SAVER: Selective Auditing of Verbal Evidence for Error Recovery in VLM Change Reasoning

Youdi Li
Panasonic Connect Co., Ltd.
Tokyo, Japan
ri.yutei@jp.panasonic.com

Abstract

Vision-language models (VLMs) frequently fail at visual change reasoning, even when their vision encoders contain sufficient information. We observe that correct VLM outputs tend to contain explicit verbal evidence (object names, colors, spatial locations) that supports the claimed change, while incorrect outputs often lack such evidence. We propose SAVER (Selective Auditing of Verbal Evidence for Error Recovery), a lightweight, rule-based method that parses VLM responses for this evidence and triggers structured reprompting only when evidence is missing or inconsistent. Across three change detection benchmarks and four VLMs, SAVER significantly improves accuracy on tasks where errors stem from the model failing to articulate what it saw (expression failures), with gains up to +25.8% on CLEVR-Change. The evidence patterns can also be generated by an LLM in a single call, matching the hand-tuned gate on CLEVR-Change. Ablation experiments confirm that the evidence gate, not reprompting alone, drives the improvement.

1 Introduction

Vision-language models (VLMs) fail at change detection. Given two images of a scene, they miss obvious differences, hallucinate changes that do not exist, or describe changes in vague terms that lack specific detail (Liu et al., 2025a; Rahmanzadehgervi et al., 2024). Figure 1 shows a typical failure: GPT-4o identifies the wrong object ("green cube" instead of the green cylinder) and the wrong change type (material change instead of removal), despite the change being visually obvious.

Why do VLMs fail? Recent work points to a gap between perception and expression. Rahmanzadehgervi et al. (2024) showed that VLMs fail on simple visual tasks despite the visual information being present in the input. Orgad et al. (2025) showed that LLM internal representations encode truthfulness information not reflected in outputs. Balasubramanian et al. (2025) found that VLMs sometimes reason correctly toward the ground truth but then change their answer, a pattern they call *inconsistent reasoning*.

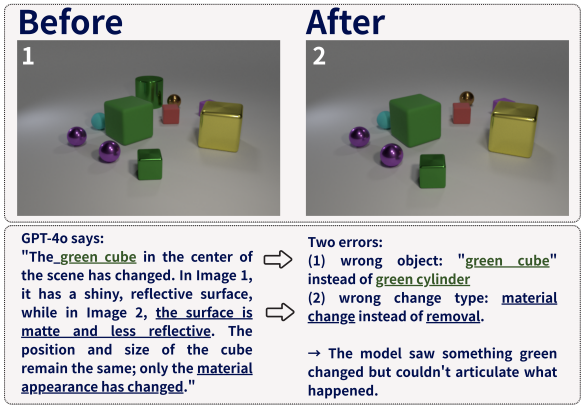


Figure 1: Expression failure in VLM change reasoning (one sample from CLEVR-Change). The ground truth is removal of the green cylinder, but GPT-4o misidentifies both the object and the change type. The model detects a change in the green region but cannot express it correctly.

We propose SAVER (**S**elective **A**uditing of **V**erbal **E**vidence for Error **R**ecovery), a lightweight method that uses this perception-expression gap at test time. SAVER sends a standard prompt and applies a rule-based *evidence gate* that parses the response for key verbal evidence. If the required evidence is present and consistent with the claimed change type, the gate accepts the response. If evidence is missing or conflicting, the gate triggers a structured reprompt that guides the model through systematic comparison.

The evidence check itself adds no cost; only the reprompt does. Because the gate is a short list of keywords and rules, every decision can be inspected to see which evidence was missing and why a reprompt was issued. Unlike learned verification methods where decisions cannot be inspected

Findings of the Association for Computational Linguistics: IJCNLP-AACL 2026, pages 3209–3227
November 6-10, 2026 ©2026 Association for Computational Linguistics

(Khan and Fu, 2024; Wu et al., 2025b), SAVER's gate logic is fully transparent.

SAVER is based on two observations about expression failures. First, correct and incorrect VLM outputs differ in a measurable way: correct answers contain explicit verbal evidence (object names, colors, spatial locations) that supports the claimed change, while incorrect answers often lack such evidence. This gap can be measured and varies by model and dataset (Section 5). Second, when a model produces an evidence-poor response, a structured reprompt that forces explicit object enumeration often recovers the correct answer. The model saw the relevant objects but did not express the right evidence until asked to enumerate systematically (Section 4.3).

We evaluate SAVER on three change detection benchmarks (CLEVR-Change (Park et al., 2019), MagicBrush (Zhang et al., 2023), and Spot-the-Diff (Jhamtani and Berg-Kirkpatrick, 2018)) across four VLMs (GPT-4o, Gemini 2.0 Flash, Qwen2.5-VL-7B, and LLaMA-4-Scout). SAVER improves GPT-4o accuracy by +13.4% on CLEVR-Change and +5.3% on MagicBrush over the baseline, with larger gains for weaker models (up to +25.8% for Qwen), while breaking 3–9.5× fewer correct answers than always-structured prompting. However, SAVER shows no significant effect on Spot-the-Diff.

This work makes three contributions. First, we show that VLMs sometimes see the change but omit key evidence from their text output. This expression failure is systematic and detectable from the output text alone, without access to model internals. Second, we propose SAVER, a rule-based method that uses verbal evidence as a diagnostic signal to selectively trigger structured reprompting at test time. Third, we evaluate SAVER across three datasets and four models, showing when selective auditing helps (expression-dominant errors), when it does not (perception-dominant errors), that it consistently fixes more samples than it degrades, and that its patterns can be LLM-generated.

2 Related Work

Image difference captioning. Introduced by Jhamtani and Berg-Kirkpatrick (2018) and established with CLEVR-Change (Park et al., 2019), image difference captioning has driven specialized architectures (Hu et al., 2024; Evennou et al., 2025; Li et al., 2025; Huang et al., 2025) and multi-image benchmarks showing that general-purpose VLMs lag well behind humans on cross-image comparison tasks (Zhang et al., 2025). These methods improve the model itself through training, while our work operates at inference time on frozen models, checking whether the output contains evidence that supports its claim.

Perception–expression gap in VLMs. Multiple lines of evidence show that VLM errors often originate not in perception but in generation. Vision encoders contain sufficient information that the language model fails to decode (Rahmanzadehgervi et al., 2024; Tong et al., 2024). Internal representations encode correct answers that do not reach the output (Orgad et al., 2025; Simhi et al., 2024; Balasubramanian et al., 2025). Reasoning traces can be unfaithful to the model's own knowledge (Turpin et al., 2023; Lanham et al., 2023). Benchmarks confirm the scale of the problem: GPT-4V scores 31.4% on HallusionBench question pairs (Guan et al., 2024), and GPT-4o achieves only 5.9 BLEU-4 on image difference captioning (Liu et al., 2025a). Liu et al. (2025b) showed that VLMs perceive visual evidence in intermediate representations but fail to use it during generation.

Selective verification. Always-verify strategies waste computation on inputs the model already handles (Snell et al., 2025). Recent methods implement selective triggers: dynamic early exit (Yang et al., 2025), adaptive reasoning escalation (Wu et al., 2025a), and RL-based reasoning selection (Wang et al., 2025). MM-Verify (Sun et al., 2025) applies output-level parsing to mathematical reasoning. Methods closer to our setting each require resources beyond the output text: REVERSE (Wu et al., 2025b) needs access to token-level logits during generation, VERA (Pei et al., 2026) needs attention maps, ReCoVERR (Srinivasan et al., 2024) deploys two additional models, and Khan and Fu (2024) require multiple API calls for paraphrase consistency. In contrast, our approach parses only the response already produced, adding zero cost unless the gate triggers a reprompt.

3 Method: SAVER

SAVER is a test-time method that checks whether a VLM response contains the verbal evidence its own claim requires. Given a task with known evidence categories, the user defines keyword pattern sets and gate rules. SAVER sends a baseline prompt, parses the response for the specified evidence, and

decides whether to accept or issue a structured reprompt. These evidence definitions can also be generated automatically by prompting an LLM with a task description and example outputs (evaluated in Section 4.5). Once generated, the rules are applied deterministically at inference time with no LLM involvement in the gate itself. Figure 2 illustrates the pipeline.

The underlying observation is that correct and incorrect VLM outputs differ systematically in verbal evidence. When a model answers correctly, it tends to state specific details that support its claim. When it errs, these details are often absent or inconsistent. We demonstrate SAVER on VLM change detection, where evidence categories (object names, colors, spatial terms) and gate rules are based on the structure of change types.

3.1 Evidence Extraction

Evidence extraction converts the VLM's free-text response into a binary evidence vector using case-insensitive pattern matching. Let $\mathcal{P} = \{\mathcal{P}_1, \ldots, \mathcal{P}_K\}$ denote the keyword pattern sets for K evidence categories. Given a response r, we extract a binary vector $\mathbf{e}(r) \in \{0,1\}^K$:

$$e_k(r) = \begin{cases} 1 & \text{if } \exists\, p \in \mathcal{P}_k \text{ s.t. } p \text{ matches } r \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Each category captures a type of visual evidence relevant to the dataset. For CLEVR-Change (K=4), the categories are: `absence` (e.g., *missing*, *removed*), `presence` (e.g., *new*, *added*), `color_change` (change verbs combined with color terms), and `material_change` (change verbs combined with material terms). MagicBrush uses five categories. It keeps `absence`, `presence`, and `color_change`, drops `material_change`, and adds `replacement` and `attribute_change`. Spot-the-Diff contains multiple simultaneous changes per image pair with no fixed type taxonomy, so the gate checks only whether the response mentions any change evidence at all rather than matching evidence to a specific change type. The complete keyword lists are in Appendix C.

In addition to evidence vectors, the gate extracts a predicted change type $\hat{t}(r) \in \mathcal{T} \cup \{\texttt{unclear}\}$, where $\mathcal{T}$ is the set of valid change types for the dataset (e.g., $\mathcal{T} = \{\texttt{drop}, \texttt{add}, \texttt{color}, \texttt{texture}\}$ for CLEVR-Change). The type is determined by the same keyword matching used for evidence extraction: if the response contains predefined removal phrases (e.g., "was removed," "no longer present"), the predicted type is `drop`; if it contains addition phrases (e.g., "was added," "appeared"), the predicted type is `add`; and so on. If no type-indicating phrase is found, or if the response matches multiple conflicting types, the predicted type defaults to `unclear`.

$\hat{t}$	$\text{req}(\hat{t})$	$\text{conf}(\hat{t})$
`drop`	`absence`	`presence`
`add`	`presence`	`absence`
`color`	`color_change`	`material_change`
`texture`	`material_change`	`color_change`

Table 1: CLEVR-Change gate mappings. Each predicted type requires specific evidence and conflicts with a different category.

3.2 Gate Decision

For each change type $t \in \mathcal{T}$, we define a required evidence index $\text{req}(t) \in \{1, \ldots, K\}$ and a conflicting evidence index $\text{conf}(t) \in \{1, \ldots, K\}$. Table 1 shows these mappings for CLEVR-Change.

The gate maps $(\hat{t}, \mathbf{e})$ to a binary decision:

$$G(r) = \begin{cases} \text{TRIGGER} & \text{if } \hat{t}(r) = \texttt{unclear} \\ \text{TRIGGER} & \text{if } e_{\text{req}(\hat{t})} = 0 \\ \text{TRIGGER} & \text{if } e_{\text{conf}(\hat{t})} = 1 \\ \text{ACCEPT} & \text{otherwise} \end{cases} \quad (2)$$

The gate triggers when (1) no change type can be identified, (2) the predicted type's required evidence is absent, or (3) conflicting evidence is detected. If none of these conditions holds, the baseline response is accepted. MagicBrush and Spot-the-Diff use analogous mappings adapted to their change types (Appendix A).

3.3 Selective Reprompting

When the gate triggers, SAVER sends a structured reprompt that guides the model through systematic comparison: first enumerate objects in Image 1, then enumerate objects in Image 2, then compare systematically, and finally state the answer. This design draws on chain-of-thought prompting (Wei et al., 2022; Kojima et al., 2022), which improves reasoning by eliciting intermediate steps, and extends it to multimodal settings where decomposing visual input into explicit object descriptions before comparison has been shown to improve compositional reasoning (Zhang et al., 2024; Mitra et al., 2024). Ablation C4 confirms that this structure is the active ingredient: a generic retry prompt

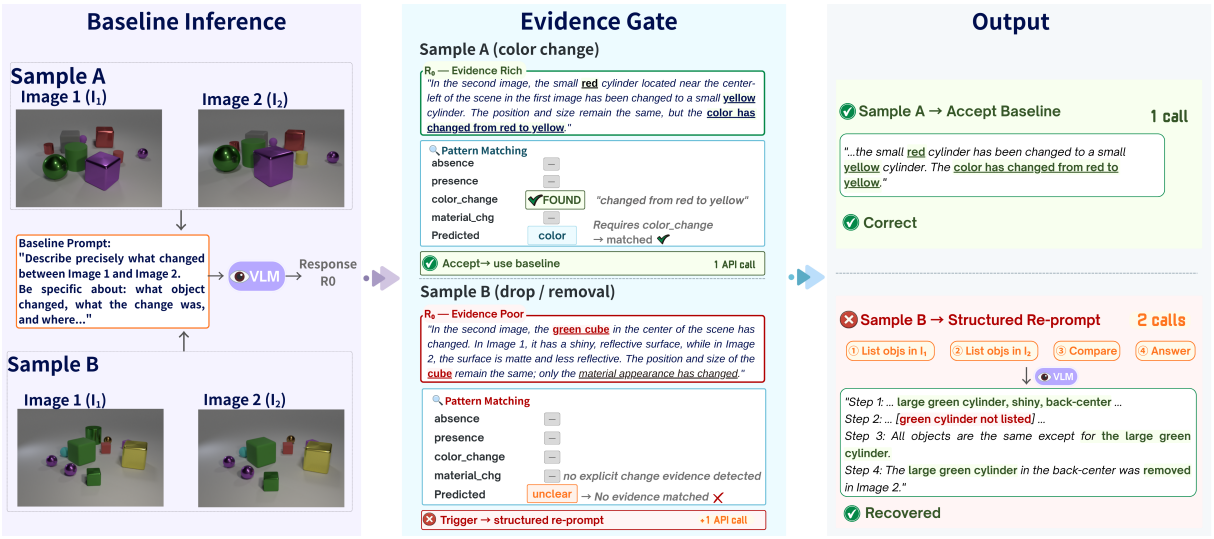


Figure 2: The SAVER pipeline. Given an image pair, SAVER sends a baseline prompt and parses the response for verbal evidence using rule-based pattern matching. **Sample A** (color change): the response states "changed from red to yellow," so the gate finds `color_change` evidence matching the predicted type and accepts the baseline (1 call). **Sample B** (drop): the baseline claims "the green cube. . . has changed" and mentions material terms ("matte," "reflective"), but no evidence vector registers a match; the response never makes an explicit change claim such as "changed from matte to shiny." The predicted type is `unclear` and no evidence is matched, so the gate triggers. The structured reprompt forces object enumeration, and the model correctly identifies the removed cylinder (2 calls).

with the same gate timing fails to replicate the improvement (Section 4.4). The full prompts are in Appendix B.

Given an image pair $x = (I_1, I_2)$, the final output is:

$$\text{SAVER}(x) = \begin{cases} r_0 & \text{if } G(r_0) = \text{ACCEPT} \\ r_1 & \text{if } G(r_0) = \text{TRIGGER} \end{cases} \quad (3)$$

where $r_0 = \text{VLM}(x, p_{\text{base}})$ is the baseline response and $r_1 = \text{VLM}(x, p_{\text{struct}})$ is the structured reprompt response. The expected API cost per sample is $1 + \tau$, where τ is the trigger rate. In most settings, the majority of samples require only the baseline call (Section 4.2). Algorithm 1 summarizes the full procedure.

3.4 Case Study

We trace SAVER through two CLEVR-Change samples to illustrate the gate logic.

Sample A (gate accepts). A color change: a small red cylinder turns yellow (upper path in Figure 2). GPT-4o responds: "*the small red cylinder has been changed to a small yellow cylinder. The color has changed from red to yellow.*" The gate scores the predicted type as `color` and scans for `color_change` evidence. The phrase "from red to yellow" is a specific color transition claim. Required evidence is present with no conflict, so the gate returns ACCEPT. Cost: 1 API call.

Sample B (gate triggers). A green cylinder is removed (lower path in Figure 2), but GPT-4o responds: "*the green cube in the center of the scene has changed. . . the surface is matte and less reflective. . . only the material appearance has changed.*" The response uses material-related vocabulary ("matte," "reflective") but never makes an explicit change claim that would register as evidence (e.g., "changed from matte to shiny"). No evidence vector is matched, so the predicted type is `unclear` and the gate returns TRIGGER. The structured reprompt forces GPT-4o to enumerate all objects in both images. It responds: "*The large green cylinder in the back-center was removed in Image 2.*" Cost: 2 API calls.

The general principle behind both cases is the same: the gate does not judge whether the answer is correct, but whether the response contains the type of evidence that a correct answer would require. A color change claim needs a color transition phrase; a removal claim needs absence language; a texture change claim needs material terms in a change context. For example, the `material_change` pattern requires a change verb (e.g., "changed," "switched") to appear near a material term (e.g., "matte," "shiny") within a short

Algorithm 1 SAVER Inference Protocol

Require: Image pair x, VLM $\mathcal{M}$, patterns $\mathcal{P}$, mappings $\text{req}(\cdot)$, $\text{conf}(\cdot)$
Ensure: Final response r^*

$r_0 \leftarrow \mathcal{M}(x,\ p_{\text{base}})$ ▷ Baseline call
$\mathbf{e} \leftarrow \text{Extract}(r_0,\ \mathcal{P})$ ▷ Eq. 1
$\hat{t} \leftarrow \text{TypeExtract}(r_0)$
if $\hat{t} = \texttt{unclear}$ **then**
  $G \leftarrow$ TRIGGER
else if $e_{\text{req}(\hat{t})} = 0$ **then**
  $G \leftarrow$ TRIGGER ▷ Missing evidence
else if $e_{\text{conf}(\hat{t})} = 1$ **then**
  $G \leftarrow$ TRIGGER ▷ Conflicting evidence
else
  $G \leftarrow$ ACCEPT
end if
if $G =$ TRIGGER **then**
  $r^* \leftarrow \mathcal{M}(x,\ p_{\text{struct}})$ ▷ Structured reprompt
else
  $r^* \leftarrow r_0$ ▷ Keep baseline
end if
return r^*

text window. In Sample B, the words "matte" and "reflective" appear, but never close to a change verb, so the pattern does not match and the evidence vector remains zero. This design means the gate catches responses that *sound* relevant but fail to commit to a specific, checkable claim. The reprompt then forces the model to enumerate objects and compare them, producing the structured evidence that the baseline lacked.

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate on three benchmark datasets spanning synthetic, edited, and natural images (Table 2): CLEVR-Change, MagicBrush, and Spot-the-Diff. CLEVR-Change and MagicBrush define fixed change-type taxonomies with four and five categories respectively. Our MagicBrush sample contains no replace instances (attribute 100, add 97, remove 94, color 49). In contrast, Spot-the-Diff contains multiple simultaneous changes without a predefined taxonomy and is evaluated on recall: how many of the true changes the model correctly mentions.

Models. We test four VLMs: GPT-4o (OpenAI, 2024), Gemini 2.0 Flash (Google DeepMind, 2024), Qwen2.5-VL-7B (Bai et al., 2025), and LLaMA-4-Scout (Meta AI, 2025). The latter three are accessed via OpenRouter,[1] an API aggregator for hosted models. All use temperature 0. To check whether SAVER generalizes beyond these four models, we also run Claude 3.5 Sonnet (Anthropic, 2024) on a 200-sample CLEVR-Change subset, where SAVER improves over both baselines with zero Broke cases (Appendix J).

Dataset	Domain	N
CLEVR-Change	Synthetic 3D	632
MagicBrush	Edited photos	340
Spot-the-Diff	Natural photos	197

Table 2: Dataset summary.

Conditions.
- **Baseline (B0):** direct prompt, one call.
- **Structured (B1):** enumerate-compare prompt, one call.
- **Standard CoT (B2):** "think step by step," one call (CLEVR only).
- **SAVER:** B0 first; if the gate triggers, B1 follows (one or two calls).

Ablations. Four controls isolate the gate's and prompt's contributions:
- **Random (C1):** reprompts the same count as SAVER but at random.
- **Length (C2):** reprompts short responses.
- **Inverted (C3):** reprompts the opposite set from SAVER.
- **Generic reprompt (C4):** same gate as SAVER, but replaces structured enumeration with "reconsider your previous answer."

C1–C3 use cached responses (zero additional API cost). C4 requires new API calls (multi-turn: baseline then generic reprompt). We also compare against **Self-Consistency (SC-5)** (Wang et al., 2023): sample the model 5 times at temperature 0.7, majority vote on detected change type, and return the longest response of the winning type.

Evaluation. Rule-based keyword matching: a response is correct if it names the correct change type and at least one ground-truth object keyword. We manually verified evaluation accuracy on 50 random CLEVR-Change samples and found 98% agreement with human judgment. Note that gate keywords and evaluation keywords are different sets: the gate checks evidence *categories* (e.g., color change verbs) while evaluation checks *specific ground-truth terms* (e.g., exact colors). See

[1] https://openrouter.ai

Model	B0	B1	B2	SAVER	Trig.
GPT-4o	72.8	80.7	77.5	**86.2**	39%
Gemini	81.5	**95.6**	87.3	93.0	36%
Qwen	56.5	**87.0**	73.4	82.3	49%
LLaMA	82.3	86.2	**90.5**	88.9	37%

Table 3: CLEVR-Change accuracy (%, $N = 632$). **Bold** = best; underline = SAVER. Trig. = gate trigger rate. All SAVER vs. B0: $p < 0.001$.

Model	B0	B1	SAVER	Trig.
GPT-4o	57.6	**70.3**	62.9	22%
Gemini	67.9	**79.7**	68.8	11%
Qwen	38.5	**68.8**	56.2	38%
LLaMA	58.5	**74.4**	63.5	30%

Table 4: MagicBrush accuracy (%, $N = 340$). **Bold** = best; underline = SAVER.

Model	B0	SAVER	Δ	Trig.
GPT-4o	77.2	78.2	+1.0	14%
Gemini	67.5	70.1	+2.5	17%
Qwen	61.4	62.9	+1.5	12%
LLaMA	73.1	78.2	+5.1	57%

Table 5: Spot-the-Diff results (% recall, $N = 197$). All $p > 0.05$ (Bonferroni). Fix/Broke counts: GPT-4o 2/0, Gemini 5/0, Qwen 3/0, LLaMA 14/4.

Limitations for further discussion. We report bootstrap 95% CIs and Bonferroni-corrected McNemar's tests.

4.2 Main Results

CLEVR-Change. Table 3 shows accuracy across all four models. SAVER improves over baseline (B0) for all four models (all $p < 0.001$). The effect is largest for Qwen, which has the lowest baseline accuracy and the highest trigger rate, because its baseline outputs frequently lack explicit evidence. For GPT-4o, SAVER outperforms B1 by 5.5%, because structured prompting hurts drop and add types while the gate preserves baseline answers for these types (Table 7). For Gemini and Qwen, B1 yields higher accuracy than SAVER, but also overwrites more correct baselines (Table 6); we detail this below. Standard CoT (B2) falls between B0 and B1 for three models, suggesting that "think step by step" provides some but not all of the benefit of explicit object enumeration. LLaMA is the exception: B2 (90.5%) exceeds B1 (86.2%). As the only mixture-of-experts model tested, LLaMA may respond better to open-ended reasoning than to rigid enumeration instructions.

MagicBrush. On real photographs with instruction-guided edits (Table 4), three of four models improve significantly over baseline ($p < 0.05$). Gemini shows a non-significant gain, likely because its baseline outputs are already evidence rich on this dataset (trigger rate 11%). B1 yields higher accuracy than SAVER for all four models, because most MagicBrush errors are perception failures where models misidentify the edited region rather than omit evidence. The gate's low trigger rates (11–38%) confirm this: models produce evidence-rich but factually wrong descriptions, so the gate accepts responses that are confident but incorrect. However, B1's accuracy advantage again comes with higher breakage (Table 6).

Statistical summary. All CLEVR-Change improvements are significant ($p < 0.001$, McNemar's test with Bonferroni correction). On MagicBrush, three of four models show significant gains ($p < 0.05$). Gemini's confidence interval crosses zero. Full bootstrap CIs are in Table 13 (Appendix D).

Spot-the-Diff. SAVER shows no significant improvement on this dataset for any model (all $p > 0.05$; Table 5). Trigger rates are low for GPT-4o, Gemini, and Qwen (12–17%), because baseline responses contain change-related keywords even when incorrect: models describe plausible but wrong changes in detail, producing evidence-rich outputs that the gate accepts. LLaMA is an outlier with 57% trigger rate but only a 12% fix rate among triggered samples, indicating that even when the gate detects sparse outputs, the underlying errors are perceptual rather than expressive. Appendix L shows this directly: 27–46% of gate-accepted responses are wrong despite containing evidence language. We analyze the boundary further in Section 5.

Robustness: Fix/Broke analysis. A reprompting strategy may degrade already-correct answers. We count *Fix* (baseline wrong, method correct) and *Broke* (baseline correct, method wrong). Table 6 compares SAVER and B1. SAVER fixes more than it breaks for every model on every dataset, with Fix:Broke ratios from 2.5:1 to 74:1. B1 breaks 3–9.5× more correct baselines than SAVER, because B1 applies enumeration to *all* samples including those where baseline is already correct. For GPT-4o on CLEVR-Change, B1 breaks 76 samples (12%) while SAVER breaks only 8 (1.3%). On

	Model	Fix	Broke	B1 Broke
CLEVR	GPT-4o	93	8	76
	Gemini	74	1	9
	Qwen	173	10	30
	LLaMA	59	17	67
MB	GPT-4o	21	3	21
	Gemini	5	2	14
	Qwen	64	4	15
	LLaMA	20	3	18

Table 6: Fix/Broke counts for SAVER and always-structured (B1) across CLEVR-Change and MagicBrush (MB). Fix and Broke are for SAVER; B1 Broke shows how many correct baselines B1 overwrites.

MagicBrush, B1 breaks 14–21 correct baselines per model while SAVER breaks only 2–4. Even where B1 yields higher aggregate accuracy (Gemini and Qwen on CLEVR, all models on MagicBrush), it does so by overwriting many correct baselines that SAVER would have preserved.

4.3 Per-Change-Type Analysis

To understand why SAVER outperforms B1 for GPT-4o, we examine the per-type breakdown (Table 7). Structured prompting helps color (+17%) and texture (+49%) but hurts drop (-22%) and add (-13%). We hypothesize that structured enumeration forces the model to list all objects, which for drop/add changes introduces confusion: the model lists objects present in both images and struggles to identify which one was added or removed. SAVER selectively applies structured reasoning where it helps, preserving baseline performance on drop/add while capturing most of the gains on color/texture.

The per-type results also reveal why the gate works. For drop changes, correct baselines typically contain absence evidence ("the red cube is no longer present"), which the gate detects and preserves. For color/texture changes, correct answers require specific property transitions ("changed from red to blue"), which baselines often replace with vague descriptions ("something changed about the sphere"). A key finding: in 68% of GPT-4o cases where structured prompting corrected a baseline error (86/126), the baseline already identified the correct object but mischaracterized the change type. Texture changes dominate this pattern (59% of mischaracterizations), as expected for visually subtle material changes. Similar rates hold for Gemini (62%) and LLaMA (65%). Qwen's lower rate (22%) reflects more diverse errors from the smaller 7B model.

Condition	Drop	Add	Color	Texture
B0 (baseline)	87.3	76.9	75.9	51.0
B1 (structured)	65.6†	64.4†	**93.0**	**100.0**
SC-5	**93.0**	**82.5**	80.4	53.5†
SAVER	89.2	76.9	84.8	94.3

Table 7: CLEVR-Change per-type accuracy (%, GPT-4o). **Bold** = best; underline = SAVER; † = below baseline. SC-5 = self-consistency (k=5).

Condition	CLEVR	MB	Calls
B0 (baseline)	72.8	57.6	1.0
B1 (always-structured)	80.7	70.3	1.0
SAVER	**86.2**	**62.9**	1.4
SC-5	77.4	—	5.0
C1: Random	74.7	60.3	1.4
C2: Length	69.0	61.2	1.4
C3: Inverted	68.4†	59.7†	1.4
C4: Generic reprompt	70.3†	54.7†	1.4

Table 8: Ablation results (% accuracy, GPT-4o). Calls = API calls per sample. MB = MagicBrush. **Bold** = best; † = below baseline. SC-5 = self-consistency with k=5. C4 uses SAVER's gate but replaces the structured prompt with a generic retry.

4.4 Ablation: Does the Gate Matter?

The ablations test two dimensions: gate selection quality and reprompt content quality. SAVER outperforms all four ablations on both datasets (Table 8).

For gate selection, Random (C1) and Inverted (C3) gates fall 11.5% and 17.8% below SAVER on CLEVR respectively, with C3 scoring below baseline (68.4% vs. B0 72.8%). For reprompt content, C4 uses SAVER's identical gate but replaces the structured enumeration with a generic retry. C4 is significantly worse than SAVER in 6 of 12 model–dataset combinations (Bonferroni-corrected $p < 0.05$) and sometimes falls below baseline (e.g., GPT-4o CLEVR: 70.3% vs. B0 72.8%), showing that the structured enumeration prompt, not the second chance itself, drives the improvement.

This pattern holds across all four models on CLEVR. On MagicBrush, the Inverted gate slightly outperforms SAVER for Gemini and LLaMA by 2%, consistent with the lower gate precision on real photographs discussed above. Appendix H reports three further gate variants (C5–C7): non-critical keyword triggers, a strict gate, and a loose gate. SAVER outperforms all three on CLEVR; on MagicBrush, the strict C6 gate performs slightly better. The self-consistency comparison is also in Appendix H.

	Model	B0	SAVER	SAVER-L	Trig.
CLEVR	GPT-4o	72.8	86.2	**87.0**	36%
	Gemini	81.5	**93.0**	92.4	23%
	Qwen	56.5	82.3	**84.2**	65%
	LLaMA	82.3	**88.9**	88.8	35%
MB	GPT-4o	57.6	**62.9**	58.5	9%
	Gemini	67.9	**68.8**	67.4	7%
	Qwen	38.5	**56.2**	40.6	14%
	LLaMA	58.5	**63.5**	61.2	26%
STD	GPT-4o	77.2	**78.2**	77.2	14%
	Gemini	67.5	**70.1**	69.0	3%
	Qwen	61.4	**62.9**	62.9‡	16%
	LLaMA	73.1	**78.2**	76.1	27%

Table 9: SAVER (LLM-gen) results (% accuracy). SAVER-L: SAVER with LLM-generated patterns. SAVER = manual patterns. Trig. = SAVER-L trigger rate. **Bold** = best. On CLEVR-Change, all SAVER-L vs. B0 gains are significant ($p < 0.001$, Bonferroni) and no SAVER-L vs. SAVER difference is significant. On MagicBrush, SAVER-L is significantly below SAVER for Qwen ($p < 0.001$, Bonferroni). ‡ = lower bound: Qwen2.5-VL-7B was retired before these experiments, so 23 of 31 gate-triggered samples could not be re-prompted and keep their baseline response. MB = MagicBrush; STD = Spot-the-Diff.

4.5 LLM-Generated Evidence Patterns

The manual gate requires hand-tuned keyword patterns. We test whether an LLM can generate them instead. For each dataset, we prompt Claude Sonnet 4.5 once with the task description, the change-type taxonomy, and 16 example baseline responses (8 correct, 8 incorrect, sampled with seed 42). The output is a complete pattern set in the same format as the manual gate. We use it as-is: no iteration, no manual edits. The generator is not among the four evaluated VLMs. We call this condition SAVER (LLM-gen). Table 9 shows the results for all 12 model–dataset cells.

The result splits by dataset. On CLEVR-Change, single-shot generated patterns fully replace the manual gate for all four models. No difference from the manual gate is significant, and the generated gate is nominally better for GPT-4o and Qwen. The example responses shown to the generator came from GPT-4o logs only, so the patterns also transfer across models. On MagicBrush, the generated patterns fail for all four models. The generated `attribute_change` patterns include bare terms like "changed" and "different" that match nearly any response, so missing evidence becomes undetectable and the gate under-triggers (trigger rates drop to 7–26%, against 11–38% for the manual gate). SAVER (LLM-gen) recovers only a fraction of the manual gain and is significantly below the manual gate for Qwen. On Spot-the-Diff, the generated gate changes nothing. The null result holds under both manual and generated patterns.

Model	B0	SAVER	SAVER-L	+Critique
GPT-4o	57.6	**62.9**	58.5	60.0
Gemini	67.9	**68.8**	67.4	**68.8**
Qwen	38.5	**56.2**	40.6	39.1
LLaMA	58.5	**63.5**	61.2	**63.5**

Table 10: Self-critique on MagicBrush (% accuracy). SAVER-L: SAVER with LLM-generated patterns. +Critique = the SAVER-L patterns after one self-critique pass. **Bold** = best. The critique restores manual-gate accuracy for Gemini and LLaMA, roughly halves the gap for GPT-4o, and fails for Qwen.

Self-critique pass. The MagicBrush failure traces to specific, identifiable patterns, the bare "changed" and "different" terms above. This raises a natural question: can the generator find and fix these patterns itself, without being told what went wrong? We test this with one additional step. The same LLM receives its own generated patterns and one generic instruction: flag any pattern broad enough to match almost any response, and tighten or remove it. The critique is a single pass. The LLM receives no accuracy feedback and no dataset-specific hints. Under this instruction, it removed exactly the bare "changed" and "different" patterns on its own. Table 10 shows the effect.

The result is mixed. For Gemini and LLaMA, one critique pass restores manual-gate accuracy exactly. For GPT-4o, it roughly halves the gap (58.5 to 60.0, against 62.9 manual) and makes the gain over baseline significant ($p = 0.027$, uncorrected). For Qwen, it fails: the critiqued gate still triggers on only 10% of samples, against 38% for the manual gate, and accuracy stays near baseline. Qwen errs on 61% of MagicBrush samples, and removing over-broad patterns cannot by itself restore a trigger rate that high. As a sanity check, we also ran the critique on the CLEVR-Change and Spot-the-Diff pattern sets. Accuracy is unchanged in seven of the eight cells and moves by $+1.0$ in one (Gemini on Spot-the-Diff, still not significant). The critique repairs broken patterns and does not harm working ones. We report single-shot and critiqued results as separate conditions throughout.

5 Discussion

Expression vs. Perception Failures. SAVER's results across three datasets reveal two failure modes. **Expression failures** occur when the model

captures the visual signal but omits key evidence from its output; the gate detects these and reprompting recovers the answer. **Perception failures** occur when the model misses the signal entirely, producing confident, evidence-rich but incorrect outputs that pass the gate. On CLEVR-Change, expression failures dominate (high trigger rates, high fix rates). On Spot-the-Diff, perception failures dominate (low trigger rates or low fix rates). This distinction determines when SAVER helps: the gate adds value when structured prompting helps some input types but hurts others (e.g., GPT-4o on CLEVR-Change), because it selectively applies structured reasoning only where evidence is missing. When errors are predominantly perceptual, B1 outperforms SAVER in aggregate accuracy because even evidence-rich responses may be wrong and the gate's selectivity provides no advantage. However, as the Fix/Broke analysis shows (Section 4.2), B1's accuracy comes at the cost of overwriting correct baselines at 3–9.5$\times$ the rate of SAVER. Which method to prefer depends on the deployment setting: B1 when aggregate accuracy is the only metric, SAVER when overwriting a correct answer carries a higher cost than missing an improvement. More generally, SAVER's type-specific gating is most effective when the task has a structured taxonomy of expected outputs and known evidence requirements (Appendix K).

Measuring the boundary. This boundary can be measured. For each cell we compute the gate's precision at catching baseline errors and compare it to a random gate triggering at the same rate, whose precision equals the baseline error rate (Appendix E). The comparison separates the three datasets cleanly. The gate is above random on CLEVR-Change for all four models, at or below random on Spot-the-Diff, and in between on MagicBrush. The margin between a gate's own precision and the random baseline also predicts the end-to-end gain. For the 12 SAVER (LLM-gen) gates, the Spearman correlation between this margin (Table 15) and the gain over baseline is 0.68, rising to 0.74 when the self-critiqued gates are pooled in. A practitioner can therefore measure gate precision on a small labeled sample and predict whether SAVER will help before full deployment. The keyword quality analysis (Appendix F) adds a second diagnostic. Pattern quality failures come in two kinds, and only one of them is dangerous. With conflict checks intact, over-broad patterns over-trigger and accuracy degrades gracefully toward always-structured prompting (B1). Without conflict checks, the trigger rate collapses silently and accuracy falls back to baseline, the same structure as the MagicBrush failure.

Failure modes. SAVER fails in two ways. First, when the model produces a confident but factually wrong response with correct-sounding evidence (perception failure), the gate accepts it. Second, on rare occasions the structured reprompt introduces errors: the forced enumeration step causes the model to second-guess correct baseline answers, particularly for drop/add types where listing all objects creates confusion about which one changed.

6 Conclusion

We introduced SAVER, a lightweight, rule-based method that checks VLM outputs for task-critical verbal evidence before deciding whether to trigger structured reprompting. SAVER significantly improves accuracy on expression-dominated tasks (up to +25.8% on CLEVR-Change) while showing when and why the approach does not help (perception-dominated errors on Spot-the-Diff). Ablation experiments confirm that the evidence gate's selectivity, not reprompting alone, drives the improvement, and Fix/Broke analysis shows that SAVER degrades far fewer correct answers than always-structured prompting.

Our results reveal a practical lesson: VLMs often know more than they say, and the gap between perception and expression is structured enough to detect with simple keyword rules. Adapting SAVER to a new domain requires only specifying evidence patterns, which can be hand-written or LLM-generated and then checked (Section 4.5). We anticipate the approach to be useful in high-stakes visual comparison settings where transparency and controllable verification are critical.

Limitations

The SAVER gate relies on predefined keyword vocabularies and a fixed proximity window for pattern matching that were set once and not tuned per dataset. Applying SAVER to a new domain requires defining new evidence patterns, and the patterns that work best likely differ across datasets and languages. However, the ablation results (Section 4.4) show that the gate's *logic* matters more than any single keyword: random and inverted gates using the same keywords perform far worse.

A second concern is circularity: the gate uses keyword matching and so does evaluation. We note that the gate checks *evidence categories* (e.g., color change verbs like "changed," "switched") while evaluation checks *specific ground-truth terms* (e.g., exact colors like "red," "blue"). The two keyword sets share zero exact matches for CLEVR-Change: the gate's 23 patterns (Appendix C) contain no object names, colors, or shapes, while the evaluation vocabulary consists entirely of dataset-specific ground-truth terms. The inverted gate (C3) directly tests this: it uses the same 23 keywords but selects the opposite samples. If keyword overlap with evaluation drove the results, C3 should perform comparably to SAVER; instead it scores 17.8% lower (68.4% vs. 86.2%), confirming that gate logic rather than keyword presence drives the improvement. Finally, although we discuss potential applications in high-stakes domains such as medical imaging (Appendix K), SAVER should not be deployed in safety-critical settings without domain-specific validation. More fundamentally, SAVER addresses expression failures only. The gate accepts any response that contains the expected evidence patterns, so confident but factually wrong outputs pass unchecked when errors are perceptual, as the Spot-the-Diff null result illustrates. This scope boundary is measurable before deployment. On a small labeled validation set, gate precision must exceed the random-gate baseline (Appendix E), otherwise errors are not expressive and SAVER should not be expected to help.

Two further limitations come from the LLM-generated pattern experiments. First, one self-critique pass is not a universal fix. It restores manual-gate accuracy on MagicBrush for Gemini and LLaMA and halves the gap for GPT-4o, but it fails for Qwen2.5-VL-7B, where the critiqued gate still under-triggers. Second, the Spot-the-Diff SAVER (LLM-gen) result for Qwen2.5-VL-7B is a lower bound. The model was retired before the revision experiments, so 23 of its 31 gate-triggered samples could not be reprompted and keep their baseline responses.

References

Anthropic. 2024. The Claude model family. Model card, https://platform.claude.com/docs/en/models/overview.

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, and 8 others. 2025. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*. Preprint.

Sriram Balasubramanian, Samyadeep Basu, and Soheil Feizi. 2025. A closer look at bias and chain-of-thought faithfulness of large (vision) language models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 13406–13439, Suzhou, China. Association for Computational Linguistics.

Gautier Evennou, Antoine Chaffin, Vivien Chappelier, and Ewa Kijak. 2025. Reframing image difference captioning with BLIP2IDC and synthetic augmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1392–1402.

Google DeepMind. 2024. Gemini 2.0 flash. Technical blog post, https://deepmind.google/models/gemini/flash/.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, Dinesh Manocha, and Tianyi Zhou. 2024. HallusionBench: An advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14375–14385.

Erdong Hu, Longteng Guo, Tongtian Yue, Zijia Zhao, Shuning Xue, and Jing Liu. 2024. OneDiff: A generalist model for image difference captioning. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, pages 2439–2455.

Zihan Huang, Junda Wu, Rohan Surana, Tong Yu, David Arbour, Ritwik Sinha, and Julian McAuley. 2025. Image difference captioning via adversarial preference optimization. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 33758–33770, Suzhou, China. Association for Computational Linguistics.

Harsh Jhamtani and Taylor Berg-Kirkpatrick. 2018. Learning to describe differences between pairs of similar images. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4024–4034, Brussels, Belgium. Association for Computational Linguistics.

Zaid Khan and Yun Fu. 2024. Consistency and uncertainty: Identifying unreliable responses from black-box vision-language models for selective visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10854–10863.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances*

in Neural Information Processing Systems (NeurIPS), pages 22199–22213.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilė Lukošiūtė, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam McCandlish, Sandipan Kundu, and 11 others. 2023. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*. Preprint.

Yi Li, Yunbin Tu, Liang Li, Li Su, and Qingming Huang. 2025. Change entity-guided heterogeneous representation disentangling for change captioning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 17050–17060, Vienna, Austria. Association for Computational Linguistics.

Yuan Liu, Saihui Hou, Saijie Hou, Jiabao Du, Shibei Meng, and Yongzhen Huang. 2025a. OmniDiff: A comprehensive benchmark for fine-grained image difference captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 21440–21449.

Zhining Liu, Ziyi Chen, Hui Liu, Chen Luo, Xianfeng Tang, Suhang Wang, Joy Zeng, Zhenwei Dai, Zhan Shi, Tianxin Wei, Benoit Dumoulin, and Hanghang Tong. 2025b. Seeing but not believing: Probing the disconnect between visual attention and answer correctness in VLMs. *arXiv preprint arXiv:2510.17771*. Preprint.

Meta AI. 2025. The Llama 4 herd of models. Technical blog post, `https://ai.meta.com/blog/llama-4-multimodal-intelligence/`.

Chancharik Mitra, Brandon Huang, Trevor Darrell, and Roei Herzig. 2024. Compositional chain-of-thought prompting for large multimodal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14420–14431.

OpenAI. 2024. GPT-4o system card. *arXiv preprint arXiv:2410.21276*.

Hadas Orgad, Michael Toker, Zorik Gekhman, Roi Reichart, Idan Szpektor, Hadas Kotek, and Yonatan Belinkov. 2025. LLMs know more than they show: On the intrinsic representation of LLM hallucinations. In *International Conference on Learning Representations (ICLR)*.

Dong Huk Park, Trevor Darrell, and Anna Rohrbach. 2019. Robust change captioning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4624–4633.

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

Pooyan Rahmanzadehgervi, Logan Bolton, Mohammad Reza Taesiri, and Anh Totti Nguyen. 2024. Vision language models are blind. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*.

Adi Simhi, Jonathan Herzig, Idan Szpektor, and Yonatan Belinkov. 2024. Distinguishing ignorance from error in LLM hallucinations. *arXiv preprint arXiv:2410.22071*. Preprint.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2025. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations (ICLR)*.

Tejas Srinivasan, Jack Hessel, Tanmay Gupta, Bill Yuchen Lin, Yejin Choi, Jesse Thomason, and Khyathi Chandu. 2024. Selective "selective prediction": Reducing unnecessary abstention in vision-language reasoning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12935–12948, Bangkok, Thailand. Association for Computational Linguistics.

Linzhuang Sun, Hao Liang, Jingxuan Wei, Bihui Yu, Tianpeng Li, Fan Yang, Zenan Zhou, and Wentao Zhang. 2025. MM-verify: Enhancing multimodal reasoning with chain-of-thought verification. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14100–14115, Vienna, Austria. Association for Computational Linguistics.

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. 2024. Eyes wide shut? exploring the visual shortcomings of multimodal LLMs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9568–9578.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Jiaqi Wang, Kevin Qinghong Lin, James Cheng, and Mike Zheng Shou. 2025. Think or not? selective reasoning via reinforcement learning for vision-language models. *arXiv preprint arXiv:2505.16854*. Preprint.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 24824–24837.

Siye Wu, Jian Xie, Yikai Zhang, Aili Chen, Kai Zhang, Yu Su, and Yanghua Xiao. 2025a. ARM: Adaptive reasoning model. In *Advances in Neural Information Processing Systems 38 (NeurIPS 2025)*.

Tsung-Han Wu, Heekyung Lee, Jiaxin Ge, Joseph E. Gonzalez, Trevor Darrell, and David M. Chan. 2025b. Generate, but verify: Reducing hallucination in vision-language models with retrospective resampling. In *Advances in Neural Information Processing Systems 38 (NeurIPS 2025)*.

Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Qiaowei Li, Minghui Chen, Zheng Lin, and Weiping Wang. 2025. Dynamic early exit in reasoning models. *arXiv preprint arXiv:2504.15895*. Preprint.

Jianshu Zhang, Dongyu Yao, Renjie Pi, Paul Pu Liang, and Yi R. Fung. 2025. VLM2-bench: A closer look at how well VLMs implicitly link explicit matching visual cues. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7510–7545, Vienna, Austria. Association for Computational Linguistics.

Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. 2023. MagicBrush: A manually annotated dataset for instruction-guided image editing. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. 2024. Multimodal chain-of-thought reasoning in language models. *Transactions on Machine Learning Research (TMLR)*.

A Gate Decision Rules

The gate maps extracted evidence vectors to a binary decision per dataset.

CLEVR-Change. The gate identifies the model's predicted change type, then checks:

- **Drop** requires `absence` evidence. `presence` detected = conflict.
- **Add** requires `presence` evidence. `absence` detected = conflict.
- **Color** requires `color_change` evidence. `material_change` = conflict.
- **Texture** requires `material_change` evidence. `color_change` = conflict.

MagicBrush. Five change types map to required vectors: remove→`absence`, add→`presence`, color→`color_change`, replace→`replacement`, attribute→`attribute_change`. Only remove and add have conflicting-evidence entries (`presence` and `absence` respectively). Color, replace, and attribute have none.

Spot-the-Diff. The gate checks whether the response contains *any* change evidence. If the model reports "no difference" while mentioning specific changes, this conflict triggers a reprompt.

B Prompts

We provide the exact prompts used for each dataset and condition. All prompts are sent with the image pair as visual input.

B.1 CLEVR-Change Prompts

Baseline (B0).

> You are looking at two images of a scene with 3D objects. The second image shows the same scene after ONE change was made. Describe precisely what changed between Image 1 and Image 2. Be specific about: what object changed, what the change was (added, removed, moved, color changed, material changed), and where in the scene the change occurred.

Structured (B1 / SAVER reprompt).

> You are looking at two images of a scene with 3D objects. The second image shows the same scene after ONE change was made. Follow these steps carefully:
>
> Step 1: Describe ALL objects in Image 1, listing each object's shape, color, material (shiny/matte), size (small/large), and approximate position.
>
> Step 2: Describe ALL objects in Image 2, listing the same properties.
>
> Step 3: Compare the two descriptions systematically to identify what changed.
>
> Step 4: State your final answer: what object changed, what the change was (added, removed, moved, color changed, material changed), and where in the scene the change occurred.

Standard CoT (B2).

> You are looking at two images of a scene with 3D objects. The second image shows the same scene after ONE change was made. Think step by step about what changed between Image 1 and Image 2. Consider each object carefully and compare their properties. Then provide your final answer: what object changed, what the change was (added, removed, moved, color changed, material changed), and where in the scene the change occurred.

B.2 MagicBrush Prompts

Baseline (B0).

> You are looking at two images. The second image shows the same scene after ONE change was made. Describe precisely what changed between Image 1 and Image 2. Be specific about: what object changed, what the change was (added, removed, replaced, color changed, modified), and where in the scene the change occurred.

Structured (B1 / SAVER reprompt).

> You are looking at two images. The second image shows the same scene after ONE change was made. Follow these steps carefully:
>
> Step 1: Describe the key objects and elements in Image 1, noting their appearance, color, position, and state.
>
> Step 2: Describe the key objects and elements in Image 2, noting the same properties.
>
> Step 3: Compare the two descriptions systematically to identify what changed.
>
> Step 4: State your final answer: what object changed, what the change was (added, removed, replaced, color changed, modified), and where in the scene the change occurred.

B.3 Spot-the-Diff Prompts

Baseline (B0).

> You are looking at two images of the same scene taken at different times. Identify ALL differences between Image 1 and Image 2. For each difference, describe what changed: what object appeared, disappeared, moved, or changed in some way. Be specific about the objects and their locations.

Structured (SAVER reprompt).

> You are looking at two images of the same scene taken at different times. Follow these steps carefully:
>
> Step 1: Describe the main objects and people visible in Image 1, noting their positions (left/right/center, foreground/background).
>
> Step 2: Describe the main objects and people visible in Image 2, noting the same properties.
>
> Step 3: Compare systematically — for each object in Image 1, check if it exists in Image 2 and vice versa. Note any objects that appeared, disappeared, or moved.
>
> Step 4: State your final answer: list ALL differences found, specifying for each: what object, what type of change (appeared, disappeared, moved, changed), and where in the scene.

B.4 Generic Reprompt (C4)

The C4 ablation uses a multi-turn format: the baseline prompt and response are included as context, followed by this generic reprompt.

> I need you to reconsider your previous answer. Please look at the two images again carefully and provide a more thorough response.

C Evidence Gate Keywords

Table 11 and Table 12 list the keyword patterns used by the SAVER evidence gate. All matching is case insensitive. Proximity patterns (marked $\sim$) match a change verb near a property term, allowing up to 20 intervening characters (e.g., "changed the color" matches `color_change`).

Vector	Keywords / Patterns
`absence`	missing, absent, gone, disappeared, removed, no longer, not present
`presence`	new, added, appeared, introduced, additional, placed
`color_change`	{changed, switched, turned} $\sim$ {color}; also detects color-pair mentions (e.g., "red to blue")
`material_change`	{changed, switched} $\sim$ {material, texture, matte, shiny}

Table 11: CLEVR-Change evidence keywords.

Vector	Keywords / Patterns
`absence`	missing, absent, gone, disappeared, removed, no longer, not present, deleted, taken away, erased
`presence`	new, added, appeared, introduced, additional, placed, now has/contains/includes/shows
`color_change`	{changed, different, switched, turned, became} $\sim$ {color, colour, hue}; also from-to patterns
`replacement`	replaced, substituted, swapped, changed … to/into/for/with
`attribute_change`	modified, altered, transformed, different … shape/size/position

Table 12: MagicBrush evidence keywords.

D Statistical Details

Table 13 reports bootstrap confidence intervals and effect sizes for all model–dataset pairs. Figure 3 shows the forest plot.

	Model	Δ%	95% CI
CLEVR	GPT-4o	+13.4***	[10.8, 16.5]
	Gemini	+11.6***	[9.3, 13.9]
	Qwen	+25.8***	[22.2, 29.4]
	LLaMA	+6.6***	[3.8, 9.3]
MB	GPT-4o	+5.3**	[2.6, 8.2]
	Gemini	+0.9	[$-$0.6, 2.4]
	Qwen	+17.6***	[13.8, 22.4]
	LLaMA	+5.0**	[2.4, 7.6]

Table 13: SAVER vs. baseline (Δ% with bootstrap 95% CIs). $^{***}\,p < 0.001$; $^{**}\,p < 0.05$; no mark = n.s. MB = MagicBrush. Cohen's h: 0.19–0.57 (CLEVR), 0.02–0.36 (MB).

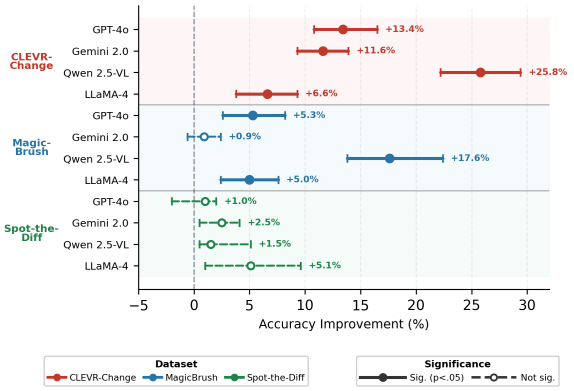


Figure 3: SAVER improvement over B0 (baseline) in percentage points, with 95% bootstrap CIs across all 12 model–dataset combinations. Solid/filled = significant ($p < 0.05$); dashed/hollow = non-significant.

E Gate Precision Analysis

We evaluate how well the manual gate separates wrong baseline responses from correct ones. For each model–dataset cell we run the gate on the cached baseline responses and label each sample against baseline correctness. A true positive is a triggered sample whose baseline response is wrong. A false positive is a triggered sample whose baseline response is correct. Gate precision is the fraction of triggered samples that are true positives. The reference point is a random gate triggering at the same rate. Its precision equals the baseline error rate, denoted P_{rand}. A gate is informative only if its precision exceeds P_{rand}. This analysis uses existing logs and requires no new API calls.

Table 14 shows the result. On CLEVR-Change, precision exceeds P_{rand} by 0.08 to 0.26 absolute for all four models. On Spot-the-Diff, precision is at or below P_{rand} for all four models. MagicBrush lies in between, with margins from -0.08 to $+0.18$. The cells where the gate beats the random baseline are the cells where SAVER helps end-to-end (Section 5).

	Model	Trig.	Prec.	Rec.	P_{rand}
CLEVR	GPT-4o	0.39	**0.508**	0.733	0.272
	Gemini	0.36	**0.371**	0.726	0.185
	Qwen	0.49	**0.691**	0.782	0.435
	LLaMA	0.37	**0.255**	0.536	0.177
MB	GPT-4o	0.22	**0.500**	0.257	0.424
	Gemini	0.11	0.243	0.083	0.321
	Qwen	0.38	**0.792**	0.493	0.615
	LLaMA	0.30	**0.446**	0.319	0.415
STD	GPT-4o	0.14	0.143	0.089	0.228
	Gemini	0.17	0.294	0.156	0.325
	Qwen	0.12	0.208	0.066	0.386
	LLaMA	0.57	0.239	0.509	0.269

Table 14: Manual-gate precision and recall against the baseline-correctness oracle, all 12 cells. Trig. = trigger rate. P_{rand} = precision of a random gate at the same trigger rate (the baseline error rate). **Bold** = precision above P_{rand}. MB = MagicBrush; STD = Spot-the-Diff.

Table 15 reports the same precision measure for the generated gates of Section 4.5, single-shot and after the self-critique pass. P_{rand} is shared with Table 14, since it depends only on the baseline error rate. These are the precision values behind the correlation in Section 5.

	Model	LLM-gen	+Critique
CLEVR	GPT-4o	0.578	0.578
	Gemini	0.549	0.549
	Qwen	0.571	0.571
	LLaMA	0.287	0.278
MB	GPT-4o	0.355	0.487
	Gemini	0.217	0.353
	Qwen	0.521	0.457
	LLaMA	0.333	0.384
STD	GPT-4o	0.111	0.088
	Gemini	1.000	0.500
	Qwen	0.290	0.289
	LLaMA	0.222	0.241

Table 15: Gate precision for the LLM-generated gates, single-shot (LLM-gen) and self-critiqued (+Critique), all 12 cells. P_{rand} as in Table 14. The Spot-the-Diff Gemini single-shot gate triggers on only 6 of 197 samples. MB = MagicBrush; STD = Spot-the-Diff.

F Keyword Quality Analysis

The MagicBrush result in Section 4.5 shows that pattern quality determines whether SAVER works. Here we vary pattern quality directly. We re-implement the hand-tuned CLEVR-Change gate in a parameterized form that reproduces its per-sample decisions exactly, then degrade the evidence patterns in controlled steps and measure accuracy, trigger rate, and gate precision for GPT-4o ($N = 632$). All conditions reuse cached responses.

Breadth. We broaden the patterns in four cumulative levels. L0 is the manual gate. L1 widens the

proximity windows by 2.5×. L2 adds bare generic change verbs ("changed", "different") to the color and material categories, the same shape as the over-broad generated MagicBrush patterns. L3 adds near-universal words ("now", "image", "second") to every category. We run each level twice: once with the conflict checks intact (CLEVR-Change's gate design) and once with them disabled, which is the structure of MagicBrush's `attribute_change` category (Appendix A). Table 16 shows both regimes.

	Level	Trig.	Prec.	Rec.	Acc.
Conflict checks intact	L0 manual	0.39	0.508	0.733	86.2
	L1 windows	0.37	0.532	0.721	85.9
	L2 verbs	0.60	0.382	0.843	87.3
	L3 ubiq.	1.00	0.272	1.000	80.2
Conflict checks disabled	L0 manual	0.34	0.583	0.733	87.3
	L1 windows	0.29	0.663	0.709	87.0
	L2 verbs	0.12	1.000	0.430	80.5
	L3 ubiq.	0.00	—	0.000	72.8

Table 16: Breadth degradation of the hand-tuned CLEVR-Change gate (GPT-4o). Levels are cumulative. $P_{\text{rand}} = 0.272$ throughout. Reference points: B0 = 72.8, B1 = 80.7.

The two regimes fail differently. With conflict checks intact, broad patterns make several evidence categories fire at once. The gate over-triggers, and at L3 every sample is reprompted. Accuracy degrades gracefully toward always-structured prompting (80.2 at L3, against B1 at 80.7). The cost is wasted calls, not silent failure. With conflict checks disabled, the same broadening makes the required evidence look present in nearly every response. Missing evidence becomes undetectable, the trigger rate collapses from 0.34 to zero, and accuracy falls back to the 72.8 baseline. Nothing in the gate's own output signals the failure, since the few remaining triggers have perfect precision. This is the same structure as the MagicBrush failure in Section 4.5, reproduced in a controlled setting. Figure 4 plots both regimes.

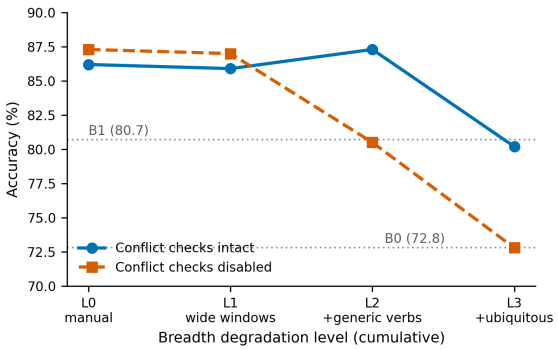


Figure 4: Breadth degradation of the hand-tuned CLEVR-Change gate (GPT-4o), both conflict-check regimes. Dotted lines mark the baseline (B0) and always-structured (B1) accuracy. With conflict checks intact, broadening over-triggers and degrades gracefully toward B1. Without them, the gate under-triggers silently and falls back to B0.

Coverage. We remove 25%, 50%, or 75% of the pattern branches in each evidence category (three random draws, seed 42, at least one branch kept per category). Table 17 shows the result. Missing keywords push the gate the other way. Evidence looks absent more often, the gate over-triggers, precision falls toward P_{rand}, and accuracy stays in the 81.6–86.6 band, well above baseline. Missing keywords are therefore the graceful failure mode. Over-broad keywords without conflict checks are the silent one.

Dropped	Trig.	Prec.	Acc.
0% (manual)	0.39	0.508	86.2
25%	0.66 ± 0.09	0.36 ± 0.04	82.8 ± 1.7
50%	0.53 ± 0.06	0.44 ± 0.05	86.6 ± 3.0
75%	0.92 ± 0.10	0.29 ± 0.02	81.6 ± 1.7

Table 17: Coverage degradation of the hand-tuned CLEVR-Change gate (GPT-4o). Mean ± sd over three draws. $P_{\text{rand}} = 0.272$.

G Cost Comparison

Table 18 compares SAVER to always-structured (B1) and self-consistency (SC-5, GPT-4o only). On CLEVR-Change, SAVER outperforms B1 for GPT-4o (+5.5%) and LLaMA (+2.7%) while using only 1.37–1.39 calls, because the gate avoids structured prompting where it hurts. SC-5 achieves 77.4% at 5.0 calls/sample, below both SAVER (86.2% at 1.39 calls) and B1 (80.7% at 1.0 calls), making it the least cost-effective strategy on this dataset. For Gemini and Qwen, B1 is more accurate, meaning always structured is the better strategy. On MagicBrush, B1 outperforms SAVER for all models by 7–13%, because errors there are predominantly

perceptual and benefit from universal structured prompting. Thus, SAVER works best when structured prompting helps some input types but hurts others.

	Model	B1	SC-5	SAVER	Calls
CLEVR	GPT-4o	80.7	77.4	**86.2**	1.39
	Gemini	**95.6**	—	93.0	1.36
	Qwen	**87.0**	—	82.3	1.49
	LLaMA	86.2	—	**88.9**	1.37
MB	GPT-4o	**70.3**	—	62.9	1.22
	Gemini	**79.7**	—	68.8	1.11
	Qwen	**68.8**	—	56.2	1.38
	LLaMA	**74.4**	—	63.5	1.30

Table 18: Accuracy (%) vs. API cost. B1 uses 1.0 calls; SAVER averages 1.11–1.49 (Calls column); SC-5 uses 5.0 calls (GPT-4o CLEVR only). **Bold** = highest accuracy.

H Ablation Details

Beyond the GPT-4o ablations in the main paper, we tested three additional gate configurations (Table 19).

C5: Non-critical keyword trigger. C5 replaces SAVER's evidence-based trigger with spatial and background keywords (e.g., *left*, *right*, *behind*, *center*). The trigger count is matched to SAVER's so the two conditions differ only in *which* samples are reprompted. On CLEVR-Change, C5 underperforms SAVER by 4–8% across all four models (Table 19). The gap is largest for GPT-4o (78.5% vs. 86.2%). Spatial keywords appear in both correct and incorrect responses, so C5 reprompts many samples that do not need it while missing samples that do. On MagicBrush the gap narrows, and LLaMA is the one exception where C5 slightly outperforms SAVER (64.1% vs. 63.5%). This result shows that evidence-type matching, not just keyword presence, is what makes the gate effective.

C6: Strict gate. C6 triggers whenever *any* evidence dimension is missing, regardless of the predicted change type. This is the most aggressive variant: trigger rates reach 88–100% on CLEVR and 55–79% on MagicBrush, meaning nearly every sample gets reprompted. On CLEVR, C6 approximates always-structured (B1) and scores below SAVER for GPT-4o and LLaMA, the two models where B1 hurts drop/add types. For Gemini and Qwen, C6 slightly exceeds SAVER because these models benefit from structured reasoning on all input types. On MagicBrush, C6 outperforms SAVER for all four models. MagicBrush errors are predominantly perceptual, so broader triggering compensates: even when the gate cannot distinguish expression from perception failures, the structured prompt still helps by forcing more careful descriptions of real photographs.

C7: Loose gate. C7 triggers only when conflicting evidence is detected (e.g., the response claims "color change" but contains material-related terms). This is the most conservative variant, with trigger rates of 5–33%. On CLEVR, C7 scores below SAVER by 4–16% across all models. The largest drop is for Qwen (66.0% vs. 82.3%), where many incorrect baselines simply lack evidence rather than containing conflicting evidence, so C7 misses them. On MagicBrush, C7 performs close to baseline for most models. The contrast between C6 and C7 shows that SAVER's standard gate occupies a productive middle ground: strict enough to catch evidence-poor responses, but not so strict that it reprompts samples the model already handles well.

Self-consistency comparison. SC-5 scores below both SAVER and B1 on CLEVR-Change despite 3.6× more API calls (Table 8). The per-type breakdown (Table 7) explains why: SC-5 barely helps texture, where GPT-4o consistently mislabels texture as color. Majority voting amplifies this confusion. SAVER sidesteps the problem by checking for material/texture keywords directly, correcting 88 samples that SC-5 misses while SC-5 corrects only 32 that SAVER misses.

SAVER averages 1 + trigger_rate calls per sample. For GPT-4o on CLEVR-Change (39% trigger rate), this is 1.39 calls, a 39% overhead that yields +13.4% accuracy. Yet SC-5 requires 5 calls for a smaller gain. The full cost–accuracy comparison is in Appendix G.

I Per-Type Visualization

Figure 5 visualizes the per-type accuracy breakdown from Table 7 in the main paper.

Condition	Trig.	CLEVR-Change GPT-4o	Gemini	Qwen	LLaMA	MagicBrush GPT-4o	Gemini	Qwen	LLaMA
SAVER (ref)	11–49%	**86.2**	93.0	82.3	**88.9**	62.9	68.8	56.2	63.5
C5 Non-critical	matched	78.5	86.4	76.3	84.5	61.5	69.1	52.4	64.1
C6 Strict gate	55–100%	81.8	**95.6**	**87.2**	87.0	**66.8**	**77.1**	**64.7**	**67.4**
C7 Loose gate	5–33%	81.5	88.0	66.0	85.0	61.5	69.1	55.0	59.4

Table 19: Gate design sensitivity analysis (% accuracy). C5 = non-critical keyword trigger; C6 = strict gate (any missing evidence); C7 = loose gate (conflict only). **Bold** = best per column. Spot-the-Diff is excluded (no B1 structured data for these conditions).

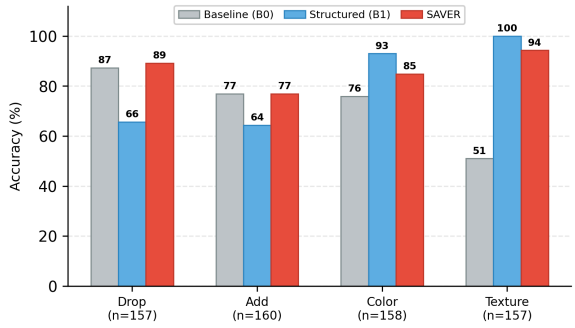


Figure 5: CLEVR-Change per-type accuracy for GPT-4o. Structured prompting (B1) hurts drop and add types but helps color and texture. SAVER selectively applies structured reasoning where it helps.

J Exploratory: Claude 3.5 Sonnet

We ran Claude 3.5 Sonnet (Anthropic, 2024) on a 200-sample CLEVR-Change subset as an exploratory comparison. Table 20 shows the results. This is one of the cleanest cases for selective gating: SAVER outperforms B1 on accuracy (77.5% vs. 76.5%) while breaking zero correct baselines. B1 fixes 27 samples but breaks 12 (6.0% broke rate), yielding a net gain of +15. SAVER fixes 17 and breaks 0, for a net gain of +17. SAVER achieves both higher accuracy and better robustness simultaneously because the gate avoids all collateral damage from unnecessary reprompting.

Condition	Accuracy	Fix	Broke	Trigger
Baseline (B0)	69.0%	—	—	—
Structured (B1)	76.5%	27	12	—
SAVER	**77.5%**	17	0	20%

Table 20: Claude 3.5 Sonnet on CLEVR-Change (200 samples). **Bold** = best. SAVER outperforms B1 with zero breaks.

K Domain Applicability

SAVER applies when a task has a structured output taxonomy and known evidence requirements per output type. Change detection satisfies both: change types (drop, add, color, texture) form the taxonomy, and each type requires specific verbal evidence (absence language, color transition phrases). We describe three other domains where these conditions hold and what adaptation would involve.

Manufacturing defect inspection. Defect categories such as scratch, dent, discoloration, and missing component form a taxonomy. Each category maps to visual evidence: scratches require directional terms ("along the edge," "linear mark"), discoloration requires color deviation terms ("darker region," "staining"). A gate would check whether the VLM's defect report contains the evidence that the claimed category requires. In manufacturing, a missed defect is typically far costlier than a false alarm. This favors higher trigger rates, closer to our C6 strict-gate variant (Appendix H), where the gate reprompts aggressively and accepts the cost of redundant calls.

Medical image comparison. Comparing longitudinal medical images (e.g., chest X-rays or MRI scans over time) involves detecting changes in lesion size, new findings, or resolved conditions. The output taxonomy includes categories like new lesion, growth, shrinkage, and stable. A "growth" claim should reference size measurements or comparative terms ("larger," "increased in diameter"), giving clear evidence requirements. Two challenges separate this domain from change detection. First, subtle changes may fall below VLM perceptual thresholds, so errors would be perceptual rather than expressive. Second, image registration artifacts can produce spurious visual differences. Our Spot-the-Diff results, where perception failures dominate, suggest SAVER would provide less benefit in such settings unless paired with domain-specific preprocessing to reduce perceptual noise.

Document change tracking. Comparing document versions (contracts, regulatory filings, technical specifications) is a structured comparison task with clear change types: additions, deletions, modifications, and reformulations. A "modification"

claim should cite the specific text that changed and how. Because document comparison is predominantly textual, the perception failure mode that limits SAVER on natural images is less relevant. This makes document comparison a strong candidate for evidence gating.

General adaptation. Applying SAVER to a new domain requires four steps: (1) define the output taxonomy, (2) specify required evidence keywords per type, (3) optionally define conflicting evidence pairs, and (4) validate the gate against the random-gate baseline on a small labeled sample (Appendix E). Steps 1–3 can be automated by prompting an LLM with a task description and example outputs (Section 4.5). The main requirement is that the domain exhibits expression failures, where the model has the right information but omits it from its output. Domains where errors are mostly perceptual will benefit less, as our results on Spot-the-Diff confirm, and on MagicBrush to a lesser degree.

L Spot-the-Diff Case Study: Perception Failures

To verify that Spot-the-Diff errors are perceptual rather than expressive, we examined gate-accepted responses that were scored as incorrect. Table 21 shows the perception failure rate across all four models: 27–46% of gate-accepted responses are wrong despite containing evidence language. The gate finds change-related keywords (disappearance, movement, appearance) and accepts the response, but the described changes do not match the ground truth. Note that these accepted-but-wrong cases do not appear in the Fix/Broke counts (Table 6), which only count samples where SAVER changed the outcome through reprompting. The cases here were never reprompted.

Model	Accepted	Wrong	Perc. Fail. %
GPT-4o	169	45	26.6
Gemini	163	61	37.4
Qwen	173	79	45.7
LLaMA	84	28	33.3

Table 21: Perception failure rates on Spot-the-Diff. "Accepted" = gate found evidence and did not trigger. "Wrong" = accepted but incorrect. These are cases where the model describes changes confidently but inaccurately.

Three GPT-4o examples illustrate the pattern. In each case, the gate accepts the response because evidence keywords are present, but the described changes do not match the ground truth.

Example 1 (ID 10313)

Ground truth: the red car is no longer there.
GPT-4o: A white car moved forward, a black car shifted right, a person appeared.
Gate: Accepts (movement + appearance evidence found). None of the three described changes match the actual disappearance.

Example 2 (ID 8648)

Ground truth: Several people are missing; a few new people appear.
GPT-4o: A cyclist disappeared; a walking person shifted left.
Gate: Accepts (disappearance + movement evidence found). The response describes the wrong people and wrong changes.

Example 3 (ID 9900)

Ground truth: A person appeared by the dumpster; two background people disappeared; a man in yellow appeared.
GPT-4o: A white car disappeared; a silver car moved forward.
Gate: Accepts (disappearance + movement evidence found). The actual changes involve people, but the model describes vehicles.

The model produces confident, evidence-rich descriptions in all three cases. The gate cannot catch these errors because the responses contain valid evidence keywords. The failures are perceptual, not expressive.

M Hyperparameters and Reproducibility

All experiments use the following settings:

- **Temperature**: 0 for all conditions except self-consistency (k=5, temperature 0.7)
- **Max tokens**: 300 (baseline), 600 (structured/SAVER reprompt), 400 (standard CoT)
- **Image detail**: `low` for GPT-4o, `auto` for OpenRouter models. Images are base64-encoded PNG.
- **Retry policy**: up to 3 retries with exponential backoff (30–120 s) on rate-limit or server errors
- **Random seed**: 42 for all stochastic operations (sample selection, bootstrap, ablation randomization)
- **Bootstrap**: 1,000 resamples, 95% confidence intervals

- **Significance testing**: McNemar's test with continuity correction, Bonferroni adjusted across 12 model–dataset pairs. The LLM-generated pattern and self-critique experiments form a separate test family with their own Bonferroni correction (24 tests). The original tests and their significance markers are unchanged.

Table 22 lists the exact model IDs and API providers. The main experiments ran between January and February 2026. The LLM-generated pattern and self-critique experiments ran in July 2026. They reuse the cached January–February responses wherever a gate decision matches an already-run condition. Only 38 new Spot-the-Diff structured reprompts were needed: 28 for GPT-4o and 10 for LLaMA-4-Scout. By July 2026, the original GPT-4o endpoint was replaced by OpenRouter's `openai/gpt-4o` (same underlying model, image detail `low` as before). Gemini 2.0 Flash and Qwen2.5-VL-7B had been retired by their providers before the revision experiments, so their Spot-the-Diff cells rely on cached responses only. Pattern generation and self-critique used Claude Sonnet 4.5, which is not among the evaluated VLMs. Per dataset, single-shot pattern generation cost \$0.02–\$0.04 and 16–28 seconds, and the self-critique pass cost \$0.02–\$0.05 and 14–29 seconds.

Model	API / Model ID
GPT-4o	OpenAI: `gpt-4o`
Gemini 2.0 Flash	OpenRouter: `gemini-2.0-flash-001`†
Qwen2.5-VL-7B	OpenRouter: `qwen-2.5-vl-7b-instruct`†
LLaMA-4-Scout	OpenRouter: `llama-4-scout`
Claude 3.5 Sonnet	OpenRouter: `claude-3.5-sonnet`
GPT-4o (revision)	OpenRouter: `openai/gpt-4o`
Claude Sonnet 4.5	Anthropic: `claude-sonnet-4-5`

Table 22: Model versions and API endpoints. Top block: main experiments, January–February 2026. Bottom block: revision experiments, July 2026 (28 new GPT-4o Spot-the-Diff reprompts via OpenRouter; Claude Sonnet 4.5 as pattern generator and critic). † = retired by the provider before the revision experiments.