WHICH ATTENTION HEADS IN VLMS SHOULD WE ADAPT AT TEST TIME?

Anonymous authors
Paper under double-blind review

ABSTRACT

Training vision–language models (VLMs) at test time has emerged as a promising approach to improving visual understanding and mitigating prediction failures during inference. However, existing methods largely predefine the model components to be adapted, risking inefficient parameter updates and unnecessary model drift, which are particularly problematic in the data-scarce test-time setting. We introduce **TTT-Head** (**T**est-**T**ime **T**raining of Attention **Head**s), a fine-grained TTT framework that automatically identifies tunable attention heads and performs instance-specific adaptation in VLMs. Experiments on seven benchmarks covering diverse VQA scenarios—including visual reasoning and compositional question answering, hallucination recognition, multimodal perception and reasoning, and optical character recognition—show that TTT-Head consistently improves the corresponding frozen baselines across all evaluated VLMs, with an average gain of 2.65 points on the LLaVA-1.5-7B backbone. We further demonstrate its generalizability across three widely used VLM families spanning 7B and 13B model scales. In-depth analysis reveals that suppressive interventions on identified key heads are more frequently associated with successful corrections than amplifying interventions, while tunable heads exhibit structured layer-wise preferences across different answer formats.

1 INTRODUCTION

Large vision–language models (VLMs) have become powerful general-purpose models for visual understanding, supporting a broad range of tasks in visual perception and multimodal reasoning. Yet, they remain prone to prediction failures at test time, arising from factors such as insufficient use of visual evidence and distribution shifts (Qiu et al., 2025; Tavella et al., 2025). Test-time training (TTT) has recently emerged as a promising paradigm for mitigating such failures by adapting a pretrained model during inference without requiring labeled data.

Despite remarkable advances, existing TTT methods for VLMs generally determine the adaptation scope in advance, ranging from the entire model (Kaya et al., 2026) to specific modules such as the language model or vision connector (Rajaneesh et al., 2025). The selected parameters are then updated using label-free signals, such as prediction confidence or likelihood (Wang et al., 2021; Hu et al., 2025), auxiliary supervision (Rajaneesh et al., 2025), and prediction consensus (Zuo et al., 2025; Kaya et al., 2026). Although such signals can effectively determine *how* the selected parameters should be updated, they provide limited guidance on *whether* those parameters are appropriate for adaptation on the target task or even for a particular test instance. This fixed adaptation scope may therefore result in inefficient parameter updates and unnecessary model drift, which are especially problematic in the data-scarce test-time setting. This motivates us to think of a fundamental question: *where should adaptation actually take place?*

To shed light on the above question, recent studies on VLM mechanistic interpretability provide a natural starting point, revealing that specific attention heads capture critical context from inputs and route relevant information to subsequent feed-forward networks for prediction (Wang et al., 2025; Jiang et al., 2025). Interestingly, our empirical study shows that *context relevance alone does not guarantee adaptation effectiveness*. Using LLaVA-1.5-7B on three representative VQA benchmarks: GQA (Hudson & Manning, 2019), AMBER (Wang et al., 2023), and MMStar (Chen et al., 2024), we first identified attention heads associated with question-relevant visual evidence. We then

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.

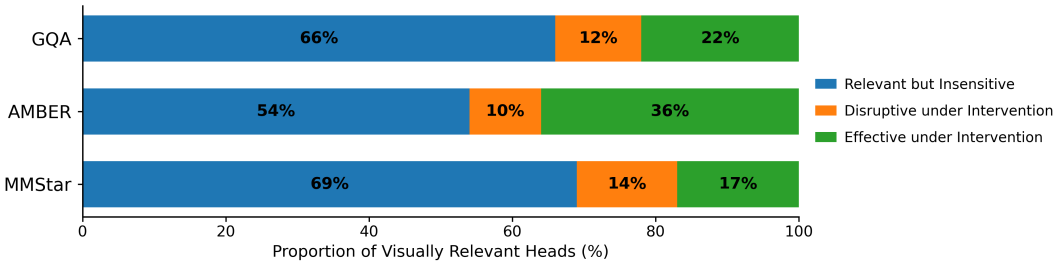

Figure 1: **Visual relevance alone does not guarantee adaptation effectiveness.** Attention heads associated with question-relevant visual evidence are first identified and then evaluated under bounded head interventions. Ground-truth answers are used only for this post-hoc diagnostic. Only a minority of visually relevant heads are effective under intervention, while the remaining heads are largely insensitive or disruptive.

intervened on these heads by amplifying or suppressing their outputs and measured the resulting changes in model predictions. As shown in Fig. 1, only $\sim$25% of visually relevant heads, on average, exhibit a positive intervention effect, while the remaining heads are either largely insensitive or even disruptive. These findings motivate us to identify *adaptation-eligible heads* that exhibit sufficient positive responsiveness to adaptation, rather than selecting heads based solely on their visual relevance. Furthermore, considering the adaptation needs of individual test instances can vary substantially, we seek to make adaptation decisions at the *instance level* rather than the task level.

In this study, we introduce **TTT-Head** (**T**est-**T**ime **T**raining of Attention **Head**s), a fine-grained TTT framework that automatically identifies attention heads suitable for adaptation and performs instance-specific adjustment in VLMs. Specifically, to identify adaptable heads for a test instance, we propose a label-free task-to-instance pipeline. Given a small set of unlabeled calibration samples from the target test distribution, we first construct a task-level candidate pool of adaptable heads based on two criteria: (1) functional association with task-relevant visual evidence; and (2) responsiveness to intervention under relatively low-confidence predictions, while remaining insensitive to intervention under high-confidence predictions. At the instance level, where test labels are unavailable, we seek an internally supported alternative prediction to provide an instance-specific corrective signal, while retaining the frozen prediction as an anchor to avoid overcorrecting potentially correct predictions. Guided by this intuition, for each test instance, we design lightweight, reversible *Head–Action* probes to the candidate heads to identify an alternative prediction as a potential adaptation target and regard its supporting heads as the instance-specific adaptable heads. We then optimize only scalar gates applied to the selected head outputs to steer the model toward the adaptation target. To avoid overcorrection, we calibrate the adaptation strength according to the support for the selected alternative.

We evaluate TTT-Head on seven benchmarks covering diverse VQA settings, using VLMs from three model families and multiple model scales. TTT-Head consistently improves over existing TTT baselines across models and benchmarks; on LLaVA-1.5-7B, it improves the seven-benchmark average by 2.65 percentage points, with similar gains observed across the other model configurations.

Our main contributions are as follows:

- We reveal that functional association with question-relevant visual evidence alone does not guarantee effective adaptation.
- We introduce TTT-Head, a novel fine-grained TTT paradigm that can dynamically determine which attention heads to adapt and how to adjust them for each test instance, enabling instance-specific and targeted adaptation in VLMs.
- We demonstrate the effectiveness and generalizability of TTT-Head across diverse VQA benchmarks and VLMs spanning multiple model families and sizes. Further analysis reveals two consistent patterns in its adaptation behavior: successful corrections more often arise from suppressive than amplifying interventions, while adaptable heads exhibit structured layer-wise preferences across answer formats.

2 Related Work

Test-Time Training for Vision–Language Models. Test-time training (TTT) leverages unlabeled target data during inference to adapt pretrained models using surrogate supervision derived from the test distribution. Existing methods primarily differ in the source of this supervision. Uncertainty- and likelihood-based approaches optimize predictive entropy or test-input likelihood (Wang et al., 2021; Niu et al., 2022; 2023; Hu et al., 2025), while consensus-based methods exploit agreement across repeated predictions or augmented multimodal inputs (Zuo et al., 2025; Kaya et al., 2026). Other approaches incorporate retrieved information (Hardt & Sun, 2024; Lee et al., 2025), instance-derived auxiliary supervision (Rajaneesh et al., 2025), or external vision–language alignment (Nath et al., 2026). Across these approaches, the adaptation interface is typically specified in advance, ranging from test-time prompts (Shu et al., 2022; Samadh et al., 2023) and prescribed architectural modules (Rajaneesh et al., 2025) to broader model parameters (Kaya et al., 2026). In most existing approaches, the main focus is therefore on how to optimize a predefined adaptation interface using label-free test-time signals. TTT-Head instead focuses on identifying which fine-grained internal components constitute an appropriate adaptation interface in the first place.

Mechanistic Interpretability and Internal Intervention. Mechanistic interpretability studies have identified sparse attention heads associated with visual grounding, OCR, evidence retrieval, and multimodal reasoning (Golovanevsky et al., 2025; Bi et al., 2025; Baek et al., 2025; Pei et al., 2026; Jiang et al., 2025). V-SEAM (Wang et al., 2025), for example, combines semantic visual editing with causal intervention to identify heads with positive or negative contributions to object-, attribute-, and relation-level predictions. Recent analyses further show that similar visual-attention magnitudes can correspond to different semantic support for the generated output (Wang et al., 2026b), suggesting that attention strength alone is insufficient to characterize how attended visual information supports the current prediction.

Internal intervention provides a complementary means of controlling model behavior during inference. ITI (Li et al., 2023a) and VTI (Liu et al., 2025) steer internal activations along learned behavior-related directions, while V-SEAM (Wang et al., 2025), SPIN (Sarkar et al., 2025), and AdaIAT (Zhong et al., 2026) manipulate selected internal components through suppression, amplification, or adaptive intervention. These studies show that localized internal components can serve as effective control interfaces. TTT-Head instead asks which heads should form the trainable interface at test time and which should be adjusted and in what direction for the current unlabeled instance. It therefore combines task-level adaptation eligibility with instance-specific Head–Action responses, rather than treating functional importance or controllability alone as sufficient for deciding where and how to train at test time.

3 Method

Fig. 2 displays the overall workflow of TTT-Head, including two stages: (1) Task-level head discovery identifies a small pool of head candidates that are visually relevant and suitable for adaptation; and (2) Instance-specific head selection and adaptation determines which heads to adjust and in which directions for each test instance, and constructs the corresponding corrective supervision.

Formally, we consider a VQA dataset $\mathcal{D}$ consisting of image–question–answer triplets $\{(x, y)\}$, where $x = (I, q)$ contains an image I and a question q. We randomly partition $\mathcal{D}$ into a small calibration split $\mathcal{D}_c$ for task-level head discovery and response calibration, and a held-out evaluation split $\mathcal{D}_e$ for instance-specific adaptation, ensuring that no images are shared between the two splits. For a target pretrained VLM f_θ, let $\mathcal{H}$ denote the set of attention heads in its language decoder. Given a test instance $x \in \mathcal{D}_e$, our goal is to identify an instance-specific subset of heads $\mathcal{H}_x \subseteq \mathcal{H}$, determine how these heads should be adjusted, and construct a label-free corrective signal to guide their test-time training. Within each reported benchmark, ground-truth answers in $\mathcal{D}_c$ are not used for head discovery or response calibration, while those in $\mathcal{D}_e$ are used only for final evaluation.

3.1 Task-Level Head Discovery

Evidence-Grounded Head Localization. We first identify attention heads that are functionally associated with question-relevant visual evidence. For each calibration sample $x = (I, q) \in \mathcal{D}_c$, we

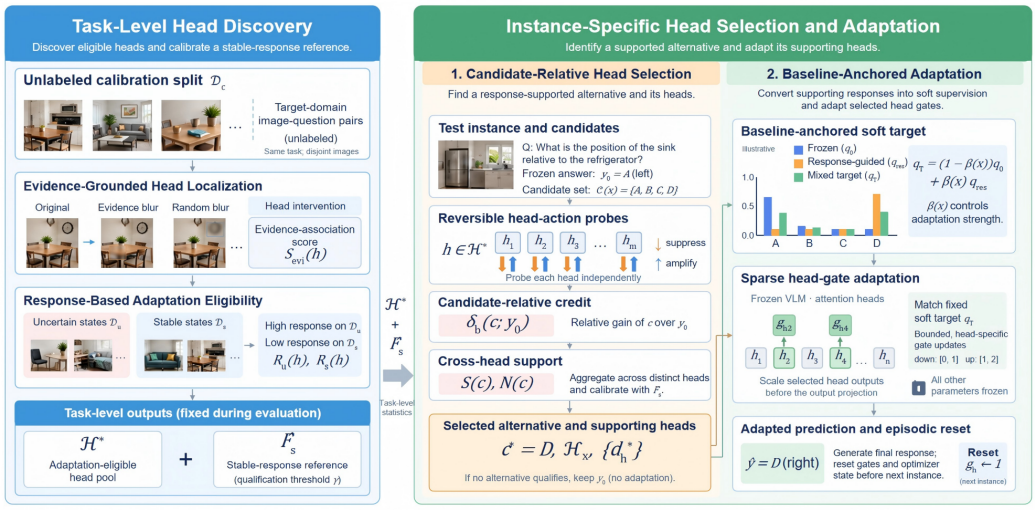

Figure 2: **Overview of TTT-Head.** Task-level head discovery identifies adaptation-eligible heads and calibrates a stable support reference from unlabeled calibration samples. For each test instance, reversible Head–Action probing identifies a qualified alternative, its supporting heads, and their adjustment directions. The resulting responses construct baseline-anchored soft supervision for sparse head-gate adaptation.

use question-to-image attention to localize visual regions relevant to the current question. Specifically, we normalize the attention maps over the visual tokens exposed to the language model and average them across question tokens, attention heads, and language-model layers. The resulting scores are used to identify the top 10% of visual tokens as potential question-relevant visual evidence. To further assess the model's functional dependence on this visual evidence, following prior visual-intervention studies (Wang et al., 2026b; Zhang, 2026), we construct two perturbed images, I^e and I^r, by applying localized Gaussian blur to the identified evidence regions and to an equally sized, non-overlapping random region, respectively. We use I^r as a reference to account for prediction changes induced by random image perturbations.

To identify attention heads causally associated with key visual evidence, for each head $h \in \mathcal{H}$, we apply a mean-replacement intervention to disrupt its individual contribution. Specifically, we replace a head's pre-projection output with the average output of the remaining heads in the same layer. Let y_0 denote the frozen-model response, and let p_h^o, p_h^e, and p_h^r denote the corresponding token prediction distribution on I, I^e, and I^r, respectively, under the same intervention of head h. We define:

$$E_h(x) = \overline{D}_{\mathrm{JS}}(p_h^o, p_h^e) - \overline{D}_{\mathrm{JS}}(p_h^o, p_h^r), \tag{1}$$

$$S_{\mathrm{evi}}(h) = \frac{1}{|\mathcal{D}_c|} \sum_{x \in \mathcal{D}_c} \left| E_h(x) - E_\varnothing(x) \right|, \tag{2}$$

where $\overline{D}_{\mathrm{JS}}$ denotes Jensen–Shannon divergence of two distributions, and $E_\varnothing(x)$ is the corresponding quantity without head intervention. Accordingly, we calculate $S_{\mathrm{evi}}(h)$ to quantify how strongly intervening on head h affects the model's dependence on question-relevant visual evidence. We then rank all heads by $S_{\mathrm{evi}}(h)$ and retain the top 10% as the evidence-grounded head set $\mathcal{H}_{\mathrm{evi}}$. The details of our implementation are provided in Appendix B.2.

Response-Based Adaptation Eligibility. Visual relevance alone does not establish adaptation eligibility: a visually involved head may respond weakly when modified or disturb otherwise stable predictions. We therefore seek heads that respond sufficiently when the model is uncertain while remaining relatively stable under confident predictions. To characterize these two predictive states, we use frozen-response entropy to form high- and low-uncertainty calibration subsets $\mathcal{D}_u$ and $\mathcal{D}_s$, respectively. These subsets reflect predictive states rather than correctness labels.

For each $h \in \mathcal{H}_{\mathrm{evi}}$, we suppress and amplify one head at a time while keeping all others unchanged, so that the resulting response can be attributed to that head. Let $R_u(h)$ and $R_s(h)$ denote its mean

intervention response over $\mathcal{D}_u$ and $\mathcal{D}_s$, respectively, averaging suppressive and amplifying responses in each state. We retain heads that respond sufficiently under uncertain predictions while excluding those that strongly disturb stable states:

$$\begin{aligned} \mathcal{H}_{\text{resp}} &= \{h \in \mathcal{H}_{\text{evi}} : R_u(h) \geq \eta_u\}, \\ \mathcal{H}^* &= \{h \in \mathcal{H}_{\text{resp}} : R_s(h) \leq \eta_s\}. \end{aligned} \tag{3}$$

Here, η_u and η_s are responsiveness and preservation thresholds. Their percentile levels are selected once using separate development data, while their numerical values and the resulting $\mathcal{H}^*$ are estimated from each model–task calibration split and fixed before evaluation. Full details are provided in Appendices A.2 and B.2.

3.2 Instance-Specific Head Selection and Adaptation

The task-level stage yields $\mathcal{H}^*$, a compact pool of heads that are generally suitable for adaptation, but does not determine their instance-specific utility, directions, or corrective signal. Static head information alone is insufficient to resolve these choices. We therefore use reversible Head–Action probing, temporarily suppressing or amplifying individual eligible heads and observing how the model's output preference responds.

These intervention responses are then compared in a common candidate space to determine whether adaptation is sufficiently supported and, if so, to identify a response-supported alternative, its supporting heads, and their directions. The same responses are subsequently converted into baseline-anchored soft supervision for sparse joint test-time training.

Candidate-Relative Head Responses. Different Head–Actions may induce different output changes, making their effects difficult to compare directly in the open response space. We therefore construct a compact candidate set $\mathcal{C}(x)$ as a common reference space for comparing intervention-induced preference shifts. For selection-based tasks, $\mathcal{C}(x)$ is the legal answer space; for open-ended tasks, it contains high-probability frozen-model alternatives together with the original response y_0.

A Head–Action branch $b = (h, a)$ temporarily scales only the pre-projection output of head h, with $a < 1$ and $a > 1$ denoting suppression and amplification, respectively. Let $s_b(c)$ and $s_0(c)$ denote the length-normalized sequence log-probabilities of candidate c under branch b and the frozen model.

Because an intervention may increase the scores of both c and y_0, the absolute change of c alone is insufficient. We therefore measure the change in their relative margin:

$$\delta_b(c; y_0) = [s_b(c) - s_b(y_0)] - [s_0(c) - s_0(y_0)]. \tag{4}$$

A positive $\delta_b(c; y_0)$ indicates an intervention-induced preference shift toward c relative to y_0, rather than evidence that c is correct.

Response-Supported Head Selection. A positive response from a single Head–Action only shows that one intervention can shift preference toward a candidate; it does not establish that the shift is supported by multiple eligible heads. Moreover, multiple probe strengths of the same head should not be counted as independent support. We therefore first summarize the tested actions of each head into a candidate-specific response and then aggregate support across distinct heads.

For each head h and candidate c, let $s_h(c)$ denote the largest candidate-relative response among its tested Head–Actions, and let $d_h(c) \in \{\text{down}, \text{up}\}$ denote the corresponding direction. We aggregate positive response strength and the number of distinct supporting heads as

$$S(c) = \sum_{h \in \mathcal{H}^*} [s_h(c)]_+, \qquad N(c) = \sum_{h \in \mathcal{H}^*} \mathbb{I}[s_h(c) > 0], \tag{5}$$

where $[z]_+ = \max(z, 0)$ and $\mathbb{I}[\cdot]$ is the indicator function.

Simply maximizing $S(c)$ could select an alternative even when all responses are weak. We therefore qualify $c \neq y_0$ only if $S(c) > \gamma$ and $N(c) \geq 2$, where γ is calibrated from alternative support observed on stable samples in $\mathcal{D}_s$. If no alternative qualifies, we retain y_0 without adaptation. Otherwise, the qualified candidate with the largest $S(c)$ is selected as c^*, with supporting heads $\mathcal{H}_x = \{h \in \mathcal{H}^* : s_h(c^*) > 0\}$ and directions $d_h^* = d_h(c^*)$. Thus, c^*, $\mathcal{H}_x$, and d_h^* are all derived from the same candidate-relative Head–Action responses.

Baseline-Anchored Sparse Adaptation. Internal intervention responses reveal preference shifts but do not establish that c^* is correct. We therefore avoid treating c^* as a hard pseudo-label and instead use its supporting Head–Actions to construct a graded corrective signal.

Let q_0 denote the frozen-model distribution over $\mathcal{C}(x)$. For each supporting head, we retain the strongest tested probe in its selected direction and aggregate its candidate-score changes using normalized weights proportional to its support for c^*. Adding the resulting response residual to the frozen candidate scores and applying temperature-scaled softmax yields the response-derived distribution q_{res}.

Although q_{res} is derived from the frozen scores, the response-induced shift may still move it substantially away from q_0, while the original prediction may already be correct. We therefore explicitly control the correction strength:

$$q_T = (1 - \beta(x))q_0 + \beta(x)q_{\mathrm{res}}, \tag{6}$$

where $\beta(x)$ increases with the calibrated support of c^* while remaining bounded. Stronger support permits a larger correction while retaining q_0 as an explicit anchor.

The probes identify locally supportive heads and directions, but their discrete single-head strengths do not determine optimal adjustment magnitudes when multiple heads are modified jointly. We therefore jointly optimize the selected scalar gates toward q_T while constraining each gate to its probe-supported direction. Let g_h denote the scalar gate applied to selected head h, and let q_g denote the candidate distribution under the jointly gated model. With all pretrained parameters frozen, we optimize

$$\min_{\{g_h\}_{h\in\mathcal{H}_x}} D_{\mathrm{KL}}(q_T \| q_g). \tag{7}$$

Each g_h is constrained to $[0, 1]$ for $d_h^* = \mathrm{down}$ and to $[1, 2]$ for $d_h^* = \mathrm{up}$, preventing optimization from reversing the probe-supported direction.

After adaptation, decoding follows the frozen baseline protocol; $\mathcal{C}(x)$ is used only for supervision and does not restrict the final output. Implementation details are provided in Appendix B.3.

4 Experiments

We evaluate **TTT-Head** across seven vision–language benchmarks and five model configurations. Our experiments ask three questions: (1) whether its gains extend across heterogeneous tasks, architectures, and model scales; (2) which components are necessary for effective head-level test-time training; and (3) what patterns characterize the discovered head pools and successful corrections.

4.1 Experimental Setup

Models and Benchmarks. We evaluate TTT-Head on three VLM families with different vision–language interfaces: LLaVA-1.5 (Liu et al., 2024), InstructBLIP (Dai et al., 2023), and Qwen2.5-VL (Bai et al., 2025). LLaVA-1.5-7B serves as the primary backbone, while LLaVA-1.5-13B, InstructBLIP-7B/13B, and Qwen2.5-VL-7B assess robustness across architectures and scales. The seven benchmarks cover open-ended/compositional VQA (GQA Open (Hudson & Manning, 2019), COCO-QA (Ren et al., 2015)), hallucination-sensitive judgment (POPE (Li et al., 2023b), AMBER (Wang et al., 2023)), multimodal perception and reasoning (MME (Fu et al., 2025), MMStar (Chen et al., 2024)), and scene-text understanding (TextVQA (Singh et al., 2019)).

Evaluation Protocol. Because task-level discovery precedes instance-level test-time training, we use approximately 20%/80% image-disjoint unlabeled-calibration and held-out-evaluation splits, with the discovered adaptation space fixed before evaluation. Calibration labels are not used; global percentile hyperparameters are selected once on a separate labeled development set and fixed across all model–task configurations. We evaluate 1,000 held-out examples for GQA Open, COCO-QA, POPE, AMBER, and MME, and 500 for MMStar and TextVQA. All methods use the same held-out instances, prompts, decoding settings, and evaluation pipeline.

Table 1: **Main results on LLaVA-1.5-7B.** GQA Open, COCO-QA, POPE, AMBER, and MME use 1K evaluation examples; MMStar and TextVQA use 500. All entries are reported on a 0–100 scale. Among label-free methods, the best and second-best results are shown in **bold** and underlined, respectively. † denotes a supervised head-intervention reference excluded from this ranking.

Method	GQA Open	COCO-QA	POPE	AMBER	MME-1K	MMStar	TextVQA	Avg.
Frozen VLM	58.40	75.40	84.90	73.50	78.40	29.60	45.54	63.68
Entropy Minimization	59.00	76.10	85.20	73.30	79.00	29.80	45.18	63.94
SAR	59.20	76.30	85.30	73.80	79.20	30.00	45.60	64.20
TTW	59.00	76.10	85.50	73.70	78.90	30.00	45.78	64.14
TTAdapt	59.80	77.00	86.20	75.20	79.60	30.80	46.62	65.03
SPIN	58.50	75.60	**87.00**	75.40	78.30	29.80	45.48	64.30
V-SEAM†	60.10	76.60	87.30	77.20	79.60	30.80	46.20	65.40
TTT-Head (Ours)	**61.20**	**78.00**	86.50	**76.70**	**80.80**	**32.40**	**48.66**	**66.32**

Baselines and Metrics. We compare generic label-free test-time methods (Entropy Minimization (ENT) (Wang et al., 2021), SAR (Niu et al., 2023)), VLM-specific test-time training methods (TTW (Rajaneesh et al., 2025), TTAdapt (Kaya et al., 2026)), and head-level intervention (SPIN (Sarkar et al., 2025)). V-SEAM (Wang et al., 2025) is reported separately as a supervised head-intervention reference and excluded from the label-free ranking. Together, these baselines test whether the gains of TTT-Head can be explained by generic test-time optimization, VLM-specific supervision, or simply by operating at the attention-head level. Results are reported on a 0–100 scale; MME uses accuracy on the fixed 1K subset (*MME-1K Acc.*), and *Avg.* is the unweighted macro-average across seven benchmarks. Further details are provided in Appendices A.2–B.5.

4.2 Results across Tasks, Architectures, and Scales

The gains extend across heterogeneous task formats. Table 1 examines whether TTT-Head remains effective across tasks with substantially different visual demands and answer spaces.

TTT-Head improves the frozen LLaVA-1.5-7B on all seven benchmarks, increasing the macro-average from 63.68 to 66.32 ($+2.65$ points), and achieves the best label-free result on six tasks. The gains span open-ended VQA, hallucination-sensitive judgment, multimodal reasoning, and scene-text understanding, indicating that the same head-level training mechanism remains useful across substantially different prediction spaces and visual demands.

POPE is the only benchmark where TTT-Head is not the strongest label-free method, with the suppression-oriented SPIN performing slightly better, consistent with its specialized hallucination-mitigation design. Across the full suite, TTT-Head exceeds TTAdapt by 1.29 points on average and also achieves a higher overall average than the supervised V-SEAM reference without using labels for head discovery.

The task-to-instance procedure generalizes across architectures and scales. Figure 3 examines whether the improvement depends on a particular backbone or a fixed discovered head set. TTT-Head yields positive gains across all five model configurations, with seven-benchmark macro-average improvements of 1.90–2.77 points. The effect persists across the 7B and 13B variants of LLaVA and InstructBLIP and also holds on Qwen2.5-VL-7B.

Crucially, the adaptation space is rediscovered independently for every model–task pair. The cross-model gains therefore demonstrate generalization of the task-to-instance discovery-and-training procedure rather than transfer of a fixed head pool. Complete results are provided in Appendix E.1.

4.3 Ablation Studies

We next isolate three questions: *which heads are suitable for training*, *what supervision should guide them*, and *how the correction should be realized*. GQA, AMBER, and MMStar represent open-ended VQA, hallucination-sensitive understanding, and multiple-choice multimodal reasoning, respectively. All variants share the same evaluation protocol, and components not targeted by an ablation retain the full-method settings.



Figure 3: **Cross-model robustness of TTT-Head.** Gains over the corresponding frozen models are reported in percentage points; *Average* is the macro-average over seven benchmarks. Adaptation spaces are independently rediscovered for each backbone.

Table 2: **Ablation of TTT-Head on three representative benchmarks.** The three groups examine task-level head discovery, corrective supervision, and adaptation realization, respectively.

Variant	GQA	AMBER	MMStar
Task-Level Discovery			
Random Initial Shortlist	58.60	73.90	29.60
Matched-Budget Visual-Only	60.10	75.40	30.80
w/o Responsiveness Screening	59.80	75.60	30.40
w/o Preservation Screening	60.30	75.10	31.20
Corrective Supervision			
Same Heads + Entropy	59.40	74.80	30.60
Hard c^* Target	59.50	75.20	30.80
w/o Candidate-Relative Credit	59.80	75.60	30.60
w/o Baseline Mixing	59.60	75.70	30.60
Adaptation Realization			
Direct $\arg\max q_T$	59.60	75.20	31.20
Static Rescaling	59.20	74.50	30.20
Joint Probe Gates (No Optimization)	60.00	75.40	31.00
w/o Direction Constraint	60.30	75.30	31.40
Update All Eligible Heads	60.20	75.00	31.40
Full TTT-Head	**61.20**	**76.70**	**32.40**

Visual relevance alone is insufficient for selecting trainable heads. As shown in Table 2, replacing the evidence-guided shortlist with a random one reduces performance by 2.6–2.8 points, showing that arbitrary sparsity is insufficient. More importantly, even with the final head budget matched to the full method, visual-only selection remains 1.1–1.6 points lower. Removing responsiveness or preservation screening also consistently hurts performance. Thus, visual relevance identifies potentially involved components, but response-based screening is needed to distinguish heads suitable for test-time training.

Head selection alone is insufficient without response-derived supervision. The second group asks whether the proposed corrective supervision can be replaced or simplified. In the most controlled comparison, *Same Heads + Entropy* keeps the supporting heads and training interface fixed while replacing our target with entropy minimization, reducing performance by 1.8–1.9 points. A one-hot target on c^* remains 1.5–1.7 points below the full method, while removing candidate-relative credit or baseline mixing produces the same consistent degradation. These results show that the gain does not arise from head selection alone: intervention responses provide graded corrective supervision, while the frozen prediction prevents an internally supported alternative from being treated as an unquestioned pseudo-label.

Probing identifies a supported correction, while sparse optimization realizes it. Directly decoding from $\arg\max q_T$ or jointly applying the strongest probed gates without optimization re-

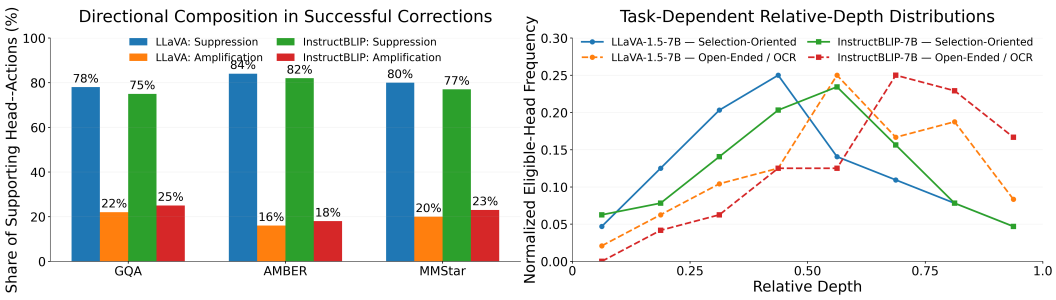

Figure 4: **Head-level regularities across architectures. Left:** directional composition of supporting Head–Actions in successful corrections; mixed-direction instances contribute separately for each supporting head. **Right:** 8-bin relative-depth distributions of task-level adaptation-eligible heads. Within each benchmark, eligible-head counts are normalized by the discovered pool size, and benchmark distributions are then averaged with equal weight within each task group.

mains 1.2–1.6 and 1.2–1.4 points below the full method, respectively. Removing the probe-derived direction constraint also degrades performance, showing that unconstrained optimization does not replace the intervention signal. Updating all eligible heads further underperforms instance-specific selection. Together, these results indicate complementary roles: probing identifies an internally supported alternative, its supporting heads, and adjustment directions, whereas sparse gate optimization determines the instance-specific adjustment magnitude.

Implementation details and additional analyses are provided in Appendices B.4, C.2, D.2, and D.3.

4.4 Analysis of Head-Level Adaptation

We next examine two head-level patterns: the directions of supporting Head–Actions in successful corrections and the depth distribution of adaptation-eligible heads.

Suppressive Head–Actions Are More Frequent among Successful Corrections. Figure 4 (left) shows that suppressive directions account for the majority of supporting Head–Actions in successful corrections on GQA, AMBER, and MMStar for both LLaVA-1.5-7B and InstructBLIP-7B. This directional asymmetry suggests that attenuation is the more common adjustment among supporting heads, while amplification remains present in a substantial minority.

Adaptation-Eligible Heads Exhibit Task-Dependent Depth Distributions. Figure 4 (right) shows that task-level pools $\mathcal{H}^*$ occupy different relative-depth regions across task groups. After benchmark-wise normalization and equal-weight averaging, selection-oriented tasks shift earlier, whereas open-ended and OCR-oriented tasks shift later within each backbone. This suggests that test-time adaptation may benefit from task-specific adaptation regions rather than a fixed layer range.

Representative corrections and additional head-level analyses are provided in Appendix F; complete cross-model results are in Appendix E.1. Case visualizations are post-hoc only.

5 Conclusion

In this paper, we study *where* adaptation should take place in vision–language model test-time training and introduce **TTT-Head**, which separates task-level adaptation eligibility from instance-specific adaptation decisions. Rather than relying on visual relevance or a predefined adaptation interface, TTT-Head uses intervention responses to identify adaptation-eligible heads and construct corrective supervision from sparse supporting heads. Across seven benchmarks and multiple VLM families and scales, this task-to-instance strategy yields consistent improvements. Our analyses further distinguish visual relevance from adaptation eligibility and reveal recurring directional and depth-related patterns in successful corrections.

AI USE STATEMENT

Generative AI tools were used only to aid or polish the writing of this paper, including improving grammar, clarity, and presentation. All scientific content and technical contributions were developed by the authors.

REFERENCES

Ingeol Baek, Hwan Chang, Sunghyun Ryu, and Hwanhee Lee. How do large vision-language models see text in image? unveiling the distinctive role of OCR heads. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20441–20453, 2025. doi: 10.18653/v1/2025.emnlp-main.1032.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Jing Bi, Junjia Guo, Yunlong Tang, Lianggong Bruce Wen, Zhang Liu, Bingjie Wang, and Chenliang Xu. Unveiling visual perception in language models: An attention head analysis approach. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4135–4144, 2025.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. Are we on the right way for evaluating large vision-language models? In *Advances in Neural Information Processing Systems*, volume 37, pp. 27056–27087, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/2f8ee6a3d766b426d2618e555b5aeb39-Abstract-Conference.html.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven C. H. Hoi. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 49250–49267, 2023.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, Rongrong Ji, Caifeng Shan, and Ran He. MME: A comprehensive evaluation benchmark for multimodal large language models. In *Advances in Neural Information Processing Systems*, volume 38, pp. 162549–162567, 2025. doi: 10.52202/085713-4899. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/d79a27cf2772fe00be7f341efc0eb517-Abstract-Datasets_and_Benchmarks_Track.html.

Michal Golovanevsky, William Rudman, Vedant Palit, Carsten Eickhoff, and Ritambhara Singh. What do VLMs NOTICE? a mechanistic interpretability pipeline for Gaussian-noise-free text-image corruption and evaluation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 11462–11482, 2025. doi: 10.18653/v1/2025.naacl-long.571. URL https://aclanthology.org/2025.naacl-long.571/.

Moritz Hardt and Yu Sun. Test-time training on nearest neighbors for large language models. In *International Conference on Learning Representations*, 2024.

Jinwu Hu, Zitian Zhang, Guohao Chen, Xutao Wen, Chao Shuai, Wei Luo, Bin Xiao, Yuanqing Li, and Mingkui Tan. Test-time learning for large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 24823–24849, 2025.

Drew A. Hudson and Christopher D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, pp. 6700–6709, 2019. URL https://openaccess.thecvf.com/content_CVPR_2019/html/Hudson_GQA_A_New_Dataset_for_Real-World_Visual_Reasoning_and_Compositional_CVPR_2019_paper.html.

Yanbei Jiang, Xueqi Ma, Shu Liu, Sarah Monazam Erfani, Tongliang Liu, James Bailey, Jey Han Lau, and Krista A. Ehinger. Investigating the functional roles of attention heads in vision language models: Evidence for reasoning modules. *arXiv preprint arXiv:2512.10300*, 2025.

Mehmet Onurcan Kaya, Desmond Elliott, and Dim P. Papadopoulos. Efficient test-time scaling for small vision-language models. In *International Conference on Learning Representations*, 2026.

Youngjun Lee, Doyoung Kim, Junhyeok Kang, Jihwan Bang, Hwanjun Song, and Jae-Gil Lee. RA-TTA: Retrieval-augmented test-time adaptation for vision-language models. In *International Conference on Learning Representations*, 2025.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems*, volume 36, pp. 41451–41530, 2023a. doi: 10.52202/075280-1797. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/81b8390039b7302c909cb769f8b6cd93-Abstract-Conference.html.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 292–305, 2023b. doi: 10.18653/v1/2023.emnlp-main.20.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26296–26306, 2024.

Sheng Liu, Haotian Ye, and James Y. Zou. Reducing hallucinations in large vision-language models via latent space steering. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/b4008025c2182bfe16fcc8566ee14d64-Abstract-Conference.html.

Mriganka Nath, Anurag Das, Jiahao Xie, and Bernt Schiele. ClipTTT: CLIP-guided test-time training helps LVLMs see better. *arXiv preprint arXiv:2603.26486*, 2026. doi: 10.48550/arXiv.2603.26486.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. Efficient test-time model adaptation without forgetting. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 16888–16905, 2022.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. Towards stable test-time adaptation in dynamic wild world. In *International Conference on Learning Representations*, 2023.

Rongcan Pei, Huan Li, Fang Guo, and Qi Zhu. VERA: Identifying and leveraging visual evidence retrieval heads in long-context understanding. *arXiv preprint arXiv:2602.10146*, 2026. doi: 10.48550/arXiv.2602.10146.

Xinkuan Qiu, Meina Kan, Yongbin Zhou, and Shiguang Shan. Benchmarking multimodal large language models against image corruptions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

Nikita Rajaneesh, Thomas Zollo, and Richard Zemel. Test-time warmup for multimodal large language models. *arXiv preprint arXiv:2509.10641*, 2025.

Mengye Ren, Ryan Kiros, and Richard Zemel. Exploring models and data for image question answering. In *Advances in Neural Information Processing Systems*, volume 28, pp. 2953–2961, 2015.

Jameel Hassan Abdul Samadh, Hanan Gani, Noor Hazim Hussein, Muhammad Uzair Khattak, Muzammal Naseer, Fahad Shahbaz Khan, and Salman Khan. Align your prompts: Test-time prompting with distribution alignment for zero-shot generalization. In *Advances in Neural Information Processing Systems*, volume 36, pp. 80396–80413, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/fe8debfd5a36ada52e038c8b2078b2ce-Abstract-Conference.html.

Sreetama Sarkar, Yue Che, Alex Gavin, Peter Anthony Beerel, and Souvik Kundu. Mitigating hallucinations in vision-language models through image-guided head suppression. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 12481–12500, Suzhou, China, November 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.631. URL https://aclanthology.org/2025.emnlp-main.631/.

Manli Shu, Weili Nie, De-An Huang, Zhiding Yu, Tom Goldstein, Anima Anandkumar, and Chaowei Xiao. Test-time prompt tuning for zero-shot generalization in vision-language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 14274–14289, 2022.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards VQA models that can read. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8317–8326, 2019.

Federico Tavella, Amber Drinkwater, and Angelo Cangelosi. Evaluating the robustness of open-source vision-language models to domain shift in object captioning. *arXiv preprint arXiv:2506.19579v2*, 2025. URL https://arxiv.org/abs/2506.19579v2.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=uXl3bZLkr3c.

Junyang Wang, Yuhang Wang, Guohai Xu, Jing Zhang, Yukai Gu, Haitao Jia, Jiaqi Wang, Haiyang Xu, Ming Yan, Ji Zhang, and Jitao Sang. AMBER: An LLM-free multi-dimensional benchmark for MLLMs hallucination evaluation. *arXiv preprint arXiv:2311.07397*, 2023.

Qidong Wang, Junjie Hu, and Ming Jiang. V-SEAM: Visual semantic editing and attention modulating for causal interpretability of vision-language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 17396–17420, 2025. doi: 10.18653/v1/2025.emnlp-main.880.

Qidong Wang, Junjie Hu, and Ming Jiang. From heads to neurons: Causal attribution and steering in multi-task vision–language models. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 36151–36175, San Diego, California, United States, July 2026a. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-acl.1802. URL https://aclanthology.org/2026.findings-acl.1802/.

Zichuan Wang, Songlin Yang, Bo Peng, Zhenchen Tang, Yang Li, Beibei Dong, and Jing Dong. Same attention, different truths: Put logit-lens over visual attention to detect and mitigate LVLM object hallucination. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25315–25325, 2026b.

Genpei Zhang. The visual insensitivity gap: Diagnosing when vision-language models fail to use visual evidence. *arXiv preprint arXiv:2609.00868*, 2026. doi: 10.48550/arXiv.2609.00868. URL https://arxiv.org/abs/2609.00868.

Li'an Zhong, Ziqiang He, Jibin Zheng, Jin Li, Z. Jane Wang, and Xiangui Kang. AdaIAT: Adaptively increasing attention to generated text to alleviate hallucinations in LVLM. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11076–11085, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Zhong_AdaIAT_Adaptively_Increasing_Attention_to_Generated_Text_to_Alleviate_Hallucinations_CVPR_2026_paper.html.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, Biqing Qi, Youbang Sun, Zhiyuan Ma, Lifan Yuan, Ning Ding, and Bowen Zhou. TTRL: Test-time reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 145667–145691, 2025. doi: 10.52202/085713-4376. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/be690ea16f005c174f6c4102a5970e67-Abstract-Conference.html.

A Experimental Protocol and Evaluation Details

A.1 Models and Benchmarks

Models. We evaluate TTT-Head on three representative VLM families: LLaVA-1.5 (Liu et al., 2024), InstructBLIP (Dai et al., 2023), and Qwen2.5-VL (Bai et al., 2025). These models are selected to examine whether TTT-Head generalizes across VLMs with different visual representations and cross-modal modeling mechanisms, rather than relying on a particular model architecture.

Specifically, LLaVA-1.5 extracts image features with a visual encoder and maps them into the language-model representation space through a projection module, representing a typical vision-encoder–projector VLM architecture. InstructBLIP builds upon the BLIP-2 framework and introduces an instruction-aware Q-Former to extract visual information relevant to the current instruction, resulting in a different vision–language interaction mechanism from LLaVA. Qwen2.5-VL employs a vision Transformer that supports native dynamic-resolution inputs and incorporates local window attention to process visual inputs of different sizes and resolutions, representing a more recent VLM design with a substantially different visual modeling architecture. Together, these three model families cover distinct visual representation and cross-modal modeling paradigms, allowing us to assess the cross-architecture applicability of TTT-Head.

We further evaluate two model scales for both LLaVA-1.5 and InstructBLIP, including LLaVA-1.5-7B/13B and InstructBLIP-Vicuna-7B/13B, to examine whether the effectiveness of test-time training persists across model scales. Qwen2.5-VL-7B is additionally included for cross-architecture evaluation. LLaVA-1.5-7B serves as the primary backbone for the main comparison and controlled ablations, while the remaining four configurations are used to evaluate robustness across architectures and model scales.

Benchmarks. We select seven vision–language benchmarks with complementary task characteristics, covering open-ended visual question answering, hallucination-sensitive visual understanding, multimodal perception and reasoning, and scene-text understanding.

GQA Open (Hudson & Manning, 2019) focuses on compositional visual question answering in real-world scenes, with questions involving objects, attributes, and relations, and is therefore used to evaluate compositional reasoning over complex visual information. COCO-QA (Ren et al., 2015) constructs question-answer pairs from natural images and their descriptions, providing a broader and relatively more fundamental open-ended VQA setting that complements the stronger compositional reasoning requirements of GQA.

POPE (Li et al., 2023b) evaluates object hallucination through object-existence questions and measures factual consistency in binary visual judgments. AMBER (Wang et al., 2023) further covers multiple hallucination types, including object existence, attributes, and relations, providing a broader hallucination-sensitive evaluation setting.

MME (Fu et al., 2025) evaluates VLMs from both perception and cognition perspectives and covers a diverse range of visual perception and reasoning capabilities. MMStar (Chen et al., 2024) places stronger emphasis on genuine dependence on visual information by reducing cases that can be solved primarily from the question text or prior knowledge, thereby providing a more targeted evaluation of multimodal perception and reasoning.

Finally, TextVQA (Singh et al., 2019) requires models to recognize and use scene text in images to answer questions, and is used to evaluate the generalization of TTT-Head to OCR-oriented and scene-text understanding tasks.

These seven benchmarks span open-ended visual question answering, compositional reasoning, hallucination-sensitive judgment, multimodal perception and reasoning, and scene-text understanding, enabling evaluation of TTT-Head across complementary task settings.

A.2 Unlabeled Calibration and Held-Out Evaluation Protocol

Evaluation Set Size. Test-time methods incur additional inference-time computation, and recent multimodal test-time studies therefore commonly evaluate on fixed-size subsets. For example, TTW (Rajaneesh et al., 2025) evaluates 500 test images per dataset. TTAug/TTAdapt (Kaya et al., 2026) uses 1,000 examples per benchmark in its main evaluation. Following a comparable evaluation scale, we use 1,000 held-out test examples for GQA Open, COCO-QA, POPE, AMBER, and MME, and 500 held-out examples for MMStar and TextVQA.

These evaluation subsets are fixed once for our experiments. For each model–benchmark pair, the frozen model, all comparison methods, and TTT-Head are evaluated on exactly the same images, questions, and held-out sample IDs. For POPE, the evaluation pool is constructed by approximately balanced sampling from the random, popular, and adversarial subsets so that all three hallucination settings are represented.

Unlabeled Calibration and Held-Out Split. Unlike methods that operate only on an individual test instance, TTT-Head requires a small set of unlabeled target-benchmark examples before test-time training to perform task-level head discovery and estimate the associated calibration statistics. Using the final evaluation instances for these task-level operations would introduce an unnecessary dependency between construction of the training space and final performance evaluation. We therefore partition a fixed evaluation pool, before any task-level head discovery or calibration, into an unlabeled calibration split $\mathcal{D}_c$ and a held-out evaluation split $\mathcal{D}_e$.

Specifically, we target an approximately 20%/80% instance ratio while enforcing image-level disjointness. All question instances associated with the same image are assigned to only one split, preventing visual content used during task-level discovery from reappearing in held-out evaluation.

For GQA Open, COCO-QA, POPE, AMBER, and MME, we construct a fixed pool of 1,250 instances, of which 250 are used for unlabeled calibration and the remaining 1,000 are reserved for held-out evaluation. For MMStar and TextVQA, we use a pool of 625 instances, with 125 examples for calibration and 500 for held-out evaluation. Thus, the "1K" and "500" settings reported in the main paper refer to the final held-out evaluation split rather than the total pool before partitioning. The resulting split sizes are summarized in Table 3.

Table 3: **Calibration and held-out evaluation splits.** Calibration and held-out evaluation are image-disjoint, and ground-truth answers from the calibration split are never used by TTT-Head.

Benchmark	Total Pool	Calibration	Held-out Eval.
GQA Open	1,250	250	1,000
COCO-QA	1,250	250	1,000
POPE	1,250	250	1,000
AMBER	1,250	250	1,000
MME	1,250	250	1,000
MMStar	625	125	500
TextVQA	625	125	500

Global Hyperparameter Selection. We use a separate labeled development set only for one-time selection of global hyperparameters. The development set contains 500 examples for GQA Open, COCO-QA, POPE, AMBER, and MME, and 300 examples for MMStar and TextVQA, and is separate from the calibration and held-out evaluation pools used for the reported experiments. Within each benchmark, the development set is further partitioned into a development calibration split and a development evaluation split so that each candidate configuration follows the same task-level discovery/calibration and instance-level test-time training protocol as the main experiments. The response-screening percentile levels, qualification quantile q_γ, maximum response weight $\beta_{\max}$, and number of probe strengths per direction are selected once on LLaVA-1.5-7B using this development pro-

tocol and are then fixed across all reported model–task configurations; no label-based retuning is performed for the remaining model families or scales.

Data Access and Test-Time Protocol. For every reported benchmark, the target-domain calibration split is treated as unlabeled and is used only for task-level head discovery and response calibration. The held-out evaluation inputs do not contribute to task-level head discovery or calibration. The task-level head pool and calibration-dependent statistics determined from $\mathcal{D}_c$ are fixed before held-out evaluation begins and remain unchanged throughout the test run.

For each instance in $\mathcal{D}_e$, TTT-Head performs instance-level test-time training independently. The corresponding ground-truth answer is available only to the evaluator for final metric computation and post-hoc analysis, and is never used for candidate construction, head intervention, training-signal construction, or test-time parameter updates.

For each model–benchmark pair, the frozen model, all comparison methods, and TTT-Head use exactly the same held-out images, questions, sample IDs, prompts, and decoding settings. The comparison methods are therefore re-evaluated on our fixed held-out subsets rather than using method-specific evaluation samples from their original studies.

A.3 Evaluation Metrics and Answer Processing

Evaluation Metrics. We use accuracy-based metrics as the primary evaluation criteria across all seven benchmarks and report all results on a 0–100 scale. GQA Open and COCO-QA use normalized answer accuracy; POPE and AMBER use binary classification accuracy; MMStar uses option accuracy; and TextVQA uses the standard VQA-style soft accuracy. For MME, following the fixed-subset evaluation practice adopted in recent test-time studies, we report question-level accuracy on a fixed subset of 1,000 questions and denote it as *MME-1K Acc.*. This metric is distinct from the official aggregate score computed over the full MME benchmark.

For the binary evaluation settings, the fixed evaluation subsets are evenly balanced between `yes` and `no` target labels, avoiding majority-answer bias in binary accuracy.

Avg. denotes the unweighted macro-average across the seven benchmark scores. All averages and performance gains are computed from unrounded values, with rounding applied only for display.

Answer Processing. All methods use the same answer parsing and normalization pipeline. For binary tasks such as POPE, AMBER, and MME, model outputs are deterministically mapped to `yes`/`no`. For MMStar, multiple-choice outputs are mapped to the corresponding option letter. Open-ended predictions for GQA Open, COCO-QA, and TextVQA are normalized using the corresponding benchmark evaluator, including case normalization, punctuation handling, and other benchmark-specific processing when applicable. Ground-truth answers are never used to resolve ambiguous model outputs, and unparseable predictions are treated as incorrect.

TextVQA. TextVQA is evaluated using the standard VQA-style soft accuracy, which scores a prediction according to its agreement with multiple human reference answers rather than strict exact matching. The same TextVQA evaluator is used for the frozen model, all comparison methods, and TTT-Head.

A.4 Benchmark Input and Output Formats

The seven benchmarks adopt different question and answer formats. Table 4 presents representative examples from the corresponding public benchmark data and summarizes the output format used for evaluation.

The AMBER example corresponds to an attribute-verification case. Our evaluation additionally includes existence and relation questions, which follow the same binary answer format.

Table 4: **Representative input and output formats for the seven benchmarks.** Questions and reference answers are taken from publicly available benchmark examples.

Benchmark	Example Question	Reference Answer	Output Format
GQA Open	`What is on the street?`	`car`	Short answer
COCO-QA	`what walks toward the rope perimeter fence`	`elephant`	One word
POPE	`Is there a snowboard in the image?`	`yes`	Yes / No
AMBER	`Is the sky sunny in this image?`	`yes`	Yes / No
MME	`Is a python code shown in the picture?`	`yes`	Yes / No
MMStar	`Which option describes the object relationship in the image correctly? A: The suitcase is on the book. B: The suitcase is beneath the cat. C: The suitcase is beneath the bed. D: The suitcase is beneath the book.`	`A`	A / B / C / D
TextVQA	`What is the time on bottom middle phone?`	`15:20`	Short text

A.5 Prompt Templates and Model Serialization

Task Instructions. We use concise task-specific instructions to constrain the expected output format while preserving the original benchmark question. Table 5 summarizes the prompt templates used for the four task formats in our evaluation.

Table 5: **Task-level prompt templates used in our experiments.** The original benchmark question is preserved, and only a short format instruction is appended to constrain the expected answer form.

Task Format	Benchmarks	Task Instruction
Binary judgment	POPE, AMBER, MME	`{Question}` `Please answer yes or no.`
Multiple choice	MMStar	`{Question with A/B/C/D options}` `Please answer with the option letter only, one of A, B, C, or D.`
Open-ended VQA	GQA Open, COCO-QA	`{Question}` `Answer the question using a single word or phrase.`
OCR-oriented VQA	TextVQA	`{Question}` `Answer with only the short answer, without explanation.`

Model-Specific Serialization. The task instruction is serialized using the native input interface of each model family. For LLaVA-1.5, we follow its standard single-turn conversation format, `USER: <image> \n {task instruction} ASSISTANT:`. For InstructBLIP, the image and textual instruction are jointly processed by the corresponding InstructBLIP processor; the instruction is also provided to the instruction-aware Q-Former, and we do not manually impose a LLaVA-style conversation wrapper. For Qwen2.5-VL, the image and textual instruction are represented as separate multimodal content items within the same user message and serialized using the model's native chat template with `add_generation_prompt=True`.

For a given model–benchmark pair, the same task instruction and model-specific serialization are used for the frozen model, all intervention probes, candidate scoring, and the final generation produced by TTT-Head. No ground-truth answer, candidate hint, or additional external evidence is included in the benchmark prompt.

B Implementation Details

B.1 Head-Level Training Interface

TTT-Head performs test-time training only on selected attention heads in the language-model self-attention modules. Let $h = (\ell, j)$ denote head j in language-model layer ℓ. We introduce a learnable

scalar gate g_h and apply it uniformly to the corresponding pre-projection head output:

$$\widetilde{\mathbf{o}}_{\ell,j,t} = g_h \mathbf{o}_{\ell,j,t},$$

where $\mathbf{o}_{\ell,j,t}$ is the original output of head h at token position t. The identity setting is $g_h = 1$, while $g_h < 1$ and $g_h > 1$ correspond to suppressing and amplifying the contribution of that head, respectively.

During test-time training, only the scalar gates associated with the selected heads are optimized. All pretrained model parameters, including the Q/K/V projections, attention output projections, vision encoder, and cross-modal connector, remain frozen.

The gate is applied while individual head outputs are still explicitly separable, before concatenation and the subsequent attention output projection. This placement allows individual heads to be controlled before their contributions are mixed by the output projection. For architectures using grouped-query attention, the gate is applied to the separable query-head output slices before the output projection; shared key/value groups remain unchanged. The same head-level interface is used for intervention probing, candidate scoring, gate optimization, and final decoding.

B.2 Task-Level Head Discovery Details

Evidence Localization and Perturbation. The first stage identifies heads that are functionally associated with visual evidence relevant to the current question. For each unlabeled calibration instance $x = (I, q) \in \mathcal{D}_c$, let $\mathcal{T}_q$ denote the question-token positions and $A_{\ell,j,t}(u)$ the attention weight from question token t to an architecture-specific language-model visual unit u at layer ℓ and head j. We normalize each head–token attention map over visual units to obtain $\bar{A}_{\ell,j,t}(u)$ and aggregate them as

$$A_x(u) = \frac{1}{L} \sum_{\ell=1}^{L} \frac{1}{H_\ell} \sum_{j=1}^{H_\ell} \frac{1}{|\mathcal{T}_q|} \sum_{t \in \mathcal{T}_q} \bar{A}_{\ell,j,t}(u),$$

where L is the number of language-model layers and H_ℓ is the number of attention heads in layer ℓ.

Because the visual units exposed to the language model have different spatial semantics across architectures, we map $A_x(u)$ to a relevance score $R_x(p)$ over spatial units p on the corresponding vision-encoder grid. For LLaVA-1.5, projected visual tokens preserve the spatial ordering of the vision-encoder patch grid, so the relevance score is directly assigned to the corresponding spatial patch.

For InstructBLIP, the language model receives projected Q-Former queries rather than spatial patch tokens. We therefore back-project the query relevance through the last Q-Former layer containing cross-attention. Letting $C_x(u, p)$ denote the cross-attention from query u to spatial vision unit p, averaged across attention heads, we use

$$R_x(p) = \sum_u A_x(u) C_x(u, p).$$

For Qwen2.5-VL, spatial units are defined at the merged-cell level using the processor-provided visual grid and spatial-merge metadata. Relevance is ranked over the merged cells; when a merged cell is selected, all pre-merge patches belonging to that cell are included in the corresponding image-space mask.

For each instance, we set $K_{\text{evi}} = \lceil 0.10 |\mathcal{V}_x| \rceil$, where $\mathcal{V}_x$ denotes the architecture-specific spatial-unit set. The K_{evi} highest-scoring units form the evidence proposal $\mathcal{V}_e(x)$. We sample an equally sized non-overlapping control set $\mathcal{V}_r(x)$ from the remaining spatial units:

$$|\mathcal{V}_r(x)| = |\mathcal{V}_e(x)|, \qquad \mathcal{V}_r(x) \cap \mathcal{V}_e(x) = \varnothing.$$

Following prior localized visual-intervention practice (Zhang, 2026), we map the selected units to image regions and construct I^e and I^r separately from the original image using the same Gaussian blur ($\sigma = 20$, kernel size 81×81). Each perturbed image is passed through the complete visual processing pipeline again rather than reusing the original visual features. Attention therefore determines the candidate evidence regions, whereas head relevance is measured through the intervention-induced evidence dependency in Eqs. 1–2.

Mean-Replacement Intervention. For head $h = (\ell, j)$, let $\mathbf{o}_{\ell,j,t}$ denote its pre-projection output at token position t. Following mean-replacement intervention (Wang et al., 2025; 2026a), we replace the selected head with the same-token mean of the remaining heads in the same layer:

$$\mathbf{o}_{\ell,j,t} \leftarrow \frac{1}{H_\ell - 1} \sum_{k \neq j} \mathbf{o}_{\ell,k,t}.$$

The mean is computed from the same forward pass, separately for the original, evidence-perturbed, and control-perturbed inputs; it is not a stored cross-example mean. Mean replacement is used only for evidence-grounded head localization. Response screening and instance-level Head–Action probing instead use the scalar gate defined in Appendix B.1.

Shared Distributional Response Measure. For the frozen response $y_0 = (y_{0,1}, \ldots, y_{0,T})$, we compare two input or intervention conditions a and b using token-averaged Jensen–Shannon divergence:

$$\overline{D}_{\mathrm{JS}}(p^a, p^b) = \frac{1}{T} \sum_{t=1}^{T} D_{\mathrm{JS}}\big(p_t^a(\cdot \mid y_{0,<t}), p_t^b(\cdot \mid y_{0,<t})\big)\,.$$

Both conditions use the same frozen-response prefixes and compare the full next-token distributions rather than independently decoded sequences. This measure is used both in the evidence-dependency calculation and in state-conditioned response screening.

Uncertainty-Based Calibration States. Visual relevance alone does not establish adaptation eligibility. We therefore use frozen-model uncertainty to distinguish states in which a head should be responsive from those in which its intervention should remain relatively limited.

Let $p_{0,t}(\cdot \mid x, y_{0,<t})$ denote the frozen model's next-token distribution at position t. We define

$$\begin{aligned} U(x) &= \frac{1}{T} \sum_{t=1}^{T} \mathrm{Ent}(p_{0,t}(\cdot \mid x, y_{0,<t}))\,, \\ \mathcal{D}_u &= \{x \in \mathcal{D}_c : U(x) \geq Q_{1-\alpha}\}, \\ \mathcal{D}_s &= \{x \in \mathcal{D}_c : U(x) \leq Q_\alpha\}, \end{aligned} \tag{8}$$

where Q_r denotes the r-quantile of $\{U(x) : x \in \mathcal{D}_c\}$ and $\alpha = 0.30$. Thus, $\mathcal{D}_u$ and $\mathcal{D}_s$ are the highest- and lowest-entropy 30% of calibration samples, respectively. They represent predictive states rather than ground-truth correctness.

Response-Based Head Screening. For each $h \in \mathcal{H}_{\mathrm{evi}}$ and calibration instance, we sample one suppressive gate from $[0, 1)$ and one amplifying gate from $(1, 2]$. A fresh pair is sampled for each instance and shared across shortlisted heads to provide matched-strength comparisons. Each probe modifies only one head, and its gate is restored to one before the next probe.

Let p_0 denote the frozen next-token distribution sequence along y_0, and let p_h^- and p_h^+ denote the corresponding sequences after suppressing and amplifying head h. The bidirectional response is

$$r_h(x) = \frac{1}{2}\left[\overline{D}_{\mathrm{JS}}(p_h^-, p_0) + \overline{D}_{\mathrm{JS}}(p_h^+, p_0)\right],$$

and its state-conditioned averages are

$$R_u(h) = \mathbb{E}_{x \in \mathcal{D}_u}[r_h(x)], \qquad R_s(h) = \mathbb{E}_{x \in \mathcal{D}_s}[r_h(x)].$$

$R_u(h)$ measures whether a head responds sufficiently in uncertain states, whereas $R_s(h)$ measures how strongly the same head affects relatively stable predictions. The sequential head-screening rule is given in Eq. 3. Its numerical thresholds are computed as

$$\eta_u = \mathrm{Quantile}_{0.60}\{R_u(h) : h \in \mathcal{H}_{\mathrm{evi}}\}, \qquad \eta_s = \mathrm{Quantile}_{0.40}\{R_s(h) : h \in \mathcal{H}_{\mathrm{resp}}\}.$$

The response threshold is therefore computed over $\mathcal{H}_{\mathrm{evi}}$, whereas the preservation threshold is computed after $\mathcal{H}_{\mathrm{resp}}$ has been formed. The 60th- and 40th-percentile levels are selected once using the development protocol in Appendix A.2 and then fixed across all reported model–task configurations. Only their numerical values are estimated separately from each unlabeled calibration split.

Stable-Response Calibration. After $\mathcal{H}^*$ has been fixed, we use the stable subset $\mathcal{D}_s$ to calibrate how much candidate-relative support is required before instance-level adaptation is triggered. This step serves a different purpose from the response screening above: it applies the same candidate construction and Head–Action probing used at test time and measures support in the common candidate space.

For each $x' \in \mathcal{D}_s$, we use three independently sampled probe strengths per direction, shared across all heads in $\mathcal{H}^*$, and compute the strongest alternative support. Let y'_0 denote the frozen response for x', and let $\widehat{F}_s$ denote the empirical CDF of the resulting values $\{M(x') : x' \in \mathcal{D}_s\}$. We define

$$M(x') = \max_{c \in \mathcal{C}(x') \setminus \{y'_0\}} S(c), \qquad \gamma = \widehat{F}_s^{-1}(q_\gamma), \tag{9}$$

with $q_\gamma = 0.70$.

The stable-response reference $\widehat{F}_s$ and threshold γ are fixed before held-out evaluation begins. The same reference is also used to control the strength of baseline anchoring during instance-level adaptation; it is an empirical reference for intervention-supported preference shifts, not an estimate of candidate correctness.

All task-level discovery and calibration are completed before evaluation. The resulting $\mathcal{H}^*$, $\widehat{F}_s$, and calibration-dependent thresholds remain fixed throughout the held-out test run.

B.3 Instance-Level Test-Time Training Details

Candidate Construction and Reversible Probing. For each test instance $x = (I, q)$, we first obtain the frozen response y_0 and construct $\mathcal{C}(x)$ before any head intervention. The candidate set provides a common space in which Head–Action responses can be compared.

For selection-based tasks, $\mathcal{C}(x)$ is the legal answer space. For open-ended tasks, we use $K_{\text{cand}} = 4$. At the first answer position, we exclude the frozen model's selected token $y_{0,1}$ and take the four highest-probability remaining tokens as alternative prefixes. Each prefix is continued into a complete response using the frozen model. We then add y_0, apply the same answer normalization used by the evaluation pipeline, and remove duplicate responses. The resulting candidate set remains fixed throughout probing and adaptation.

For a candidate $c = (c_1, \ldots, c_{|c|})$, the score under Head–Action branch $b = (h, a)$ is the length-normalized sequence log-probability

$$s_b(c) = \frac{1}{|c|} \sum_{t=1}^{|c|} \log p_b(c_t \mid x, c_{<t}),$$

with $s_0(c)$ denoting the corresponding frozen-model score. These are the scores used in the candidate-relative credit $\delta_b(c; y_0)$ defined in Eq. 4.

For each intervention direction, we independently sample three gate strengths, with

$$\mathcal{A}_{\text{down}} \subset [0, 1), \qquad \mathcal{A}_{\text{up}} \subset (1, 2].$$

The sampled strengths are shared across all eligible heads within the same instance. A branch $b = (h, a)$ changes only head h to gate value a, while all remaining gates stay at one. The gate is reset after every branch, so the probes are reversible and non-cumulative.

Per-Head Response Consolidation. Repeated strengths of the same head correspond to different operating points of the same control variable and should not be counted as independent support. We therefore first collapse the probes within each direction:

$$s_h(c, d) = \max_{a \in \mathcal{A}_d} \delta_{(h,a)}(c; y_0), \qquad d \in \{\text{down}, \text{up}\}.$$

For each head–candidate pair, we then retain the better-supported direction:

$$d_h(c) = \arg \max_{d \in \{\text{down}, \text{up}\}} s_h(c, d), \qquad s_h(c) = s_h(c, d_h(c)).$$

Each eligible head therefore contributes at most one support value to a given candidate. Positive responses are then aggregated across distinct heads using $S(c)$ and $N(c)$ in Eq. 5.

Candidate Qualification and Supporting Heads. Selecting the alternative with the largest $S(c)$ alone could trigger an update even when all alternatives are only weakly supported. We therefore use the stable-response threshold from Eq. 9 and qualify only candidates satisfying

$$\mathcal{C}_{\text{qual}}(x) = \{c \in \mathcal{C}(x) \setminus \{y_0\} : S(c) > \gamma,\ N(c) \geq 2\}\,.$$

The support threshold rejects weak preference shifts, while $N(c) \geq 2$ requires agreement from at least two distinct eligible heads. If $\mathcal{C}_{\text{qual}}(x) = \varnothing$, we retain the frozen response y_0 and perform no adaptation.

Otherwise, we select

$$c^* = \arg\max_{c \in \mathcal{C}_{\text{qual}}(x)} S(c), \qquad \mathcal{H}_x = \{h \in \mathcal{H}^* : s_h(c^*) > 0\}, \qquad d_h^* = d_h(c^*).$$

Thus, the selected alternative c^*, its supporting-head set $\mathcal{H}_x$, and the direction d_h^* of each supporting head are all derived from the same candidate-relative Head–Action responses.

Response-Derived Soft Supervision. Although c^* is supported by multiple internal responses, this does not establish that it is correct. We therefore convert the supporting responses into a graded candidate-space distribution rather than treating c^* as a hard pseudo-label.

For each $h \in \mathcal{H}_x$, we retain the tested strength that gives the largest response for c^* in its selected direction:

$$a_h^* = \arg\max_{a \in \mathcal{A}_{d_h^*}} \delta_{(h,a)}(c^*; y_0).$$

Let

$$\mathbf{s}_0 = [s_0(c)]_{c \in \mathcal{C}(x)}$$

denote the frozen candidate-score vector, and define the score residual induced by the selected probe of head h as

$$\Delta\mathbf{s}_h = [s_{(h,a_h^*)}(c) - s_0(c)]_{c \in \mathcal{C}(x)}.$$

Candidate-relative response determines how strongly each supporting head contributes. We use

$$w_h = \frac{[s_h(c^*)]_+}{\sum_{j \in \mathcal{H}_x} [s_j(c^*)]_+}, \qquad \Delta\mathbf{s}^* = \sum_{h \in \mathcal{H}_x} w_h \Delta\mathbf{s}_h.$$

The candidate-relative responses are therefore used for head selection and weighting, while the full score residual preserves how each selected probe redistributes preference over the complete candidate set. This aggregation is used only to construct the supervision signal; the actual joint response of the selected gates is recomputed by the model during optimization.

Using temperature τ, we define

$$q_0 = \text{Softmax}(\mathbf{s}_0/\tau), \qquad q_{\text{res}} = \text{Softmax}((\mathbf{s}_0 + \Delta\mathbf{s}^*)/\tau)\,. \tag{10}$$

Here, q_0 is the frozen-model distribution over $\mathcal{C}(x)$, whereas q_{res} represents the candidate-preference shift implied by the supporting Head–Action responses.

Support-Calibrated Baseline Anchoring. Qualification determines whether adaptation is triggered. For a qualified c^*, we further use its position in the same stable-response distribution $\widehat{F}_s$ to control how strongly q_{res} influences the training target.

Specifically, we set

$$\beta(x) = \beta_{\max} \,\text{clip}\left(\frac{\widehat{F}_s(S(c^*)) - q_\gamma}{1 - q_\gamma}, 0, 1\right), \tag{11}$$

where $\text{clip}(z, 0, 1)$ truncates z to $[0, 1]$ and $\beta_{\max} = 0.5$.

The normalization maps the qualification percentile q_γ to zero and increases the response contribution as the selected candidate moves farther above that level. Substituting $\beta(x)$ into Eq. 6 gives the final target q_T. The frozen distribution therefore remains an explicit anchor even for strongly supported alternatives.

Sparse Test-Time Training. The probes determine which heads are locally supportive and in which directions they should move, but the discrete single-head probe strengths need not be optimal once several heads are modified jointly. We therefore optimize the selected gates while preserving the probe-supported adaptation space.

The candidate set $\mathcal{C}(x)$, selected alternative c^*, supporting heads $\mathcal{H}_x$, directions d_h^*, response weight $\beta(x)$, and target q_T are computed once before optimization and then kept fixed. The target is detached from gradient computation; there is no candidate regeneration, re-probing, supporter reselection, or target recomputation during the update.

Let

$$\mathbf{s}_g = [s_g(c)]_{c \in \mathcal{C}(x)}$$

denote the candidate-score vector under the jointly gated model. Using the same temperature as above,

$$q_g = \mathrm{Softmax}(\mathbf{s}_g/\tau).$$

We optimize only the gates $\{g_h : h \in \mathcal{H}_x\}$ using AdamW, while keeping all remaining gates at one and all pretrained model parameters frozen. The optimization objective is the KL divergence in Eq. 7.

Starting from $g_h = 1$, each gate is projected after every update to preserve its probe-supported direction:

$$g_h \leftarrow \begin{cases} \mathrm{clip}(g_h, 0, 1), & d_h^* = \mathrm{down}, \\ \mathrm{clip}(g_h, 1, 2), & d_h^* = \mathrm{up}. \end{cases}$$

Thus, optimization can determine the joint adjustment magnitudes but cannot reverse the intervention direction identified by the probes. Baseline anchoring is incorporated through q_T rather than through an additional regularization term.

After optimization, the adapted model generates its final response using the same prompt and decoding protocol as the frozen baseline. The candidate set is used only as the supervision space and does not restrict final decoding. All gates are reset to one before the next test instance.

B.4 Default Implementation Settings

Table 6 summarizes the default settings used throughout the main experiments. Global hyperparameters are selected once using the development protocol on LLaVA-1.5-7B and then fixed across tasks, model families, and model scales. Model–task-specific numerical response thresholds are estimated separately from the corresponding unlabeled calibration split.

For task-level eligibility screening, the sampled suppression and amplification strengths are shared across shortlisted heads within the same calibration instance, while each head is probed independently. For instance-level probing, the three sampled strengths in each direction are likewise shared across all eligible heads within the same instance. This provides matched-strength comparisons across heads.

The default screening configuration yields a sparse task-level adaptation space containing approximately 1.6% of the language-model attention heads, with small model–task variations due to finite head counts and percentile ties.

Ablation Implementation Details. For *Random Initial Shortlist*, we replace $\mathcal{H}_{\mathrm{evi}}$ with an equally sized random set of language-model attention heads and keep all subsequent stages unchanged.

For *Matched-Budget Visual-Only*, we set $K = |\mathcal{H}^*_{\mathrm{Full}}|$ for the same model–task pair and use

$$\mathcal{H}_{\text{MB-Visual}} = \mathrm{TopK}_h \, S_{\mathrm{evi}}(h),$$

without using $R_u(h)$ or $R_s(h)$ for selection. All subsequent instance-level components are identical to the full method.

For *w/o Responsiveness Screening*, we set $\mathcal{H}_{\mathrm{resp}} = \mathcal{H}_{\mathrm{evi}}$ and retain the preservation stage. For *w/o Preservation Screening*, we set $\mathcal{H}^* = \mathcal{H}_{\mathrm{resp}}$.

Table 6: **Default implementation settings of TTT-Head.** Task-level eligibility screening uses a single probe strength per direction, whereas candidate-relative support is estimated using multiple strengths.

Setting	Default
Evidence proposal size K_{evi}	Top 10% of architecture-specific spatial units
Local evidence perturbation	Gaussian blur, $\sigma = 20$, kernel size 81×81
Initial visual shortlist	Top 10% of all LM attention heads
Mean-replacement intervention	Same-layer, same-token mean of other heads
Uncertain/stable split α	0.30
Responsiveness threshold η_u	60th percentile of $\{R_u(h) : h \in \mathcal{H}_{\text{evi}}\}$
Preservation threshold η_s	40th percentile of $\{R_s(h) : h \in \mathcal{H}_{\text{resp}}\}$
Open-ended candidate budget K_{cand}	4 alternative prefixes; y_0 added separately
Probe range	suppression $g \in [0, 1)$; amplification $g \in (1, 2]$
Task-level probe sampling	1 strength per direction / instance
Instance-level probe sampling	3 strengths per direction / instance
Stable-reference probing	Same protocol as instance-level probing
Qualification quantile q_γ	0.70
Candidate qualification	$S(c) > \gamma$ and $N(c) \geq 2$
Soft-target temperature τ	0.5
Maximum response weight $\beta_{\max}$	0.5
Supporting heads	All h with $s_h(c^*) > 0$
Gate initialization	$g_h = 1$
Gate optimizer	AdamW
Gate feasible range	$[0, 1]$ for down; $[1, 2]$ for up
Gate projection	Enforced after each update
Final prediction	Baseline decoding protocol with adapted gates
Test-time protocol	Episodic; gates reset per instance

For *Same Heads + Entropy*, we keep the full-method supporting heads, probe-derived directions, gate interface, and projection constraints, but replace $D_{\text{KL}}(q_T \| q_g)$ with entropy minimization $H(q_g)$.

For *Hard c^* Target*, we keep c^*, $\mathcal{H}_x$, and d_h^* unchanged, but replace q_T with the one-hot target $\mathbf{e}_{c^*}$.

For *w/o Candidate-Relative Credit*, we replace the candidate-relative credit in Eq. 4 with the absolute candidate-score change

$$\delta_b^{\text{abs}}(c) = s_b(c) - s_0(c),$$

while keeping subsequent candidate selection, support aggregation, target construction, and optimization unchanged.

For *w/o Baseline Mixing*, we set $\beta(x) = 1$, so that $q_T = q_{\text{res}}$.

For *Direct* $\arg\max q_T$, candidate qualification and target construction are identical to the full method, but no gate optimization is performed. Qualified instances return the candidate in $\mathcal{C}(x)$ with maximum q_T, while unqualified instances retain y_0.

For *Static Rescaling*, each $h \in \mathcal{H}^*$ uses a fixed task-level direction determined from its unlabeled calibration responses. The corresponding gate is fixed to 0.5 for suppression or 1.5 for amplification and is applied unchanged to all held-out instances, without instance-specific Head–Action probing or gradient updates.

For *Joint Probe Gates (No Optimization)*, we retain the full-method c^*, $\mathcal{H}_x$, d_h^*, and a_h^*, jointly set $g_h = a_h^*$ for all $h \in \mathcal{H}_x$, and decode without gradient updates.

For *w/o Direction Constraint*, we retain c^*, $\mathcal{H}_x$, and q_T, but optimize the gates subject only to $0 \leq g_h \leq 2$.

For *Update All Eligible Heads*, c^*, the original $\mathcal{H}_x$ used to construct q_T, and q_T itself remain identical to the full method; only the trainable set is expanded from $\mathcal{H}_x$ to $\mathcal{H}^*$. Each additional

$h \in \mathcal{H}^* \setminus \mathcal{H}_x$ uses its candidate-specific direction

$$d_h(c^*) = \arg \max_{d \in \{\text{down}, \text{up}\}} s_h(c^*, d)$$

as its direction constraint.

B.5 Comparison Method Details

We implement all comparison methods following their original optimization objectives and method-specific procedures, while adapting the trainable parameter interface where necessary for transformer-based VLMs. We do not replace their native objectives or intervention mechanisms with those of TTT-Head.

For fair comparison on the target tasks, all methods use the same held-out images, questions, sample IDs, benchmark prompts described in Appendix A.5, decoding settings, and evaluation pipeline described in Appendix A.3. Method-specific auxiliary prompts, augmentations, or internal signals that form part of the original algorithms are retained.

Entropy-Based Test-Time Methods. Entropy Minimization (ENT) follows the entropy-minimization objective of TENT (Wang et al., 2021). For our transformer-based VLMs, entropy minimization updates the learnable scale parameters of RMSNorm layers while keeping the remaining model parameters frozen.

SAR (Niu et al., 2023) uses the same trainable RMSNorm parameter scope and follows its reliable-sample filtering and sharpness-aware entropy-minimization procedure. Thus, ENT and SAR retain their entropy-based training objectives and normalization-level update interface rather than being reimplemented over the TTT-Head head gates.

Multimodal Test-Time Methods. TTW (Rajaneesh et al., 2025) is implemented following its original per-instance warmup procedure. The vision encoder remains frozen, while the language model and multimodal connector are trained using cross-entropy supervision derived from the method-specific auxiliary responses for the current image. The auxiliary prompts and weakly supervised filtering procedure follow the original TTW setting, whereas the final benchmark question uses the same task prompt as all other methods. Updated parameters are discarded after inference on the current test instance.

TTAdapt (Kaya et al., 2026) follows its original iterative consensus-based training procedure. Pseudo-labels are generated using TTAug consensus and are used as supervision for parameter updates with the objective and trainable parameter configuration specified by the original method. The model is reset to its initial state before processing the next test instance. The target benchmark question itself uses the same task prompt and evaluation protocol as TTT-Head and the other comparison methods.

Head-Level Intervention Methods. SPIN (Sarkar et al., 2025) is implemented using its original image-attention-guided head-suppression rule. For each query token, heads with weaker attention to visual tokens are suppressed according to the method's prescribed selection rule, without gradient-based parameter updates. We retain this native inference-time intervention rather than replacing it with the trainable gates used by TTT-Head.

V-SEAM (Wang et al., 2025) follows its original supervised head-discovery and modulation procedure. Positive and negative attention heads are identified using labeled analysis examples and are subsequently modified according to the original modulation rule. Because its head discovery requires ground-truth supervision, V-SEAM is reported separately as a supervised reference and is excluded from the ranking of label-free methods.

Frozen Reference. The frozen VLM uses the same checkpoint, benchmark prompts, decoding settings, held-out images, questions, and evaluation samples as the test-time methods, but performs no parameter update or head intervention.

C Additional Analysis of Adaptation-Space Discovery

This section provides additional analyses of the task-level adaptation-space discovery in TTT-Head. We focus on three complementary questions. First, we extend the preliminary observation in the main paper across additional model families and scales to examine why visual relevance alone is insufficient for identifying heads suitable for test-time training. Second, we study the sensitivity of response-based head screening to the preservation criterion and the resulting adaptation-space size. Third, we compare the label-free adaptation space discovered by TTT-Head with a label-based oracle to examine how closely label-free discovery recovers heads that are truly useful for correction.

C.1 Visual Relevance Does Not Guarantee Adaptation Effectiveness

The preliminary analysis in the main paper shows that heads functionally associated with task-relevant visual evidence do not necessarily exhibit effective intervention behavior. Here, we extend this analysis from the primary LLaVA-1.5-7B setting to the other four model configurations. Specifically, we evaluate LLaVA-1.5-13B, InstructBLIP-Vicuna-7B/13B, and Qwen2.5-VL-7B, covering both model-scale and architecture variations.

For each model and task, we first retain the top 10% of attention heads according to the visual-evidence relevance score used in the first stage of TTT-Head. We use this relatively broad shortlist as an initial candidate pool, retaining a diverse set of visually associated heads while substantially reducing the cost of subsequent head-wise intervention probing. The 10% ratio is fixed across all models and tasks and is not tuned using ground-truth labels; the final adaptation-eligible head pool is determined only after the subsequent response-based screening.

This initial selection is therefore entirely label-free. We then use ground-truth answers only for this post-hoc diagnostic analysis. Each visually relevant head is reversibly intervened on while the remaining heads are kept unchanged. For each head, we evaluate its intervention behavior on two complementary subsets. On instances that are initially answered incorrectly by the frozen model, we measure the corrective response induced by the intervention toward the ground-truth answer. On instances that are initially answered correctly, we measure the extent to which the intervention disrupts the original correct prediction. We aggregate the corrective and disruptive effects separately over the two subsets and categorize heads accordingly. *Relevant but Insensitive* heads exhibit little corrective response on initially incorrect instances; *Disruptive under Intervention* heads substantially disturb initially correct predictions; and *Effective under Intervention* heads exhibit a substantial corrective response while maintaining low disruption on initially correct instances. Ground-truth labels are used only for this diagnostic categorization and are never accessed by the actual label-free discovery procedure of TTT-Head.

As shown in Figure 5, the same qualitative pattern extends across architectures and model scales. In the primary LLaVA-1.5-7B diagnostic across GQA, AMBER, and MMStar, the three categories account for approximately 63% insensitive, 12% disruptive, and 25% effective heads on average. The large insensitive fraction is consistent with the need for an additional responsiveness-screening stage, while the disruptive fraction further supports screening for preservation on stable instances. These post-hoc proportions are used only for interpretation and do not determine the responsiveness or preservation percentile levels, which are selected separately on the labeled development set and fixed before evaluation. Overall, only a subset of visually relevant heads combines meaningful corrective responses with low disruption to already-correct predictions. These results further show that visual relevance alone is insufficient for identifying reliable adaptation targets, supporting the subsequent response-based screening in TTT-Head.

C.2 Sensitivity to Response-Based Head Screening

We examine sensitivity to the preservation criterion in response-based head screening. Starting from the fixed top-10% visually relevant shortlist, we keep the responsiveness threshold at the 60th percentile and vary the preservation percentile $p_s \in \{10, 20, \ldots, 100\}$, corresponding nominally to adaptation-space ratios of 0.4%, 0.8%, …, 4.0% of all language-model attention heads.

This study is conducted on a shared 500-case analysis set, denoted $\mathcal{D}_{\mathrm{ana}}$, that is separate from the main calibration and held-out evaluation pools. The same analysis set is used for the probe-

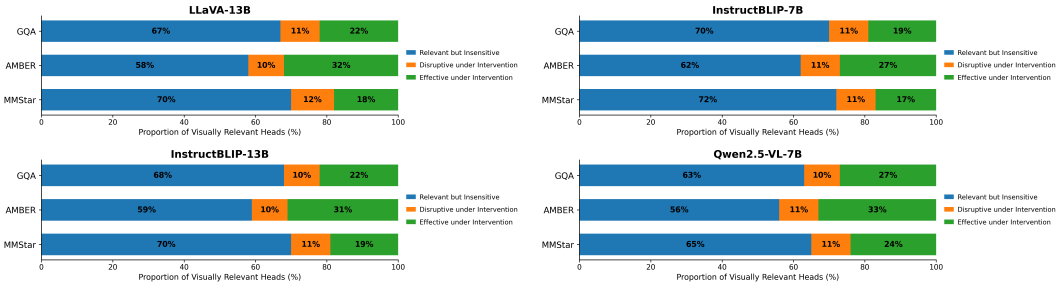

Figure 5: **Visual relevance alone does not guarantee adaptation effectiveness across architectures and model scales.** For each model, we first identify the top 10% visually relevant heads using the label-free evidence-localization stage of TTT-Head. Ground-truth answers are used only for post-hoc diagnosis of whether bounded head interventions are effective, insensitive, or disruptive.

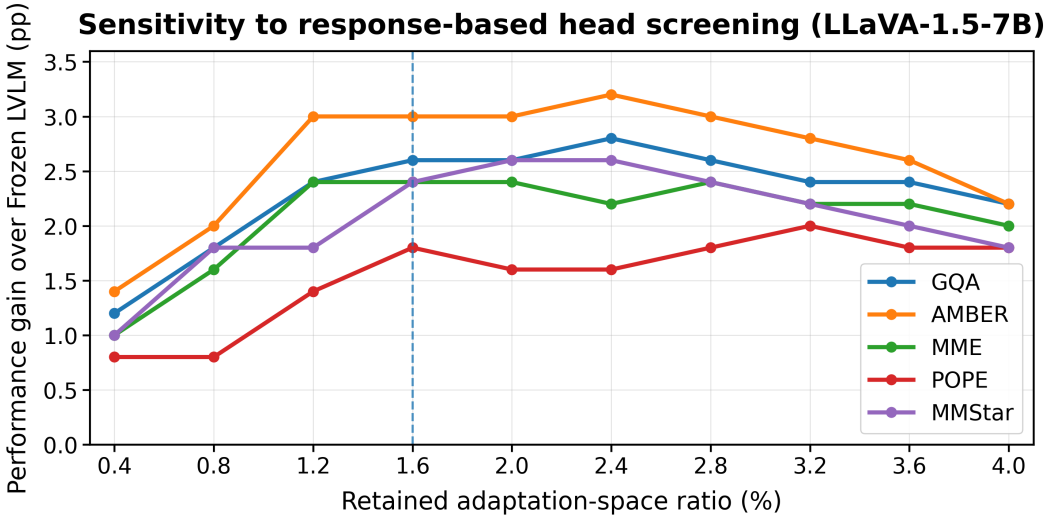

Figure 6: **Sensitivity to response-based head screening on LLaVA-1.5-7B.** Gains improve as overly restrictive adaptation spaces are relaxed, while larger pools show diminishing returns and occasional degradation. The dashed line marks the nominal 1.6% default configuration.

sensitivity study in Appendix D.2. For each screening configuration, pool-dependent calibration statistics, including the stable-response reference $\widehat{F}_s$ and qualification threshold γ, are re-estimated using the same calibration rule, while $q_\gamma = 0.70$ and all other hyperparameters remain fixed.

The default configuration uses $p_s = 40$, giving a nominal retained-head ratio of

$$0.10 \times (1 - 0.60) \times 0.40 = 0.016,$$

or approximately 1.6%. We evaluate LLaVA-1.5-7B, InstructBLIP-7B, and Qwen2.5-VL-7B on five representative benchmarks and report gains over the corresponding frozen VLM. The resulting sensitivity curves are shown in Figures 6, 7, and 8.

As shown in Figures 6– 8, overly strict preservation screening generally underperforms because the available adaptation space is too small. Relaxing the criterion toward the default configuration substantially improves performance.

Beyond the default setting, gains generally enter a broad plateau: moderately larger pools can help some tasks, whereas more permissive thresholds can also introduce mild degradation. Thus, the default 1.6% configuration provides a sparse operating point that captures a large fraction of the attainable gain while limiting the cost of subsequent instance-level probing and adaptation.

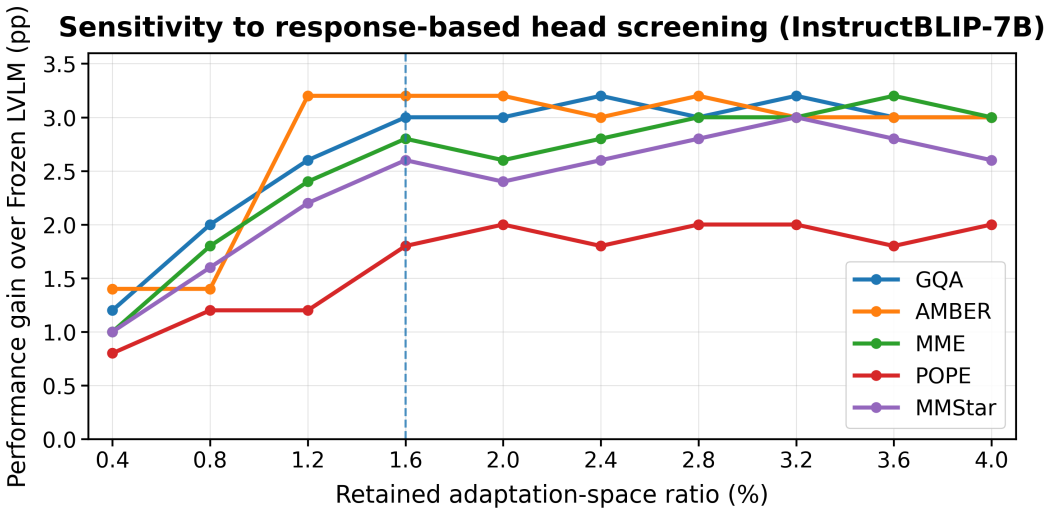

Figure 7: **Sensitivity to response-based head screening on InstructBLIP-7B.** The default configuration captures a large fraction of the attainable gain, while further relaxation yields small, task-dependent changes. The dashed line marks the nominal 1.6% default configuration.

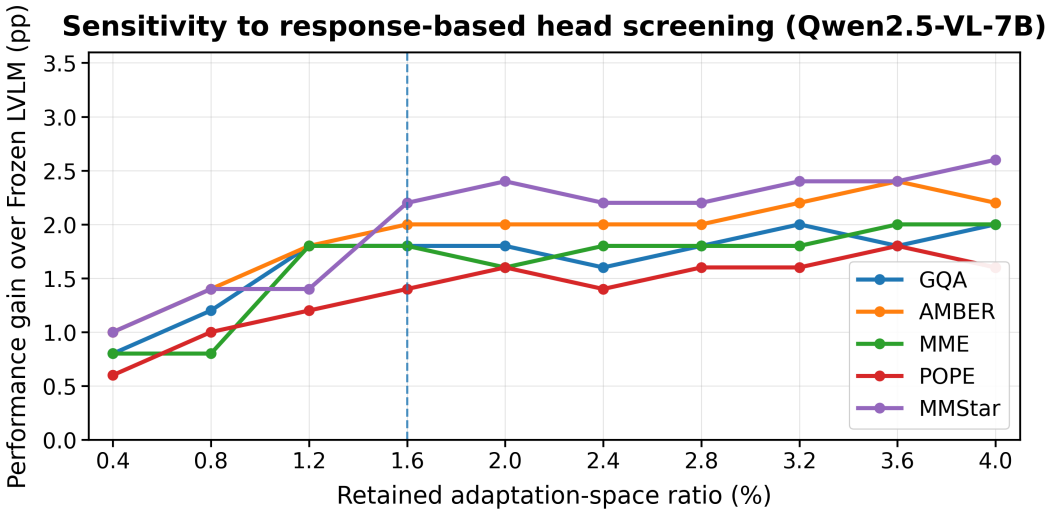

Figure 8: **Sensitivity to response-based head screening on Qwen2.5-VL-7B.** Moderate relaxation can provide additional gains on some tasks, but larger adaptation spaces exhibit diminishing returns and fluctuations. The dashed line marks the nominal 1.6% default configuration.

C.3 Agreement with Label-Based Head Discovery

We finally examine whether the label-free adaptation space discovered by TTT-Head agrees with heads identified using direct ground-truth corrective feedback. For each model–benchmark pair, the label-free set $\mathcal{H}_{\mathrm{LF}}$ is exactly the default TTT-Head pool obtained from the top-10% visually relevant shortlist, followed by the 60th-percentile responsiveness filter and the 40th-percentile preservation filter. The resulting pool is nominally about 1.6% of all language-model attention heads, with small realized variations due to finite head counts and percentile ties.

For the label-based branch, we start from the same visual shortlist $\mathcal{H}_{\mathrm{evi}}$ and use ground-truth answers only for this diagnostic analysis. Specifically, for each candidate head and intervention direction, the label-based diagnostic directly evaluates how the intervention changes the model's preference for the ground-truth answer. Heads are ranked according to this supervised corrective utility, and the

top-K heads form the label-based oracle set $\mathcal{H}_{\mathrm{GT}}$, where $K = |\mathcal{H}_{\mathrm{LF}}|$ matches the realized size of the default label-free pool.

We measure their agreement using oracle-head recovery,

$$\mathrm{Recovery} = \frac{|\mathcal{H}_{\mathrm{LF}} \cap \mathcal{H}_{\mathrm{GT}}|}{|\mathcal{H}_{\mathrm{GT}}|}. \tag{12}$$

For two independently selected sets of the same realized size K from the shared visual shortlist, the expected random recovery is

$$\mathbb{E}[\mathrm{Recovery}_{\mathrm{rand}}] = \frac{K}{|\mathcal{H}_{\mathrm{evi}}|}. \tag{13}$$

The random expectation is computed separately for each model–benchmark pair using the realized label-free pool size and the corresponding model-specific visual shortlist size. Under the default screening configuration, these values remain close to 16%. The resulting recovery rates and random expectations are reported in Table 7.

Table 7: **Agreement between label-free and label-based head discovery.** Values report oracle-head recovery (%). For each model–benchmark pair, the corresponding random baseline gives the expected recovery of an equally sized random head set sampled independently from the same visual shortlist.

Model	GQA	AMBER	MME	POPE	MMStar	Avg.
LLaVA-1.5-7B	75.0	81.3	75.0	68.8	62.5	72.5
Random expectation	15.7	15.7	15.7	15.7	15.7	15.7
InstructBLIP-7B	70.6	76.5	68.8	62.5	58.8	67.4
Random expectation	16.7	16.7	15.7	15.7	16.7	16.3
Qwen2.5-VL-7B	76.9	84.6	76.9	69.2	61.5	73.8
Random expectation	16.7	16.7	16.7	16.7	16.7	16.7

As shown in Table 7, TTT-Head recovers a substantial fraction of the heads selected using direct ground-truth corrective feedback. Across the three model architectures, the average recovery ranges from 67.4% to 73.8%, substantially above the corresponding random expectations, which average approximately 16.2% across the evaluated model–benchmark pairs. This agreement indicates that, within the shared top-10% visually relevant candidate pool used by both procedures, the label-free responsiveness and preservation signals recover many heads that also exhibit supervised corrective utility, despite having no access to ground-truth answers during actual discovery. The result therefore measures agreement within this controlled candidate space rather than global optimality over all attention heads.

D Additional Analysis of Instance-Specific Test-Time Training

This section provides four complementary analyses of the instance-specific test-time training stage. We first clarify why TTT-Head measures intervention responses relative to the frozen-model answer rather than using absolute candidate-score changes. We then examine the sensitivity of TTT-Head to the number of intervention strengths used during probing. Next, we use a supervised oracle-target diagnostic to separate limitations of label-free target estimation from limitations of the discovered adaptation space. Finally, we analyze correction and disruption behavior to examine how baseline anchoring affects overcorrection.

D.1 Candidate-Relative Supervision Construction

The instance-level probes are intended to identify interventions that shift the model's preference away from the frozen response toward an alternative candidate. Measuring only the absolute score change of a candidate is insufficient for this purpose, because the same intervention may also change the score of the frozen response.

For a test instance, let y_0 denote the frozen response and $c \in \mathcal{C}(x)$ an alternative candidate. TTT-Head therefore measures the intervention-induced change in their relative score margin:

$$\delta_b(c; y_0) = \big[s_b(c) - s_b(y_0)\big] - \big[s_0(c) - s_0(y_0)\big], \tag{14}$$

where $s_0(\cdot)$ and $s_b(\cdot)$ denote candidate scores before and after intervention branch b, respectively.

Equivalently,

$$\delta_b(c; y_0) = \big[s_b(c) - s_0(c)\big] - \big[s_b(y_0) - s_0(y_0)\big]. \tag{15}$$

This decomposition shows that candidate-relative credit discounts response changes that affect the alternative and the frozen response similarly. A common shift of equal magnitude contributes zero credit, whereas positive credit indicates that the intervention improves the candidate–frozen-response margin.

Importantly, $\delta_b(c; y_0) > 0$ measures intervention response rather than correctness: it indicates that branch b makes c more competitive relative to y_0, but does not require c to become the top-ranked candidate or imply that c is correct. TTT-Head therefore aggregates such responses across distinct heads and requires the aggregated support to exceed the calibrated stable-response threshold before adaptation is triggered, using the responses as graded supervision rather than hard pseudo-labels.

D.2 Sensitivity to Intervention Probing

Intervention probing introduces additional forward passes as the number of sampled strengths increases. We therefore evaluate 1, 3, 5, and 10 probe strengths per direction on the shared 500-case analysis set $\mathcal{D}_{\text{ana}}$ defined in Appendix C.2, while keeping the adaptation space, candidate construction, and training objective fixed.

To reduce sampling variation, we use a nested-prefix protocol. For each instance and direction, we sample a fixed ordered sequence of ten strengths, with $A_1 \subset A_3 \subset A_5 \subset A_{10}$. For each probe budget n, we re-estimate the corresponding stable-response reference $\widehat{F}_s^{(n)}$ and qualification threshold $\gamma^{(n)}$ using the same calibration rule, while keeping $q_\gamma = 0.70$ and all other hyperparameters fixed. Table 8 reports the resulting performance across probe budgets.

Table 8: **Sensitivity to the number of probe strengths per direction.** Results are reported on the shared analysis set $\mathcal{D}_{\text{ana}}$ using LLaVA-1.5-7B. All settings use nested prefixes of the same probe sequence; bold denotes the default configuration.

# Probe Samples / Direction	GQA	AMBER	MMStar
Frozen reference	57.80	72.60	30.20
1	59.00	74.40	31.60
3 (Default)	**60.40**	**75.60**	**32.60**
5	60.40	75.60	32.80
10	60.40	75.40	32.80

Increasing the probe budget from one to three provides clear gains across all three benchmarks. Increasing it further to five or ten yields at most 0.2 additional points and no consistent improvement. This sensitivity analysis confirms that three probes per direction capture most of the attainable gain while avoiding the additional inference cost of larger probe budgets.

D.3 Oracle-Target Diagnostic

To examine the remaining headroom in the label-free corrective signal while holding the discovered adaptation space fixed, we keep the TTT-Head head pool and sparse gate parameterization unchanged and replace the baseline-anchored label-free target with ground-truth supervision. This variant is used only as a supervised diagnostic and is not part of TTT-Head's test-time protocol. The corresponding results are reported in Table 9.

Replacing TTT-Head's label-free target with ground-truth supervision improves GQA, AMBER, and MMStar by 1.0, 0.7, and 1.2 points, respectively. The additional improvement indicates remaining headroom in the label-free target supervision within the same discovered head pool and sparse gate

Table 9: **Oracle-target diagnostic.** The discovered head pool and sparse gate parameterization are retained, while the label-free target is replaced by ground-truth supervision.

Variant	GQA	AMBER	MMStar
TTT-Head (label-free)	61.20	76.70	32.40
TTT-Head + GT Target‡	62.20	77.40	33.60

parameterization. This experiment is therefore intended as a supervised diagnostic rather than an upper bound on the full adaptation pipeline.

D.4 Correction–Disruption Analysis

We further examine how baseline mixing affects the balance between correcting initially wrong predictions and disrupting predictions that are already correct. This analysis is conducted on the same 1,000 held-out AMBER instances used in the main evaluation with LLaVA-1.5-7B, comparing full TTT-Head with the *w/o Baseline Mixing* variant.

For each instance, we compare the frozen and adapted predictions against the ground-truth answer. A *correction* ($\mathrm{W} \rightarrow \mathrm{C}$) denotes an instance that is incorrect under the frozen model but correct after test-time training, whereas a *disruption* ($\mathrm{C} \rightarrow \mathrm{W}$) denotes an initially correct instance that becomes incorrect after test-time training. Their difference determines the corresponding accuracy improvement over the frozen model:

$$\Delta\mathrm{Acc} = 100\,\frac{N_{\mathrm{W}\rightarrow\mathrm{C}} - N_{\mathrm{C}\rightarrow\mathrm{W}}}{N}. \tag{16}$$

The resulting correction and disruption counts are summarized in Table 10.

Table 10: **Correction–disruption analysis on AMBER.** Counts are computed over the same 1,000 held-out instances used in the main evaluation. $\mathrm{W} \rightarrow \mathrm{C}$ denotes corrected frozen-model errors, whereas $\mathrm{C} \rightarrow \mathrm{W}$ denotes disruptions of initially correct predictions.

Variant	$\mathrm{W} \rightarrow \mathrm{C} \uparrow$	$\mathrm{C} \rightarrow \mathrm{W} \downarrow$	Net Gain (pp) $\uparrow$
w/o Baseline Mixing	60	38	+2.20
TTT-Head	50	18	**+3.20**

Removing baseline mixing produces more corrections, but also substantially more disruptions of predictions that were already correct. Full TTT-Head makes fewer corrective flips (50 versus 60) while reducing disruptive flips from 38 to 18, yielding a larger net improvement. This behavior is consistent with the role of baseline anchoring in moderating aggressive response-supported updates: it sacrifices a small number of additional corrections while preventing a larger number of overcorrections.

E Supplementary Experimental Results

This section provides supplementary experimental results that complement the main comparison. We first report the complete results on the four additional model configurations summarized in the main paper, and then provide a fine-grained breakdown of POPE across its three hallucination subsets. The complete comparison on the primary LLaVA-1.5-7B backbone is reported in the main paper and is therefore not repeated here.

E.1 Complete Results Across Models

Tables 11, 12, 13, and 14 report the complete comparison on LLaVA-1.5-13B, InstructBLIP-Vicuna-7B/13B, and Qwen2.5-VL-7B. For each model–benchmark pair, all methods are evaluated using the same held-out samples, benchmark prompts, decoding protocol, and evaluation pipeline.

Table 11: **Complete results on LLaVA-1.5-13B.**

Method	GQA	COCO-QA	POPE	AMBER	MME-1K	MMStar	TextVQA	Avg.
Frozen VLM	63.10	78.30	87.00	80.20	76.20	33.60	55.98	67.77
ENT	63.80	78.70	87.10	80.30	76.20	33.80	56.02	67.99
SAR	63.80	79.20	87.30	80.10	76.80	34.00	56.00	68.17
TTW	64.10	79.00	87.80	80.70	76.50	34.00	56.42	68.36
TTAdapt	64.60	79.50	87.90	81.60	76.90	34.80	56.80	68.87
SPIN	63.70	78.40	88.40	81.90	76.60	34.00	56.08	68.44
TTT-Head (Ours)	**65.50**	**80.70**	**88.60**	**82.90**	**78.50**	**36.40**	**58.38**	**70.14**

Table 12: **Complete results on InstructBLIP-Vicuna-7B.**

Method	GQA	COCO-QA	POPE	AMBER	MME-1K	MMStar	TextVQA	Avg.
Frozen VLM	49.10	73.50	82.10	69.90	69.40	27.80	49.46	60.18
ENT	49.50	74.20	82.40	70.00	69.60	28.00	49.38	60.44
SAR	49.90	74.00	82.60	70.30	69.60	28.40	49.70	60.64
TTW	49.60	74.50	82.70	71.10	69.90	28.60	49.84	60.89
TTAdapt	50.70	74.80	83.40	71.30	70.60	28.80	50.56	61.45
SPIN	49.20	74.10	84.00	71.10	69.30	28.40	49.38	60.78
TTT-Head (Ours)	**52.30**	**76.10**	**84.10**	**73.20**	**72.10**	**30.60**	**52.26**	**62.95**

Table 13: **Complete results on InstructBLIP-Vicuna-13B.**

Method	GQA	COCO-QA	POPE	AMBER	MME-1K	MMStar	TextVQA	Avg.
Frozen VLM	49.30	74.10	83.10	76.80	73.20	29.80	50.52	62.40
ENT	49.90	74.30	83.40	76.40	73.50	30.00	50.38	62.55
SAR	49.80	74.80	83.30	77.20	73.60	30.60	50.56	62.84
TTW	50.20	74.50	83.90	76.70	74.00	30.20	50.92	62.92
TTAdapt	50.60	75.70	84.00	78.20	74.30	31.00	51.60	63.63
SPIN	49.90	74.10	84.20	78.80	73.70	30.00	50.70	63.06
TTT-Head (Ours)	**51.70**	**76.20**	**84.80**	**79.50**	**75.60**	**32.40**	**52.82**	**64.72**

Table 14: **Complete results on Qwen2.5-VL-7B.**

Method	GQA	COCO-QA	POPE	AMBER	MME-1K	MMStar	TextVQA	Avg.
Frozen VLM	63.50	77.20	86.60	81.10	79.20	62.00	82.06	75.95
ENT	63.60	77.40	86.50	81.20	79.30	62.40	82.10	76.07
SAR	64.00	77.30	86.50	81.40	79.40	62.20	82.16	76.14
TTW	63.80	77.80	87.00	81.30	79.80	62.40	82.34	76.35
TTAdapt	64.70	77.90	87.70	81.80	80.30	62.80	82.70	76.84
SPIN	64.00	77.40	87.60	82.00	79.00	62.80	82.06	76.41
TTT-Head (Ours)	**65.30**	**79.20**	**88.20**	**83.30**	**80.90**	**64.40**	**83.66**	**77.85**

Across all four additional model configurations, TTT-Head improves over the corresponding frozen VLM on every benchmark and achieves the highest seven-benchmark average for each backbone. Together with the primary LLaVA-1.5-7B results in Table 1, these results show that the gains of sparse head-level test-time training extend across three model families and two model scales.

E.2 Fine-Grained POPE Results

POPE contains three complementary subsets—Random, Popular, and Adversarial—which differ in how negative objects are selected. Random sampling draws absent objects without additional bias, Popular sampling favors frequently occurring objects, and Adversarial sampling selects challenging negative objects that are more likely to be confused with the visual context. To examine whether the aggregate POPE result is dominated by a particular sampling regime, we additionally report performance on the three subsets separately. The POPE evaluation subset is evenly balanced between `yes` and `no` ground-truth labels, so accuracy gains cannot be attributed to majority-label imbalance. The subset-level comparison is reported in Table 15.

Table 15: **Fine-grained results on the three POPE subsets for LLaVA-1.5-7B.** The 1K evaluation subset contains 333 Random, 333 Popular, and 334 Adversarial instances. Aggregate denotes the POPE score reported in the main comparison.

Method	Random	Popular	Adversarial	Aggregate
Frozen VLM	86.19	84.98	83.53	84.90
TTAdapt	87.39	86.19	85.03	86.20
SPIN	**88.59**	**86.79**	**85.63**	**87.00**
TTT-Head (Ours)	87.69	86.49	85.33	86.50

The fine-grained results show a consistent difficulty trend across the three POPE subsets, with performance decreasing from Random to Popular and further to Adversarial. TTT-Head improves over the frozen VLM on all three subsets, by 1.50, 1.50, and 1.80 points on Random, Popular, and Adversarial, respectively. Thus, its aggregate improvement is not driven by a single POPE sampling regime. SPIN remains slightly stronger than TTT-Head on all three subsets, consistent with the aggregate POPE result in Table 1.

F Supplementary Head-Level Analyses

The main paper identifies two recurring head-level patterns on LLaVA-1.5-7B and InstructBLIP-7B: suppressive Head–Actions are more frequent among the supporting actions in successful corrections, and adaptation-eligible heads exhibit task-dependent depth distributions. Here, we extend the same analysis to the remaining three model configurations: LLaVA-1.5-13B, InstructBLIP-13B, and Qwen2.5-VL-7B.

For the directional analysis, each observation is a supporting head–instance pair from an instance that is incorrect before adaptation and correct afterward; mixed-direction instances contribute separately according to each supporting head's selected direction. For the depth analysis, we map head layers to relative depth and divide the range into eight equal bins. Each benchmark distribution is normalized by its discovered pool size before equal-weight averaging within each task group. Figures 9, 10, and 11 report the resulting analyses.

Across all three additional model configurations, suppressive directions account for the majority of supporting Head–Actions in successful corrections on GQA, AMBER, and MMStar. Although the exact proportions vary across tasks and backbones, the same directional asymmetry is consistently observed. This pattern suggests that attenuation is a common adjustment among supporting heads, while amplification remains present in a substantial minority of successful corrections.

The relative-depth distributions show a similarly consistent task-dependent pattern. Within each backbone, selection-oriented tasks shift toward earlier relative depths, whereas open-ended and OCR-oriented tasks shift later. Together with the main-paper results, this suggests that test-time adaptation may benefit from discovering task-specific adaptation regions rather than relying on a fixed layer range across tasks.

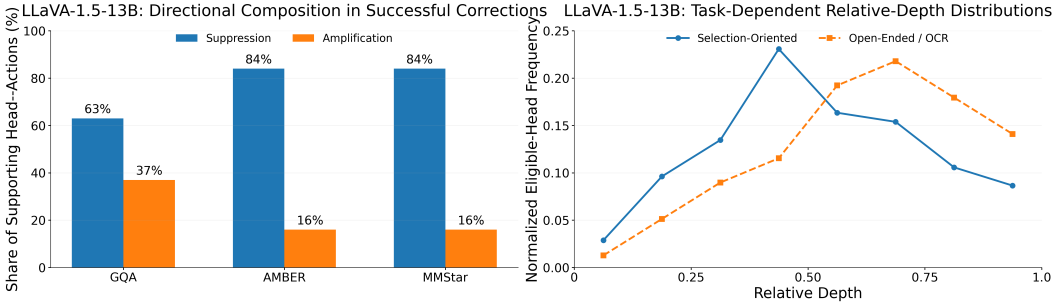

Figure 9: **Head-level findings on LLaVA-1.5-13B. Left:** directional composition of supporting Head–Actions in successful corrections on GQA, AMBER, and MMStar. **Right:** 8-bin relative-depth distributions of task-level adaptation-eligible heads, normalized per benchmark and equally weighted within each task group.

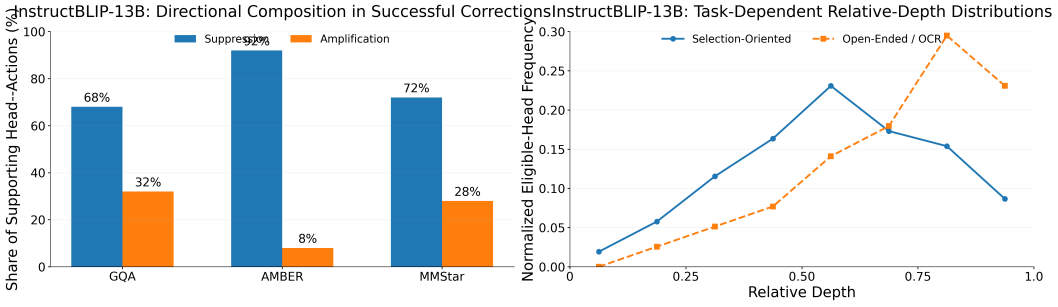

Figure 10: **Head-level findings on InstructBLIP-13B. Left:** directional composition of supporting Head–Actions in successful corrections on GQA, AMBER, and MMStar. **Right:** 8-bin relative-depth distributions of task-level adaptation-eligible heads, normalized per benchmark and equally weighted within each task group.

F.1 Illustrative Suppressive and Amplifying Corrections

We next provide two representative instance-level case studies illustrating how opposite corrective directions can both lead to successful adaptation, and why retaining both directions can be useful rather than imposing a fixed suppression-only policy. Figures 12 and 13 show the suppressive and amplifying examples, respectively. The first case shows a *suppressive correction*, in which later-layer semantic readouts initially favor baseline-consistent signals; suppressing the selected heads makes answer-consistent semantics more prominent. The second case shows an *amplifying correction*, in which relevant concepts are already visible in intermediate-layer readouts but are not dominant in the final prediction; amplifying the selected heads strengthens these answer-consistent signals.

In both examples, panel (a) shows the input instance, question, baseline prediction, and ground truth; panel (b) shows layer-wise top-5 post-hoc semantic readouts before and after adaptation; and panel (c) shows the corresponding candidate probabilities. To obtain the layer-wise semantic readouts in panel (b), we apply a logit-lens-style analysis at the final prompt position, whose hidden state is used to predict the first answer token. For each layer, the corresponding hidden state is passed through the model's final normalization and frozen language-model output head, and the top-5 vocabulary tokens are retained for visualization. This analysis is used only for post-hoc interpretation and does not participate in TTT-Head.

G Statistical Significance

We further assess whether the performance improvements of TTT-Head are robust to finite-sample variation on the fixed held-out evaluation sets. We consider five representative benchmarks spanning

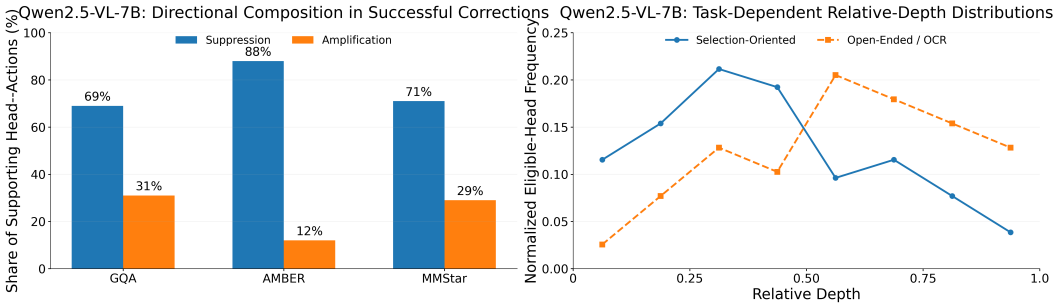

Figure 11: **Head-level findings on Qwen2.5-VL-7B. Left:** directional composition of supporting Head–Actions in successful corrections on GQA, AMBER, and MMStar. **Right:** 8-bin relative-depth distributions of task-level adaptation-eligible heads, normalized per benchmark and equally weighted within each task group.

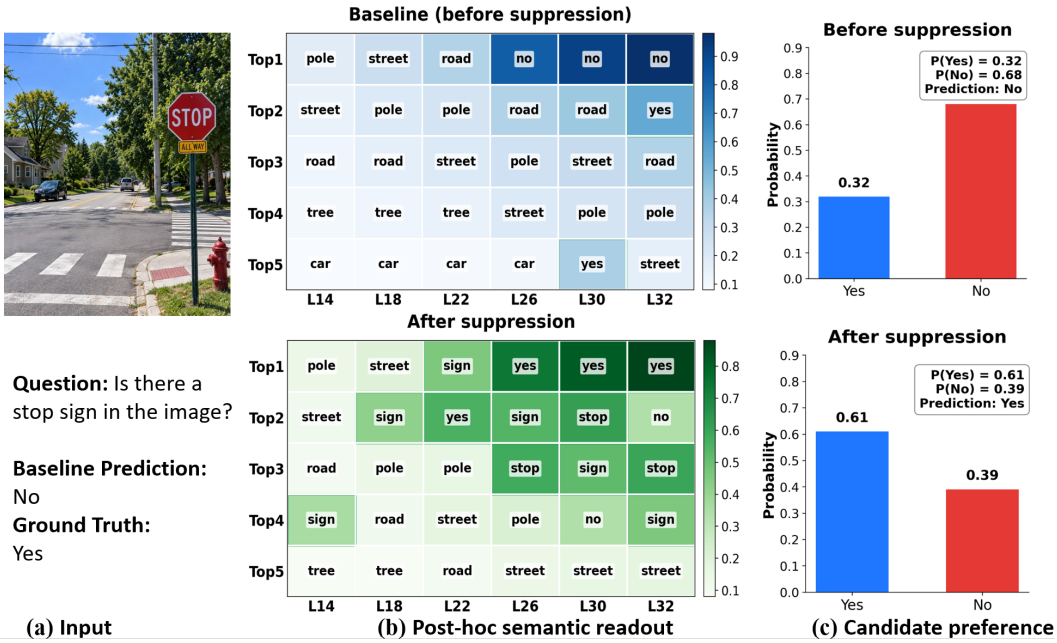

Figure 12: **Illustrative suppressive correction. (a) Input.** The baseline model predicts *No* for the question "Is there a stop sign in the image?", while the ground truth is *Yes*. **(b) Post-hoc semantic readout.** Before adaptation, later-layer readouts are dominated by the baseline-favoring token *no*, with only limited answer-consistent evidence such as *yes* visible in the displayed top-5 readouts. After jointly suppressing the selected heads, answer-consistent concepts such as *sign*, *stop*, and *yes* become more prominent in later layers. **(c) Candidate preference.** The candidate probability shifts from *No* to *Yes*, correcting the prediction. This case suggests that suppression can improve performance by reducing competing internal contributions and making answer-consistent semantics more prominent downstream.

open-ended VQA, hallucination-sensitive visual understanding, multimodal perception and reasoning, and scene-text understanding.

Because TTT-Head and the frozen VLM are evaluated on exactly the same held-out examples, we use paired bootstrap resampling with $B = 10{,}000$ resamples. For each bootstrap sample, the benchmark scores are recomputed from the paired predictions on the same resampled examples, and the corresponding performance difference is recorded. When multiple questions are associated with the same image, the image is used as the resampling unit and all corresponding questions are sampled

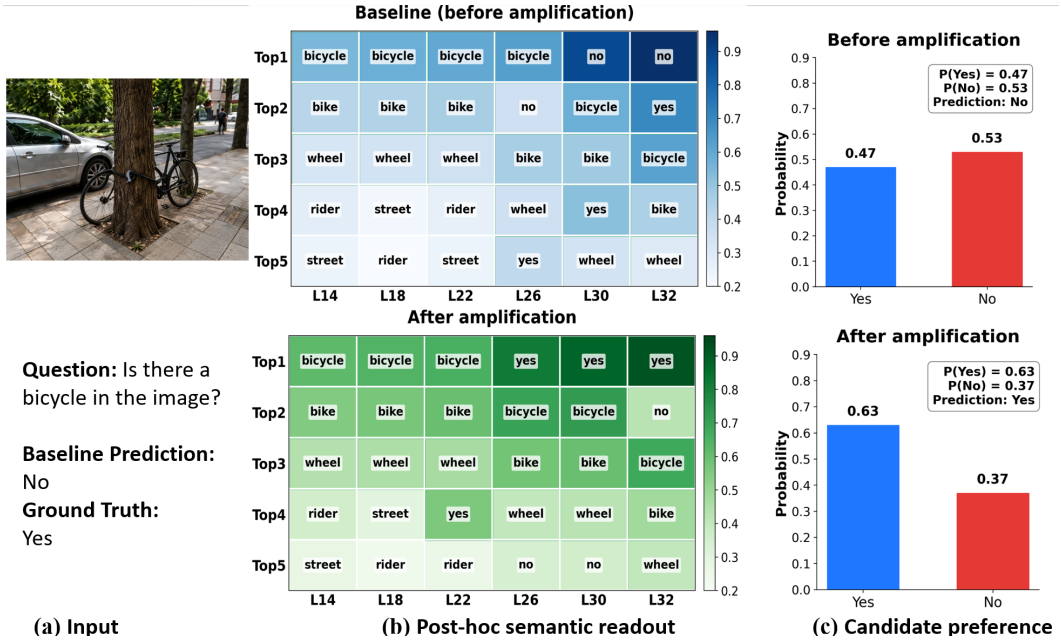

Figure 13: **Illustrative amplifying correction. (a) Input.** The baseline model predicts *No* for the question "Is there a bicycle in the image?", while the ground truth is *Yes*. **(b) Post-hoc semantic readout.** Before adaptation, bicycle-related concepts such as *bicycle*, *bike*, and *wheel* are already present, indicating that relevant visual semantics have been partially extracted but remain under-expressed in the final answer path. After jointly amplifying the selected heads, later-layer readouts increasingly favor answer-consistent tokens such as *yes*. **(c) Candidate preference.** The candidate probability shifts from *No* to *Yes*, correcting the prediction. This case suggests that amplification can improve performance by strengthening a useful but under-expressed semantic pathway.

Table 16: **Statistical significance of TTT-Head over the frozen LLaVA-1.5-7B on five representative benchmarks.** Δ denotes the observed performance improvement in percentage points on the same held-out evaluation subsets used in the main experiments. Significance is estimated using 10,000 paired image-clustered bootstrap resamples. $^{***}p < 0.001$.

Benchmark	Δ (pp)	Sig.
GQA Open	+2.80	***
AMBER	+3.20	***
MME-1K	+2.40	***
MMStar	+2.80	***
TextVQA	+3.12	***

jointly. Statistical significance is assessed from the empirical distribution of the paired performance differences. The observed gains and significance results are summarized in Table 16.

As shown in Table 16, TTT-Head consistently improves over the frozen VLM across all five representative benchmarks, with all paired improvements reaching statistical significance ($p < 0.001$). The observed gains range from 2.40 to 3.20 percentage points, with an average improvement of 2.86 points across the five benchmarks. These results indicate that the gains remain stable under resampling of the fixed evaluation subsets and are unlikely to be explained by finite-sample variation alone.

H Efficiency Analysis

We report online runtime on LLaVA-1.5-7B in the GQA short-answer setting. All methods use the same checkpoint, evaluation samples, prompts, decoding configuration, and a test-instance batch size of 1. The frozen VLM, ENT, TTAug, TTAdapt, and TTT-Head use a single NVIDIA Tesla V100 32 GB GPU, whereas TTW uses two V100 32 GB GPUs with sharding and offloading for its full-parameter test-time warmup. Reported wall-clock time includes the complete online processing required by each configuration.

For TTT-Head, latency includes candidate construction, reversible Head–Action probing, sparse test-time training for qualified instances, and final prediction, averaged over all evaluated instances. The one-time task-level head discovery and calibration cost is excluded.

We additionally report *TTAug-8* and *TTAug-16* as augmentation-only inference references. TTAdapt reports its complete test-time procedure, including pseudo-label construction and parameter updates, while TTW includes auxiliary response generation and filtering, test-time warmup, and final VQA inference. Results are reported in Table 17.

Table 17: **Representative online runtime on LLaVA-1.5-7B.** Average wall-clock time per GQA sample with batch size 1. All methods except TTW use one V100 32 GB GPU; TTW uses two with sharding and offloading. Relative latency is normalized by frozen inference; TTT-Head's one-time task-level discovery and calibration cost is excluded.

Method	Avg. Time / Sample ↓	Relative Latency
Frozen VLM	0.52 s	1.0×
Entropy Minimization (ENT)	2.34 s	4.5×
TTAug-8	5.12 s	9.8×
TTT-Head (Ours)	5.68 s	10.9×
TTAug-16	9.46 s	18.2×
TTAdapt	37.80 s	72.7×
TTW	76.84 s	147.8×

TTT-Head requires 5.68 s per sample, close to the TTAug-8 augmentation-only reference and lower than TTAug-16, full TTAdapt, and TTW. Its measured wall-clock latency is approximately 6.7× lower than TTAdapt and 13.5× lower than TTW. Since TTW uses two GPUs, its result should be interpreted as end-to-end latency under its feasible execution configuration rather than a same-hardware throughput comparison.

Task-level discovery and calibration are performed once per model–benchmark configuration and reused across test instances. For N instances, the amortized per-instance runtime is

$$\bar{T}_{\text{total}} = \bar{T}_{\text{online}} + \frac{T_{\text{cal}}}{N}, \tag{17}$$

where T_{cal} is the one-time calibration cost. Table 17 reports only $\bar{T}_{\text{online}}$.

I Computational Resources

All experiments were conducted on a compute server equipped with four NVIDIA Tesla V100 GPUs with 32 GB memory each, an Intel(R) Xeon(R) Gold 6230R CPU running at 2.10 GHz, and 128 GB of system memory. Experiments with the 13B backbones were executed using two NVIDIA Tesla V100 32 GB GPUs. TTT-Head was implemented in PyTorch with Hugging Face Transformers, including task-level head discovery, reversible head-level intervention probing, candidate-relative response estimation, and sparse instance-level test-time training. During test-time training, only the lightweight scalar gates associated with the selected attention heads are optimized, while the remaining model parameters are kept frozen.