefficients may be signed, so the effective update is not a convex mixture. Calibration starts from an equal expert mean rather than a preferred task. Appendix F gives the exact concatenated low-rank implementation. One gate evaluation per request fixes the mixture of retained LoRA updates throughout prefill and decoding along one backbone trajectory.

3.4 JOINT CALIBRATION AND HARD DEPLOYMENT

Let $\theta = (U, W_o, b)$ and let $\mathcal{D}_{\mathrm{src}}$ contain inputs, responses y and task origins t. With visual logits $a^V = r_\phi(u(x))$ and language logits a^L evaluated at $\widetilde{Z}_V$, calibration minimizes

$$\min_{\phi,\theta,\beta} \mathbb{E}_{(x,y,t)\sim\mathcal{D}_{\mathrm{src}}} \left[\ell_{\mathrm{answer}}(y \mid x; \widetilde{Z}_V, g) + \lambda_V \,\mathrm{CE}(a^V, t) + \lambda_L \,\mathrm{CE}(a^L, t)\right].$$

The answer loss trains combinations; task-origin cross-entropy, weighted by λ_V and λ_L, anchors the gates to available expertise. Joint fitting calibrates both choices against the answer objective, while the execution link specifies the language gate's observation. Deployment uses one selected output for both consumers without task labels. Appendix N gives the calibration data and training budgets.

Appendices E and G examine supervision and observation controls.

3.5 WHEN DO SELECTED-PROJECTOR FEATURES MATTER?

For finitely many execution settings with positive weights and finite candidate allocations, an observation permits an optimal policy exactly when each group it leaves indistinguishable shares an optimal choice. If one observation determines the other, the finer one cannot increase optimal risk (Appendix C.1).

A fixed nonlinear-projector example isolates a distinction that bank averaging can erase: with a fixed binary decoder and common target class, inputs with equal raw and bank-mean summaries yield selected outputs requiring different readout mixtures. A common-summary gate incurs cross-entropy at least $\log 2$, whereas an executed-output gate attains strictly less with finite logits (Appendix C.2). Nonlinearity is essential to this example: affine projectors make pooled outputs functions of the mean input.

4 EXPERIMENTS

4.1 EVALUATION PROTOCOL AND METRICS

We evaluate complete continual-learning trajectories on six-task UCIT and five-domain DCL with LLaVA-1.5-7B and InternVL-Chat-7B (Liu et al., 2024; Chen et al., 2024b; Guo et al., 2025c). Each UCIT task uses the specified 3,000-example population; DCL uses its complete native population. Final average score (FAA) averages final-stage task scores; continual average (CAA) averages stage means; backward transfer (BWT) averages each earlier task's change from acquisition to the final stage. All three are higher-is-better. Appendix B gives the datasets, exact formulas, scoring conventions and control protocols. Experiments and baseline reproductions use a shared pool of 32 NVIDIA A100 GPUs, with allocation varying by job.

4.2 BASELINES AND COMPARISON SCOPE

The benchmark tables compare complete learners under matched evaluation protocols; observation controls isolate the language gate's input. Source identifies each method. The UCIT–LLaVA extension rows include PASs-MoE, IterIS, RegMean++, and CSP, each evaluated on the same 3,000 examples per task.

Table 1: **UCIT performance on LLaVA-1.5-7B.** Complete local evaluation with 3,000 examples per task. IN-R, Arxiv, VizWiz, Icon, CLEVR and Flickr denote ImageNet-R, ArxivQA, VizWiz-caption, IconQA, CLEVR-Math and Flickr30k. Final average score (FAA) averages final task scores, continual average (CAA) averages stage means, and backward transfer (BWT) measures change since task acquisition; all are higher-is-better. Source identifies the method; FAA/CAA are percentages and BWT is in points. Dashes mark inapplicable CAA/BWT for unmerged rows. † marks the unmerged base, ‡ separate task experts, and blanks unavailable results.

Method	Source	IN-R	Arxiv	VizWiz	Icon	CLEVR	Flickr	FAA ↑	CAA ↑	BWT ↑
Zero-shot	(Guo et al., 2025c)	17.07	52.93	39.12	20.93	19.90	40.95	31.82†	—	—
Individual	(Guo et al., 2025c)	91.33	91.47	60.72	77.40	76.60	57.61	75.86‡	—	—
LoRA-FT	(Hu et al., 2021)	65.60	77.03	44.44	70.93	58.43	57.46	62.32	76.73	-16.57
O-LoRA	(Wang et al., 2023)	82.97	92.07	40.04	64.23	51.10	52.39	63.80	78.21	-7.77
MoELoRA	(Chen et al., 2024a)	61.50	77.57	43.77	70.77	59.83	57.32	61.79	76.90	-17.43
CL-MoE	(Huai et al., 2025)	61.60	77.77	44.22	68.63	54.37	57.65	60.71	75.86	-18.35
HiDE	(Guo et al., 2025a)	84.23	88.83	48.12	66.60	46.83	53.73	64.72	78.59	-7.14
DISCO	(Guo et al., 2025b)	87.53	92.63	46.62	71.23	60.90	56.23	69.19	81.39	-7.92
SiGMA	(Park & Kim, 2026)	82.43	91.13	54.47	70.30	58.50	56.02	68.81	80.61	-6.34
MAny	(Gao et al., 2026)	89.47	91.97	58.86	67.67	68.07	56.88	72.15	81.60	-1.52
PASs-MoE	(Hou et al., 2026)	76.27	78.67	45.41	65.83	52.63	57.19	62.67	76.93	-12.00
IterIS	(Chen et al., 2025)	69.90	87.23	43.23	65.47	63.13	49.45	63.07	75.44	-5.36
RegMean++	(Nguyen et al., 2025)	75.83	88.80	44.70	64.90	54.83	54.78	63.97	76.25	-4.73
CSP	(Sun et al.)	85.60	92.03	46.73	71.80	65.20	56.95	69.72	78.45	-1.22
ELGAR	Ours	90.70	92.10	62.97	74.00	77.53	56.50	75.63	83.56	-0.41

ELGAR reaches 75.63 FAA, 83.56 CAA, and -0.41 BWT, improving over the strongest other merged row by 3.48, 1.96, and 0.81 points, respectively. The complete-model gains include 9.46 points on CLEVR-Math. Appendix Table 5 tests component removals; the matched observation controls separately test which information helps the language allocator.

Table 2: **UCIT performance on InternVL-Chat-7B.** Complete local evaluation with 3,000 examples per task under the specified InternVL 1.0 interface. IN-R, Arxiv, VizWiz, Icon, CLEVR and Flickr denote ImageNet-R, ArxivQA, VizWiz-caption, IconQA, CLEVR-Math and Flickr30k. Final average score (FAA) averages final task scores, continual average (CAA) averages stage means, and backward transfer (BWT) measures change since task acquisition; all are higher-is-better. † marks the unmerged base and ‡ separate task experts. Source identifies the method; FAA/CAA are percentages and BWT is in points. Dashes mark inapplicable CAA/BWT for unmerged rows.

Method	Source	IN-R	Arxiv	VizWiz	Icon	CLEVR	Flickr	FAA ↑	CAA ↑	BWT ↑
Zero-shot	(Guo et al., 2025c)	20.80	62.80	40.83	28.33	20.13	46.19	36.51†	—	—
Individual	(Guo et al., 2025c)	94.70	93.83	62.08	75.30	61.30	58.92	74.36‡	—	—
LoRA-FT	(Hu et al., 2021)	70.23	78.17	45.54	71.07	74.40	58.32	66.29	79.08	-9.45
O-LoRA	(Wang et al., 2023)	78.47	92.63	44.33	62.13	65.73	54.68	66.33	78.25	-4.56
MoELoRA	(Chen et al., 2024a)	54.97	84.47	46.47	63.13	61.83	55.98	61.14	73.68	-10.05
CL-MoE	(Huai et al., 2025)	73.17	78.27	44.57	72.00	71.03	58.02	66.18	78.33	-9.54
HiDE	(Guo et al., 2025a)	86.47	94.17	53.34	69.07	71.17	56.07	71.71	81.86	-0.08
MAny	(Gao et al., 2026)	88.07	95.13	62.44	73.47	60.20	58.77	73.01	84.12	-1.69
ELGAR	Ours	94.77	94.80	64.67	71.80	75.87	56.71	76.44	85.37	0.07

On InternVL, ELGAR obtains 76.44 FAA, 85.37 CAA, and 0.07 BWT, exceeding the strongest other merged row by 3.43, 1.25, and 0.15 points. This transfers the complete pipeline without assuming that LLaVA projector features or weights remain unchanged.

Table 3: **DCL performance on LLaVA-1.5-7B.** Complete native evaluation populations. RS, Med, AD, Sci and Fin denote the remote-sensing, medical, autonomous-driving, science and finance domains. Final average score (FAA) averages final task scores, continual average (CAA) averages stage means, and backward transfer (BWT) measures change since task acquisition; all are higher-is-better. † marks the unmerged base and ‡ separate task experts. Zero-shot and O-LoRA use corrected native Fin endpoints. Source identifies the method; FAA/CAA are percentages and BWT is in points. Dashes mark inapplicable CAA/BWT for unmerged rows.

Method	Source	RS	Med	AD	Sci	Fin	FAA ↑	CAA ↑	BWT ↑
Zero-shot	(Guo et al., 2025c)	32.27	28.26	15.61	35.41	37.69	29.85†	—	—
Individual	(Guo et al., 2025c)	68.67	62.07	53.69	53.20	81.80	63.89‡	—	—
LoRA-FT	(Hu et al., 2021)	77.03	44.39	35.29	48.37	80.81	57.18	64.92	-10.03
O-LoRA	(Wang et al., 2023)	79.41	39.58	30.53	40.40	87.14	55.41	61.48	-8.71
MoELoRA	(Chen et al., 2024a)	73.92	42.50	29.22	47.19	79.44	54.45	64.29	-14.28
CL-MoE	(Huai et al., 2025)	77.90	46.94	36.17	48.30	80.91	58.04	66.16	-10.22
HiDE	(Guo et al., 2025a)	79.92	52.38	36.79	48.25	74.44	58.36	64.72	-3.80
DISCO	(Guo et al., 2025b)	72.00	44.50	52.97	51.61	82.80	60.78	63.27	-3.75
MAny	(Gao et al., 2026)	59.44	60.91	53.81	51.79	89.57	63.10	63.41	-3.79
ELGAR	Ours	69.11	62.18	50.49	53.03	82.87	63.54	64.18	-0.49

The native DCL matrix yields 63.54 FAA, 64.18 CAA, and -0.49 BWT: FAA and BWT exceed the strongest other merged rows by 0.44 and 3.26 points, while CAA is 1.98 points lower. Thus stronger final scores and backward transfer coexist with lower stage-mean performance in this setting.

Table 4: **DCL performance on InternVL-Chat-7B.** Complete native evaluation populations under the specified InternVL 1.0 interface. RS, Med, AD, Sci and Fin denote remote-sensing, medical, autonomous-driving, science and finance. Final average score (FAA) averages final task scores, continual average (CAA) averages stage means, and backward transfer (BWT) measures change since task acquisition; all are higher-is-better. † marks the unmerged base and ‡ separate task experts. Source identifies the method; FAA/CAA are percentages and BWT is in points. Dashes mark inapplicable CAA/BWT for unmerged rows. HiDE uses controlled task-free anchor top-1 inference.

Method	Source	RS	Med	AD	Sci	Fin	FAA ↑	CAA ↑	BWT ↑
Zero-shot	(Guo et al., 2025c)	32.05	35.90	14.39	34.90	42.83	32.01†	—	—
Individual	(Guo et al., 2025c)	73.50	66.64	54.25	53.52	82.49	66.08‡	—	—
LoRA-FT	(Hu et al., 2021)	68.85	46.86	33.72	42.26	86.65	55.67	63.66	-12.83
O-LoRA	(Wang et al., 2023)	76.91	28.38	28.95	31.99	67.10	46.67	55.05	-4.82
MoELoRA	(Chen et al., 2024a)	76.05	42.74	37.84	45.02	78.68	56.07	62.92	-8.96
CL-MoE	(Huai et al., 2025)	75.46	47.52	36.10	42.52	82.44	56.81	66.76	-13.07
HiDE	(Guo et al., 2025a)	80.36	59.23	51.20	23.91	53.97	53.73	64.34	-0.15
MAny	(Gao et al., 2026)	75.05	61.78	51.76	45.53	78.23	62.47	65.68	-0.65
ELGAR	Ours	77.59	66.03	52.76	52.73	82.27	66.28	67.70	0.66

ELGAR records 66.28 FAA, 67.70 CAA, and 0.66 BWT, gains of 3.81, 0.94, and 0.81 points over the strongest other merged rows. The complete lower-triangular matrix establishes stage-wise retention, while matched UCIT controls test the observation mechanism.

Figure 3 tests whether observing the visual features delivered to the decoder improves language allocation. With the same frozen visual route and expert bank, the three paired source-order repeats give 73.75 FAA for learned static allocation, 74.85 for request-conditioned bank-mean routing, and 75.53 for selected-output routing. Every arm receives 4,608 updates and a complete native six-task evaluation. The 1.10-point Static-to-Bank contrast identifies the benefit of dynamic allocation; the

additional 0.68-point Bank-to-Selected contrast identifies the benefit of observing the representation that the decoder actually processes. Both gains recur in every paired repeat, although task-level changes vary. This frozen-route family tests the observation choice more directly than the jointly trained feature control in Appendix L; it is distinct from the jointly trained ELGAR endpoint.

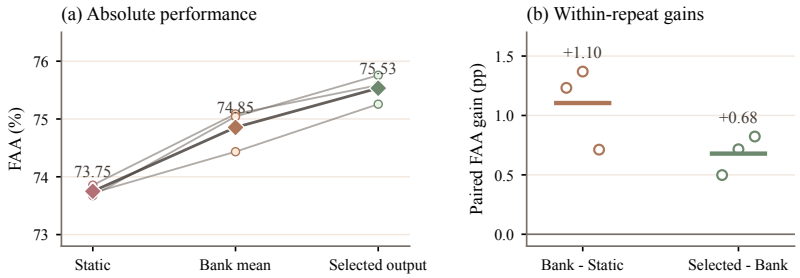


Figure 3: **The executed-output observation improves language allocation across three paired source-order repeats.** Static, Bank mean and Selected output allocators share a frozen visual route and expert bank, receive 4,608 updates, and use 3,000 examples per UCIT task on LLaVA-1.5-7B. **(a)** Thin lines with circles trace individual repeats; dark diamonds mark mean final average scores (FAA) of 73.75, 74.85 and 75.53. **(b)** Circles show paired differences and segments their means, not uncertainty intervals: +1.10 for Bank mean minus Static and +0.68 for Selected output minus Bank mean. Both are positive in every repeat; the latter isolates the executed-output observation.

Knowing which projector ran does not reveal what it produced for the current request. We therefore compare selected-output observations with projector identity while fixing the visual route and retaining the same raw-image and question summaries. In a separate three-pair study, output content improves FAA by 0.50, 0.67 and 0.68 points, averaging 0.62 points. The largest mean task gains are 1.85 on VizWiz-caption and 2.74 on IconQA. The output representation therefore adds repeatable aggregate value beyond branch identity (Appendix H).

The complete system leads the strongest other merged row in FAA and BWT across all four settings. Both UCIT backbones and DCL–InternVL also lead in CAA. DCL–LLaVA gains 0.44 FAA and 3.26 BWT points, with CAA 1.98 points lower. The radar profiles locate these aggregate patterns at the task level.

The paired study tests the observation used to train an allocator; the post-fit controls test whether hard deployment preserves its behavior. Across the four combinations of soft or selected decoder tokens and router features, FAA spans 0.16 points, with fully hard deployment highest (Appendix P). The resulting single-projector path preserves the fitted system's task performance without evaluating the full bank. Figure 4 addresses the next question: how does that fitted system distribute work between visual and language experts? Task-associated visual choices coexist with cross-expert language allocation, especially on the caption tasks.

These maps summarize different quantities: hard visual selection frequency and mean soft language-gate weight. Because each expert's effective language update also includes module coefficients and adapter weights, the maps describe allocation, not a decomposition of accuracy.

Caption tasks illustrate why the two assignments need not coincide. A task-associated visual branch can produce tokens processed by a mixture of caption-language experts. The execution link therefore couples the decisions through the selected representation without tying their expert indices. Observation controls measure the information's task value; the heatmaps reveal the resulting division of computation.

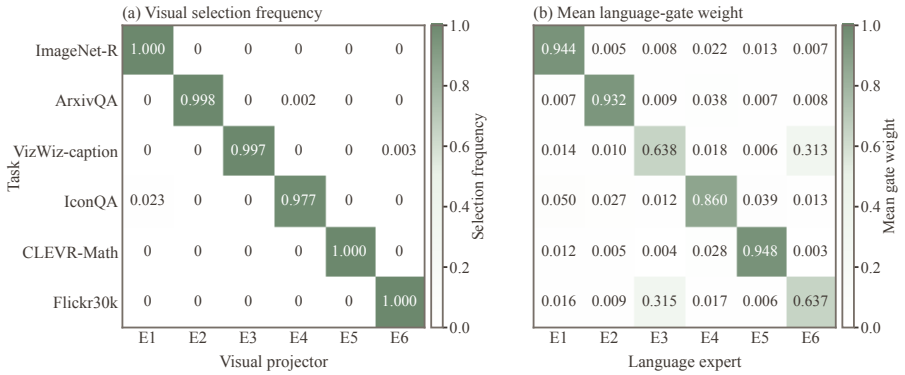


Figure 4: **Visual selection and language allocation have distinct roles in the fitted router.** Both panels summarize the same 18,000 UCIT queries from one fitted LLaVA-1.5-7B model. E1–E6 denote the ImageNet-R, ArxivQA, VizWiz-caption, IconQA, CLEVR-Math and Flickr30k experts. **(a)** Cells give hard visual selection frequency. **(b)** Cells give mean soft language-gate weight. Rows sum to one before rounding, with separate zero-to-one colorbars. Task labels group evaluation queries but are not gate inputs.

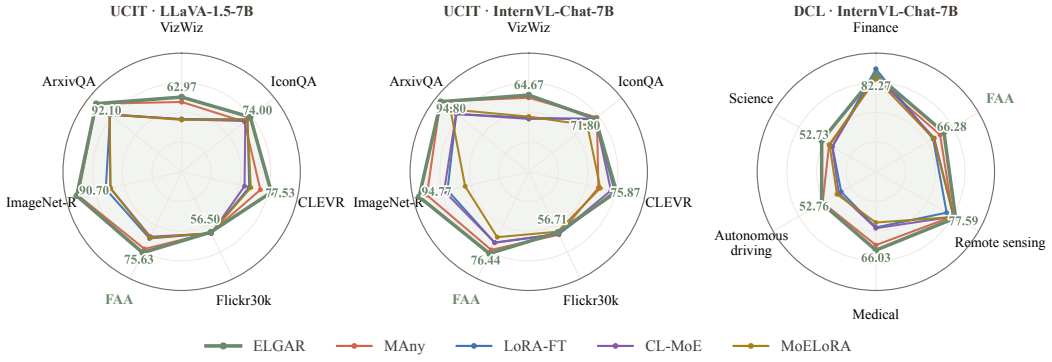


Figure 5: **Final-task profiles under three benchmark–backbone settings.** The panels show UCIT–LLaVA-1.5-7B, UCIT–InternVL-Chat-7B and DCL–InternVL-Chat-7B, the three settings with ELGAR's largest final average score (FAA) leads. Spokes use a common 0–100 scale. Lines compare ELGAR with locally evaluated MAny, LoRA-FT, CL-MoE and MoELoRA; labels give ELGAR's values. These are final endpoints; the tables establish stage-wise retention.

Figure 5 locates task gains across settings; matched training isolates the executed observation's value.

5 Conclusion

ELGAR makes the executed visual representation the language allocator's observation, with joint calibration and single-path deployment. Its finite-candidate analysis characterizes when an observation permits optimal allocation: indistinguishable requests must share an optimal choice. Matched controls improve on bank-mean routing in every UCIT repeat; a separate projector-identity comparison gains 0.62 FAA points on average. Jointly trained ELGAR reaches 75.63 on UCIT–LLaVA, with complete stage-wise evaluations across four settings. Together, these findings connect the observation's allocation value to task performance. Language experts are assigned to process the representation actually delivered for each image and question, so allocation follows the executed computation rather than a bank summary or branch identity.

REFERENCES

Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *NeurIPS*, 33:15920–15930, 2020.

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. *NeurIPS*, 37:57817–57840, 2024a.

Hongxu Chen, Zhen Wang, Runshi Li, Bowei Zhu, and Long Chen. Iteris: Iterative inference-solving alignment for lora merging. In *CVPR*, pp. 4829–4838. IEEE, 2025.

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 *CVPR*, pp. 24185–24198, 2024b.

Feng Cheng, Ziyang Wang, Yi-Lin Sung, Yan-Bo Lin, Mohit Bansal, and Gedas Bertasius. Dam: Dynamic adapter merging for continual video qa learning. *arXiv preprint arXiv:2403.08755*, 2024.

Samuel N Cohen and Eliana Fausti. Hyperbolic contractivity and the hilbert metric on probability measures. *arXiv preprint arXiv:2309.02413*, 2023.

Adam N Elmachtoub and Paul Grigas. Smart "predict, then optimize". *Management science*, 68(1): 9–26, 2022.

Zijian Gao, Wangwang Jia, Xingxing Zhang, Pengfei Qian, Tao Sun, Bo Ding, Yong Dou, Huaimin Wang, and Kele Xu. Many: Merge anything for multimodal continual instruction tuning. *arXiv preprint arXiv:2604.14016*, 2026.

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 *ACL*, pp. 13572–13586, 2025a.

Haiyang Guo, Fanhu Zeng, Fei Zhu, Wenzhuo Liu, Da-Han Wang, Jian Xu, Xu-Yao Zhang, and Cheng-Lin Liu. Federated continual instruction tuning. In *ICCV*, pp. 1325–1335. IEEE, 2025b.

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*, 2025c.

Danna Gurari, Yinan Zhao, Meng Zhang, and Nilavra Bhattacharya. Captioning images taken by people who are blind. In *ECCV*, pp. 417–434. Springer, 2020.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *ICCV*, pp. 8320–8329. IEEE, 2021.

Zhiyan Hou, Haiyun Guo, Haokai Ma, Yandu Sun, Yonghui Yang, and Jinqiao Wang. Pass-moe: Mitigating misaligned co-drift among router and experts via pathway activation subspaces for continual learning. In *ACL*, pp. 31959–31972, 2026.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

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 *CVPR*, pp. 19608–19617. IEEE, 2025.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*, 2022.

Hyundong Jin, Hyung Jin Chang, and Eunwoo Kim. Instruction-grounded visual projectors for continual learning of generative vision-language models. In *ICCV*, pp. 3466–3476. IEEE, 2025.

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. Dataless knowledge fusion by merging weights of language models. *arXiv preprint arXiv:2212.09849*, 2022.

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.

Lei Li, Yuqi Wang, Runxin Xu, Peiyi Wang, Xiachong Feng, Lingpeng Kong, and Qi Liu. Multimodal arxiv: A dataset for improving scientific comprehension of large vision-language models. In *ACL*, pp. 14369–14387, 2024.

Wenbing Li, Zikai Song, Hang Zhou, Junqing Yu, Yunyao Zhang, and Wei Yang. Lora-mixer: Coordinate modular lora experts through serial attention routing. In *ICLR*, volume 2026, pp. 14694–14716, 2026.

Zhongyang Li, Ziyue Li, and Tianyi Zhou. R2-t2: Re-routing in test-time for multimodal mixture-of-experts. *arXiv preprint arXiv:2502.20395*, 2025.

Adam Dahlgren Lindström and Savitha Sam Abraham. Clevr-math: A dataset for compositional language, visual and mathematical reasoning. *arXiv preprint arXiv:2208.05358*, 2022.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *NeurIPS*, 36:34892–34916, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *CVPR*, pp. 26286–26296. IEEE, 2024.

Pan Lu, Liang Qiu, Jiaqi Chen, Tony Xia, Yizhou Zhao, Wei Zhang, Zhou Yu, Xiaodan Liang, and Song-Chun Zhu. Iconqa: A new benchmark for abstract diagram understanding and visual language reasoning. *arXiv preprint arXiv:2110.13214*, 2021.

Zhenyi Lu, Chenghao Fan, Wei Wei, Xiaoye Qu, Dangyang Chen, and Yu Cheng. Twin-merging: Dynamic integration of modular expertise in model merging. *NeurIPS*, 37:78905–78935, 2024.

Michael S Matena and Colin Raffel. Merging models with fisher-weighted averaging. *NeurIPS*, 35: 17703–17716, 2022.

The-Hai Nguyen, Dang Huu-Tien, Takeshi Suzuki, and Le-Minh Nguyen. Regmean++: Enhancing effectiveness and generalization of regression mean for model merging. *arXiv preprint arXiv:2508.03121*, 2025.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E Gonzalez, Mohammed Kadous, and Ion Stoica. Routellm: Learning to route llms from preference data. In *ICLR*, volume 2025, pp. 34433–34448, 2025.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, pp. 311–318, 2002.

Keonhee Park and Gunhee Kim. Sigma: Sign-guided merging and adaptation for multimodal continual instruction tuning. *arXiv preprint arXiv:2607.20511*, 2026.

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

George Stoica, Pratik Ramesh, Boglarka Ecsedi, Leshem Choshen, and Judy Hoffman. Model merging with svd to tie the knots. In *ICLR*, volume 2025, pp. 4501–4519, 2025.

Wenju Sun, Qingyong Li, Tiancheng Li, Yangliao Geng, and Boyang Li. Revisiting the role of pretrained weights in model merging: On near-optimality within the core subspace. In *ICML*.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *ICCV*, pp. 4566–4575, 2015.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *TPAMI*, 46(8):5362–5383, 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 Association for Computational Linguistics: EMNLP 2023*, pp. 10658–10671, 2023.

Yimu Wang, Mozhgan Nasr Azadani, Sean Sedwards, and Krzysztof Czarnecki. Leo-mini: An efficient multimodal large language model using conditional token reduction and mixture of multimodal experts. In *EMNLP*, pp. 7257–7272, 2025.

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 *CVPR*, pp. 139–149. IEEE, 2022.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *ICML*, pp. 23965–23998. Pmlr, 2022.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. Ties-merging: Resolving interference when merging models. *NeurIPS*, 36:7093–7115, 2023.

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. Adamerging: Adaptive model merging for multi-task learning. In *ICLR*, volume 2024, pp. 22743–22763, 2024.

Youngsik Yoon, Siwei Wang, Wei Chen, and Jungseul Ok. When are experts misrouted? counterfactual routing analysis in mixture-of-experts language models. *arXiv preprint arXiv:2605.07260*, 2026.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *TACL*, 2: 67–78, 2014.

Duzhen Zhang, Yahan Yu, Qiaoyi Su, Jiahua Dong, and Tielin Zhang. Progressive multimodal alignment for continual instruction tuning. *arXiv preprint arXiv:2607.26947*, 2026.

Didi Zhu, Yibing Song, Tao Shen, Ziyu Zhao, Jinluan Yang, Min Zhang, and Chao Wu. Remedy: Recipe merging dynamics in large vision-language models. In *ICLR*, 2025.

AI USE STATEMENT

Generative AI tools assisted with manuscript language polishing and experimental code development and debugging. The authors take responsibility for the final manuscript and code.

The appendix separates the evidence for the complete router from the analysis of its observation. Appendix A details module and observation controls, and Appendix C.1 derives the allocation-value criterion. Appendices B and N specify evaluation and training conditions. The deployment analyses in Appendices O and P distinguish changes to visual tokens from changes to language allocation when a soft mixture becomes a hard choice.

A SUPPORTING EMPIRICAL EVIDENCE

Table 5 reports the four module-removal configurations on the six-task UCIT evaluation (3,000 examples per task), where the frozen full model obtains 75.63 FAA. All use the same frozen backbone and expert bank, and none uses prototype-based visual routing. Each removal cell is independently evaluated. The feature controls in Appendix L compare inputs to a retained language router, complementing the complete-module removals. The trainable removal variants use the same source data and update budget as the full method; removed parameters are fixed throughout training and deployment. The fully fixed cell requires no training. Writing s_{lv} for the FAA with language/visual modules (l, v), we measure the language effect $s_{11} - s_{01}$, the visual effect $s_{11} - s_{10}$, and the interaction $s_{11} - s_{10} - s_{01} + s_{00}$. All four cells share the evaluation protocol.

Table 5: **Both ELGAR routing modules contribute to final UCIT performance.** Each row reports final average score (FAA) on six UCIT tasks using LLaVA-1.5-7B and 3,000 examples per task. Language and Visual mark whether the corresponding learned router remains active. Visual removal uses the fixed mean projector-output tensor. Language removal fixes query weights and module coefficients to equal expert contributions. Each surviving trainable module is trained with the same source data and update budget as the full method. The fully fixed model needs no training. All scores use complete, independently scored evaluations.

Configuration	Language	Visual	FAA ↑
ELGAR	On	On	75.63
Remove language module	Off	On	62.18
Remove visual module	On	Off	69.97
Remove both modules	Off	Off	51.52

Both learned modules improve the complete system, but their effects are subadditive: the four scores give an interaction of -5.00 points. Thus the value of either module depends on whether the other is present; the interaction measures their joint contribution to the final score. The paired observation comparison in Figure 3 then asks what the retained language allocator should observe, separating dynamic allocation from the additional benefit of the executed visual output. Figure 4 describes how the fitted gates divide work through their observed expert allocations. Table 6 compares optimization paths under complete six-task UCIT evaluation. Keeping pooled selected-projector features while stopping their gradient into the visual router gives 75.64 FAA, compared with 75.63 for the full gradient path. The near-equal scores support retaining the executed-feature interface with either gradient path. Feature-removal controls instead test the information supplied to the language router. The table also compares the mixed-feature training objective with an expectation over selected-projector branches. Equal decoder pass counts do not imply equal optimizer updates or source exposure.

Table 6: **Training-path controls for ELGAR on UCIT–LLaVA.** Every reported score uses six tasks with 3,000 examples per task and hard visual selection at deployment. FAA is the final average score across tasks. Mixed-response training passes one soft visual mixture through the decoder, whereas exact branch expectation evaluates all six projector branches per update. Stopping the cross-module gradient retains the selected-projector feature vector. Exact branch expectation evaluates six decoder branches per update. Decoder passes count the training schedule and exclude qualification and evaluation. Matching source exposure does not match decoder computation between objectives. The mixed-response, six-epoch model remains the selected method; other rows are frozen controls.

Training configuration	Updates	Decoder passes	FAA ↑
ELGAR, mixed response, full gradient	4,608	4,608	75.63
Mixed response, stopped cross-module gradient	4,608	4,608	75.64
Exact branch expectation	768	4,608	72.55
Mixed response, one epoch	768	768	72.57
Exact branch expectation, six epochs	4,608	27,648	75.77

The mixed-response model is within 0.14 FAA points of six-epoch exact branch expectation while using one sixth as many training decoder passes (4,608 versus 27,648), a schedule-level reduction in decoder computation. The stopped-gradient control preserves essentially the same FAA while retaining the executed-feature interface.

B Evaluation Protocol Details

Routing summaries use length normalization $n(z) = z/\max(\|z\|_2, 10^{-12})$. Patch and instruction means are computed before normalization, as is the token mean of each projector output.

Evaluation design. Complete task evaluations measure system performance; module removals isolate each router's contribution, and matched feature and gradient controls test their connection. Additional backbones and task protocols assess transfer, with training and deployment costs reported separately. The four-cell matrix removes complete modules. Projector-ensemble mean-feature and no-projector-feature controls change the language router's input, whereas the detached-feature control retains that input but removes its cross-branch training gradient. A decoder-compute-matched training control is separate from controls matching optimization steps and source exposure. Complete evaluations populate the result tables. Appendix L gives the per-task observation comparison and paired-bootstrap protocol; runtime measurements are in Appendix M.

Backbones and datasets. The UCIT reproduction uses LLaVA-1.5-7B, following visual instruction tuning and its improved baseline (Liu et al., 2023; 2024), with a CLIP visual encoder (Radford et al., 2021). Its six tasks are ImageNet-R (Hendrycks et al., 2021), ArxivQA (Li et al., 2024), VizWiz-Captions (Gurari et al., 2020), IconQA (Lu et al., 2021), CLEVR-Math (Lindström & Abraham, 2022), and Flickr30k (Young et al., 2014). Each UCIT task uses the benchmark's specified 3,000 examples. Caption tasks use the benchmark's corpus-level scoring suite, including BLEU and CIDEr (Papineni et al., 2002; Vedantam et al., 2015), and its declared aggregation. Tables 1–4 separate UCIT and DCL by backbone. Every baseline row is independently evaluated with its original-method citation in Source; the four ELGAR rows each have a complete stage-wise evaluation under the corresponding native interface. LLaVA routing summaries use penultimate-layer patches; InternVL uses native layer -4 patches and its final classification token in both input summaries.

Metric definitions. For the score $R_{i,j}$ of task j after stage i in a T-task sequence, write $\bar{R}_i = i^{-1}\sum_{j=1}^{i} R_{i,j}$. We use

$$\mathrm{FAA} = \frac{1}{T}\sum_{j=1}^{T} R_{T,j}, \quad \mathrm{CAA} = \frac{1}{T}\sum_{i=1}^{T} \bar{R}_i, \quad \mathrm{BWT} = \frac{1}{T-1}\sum_{j=1}^{T-1}(R_{T,j} - R_{j,j}).$$

CAA averages stage means rather than all lower-triangular cells equally. BWT is in percentage points. Positive BWT means earlier tasks improve on average from acquisition to the final check-

point. Results are rounded to two decimals, while differences use unrounded scores. Missing stage evaluations remain unavailable.

C Observation Sufficiency and Allocation Value

C.1 Conditional allocation value and realized selection error

This analysis fixes one query and teacher prefix. It compares two visual interventions, Z_h and Z_s, with two existing language allocations, g_h and g_s. Using the same answer targets and loss normalization, define

$$L_{hh} = L(Z_h, g_h), \quad L_{hs} = L(Z_h, g_s), \quad L_{sh} = L(Z_s, g_h), \quad L_{ss} = L(Z_s, g_s).$$

These losses compare the same allocations under both visual representations; the candidates do not exhaust all possible language updates. Let $d_h = L_{hh} - L_{hs}$ and $d_s = L_{sh} - L_{ss}$, and assign analysis weights $p, 1-p$, with $0 < p < 1$, to the two interventions. Define the best allocation held fixed across interventions and the two-candidate, intervention-specific oracle by

$$R_{\text{fixed}} = \min\{pL_{hh} + (1-p)L_{sh},\ pL_{hs} + (1-p)L_{ss}\},$$
$$R_{\text{oracle}} = p\min\{L_{hh}, L_{hs}\} + (1-p)\min\{L_{sh}, L_{ss}\}.$$

Proposition 1 (Value of intervention-dependent allocation)**.** *Let Z_h and Z_s be two visual interventions, g_h and g_s two candidate language allocations, and $L_{ab} = L(Z_a, g_b)$ their answer losses for $a, b \in \{h, s\}$. Assign analysis weights p and $1-p$ to the two interventions, where $0 < p < 1$. Write $d_h = L_{hh} - L_{hs}$ and $d_s = L_{sh} - L_{ss}$ for the two allocation-loss margins; let R_{fixed} be the best risk from using one allocation for both interventions and R_{oracle} the risk from choosing separately. The difference $V = R_{\text{fixed}} - R_{\text{oracle}}$ is*

$$V = \tfrac{1}{2}\{p|d_h| + (1-p)|d_s| - |pd_h + (1-p)d_s|\}.$$

It is positive exactly when $d_h d_s < 0$. In that case, $V = \min\{p|d_h|, (1-p)|d_s|\}$.

Proof. Substitute $\min(a, b) = (a + b - |a - b|)/2$ and cancel common terms. The equality condition of the scalar triangle inequality gives the zero-value cases, and opposite signs yield the last expression. □

Preference reversal is therefore necessary, but its frequency alone does not quantify its value: the smaller weighted loss margin determines the cost of using one allocation.

Proposition 2 (Allocation sufficiency of an observation)**.** *For finite visual settings s, positive weights p_s and candidate allocations $a \in \mathcal{A}$, an observation map ψ groups settings into cells $C \in \mathcal{C}_\psi$. The best policy restricted to that observation has excess risk*

$$G_\psi = \sum_{C \in \mathcal{C}_\psi} \min_{a \in \mathcal{A}} \sum_{s \in C} p_s L(s, a) - \sum_s p_s \min_{a \in \mathcal{A}} L(s, a).$$

Then $G_\psi \geq 0$, with equality precisely when each cell has an allocation minimizing every positive-weight setting in that cell.

Proof. All individual excess losses are nonnegative, and their weighted sum vanishes only if each vanishes. □

The observation criterion concerns a specified summary and policy restriction. When visual execution is deterministic from the raw input, an unrestricted raw-input policy can recover it in principle. Giving its output directly to a lightweight gate changes the representation available to that gate.

Proposition 3 (Value of a finer observation)**.** *Let the finite setting set, positive setting weights p_s, candidate allocation set $\mathcal{A}$ and losses $L(s, a)$ be shared by two observations. The finer observation ψ_f refines the coarser observation ψ_c when $\psi_c = h \circ \psi_f$ for some map h. Writing $\mathcal{C}_f$ and $\mathcal{C}_c$ for their groups of indistinguishable settings, their optimal observation gaps satisfy*

$$G_{\psi_c} - G_{\psi_f} = \sum_{C \in \mathcal{C}_c} \left[\min_a \sum_{s \in C} p_s L(s, a) - \sum_{D \in \mathcal{C}_f :\, D \subseteq C} \min_a \sum_{s \in D} p_s L(s, a) \right] \geq 0.$$

Equality holds exactly when, within every coarse group, one allocation minimizes the aggregate loss of every constituent fine group.

Proof. Subtract the two observation-gap expressions in Proposition 2; the common full-information risk cancels. Each coarse group is a disjoint union of fine groups. The minimum of their summed losses is at least the sum of their minima. Because the allocation set is finite, equality holds exactly when those minima have a common optimizer. □

Thus refinement is valuable when some coarse group's constituent fine groups have no common optimal allocation. For example, if the request summary and executed-output observation determine the selected projector's identity, they refine the identity-only observation. This is a conditional comparison of optimal policies; the empirical comparison of fitted gates also depends on their parameterization and optimization.

Observation, policy class and fitted policy. For a common finite setting distribution, action set and answer loss, let $R_{\mathrm{oracle}} = \sum_s p_s \min_a L(s,a)$ and let R^*_ψ be the minimum risk of all ψ-measurable policies, as in Proposition 2. Let Π be the policy class restricted to that observation and taking actions in the *same* set, and let $\pi \in \Pi$ be its fitted member. Adding and subtracting R^*_ψ and $\inf_{\pi' \in \Pi} R(\pi')$ gives

$$R(\pi) - R_{\mathrm{oracle}} = \underbrace{R^*_\psi - R_{\mathrm{oracle}}}_{G_\psi} + \underbrace{\inf_{\pi' \in \Pi} R(\pi') - R^*_\psi}_{A_\Pi} + \underbrace{R(\pi) - \inf_{\pi' \in \Pi} R(\pi')}_{E_\pi}. \tag{2}$$

All three terms are nonnegative because the policy classes are nested. This decomposition uses one common loss population. The finite-cell theorem controls only G_ψ: a restricted class may have $A_\Pi > 0$, and a fitted policy may have $E_\pi > 0$. For the actual affine-softmax gate, the same accounting must instead use its continuous action set consistently. Population-level approximation and fitting errors are distinct from source-training loss. Near-collision tests additionally require a prespecified metric, tolerance and sensitivity restriction to connect observed feature similarity with the allocation problem.

Stability under action-dependent measurement error. For the finite observation-restricted value G_ψ above, let $e_s(a) = L(s,a) - \widehat{L}(s,a)$ and $\omega_s = \max_a e_s(a) - \min_a e_s(a)$. Then

$$|G_\psi(L) - G_\psi(\widehat{L})| \le \sum_s p_s \omega_s.$$

To prove this, subtract the action-independent quantity $\min_a e_s(a)$ from each row; such row constants cancel in G_ψ. The remaining errors lie in $[0, \omega_s]$. The observation-restricted optimum and the per-setting optimum each change by a number in $[0, \sum_s p_s \omega_s]$, which gives the bound after subtraction. Common rowwise loss offsets therefore do not alter this decision value. A uniform absolute error ϵ_s gives the weaker bound $2\sum_s p_s \epsilon_s$. Applying this conditional stability bound to measured decoder losses requires corresponding action-dependent error bounds.

Available value versus realized value. For any actual two-candidate policy π,

$$R_{\mathrm{fixed}} - R_\pi = V - \mathrm{Regret}(\pi),$$

where regret sums $p|d_h|$ and $(1-p)|d_s|$ on the respective incorrect choices. This follows by subtracting the per-setting minimum. For the native diagonal policy its value is

$$\mathrm{Regret}_{\mathrm{diag}} = p\max(d_h, 0) + (1-p)\max(-d_s, 0).$$

Thus a representation can permit useful allocation changes without a learned gate realizing that opportunity. If candidate-loss estimates have uniform error at most ϵ_s in each setting, their minimizer has excess loss at most $2\epsilon_s$, and is correct whenever the true two-candidate margin exceeds $2\epsilon_s$. Add and subtract the estimated losses of the chosen and optimal actions to prove these statements. Consequently $R_{\mathrm{fixed}} - R_{\mathrm{learned}} \ge V - 2\sum_s p_s \epsilon_s$, conditional on uniform candidate-loss error bounds over the evaluated setting population.

Connecting value to effective changes. For this bound only, write $z_h = \mathrm{vec}(Z_h)$ and $z_s = \mathrm{vec}(Z_s)$. Stack the physical, native-scaled language updates over adapted modules as $\mathcal{U}(g)$: with $U_{mi} = K\beta_{m,i}s_{m,i}B_{m,i}A_{m,i}$ and $(G_W)_{ij} = \sum_m \langle U_{mi}, U_{mj}\rangle_F$, we have $\|\mathcal{U}(g_h) - \mathcal{U}(g_s)\|_2^2 = (g_h - g_s)^\top G_W (g_h - g_s)$. Equation 3 derives this physical Gram identity without materializing dense updates. Assume the fixed-prefix loss $L(z, \mathcal{U})$ is twice continuously differentiable on a neighborhood of the product of the two line segments joining these visual and language endpoints, and that its mixed Hessian satisfies $\|\nabla^2_{z,\mathcal{U}} L\|_{\mathrm{op}} \le M$ throughout that product, in the stated Euclidean coordinates. Let $B = M\|z_h - z_s\|_2\sqrt{(g_h - g_s)^\top G_W (g_h - g_s)}$. Integrating the mixed derivative over the product of the segments and applying the operator-norm bound gives $|d_h - d_s| \le B$. Therefore $|d_h| > B$ rules out a reversal. When signs differ, $|d_h| + |d_s| = |d_h - d_s| \le B$, and

$$V \le p(1-p)B.$$

Indeed $\min(pa, (1-p)b) \le p(1-p)(a+b)$ for nonnegative a, b. This conditional upper bound links available allocation value to visual and language changes through the decoder's mixed curvature.

C.2 A FIXED-PROJECTOR CONSTRUCTION FOR OBSERVATION LOSS

This analytical construction isolates a restriction on the router's observation. The projector-ensemble mean-feature control averages frozen projector-output tensors before normalization, whereas the primary method observes the selected projector output. Both also receive raw patch-mean and question summaries. Consider two patch sets $X_s = \{(1, -s), (1, s)\}$ for $s \in \{1, 2\}$, with equal raw means and fixed question and CLS inputs. Fix the hard visual choice to the first of two projectors.

Let $f_{\mathrm{even}}(x) = \mathrm{GELU}(x) + \mathrm{GELU}(-x)$, $b = (f_{\mathrm{even}}(1) + f_{\mathrm{even}}(2))/2$ and $d = (f_{\mathrm{even}}(2) - f_{\mathrm{even}}(1))/2 > 0$. The fixed maps

$$f_+(1, x) = (1, f_{\mathrm{even}}(x) - b), \qquad f_-(1, x) = (1, b - f_{\mathrm{even}}(x))$$

are realizable by two-layer GELU projectors with biases: their hidden coordinates are $x, -x$, followed by a sum or negative sum. The normalized bank mean is $(1, 0)$ for both inputs, whereas the normalized selected means are $(1, -d)/\sqrt{1 + d^2}$ and $(1, d)/\sqrt{1 + d^2}$. Thus the selected projector output distinguishes these inputs even though the raw and bank-mean observations do not.

For a fixed binary decoder receiving the normalized pooled selected output z and a common gold class, let its logit margin be az_2, where $a \in [-1, 1]$ is the simplex-weight difference of two opposite rank-one language readout updates. Write $m = d/\sqrt{1 + d^2}$ and $\ell(v) = \log(1 + e^{-v})$. Any allocator seeing identical summaries must choose the same a on both inputs, and hence has equal-weight risk

$$\tfrac{1}{2}[\ell(am) + \ell(-am)] \ge \log 2.$$

This follows from $\ell(v) + \ell(-v) = \log(2 + 2\cosh v)$, with equality at $a = 0$. Input-independent randomization cannot improve this bound. In contrast, finite gate logits $(\kappa z_2, -\kappa z_2)$ yield $a = \tanh(\kappa z_2)$ and risk

$$\ell\big(m\tanh(\kappa m)\big) < \log 2 \qquad (\kappa > 0).$$

The separation therefore follows from fixed nonlinear projectors and ordinary cross-entropy at the projector/gate interface. Its equal-summary patch inputs define the analytical setting. Routers given complete patches, selected projector outputs or sufficiently informative hidden states may recover the same decision, as is relevant to visually conditioned and token-level routing (Wang et al., 2025; Li et al., 2026). Replacing the projectors with affine maps removes this example because mean projector outputs then depend only on mean inputs. The construction identifies an observation distinction at the projector–gate interface. The matched source-order study tests the task-level value of the observation on real inputs.

D CROSS-DOMAIN TRANSFER WITHOUT TARGET-DOMAIN TRAINING

Cross-domain transfer of the frozen model. Table 7 evaluates the UCIT-constructed model without target-domain training or model selection. It is separate from the native continual-learning results in Table 3.

Table 7: **Direct transfer to the DCL test domains.** The selected ELGAR UCIT–LLaVA model with hard visual selection remains frozen. Each domain uses its complete native test population and original question text. Accuracy uses the native domain scorer; greedy decoding retains the source-model limit of 256 generated tokens. All 47,469 predictions pass ordered-population and scorer-identity checks. These are transfer accuracies, not continual-learning trajectory metrics.

Test domain	Examples	Accuracy ↑
Remote sensing	10,000	41.02
Medical	9,790	36.74
Autonomous driving	10,000	15.10
Science	7,687	38.23
Finance	9,992	42.38

E Task-Origin Supervision Sensitivity

Figure 6 varies the task-origin loss used to fit the routers. Removing this supervision lowers six-task final average score (FAA) from 75.58 at the prespecified weight of 1 to 72.31. The positive weights score between 75.34 and 75.90, with one independent fit per weight. The weight-1 refit differs from the main-table model's 75.63 FAA by 0.05 points. This training-weight study is distinct from the paired observation comparison in the main text.

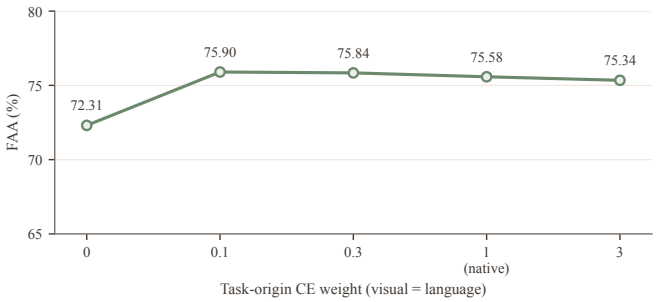


Figure 6: **Task-origin supervision changes the final UCIT score.** The visual and language task-origin cross-entropy weights vary together over 0, 0.1, 0.3, 1 and 3 while the answer-loss weight stays fixed. Each point is one independently fitted 4,608-update model scored with the native LLaVA-1.5-7B endpoint on 3,000 examples per UCIT task. Final average score (FAA) is the six-task mean; higher is better. FAA is 72.31 without task-origin supervision, 75.90 at weight 0.1, and 75.58 at the native weight 1. The line connects measured settings only; no across-fit error bars are available.

F Routing Implementation and Artifact Scope

The visual and language allocation maps are presented in the main text in Figure 4. The following implementation identities specify how the same fitted gate enters each language module.

Static composition and exact low-rank execution. Setting both input weights $U = W_o = 0$ gives a constant gate; uniform routing recovers $\sum_i \beta_{m,i} \Delta W_{m,i}$, nesting the static family. Let A_m vertically concatenate $A_{m,i}$ and B_m horizontally concatenate $B_{m,i}$. For block-diagonal $G_m(x; Z)$ with blocks $K g_i(x; Z) \beta_{m,i} s_{m,i} I_{r_i}$, block multiplication gives $B_m G_m(x; Z) A_m = \Delta W_m(x; Z)$. The block construction implements the summed updates exactly in real arithmetic; native numerical parity is checked separately. Constant-router and within-task gate-permutation controls distinguish query-specific allocation from fixed composition and task-level marginal allocation.

G Scope of the Matched Routing Controls

Figure 3 compares a trained static allocation with two query-conditioned allocators under a frozen visual route. The static arm is fitted directly rather than obtained by zeroing a trained query router. Figure 1 summarizes this same comparison. Appendix L reports the joint visual–language feature comparison. Both families retain task-origin supervision and the fitted within-task allocations, isolating observation choice under those shared training conditions.

H Projector Identity and Executed-Output Content

To test whether the language allocator benefits from output content beyond knowing the selected branch, we hold the visual route and expert bank fixed. Both allocators retain the same raw-image and question summaries. One additionally observes the normalized selected-projector output; the other observes its one-hot identity, zero-padded to the same input width. Paired fits share initialization, source examples and update budget, with three repeats varying source order. Each final state is evaluated on all six UCIT tasks with 3,000 examples per task under the same native LLaVA protocol. This comparison is independent of the Static/Bank/Selected study in Figure 3.

Selected-output and projector-identity FAA are respectively 75.59 and 75.09 in the first pair, 75.76 and 75.09 in the second, and 75.26 and 74.57 in the third. The paired gains are 0.50, 0.67 and 0.68 points, with mean 0.62 and sample standard deviation 0.10. Averaged over the three pairs, task differences are +0.12 on ImageNet-R, -0.77 on ArxivQA, +1.85 on VizWiz, +2.74 on IconQA, -0.06 on CLEVR-Math and -0.18 on Flickr30k. The consistent aggregate advantage, with task-dependent effects, supports using the executed representation to guide allocation rather than substituting its branch label.

I Full-Population Evaluation Protocol

Development and formal evaluation remain distinct populations. The main module-removal matrix fixes the expert bank, source identities, tokenizer, prompt, decoding and official full-population scoring. Trainable cells retain the declared six-epoch source schedule; the no-module cell has no optimizer. Removing a module also removes its associated auxiliary loss, and training cost is reported rather than assumed equal. The full model uses hard visual selection at deployment and a representation-conditioned language gate. Each of the six formal tasks contains 3,000 ordered predictions. Caption tasks use five references per image. QA uses uppercase whole-string matching without whitespace stripping or option extraction.

J Baseline Evaluation Details

The InternVL–DCL HiDE evaluation uses task-free anchor top-1 selection of the matching language expert in each layer and its trained projector, rather than the public inference code. Its FAA is 53.73 under this configuration. MAny uses FP32 for its core solve. Each comparison uses the same backbone, benchmark population and scoring protocol. Zero-shot evaluates the base model, whereas Individual evaluates a separate expert for each task; neither is a merged-model baseline.

K Metric Aggregation

The final average score (FAA) weights final-stage tasks equally. The continual average (CAA) weights stages equally after averaging the tasks at each stage. Backward transfer (BWT) averages acquisition-to-final changes over the $T-1$ earlier tasks. These quantities use unrounded scores from complete task evaluations.

Zero-shot and Individual means are reference summaries, not sequential acquisition-stage scores. They appear in FAA with † and ‡, respectively, in DCL; Individual additionally denotes separate experts rather than one final model. No missing CAA or BWT is inferred from final endpoints.

Table 8: **Selected-projector features improve the UCIT observation control.** Each historical arm has 3,000 evaluation examples per task, or 18,000 in total, with independently accepted scoring. Columns I-R, Arx, VZ, IQ, CM, and F30 mean ImageNet-R, ArxivQA, VizWiz-caption, IconQA, CLEVR-Math, and Flickr30k. The two captioning tasks use their native seven-metric corpus scores rather than classification accuracy. FAA is the final average score across the six tasks. All entries are point estimates.

Language-router features	I-R	Arx	VZ	IQ	CM	F30	FAA
No projector features	90.67	92.40	61.48	72.70	78.10	56.25	75.27
Projector-ensemble mean features	90.67	92.30	62.43	72.43	78.03	56.42	75.38
Pooled selected-projector features	90.70	92.10	62.97	74.00	77.53	56.50	75.63

Table 9: **Local inference time for the ELGAR module combinations.** Matched reference uses neither learned routing intervention, Language only or Visual only enables one, and Joint enables both. Each row times 96 development requests twice on the same GPU after warm-up, including the native generation path and a verification forward pass but excluding cold loading. Mean and Median are milliseconds per request, Tokens is mean generated length, and Peak is allocated GPU memory in GiB. The panel is not a new accuracy evaluation or a production-serving latency estimate.

Intervention	Mean (ms)	Median (ms)	Tokens	Peak (GiB)
Matched reference	450.18	320.33	6.17	15.28
Language only	462.35	324.20	6.47	15.28
Visual only	472.47	325.96	6.57	15.05
Joint	491.88	327.41	6.78	15.05

L Full-Population Observation Controls

Table 8 tests which pooled visual features are supplied to the language router. The training source, six-epoch budget, expert bank, evaluation population and scoring convention are matched. The no-projector-feature arm removes that feature input; the projector-ensemble mean-feature arm instead uses the same-width normalized pooled vector of the mean of all six frozen projector-output tensors. The selected-projector-feature arm uses the pooled vector from its selected visual expert's output tensor. Projector-ensemble computation differs because all six projectors are evaluated; the comparison matches the training budget and evaluation population while varying the observation.

Pooled selected-projector features increase FAA by 0.37 points over no projector features and 0.25 over projector-ensemble mean features. They improve four task scores and lower two relative to each control. The largest positive task difference is on VizWiz-caption versus no projector features and on IconQA versus projector-ensemble mean features. For pooled selected-projector versus projector-ensemble mean features, 2,000 task-stratified paired row-bootstrap replicates give a 95% percentile CI of $[0.03, 0.46]$ points for the FAA difference. Each replicate resamples the same row indices for both methods and recomputes all seven caption metrics on the resampled corpus, including repeated occurrences. The interval excludes zero for this frozen comparison, with the task-dependent effects reported above. The percentile interval summarizes row-resampling variation conditional on the two fitted models and observed benchmark. Its coverage is approximate, and its scope is the selected-versus-ensemble comparison. These development-selected checkpoints are evaluated on the reused UCIT population.

M Matched Native Runtime Controls

Table 9 measures the four module interventions on the same GPU. Each of two reverse-order rounds uses the first 16 development requests from every task, after 12 warm-up requests per intervention. All timed outputs reproduce the corresponding frozen answer text and token count. The 768 timings cover the native generation path and its parity forward after loading, with two measurements of the same 96 requests per intervention. Joint execution takes 9.26% more total measured time than the matched reference in this panel; the median paired request-time ratio is 1.02. The timing

difference includes both routing and changed generation lengths. The panel covers the four module combinations listed in the table.

N Evaluation and Numerical Protocol

Formal UCIT evaluation uses 3,000 ordered examples for each of ImageNet-R, ArxivQA, VizWiz-caption, IconQA, CLEVR-Math, and Flickr30k. Reported FAA is the equal-task mean of the six complete task scores. Caption scores use the official seven-metric corpus aggregation rather than an average of per-example correctness. ArxivQA uses uppercase whole-string comparison, without whitespace stripping or option-letter extraction.

Training and paired-repeat settings. Calibration uses AdamW with batch size one. Zero gate logits and module coefficients $\beta_{m,i} = 1/K$ initialize the equal mean of the K native-scaled experts. Answer loss and unit-weight task-origin losses train the gates and module coefficients while the backbone remains frozen. For six-task UCIT, 128 source examples per task and six passes give 4,608 updates; five-domain DCL uses the same per-task budget and 3,840 updates. Each stage-prefix fit uses its available experts and source examples.

The frozen-visual Static/Bank/Selected comparison shares the visual gate, initialization, source population, optimization settings and update budget across its three arms. Three paired repeats vary the source order while holding it identical within each repeat. Every arm uses its fixed-budget final state rather than a checkpoint selected on evaluation scores.

InternVL uses its native vision tower and projector, with direct RGB preprocessing for UCIT and the configured square padding for DCL. Learned gates and module coefficients use FP32; frozen vision and projector computations use their native FP16 execution.

Numerical conditions for controlled comparisons. Backbone execution, learned gates and stored adapters can use different precision. Matched controls preserve these settings together with the source data, initialization, objective, routing and scorer, so a change in the observation is not confounded with a numerical change.

Data roles. The development pool is sampled from UCIT training files and is training-exposed for baselines fitted on those complete files. A path-and-question audit finds no identical image–question pair between those files and the six evaluation sets, although 1,280 of 3,000 ArxivQA questions share an image path with training. This establishes question-level separation, with image-level and semantic deduplication outside the audit. Development and full-test scores use separate populations; full-test outcomes do not retroactively select development configurations.

O Visual Hardening and the Language Gate

For a fixed image and question, let $Z_i \in \mathbb{R}^{N\times H}$ be expert i's visual tokens, p the visual routing simplex, $\bar{Z} = \sum_i p_i Z_i$, and $k = \arg\max_i p_i$ with a fixed tie rule. Write $r_i = N^{-1}\sum_n Z_{i,n}$ and $n(r) = r/\max(\|r\|_2, \epsilon)$ for the implemented normalization. The language gate is $g(r) = \mathrm{softmax}(b + W_o n(r))$, where the image/question contribution b is unchanged between the two executions. Hardening changes both the tokens and the language allocation.

Two computable perturbations. The visual difference has the exact decomposition and bounds

$$Z_k - \bar{Z} = \sum_{i\neq k} p_i (Z_k - Z_i),$$

$$\|Z_k - \bar{Z}\|_F \le \sum_{i\neq k} p_i \|Z_k - Z_i\|_F,$$

$$\|r_k - \bar{r}\|_2 \le \frac{\|Z_k - \bar{Z}\|_F}{\sqrt{N}}.$$

The first inequality is the triangle inequality and the second follows from Cauchy–Schwarz. Define $\delta = W_o[n(r_k) - n(\bar{r})]$ and $R = \max_j \delta_j - \min_j \delta_j$. Then

$$\mathrm{TV}(g(r_k), g(\bar{r})) \le \tanh(R/4), \qquad \mathrm{TV}(u, v) = \tfrac{1}{2}\|u - v\|_1.$$

This is an application of the Hilbert-metric bound of Cohen & Fausti (2023, Theorem 5.1). The log partition functions contribute a common shift, so the Hilbert distance between these positive softmax vectors is exactly R. Their full-ℓ_1 convention accounts for the factor of two. For completeness, a finite-dimensional derivation sets $X_i = g(r_k)_i / g(\bar{r})_i$ under probabilities $g(\bar{r})_i$. Then $\mathbb{E}X = 1$ and $\max_i X_i / \min_i X_i \le e^R$. If $a = \min_i X_i \le 1 \le b = \max_i X_i$, convexity of $|x - 1|$ bounds its expectation by the secant through $(a, 1-a)$ and $(b, b-1)$, yielding $\mathrm{TV} \le (b-1)(1-a)/(b-a)$. This bound increases with b for fixed a; setting $b = e^R a$ and maximizing over $a \in [e^{-R}, 1]$ gives $a = e^{-R/2}$ and $(e^{R/2} - 1)/(e^{R/2} + 1) = \tanh(R/4)$. This derivation fixes the total-variation convention used here.

Normalization also satisfies

$$\|n(a) - n(b)\|_2 \le \min\left(2, \frac{2\|a - b\|_2}{\max(\|a\|_2, \|b\|_2, \epsilon)}\right).$$

To see this, take $A = \max(\|a\|_2, \epsilon) \ge B = \max(\|b\|_2, \epsilon)$ and expand $a/A - b/B = (a - b)/A + b(1/A - 1/B)$. Use $\|b\|_2 \le B$ and $A - B \le \|a - b\|_2$; the bound by two follows because both normalized vectors lie in the unit ball.

Interpretation for coordinated routing. Visual confidence alone does not control either perturbation: an expert with small routing mass can have a large output difference. Conversely, equal pooled observations can hide different token tensors. The two diagnostics therefore address different parts of the deployed method. The learned decoder coefficients remain signed. For module m, define $U_{mj} = K\beta_{m,j} s_{mj} B_{mj} A_{mj}$, with $K = 6$ for the present bank. The induced weight difference satisfies $\|\sum_j (g_{k,j} - g_{\mathrm{mix},j}) U_{mj}\|_2 \le \sum_j |g_{k,j} - g_{\mathrm{mix},j}| \|U_{mj}\|_2$. A simplex gate does not make these effective weight updates convex. These bounds quantify local token, gate and weight changes at a fixed input.

An exact metric for the represented language update. The signed coefficients and original adapter scales can be retained without materializing dense updates. Define the $K \times K$ Gram matrix

$$(G_W)_{ij} = \sum_m \langle U_{mi}, U_{mj} \rangle_F, \qquad d = g_k - g_{\mathrm{mix}}.$$

Then

$$\sum_m \left\| \sum_j d_j U_{mj} \right\|_F^2 = d^\top G_W d. \tag{3}$$

Indeed, stacking $\mathrm{vec}(U_{mj})$ over modules as column j of a matrix T gives $G_W = T^\top T$. Consequently the represented updates are identical at every adapted module exactly when $d \in \ker G_W$. Since $\mathbf{1}^\top d = 0$, only the restriction of G_W to the simplex tangent space matters. This standard Gram construction accounts for the particular operators executed by the model; a probability distance alone omits their magnitudes and cancellation. The identity concerns real arithmetic on stored factors; intermediate gain rounding is a separate runtime effect. For unequal adapter ranks, its entries can be obtained from

$$\langle B_i A_i, B_j A_j \rangle_F = \sum_{a,b} (B_i^\top B_j)_{ab} (A_i A_j^\top)_{ab}.$$

A conditional bridge to hard deployment. Write $Z_s = \bar{Z}$, $Z_h = Z_k$, and let g_s, g_h be the language allocations computed from their respective pooled observations. Stack the U_{mi} over modules into physical update atoms U_i, and write $U(g) = \sum_i g_i U_i$. Native LoRA scale, the factor K and the possibly signed $\beta_{m,i}$ are all included in these atoms. For $d = g_h - g_s$, Equation 3 gives the *exact* represented update displacement $\|U(g_h) - U(g_s)\|_2 = \sqrt{d^\top G_W d}$.

Proposition 4 (Two perturbations at the executed interface)**.** *Fix an input and answer prefix. Let visual routing probabilities p_i select projector k and define hard tokens $Z_h = Z_k$ and soft tokens $Z_s = \sum_i p_i Z_i$. Their pooled features r_k and $\bar{r}$ produce language allocations g_h and g_s; write $d = g_h - g_s$. Let U_i be the physical update from expert i, including its adapter and routing scales. Write $U(g) = \sum_i g_i U_i$ for an allocation g and let $(G_W)_{ij} = \langle U_i, U_j \rangle$ be the Gram matrix of these stacked updates across adapted modules. Define the update diameter $D_U = \max_{i,j} \|U_i - U_j\|_2$ and the language-gate logit oscillation $R = \mathrm{osc}(W_o[n(r_k) - n(\bar{r})])$, where n normalizes pooled features, W_o maps them to logits, and $\mathrm{osc}(v) = \max_j v_j - \min_j v_j$. Suppose answer loss $L(Z, U)$ is L_Z-Lipschitz in visual tokens and L_U-Lipschitz in physical language updates along the two relevant line segments. Then*

$$\begin{aligned} |L(Z_h, U(g_h)) - L(Z_s, U(g_s))| &\le L_Z \|Z_h - Z_s\|_F + L_U \sqrt{d^\top G_W d} \\ &\le L_Z \sum_i p_i \|Z_k - Z_i\|_F + L_U D_U \tanh(R/4), \end{aligned} \tag{4}$$

The first term measures the visual-token change; the second measures the corresponding change in the represented language update.

Proof. Insert $L(Z_s, U(g_h))$ between the two losses and apply the two Lipschitz assumptions. The first term is bounded by the visual mixture identity above. For the second, d sums to zero; its positive and negative parts each have mass $\mathrm{TV}(g_h, g_s)$. Their normalized update combinations are convex combinations of the *scaled* atoms, so their distance is at most D_U. The preceding softmax stability bound gives $\mathrm{TV}(g_h, g_s) \le \tanh(R/4)$. □

Under the stated Lipschitz assumptions, the bound controls fixed-prefix loss changes through visual-token distance, logit oscillation and Gram-weighted update distance. It bounds the magnitude of the hardening effect.

P Consistency of Soft Training and Hard Deployment

With the trained model, input, answer prefix, target alignment and loss normalization fixed, let Z_h and Z_s denote selected-projector tokens and the soft mixture. Their normalized pooled observations produce language allocations g_h and g_s. Write $L_{ab} = L_x(Z_a, g_b)$ for $a, b \in \{h, s\}$. The diagonal cells are native execution paths; the off-diagonal cells exchange token and router-feature choices independently, without refitting. Define

$$O_h = L_{hh} - L_{hs}, \qquad O_s = L_{sh} - L_{ss}, \qquad T_s = L_{hs} - L_{ss}.$$

The hard-versus-soft difference is exactly $O_h + T_s$, and $I = O_h - O_s$ measures how the allocation effect depends on token choice. A negative O_h means that, for selected-projector tokens, recomputing the allocation from their observation gives lower loss than retaining the soft-mixture allocation. Both routers remain fixed throughout these interventions.

Table 10: **The selected visual branch supplies both decoder tokens and router features.** The four UCIT–LLaVA rows use identical ELGAR weights and 3,000 examples per task across six tasks. FAA is their final average score. Selected-projector output tensors and soft output mixtures are varied separately as the decoder input and the source of the language router's pooled feature vector. These deployment interventions are distinct from module-removal training.

Decoder visual-token input	Language-router features	FAA ↑
Selected branch	Selected branch	75.63
Selected branch	Soft mixture	75.62
Soft mixture	Selected branch	75.51
Soft mixture	Soft mixture	75.47

Table 10 reports the four corresponding deployment endpoints. Their FAA spans 0.16 points, with fully hard deployment highest. At fixed decoder tokens, exchanging router features changes FAA by 0.01 or 0.04 points. This close agreement assesses deployment consistency for the fitted gates; Figure 3 instead tests which observation produces a better allocator when trained with it.