Air-Know: Arbiter-Calibrated Knowledge-Internalizing Robust Network for Composed Image Retrieval

Supplementary Material

This is the appendix of "Air-Know: Arbiter-Calibrated Knowledge-Internalizing Robust Network for Composed Image Retrieval".

A. Proof

In the main text, we assert that optimizing the EKI module via the Evidence Lower Bound (ELBO) is mathematically equivalent to minimizing the KL divergence between the variational distribution $q_\theta(\mathbf{W})$ and the intractable true posterior $p(\mathbf{W}|\mathcal{D}_{anchor})$. While this is a standard result in variational inference, we provide the complete, step-by-step derivation here to ensure the self-containment of our theoretical framework.

Theorem 1 (ELBO-KL Equivalence)**.** *The log-evidence (marginal likelihood) $\log p(\mathcal{D}_{anchor})$ can be rigorously decomposed into the sum of the ELBO and the KL divergence between the variational approximation and the true posterior. Consequently, since the evidence is constant with respect to the model parameters θ, maximizing the ELBO is strictly equivalent to minimizing the posterior KL divergence.*

Proof. Our starting point is the definition of the Kullback-Leibler (KL) divergence from the variational distribution $q_\theta(\mathbf{W})$ to the true posterior $p(\mathbf{W}|\mathcal{D}_{anchor})$.

$$\begin{aligned}&\mathcal{D}_{\mathrm{KL}}[q_\theta(\mathbf{W}) \parallel p(\mathbf{W}|\mathcal{D}_{anchor})] \\ &\quad= \mathbb{E}_{\mathbf{W}\sim q_\theta}\left[\log \frac{q_\theta(\mathbf{W})}{p(\mathbf{W}|\mathcal{D}_{anchor})}\right].\end{aligned} \tag{1}$$

Using the property of logarithms $\log(a/b) = \log a - \log b$, we expand the term inside the expectation into two separate components: entropy and cross-entropy-like terms:

$$\begin{aligned}\mathcal{D}_{\mathrm{KL}}[q_\theta \parallel p(\cdot|\mathcal{D})] &= \mathbb{E}_{q_\theta}[\log q_\theta(\mathbf{W})] \\ &\quad - \mathbb{E}_{q_\theta}[\log p(\mathbf{W}|\mathcal{D}_{anchor})].\end{aligned} \tag{2}$$

Next, we invoke Bayes' theorem for the posterior term:

$$p(\mathbf{W}|\mathcal{D}_{anchor}) = \frac{p(\mathcal{D}_{anchor}|\mathbf{W})p(\mathbf{W})}{p(\mathcal{D}_{anchor})}. \tag{3}$$

Substituting this expansion into the second term of Eq. (32), and utilizing the linearity of expectation, we obtain:

$$\begin{aligned}&\mathbb{E}_{q_\theta}[\log p(\mathbf{W}|\mathcal{D}_{anchor})] \\ &\quad= \mathbb{E}_{q_\theta}\left[\log \frac{p(\mathcal{D}_{anchor}|\mathbf{W})p(\mathbf{W})}{p(\mathcal{D}_{anchor})}\right] \\ &\quad= \mathbb{E}_{q_\theta}[\log p(\mathcal{D}_{anchor}|\mathbf{W})] + \mathbb{E}_{q_\theta}[\log p(\mathbf{W})] \\ &\qquad - \mathbb{E}_{q_\theta}[\log p(\mathcal{D}_{anchor})].\end{aligned} \tag{4}$$

Crucially, observe that the term $\log p(\mathcal{D}_{anchor})$ (the log-evidence) depends solely on the dataset and is independent of the weight variable $\mathbf{W}$. Therefore, it acts as a constant under the expectation $\mathbb{E}_{q_\theta}$:

$$\mathbb{E}_{q_\theta}[\log p(\mathcal{D}_{anchor})] = \log p(\mathcal{D}_{anchor}). \tag{5}$$

Now, we substitute the expanded form back into the original KL divergence equation:

$$\begin{aligned}&\mathcal{D}_{\mathrm{KL}}[q_\theta \parallel p(\cdot|\mathcal{D})] \\ &\quad= \mathbb{E}_{q_\theta}[\log q_\theta(\mathbf{W})] - \Big(\mathbb{E}_{q_\theta}[\log p(\mathcal{D}_{anchor}|\mathbf{W})] \\ &\qquad + \mathbb{E}_{q_\theta}[\log p(\mathbf{W})] - \log p(\mathcal{D}_{anchor})\Big).\end{aligned} \tag{6}$$

Rearranging the terms to isolate the log-evidence on one side, we reveal the fundamental decomposition:

$$\begin{aligned}\log p(\mathcal{D}_{anchor}) = \mathcal{D}_{\mathrm{KL}}[q_\theta \parallel p(\cdot|\mathcal{D})] \\ + \underbrace{\mathbb{E}_{q_\theta}[\log p(\mathcal{D}_{anchor}|\mathbf{W})]}_{\text{Reconstruction Term}} \\ - \underbrace{(\mathbb{E}_{q_\theta}[\log q_\theta(\mathbf{W})] - \mathbb{E}_{q_\theta}[\log p(\mathbf{W})])}_{\text{Regularization Term}}.\end{aligned} \tag{7}$$

We recognize that the last grouped term corresponds exactly to the KL divergence between the variational distribution and the prior $p(\mathbf{W})$:

$$\mathbb{E}_{q_\theta}\left[\log \frac{q_\theta(\mathbf{W})}{p(\mathbf{W})}\right] = \mathcal{D}_{\mathrm{KL}}[q_\theta(\mathbf{W}) \parallel p(\mathbf{W})]. \tag{8}$$

Thus, we arrive at the final identity:

$$\log p(\mathcal{D}_{anchor}) = \mathcal{D}_{\text{KL}}[q_\theta \parallel p(\cdot|\mathcal{D})] + \mathcal{L}_{\text{ELBO}}(\theta), \quad (9)$$

where the Evidence Lower Bound (ELBO) is defined as:

$$\begin{aligned}\mathcal{L}_{\text{ELBO}}(\theta) = \mathbb{E}_{q_\theta}[\log p(\mathcal{D}_{anchor}|\mathbf{W})] \\ - \mathcal{D}_{\text{KL}}[q_\theta(\mathbf{W}) \parallel p(\mathbf{W})].\end{aligned} \quad (10)$$

Since $\mathcal{D}_{\text{KL}} \geq 0$ (Gibbs' inequality), $\mathcal{L}_{\text{ELBO}}$ is indeed a lower bound on the log-evidence. More importantly, because $\log p(\mathcal{D}_{anchor})$ is fixed with respect to θ, maximizing $\mathcal{L}_{\text{ELBO}}(\theta)$ is strictly equivalent to minimizing the discrepancy $\mathcal{D}_{\text{KL}}[q_\theta \parallel p(\cdot|\mathcal{D})]$. □

B. Datasets

In this section, we provide a detailed introduction to two benchmark datasets widely recognized in the field of Composed Image Retrieval and adopted in our experiments, specifically FashionIQ and CIRR. The details are as follows:

- FashionIQ [46] serves as a standard dataset for evaluating Composed Image Retrieval within the fashion domain. It comprises 77,684 fashion images crawled from the web, totaling 30,134 annotated triplets. The content is primarily divided into three core categories: dresses, shirts, and toptees. This dataset is utilized mainly to assess retrieval performance in fashion scenarios, with a particular emphasis on evaluating the ability of the model to align visual content with descriptive modification texts.
- CIRR [1] utilizes real-world images derived from NLVR2 [**?**], which is a natural language visual reasoning dataset. CIRR comprises 36,554 annotated triplets and 21,552 images. Unlike the domain-specific nature of FashionIQ, CIRR places greater emphasis on complex interactions among multiple objects in natural environments. This characteristic effectively mitigates the risk of model overfitting to a single domain. Furthermore, it addresses the issue of substantial false negatives caused by incomplete annotations, a problem observed in FashionIQ, by including a dedicated subset for fine-grained contrastive evaluation. Consequently, CIRR represents an ideal choice for evaluating the capability of a model to handle complex scenarios, comprehend object interactions, and fuse multimodal information.

C. Cross-Validation of EPA

The External Prior Arbitration (EPA) module constitutes the cornerstone of the Air-Know framework. As articulated in the main text, our primary challenge lies in breaking the "self-dependent vicious cycle" between the Learner and the Arbiter. To achieve this decoupling, we introduce the EPA module, whose sole objective is to leverage Multimodal Large Language Models (MLLMs), such as GPT-4o, as an *offline expert arbiter*. Its purpose is to provide high-precision, reliable binary labels (i.e., "Clean" vs. "Noisy") for a subset of the training data. The final output of this process is a small-scale, high-precision Anchor Dataset, D_{anchor}, which is subsequently employed in the second phase (EKI) to supervise the lightweight proxy arbiter.

The "cross-validation strategy" mentioned in Sec. 3.2 relies on a multi-step, logically sophisticated prompt design. While the detailed prompts provided to the MLLM are available in Appendix H, we elaborate on the three core analysis stages executed:

Step1: Deconstruct Inputs. The objective of this phase is to compel the MLLM to analyze each component of the triplet independently and unbiasedly, establishing a factual foundation for subsequent comparisons and reasoning:

- **Input:**A multimodal triplet (I_r, T_m, I_t) randomly sampled from the training pool.
- **Process:** 1) *Analyze Reference Image (I_r).* The MLLM is instructed to first analyze the reference image in isolation, identifying key objects, salient attributes, background scenes, and their relationships. 2) *Analyze Target Image (I_t).* Subsequently, the MLLM analyzes the target image independently in the same manner. 3) *Analyze Modification Text (T_m).* Finally, the MLLM parses the modification text to accurately comprehend the intended modification actions (e.g., add, remove, replace, change color).
- **Output:** The output of this stage is a set of structured 'internal factual descriptions' $(Desc_r, Desc_m, Desc_t)$. These descriptions represent the MLLM's internal working state and are fed as a holistic input into the second phase.

By enforcing sequential processing, this phase effectively prevents "jumping to conclusions" or "confirmation bias" It ensures that subsequent reasoning is grounded in independent, objective observations of each component.

Step2: Compare & Reason. This phase constitutes the core of the EPA strategy, where the "cross-validation" occurs. The MLLM executes complex logical reasoning to judge the internal consistency of the triplet.

- **Input:** The structured descriptions from Step 1.
- **Process:** 1) *Infer Actual Differences (ΔT_I):* The MLLM's first reasoning step is to temporarily ignore T_m. It is instructed to strictly follow the Prompt requirements to compare only $Desc_r$ and $Desc_t$, aiming to objectively describe the actual visual changes that occurred from the reference to the target image. This inferred change, ΔT_I, represents the "factual change" observed by the MLLM, corresponding to the inferred instruction ΔT_I mentioned in the main text. 2) *Verify "Instruction" vs. "Fact":* The MLLM performs the critical cross-validation by compar-

ing the semantic consistency between T_m (the given instruction) and ΔT_I (the observed fact). 3) *Apply "Key Principles" (Handling Ambiguity):* In NTC scenarios, substantial noise is not entirely irrelevant but manifests as "partially match". Our Prompt enforces the principle that human annotation allows for incomplete information. For "Clean" samples: As long as the core change described in the modification text actually occurred between the reference and target images, the MLLM must classify it as "Clean", even if "unmentioned minor discrepancies" (e.g., slight pose changes, background shifts) exist. This ensures the expert correctly handles clean samples. 4) *NTC Cause Diagnosis:* If a severe logical disconnection or contradiction exists between T_m and ΔT_I, the MLLM classifies it as a noisy triplet correspondence. However, the core difficulty lies in partially matched samples, which are neither entirely noisy nor entirely correct. To ensure the integrity and correctness of the expert logic chain, we require the MLLM to further diagnose the root cause of the NTC classification and categorize it according to the NTC definitions:
  - Mismatched Modification Text: The text T_m describes an operation completely unrelated to ΔT_I. For example, the actual visual change ΔT_I is "car turns blue", but the text T_m is "add a chair".
  - Mismatch Reference Imgae: The modification text T_m and target image I_t are logically self-consistent, but the reference image is completely unrelated. For instance, T_m and I_t correspond to "church spire turns gold", but the reference image I_r is an unrelated "forest".
  - Mismatched Target Image: The reference image I_r and text T_m jointly point to a clear expectation, but I_t does not match.
- **Output:** The output of this stage is an exhaustive "Reasoning Chain", which fully documents the MLLM's logic from factual inference to cross-validation, principle application, and final diagnosis. This reasoning chain serves as the input for the final phase.

Step3: Judge & Conclude. The goal of this phase is to compile the MLLM's complex internal reasoning process into a standardized final format for dataset construction.

- **Input:** The Reasoning Chain output from Step 2.
- **Process:** The MLLM receives the complete reasoning chain and populates its analysis (including descriptions from Step 1 and reasoning from Step 2) into a strict structure defined by our required output format in the prompt.
- **Output:** The final output of the EPA Chain-of-Thought for a single triplet contains two key components: 1) *Analysis Workflow*. Contains the MLLM's complete thought process, including the structured descriptions $Desc_r, Desc_m, Desc_t$ from Step 1 and the Reasoning Chain from Step 2. This ensures the interpretability and traceability of the expert knowledge. 2) *Final Judgement*. Contains the MLLM's final verdict, consisting of a binary decision label accompanied by a concise summary of the core rationale.

Ultimately, we iterate this chain-of-thought process on the sampled small-scale subset to collect labels for individual samples, thereby constructing the anchor dataset D_{anchor}. This successfully consolidates the judgment knowledge of the MLLM expert, preparing for the training of the EKI module and achieving the thorough decoupling of the "Arbiter" and the "Learner".

D. Training Details

In this section, we provide a comprehensive description of the network architecture, the optimization procedure, and the proposed Two-Stage Progressive Training Strategy to ensure the reproducibility of Air-Know.

D.1. Architecture of the Lightweight Proxy

The Expert-Knowledge Internalization (EKI) module utilizes a lightweight Multi-Layer Perceptron (MLP) to internalize the expert priors. Given the backbone feature dimension $D = 256$, the input Geometric Deconstruction Vector (GDV) possesses a dimension of $4D = 1024$. The EKI proxy is implemented as a three-layer MLP ($1024 \to 512 \to 256 \to 1$). We apply ReLU activations and Dropout ($p = 0.1$) after the first two linear layers, while the final layer employs a Sigmoid function to output the confidence score $\hat{c}$. This lightweight design ensures the proxy captures the non-linear geometric boundaries without introducing significant computational overhead.

D.2. Two-Stage Progressive Training Strategy

To effectively decouple the arbiter from the learner and strictly prevent the "self-dependent vicious cycle", we adopt a progressive training strategy consisting of two distinct stages:

Stage 1: Expert Internalization (Warm-up). In this initial phase, our primary goal is to establish a reliable arbiter. We freeze the entire backbone network and exclusively optimize the parameters of the EKI module. Specifically, we construct the anchor dataset $\mathcal{D}_{anchor}$ by applying the EPA module to a randomly sampled subset comprising 40 batches, with a batch size of 256. Subsequently, utilizing this small-scale $\mathcal{D}_{anchor}$, the EKI module is trained to minimize the internalizing loss $\mathcal{L}_{EKI}$ (Eq. (9) in the main text) for 2 epochs. This stage enables the proxy to learn a robust probabilistic decision boundary for identifying noisy correspondence before interacting with the main model.

Stage 2: Dual-Stream Reconciliation. In the second phase, we train the representation model on the full dataset. Crucially, we **freeze** the parameters of the EKI module obtained from Stage 1. This ensures the gating signals remain stable and independent of the learner's current state. The

confidence scores $\hat{c}$ generated by the frozen EKI dynamically guide the training samples into either the clean alignment stream or the feedback reconciliation stream. The backbone is then optimized using the total objective (Eq. (14) in the main text).

E. Additional Quantitative Analysis

E.1. Efficiency Evaluation

To evaluate system efficiency and resource consumption, we compared the ordinary method SPRC, the robust method TME, and our Air-Know under identical hardware conditions with a batch size of 128. The experimental results are presented in Table 4, and the specific analysis is as follows:

1) In terms of computational cost (FLOPs), Air-Know (402.51G) exhibits the lowest computational overhead, representing a reduction of approximately 0.66% compared to TME (405.20G) and 2.63% relative to SPRC (413.38G). This result indicates that Air-Know effectively reduces the computational load while maintaining performance. Furthermore, the parameter counts for all models remain at approximately 915M, further demonstrating that the performance improvements of Air-Know do not stem from an increase in model scale.

2) Regarding GPU memory consumption, the requirement of Air-Know (16590 MiB) is higher than that of TME (12405 MiB). This increase in memory usage is within expectations and is primarily attributed to the training mechanism of Air-Know. During the training phase, the model is required to run the backbone network in parallel with the lightweight Expert-Knowledge Internalization proxy (utilized for generating matching confidence $\hat{c}$ in real time) while simultaneously maintaining gradient computations for both the clean alignment stream and the feedback reconciliation stream. We posit that exchanging moderate memory usage for enhanced training robustness and superior final feature quality represents a reasonable trade-off under current hardware conditions.

3) In terms of training efficiency, we adopt a rigorous evaluation metric that incorporates both the one-off offline MLLM labeling process and the EKI warmup phase. Under this comprehensive setting, Air-Know records a training time of 2.805 s/iter. While this presents a slight increase over SPRC (2.624 s/iter) due to the inclusion of the aforementioned components, Air-Know still demonstrates a substantial advantage over the robust baseline TME (7.858 s/iter), achieving a $\sim$3$\times$ speedup (approx. 64.9% reduction in training time). Regarding inference efficiency, the auxiliary EKI module is explicitly discarded during the test phase, allowing the system to rely solely on the optimized backbone. Consequently, Air-Know achieves a test time of 0.010 s/sample, surpassing TME (0.124 s/sample) by a margin of 12.4$\times$, ensuring high efficiency for real-world deployment.

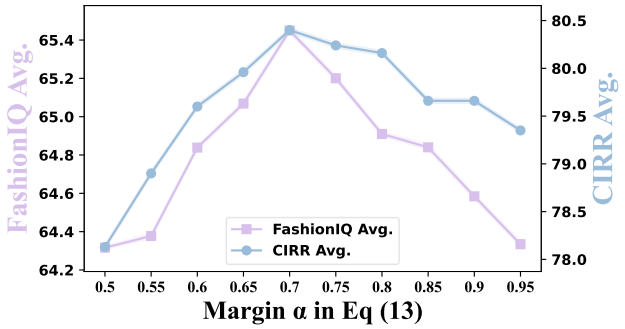

Figure 5. Sensitivity analysis of the margin α. We evaluated the impact of it in Equation (13), which serves as a parameter to control the threshold for penalizing noisy correspondence in the feedback reconciliation stream. A lower α imposes stricter filtering on semantically similar samples, while a higher α allows more samples exhibiting uncertainty to pass through, thereby creating distinct trade-offs between noise handling and sample retention.

E.2. Additional Hyperparameter Analysis

We investigated the impact of the tolerance margin α within the feedback reconciliation stream (Equation 13 in the main text). This hyperparameter acts as a threshold to penalize high-similarity noisy correspondence, specifically samples exhibiting low matching confidence $\hat{c}$ but high similarity s. As illustrated in Figure 5, we present its sensitivity curves on the FashionIQ and CIRR datasets.

It can be observed from the figure that as α increases from 0.5 to 0.95, the performance curves on both datasets exhibit a distinct inverted V-shaped trend, reaching their peak performance simultaneously at $\alpha = 0.7$. We explain this phenomenon by analyzing the gradient suppression mechanism within the Noisy Triplet Correspondence (NTC) scenario as follows:

Over-correction caused by excessively small α: When $\alpha < 0.7$, the tolerance threshold is set excessively low. Consequently, even weakly correlated noisy correspondence sharing only partial basic semantics, such as similar backgrounds or objects of the same category, triggers the penalty mechanism of $\mathcal{L}_{Recon}$. This overly aggressive gradient suppression strategy not only pushes away noise but also disrupts reasonable semantic continuity within the feature space. This hinders the model from learning intra-class commonalities, thereby leading to performance degradation, for instance, FashionIQ drops to approximately 64.3 when $\alpha = 0.5$.

Ineffective Constraints caused by excessively large α: When $\alpha > 0.7$, the tolerance threshold is set excessively high. In this case, only extreme hard noise that is highly similar to the query features triggers the loss function, while the vast majority of interfering samples with moderate sim-

Table 4. We conducted a comprehensive evaluation of computational complexity and operational efficiency, encompassing FLOPs, parameter counts, GPU memory overhead, and physical runtime. Under a unified batch size setting (Batch Size = 128), compared to the strong robust baseline TME, Air-Know achieved approximately a three-fold training acceleration with the lowest FLOPs while maintaining optimal retrieval accuracy, thereby demonstrating its optimal balance between efficiency and robustness.

Type	Method	FLOPs(G)	Parameters(M)	GPU Memory(MiB)	Test time(s/sample)	Train Time(s/iteration)	FashionIQ-Avg	CIRR-Avg
Ordinary	SPRC	413.38	915.69	24478(bs=128)	0.011	**2.624(bs=128)**	56.33	76.98
Robust	TME	405.20	**915.68**	**12405**(bs=128)	0.124	7.858(bs=128)	63.97	79.74
	Air-Know(Ours)	**402.51**	915.99	16590(bs=128)	**0.010**	2.805(bs=128)	**65.45**	**80.40**

ilarity are ignored as they satisfy the condition $s/\tau < \alpha$ (i.e., the loss becomes 0). This results in sparse or vanishing gradients within the feedback reconciliation stream, rendering it unable to effectively execute the de-pollution task and causing gradual degradation in model performance.

Consequently, $\alpha = 0.7$ represents an optimal balance point. It allows noisy correspondence to retain reasonable basic visual similarity while effectively penalizing erroneous high-similarity matches, thereby enhancing the robustness of Air-Know against noise.

F. Additional Ablation Study

F.1. Ablation Study of MLLMs

In the Air-Know framework, the External Prior Arbitration (EPA) module plays a pivotal role. Its core task is to utilize the MLLM as an offline expert to construct a high-precision Anchor Dataset (D_{anchor}) and through this dataset guide the subsequent lightweight agent (the EKI module) in learning noise discrimination logic. Although the EPA operates offline, constructing D_{anchor} still necessitates the processing of large volumes of data. The substantial inference overhead of MLLMs serves as the primary bottleneck. Consequently, it is necessary to find the optimal balance between expert-level discrimination quality and acceptable temporal and economic costs.

Based on the aforementioned requirements, as shown in Table 5, we compared mainstream open-source MLLMs (Llama-3.2-Vision-90B, Qwen3-VL-235B-Instruct) and closed-source MLLMs (GPT-4o, Claude-Sonnet-4.5, Gemini-2.5-Pro, GPT-5, etc.).We obtained the following observations: **1) Complex instruction-following capability:** When executing the deconstruction-reasoning-determination chain of EPA, GPT-5 and Claude-Sonnet-4.5 demonstrated extremely high accuracy, capable of precisely identifying highly deceptive noisy correspondences. **2) Reasoning efficiency and throughput:** Efficiency is a key consideration in our model selection. Although GPT-5 demonstrates superior performance, its batch processing time ($s/batch$) reaches 129 seconds, which is too slow and expensive for constructing D_{anchor}. In contrast, GPT-4o requires only 20 seconds for batch processing, making its efficiency more than six times that of GPT-5. **3) Robustness trade-off:** Under tests with different noise ratios (20%, 50%, 80%), GPT-4o exhibited excellent stability. Particularly under the extreme noise level of 80%, GPT-4o maintained an accuracy of 91.41%, which is only slightly lower than that of GPT-5 (94.43%) but higher than other lightweight models. This indicates that GPT-4o can provide sufficiently clean supervision signals for the EKI module to internalize while maintaining high efficiency.

F.2. Ablation Study of EPA

In the External Prior Arbitration (EPA) module, we employ a three stage cross validation strategy as follows:

Step 1: Deconstruct Inputs. The model independently analyzes the visual content of the reference image (I_r) and the target image (I_t) and separately comprehends the modification intent of the modification text (T_m), establishing objective factual descriptions.

Step 2: Compare & Reason. The model infers the actual visual change (ΔT_I) between the images and cross-verifies its semantic consistency with the textual instruction (T_m) to diagnose the presence of NTC noisy correspondence.

Step 3: Judge & Conclude. Based on the reasoning diagnosis described above, the model outputs a final binary determination (Clean or Noisy), thereby constructing a high-precision anchor dataset.

F.2.1. Quantitative Results

To verify the necessity of our three-stage cross-validation strategy, we conducted ablation experiments on the CIRR dataset across varying noise ratios ($\sigma \in \{0.2, 0.5, 0.8\}$). Our core hypothesis is that a reliable arbiter must simultaneously possess two capabilities: **1) Robustness against high-level semantic contradictions (NTC)** and **2) Tolerance for minor visual discrepancies (partial matches)**.

As shown in Table 6, we compared our complete prompt (Figure 8) against three variants: removing independent deconstruction (w/o Step 1, Figure 9), removing explicit reasoning (w/o Step 2, Figure 10), and an end-to-end baseline (w/o Step 1&2, Figure 11). These comparisons yielded the following observations:

1) Absence of objective anchors leads to text-induced confirmation bias: Removing Step 1 forces the MLLM to compare inputs without establishing independent objective factual descriptions. In high-noise environments ($\sigma = 0.8$), its accuracy exhibited a significant decline com-

Table 5. Performance and Efficiency Analysis of MLLMs in the EPA Module. We evaluated the discrimination accuracy and inference speed of various MLLMs across different noise levels. Under a unified experimental setting with a batch size of 256, we employed 256 concurrent threads to invoke the corresponding APIs, thereby achieving an efficient experimental workflow.

Noise	MLLM	$s/sample \downarrow$	FIQ-$Accuracy(\%) \uparrow$	CIRR-$Accuracy(\%) \uparrow$	$s/batch \downarrow$	FIQ-R@10-Avg	FIQ-R@50-Avg	CIRR-Avg
20%	gpt-5	20	89.85	**90.62**	129	**55.45**	**75.88**	**80.61**
	gpt-5-mini	**9**	79.14	80.47	41	54.96	75.55	80.22
	Llama-3.2-Vision-90B	17	75.32	76.95	80	54.78	75.42	80.10
	claude-sonnet-4.5	17	85.22	86.00	57	55.22	75.74	80.45
	Qwen3-VL-235B-Instruct	32	73.95	75.78	204	54.73	75.39	80.05
	gpt-5-nano	10	81.66	82.81	37	55.07	75.63	80.31
	gemini-2.5-pro	15	72.58	74.22	68	54.65	75.32	79.98
	gpt-4o	10	84.09	85.16	**20**	55.18	75.71	80.40
50%	gpt-5	20	92.51	**93.15**	129	**53.68**	**74.22**	**79.15**
	gpt-5-mini	**9**	82.68	84.03	41	53.20	73.88	78.75
	Llama-3.2-Vision-90B	17	78.45	80.12	80	52.98	73.72	78.56
	claude-sonnet-4.5	17	89.12	89.95	57	53.51	74.10	79.01
	Qwen3-VL-235B-Instruct	32	78.56	80.44	204	53.01	73.75	78.59
	gpt-5-nano	10	84.39	85.67	37	53.28	73.94	78.82
	gemini-2.5-pro	15	75.35	77.08	68	52.82	73.61	78.43
	gpt-4o	10	87.15	88.28	**20**	53.42	74.04	78.93
80%	gpt-5	20	93.88	**94.43**	129	**50.18**	**70.72**	**77.05**
	gpt-5-mini	**9**	85.52	86.88	41	49.75	70.41	76.68
	Llama-3.2-Vision-90B	17	81.50	83.25	80	49.54	70.26	76.50
	claude-sonnet-4.5	17	91.85	92.64	57	50.08	70.65	76.96
	Qwen3-VL-235B-Instruct	32	80.35	82.30	204	49.49	70.22	76.45
	gpt-5-nano	10	87.18	88.52	37	49.85	70.48	76.78
	gemini-2.5-pro	15	79.42	81.15	68	49.42	70.18	76.39
	gpt-4o	10	90.25	91.41	**20**	50.01	70.60	76.90

Table 6. Ablation results of EPA module on the CIRR dataset under different noise ratios ($\sigma = 0.2, 0.5, 0.8$). Specifically, w/o Step 1 removes the Deconstruct Inputs, compelling the model to execute comparisons directly without establishing objective factual anchor points. w/o Step 2 removes the Compare & Reason, omitting the crucial logical cross-verification stage and jumping directly from observation to determination. w/o Step 1 & 2 removes all structured intermediate steps, retaining only the naive end-to-end binary decision.

Variant	$\sigma = 0.2$	$\sigma = 0.5$	$\sigma = 0.8$
w/o Step1	83.38	86.84	89.25
w/o Step2	76.56	87.50	89.72
w/o Step1&2	75.78	86.72	87.28
Ours	85.16	88.28	91.41

pared to the complete model (91.41%), dropping to 89.25% ($\Delta = -2.16\%$). In NTC scenarios, particularly where the actual visual change is completely unrelated to the text instruction, misleading text (T_m) often causes the model to "hallucinate" non-existent visual changes. By generating independent factual descriptions ($Desc_r, Desc_m, Desc_t$), Step 1 effectively severs this interference, ensuring that the determination is based on objective facts.

2) Absence of reasoning chain causes rigid rejections of valid partial matches: Skipping the "Compare & Reason" stage (w/o Step 2) caused accuracy at $\sigma = 0.2$ to plummet to 76.56%, compared to Air-Know (85.16%). Under low-noise ($\sigma = 0.2$) setting, the dataset contains a large number of partial match samples. Lacking the reasoning process in Step 2 that infers the actual difference (ΔT_I) and performs cross validation with T_m, the model struggles to distinguish between "tolerable visual discrepancies" and "fundamental semantic contradictions", mistakenly discarding valid samples as noise.

3) Dual limitations of unstructured determination: The end-to-end variant (w/o Step 1&2) performed worst across all settings. Unstructured prompting fails the core dilemma under NTC scenarios: balancing skepticism against false text with tolerance for partial matches. Lacking Step 1, the model cannot resist text inducement in high-noise scenarios; it is extremely prone to the misdetermination of noise where "the text describes a change but the visual change does not occur" as Clean. Lacking Step 2, it cannot understand the rationality of "matching primary intent but flawed details" in low-noise scenarios; it tends to capture all minute visual discrepancies and determine them as Noisy.

F.2.2. Qualitative Results

To demonstrate the effectiveness of our prompt strategy, Figure 6 visualizes a highly deceptive partially mismatched sample. The reference image (I_r) shows a green cantaloupe, but the text (T_m) reads "Slice open the orange", and the target (I_t) depicts a sliced blood orange. Despite perfect text-target alignment, the operational premise contradicts I_r. Thus, it is a noisy triplet. We analyze the four variants' responses below:

1) Three stage cross validation prompt (First row, correct determination): The complete EPA model correctly

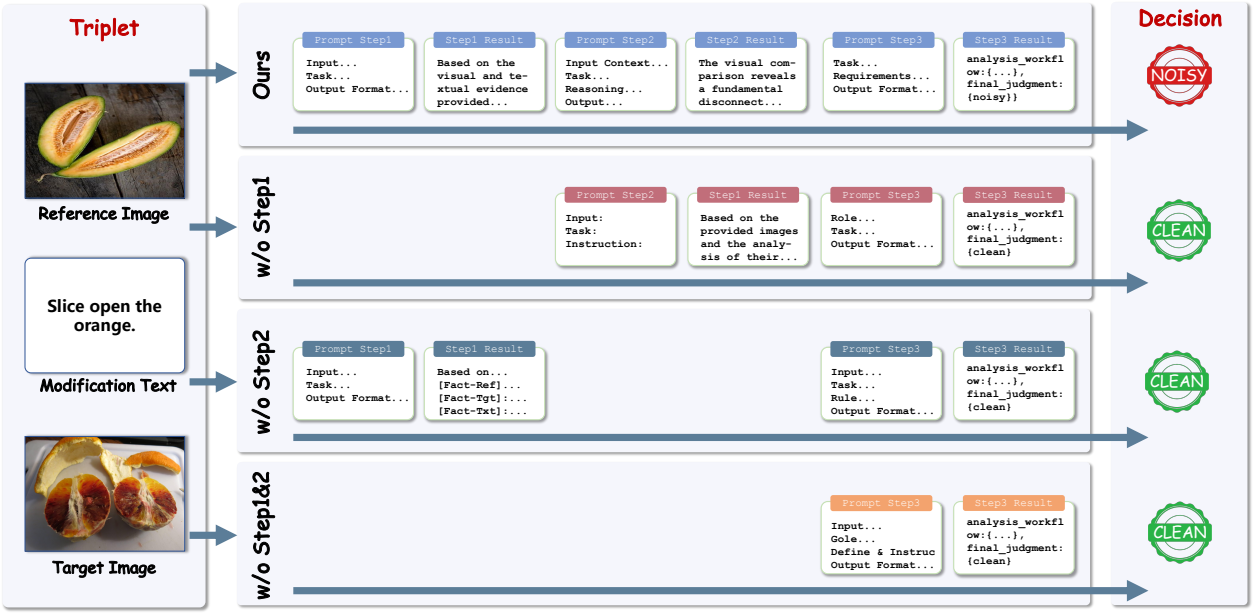

Figure 6. Visualization of prompt design and ablation study on a real-world NTC case. We present a comparison of the reasoning process between the full Air-Know (Ours) and three ablation variants on a typical "reference image mismatch" NTC sample. The left side displays the input triplet; the center illustrates the prompt execution flow (with specific steps retained or removed); and the right presents the final determination. In this case, only the complete model successfully identified the logical conflict between the reference image "a sliced cantaloupe" and the instruction "Slice open the orange". through the complete "Deconstruct-Reason-Determine" chain, while all ablation variants fell into confirmation bias and provided an incorrect "Clean" determination.

labels I_r as a melon in Step 1. In Step 2, it detects the conflict between the text's operational object ("orange") and the I_r anchor (melon), successfully diagnosing a Reference Image Mismatch and outputting Noisy.

2) w/o Step 1 (second row, misdetermination): Upon removing Step 1, the model loses its objective cognition of the reference image and succumbs to typical Confirmation Bias. Due to the absence of a pre-established factual anchor stating "this is a cantaloupe", the model focuses its attention entirely on the correspondence between the modification text (T_m) and the target image (I_t). It observes that the instruction "Slice open the orange" is perfectly executed in the target image, thereby generating a visual hallucination; consequently, it subjectively ignores the actual content of the reference image or erroneously assumes that the reference image is an unsliced orange.

3) w/o Step 2 (Third row, misdetermination): The failure of this variant reveals the importance of logical cross-validation in Step 2. Although the model might have identified the melon and the orange during the observation phase, the absence of an explicit reasoning mechanism for verifying the consistency between the instruction and the facts prevented the model from performing the critical logical check, specifically asking whether the instruction could be applied to the reference image. Faced with the strong semantic correlation between the text and the target image, the model, lacking logical scrutiny, tended to take shortcuts. It made determinations directly based on the matching degree between T_m and I_t, thereby ignoring the premise conflict between I_r and T_m.

4) w/o Step 1&2 (fourth row, misdetermination): As an end-to-end baseline, in the absence of structured guidance, the model is completely dominated by the evident semantic alignment between the text and the target. It is incapable of processing the complex dependencies within the triplet and fails to recognize the disconnection of the reference image within the logical chain, ultimately blindly rendering an erroneous determination of Clean. This result compellingly demonstrates that relying solely on perceptual capabilities is insufficient when confronting covert noise such as Partially Mismatch. By anchoring reference facts and conducting logical verification, the EPA strategy effectively prevents the model from succumbing to misdetermination caused by local semantic matching.

G. More Qualitative Results

As illustrated in Figure 7, we visualize the sample discrimination results of Air-Know in NTC scenarios and the cleanliness estimation output by the *Expert-Knowledge Internalization* (EKI) module. The visualization results intuitively confirm that, benefiting from the internalization of expert

discriminative logic by EKI, the model accurately distinguishes between logically consistent clean samples and NTC noise containing semantic conflicts. Consequently, it outputs highly credible confidence scores, exhibiting superior noise-robust discriminative capabilities.

Specifically, the blue region on the left illustrates the triplets identified as "Clean" by the model, which generally achieved exceptionally high confidence scores. Taking the second row on the left as an example, when presenting the subtle spatial positional and attribute transformations between the reference image and the target image (specifically, the modification text describing "Orange pouch is in front of blue pouch"), the model assigned a high score of 0.96. This demonstrates that Air-Know accurately captures fine-grained semantic consistency among multimodal inputs rather than merely relying on superficial visual similarity, thereby validating its profound understanding of effective semantic variations.

In contrast, the orange region on the right displays typical NTC noisy correspondence, to which the model assigns significantly low confidence. For instance, in the middle case of the first row, although the modification text describes a certain renovation change, a fundamental Cross-category Semantic Mismatch exists between the reference image (two pugs) and the target image (shelves). This severe logical disconnection is keenly identified by the model, resulting in an extremely low score of 0.17. Similarly, in the case on the right of the second row, the modification text instruction "Show shorn dog" is completely disconnected from the actual visual content presented (bottles and vases). The model assigns a low score of 0.19 to this, demonstrating its high sensitivity to the consistency between the modification text and the target image.

H. Prompt

We provide a detailed visualization of the specific prompt design logic utilized in the experiments in Figures 8, 9, 10, and 11.

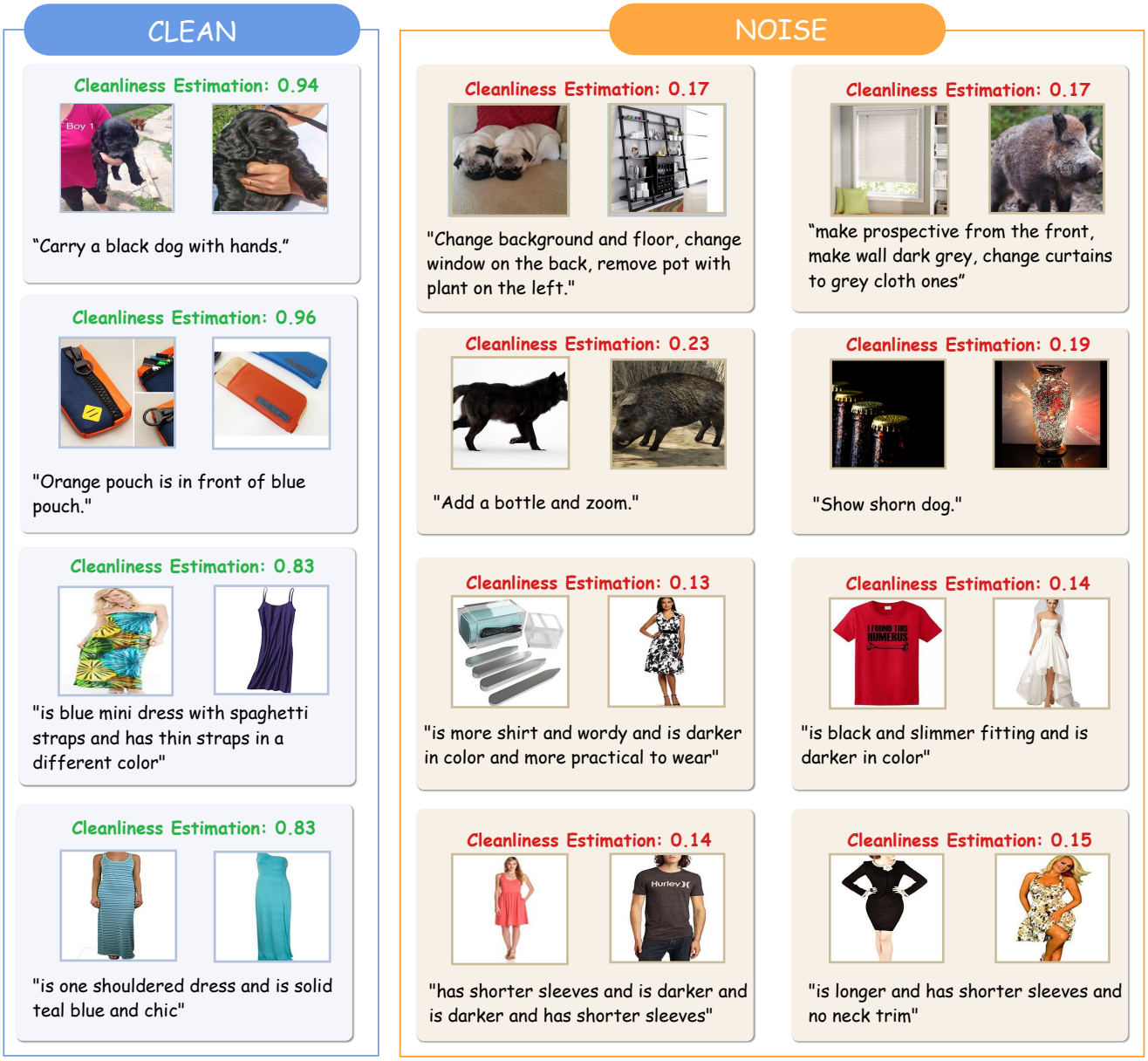

Figure 7. Visualization of NTC recognition results by the EKI module. We present the discrimination results of the EKI module for triplets. Left (Blue): Semantically consistent samples are mapped to high-confidence regions. Right (Orange): NTC exhibiting mismatch or text-image inconsistency is precisely suppressed, receiving extremely low scores. These reliable estimation values serve as dynamic gating signals for the DSR module.

Prompt Step1

Input:
- Reference Image <Image 1>
- Target Image <Image 2>
- Modification Text <Text>

Task:
You must analyze these three components **independently** and strictly. Do not compare them yet. Do not make assumptions. Just describe what you see.

1. **Deconstruct Reference Image:** Identify the main object, its attributes (color, shape, texture), the background, and the scene layout.
2. **Deconstruct Target Image:** Identify the main object, its attributes, background, and scene layout.
3. **Deconstruct Modification Text:** Analyze the text linguistically. What specific action is it requesting? (e.g., add object, change color, remove item, replace background).

Output Format:
Please provide your observations in the following structure:
- **[Fact-Ref]:** (Description of Reference Image)
- **[Fact-Tgt]:** (Description of Target Image)
- **[Fact-Txt]:** (Analysis of the Text Instruction)

Prompt Step2

Input Context:
You have previously extracted the objective facts: [Fact-Ref], [Fact-Tgt], and [Fact-Txt].

Task:
Perform a logical cross-validation to generate a "Reasoning Chain". You must apply the following **Strict Standards** derived from high-quality annotation guidelines.

1. The "Clean" Standard (Partial Match Principle):
Definition:** A triplet is Clean if the **core visual change** described in the text actually occurred.
Crucial Tolerance: Human annotations are imperfect. Ignore minor discrepancies (e.g., slight pose shifts, background noise, lighting changes). **Do not nitpick.**
Fashion Domain Note: In fashion, if the text says "change color", and the item changes color but retains the same style, it is Clean. If the item structure changes completely (e.g., from a T-shirt to a dress) without text instruction, it is Noisy.
2. The "Noisy" Standard (NTC Definitions):
A triplet is Noisy ONLY IF there is a fundamental logical break. Classify into one of these three:
Type A: Mismatched Modification Text: The actual visual change (ΔI) is completely unrelated to the text instruction (e.g., Image: "Car turns blue", Text: "Add a chair").
Type B: Mismatched Reference Image: The Ref image is irrelevant to the transformation logic (e.g., Text/Target refer to a "Church", but Ref is a "Forest").
Type C: Mismatched Target Image: The Ref + Text create a clear expectation, but the Target is completely different.
Reasoning Steps
1. **Identify Actual Δ:** Compare [Fact-Ref] and [Fact-Tgt] to find the **real** visual change.
2. **Validate Text:**** Does [Fact-Txt] match this real change?
3. **Apply Tolerance:**** Is it a "Partial Match" (Clean) or a "Fundamental Break" (Noisy)?

Output:
Provide your detailed **Reasoning Chain** text. Explain **why** it fits one of the definitions above.

Prompt Step3

Task:
Based on the detailed reasoning chain and analysis you just generated, summarize the findings and output the final verdict in the specified JSON format.

Requirements:
- **Verdict:** Must be strictly "Clean" or "Noisy".
- **Rationale:** A concise 1-2 sentence summary of the core reason.
- **Consistency:** Ensure the JSON content matches your previous reasoning perfectly.

Output Format:
```json
{
  analysis_workflow": {
    "step_1_reference_image_description": "A brief description of the reference image's content.",
    "step_2_target_image_description": "A brief description of the target image's content.",
    "step_3_modification_text_analysis": "An analysis of the instruction in the modification text.",
    "step_4_synthesis_and_reasoning": "The detailed synthesis and reasoning process, including identification of actual differences, text-to-difference validation, and the logic for applying the NTC definitions."
  },
  "final_judgment": {
    "verdict": "Clean | Noisy",
    "rationale": "A concise summary of the core reason for the verdict."
  }
}
```

Figure 8. The complete three stage cross-validation prompt architecture. This design enforces a Deconstruct-Reason-Determine process. Specifically, it guides the model to first deconstruct Inputs to establish objective visual factual anchors, subsequently to diagnose NTC types and verify semantic consistency, and finally to output a determination based on a comprehensive chain of evidence.

Prompt w/o Step1

Input:
- Reference Image <Image 1>
- Target Image <Image 2>
- Modification Text <Text>

Task:
Directly analyze the logical coherence of this triplet.

STRICT JUDGMENT CRITERIA (Must Follow):
1. Core Definitions:
Clean: The modification text accurately describes the **major visual change.**
Note: Partial Matches are ACCEPTABLE. Do not penalize minor unmentioned differences (background, angle, etc.).
Noisy (NTC):A logical contradiction exists.
Mismatched Ref: Ref image is unrelated to the task.
Mismatched Text: Text describes a wrong action.
Mismatched Target: Target image is wrong.

2. Common Pitfalls:
Avoid Over-Imagination: If you need complex creative thinking to connect the images, it is likely Noisy.
Focus on the Main Object: Ignore background clutter unless the text specifically mentions it.

Instruction:
Compare the images, read the text, and apply the criteria above. Generate a detailed **Reasoning Chain** explaining your decision process.

Role: You are a Data Formatting Specialist.

Task:
Based on the reasoning you just provided, verify the conclusion and output the final verdict in the specified JSON format.

Output Format:
```json
{
  "analysis_workflow": {
    "reasoning_summary": "Summarize your direct comparison logic here."
  },
  "final_judgment": {
    "verdict": "Clean | Noisy",
    "rationale": "Concise summary of the core reason."
  }
}
```

Figure 9. The prompt variant in which Step 1 (input deconstruction) is removed. In this setting, we eliminated the instruction requiring the model to independently parse the image and text content, thereby compelling the model to execute the comparison directly without establishing objective factual anchors.

Prompt w/o Step2

Input:
- Reference Image (Image 1)
- Target Image (Image 2)
- Modification Text

Task:
You must analyze these three components **independently** and strictly. Do not compare them yet. Do not make assumptions. Just describe what you see.

1. **Deconstruct Reference Image:** Identify the main object, its attributes (color, shape, texture), the background, and the scene layout.
2. **Deconstruct Target Image:** Identify the main object, its attributes, background, and scene layout.
3. **Deconstruct Modification Text:** Analyze the text linguistically. What specific action is it requesting? (e.g., add object, change color, remove item, replace background).

Output Format:
Please provide your observations in the following structure:
- **[Fact-Ref]:** (Description of Reference Image)
- **[Fact-Tgt]:** (Description of Target Image)
- **[Fact-Txt]:** (Analysis of the Text Instruction)

Input:
Using the facts extracted in the previous turn ([Fact-Ref], [Fact-Tgt], [Fact-Txt]).

Task:
Immediately classify the triplet based on these facts. **Do not output a reasoning chain.**

DECISION RULES (Internalize these before judging):

RULE 1: The Tolerance Rule (For "Clean"):
Is the **primary** action in [Fact-Txt] visible in the change from [Fact-Ref] to [Fact-Tgt]?
If YES, output "Clean" (even if there are other minor errors).
Example: Text: "Change to red". Ref: Blue shirt. Tgt: Red shirt (but folded differently). -> **Verdict: Clean.**

RULE 2: The NTC Rule (For "Noisy"):
Is there a total disconnect?
Example: Text: "Change to red". Ref: Car. Tgt: Dog. -> **Verdict: Noisy.**

Output Format:
```json
{
  "analysis_workflow": {
    "step_1_observations": "Summary of facts",
    "skipped_reasoning": "Reasoning skipped as per instruction."
  },
  "final_judgment": {
    "verdict": "Clean | Noisy",
    "rationale": "A single sentence summary based on Rule 1 or Rule 2."
   }
}
```

Figure 10. The prompt variant in which Step 2 (Comparison and Reasoning) is removed. While this design retains the preliminary observation of the inputs, it omits the core reasoning chain involving the inference of the actual visual change (ΔT_I) and the execution of cross validation. The model is required to proceed directly from observation to conclusion.

Prompt w/o Step1&2

Input: Reference Image, Target Image, Modification Text.

Goal: Classify this triplet as "Clean" or "Noisy" and output the result in JSON.

Definitions:
- **Clean:** The text describes the major visual change from Reference to Target. Partial matches (ignoring background noise) are acceptable.
- **Noisy (NTC):** 1. Mismatched Reference (Ref is irrelevant).
  2. Mismatched Text (Text describes a different action than what happened).
  3. Mismatched Target (Target is wrong).

Instructions:
Analyze the images and text. Deconstruct their content, compare the differences, check if the text matches the visual difference, and apply the NTC definitions.

Output Format:

```json
{
  "analysis_workflow": {
    "full_analysis": "Include your observation, comparison, and reasoning here."
  },
  "final_judgment": {
    "verdict": "Clean | Noisy",
    "rationale": "The core reason."
  }
}
```

Figure 11. The end-to-end prompt variant in which both Step 1 and Step 2 are removed. This variant strips away all structured intermediate steps, requiring the model to directly output a binary classification result (Clean/Noisy) based on the raw triplet input.