REKEY: Metadata-Grounded Visual-Key Regeneration for Contamination-Resistant VQA Evaluation

Anonymous ACL submission

Abstract

Static VQA benchmarks age quickly: once items leak into training corpora, scores can reflect memorization rather than genuine visual ability, obscuring real progress. Rebuilding high-quality benchmarks such as V*Bench (Wu and Xie, 2024) requires substantial human annotation, yet each static release can quickly become another leaked artifact. We propose REKEY, a live benchmark protocol that randomly regenerates the answer-bearing local detail, or visual key, in real images at evaluation time. Using human-validated edit slots, REKEY samples fresh instances with new answers, construction-grounded labels, and controlled visual-search difficulty. On V*Bench, the REKEY regenerated benchmark reveals a sharp score jump across eight frontier VLMs: original items score 9.5–18.8 percentage points higher than regenerated variants. By making the visual key renewable, REKEY keeps evaluation fresh as models and training data evolve.[1]

1 Introduction

Static VQA benchmarks become less reliable as soon as benchmark items leak into training corpora or repeated development cycles. At that point, a score can mix perception with recall, and it becomes harder to tell whether a model is actually searching the image or just remembering the answer. That problem is sharpest when the label depends on a small visual detail that is hard to locate.

V*Bench (Wu and Xie, 2024) makes this failure mode visible. Each of its 191 questions depends on locating a *visual key*: a small, peripheral detail in the image that determines the correct answer. When we pair the same images with fresh questions that target different visual keys, every evaluated model drops by 9.5–18.8 percentage points (Figure 1). That gap suggests that the original scores are not only measuring visual search, but also benefiting from memorized answer-bearing details.

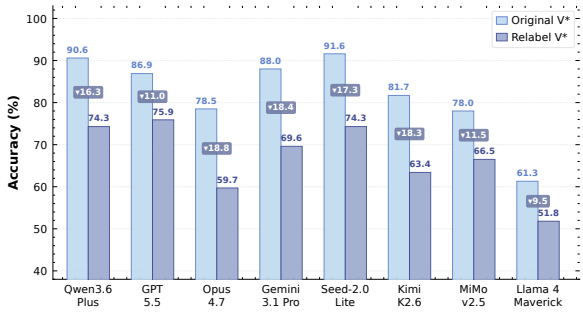

Figure 1: Across eight evaluated models, accuracy drops by 9.5–18.8 percentage points when the same images are paired with fresh human-written questions that target different visual keys, suggesting that original V*Bench scores are partly inflated by memorized answer-bearing details (details in Appendix A).

Existing dynamic evaluation methods address parts of this problem but none resolves all three needs simultaneously: preserving the original visual-search difficulty, renewing the leaked answer, and keeping labels reliable without a model in the verification loop. Approaches that source new content, whether from recent publications (Shabtay et al., 2025; White et al., 2025) or from text-to-image synthesis (Bao et al., 2024), can produce fresh answers but sacrifice control over whether each item matches the original visual difficulty. Methods that preserve the benchmark structure, such as Vision-Language Bootstrapping (Yang et al., 2025), maintain difficulty but often keep the original answer, which is exactly what a contaminated model may remember. Parametric regeneration shows that answers can be derived from the construction process (Zou et al., 2025), but existing examples are limited to math or diagrammatic settings.

For natural-image VQA, resolving these gaps simultaneously yields three constraints. (1) *Pre-*

[1]Code and data: https://anonymous.4open.science/r/rekey

serve difficulty: the regenerated key must remain a small, plausible target in the original cluttered scene, rather than becoming a salient insertion or a synthetic replacement for the image. (2) *Renew the answer*: each benchmark instance must compose its answer from newly sampled visual evidence, not reuse the leaked object–attribute association or question–answer pair. (3) *Keep labels reliable*: the correct answer should follow from the construction process itself, since a VLM asked to inspect small edited regions can miss details or inherit the same visual biases we aim to evaluate.

We meet these constraints with REKEY, short for Metadata-Grounded Visual-Key Regeneration. REKEY treats the visual key, rather than the whole VQA item, as the unit of renewal. We instantiate the framework on V*Bench, which exemplifies the small-target visual-search setting where contamination poses the greatest threat to evaluation validity. Human annotators mark edit slots in the V*Bench images once; at evaluation time, a rule-based pipeline samples fresh visual content, applies a localized edit, and derives the ground-truth label from the same sampling record that produced the edit. A single set of annotations thus supports many fresh benchmark instances, each with a different correct answer derived directly from the construction process.

Our contributions are:

1. **Visual-key regeneration.** We introduce visual-key regeneration as a contamination-resistant evaluation principle for VQA: renewing the specific detail that determines the answer while preserving the visual-search difficulty, turning each source image into many fresh instances with different correct answers.

2. **Metadata-grounded answer construction.** We propose a construction process that produces both the edited image and its ground-truth label jointly, so answers are traceable to the construction rather than dependent on model judgment.

3. **Comprehensive V*Bench evaluation.** Pairing the same images with fresh human-written questions reveals that original V*Bench scores across eight VLMs are 9.5–18.8 percentage points higher than warranted. REKEY closes this gap while preserving the original difficulty and consistent scores, and we further validate the necessity of human curation over VLM self-annotation.

Overall, REKEY shows that contamination resistance and evaluation fidelity need not be in tension: by renewing only the visual key that determines the answer, the same source images can support fresh, comparable evaluation indefinitely.

2 Related Work

Multimodal evaluation benchmarks. Multimodal evaluation has moved from broad capability suites to benchmarks that stress fine-grained evidence use. V*Bench (Wu and Xie, 2024) frames visual search as a core VLM capability, with each question requiring the model to locate a visual key in a cluttered image. Recent benchmarks extend this pressure to high-resolution and dense scenes (Zhang et al., 2025b,c; Gavrikov et al., 2025). These benchmarks motivate our setting, but they remain static once their fixed image–question–answer items are exposed.

Data contamination and dynamic evaluation. Data contamination has become a central concern for multimodal evaluation: recent studies show leakage through both image and text channels (Song et al., 2025), and show that models can exploit non-visual shortcuts when benchmark items are exposed (Brown et al., 2025). A recent survey therefore treats dynamic evaluation as a central response to contamination (Chen et al., 2025). One line uses temporal freshness: LiveBench (White et al., 2025) and LiveXiv (Shabtay et al., 2025) collect new items from recent sources. Another line uses regeneration or transformation: DynaMath (Zou et al., 2025) instantiates mathematical VQA problems from programs, while Vision-Language Bootstrapping (Yang et al., 2025) rebuilds image-question pairs through visual and language-side transformations. Together, these methods show three useful directions: sourcing unseen content, varying existing items, and constructing answers from generation records. Scarce-visual-key evaluation requires these directions to meet at a more specific point: the benchmark must keep the original visual-search problem while replacing the small piece of evidence that determines the answer.

Automatic benchmark construction and verification. Structured VQA provides a long-standing basis for reliable benchmark construction. CLEVR (Johnson et al., 2017) and GQA (Hudson and Manning, 2019) derive answers from structured scene

representations, making labels explicit rather than post-hoc. Recent work further automates benchmark construction with models: AutoBench-V (Bao et al., 2024) uses text-to-image generation and VLM orchestration to create on-demand VQA evaluations, while KBE-DME (Zhang et al., 2025a) evolves static VQA samples by re-selecting visual information or adding external knowledge. Such pipelines scale benchmark renewal, but scarce-visual-key evaluation adds a specific constraint: the construction process must not move the evidence toward salient or easy regions, and the label should not depend on a model judge that may inherit evaluation-time biases. Recent VLM-as-a-judge work reports modality neglect, instability, and systematic preferences under controlled perturbations (Lee et al., 2026). REKEY therefore uses human annotation to identify where a difficult visual key can exist, then uses rule-based sampling and metadata-grounded answer construction to instantiate fresh evidence.

3 Method

Refreshing a VQA item means changing the detail that determines the answer without changing how hard it is to find. Given a source image I, question Q, and answer A, REKEY produces a fresh item (I', Q', A') with a different visual key but comparable visual-search difficulty, by separating a one-time annotation step from a fully automated evaluation-time pipeline. Human annotators examine each V*Bench source image and mark edit slots $\{s_1, \ldots, s_K\}$, compact regions where a visual key can be renewed while remaining natural and non-obvious (§3.1). At evaluation time, a *sampling record* r selects a slot and assigns it fresh visual content (§3.2); a localized edit produces I' from I (§3.3), and a rule-based constructor derives Q' and A' directly from r (§3.4). Because the edit and the label are both determined by r, the ground truth is fixed before any model sees the result.

3.1 Human-Annotated Edit Slots

Each edit slot s_k is an annotated region of the source image that specifies where a visual key may be renewed without changing the visual-search difficulty of the original item. Within s_k, a smaller edit region s_k^{edit} marks the pixels that may change; the surrounding area in s_k provides the image editor with scene context. Each slot operates in one of two modes, differing in how the object candidate is determined. A *replace* slot targets a single existing object identified by a description t_k. An *add* slot provides multiple object candidates in a per-slot set $\mathcal{C}_k$. The sampler later selects content and assigns visual attributes for each slot (§3.2).

Selecting where to place a slot requires human judgment: the location must be plausible in the scene, unambiguous to reference, and difficult relative to surrounding clutter. Model-selected regions can drift toward central or salient objects, as observed in VLB's GPT-4V-guided region selection (Yang et al., 2025); such regions would make the refreshed item easier than the original small-target task. Human annotators instead mark compact locations where a new or modified key can remain natural and non-obvious.

In *replace* mode, the visual key already exists in the source image; its visual attribute will be modified while the object identity, location, and surrounding clutter remain the same, preserving the search difficulty by construction. t_k is needed because s_k may crop only a small or ambiguous region: it tells the editor which object to modify, and provides a stable referent for Q' afterward. In Figure 2, t_{A} = "blue jacket on the cyclist" identifies the jacket as the target within s_{A}.

In *add* mode, no existing object serves as the key; instead, a new object will be inserted at an empty region. The annotator curates $\mathcal{C}_k$ with objects that are plausible in the local scene, absent from the source image, and small enough to preserve the search challenge. In Figure 2, $\mathcal{C}_{\text{C}}$ lists objects such as a bucket, a cat, or a dog for the lawn patch, while $\mathcal{C}_{\text{D}}$ lists a stroller, a skateboard, or a bottle for the space beside the stairs.

Together, the annotated slots encode where and how a visual key can be renewed. This annotation is a one-time effort: once the slots are marked, the pipeline can generate fresh items indefinitely while each inherits the difficulty and naturalness constraints that the annotator established.

3.2 Edit-Slot Sampling

The annotated slots define what can change; sampling decides what actually changes in each benchmark instance. At evaluation time, a random seed σ determines a sampling record r through two steps: slot selection and content assignment.

REKEY supports two question types, and the question type determines how many slots are selected. An *attribute* question (Q_{attr}) asks about a single object's property, so it selects one slot, yield-

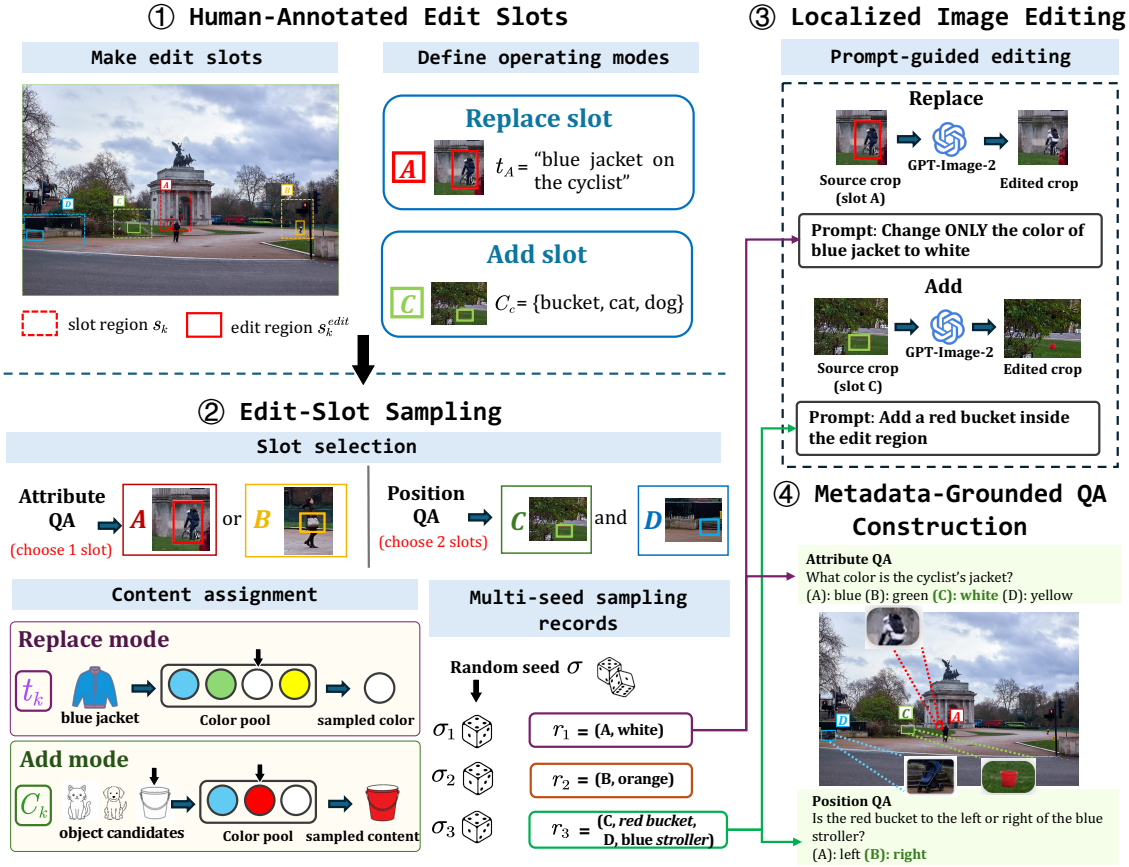

Figure 2: Overview of REKEY on one V*Bench source image with four annotated edit slots: replace slots A (blue jacket) and B (pink handbag), add slots C (lawn) and D (beside stairs). Three seeds produce three records: r_1=(A, *white*), r_2=(B, *orange*), r_3=(C, *red bucket*, D, *blue stroller*). Each r_i determines both the localized edit to I and the fresh (Q', A') pair.

ing $r = (s, c)$ where s is the slot and c the sampled content. A *position* question (Q_{pos}) asks about the spatial relation between two objects, so it selects a pair, yielding $r = (s_j, c_j,\ s_k, c_k)$; the sampler ensures the two visual keys are distinct.

For each selected slot, content is assigned according to its mode: a replace slot keeps its existing object and draws a new color from the shared pool; an add slot first draws an object from $\mathcal{C}_k$, then draws a color from the same pool. In Figure 2, three seeds produce three different records: $r_1 =$ (A, *white*) recolors the blue jacket (t_{A}) to white; $r_2 =$ (B, *orange*) recolors the pink handbag (t_{B}) to orange; $r_3 =$ (C, *red bucket*, D, *blue stroller*) selects add slots C and D with their respective objects and colors.

Slot selection and content assignment together determine the pool of fresh items derivable from one source image. Let $|\mathcal{A}|$ denote the color pool size and n_k the content choices per slot ($|\mathcal{A}|$ for replace, $|\mathcal{C}_k| \cdot |\mathcal{A}|$ for add). The number of distinct attribute items is $\sum_k n_k$ and the number of distinct position items is $\sum_{j<k} n_j \cdot n_k$, so the pool grows combinatorially (practical constraints such as object uniqueness and spatial separation reduce the pool slightly; Appendix B). Each seed thus produces a fresh (I', Q', A') with a different correct answer, all from the same one-time annotations.

3.3 Localized Image Editing

The sampling record specifies what should change; the image editor realizes it as a localized edit in the source image. We use GPT-Image-2 (OpenAI, 2026a) as the editing backbone. The pipeline processes each selected slot through three stages: *cropping*, *prompt-guided editing*, and *compositing*.

Cropping. For each selected slot s, the pipeline crops s from I and resizes it to a standard canvas. The full slot region provides scene context, but the editor is instructed to modify only the edit sub-

region s^{edit}.

Prompt-guided editing. A rule-based prompt generator constructs an instruction from $r = (s, c)$ and t_s by filling a template:

Mode	Core instruction template
Replace	"Change ONLY the color of $\{t_s\}$ to $\{c\}$"
Add	"Add a $\{c\}$ in the marked area"

t_s and c fully determine the instruction; no prompt is manually written. In Figure 2, r_1 fills the replace template as "Change ONLY the color of blue jacket to white"; r_3 fills the add template for slot C as "Add a red bucket in the marked area."

Compositing. Each edited crop is resized to the original dimensions and composited back into I. For a single-slot record such as r_1, the edited region I'_{A} replaces the A area in I, producing I'. For a two-slot record such as r_3, each slot is edited independently, yielding I'_{C} and I'_{D}, and both are composited into I to produce the final I'.

Because prompts are constructed mechanically and each edit is confined to s^{edit}, the pipeline is fully automated, reproducible, and preserves the original visual-search difficulty.

3.4 Metadata-Grounded QA Construction

With I' produced, the final step constructs Q' and A' directly from r. Because r already encodes the slot, the sampled content, and the edit-region coordinates, both the question and the answer follow deterministically. For an attribute question, the sampled content c is itself the answer, so there is no gap between what was edited and what the label says. When constructing Q', the original color is stripped from t_s to keep the question free of color cues. For a position question, the answer follows from comparing the spatial locations of the two selected slots, independent of the edited image content. The two constructions are illustrated below using records from §3.2: r_1 for Q_{attr} and r_3 for Q_{pos} (Figure 2):

Q_{attr}: Q': "What color is $\{t_s\}$?" $A' = c$
Example: $r_1 = (\text{A}, \textit{white})$, $t_{\text{A}} =$ "blue jacket on the cyclist"
Q' What color is the jacket on the cyclist? (A) blue (B) green (C) white (D) yellow
A' (C) white

Q_{pos}: Q': "Is $\{c_j\}$ to the left or right of $\{c_k\}$?" A': x_j vs. x_k
Example: $r_3 = (\text{C}, \textit{red bucket}, \text{D}, \textit{blue stroller})$, $x_{\text{C}} > x_{\text{D}}$
Q' Is the red bucket to the left or right of the blue stroller? (A) left (B) right
A' (B) right

The same r that drives the image edit also fixes the label: attribute answers come from the sampled content c, and position answers from the annotated slot locations. Together with human-curated slots (§3.1) and localized editing (§3.3), this ensures that every fresh (I', Q', A') preserves the original visual-search difficulty, carries a different correct answer, and has traceable ground truth.

Summary. REKEY converts a static VQA benchmark into a dynamic one through a single design principle: a shared sampling record r connects human-curated edit slots to both the image edit and the ground-truth label. Each seed σ draws a different r, producing a fresh (I', Q', A') whose answer is correct by construction. Because the slots are annotated once and reused across seeds, the same source images support independent evaluation runs with different visual keys, different correct answers, and consistent difficulty.

4 Experiments

The gap between original and renewed V*Bench scores (Figure 1) suggests that a substantial portion of current performance reflects memorization rather than genuine visual search. REKEY is designed to close this gap, but a dynamic benchmark must also earn trust: the difficulty should not shift, scores should remain consistent across seeds, and the human-in-the-loop design should be justified rather than replaceable. We organize our evaluation around three research questions: Does REKEY close the contamination gap (RQ1)? Does it preserve V*Bench's original difficulty (RQ2)? And is human annotation necessary, or can a VLM serve as its own annotator (RQ3)?

4.1 Setup

Setup. We annotate all 191 V*Bench (Wu and Xie, 2024) images with edit slots (§3.1), yielding 267 replace and 485 add slots, and generate three benchmark instances with different seeds (each ∼114 attribute + ∼77 position items, edited with GPT-Image-2 (OpenAI, 2026a)). We evaluate eight VLMs from eight families (Table 4 in Appendix C) and, for difficulty calibration (RQ2), LLaVA-1.5-13B (Liu et al., 2024). All models are queried under identical conditions matching the official V*Bench format: original resolution, no system prompt, temperature 0, `max_tokens` = 8192 (details in Appendices B and C).

4.2 RQ1: Does REKEY Close the Contamination Gap?

Setup. We generate three independent REKEY benchmark instances from the same 191 source images using three different random seeds σ. Each seed produces a fresh sampling record r for every item, yielding different visual keys and correct answers while keeping the source images fixed. All eight models (Table 4) are evaluated on all three seeds under the same protocol as the contamination audit.

Results. If the gap identified in Figure 1 reflects memorization, renewing the visual key should close it: REKEY scores should align with Relabel V*, the accuracy obtained when the same images are paired with fresh human-written questions. Table 1 supports this.

All eight models drop from Original V* on REKEY. The largest drops appear in models with the highest Original scores (Qwen3.6-Plus: -14.9 pp from 90.6%; Gemini: -13.0 pp from 88.0%), consistent with these models benefiting most from memorization. Models with lower Original scores show smaller drops (MiMo and Llama: -4.4 pp each), suggesting less contamination exposure.

The Relabel column provides a cross-check. Two models land almost exactly at the Relabel baseline (Qwen: +1.4 pp; GPT-5.5: +1.2 pp), indicating that REKEY recovers the same difficulty level as fresh human-written questions. The remaining models score somewhat above Relabel (mean +5.3 pp), which we attribute to Relabel using a single fixed question set whereas REKEY benefits from sampling diversity. Cross-seed standard deviations range from 0.6 to 5.4 pp, confirming that individual seeds produce consistent measurements (Figure 6 in Appendix E).

The alignment with Relabel is consistent with the drop reflecting contamination rather than a difficulty shift. A per-category decomposition (Figure 6) reinforces this interpretation: the drop concentrates in attribute questions, where memorizing a single visual property provides the most direct shortcut.

Chain-of-thought amplifies contamination. A natural concern is whether chain-of-thought (CoT) prompting could circumvent REKEY's decontamination: if step-by-step reasoning helps a model *genuinely* analyze the edited image, CoT might recover performance on REKEY just as it boosts scores on static benchmarks. To test this, we select three reasoning-capable models, turn off their internal thinking mode so that all reasoning must appear in the visible output, and compare two prompt variants: the standard direct-answer suffix (*Direct*) and a step-by-step instruction (*CoT*: "Think step by step, then answer...") (Figure 3a; setup in Appendix C, full results in Table 5).

On static benchmarks, CoT inflates scores substantially: Original V* rises by up to 6.8 pp, and, critically, Relabel V* rises by an even larger margin (up to 10.5 pp). The fact that Relabel also benefits is revealing: Relabel uses *fresh* questions on the same images, so the boost cannot come from memorizing the original question–answer pair. Instead, CoT appears to help models retrieve benchmark-associated knowledge more broadly, recognizing familiar images and leveraging any static association between image content and answer patterns.

On REKEY, this mechanism breaks down. The CoT effect turns negative for all three models (approximately -0.5 to -3.1 pp), meaning step-by-step reasoning actively hurts on renewed visual keys. The contrast is clear: CoT boosts most static evaluations yet provides no benefit once the visual key is renewed. This suggests that CoT's apparent strength on V*Bench is not visual reasoning but structured recall of benchmark-associated knowledge. By stripping away this shortcut, REKEY reveals that explicit step-by-step reasoning does not aid genuine visual search: the task is fundamentally perceptual, and CoT's contribution was an artifact of contamination.

Agentic VLMs. A parallel concern is whether iterative tool use could circumvent REKEY's decontamination: if a zoom tool helps a model systematically search the image, it might locate renewed visual keys just as effectively as static ones. To test this, we evaluate six models in a ReAct harness that provides a crop-and-zoom tool for iterative inspection (Figure 3b; full results in Table 6).

On static benchmarks, tool use raises accuracy substantially (Original +1.4–9.5 pp, Relabel +7.2–12.4 pp for four of six models). Since Relabel uses fresh questions on the same images, the tool provides genuine visual-search capability beyond memorized answers, guided by the model's familiarity with the scene.

On REKEY, this advantage does not transfer: the tool boost shrinks to -4.0 to $+3.6$ pp for five

Model	Static anchors		REKEY renewal		
	Original V*	Relabel V*	Mean ± std	Orig. Drop	vs. Relabel ≈0
Qwen3.6-Plus	90.6	74.3	75.7 ± 5.4	↓14.9	+1.4
GPT-5.5	86.9	75.9	77.1 ± 2.1	↓9.8	+1.2
Claude Opus 4.7	78.5	59.7	67.2 ± 3.9	↓11.3	+7.5
Gemini 3.1 Pro	88.0	69.6	75.0 ± 3.3	↓13.0	+5.4
Seed-2.0-Lite	91.6	74.3	79.9 ± 2.6	↓11.7	+5.6
Kimi K2.6	81.7	63.4	72.6 ± 0.8	↓9.1	+9.2
MiMo v2.5	78.0	66.5	73.6 ± 4.7	↓4.4	+7.1
Llama 4 Maverick	61.3	51.8	56.9 ± 0.6	↓4.4	+5.1
Average	*82.1*	*66.9*	*72.2*	↓9.9	+5.3

Table 1: REKEY evaluation (RQ1). **Relabel V***: accuracy when the same images are paired with fresh human-written questions targeting different visual keys (Appendix A). Red ↓: Original-to-REKEY drop. Teal: proximity to Relabel (values near zero = aligned). **REKEY** reports 3-seed mean ± standard deviation. Accuracy in %; differences in percentage points.

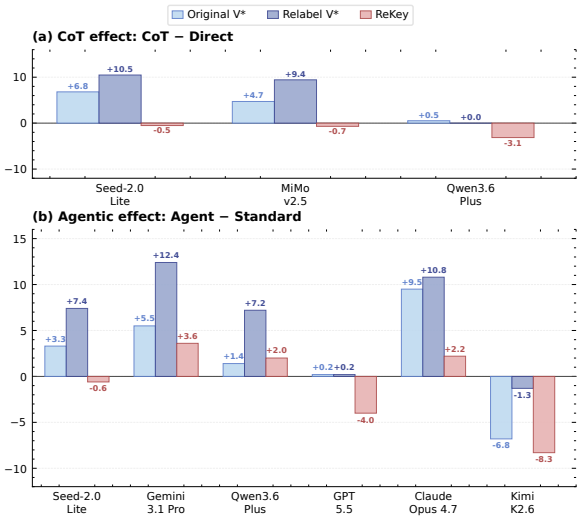

Figure 3: Two capability boosts that disappear under REKEY (RQ1). **(a)** CoT effect (CoT − Direct): three reasoning models with internal thinking disabled. **(b)** Agentic effect (Agent − Standard): six models in a ReAct harness with iterative zoom. Both capabilities boost most static evaluations (Original and Relabel, blue bars) but the effect shrinks or turns negative on REKEY (red). Effects are in percentage points.

of six models. Both Relabel and REKEY require locating a different visual key, but Relabel preserves the original image while REKEY locally edits it. On Relabel, the model's scene understanding guides the tool to relevant regions; on REKEY, the visual key itself is novel, so even zooming into the correct area does not yield a recognizable target. This shows that REKEY is robust against tool-augmented evaluation.

4.3 RQ2: Does REKEY Preserve the Original Difficulty?

The accuracy drop from Original V* to REKEY could reflect a difficulty shift rather than contamination removal. We examine this from three angles.

Edit-region scale. By design, each edit modifies only the slot region s_k^{edit}, whose median area is approximately 3,500 px^2 in a ∼2250×1500 image, roughly 0.1% of the total pixels. As described in §3.1, replace slots target the same objects that V*Bench's original questions ask about, so their edit regions directly reflect V*Bench's small-target scale. Human-annotated add regions are of similar size (∼3,300 px^2 vs. ∼3,800 px^2 for replace; Table 3), confirming that annotators select insertion locations that match the original difficulty standard. The vast majority of the image remains untouched, limiting the scope for difficulty shift.

Model calibration. To verify empirically, we evaluate LLaVA-1.5-13B, the strongest open-source baseline in the original V*Bench paper (Wu and Xie, 2024). This model anchors V*Bench's published difficulty and carries lower contamination risk, so a large drop would signal a task change rather than decontamination. Table 2 shows that LLaVA-1.5-13B drops by only 6.1 pp overall, smaller than most frontier models in RQ1 (9.1–14.9 pp for six of eight models). Position accuracy is nearly unchanged (−0.4 pp), which is particularly important because spatial relations are the aspect most sensitive to editing artifacts. The attr/pos asymmetry (Figure 6 in Appendix E) reinforces this: attribute questions drop more because memorizing a single color provides a direct shortcut, while position questions barely change. Together,

Split	Original V*	REKEY	Drop
Overall	48.68	42.6 ± 2.4	↓6.1
Attr	43.47	33.2 ± 3.9	↓10.3
Pos	56.57	56.2 ± 3.3	↓0.4

Table 2: Difficulty calibration on LLaVA-1.5-13B (RQ2). **Original V*** uses the published result from Wu and Xie (2024); **REKEY** reports the 3-seed mean ± standard deviation. Accuracy in %; drops in percentage points.

the small edit-region footprint, the modest LLaVA calibration drop, and the stability of position accuracy indicate that the RQ1 drops are contamination-driven rather than difficulty-driven.

4.4 RQ3: Is Human Annotation Necessary?

Setup. A natural question is whether the human annotation stage can be replaced by a VLM, reducing cost and increasing scale. Vision-Language Bootstrapping (Yang et al., 2025), for instance, uses GPT-4V to select regions for perturbation. We prompt GPT-5.5 with each V*Bench image and the same annotation instructions given to human annotators. However, only 33.6% of the VLM's replace annotations are correctly localized (Table 3): GPT-5.5 frequently places bounding boxes at incorrect locations, failing to locate the small, peripheral targets that V*Bench requires. We therefore construct the VLM-annotated benchmark from add slots only (189 items) and evaluate all eight models on both benchmarks.

Results. All eight models score 8.4–18.5 pp higher on the VLM-annotated benchmark (Figure 4). Table 3 reveals the cause: human-annotated add regions (∼3,300 px^2) match the small-target scale of V*Bench's replace regions (∼3,800 px^2), but VLM-annotated add regions are 3.2× larger (∼10,500 px^2), because GPT-5.5 favors prominent, spacious areas where inserted objects are easy to spot. This saliency bias systematically inflates accuracy, undermining the difficulty preservation that human curation provides.

5 Conclusion

V*Bench scores are substantially higher than warranted, consistent with contamination. REKEY addresses this by treating the visual key as the renewable unit: human annotators mark edit slots once, and a rule-based pipeline generates fresh instances whose ground truth is correct by construction. Our experiments show that REKEY closes the contamination gap while preserving difficulty, and that neither chain-of-thought prompting nor agentic tool use restores the static-benchmark gains in our experiments. A VLM self-annotation ablation further supports the need for human curation, as automated annotation introduces systematic saliency bias. More broadly, one-time human curation combined with automated renewal shows that curated visual benchmarks need not be disposable: with the right design, the same source images can support fresh, reliable evaluation as models and training data continue to evolve.

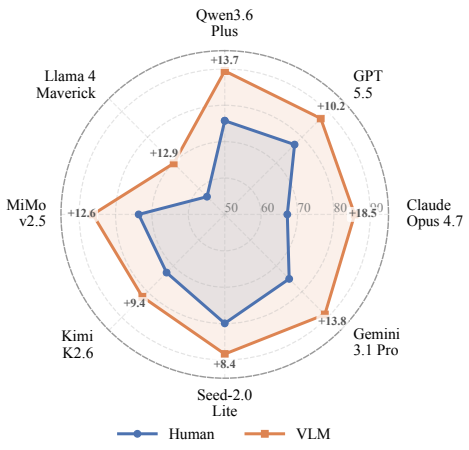

Figure 4: Human vs. VLM annotation (RQ3). **Human** (blue): mean accuracy on the human-annotated REKEY benchmark (3 seeds). **VLM** (orange): accuracy on the GPT-5.5-annotated benchmark using add slots only. All eight models score +8.4–18.5 pp higher on the VLM-annotated version. Accuracy in %.

	Replace	Add
Human size (px^2)	3,843	3,256
VLM size (px^2)	3,531	10,504
Size ratio	0.92×	**3.2**×
Usability	33.6%	∼94%

Table 3: VLM annotation quality by slot type (RQ3). Human replace regions serve as a difficulty baseline: they target the original V*Bench visual keys and reflect the benchmark's small-target scale. **Size ratio**: VLM median / Human median. **Usability**: fraction of VLM-annotated slots whose bounding box correctly covers the intended target (manual inspection). Replace annotations are mostly unusable (33.6%); add annotations are largely usable (∼94%) but 3.2× larger than the human baseline.

Limitations

Single generator. All edits use GPT-Image-2. Instruction-based editing does not guarantee pixel preservation outside the edit region: visible seams at crop boundaries and occasional object size mismatches are observed. Switching to a different editing backend may introduce new artifact patterns that require re-validation.

Attribute coverage. The current implementation varies only color. The method framework supports other visual attributes (material, texture), but these require corresponding attribute pools and have not been validated.

Cost. Annotating edit slots for all 191 V*Bench images required approximately 16 person-hours (two annotators, one day each). This is a one-time effort: once annotated, the same slots support many independent benchmark instances across different seeds. Generating one instance costs approximately $21 in GPT-Image-2 API calls, and evaluating eight models on one seed costs approximately $14 in VLM API calls.

Difficulty control. The pipeline's design implicitly preserves visual-search difficulty: replace slots keep the target at its original location, and add slots use human-curated regions that match V*Bench's small-target scale (RQ1 and RQ2 confirm this at the aggregate level). However, the pipeline treats all items uniformly and provides no mechanism for explicitly controlling or grading item difficulty. It cannot generate a harder or easier variant on demand, nor distinguish easy items from hard ones before evaluation. This limits applications that require fine-grained difficulty profiling, such as adaptive testing or curriculum-based evaluation. Introducing difficulty-aware sampling, for instance by conditioning on slot location, object size, or scene clutter, is a natural direction for future work.

V*Bench scope. We validate REKEY exclusively on V*Bench (191 images). Generalization to other visual benchmarks with different image characteristics or question formats has not been tested.

References

Anthropic. 2026. Introducing Claude Opus 4.7.

Han Bao, Yue Huang, Yanbo Wang, Jiayi Ye, Xiangqi Wang, Xiuying Chen, Mohamed Elhoseiny, and Xiangliang Zhang. 2024. Autobench-v: Can large vision-language models benchmark themselves? *CoRR*, abs/2410.21259.

Ellis Brown, Jihan Yang, Shusheng Yang, Rob Fergus, and Saining Xie. 2025. Benchmark designers should "train on the test set" to expose exploitable non-visual shortcuts. *CoRR*, abs/2511.04655.

ByteDance Seed Team. 2026. Seed 2.0 official launch.

Simin Chen, Yiming Chen, Zexin Li, Yifan Jiang, Zhongwei Wan, Yixin He, Dezhi Ran, Tianle Gu, Haizhou Li, Tao Xie, and Baishakhi Ray. 2025. Recent advances in large langauge model benchmarks against data contamination: From static to dynamic evaluation. *Preprint*, arXiv:2502.17521.

Paul Gavrikov, Wei Lin, Muhammad Jehanzeb Mirza, Soumya Jahagirdar, Muhammad Huzaifa, Sivan Doveh, Serena Yeung-Levy, James R. Glass, and Hilde Kuehne. 2025. Visualoverload: Probing visual understanding of vlms in really dense scenes. *CoRR*, abs/2509.25339.

Gemini Team. 2026. Gemini 3.1 Pro: A smarter model for your most complex tasks.

Drew A. Hudson and Christopher D. Manning. 2019. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 6700–6709. Computer Vision Foundation / IEEE.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross B. Girshick. 2017. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 1988–1997. IEEE Computer Society.

Sua Lee, Sanghee Park, and Jinbae Im. 2026. Mm-judgebias: A benchmark for evaluating compositional biases in mllm-as-a-judge. *CoRR*, abs/2604.18164.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024. Improved baselines with visual instruction tuning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 26286–26296. IEEE.

Meta AI. 2025. The Llama 4 Herd: The beginning of a new era of natively multimodal AI innovation.

Moonshot AI. 2026. Kimi K2.6: Advancing open-source coding.

OpenAI. 2026a. Introducing ChatGPT images 2.0.

OpenAI. 2026b. Introducing GPT-5.5.

Qwen Team. 2026. Qwen3.6-Plus: Towards real world agents.

Nimrod Shabtay, Felipe Maia Polo, Sivan Doveh, Wei Lin, Muhammad Jehanzeb Mirza, Leshem Choshen, Mikhail Yurochkin, Yuekai Sun, Assaf Arbelle, Leonid Karlinsky, and Raja Giryes. 2025. Livexiv - A multi-modal live benchmark based on arxiv papers content. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Dingjie Song, Sicheng Lai, Mingxuan Wang, Shunian Chen, Lichao Sun, and Benyou Wang. 2025. Both text and images leaked! A systematic analysis of data contamination in multimodal LLM. In *Findings of the Association for Computational Linguistics: EMNLP 2025, Suzhou, China, November 4-9, 2025*, pages 10527–10542. Association for Computational Linguistics.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Ben Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Sreemanti Dey, Shubh-Agrawal, Sandeep Singh Sandha, Siddartha Naidu, Chinmay Hegde, Yann LeCun, Tom Goldstein, Willie Neiswanger, and Micah Goldblum. 2025. Livebench: A challenging, contamination-limited llm benchmark. *Preprint*, arXiv:2406.19314.

Penghao Wu and Saining Xie. 2024. V*: Guided visual search as a core mechanism in multimodal llms. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 13084–13094. IEEE.

Xiaomi MiMo Team. 2026. Xiaomi MiMo-V2.5.

Yue Yang, Shuibo Zhang, Kaipeng Zhang, Yi Bin, Yu Wang, Ping Luo, and Wenqi Shao. 2025. Dynamic multimodal evaluation with flexible complexity by vision-language bootstrapping. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Junzhe Zhang, Huixuan Zhang, and Xiaojun Wan. 2025a. KBE-DME: dynamic multimodal evaluation via knowledge enhanced benchmark evolution. *CoRR*, abs/2510.21182.

Yifan Zhang, Huanyu Zhang, Haochen Tian, Chaoyou Fu, Shuangqing Zhang, Junfei Wu, Feng Li, Kun Wang, Qingsong Wen, Zhang Zhang, Liang Wang, and Rong Jin. 2025b. Mme-realworld: Could your multimodal LLM challenge high-resolution real-world scenarios that are difficult for humans? In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Yusen Zhang, Wenliang Zheng, Aashrith Madasu, Peng Shi, Ryo Kamoi, Hao Zhou, Zhuoyang Zou, Shu Zhao, Sarkar Snigdha Sarathi Das, Vipul Gupta, Xiaoxin Lu, Nan Zhang, Ranran Haoran Zhang, Avitej Iyer, Renze Lou, Wenpeng Yin, and Rui Zhang. 2025c. Hrscene: How far are vlms from effective high-resolution image understanding? In *IEEE/CVF International Conference on Computer Vision, ICCV 2025, Honolulu, HI, USA, October 19-25, 2025*, pages 22922–22933. IEEE.

Chengke Zou, Xingang Guo, Rui Yang, Junyu Zhang, Bin Hu, and Huan Zhang. 2025. Dynamath: A dynamic visual benchmark for evaluating mathematical reasoning robustness of vision language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

A V*Bench Contamination Audit

To measure how much current V*Bench performance reflects memorization, we construct *Relabel V**: the same 191 images paired with fresh human-written questions that target different visual keys while preserving the original difficulty regime (small, peripheral targets in cluttered scenes).

Figure 5 walks through a concrete example. On vstar_0101, all eight models correctly answer the original question ("What color is the life buoy?" → red/white, 8/8). The life buoy is a small, peripheral detail in a cluttered lakeside scene, exactly the kind of target V*Bench is designed to test. Yet when a human annotator writes a fresh question about a different visual key in the same image ("What color is the direction sign?" → yellow), every model fails (0/8). The image has not changed; only the question target has. This sharp drop on a single item suggests that models recalled a memorized answer rather than locating the visual key through genuine visual search. The bottom half of the figure shows the REKEY counterpart: a purple dog is inserted via localized editing, and the question and answer are derived from the sampling record. All models fail again (0/8), but unlike Relabel, REKEY produces this effect automatically from a single annotation.

The pattern illustrated by vstar_0101 is not isolated. Figure 1 aggregates the results across all 191 items and eight models: every model's overall accuracy drops when the same images are paired with fresh questions targeting different visual keys. Seven of eight drops are statistically significant ($p < 0.01$, McNemar's exact test); only Llama 4 Maverick (-9.5 pp, $p = 0.09$) falls short of significance, consistent with its lower baseline and lower contamination exposure. The largest drops appear in models with the highest Original scores (Claude Opus 4.7: -18.8 pp from 78.5%; Gemini 3.1 Pro (-18.4 pp) and Kimi K2.6

(−18.3 pp)), consistent with these models benefiting most from memorization. A detailed breakdown by question type is provided in Appendix E.

B Annotation and Construction Details

Slot design rationale. The two slot modes serve complementary roles in breaking memorization while preserving the visual-search task.

Replace slots target the object that the original V*Bench question asks about. Most V*Bench questions reference a unique object in the scene (e.g. "What color is the life buoy?"), so modifying that specific object directly invalidates the memorized answer while largely preserving the search task: the model must still locate the same small, peripheral target in the same cluttered scene. Because the object's identity and location are unchanged, a performance drop is consistent with loss of the memorized answer rather than a change in difficulty.

Add slots introduce a new object that is absent from the source image, reducing the chance of a fixed scene-object association in training data. Annotators curate the candidate set $\mathcal{C}_k$ to contain only objects that are plausible in the local context, absent from the source image, and small enough to preserve the visual-search difficulty (§3.1). The absence constraint is enforced during sampling: each candidate is checked against all replace targets in the scene (both the full description and its last word) and excluded if a match is found. For position questions, an additional spatial-separation constraint requires the two selected slot centers to be at least 20 px apart, ensuring that the directional answer (left/right or above/below) is unambiguous. Together, these constraints ensure that answering a REKEY question requires genuine visual search rather than recalling typical scene compositions.

Position-question format. Position questions ask about the spatial relation between two objects, so both objects must be unambiguously identifiable in the question text. The format follows from the slot design above.

A replace object retains its identity and location; only its color changes. Because the scene may contain other instances of the same category, the question includes the *new* color to let the model distinguish the edited object from its surroundings (e.g. "the **orange** phone" after recoloring from black). An add object, by contrast, is the only instance of its category in the scene (guaranteed by the absence constraint), so the category name alone is sufficient (e.g. "the cat"). Including the color for add objects would be redundant and could introduce an unintended cue that makes the object easier to locate.

Model	Type	Reasoning
Qwen3.6-Plus (Qwen Team, 2026)	✗	Always-on CoT
GPT-5.5 (OpenAI, 2026b)	✗	Medium
Claude Opus 4.7 (Anthropic, 2026)	✗	Adaptive
Gemini 3.1 Pro (Gemini Team, 2026)	✗	Medium
Seed-2.0-Lite (ByteDance Seed Team, 2026)	✗	None
Kimi K2.6 (Moonshot AI, 2026)	✗	Default
MiMo v2.5 (Xiaomi MiMo Team, 2026)	✓	Default
Llama 4 Maverick (Meta AI, 2025)	✓	None

Table 4: Evaluated models spanning eight families. **Type**: ✗ closed-weight, ✓ open-weight. **Reasoning**: provider-default strategy (not overridden).

C Evaluation Protocol Details

All models are queried through the OpenRouter API under identical conditions: original image resolution, no system prompt, temperature 0, and `max_tokens` = 8192. Each provider exposes a default reasoning strategy that we do not override (Table 4); this mirrors a realistic deployment setting where the user accepts the provider's default. Source images are re-encoded as JPEG (quality 95) for API compatibility; no resizing is applied.

Prompt format. Each item is presented as a single-turn message containing the image and a text prompt. For the standard (direct) evaluation, the prompt follows the official V*Bench format:

```
{question} (A) {a} (B) {b} . . . Answer with the
option's letter from the given choices directly.
```

For the CoT experiments (§4.2), the suffix is replaced with:

```
Think step by step, then answer with the option's
letter from the given choices.
```

Responses are parsed by extracting the *last* option letter (A/B/C/D) in the output, which avoids matching letters that appear in intermediate reasoning text.

CoT experimental setup. The chain-of-thought analysis in §4.2 uses three reasoning-capable models with their internal thinking mode disabled (OpenRouter `reasoning.effort` = `none`), so that all reasoning must appear in the visible output. Two prompt variants are compared: the standard direct-answer suffix above (*Direct*) and a step-by-step instruction (*CoT*) shown above. This setup isolates

the effect of explicit chain-of-thought reasoning from provider-internal thinking tokens.

D CoT and Agentic Detailed Results

Table 5 reports the CoT experiment from §4.2: three reasoning models with internal thinking disabled, evaluated under Direct and CoT prompting across three benchmark conditions. Table 6 reports the agentic experiment: six models in a ReAct harness with iterative zoom, compared with their standard (no-tool) counterparts.

Model	Mode	Orig.	Relab.	REKEY
Seed-2.0-Lite	Direct	82.7	62.8	79.8 ± 3.0
	CoT	89.5	73.3	79.2 ± 4.3
MiMo v2.5	Direct	74.3	55.0	68.2 ± 4.7
	CoT	79.1	64.4	67.5 ± 3.1
Qwen3.6-Plus	Direct	85.9	66.5	78.9 ± 4.1
	CoT	86.4	66.5	75.7 ± 4.5

Table 5: CoT vs. Direct accuracy with internal thinking disabled (§4.2). **Orig.**: Original V*. **Relab.**: Relabel V*. **REKEY**: 3-seed mean ± standard deviation. Accuracy in %.

	Original		Relabel		REKEY	
Model	Ag.	Base	Ag.	Base	Ag.	Base
Seed-2.0-Lite	94.9	91.6	81.7	74.3	79.3±3.4	79.9±2.6
Gemini 3.1 Pro	93.5	88.0	82.0	69.6	78.6±2.0	75.0±3.3
Qwen3.6-Plus	92.0	90.6	81.5	74.3	77.7±4.0	75.7±5.4
GPT-5.5	87.1	86.9	76.1	75.9	73.1±2.0	77.1±2.1
Claude Opus 4.7	88.0	78.5	70.5	59.7	69.4±2.6	67.2±3.9
Kimi K2.6	74.9	81.7	62.1	63.4	64.3±2.0	72.6±0.8

Table 6: Agentic vs. standard VLMs (§4.2). **Ag.**: ReAct harness with iterative zoom tool. **Base**: standard single-pass evaluation (Table 1). REKEY columns report 3-seed mean ± standard deviation; Original and Relabel are from single evaluations. Accuracy in %.

E Additional Analysis

Attribute vs. position asymmetry. Figure 6 decomposes REKEY accuracy into attribute (Q_{attr}) and position (Q_{pos}) questions. Compared to Original V* (Table 1), attribute accuracy drops 14.8 pp on average while position drops only 2.6 pp, a 5.7× asymmetry discussed in §4.2. Three factors explain why attribute questions are more susceptible.

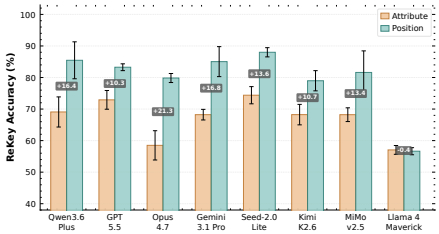

Figure 6: REKEY accuracy by question type (3-seed mean ± σ). Gray tags show the position–attribute gap in percentage points. Llama 4 Maverick is the only model without a positive gap, consistent with its low contamination exposure. Accuracy in %.

First, attribute answers are compact, single-token facts (e.g. "blue") that are easy to memorize from a fixed benchmark. REKEY replaces each memorized value with a freshly sampled color, so the cached answer has only a $1/|\mathcal{A}|$ chance (≈9%) of remaining correct. Position answers, by contrast, depend on the spatial relation between two objects—a relational judgment that is harder to encode as a training-data shortcut.

Second, the two question types differ in baseline chance. Attribute questions offer four options (25% chance), leaving a wide margin for contamination-driven inflation. Position questions offer two (50%), compressing the range over which memorization can inflate scores.

Third, REKEY's renewal mechanism preserves spatial layout by construction. Replace slots keep the target object at its original location; add slots insert objects at annotated coordinates. The relative position of any slot pair is therefore fixed across seeds, so position accuracy reflects genuine spatial reasoning rather than memorized layout. The small mean position drop (−2.6 pp) provides independent support for the difficulty-preservation argument (RQ2): spatial relations are the aspect most sensitive to editing artifacts, and their stability indicates that the localized editing process preserves the scene's spatial structure.

Llama 4 Maverick is the sole exception to the pattern: its position–attribute gap is −0.4 pp, compared to +10.3 to +21.3 for all other models (Figure 6). This is consistent with Llama having the smallest and only non-significant contamination drop (−9.5 pp, $p = 0.09$; Figure 1): without a strong memorization advantage on attribute questions, its attr and pos accuracy are roughly equal under REKEY.

F Qualitative Examples

We examine individual REKEY items to illustrate how visual-key renewal breaks memorization while preserving visual-search difficulty. The two slot modes provide complementary mechanisms: inserting a novel object (add mode) and recoloring an existing target (replace mode).

Add mode: novel object in a cluttered scene. Figure 7 shows vstar_0093. The original question asks about a helmet's color in an urban street scene, and all eight models answer correctly (8/8). REKEY inserts a small brown stroller in the corner of a parking area; the generated question asks its color. No model answers correctly (0/8). The stroller is small, placed at the periphery between a parked car and a crosswalk, and surrounded by visual clutter (signage, building facades, street furniture). Because the object is absent from the source image, no model has a prior association between this scene and the stroller, so answering requires genuine visual search.

Replace mode: color renewal on the original target. Figure 8 shows vstar_0046. The original question asks about a man's cap color in a resort scene; seven of eight models answer correctly (88%). REKEY recolors the cap from white to purple. The object's identity and location are unchanged, so the visual-search task is largely preserved; yet only three models identify the new color (38%). This is consistent with the original high accuracy reflecting a memorized answer rather than genuine localization of the cap.

Together, these two cases illustrate why both slot modes contribute to decontamination: add mode eliminates scene-object associations, and replace mode invalidates memorized attributes, while both preserve the small-target, cluttered-scene regime that defines V*Bench's difficulty.

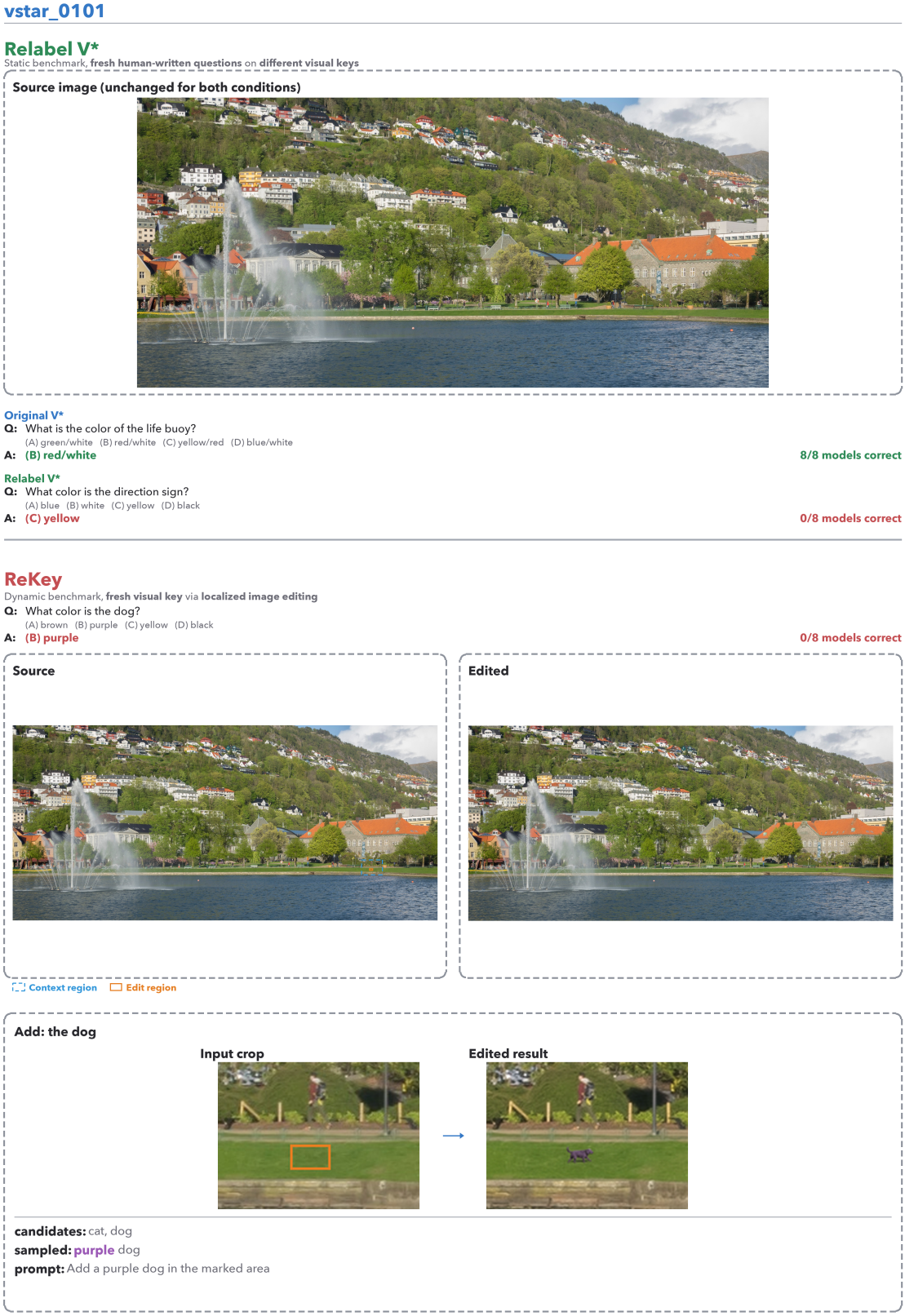

Figure 5: Two ways to break memorization on the same V*Bench image (vstar_0101). **Relabel V*** (top) keeps the source image unchanged and pairs it with a fresh human-written question about a different visual key (direction sign instead of life buoy). **ReKey** (bottom) inserts a new visual key (purple dog) via localized editing and derives both the question and the answer from the sampling record r. All eight models answer the original question correctly (8/8) but fail on both renewed conditions (0/8).

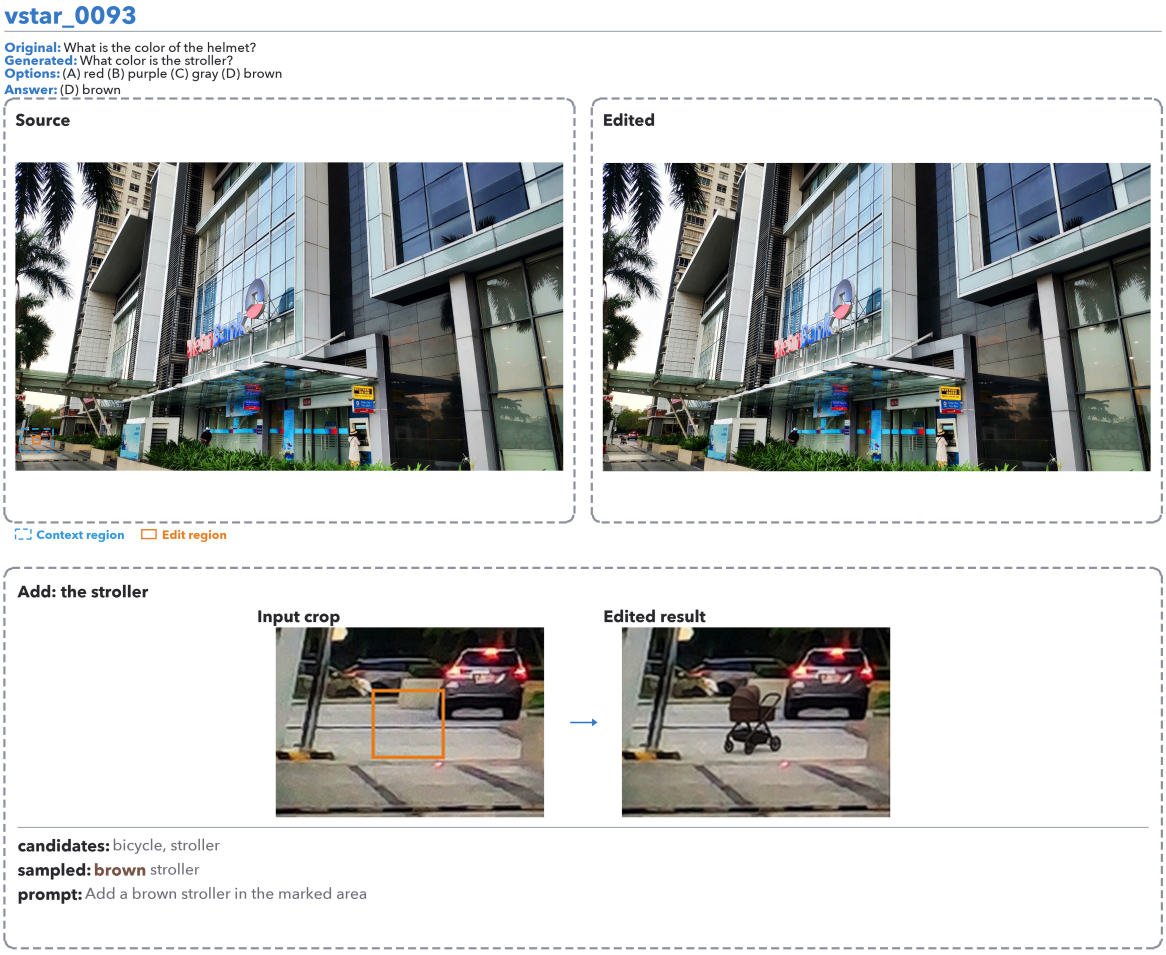

Figure 7: Success: add mode (vstar_0093, bench2). A small brown stroller is inserted in a parking-area corner. Original accuracy: 8/8; REKEY accuracy: 0/8.

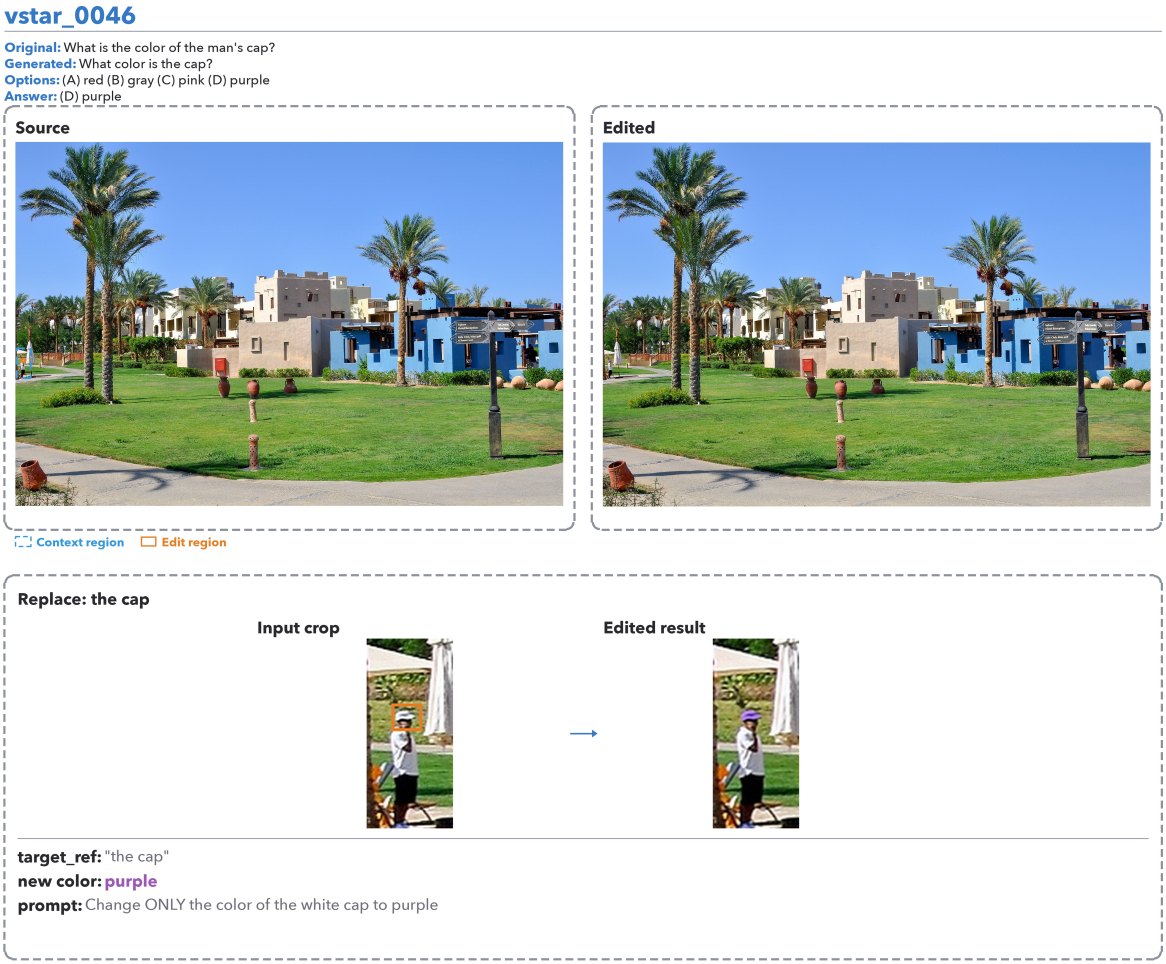

Figure 8: Success: replace mode (vstar_0046). A white cap in a resort scene is recolored to purple. Original accuracy: 88%; REKEY accuracy: 38%.