SOLVEEDIT: BENCHMARKING VISUAL PROBLEM SOLVING IN GENERATIVE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Machine intelligence is often evaluated through abstract reasoning problems, yet many real-world problems are visual, such as arranging objects, repairing layouts, or tracing routes. Solving these problems requires understanding a scene, inferring what must change to achieve a goal, and realizing that change without disturbing unrelated content. However, existing benchmarks mainly evaluate perception, generation, or explicitly specified transformations, leaving goal-driven visual problem solving underexplored. To bridge this gap, we introduce SOLVEEDIT, a benchmark for visual problem solving through scene transformation. Given an image and a goal, a model must infer a valid transformation from the request, the scene, or a visually expressed rule, then execute it while preserving unrelated content. SOLVEEDIT contains 2,728 cases across 10 domains and 54 subdomains. Atomic transition contracts specify required and protected conditions, enabling SOLVESCORE to measure completion and unintended changes without a single reference output. The strongest evaluated model achieves only 57.0% SOLVESCORE. We further introduce SOLVEEDIT-PLAN, a two-stage visual planner that instantiates the transition before generation. Under matched single-generation evaluation, it improves SOLVESCORE by 9.1 points on average across three tested generators, including a gain from 57.0% to 71.6% for GPT-Image-2, without modifying the editor.

1 INTRODUCTION

Machine intelligence is usually tested with abstract reasoning problems, but much real-world problem solving starts in front of a visual scene, as in Figure 1. Given the scene and a goal, the system has to understand the current state, work out what should change, produce a valid final state, and leave unrelated content alone. Visual interpretation, transition determination, and execution all enter the process. We study this capability as *visual problem solving through scene transformation*: the model answers by changing the scene, not by returning text.

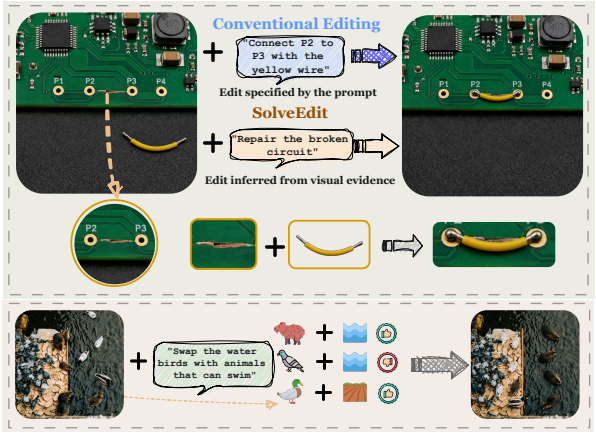


Figure 1: From executing a specified edit to solving a visual problem. In SOLVEEDIT, the request states a goal, while the source image and any rule depicted in it may be needed to determine and realize a valid solution.

Current evaluations tend to isolate perception, abstract reasoning, generation, or the execution of explicitly specified transformations. Instruction-based editing benchmarks, for example, check whether a model can carry out a stated change while preserving the rest of an image (Brooks et al., 2023; Zhang et al., 2023; Wang et al., 2023; Ku et al., 2024b; Ma et al., 2024). Newer image-to-image benchmarks add temporal, causal, spatial, logical, factual, procedural, and planning tasks (Zhao et al., 2026b; Wu et al., 2026b; Han et al., 2025). What they do not isolate is where the required transformation comes from: the request, the current scene, or a rule expressed in the image. Executing a specified transformation and inferring an unresolved one can therefore be scored together.

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

Reference-based scoring adds a second problem: valid solutions that differ from the single authored output may be penalized. These gaps lead to our central question: *Can a generative model determine a valid transformation from a source image and a goal, and then realize it in the same scene?*

We answer with SOLVEEDIT, which puts transition determination inside the visual problem instead of supplying it through the prompt. The distinction matters. A model can execute a familiar edit pattern without working out what the current scene requires, just as a language model that has memorized facts or solution templates need not generalize to a new configuration. We therefore control the source of the information that fixes a valid transition. The 2,728 cases, spread over 10 domains and 54 subdomains, fall into three regimes. In **Instruction-Specified (IS)** cases the prompt specifies the change. In **State-Dependent (SD)** cases the model resolves a missing variable from observable scene state or basic spatial and structural relations. In **Rule-Dependent (RD)** cases it interprets a rule expressed in the image, sometimes with knowledge beyond the immediate scene. Removing a grounded marker is IS; placing a block in the only empty slot is SD; sorting pieces by a depicted legend is RD. Instruction following, scene understanding, and rule inference are thus separated within one generative task. The axis is orthogonal to content domain and task type. The same spatial operation can land in IS, SD, or RD depending on what fixes its valid outcome.

More than one solution can be valid, so each case carries an atomic *transition contract*. Required conditions state what the output must accomplish; protected conditions state what must remain intact. SOLVESCORE measures required completion, applies a bounded deduction for unintended changes, and reports six diagnostic scores across the required and protected semantic, relational, and visual-quality conditions. The same contracts score completed states from image-to-image models and final frames from image-to-video models.

We evaluate nine image-to-image models and two image-to-video models on the same contracts. The strongest of them reaches only 57.0% SOLVESCORE, which leaves substantial room for improvement. Its diagnostic scores show larger deficits in required semantic and relational conditions than in visual-quality ones; solution recovery appears to be a large part of the failure. That diagnosis suggests a direct test: determine the transition before generation. We run it with SOLVEEDIT-PLAN, a two-stage agentic visual planner that instantiates scene-dependent transition variables and writes an evidence-grounded editing instruction, with no training or changes to the editor. Under matched single-generation evaluation, it takes GPT-Image-2 from 57.0% to 71.6% SOLVESCORE, transfers to the open-source editor and the video generator, and beats a call-matched GENERIC VISION REWRITE control by 6.3 points on GPT-Image-2, with higher required completion and less collateral damage.

Our contributions are threefold:

- ❶ **A problem formulation and benchmark.** We formulate visual problem solving through scene transformation and introduce SOLVEEDIT, a benchmark of 2,728 cases across 10 application domains and 54 subdomains, organized by the information source that determines the required transition.
- ❷ **A contract-based evaluation framework.** We define atomic transition contracts and SOLVESCORE to accommodate multiple valid visual outcomes, separate required completion from collateral changes, and give six diagnostic views of model behavior.
- ❸ **A diagnosis and targeted intervention.** We analyze current image and video generators across the IS, SD, and RD regimes, and introduce SOLVEEDIT-PLAN to test whether explicitly instantiating the transition addresses the observed deficits in required semantic and relational conditions.

2 RELATED WORK

Visual Reasoning and Image Editing. Visual reasoning has progressed from language or discrete-answer benchmarks to tasks that infer transformations or realize answers as images and videos (Goyal et al., 2017; Johnson et al., 2017; Hudson & Manning, 2019; Suhr et al., 2019; Marino et al., 2019; Yue et al., 2024; Chollet, 2019; Chen et al., 2026; Li et al., 2026; Chen et al., 2025). Image-editing methods generally receive the desired change explicitly, including both instruction-based (Brooks et al., 2023; Zhang et al., 2023; Sheynin et al., 2024; Hui et al., 2024; Zhao et al., 2024; Yu et al., 2025) and training-free approaches (Meng et al., 2021; Hertz et al., 2022; Mokady et al., 2023; Kawar et al., 2023; Cao et al., 2023; Parmar et al., 2023; Tumanyan et al., 2023; Brack et al., 2024).

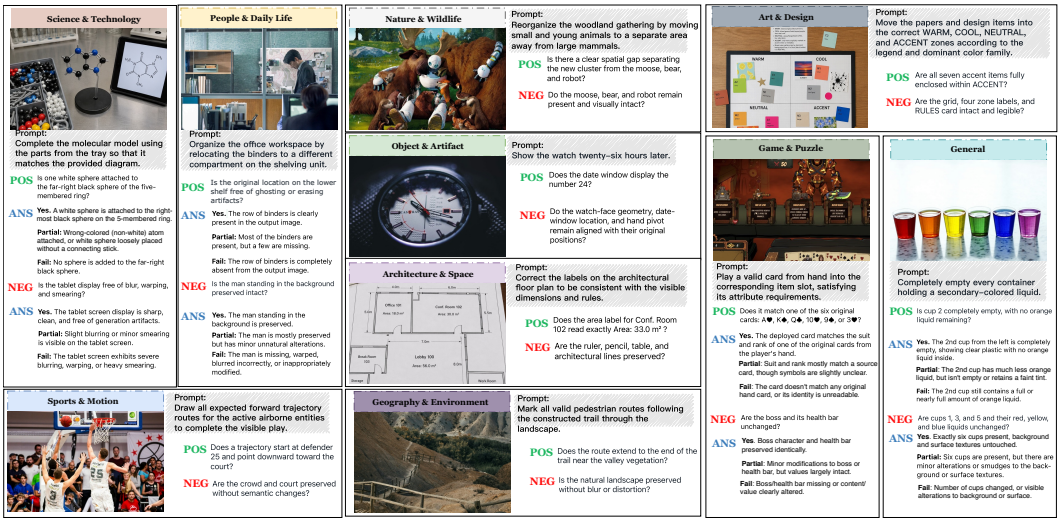


Figure 2: **Application breadth of SOLVEEDIT.** One representative case from each of ten domains. Each panel shows the source image, task instruction (Q), and evaluator questions (POS/NEG) with reference answers (ANS). The examples span semantic grouping, numerical correction, structural assembly, route tracing, sequence completion, trajectory prediction, and constrained action.

More recent benchmarks extend editing evaluation to grounded correctness, preservation, knowledge, reasoning, and planning (Wang et al., 2023; Ku et al., 2024b; Ma et al., 2024; Qian et al., 2025; Zhao et al., 2026b; Wu et al., 2026b; Han et al., 2025; Zhang et al., 2026; Zheng et al., 2026). Our focus is complementary: SOLVEEDIT makes the source of transition-defining information an explicit evaluation axis and uses required and protected contracts to allow multiple valid renderings while measuring collateral changes. It also spans heterogeneous visual sources, including photographs, animation, games, cinematic frames, and purpose-built scenes, testing whether transition recovery transfers across visual styles. Figure 2 illustrates the resulting breadth of visual problems.

Evaluation of Generated Visual Content. Distributional and embedding metrics measure fidelity or cross-modal alignment (Heusel et al., 2017; Radford et al., 2021; Hessel et al., 2021); learned preference and object-focused evaluators provide more targeted text-to-image signals (Xu et al., 2023; Wu et al., 2023b; Ghosh et al., 2023); and question-based evaluators decompose outputs into interpretable judgments (Hu et al., 2023; Ku et al., 2024a; Cho et al., 2024; Wu et al., 2024). These metrics primarily score an image or its alignment with text. In contrast, SOLVEEDIT evaluates a transition between an input and an output. Its atomic contracts separate required completion from independently protected content, allow multiple valid final states, and report collateral damage separately from task completion.

Multimodal Reasoning for Visual Generation. Reasoning-action agents, tool-using models, and multimodal language models provide foundations for visually grounded planning (Yao et al., 2022; Schick et al., 2023; Driess et al., 2023; Wang et al., 2024a; Chen et al., 2023). Visual agents compose tools or foundation models for multi-step generation and editing (Yang et al., 2023; Wu et al., 2023a; Wang et al., 2024b), while editing systems increasingly use multimodal understanding, intent interpretation, or intermediate plans (Fu et al., 2024; Huang et al., 2024a; Hu et al., 2024; Ji et al., 2025; Ci et al., 2025). SolveEdit-Plan addresses a narrower question: whether a task-specific transition can be recovered before one call to an unchanged generator. It uses targeted observation, candidate comparison, and explicit preservation obligations, with a matched planning backend, call budget, and final generator. For video, we use terminal frames only to test transfer of the same transition contracts across modalities; temporal generation is outside our scope (Kondratyuk et al., 2023; Bar-Tal et al., 2024; Yang et al., 2025; Kong et al., 2024; Huang et al., 2024b; Fu et al., 2025).

3 SOLVEEDIT: FORMULATION, BENCHMARK, AND EVALUATION

3.1 TASK FORMULATION AND DEPENDENCY REGIMES

Let I be an input image and x a natural-language request. Writing the editor as $Y = f(I, x)$ hides the edit that satisfies x in this particular scene. We make that edit explicit as a specification z listing the relevant entities, the action, the final state, and the preservation scope. Correctness is not one

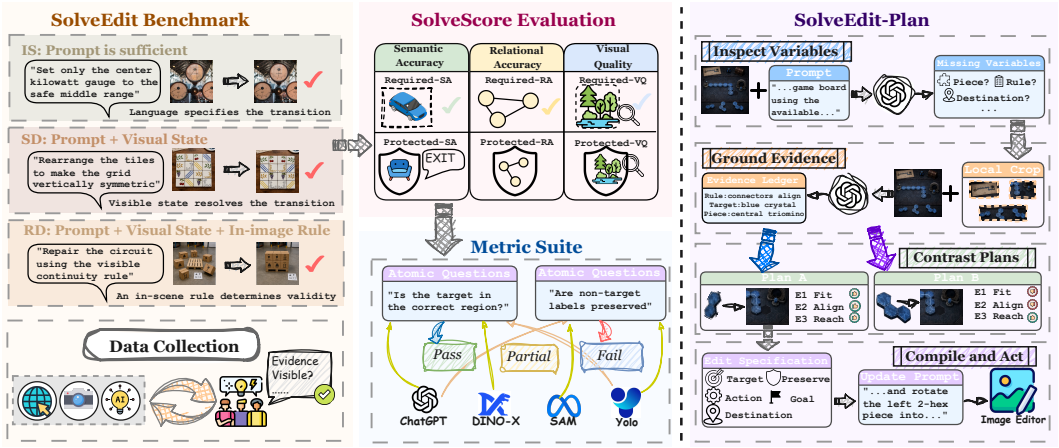


Figure 3: **Overview of SOLVEEDIT, SOLVESCORE, and SOLVEEDIT-PLAN. Left:** cases are grouped by the information needed to determine a valid edit. **Center:** atomic criteria specify the required change and the visual state to preserve; SOLVESCORE aggregates their satisfaction. **Right:** SOLVEEDIT-PLAN instantiates the transition and compiles an instruction for an unmodified editor.

target image. Several specifications z can satisfy the same request, and we collect them in the set $\mathcal{Z}^*(I, x)$. Figure 3 places this set next to the transition contract and the planning interface.

Once visible entities are grounded, the remaining question is what fixes z. The image is always used for localization and rendering, so it stays in the pipeline regardless of regime. Four ingredients can enter: request-to-entity bindings $B(I, x)$, background knowledge K, current-scene facts $S(I)$, and an in-scene rule $\mathcal{R}(I)$ (a legend, a constraint, or a repeated pattern). Eq. equation 1 writes the admissible set as a function Φ of these four:

$$\mathcal{Z}^*(I, x) = \Phi(x, B(I, x), K, S(I), \mathcal{R}(I)) \,. \tag{1}$$

A generated image Y belongs to $\mathcal{Y}^*(I, x)$ under two conditions: it realizes some $z \in \mathcal{Z}^*(I, x)$, and visual state outside that z is preserved:

$$\mathcal{Y}^*(I, x) = \{Y : \exists z \in \mathcal{Z}^*(I, x),\ \text{Realize}(Y; I, z) \wedge \text{Preserve}(Y; I, z)\} \,. \tag{2}$$

Two renderings of the same z both count. A plausible rendering of the wrong z does not.

The three regimes do not overlap, even though background knowledge K may be used in any of them. The label is assigned after grounding and K are applied: it records which case-specific piece of information fixes the solution. Under that rule, **IS** means the grounded request already fixes z; **SD** means a current-scene fact resolves a variable the request leaves open (target, action, destination, route, or final state); **RD** means the model also reads a rule shown or instantiated in the image. The label uses the minimum information needed, so it is not a difficulty rating. Appendix A.4 gives the decision protocol, including examples that distinguish adjacent regimes.

Annotators follow a fixed procedure. They first bind the request to visible entities, then ask whether the grounded request together with K fixes the transition. If it does not, a current-scene fact or a case-local rule must resolve the remaining variable, and that fact decides the label. Ordinary localization stays in IS; knowledge that never appears in the image cannot make a case RD. The label is therefore operational, not a claim about model architecture: IS cases still require visual grounding, and an RD case can be easier than an IS one.

A case can fail in two distinct ways. The model may settle on an invalid z, choosing an occupied destination or applying the wrong visible rule. Or it may recover a valid z and fail to render it: the route comes back disconnected, an object lands in the wrong place, or unrelated content changes. We call the first a solution-recovery error and the second a visual-execution error. The evaluation covers both: it scores the solved state and, separately, the content that should have stayed unchanged.

3.2 BENCHMARK CONSTRUCTION AND COVERAGE

The 2,728 cases are each built around a visual problem in a concrete application. The request given to the model states only the goal. It never reveals the scene state or rule attached to the case's regime. Behind each case, the authors record the evidence needed to resolve the request, mark one or

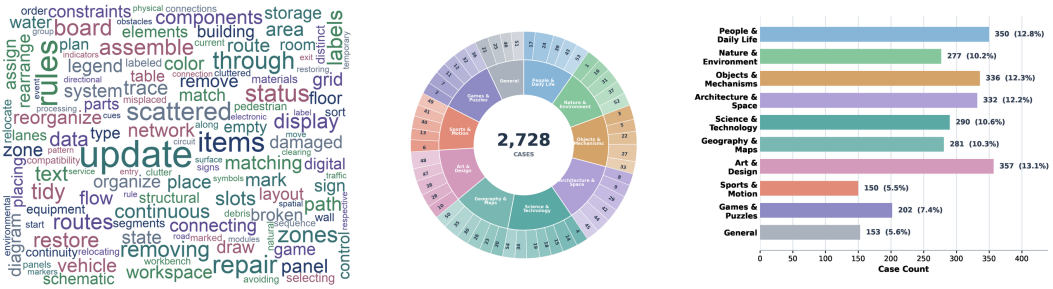


Figure 4: Statistics of the 2,728-case SOLVEEDIT benchmark. **Left:** frequent terms in editing requests. **Center:** coverage across 10 application domains and 54 canonical subdomains. **Right:** case counts across the application domains.

more admissible final states, and mark the content allowed to change. A case survives review only when four conditions hold: the intended change is semantically checkable; its evidence is visible in the image; correctness can be stated without matching an exact reference; and the change can be separated from protected content. Cases that fail, or rest on private assumptions, are revised or rejected before entering the manifest.

Each retained case is stored as

$$c = (I, x, m, \mathcal{C}^{\text{req}}, \mathcal{C}^{\text{pro}}, \mathcal{A}) \,, \tag{3}$$

Here m holds descriptive metadata, $\mathcal{C}^{\text{req}}$ and $\mathcal{C}^{\text{pro}}$ hold the required and protected conditions, and $\mathcal{A}$ holds optional audit assets. A reference edit may be attached to show one admissible solution; it has no role in defining correctness. The request shown to the model and the evidence used to annotate the case are stored separately.

The 10 domains and 54 subdomains cover people and everyday life, natural and environmental systems, objects and mechanisms, architecture and space, science and technology, geography and maps, art and design, sports and motion, games and puzzles, and general scenes. A domain label records the setting rather than the entities, and it is independent of the regime. Photographic, animated, cinematic, game, and purpose-built imagery all appear, with underrepresented settings and transitions authored on purpose. Figure 4 gives this coverage and the request distribution. Source diversity alone does not qualify a case: the image has to support a well-defined problem that can be checked visually.

Quality control checks five things: image validity, request grounding, contract coverage, dependency evidence, and split integrity. Shared scenes, near duplicates, and common generation templates are grouped before the data are split. Automated tools may order the review queue; they never decide a released label. Before manifests are produced, a release validator checks schema validity, criterion coverage, image paths, and hashes. The request hides the state or rule of its regime, and the hidden contract records the observable required and protected consequences. A case is revised or withheld when that correspondence breaks or any check fails. Appendix A gives the full procedures.

3.3 ATOMIC EVALUATION AND SOLVESCORE

No one reference image covers $\mathcal{Y}^*(I, x)$. We therefore pair each case with an atomic *transition contract*. Every criterion q makes a single observable claim, with the pass, partial, and fail conditions and the evidence needed to judge it written out. The evaluator returns $a_q \in \{1, 0.5, 0\}$. It abstains when the visible evidence is insufficient.

Criteria have two roles and three diagnostic properties. Required criteria state what the edit must accomplish, and protected criteria state what must remain intact. A target car left outside a valid space fails a required condition; a second car moved by mistake fails an independent protected one. We keep a protected criterion only when it can fail with every required condition passing, so one error is never counted twice.

SA covers entities, identities, counts, attributes, text, symbols, and intrinsic states. RA covers spatial and structural relations: placement, ordering, containment, connectivity, topology. VQ covers visible rendering defects such as residual traces, malformed geometry, and illegible content. Crossing the two roles with the three properties gives six cells. For role $r \in \{\text{req}, \text{pro}\}$ and property

$k \in \{\mathrm{SA}, \mathrm{RA}, \mathrm{VQ}\}$:

$$S_{r,k} = \frac{\sum_{q:r(q)=r,\, d(q)=k,\, a_q \neq \perp} \omega_q a_q}{\sum_{q:r(q)=r,\, d(q)=k,\, a_q \neq \perp} \omega_q}, \tag{4}$$

$a_q = \perp$ marks an abstention, and only non-abstained atoms enter the sums. All six scores are higher-is-better. The weights are hierarchical, so redundant criteria or mechanically split atoms cannot inflate a requirement.

With normalized weights w_q and u_q, each summing to one over the required and protected criteria, completion and damage are

$$R = \sum_{q \in \mathcal{C}^{\mathrm{req}}} w_q a_q, \qquad D = \sum_{q \in \mathcal{C}^{\mathrm{pro}}} u_q (1 - a_q). \tag{5}$$

Abstained criteria add zero without redistributing weight. The case score is

$$\textsc{SolveScore}_\lambda = G_{\mathrm{quality}} \max(0, R - \lambda D)\,, \tag{6}$$

G_{quality} rejects outputs that are missing, blank, severely corrupted, unrelated to the request, or unusable. The primary protocol uses $\lambda = 0.5$: completion stays the main objective, and damage can remove at most half the score. Cases are scored individually before averaging. We report R, D, the six diagnostics, the quality-gate pass rate, and coverage separately; coverage is the share of criterion weight with a non-abstained verdict. Appendix B covers weights, abstentions, and authoring rules.

A VLM performs most of the evaluation. A fixed subset of criteria is also assigned to specialized checkers beforehand, and the assignment is the same for every model. These are the conditions where VLM judgments are less reliable or a tool gives a more objective check. The evaluator sees the input, output, request, and full contract; the generator never sees the contract. For assigned criteria, SAM and YOLO supply segmentation and detection evidence next to the VLM's interpretation. Their output is a quality gate plus atomic verdicts with evidence, and a deterministic scorer aggregates them without further judgment. All editors use the same protocol, VLM version, and scoring code. Generation failures, gate failures, coverage, and abstentions are reported apart from the aggregate score, making the limits of the available evaluation evidence visible.

4 Planning Before Editing

SolveEdit-Plan tests whether a model can determine the transition before it renders anything. It is a two-stage test-time planner with no learned parameters, and the editor is left unchanged. The first stage, Inspect, finds the unresolved variables and gathers targeted visual evidence. The second, Resolve, compares plausible transitions and compiles the chosen one into an instruction for a single call to the original editor:

$$(\hat{z}, \hat{O}, \tilde{x}) = H(I, x), \qquad \hat{Y} = f(I, \tilde{x}), \tag{7}$$

Here $\hat{z}$ is the selected transition, $\hat{O}$ the observable completion and preservation obligations, and $\tilde{x}$ the instruction handed to the generator.

Transition planning. Inspect looks for variables the request leaves unresolved: target, action, destination, final state, preservation scope. It requests targeted crops for those variables and records the evidence each crop provides. Resolve then compares the plausible transitions against that evidence. The transition it selects is compiled into completion conditions and preservation obligations for the final editing call to the unchanged generator.

Generation interface and control. The compiled instruction goes to the original generator in one call. The obligations in $\hat{O}$ shape that instruction, but they are not the human-authored contract SolveScore uses, and the intermediate transition gets no supervision or evaluator feedback. At inference the planner sees only the source image and the goal: no reference outputs, contracts, authored regions, manual facts, dependency labels, or evaluator responses. The matched control, Generic Vision Rewrite, runs on the same backend with two agent calls and one final generation. It returns an unconstrained rewritten instruction, without explicit transition variables, candidate comparison, or preservation obligations. Appendix E gives the agent configuration, schemas, accounting rules, and implementation details.

Table 1: Performance on SOLVEEDIT (%, higher is better unless marked otherwise). Evaluation coverage is reported in Appendix D.1; video outputs are scored from their final frames. SA, RA, and VQ denote Semantic Accuracy, Relational Accuracy, and Visual Quality.

Model	Required			Protected			R↑	D↓	SOLVESCORE↑
	SA↑	RA↑	VQ↑	SA↑	RA↑	VQ↑			
Open-Source Models									
OmniGen2 (Wu et al., 2026a)	19.8	16.2	25.4	49.9	64.3	53.4	18.1	45.9	9.7
BAGEL (Deng et al., 2025)	27.9	20.0	24.2	51.7	60.7	53.6	22.4	45.5	12.1
FLUX.1 Kontext [dev] (Labs et al., 2025)	25.4	20.5	40.3	76.4	75.8	80.0	24.2	22.5	18.7
Qwen-Image-Edit-2509 (Wu et al., 2025b)	36.4	28.2	45.8	48.1	53.9	56.4	32.7	47.5	20.8
FLUX.2 [dev] (Labs, 2025)	38.9	30.6	51.7	62.9	65.3	77.3	31.7	32.3	22.8
Commercial Models									
Qwen Image 2.0 Pro (Zhao et al., 2026a)	53.0	47.6	68.0	65.0	68.7	81.6	50.7	28.8	40.6
Gemini 3.1 Flash Image (Google DeepMind, 2026)	68.3	64.0	78.0	71.5	72.3	88.4	66.0	23.1	55.5
Seedream 5.0 Pro (ByteDance Seed, 2026)	65.1	63.0	76.0	79.4	75.5	92.2	64.3	17.6	56.6
GPT-Image-2 (OpenAI, 2026)	69.6	64.6	81.7	71.7	65.2	90.7	67.6	23.9	57.0
Image-to-Video Models									
HunyuanVideo-1.5 (Wu et al., 2025a)	16.9	17.9	22.5	42.2	57.4	41.9	17.3	55.5	7.4
Kling V3 (Kuaishou Technology, 2026)	37.6	37.1	54.7	64.1	59.7	75.5	39.1	34.0	27.5

5 EXPERIMENTS

We organize our experiments around three questions linking performance, diagnosis, and planning. RQ1: How well do current generative models solve SOLVEEDIT tasks? RQ2: What do visually plausible failures reveal about the limitations of current models? RQ3: Does explicitly determining the transition before generation improve task completion and preservation with a fixed generator?

5.1 EXPERIMENTAL SETUP

Models and inputs. We evaluate nine image-to-image models (five open-weight and four commercial) and two image-to-video models on the same 2,728-case benchmark manifest. Table 1 lists all evaluated systems. In the direct evaluation, every model receives the same source image and intent-level request, and we score one output per case. The planning comparisons use GPT-Image-2, Qwen-Image-Edit-2509, and HunyuanVideo-1.5 with fixed final generators; controls and planning budgets are described in RQ3.

Evaluation protocol. Outputs are scored with the same hidden transition contracts and deterministic SOLVESCORE aggregation, using $\lambda = 0.5$ throughout. The evaluator receives the source image, request, model output, and contract; the contract is withheld from the generator. Scores are computed per case and averaged over the full manifest, giving each case equal overall weight. Failed generations are retried at most twice; missing or unusable outputs receive zero scores and remain in the denominator. Video outputs are scored from their final frame with the same contract; this evaluates the completed visual state and treats video results as a complementary cross-modal study, not a temporal-reasoning benchmark. Generation settings, retry policy, coverage, checkpoints, and hardware are provided in Appendix D.1 and the release artifact.

5.2 HOW WELL DO CURRENT MODELS SOLVE VISUAL PROBLEMS? (RQ1)

Overall performance. Table 1 reports semantic, relational, and visual-quality scores for required and protected content, together with task completion R and collateral damage D. GPT-Image-2 obtains the highest SOLVESCORE at 57.0% among the eleven systems evaluated here, followed closely by Seedream 5.0 Pro (56.6%) and Gemini 3.1 Flash Image (55.5%). Commercial image editors substantially outperform the open-source systems, while the image-to-video models remain weaker under the final-frame protocol. Even the strongest model therefore solves only part of the required transition while preserving the surrounding scene.

Completion and preservation. Similar aggregate scores can conceal different completion and preservation profiles. GPT-Image-2 has the highest task-completion score ($R = 67.6\%$), but incurs more collateral damage than Seedream ($D = 23.9\%$ versus $D = 17.6\%$). Seedream therefore reaches a similar overall score through better preservation rather than higher completion. The near tie does not

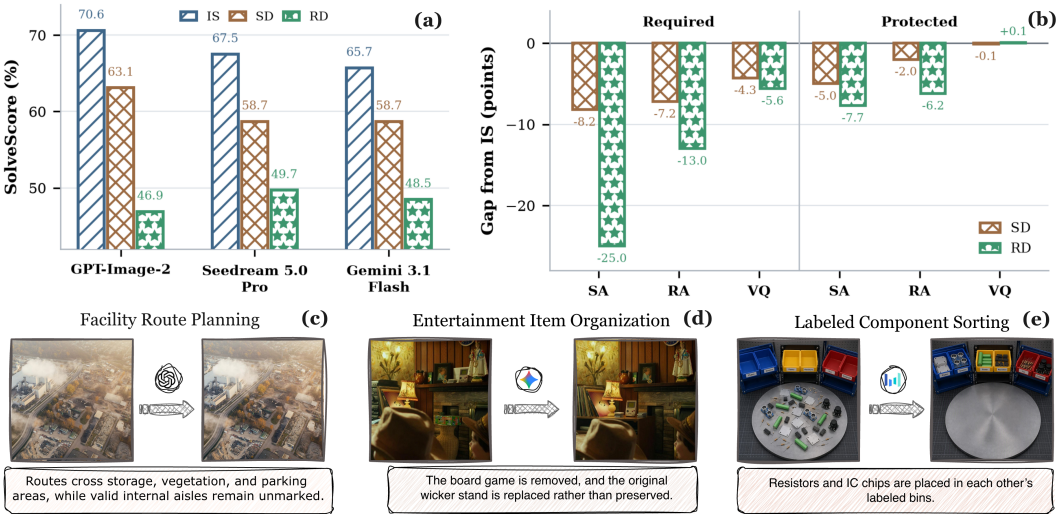


Figure 5: Reasoning failures behind visually plausible outputs. **Top left:** SOLVESCORE across IS, SD, and RD for the three leading models. **Top right:** application-domain-adjusted diagnostic gaps from IS, pooled across models and separated into required and protected conditions. **Bottom:** representative outputs that remain visually coherent but violate task-specific spatial, semantic, or rule-based requirements.

imply interchangeable behavior: improving task completion alone can leave substantial unintended changes unaddressed. Moreover, required visual quality exceeds required semantic and relational accuracy for all three leading models in Table 1. Visually coherent outputs thus remain an incomplete indicator of task success. These diagnostics identify which conditions fail; the regime-wise analysis below examines how these failures relate to the information needed to determine the transition.

5.3 WHAT DO VISUALLY PLAUSIBLE FAILURES REVEAL? (RQ2)

Performance declines with information dependence. Current models often produce coherent edits that nevertheless fail the task. We compare the three information-dependence regimes, IS, SD, and RD, and separate failures to satisfy required conditions from collateral changes to protected content. For each diagnostic cell, we compare regimes within application domains and aggregate the differences using common domain weights. This adjustment reduces the influence of domain composition, helping distinguish the observed regime gaps from differences in the mixture of task categories. Appendix D.2 gives the case counts, coverage, and domain-adjustment procedure.

Diagnostic gaps. Figure 5(a) shows a consistent gap from IS to RD for three leading models: RD trails IS by 17.2 to 23.7 points, while the SD gaps range from 7.0 to 8.8 points. This ordering is consistent with the additional information required to determine the transition: IS cases can be solved from the grounded request, whereas SD and RD cases require recovery of scene state or an in-image rule. After adjusting for domain composition, Figure 5(b) shows that the largest deficits from IS to RD occur in required semantic (-25.0 points) and relational (-13.0 points) conditions, compared with -5.6 points for required visual quality; protected dimensions show smaller deficits (-7.7, -6.2, +0.1 for SA, RA, VQ), with visual-quality preservation near neutral. This asymmetry indicates that the regime gap is concentrated in satisfying the requested change, with a much smaller deterioration in preserving unrelated content. Together with the qualitative examples in Figure 5(c) to (e), this pattern suggests that solution-recovery errors are a major contributor, rather than the failures arising only from visual execution. The regimes describe information dependence, not a prescribed difficulty ranking. This motivates testing explicit transition planning over generic quality interventions.

5.4 DOES EXPLICIT TRANSITION DETERMINATION IMPROVE A FIXED GENERATOR? (RQ3)

Matched comparison. We compare Direct, Text-only Rewrite, GENERIC VISION REWRITE, and SOLVEEDIT-PLAN with a matched one-call final-generation budget from the same fixed editor. Planning-call budgets are reported separately, so this comparison holds the case manifest, output count, and scoring contract constant while testing whether an explicit transition improves the editor beyond longer or more descriptive instructions. The ablations remove contrastive checking or preservation inference; protocol details are provided in Appendix E.

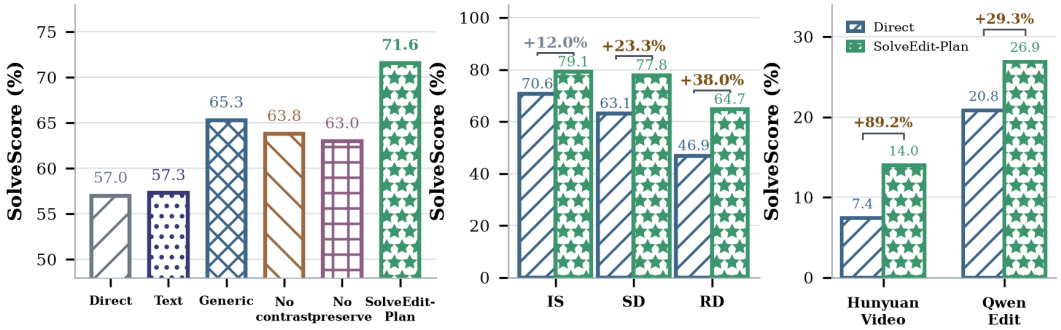


Figure 6: Effect of explicit transition inference. **Left:** GPT-Image-2 controls and component ablations. **Middle:** GPT-Image-2 gains by dependency level. **Right:** Direct versus SOLVEEDIT-PLAN across fixed generators. Every method uses one final generation.

Main gains and ablations. SOLVEEDIT-PLAN improves GPT-Image-2 from 57.0 to 71.6 SOLVESCORE, outperforming the budget-matched GENERIC VISION REWRITE by 6.3 points. The gain transfers to the tested open-source image editor and video generator without changing the parameters of either final generator. Compared with generic rewriting, required completion increases from 72.8 to 75.8, while collateral damage decreases from 16.3 to 8.6. The larger gains on SD and RD are consistent with the information-dependent failures diagnosed in RQ2; full ablations and regime-wise results are reported in Appendix D.1. Figure 6 summarizes the controls, ablations, regime-wise gains, and cross-generator transfer. Additional per-category and per-regime results, generation success, evaluator coverage, λ sensitivity, and component ablations are reported in Appendix D.1.

5.5 FURTHER ANALYSIS: COST AND CROSS-GENERATOR TRANSFER

Planning cost. The planning intervention adds a measurable test-time cost. Each case uses two LLM calls, one for Inspect and one for Resolve, with an average of 3,207 input tokens and 635 output tokens. This adds computation before the same single final generation, so the gains over Direct are obtained with additional planning computation. The stronger control is GENERIC VISION REWRITE: with the same planning backend and call budget, SOLVEEDIT-PLAN achieves a higher score on GPT-Image-2, supporting the value of structured transition planning within that budget. Equal call counts do not imply equal token use; implementation-dependent latency and cost accounting are provided in Appendix E.

Transfer across generators. The improvement is not limited to GPT-Image-2: under the same case manifest and transition contracts, SOLVEEDIT-PLAN increases SOLVESCORE from 20.8% to 26.9% for Qwen-Image-Edit-2509 and from 7.4% to 14.0% for HunyuanVideo-1.5. Both generators improve required completion and reduce collateral damage (Table 13), showing that the gains extend to both parts of the transition contract. This supports the applicability of the planning procedure through the instruction interface for the tested image and video generators, despite their different generation architectures and output modalities. Their low absolute scores nevertheless leave substantial residual error: these results establish neither reliable task completion nor whether the remaining failures originate in planning or generation.

6 CONCLUSION

We studied visual problem solving through generative transformation of an existing scene: the model determines a valid transition from the request and visual evidence, carries it out, and leaves unrelated content in place. SOLVEEDIT supplies 2,728 such cases organized by the information that fixes the required change, and its atomic contracts together with SOLVESCORE measure completion and preservation without relying on one reference output. The strongest evaluated model still reaches only 57.0%, with the largest gaps on scene- and rule-dependent cases. Recovering the transition before a single call to an unchanged generator closes part of that gap: GPT-Image-2 goes from 57.0% to 71.6% under a matched budget, and the same procedure helps the tested open-source editor and video generator. We hope the benchmark, with its explicit split between transition determination, visual execution, and preservation, gives a useful basis for distinguishing an incorrect transition from damage incurred during an otherwise correct edit.

AI USE STATEMENT

Generative AI models are used as integral components of the proposed system and the evaluation pipeline: GPT-5.6 Sol serves as the primary VLM evaluator and the resulting verdicts are aggregated by a deterministic scorer, as described throughout the paper and validated in Appendix C. Outside these research-specific uses, generative AI tools were used only to assist with software implementation, experiment automation, and to polish the writing. They were not used to originate research ideas, design the benchmark or method, assign benchmark labels, make experimental decisions, interpret results, or derive scientific conclusions. All AI-assisted code, labels, and text were reviewed and verified by the authors, and all reported results were derived from recorded experimental artifacts. The authors take full responsibility for the final text, claims, code, and other artifacts.

ETHICS STATEMENT

This work introduces a benchmark and an evaluation protocol for generative image editing; it does not train a new generative model and does not collect private personal data. The benchmark combines author-captured photographs, documented online sources, animation and game captures, cinematic frames, and AI-generated inputs. We record provenance and licensing notes for released cases, group near-duplicates before partitioning, and exclude sensitive identity transformations; cases that depend on inaccessible private assumptions or cannot be visually checked are revised or withheld. Because the benchmark spans diagrams, public spaces, and social or professional settings, some cases may reflect cultural or domain-specific conventions, which we document together with a process for reporting ambiguous or harmful content.

Human judgments used for metric validation were provided by professional annotators at a non-profit annotation organization. All annotators were adults, were informed that their judgments would be used for research, and participated under the organization's standard labor terms, including fair compensation. They reviewed only source images, task descriptions, and model outputs, and no personal identifying information of annotators or third parties was collected or stored. The task consisted solely of minimal-risk technical judgments about image quality and involved no intervention with, or data about, identifiable individuals.

Because the evaluated systems build on existing image and video generators, they may inherit the risks of those components, including harmful or misleading content, representational bias, and potential misuse. We encourage deployments to retain the safety mechanisms and usage policies of the underlying models and to comply with applicable licensing and data-use requirements.

REPRODUCIBILITY STATEMENT

The paper specifies the task formulation, dependency-regime protocol, atomic contract schema, scoring equations, evaluator boundaries, and generation protocol. Appendix A documents benchmark construction, provenance, annotation, and quality control; Appendix C describes the human metric-validation protocol; Appendix D.1 reports model configurations, coverage, per-domain results, and sensitivity analyses; and Appendix E reports the matched planning comparisons. An anonymous code repository accompanying the submission contains versioned prompts, schemas, manifests, evaluation code, random seeds, intermediate records, and generation logs, together with instructions for reproducing the reported analyses. Results for API-based generators depend on provider endpoints as of September 2026; we document endpoint identifiers and model versions, and note that later provider updates may change these numbers.

REFERENCES

Omer Bar-Tal, Hila Chefer, Omer Tov, Charles Herrmann, Roni Paiss, Shiran Zada, Ariel Ephrat, Junhwa Hur, Guanghui Liu, Amit Raj, et al. Lumiere: A space-time diffusion model for video generation. In *SIGGRAPH Asia 2024 conference papers*, pp. 1–11, 2024.

Manuel Brack, Felix Friedrich, Katharia Kornmeier, Linoy Tsaban, Patrick Schramowski, Kristian Kersting, and Apolinário Passos. Ledits++: Limitless image editing using text-to-image models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 8861–8870, 2024.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18392–18402. IEEE, 2023.

ByteDance Seed. Seedream 5.0 pro. https://seed.bytedance.com/en/seedream5_0_pro, 7 2026.

Mingdeng Cao, Xintao Wang, Zhongang Qi, Ying Shan, Xiaohu Qie, and Yinqiang Zheng. Masactrl: Tuning-free mutual self-attention control for consistent image synthesis and editing. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 22503–22513. IEEE, 2023.

Harold Haodong Chen, Disen Lan, Wen-Jie Shu, Qingyang Liu, Zihan Wang, Sirui Chen, Wenkai Cheng, Kanghao Chen, Hongfei Zhang, Zixin Zhang, Rongjin Guo, Yu Cheng, and Ying-Cong Chen. TiViBench: Benchmarking think-in-video reasoning for video generative models. *arXiv preprint arXiv:2511.13704*, 2025.

Liang Chen, Weichu Xie, Yiyan Liang, Hongfeng He, Hans Zhao, Zhibo Yang, Zhiqi Huang, Haoning Wu, Haoyu Lu, Yiping Bao, et al. Babyvision: Visual reasoning beyond language. *arXiv preprint arXiv:2601.06521*, 2026.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Zhong Muyan, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. Intern vl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24185–24198, 2023. URL https://api.semanticscholar.org/CorpusID:266521410.

Jaemin Cho, Yushi Hu, Jason Baldridge, Roopal Garg, Peter Anderson, Ranjay Krishna, Mohit Bansal, Jordi Pont-Tuset, and Su Wang. Davidsonian scene graph: Improving reliability in fine-grained evaluation for text-to-image generation. In *International conference on learning representations*, volume 2024, pp. 15625–15645, 2024.

François Chollet. On the measure of intelligence. *arXiv preprint arXiv:1911.01547*, 2019.

En Ci, Shanyan Guan, Yanhao Ge, Yilin Zhang, Wei Li, Zhenyu Zhang, Jian Yang, and Ying Tai. Describe, don't dictate: Semantic image editing with natural language intent. In *IEEE/CVF International Conference on Computer Vision*, 2025. doi: 10.1109/ICCV51701.2025.01783.

Chaorui Deng, Deyao Zhu, Kunchang Li, Chenhui Gou, Feng Li, Zeyu Wang, Shu Zhong, Weihao Yu, Xiaonan Nie, Ziang Song, et al. Emerging properties in unified multimodal pretraining. *arXiv preprint arXiv:2505.14683*, 2025.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, Peixian Chen, Yanwei Li, Shaohui Lin, Sirui Zhao, Ke Li, Tong Xu, Xiawu Zheng, Enhong Chen, Caifeng Shan, Ran He, and Xing Sun. Video-MME: The first-ever comprehensive evaluation benchmark of multi-modal LLMs in video analysis. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025. doi: 10.1109/CVPR52734.2025.02245.

Tsu-Jui Fu, Wenze Hu, Xianzhi Du, William Wang, Yinfei Yang, and Zhe Gan. Guiding instruction-based image editing via multimodal large language models. In *International Conference on Learning Representations*, volume 2024, pp. 54820–54833, 2024.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. Geneval: An object-focused framework for evaluating text-to-image alignment. *Advances in Neural Information Processing Systems*, 36: 52132–52152, 2023.

Google DeepMind. Gemini 3.1 flash image model card. https://deepmind.google/models/model-cards/gemini-3-1-flash-image/, 2 2026.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6904–6913, 2017.

Feng Han, Yibin Wang, Chenglin Li, Zheming Liang, Dianyi Wang, Yang Jiao, Zhipeng Wei, Chao Gong, Cheng Jin, and Jiaqi Wang. Unireditbench: A unified reasoning-based image editing benchmark. *arXiv preprint arXiv:2511.01295*, 2025.

Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. Clipscore: A reference-free evaluation metric for image captioning. In *Proceedings of the 2021 conference on empirical methods in natural language processing*, pp. 7514–7528, 2021.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Hexiang Hu, Kelvin C. K. Chan, Yu-Chuan Su, Wenhu Chen, Yandong Li, Kihyuk Sohn, Yang Zhao, Xue Ben, Boqing Gong, William Cohen, Ming-Wei Chang, and Xuhui Jia. Instruct-imagen: Image generation with multi-modal instruction, 2024. URL https://arxiv.org/abs/2401.01952.

Yushi Hu, Benlin Liu, Jungo Kasai, Yizhong Wang, Mari Ostendorf, Ranjay Krishna, and Noah A Smith. Tifa: Accurate and interpretable text-to-image faithfulness evaluation with question answering. In *2023 ieee/cvf international conference on computer vision (iccv)*, pp. 20349–20360. IEEE, 2023.

Yuzhou Huang, Liangbin Xie, Xintao Wang, Ziyang Yuan, Xiaodong Cun, Yixiao Ge, Jiantao Zhou, Chao Dong, Rui Huang, Ruimao Zhang, et al. Smartedit: Exploring complex instruction-based image editing with multimodal large language models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8362–8371. IEEE, 2024a.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818. IEEE, 2024b.

Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6693–6702. IEEE, 2019.

Mude Hui, Siwei Yang, Bingchen Zhao, Yichun Shi, Heng Wang, Peng Wang, Yuyin Zhou, and Cihang Xie. Hq-edit: A high-quality dataset for instruction-based image editing. *arXiv preprint arXiv:2404.09990*, 2024.

Liya Ji, Chenyang Qi, and Qifeng Chen. Instruction-based image editing with planning, reasoning, and generation. In *IEEE/CVF International Conference on Computer Vision*, 2025. doi: 10.1109/ICCV51701.2025.01626.

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2901–2910, 2017.

Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6007–6017. IEEE, 2023.

Dan Kondratyuk, Lijun Yu, Xiuye Gu, José Lezama, Jonathan Huang, Grant Schindler, Rachel Hornung, Vighnesh Birodkar, Jimmy Yan, Ming-Chang Chiu, et al. Videopoet: A large language model for zero-shot video generation. *arXiv preprint arXiv:2312.14125*, 2023.

Weijie Kong, Qi Tian, Zijian Zhang, Rox Min, Zuozhuo Dai, Jin Zhou, Jiangfeng Xiong, Xin Li, Bo Wu, Jianwei Zhang, et al. Hunyuanvideo: A systematic framework for large video generative models. *arXiv preprint arXiv:2412.03603*, 2024.

Max Ku, Dongfu Jiang, Cong Wei, Xiang Yue, and Wenhu Chen. Viescore: Towards explainable metrics for conditional image synthesis evaluation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12268–12290, 2024a.

Max Ku, Tianle Li, Kai Zhang, Yujie Lu, Xingyu Fu, Wenwen Zhuang, and Wenhu Chen. Imagenhub: Standardizing the evaluation of conditional image generation models. In *International Conference on Learning Representations*, volume 2024, pp. 46689–46722, 2024b.

Kuaishou Technology. Kling Video 3.0 model user guide. Kling AI documentation, 2 2026. URL https://kling.ai/quickstart/klingai-video-3-model-user-guide.

Black Forest Labs. FLUX.2: Frontier Visual Intelligence. https://bfl.ai/blog/flux-2, 2025.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, et al. Flux. 1 kontext: Flow matching for in-context image generation and editing in latent space. *arXiv preprint arXiv:2506.15742*, 2025.

Ouxiang Li, Yuan Wang, Xinting Hu, Huijuan Huang, Rui Chen, Jiarong Ou, Xin Tao, Pengfei Wan, Xiaojuan Qi, and Fuli Feng. Easier painting than thinking: Can text-to-image models set the stage, but not direct the play? In *International Conference on Learning Representations*, volume 2026, pp. 86729–86758, 2026.

Yiwei Ma, Jiayi Ji, Ke Ye, Weihuang Lin, Zhibin Wang, Yonghan Zheng, Qiang Zhou, Xiaoshuai Sun, and Rongrong Ji. I2ebench: A comprehensive benchmark for instruction-based image editing. *Advances in Neural Information Processing Systems*, 37:41494–41516, 2024.

Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge. In *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 3190–3199. IEEE, 2019.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. *arXiv preprint arXiv:2108.01073*, 2021.

Ron Mokady, Amir Hertz, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Null-text inversion for editing real images using guided diffusion models. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6038–6047. IEEE, 2023.

OpenAI. Introducing chatgpt images 2. https://platform.openai.com/docs/models/gpt-image-2, 2026.

Gaurav Parmar, Krishna Kumar Singh, Richard Zhang, Yijun Li, Jingwan Lu, and Jun-Yan Zhu. Zero-shot image-to-image translation. In *ACM SIGGRAPH 2023 conference proceedings*, pp. 1–11, 2023.

Yusu Qian, Jiasen Lu, Tsu-Jui Fu, Xinze Wang, Chen Chen, Yinfei Yang, Wenze Hu, and Zhe Gan. Gie-bench: Towards grounded evaluation for text-guided image editing. *arXiv preprint arXiv:2505.11493*, 2025.

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

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36:68539–68551, 2023.

Shelly Sheynin, Adam Polyak, Uriel Singer, Yuval Kirstain, Amit Zohar, Oron Ashual, Devi Parikh, and Yaniv Taigman. Emu edit: Precise image editing via recognition and generation tasks. In *2024 ieee/cvf conference on computer vision and pattern recognition (cvpr)*, pp. 8871–8879. IEEE, 2024.

Alane Suhr, Stephanie Zhou, Ally Zhang, Iris Zhang, Huajun Bai, and Yoav Artzi. A corpus for reasoning about natural language grounded in photographs. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 6418–6428, 2019.

Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1921–1930. IEEE, 2023.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024a.

Su Wang, Chitwan Saharia, Ceslee Montgomery, Jordi Pont-Tuset, Shai Noy, Stefano Pellegrini, Yasumasa Onoe, Sarah Laszlo, David J Fleet, Radu Soricut, et al. Imagen editor and editbench: Advancing and evaluating text-guided image inpainting. In *2023 ieee/cvf conference on computer vision and pattern recognition (cvpr)*, pp. 18359–18369. IEEE, 2023.

Zhenyu Wang, Aoxue Li, Zhenguo Li, and Xihui Liu. Genartist: Multimodal llm as an agent for unified image generation and editing. *Advances in Neural Information Processing Systems*, 37: 128374–128395, 2024b.

Bing Wu, Chang Zou, Changlin Li, Duojun Huang, Fang Yang, Hao Tan, Jack Peng, Jianbing Wu, Jiangfeng Xiong, Jie Jiang, et al. Hunyuanvideo 1.5 technical report. *arXiv preprint arXiv:2511.18870*, 2025a.

Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. Visual chatgpt: Talking, drawing and editing with visual foundation models. *arXiv preprint arXiv:2303.04671*, 2023a.

Chenfei Wu, Jiahao Li, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kun Yan, Sheng ming Yin, Shuai Bai, Xiao Xu, Yilei Chen, Yuxiang Chen, Zecheng Tang, Zekai Zhang, Zhengyi Wang, An Yang, Bowen Yu, Chen Cheng, Dayiheng Liu, Deqing Li, Hang Zhang, Hao Meng, Hu Wei, Jingyuan Ni, Kai Chen, Kuan Cao, Liang Peng, Lin Qu, Minggang Wu, Peng Wang, Shuting Yu, Tingkun Wen, Wensen Feng, Xiaoxiao Xu, Yi Wang, Yichang Zhang, Yongqiang Zhu, Yujia Wu, Yuxuan Cai, and Zenan Liu. Qwen-image technical report, 2025b. URL https://arxiv.org/abs/2508.02324.

Chenyuan Wu, Jiahao Wang, Pengfei Zheng, Ruiran Yan, Shitao Xiao, Xin Luo, Yueze Wang, Wanli Li, Xiyan Jiang, Yexin Liu, et al. Omnigen2: Towards instruction-aligned multimodal generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 21964–21975, 2026a.

Haoning Wu, Zicheng Zhang, Erli Zhang, Chaofeng Chen, Liang Liao, Annan Wang, Chunyi Li, Wenxiu Sun, Qiong Yan, Guangtao Zhai, et al. Q-bench: A benchmark for general-purpose foundation models on low-level vision. In *International Conference on Learning Representations*, volume 2024, pp. 12547–12573, 2024.

Xiaoshi Wu, Keqiang Sun, Feng Zhu, Rui Zhao, and Hongsheng Li. Human preference score: Better aligning text-to-image models with human preference. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2096–2105. IEEE, 2023b.

Yongliang Wu, Zonghui Li, Xinting Hu, Xinyu Ye, Xianfang Zeng, Gang Yu, Wenbo Zhu, Bernt Schiele, Ming-Hsuan Yang, and Xu Yang. Kris-bench: Benchmarking next-level intelligent image editing models. *Advances in Neural Information Processing Systems*, 38, 2026b.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation. *Advances in Neural Information Processing Systems*, 36:15903–15935, 2023.

Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. Mm-react: Prompting chatgpt for multimodal reasoning and action. *arXiv preprint arXiv:2303.11381*, 2023.

Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. In *International Conference on Learning Representations*, volume 2025, pp. 83048–83077, 2025.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Qifan Yu, Wei Chow, Zhongqi Yue, Kaihang Pan, Yang Wu, Xiaoyang Wan, Juncheng Li, Siliang Tang, Hanwang Zhang, and Yueting Zhuang. Anyedit: Mastering unified high-quality image editing for any idea. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26125–26135. IEEE, 2025.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9556–9567, 2024.

Huanyu Zhang, Xuehai Bai, Chengzu Li, Chen Liang, Haochen Tian, Haodong Li, Ruichuan An, Yifan Zhang, Anna Korhonen, Zhang Zhang, et al. How well do models follow visual instructions? vibe: A systematic benchmark for visual instruction-driven image editing. *arXiv preprint arXiv:2602.01851*, 2026.

Kai Zhang, Lingbo Mo, Wenhu Chen, Huan Sun, and Yu Su. Magicbrush: A manually annotated dataset for instruction-guided image editing. *Advances in neural information processing systems*, 36:31428–31449, 2023.

Bing Zhao, Chenfei Wu, Deqing Li, Hao Meng, Jiahao Li, Jie Zhang, Jingren Zhou, Junyang Lin, Kaiyuan Gao, Kuan Cao, et al. Qwen-image-2.0 technical report. *arXiv preprint arXiv:2605.10730*, 2026a.

Haozhe Zhao, Xiaojian Ma, Liang Chen, Shuzheng Si, Rujie Wu, Kaikai An, Peiyu Yu, Minjia Zhang, Qing Li, and Baobao Chang. Ultraedit: Instruction-based fine-grained image editing at scale. *Advances in Neural Information Processing Systems*, 37:3058–3093, 2024.

Xiangyu Zhao, Peiyuan Zhang, Kexian Tang, Xiaorong Zhu, Hao Li, Wenhao Chai, Zicheng Zhang, Renqiu Xia, Guangtao Zhai, Junchi Yan, et al. Envisioning beyond the pixels: Benchmarking reasoning-informed visual editing. *Advances in Neural Information Processing Systems*, 38, 2026b.

Dian Zheng, Manyuan Zhang, Hongyu Li, Hongbo Liu, Kai Zou, Kaituo Feng, and Hongsheng Li. Uni-edit: Intelligent editing is a general task for unified model tuning. *arXiv preprint arXiv:2605.21487*, 2026.

A Benchmark Composition and Annotation

A.1 Data Sources and Final Composition

SolveEdit contains 2,728 cases across 10 application domains and 54 canonical subdomains. Application domain, solution-dependency regime, and image provenance are separate annotations. The table below reports only image provenance, namely how the source image was obtained. It therefore does not mix the benchmark's information-dependence labels with its media sources.

The real-image subset contains photographs obtained from publicly accessible online sources and photographs captured by the authors. Online photographs are retained only when their source and usage status can be documented; author- captured photographs are identified separately in the release metadata. The animation and game category contains frames or captures from animated media, games, and game-like interfaces, while film and cinematic cases are retained as a separate media category.

Table 2: Source and acquisition composition of the 2,728 benchmark cases. Categories are mutually exclusive and independent of application domain, task family, and solution-dependency regime.

Source and acquisition method	Cases
AI-generated inputs, purpose-built or image-conditioned	1,375
Real photographs, collected online or captured by the authors	646
Animation and game captures	377
Film and cinematic frames	330
Total	2,728

AI-generated cases are not a single visual style. They include purpose-built synthetic scenes authored from text specifications and image-conditioned cases in which a real or sourced image is transformed or extended before being used as a benchmark input. The canonical 2,728-case manifest records these cases under the common model-generated provenance label, so the table reports their aggregate count rather than introducing unsupported subcounts. The release metadata records the generation model, conditioning mode when applicable, source identifier, acquisition method, and applicable license or usage note for each case.

These source categories describe acquisition rather than visual content. For example, a photograph can depict a game-like board, and an AI-generated image can depict any application domain; neither fact changes its provenance label. Likewise, IS, SD, and RD indicate where the information needed to determine a transition resides and are not alternative names for the source categories.

A.2 Data Production and Filtering

The benchmark was assembled through a staged production pipeline rather than by directly selecting the final cases. For sourced media, we acquired images from documented sources or captured them directly, decoded and cropped usable frames, applied an automatic visual filter, sent surviving candidates to VLM review, and then retained cases after manual task authoring and contract review. AI-generated cases followed the same review stages after generation, except that web collection and frame cropping were not required for every batch. The final manifest was then deduplicated and quality-checked before the composition in Table 2 was frozen.

Table 3 reports the source-level production logs used to characterize this process. These are candidate-pool counts from multiple collection rounds, not alternative definitions of the final benchmark size. The VLM-reviewed candidates were subsequently inspected and selected by human annotators. The final human-picked column is the count that enters the canonical manifest and therefore sums exactly to the 2,728 cases in Table 2. Animation and game captures are shown separately here because their collection logs were maintained independently, although they form one source category in the final composition table.

Production pipeline. Figure 7 summarizes the staged construction flow.

Data governance. **Duplicate control:** near-duplicate detection uses perceptual hashing on source images plus instruction similarity; shared scenes are grouped and at most one representative per group is retained unless the tasks are genuinely distinct. **Source and license metadata:** every case records its provenance (real photograph, animation, game capture, film frame, or AI-generated), source identifier, input image hash, and a content note for third-party frames. **Author-captured and**

Table 3: Source-level production statistics and human selection from the benchmark construction logs. Counts before the final column describe candidate pools across collection rounds; the final column is the number of human-picked cases retained in the canonical 2,728-case manifest.

Source	Initial candidates	Cropped	Filtered	VLM reviewed	Human picked
Real photographs	42,776	34,078	1,335	1,032	646
Animation frames	118,589	62,507	1,221	944	369
Game captures	1,260	325	219	176	8
Film/cinematic frames	41,467	26,608	1,158	825	330
AI-generated inputs	5,608	–	2,096	1,607	1,375

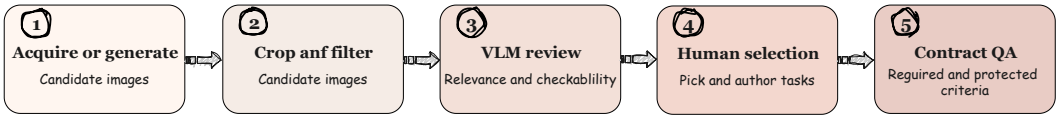


Figure 7: Benchmark production pipeline. Candidates are acquired or generated, screened, selected by humans, converted into contracts, and validated before the manifest is frozen. Failed candidates are rejected or revised.

AI-generated images: AI-generated images are produced with task-specific conditioning prompts and reviewed for visual checkability before authoring; no model output from the evaluated systems is used as a benchmark source. **Train/eval leakage:** source images may overlap with web-scale corpora, but the task requests, transition contracts, and scoring criteria are authored specifically for this benchmark and are not derivable from image captions alone. We do not claim decontamination against all training sets; the contract-based protocol reduces the risk that a model can succeed by recalling a specific image–text pair.

The pass rate is substantially higher for AI-generated candidates because they were produced from task-specific specifications, whereas sourced media were searched broadly and then filtered for visual checkability, usable resolution, and task relevance. The reported rate averages batch-level final pass rates and is therefore not obtained by dividing two aggregate columns in the table. It characterizes production selectivity across the two acquisition pipelines and should not be interpreted as a difference in benchmark difficulty or model performance on the retained benchmark cases.

A.3 Case Construction and Quality Assurance

Cases combine sourced real-world imagery, animation and cinematic frames, game scenes, and purpose-built visual problems. For each candidate, the authors review the source image, intent-level request, dependency label, and transition contract. A case is retained only when the intended outcome is visually checkable, the information needed to determine it is available, correctness can be expressed without exact reference matching, and required changes can be separated from protected content. Ambiguous or inaccessible cases are revised or withheld.

Exact duplicates, near-duplicate frames, and cases derived from the same underlying scene are grouped before evaluation partitions are formed. The release records case identifiers, source and license metadata, group identifiers, safety exclusions, and image hashes. A release validator checks schema validity, criterion coverage, image paths, and manifest integrity before evaluation.

A.4 Dependency-Regime Design

Annotators first bind request terms to visible entities and regions. They then apply the following decision sequence:

1. Assign IS when the grounded request and relevant background knowledge determine the semantic action and valid outcome.
2. Otherwise, assign SD when observable current-scene facts determine the missing target, action, destination, route, or final state.

3. Assign RD when a case-local rule, legend, reference relation, capacity, pattern, or compatibility condition must additionally be read or induced from the image.

The label records the minimum case-specific information needed to determine the transition, not task difficulty. Ordinary grounding remains IS. Background knowledge alone does not make a case RD unless a case-local rule is shown or instantiated in the image. Each assignment records the unresolved variable and the visual evidence used to resolve it.

Boundary examples. "Move the red marker to the left side of the board" is IS after grounding. "Move the marker to the available slot" is SD when occupancy determines the destination. "Arrange the markers according to the legend shown in the image" is RD when the visual legend determines the valid arrangement.

A.5 Prompt Suite for Visual Evaluation

The benchmark uses VLMs at two distinct stages: candidate screening during construction and evidence extraction during evaluation. These prompts are not the task instructions given to image editors. The editor receives only the case image and intent-level request; the VLM receives the image, the relevant case-specific question or contract atom, and an output schema. We keep the prompt families fixed across models and cases, while the nouns, regions, and criterion text are instantiated from the case record.

Table 4: Prompt suite used in benchmark construction and evaluation.

Prompt family	Inputs	Required output
Candidate screening	Candidate image, source metadata, and a short task brief	Image usability, task relevance, duplicate or near-duplicate flag, and a keep/reject decision
Case authoring check	Input image, intent-level request, and proposed transition	Whether the request is visually checkable, whether the target is under-specified as intended, and a concise issue list
Contract evidence	Input image, model output, the full set of atomic criteria, and pass/partial/fail definitions	One verdict, confidence, and localized visual evidence per criterion, returned in a single structured JSON response
Quality gate	Input image and model output	Valid, corrupted, unrelated, or unusable status with a failure reason
Dependency annotation	Input image, request, and IS/SD/RD decision rules	One dependency label, unresolved variable, and image evidence that resolves it

All VLM calls use structured JSON responses. The evaluator cannot change the contract, revise the request, or use a reference output to invent a new target. When evidence is insufficient, it returns "abstain" rather than guessing; the abstention is retained in the audit log and lowers evidence coverage, which measures the fraction of total criterion weight receiving a non-abstained verdict. The full versioned prompts and schemas are released with the evaluation code, while Table 4 gives the reviewer-facing prompt roles and information boundaries.

Representative contract-evidence prompt. Given the input image, the model output, and the list of criteria below, decide for each criterion whether the output is `pass`, `partial`, `fail`, or `abstain`. Use only visible evidence in the two images and the criterion definitions. Do not infer missing pixels, use the model identity, or compare against an unprovided reference output. Return a JSON array with one entry per criterion, each containing `verdict`, `confidence`, and `visual_evidence`. Criterion `i`: {`criterion_question`}. Pass condition: {`pass_criteria`}. Partial condition: {`partial_criteria`}. Fail condition: {`fail_criteria`}.

Example 1: Required RA criterion (partial verdict). The contract supplies the atomic criterion as input; the evaluator returns a structured verdict.

Input: contract atom

`id`: atom_cross_aisles_route
`role`: required, `property`: RA

Question: Are transverse (cross) routes marked connecting the main longitudinal routes across the clear floor?

Pass: Transverse or branching lines connect the main longitudinal routes.
Partial: Only one minor branching line, missing obvious connections.
Fail: No transverse or branching routes drawn.

Output: evaluator verdict

`verdict`: **partial**
`trigger_score`: 0.5
`confidence`: 0.82

Reason: Some branching is present, but the markings do not clearly provide cross-aisle connections.

Evidence: Yellow curved segments span portions of the open floor but do not form clear connections.

Example 2: Protected RA criterion (fail verdict). A protected criterion is scored positively (intact = pass); the deterministic scorer converts a fail into a deduction through $1 - a_q$.

Input: contract atom

`id`: atom_cables_layout_preserved
`role`: protected, `property`: RA

Question: Is the original structural layout and connectivity of the utility cables and clotheslines completely preserved?

Pass: The structural layout and connectivity remain completely preserved.
Fail: The layout is heavily distorted or broken.

Output: evaluator verdict

`verdict`: **fail**
`trigger_score`: 1.0
`confidence`: 0.93

Reason: The clothesline arrangement is substantially changed rather than merely rearranged on the original lines.

B More Details of SolveScore

Each case stores required postconditions and independent protected conditions. Every atomic criterion has an observable question, pass/partial/fail conditions, an evidence scope, a role, and one diagnostic property. Required criteria describe what the edit must accomplish; protected criteria describe what must remain intact. A protected criterion is retained only when it can fail while all required conditions pass, preventing duplicate counting.

B.1 Scoring Protocol

The evaluator first applies the quality gate to reject missing, blank, severely corrupted, unrelated, or unusable outputs. It then assigns each applicable atom one of {pass, partial, fail, abstain}. The deterministic scorer maps pass, partial, and fail to $a_q \in \{1, 0.5, 0\}$, respectively, and computes

$$R = \sum_{q \in \mathcal{C}^{\text{req}}} w_q a_q, \qquad D = \sum_{q \in \mathcal{C}^{\text{pro}}} u_q (1 - a_q), \tag{8}$$

$$\text{SolveScore}_\lambda = G_{\text{quality}} \max(0, R - \lambda D). \tag{9}$$

The primary protocol fixes $\lambda = 0.5$. Required and protected weights are normalized separately. Applicable SA, RA, and VQ properties receive equal mass within each role; atoms split from one criterion share their parent group mass. Abstained criteria retain their original weights but contribute zero to the corresponding sum: required abstentions earn no completion credit, and protected abstentions trigger no damage deduction. No weights are redistributed. If all required criteria abstain, $R = 0$; if all protected criteria abstain, $D = 0$, which indicates no established violation rather than verified preservation. Output-induced deletion, occlusion, or distortion that violates a criterion is scored as fail or partial according to its rubric, rather than abstain. Evidence coverage is the non-abstained criterion weight divided by the total criterion weight across both roles. Abstentions remain in this denominator and in the audit log. Coverage and generation failures are reported separately from the score.

Edge-case handling. **Quality-gate failure:** if the output is missing, blank, severely corrupted, unrelated to the input, or no longer a plausible edit, $G_{\text{quality}} = 0$ and the case receives SolveScore 0 regardless of atomic verdicts. **Generation failure:** a model that produces no usable image after at most two retries is scored as a quality-gate failure (0), not omitted from the denominator. **All-criteria abstain:** if every applicable atom returns `abstain`, the case is flagged for human review and scored as 0, consistent with the full 2,728-case denominator; this occurs in $< 0.1\%$ of evaluated outputs. **Partial coverage:** models are evaluated on the full 2,728 cases; any case without a successful

generation is counted as score 0, so coverage differences are reflected in the score rather than hidden by subset averaging. **Per-criterion abstain:** each atomic criterion returns `pass`, `partial`, `fail`, or `abstain`. For required (reward) criteria, abstention contributes zero credit while its weight remains in the denominator; for protected (penalty) criteria, abstention triggers no deduction. If all required criteria abstain, $R = 0$; if all protected criteria abstain, $D = 0$. For the RQ2 diagnostic decomposition, dimension scores are computed over non-abstained criteria only, so that abstention does not artificially depress one regime relative to another. The same rule applies to every model, with each abstention retained in the criterion-level audit record.

Evaluation flow. The evaluator receives the source image, request, model output, and hidden transition contract. It first applies the quality gate, then judges each atomic criterion, and finally passes the structured verdicts to the deterministic scorer. Reference images are not required by the primary protocol.

B.2 VLM Evaluation with Specialized Visual Tools

SolveScore is a deterministic aggregation layer over visual evidence. The main evaluation primarily uses GPT-5.6 Sol (temperature 0), which receives the source image, edited output, request, and full contract and produces structured judgments with supporting evidence. Checker assignments are fixed for each atomic criterion before model evaluation and are shared across evaluated models. They target conditions for which VLM judgments are less reliable or a specialized tool provides a more reliable, objective check. For example, SAM and YOLO provide segmentation and detection evidence for assigned object and region conditions. Thus, checker use is determined by the criterion's fixed assignment, independently of the initial VLM verdict or self-reported confidence. Tool predictions supply evidence for criterion assessment; they are not assumed to be error-free. Table 5 summarizes the evidence families. The resulting atomic verdicts use the same pass/partial/fail/abstain schema and are aggregated by the same deterministic scorer.

Table 5: Evidence families for atomic criteria. Specialized checkers are preassigned to a fixed subset of criteria; the rest use VLM judgments.

Evidence family	Typical criteria	Evidence produced
OCR and parsers	Text, symbols, digits, labels, state updates	Recognized string/state and normalized comparison with text-region evidence
Grounding and segmentation	Entity presence, count, target zones	Instances, boxes/masks, count and containment relations
Geometry and topology	Position, alignment, routes, occupancy, assembly	Distances, overlaps, adjacency and graph/route relations
Preservation and similarity	Protected objects, layout, background, quality	Region-level change and similarity signals with artifact evidence
VLM contract audit	Scene interpretation, criterion assessment	Pass/partial/fail/abstain verdict, confidence and visible evidence

The contract defines what must be checked; VLM judgments and evidence from preassigned checkers support the assessment; and the deterministic scorer maps the resulting verdicts to contributions to R or D. For each criterion that has an assigned checker, both the checker and the VLM produce a verdict. If they agree, the shared verdict is used. If they disagree, the VLM is called once more with both the checker result and its own initial judgment, and the adjudicated verdict is used. For criteria without an assigned checker, the VLM returns a verdict, a self-reported confidence score in $[0, 1]$, and a natural-language reason with visible evidence; the verdict is used directly. Confidence scores and reasons are retained in the release artifact for audit, and the human evaluation in Appendix C validates overall VLM agreement against case-level review. Table 6 summarizes evidence-family annotations across the benchmark. These categories describe intended evidence, not measured tool invocation or successful-execution rates; criterion-level checker assignments are fixed as described above for each atomic criterion, independently of which model generated the output.

B.3 Diagnostic Dimensions and Contract Statistics

Semantic Accuracy (SA) covers entities, identity, count, attributes, text, symbols, and intrinsic state. Relational Accuracy (RA) covers position, order, containment, alignment, connectivity, topology, and occupancy. Visual Quality (VQ) covers rendering defects such as residue, blur, malformed local geometry, and illegible content. The canonical contract inventory contains 29,460 atomic criteria:

14,831 required and 14,629 protected; 14,322 are SA, 10,520 are RA, and 4,618 are VQ. The mean and median numbers of criteria per case are 10.8 and 10, respectively, and every case has at least one protected criterion, enabling preservation to be evaluated for every generated output.

B.4 Worked Score Calculation

Suppose a case has four equally weighted required atoms with verdicts pass, pass, partial, and fail, and two equally weighted protected atoms with verdicts pass and partial. Then $R = 0.625$, $D = 0.25$, and the primary score is $\max(0, 0.625 - 0.5 \times 0.25) = 0.50$, assuming the quality gate passes. This illustrates that a visually plausible output can receive a low score when it realizes the wrong transition or changes protected content.

B.5 Representative Case Cards

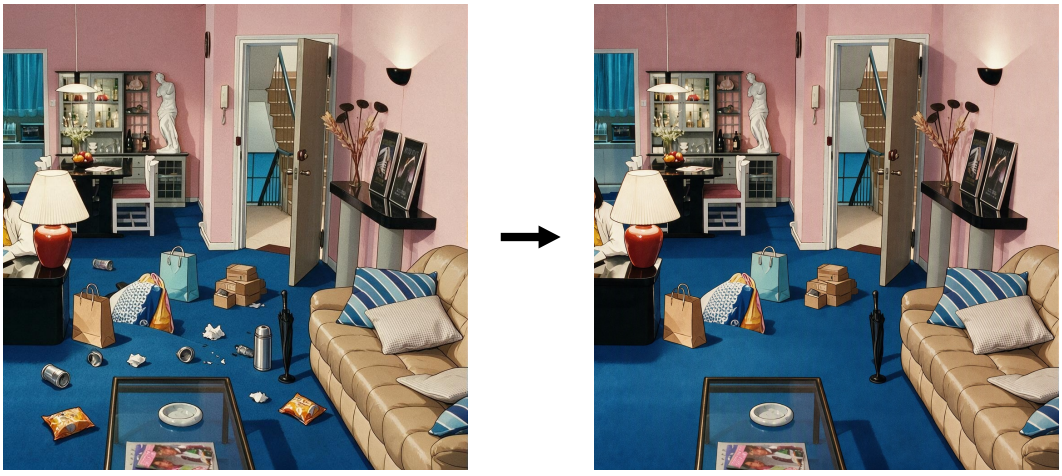
Each case below shows a representative subset of its atomic criteria; criterion weights are non-uniform (required SA/RA/VQ and protected SA/RA/VQ are normalized separately), so the displayed verdicts do not average directly to the reported R, D, and SolveScore.

Three representative cases illustrate the contract structure and evaluator output across information-dependence regimes. Each card lists the instruction, required and protected criteria, the GPT-Image-2 verdict for each criterion, and the resulting R, D, and SolveScore.

Table 6: Evidence-family annotations across the 29,460 atomic criteria in the 2,728-case benchmark. Counts describe intended evidence rather than measured tool execution counts. Checker assignments are fixed across evaluated models.

Evidence family	Criteria	%
VLM contract audit	24,034	81.6
Grounding and segmentation	1,768	6.0
Preservation and similarity	1,325	4.5
Geometry and topology	1,120	3.8
OCR and text parsing	1,213	4.1
Total	**29,460**	**100.0**

Case 1 (IS): Selective cleanup. *Instruction:* Remove all fallen debris, crushed cans, and crumpled trash from the blue carpet, but preserve all intact upright items and furniture. *Regime:* the request explicitly specifies which items to remove; the image only grounds the referenced categories.



Input

GPT-Image-2 output: R=100, D=12.5, SS=93.8

Criterion	Role	Dim	Verdict
All crushed cans on carpet removed	Req	SA	pass
All crumpled paper pieces removed	Req	SA	pass
Fallen snack bags removed	Req	SA	pass
Fallen spray can removed	Req	SA	pass
Carpet inpainted seamlessly	Req	VQ	pass
Upright bags preserved	Pro	SA	pass
Cardboard boxes preserved	Pro	SA	pass
Silver thermos preserved	Pro	SA	fail
Black umbrella preserved	Pro	SA	pass
Furniture preserved	Pro	SA	pass
Items on surfaces preserved	Pro	SA	pass

Case 2 (SD): Road bridge repair. *Instruction:* Repair the missing roads in the image. *Regime:* the correct bridge segment and its placement must be determined from the current scene state (matching cobblestone texture, path endpoints, and river gap).



Input



GPT-Image-2 output: R=25, D=66.7, SS=0.0

Criterion	Role	Dim	Verdict
Bridge segment placed over river gap	Req	RA	fail
Cobblestone texture matches existing path	Req	SA	fail
Segment aligns with both path ends	Req	RA	fail
Original segment removed from initial pos	Req	SA	pass
Decoy segments on right bank preserved	Pro	SA	fail
Decoy positions unchanged	Pro	RA	fail
Background elements preserved	Pro	SA	fail
No rendering artifacts	Pro	VQ	pass

Case 3 (RD): Storyboard connection repair. *Instruction:* Repair broken storyboard connections and align cue cards and tempo markers to correct panels based on sequence, character consistency, and the visible grid. *Regime:* the in-image grid and sequence rules determine which card belongs to which panel.



Input



GPT-Image-2 output: R=66.7, D=33.3, SS=50.0

Criterion	Role	Dim	Verdict
Red connection line gap closed	Req	RA	partial
'Sparky appears!' card to Panel 2	Req	RA	pass
'Hello there!' card to target panel	Req	RA	fail
'Andante' marker aligned to grid	Req	RA	pass
Floating flame marker aligned	Req	RA	pass
Storyboard panels intact	Pro	SA	fail
Existing connection lines preserved	Pro	RA	fail
Card text content unchanged	Pro	SA	pass
Text legible	Pro	VQ	pass
Panels not occluded	Pro	RA	pass

B.6 Sensitivity to the Preservation Weight

Figure 8 visualizes the sensitivity sweep, while Table 7 reports the corresponding values for $\lambda \in [0, 1]$ on the canonical 2,728-case cohort. Per-case scores are clamped at zero before averaging ($\max(0, R - \lambda D)$). The leaderboard ranking remains stable across all values (rank correlation ≥ 0.909). We fix $\lambda = 0.5$ before evaluating the leaderboard.

C Human Judge Alignment and Metric Validation

This study validates the computed SolveScore against independent human judgments.

C.1 Protocol

We sample 360 model-case pairs from the frozen 2,728-case manifest, stratified by model family, source type, dependency regime, and score bin. Each pair contains multiple atomic questions, yielding 3,832 atomic judgments. Two independent panels of four annotators each review the sampled pairs. Within each panel, annotators independently label every atomic criterion using the same pass/partial/fail/abstain rubric as the automatic evaluator; the panel reaches a consensus verdict per criterion by discussion. The question-level reference is the merged consensus across both panels. Separately, each panel judges whether the overall SolveScore for a case is reasonable, partly reasonable, or unreasonable; case-level disagreements between panels on this binary judgment (acceptable vs. unreasonable) are resolved by a third review. Each judge receives the source image, request, model output, and transition contract, but not the evaluator verdict or another panel's decision.

Table 7: SolveScore sensitivity to λ on the canonical 2,728-case cohort. Mean SolveScore is averaged across all 11 evaluated models with per-case zero-clamping. Rank correlation is computed against the primary $\lambda = 0.5$ ranking.

λ	Mean SS (%)	Rank corr.
0.00	39.5	0.909
0.25	34.4	0.973
0.50	29.8	1.000
0.75	26.1	0.973
1.00	23.0	0.945

Consensus merging. When the two panels agree on a question-level verdict, that label is taken as the reference; when they disagree, a third reviewer resolves the discrepancy. The reported 95.3% agreement rate compares this merged reference against the automatic evaluator's final tool-assisted verdict. **Abstain** refers to cases where the evaluator itself returned `abstain`; human judges rarely abstain.

Bootstrap. The 95% confidence interval on the case-level acceptable rate uses 1,000 case-level bootstrap replicates (percentile interval), resampling cases (not individual questions) to account for the within-case correlation across multiple atomic criteria per output.

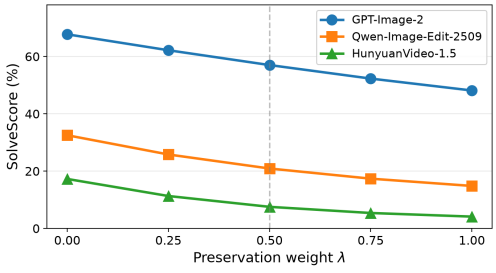


Figure 8: SolveScore as a function of the preservation weight λ for three representative generators on the 2,728-case cohort. The relative ordering is stable across the full range.

C.2 Human–Evaluator Agreement

Question-level agreement. The merged human consensus on each of the 3,832 atomic criteria is compared against the automatic evaluator's final tool-assisted verdict. The evaluator's verdict agrees with the human consensus on 95.3% of atomic questions, with 4.2% disagreement and 0.5% abstentions.

Table 8: Question-level human–evaluator agreement, overall and by diagnostic dimension.

Question-level metric	Value	Count
Evaluator agrees with human	95.3%	3,652/3,832
Evaluator disagrees with human	4.2%	161/3,832
Evaluator abstains	0.5%	19/3,832
Semantic accuracy (SA)	96.6%	1,449/1,500
Relational accuracy (RA)	91.8%	1,102/1,200
Visual quality (VQ)	97.3%	1,101/1,132

Case-level judgments. Human judges deemed the overall SOLVESCORE reasonable or partly reasonable on 93.6% of 360 reviewed pairs (337/360), with 6.4% (23/360) judged unreasonable. The two panels agree on this binary case-level judgment (acceptable vs. unreasonable) on 344/360 = 95.6% of pairs.

D MORE EXPERIMENTAL ANALYSIS

D.1 MODEL CONFIGURATION AND COVERAGE

All main results use the same 2,728 case identifiers. A failed generation is retried at most twice. Table 10 summarizes the generation configuration and coverage.

Table 9: Case-level human judgments on the 360 reviewed pairs.

Case-level metric	Value	Count
Reasonable + partly reasonable	93.6%	337/360
Unreasonable	6.4%	23/360
95% bootstrap CI on acceptable rate	[90.5, 96.4]	
Two-panel binary agreement	95.6%	344/360

No missing-generation indicator is used as a task score. API models were evaluated in September 2026 against the provider endpoints listed above (gpt-image-2, seedream-5.0-pro, gemini-3.1-flash-image, qwen-image-2.0-pro, kling-v3) with provider-default quality and aspect-ratio settings; open-weight models use the step counts and resolutions listed above. Video models are scored from their final frame. The evaluator is GPT-5.6 Sol (temperature 0, structured JSON output); all models are scored with the same evaluator version. Qwen-Image-Edit-2509 refers to the open-weight checkpoint `qwen2.5-image-edit`. Per-model quality-gate details are reported in the release artifact.

Table 10: Generation configuration and coverage on the 2,728-case benchmark.

Model	Access	Checkpoint / endpoint	Generated	Evaluated	QG pass	Evidence cov.
GPT-Image-2	API	gpt-image-2, 1024×1024	2,728	2,728	99.8%	99.1%
Seedream 5.0 Pro	API	seedream-5.0-pro, 1024×1024	2,728	2,728	99.8%	99.3%
Gemini 3.1 Flash Image	API	gemini-3.1-flash-image, 1024×1024	2,728	2,728	100.0%	98.8%
Qwen Image 2.0 Pro	API	qwen-image-2.0-pro, 1024×1024	2,728	2,728	99.3%	99.1%
Qwen-Image-Edit-2509	Open-weight	qwen2.5-image-edit, 50 steps, 1024×1024	2,728	2,728	90.8%	99.8%
OmniGen2	Open-weight	omnigen2, 50 steps, 1024×1024	2,728	2,728	84.9%	99.7%
BAGEL	Open-weight	bagel, 50 steps, 1024×1024	2,728	2,728	87.8%	99.6%
FLUX.1 Kontext [dev]	Open-weight	flux.1-dev-kontext, 28 steps, 1024×1024	2,728	2,728	98.7%	99.5%
FLUX.2 [dev]	Open-weight	flux.2-dev, 28 steps, 1024×1024	2,728	2,728	97.1%	99.7%
Kling V3	API	kling-v3, 3s video 720p, final frame	2,728	2,728	97.0%	99.0%
HunyuanVideo-1.5	Open-weight	hunyuan-video-1.5, 5s video, final frame	2,728	2,728	91.5%	99.2%

D.2 PER-DOMAIN AND DEPENDENCY RESULTS

Table 11 reports SOLVESCORE by content domain on the 2,728-case cohort. The three leading models remain ahead of the remaining systems in most domains, although Sci&Tech and Game are substantially harder for most models.

For the dependency analysis, we report both absolute scores and domain-adjusted gaps relative to IS. For model m, regime ℓ, diagnostic cell k, and domain d, let $S_{m,\ell,k,d}$ denote the corresponding score. Let $\mathcal{D}_{m,\ell,k}$ contain domains with defined cell means for both regime ℓ and IS. Let π_d be the fraction of total benchmark cases in domain d, fixed across regimes. We renormalize these weights over the shared support before averaging across the $M = 3$ models:

$$\Delta_{\ell,k} = \frac{1}{M} \sum_{m=1}^{M} \frac{\sum_{d \in \mathcal{D}_{m,\ell,k}} \pi_d \left(S_{m,\ell,k,d} - S_{m,\mathrm{IS},k,d} \right)}{\sum_{d \in \mathcal{D}_{m,\ell,k}} \pi_d}. \tag{10}$$

Table 11: SOLVESCORE (%) by content domain on the 2,728-case cohort.

Domain	N	GPT	Seedream	Gemini	Qwen2.0	FLUX.2	Qwen-Edit	FLUX.1	OmniGen2	BAGEL	Kling	Hunyuan
People	350	69.2	60.0	61.6	43.3	41.9	42.6	38.6	19.0	22.6	31.2	5.6
Architecture	332	57.1	56.3	57.1	42.8	30.2	24.4	21.7	10.9	14.4	29.4	4.2
Sci&Tech	290	52.8	55.5	55.2	27.1	9.6	7.6	9.2	4.1	7.1	18.9	8.2
Nature	277	59.7	55.4	55.9	53.2	22.3	21.1	18.3	8.4	10.9	30.9	4.2
Geography	281	56.4	53.8	51.7	52.7	22.4	21.1	15.1	7.6	9.9	30.7	0.0
Art	357	58.6	61.8	59.9	48.2	17.5	17.1	16.0	9.2	12.1	24.6	8.9
Object	336	53.6	54.8	51.2	35.0	17.5	14.8	14.1	10.3	9.2	31.3	15.2
Game	202	38.8	47.5	42.0	6.5	9.3	6.8	8.2	3.7	5.4	21.7	18.5
Sports	150	49.7	49.9	54.5	43.3	13.5	12.1	11.7	3.8	6.4	16.4	0.0
General	153	67.8	69.5	63.0	47.4	40.7	36.1	29.4	16.2	19.5	36.0	8.1
Overall	2,728	57.0	56.6	55.5	40.6	22.8	20.8	18.7	9.7	12.1	27.5	7.4

When all ten domains have defined means, the denominator is one. The figure reports point estimates. The regimes are information-source labels and are not interpreted as an intrinsic difficulty ordering. The 2,728 cases split as IS $= 736$, SD $= 629$, and RD $= 1{,}363$. Per-regime SolveScore for the top models is reported in Figure 5 of the main paper.

D.3 ROBUSTNESS AND FAILURE ANALYSIS

We report leaderboard sensitivity to $\lambda \in \{0, 0.25, 0.5, 0.75, 1\}$ in Section B.6. Table 12 groups observable failures by atomic criterion: required semantic (SA), required relational (RA), required visual-quality (VQ), protected semantic/relational, protected visual-quality, and quality-gate rejection. A case is counted in a category if at least one atomic criterion of that type fails. These are observable failure attributions from the contract evaluator and do not claim access to a model's internal reasoning process. Required relational failures are the most common category across all three top models, consistent with the RQ2 finding that transition determination is a substantial contributor to the difficulty.

Table 12: Failure taxonomy (% of cases with at least one failed criterion in each category) for the three highest-scoring models on the 2,728-case cohort. Categories are not mutually exclusive; a case may fail in multiple categories.

Failure category	GPT-Image-2	Seedream 5.0 Pro	Gemini 3.1 Flash
Required semantic (SA)	60.7	59.6	63.3
Required relational (RA)	69.3	68.5	71.6
Required visual quality (VQ)	29.4	28.8	31.1
Protected semantic/relational	69.1	53.1	66.6
Protected visual quality	13.0	11.6	15.8
Quality-gate rejection	0.2	0.2	0.0

E MORE DETAILS OF SOLVEEDIT-PLAN

Table 13: Matched one-generation evaluation of SOLVEEDIT-PLAN and controls (%).

Generator	Instruction method	$R \uparrow$	$D \downarrow$	SOLVESCORE ↑	Δ
GPT-Image-2	Direct	67.6	23.9	57.0	–
	SOLVEEDIT-PLAN	75.8	8.6	71.6	+14.6
Qwen-Image-Edit-2509	Direct	32.7	47.5	20.8	–
	SOLVEEDIT-PLAN	40.5	38.5	26.9	+6.1
HunyuanVideo-1.5	Direct	17.3	55.5	7.4	–
	SOLVEEDIT-PLAN	23.7	39.7	14.0	+6.6
GPT-Image-2 controls and ablations					
	Text-only Rewrite	67.9	23.2	57.3	+0.3
	GENERIC VISION REWRITE	72.8	16.3	65.3	+8.3
	w/o Preservation	68.0	18.1	63.0	+6.0
	w/o Contrast	69.1	11.5	63.8	+6.8
	SOLVEEDIT-PLAN	75.8	8.6	71.6	+14.6

SOLVEEDIT-PLAN receives only the source image and task request. It cannot access the evaluation contract, reference output, authored regions, manual facts, dependency label, or evaluator response. Inspect identifies unresolved variables and crop queries; Resolve compares candidate transitions and compiles one instruction for one final generator call. Direct, Text-only Rewrite, GENERIC VISION REWRITE, ablations, and SOLVEEDIT-PLAN use the same final generation budget (one generator call) in the matched comparison. The planning budgets are: Direct (0 agent calls), Text-only Rewrite (1 text-only call), and GENERIC VISION REWRITE, the ablations, and SOLVEEDIT-PLAN (2 vision-enabled agent

calls). SOLVEEDIT-PLAN uses one call for Inspect and one for Resolve, which internally performs candidate comparison and contrast checks.

Planner configuration. The released implementation uses GPT-5.6 Sol (API identifier `gpt-5.6-sol`) with temperature 0. All planner and control calls share the same output-token limit of 5,000 and the same backend. Inspect receives the source image; Resolve receives it together with at most six valid crops requested by Inspect using normalized bounding boxes. GENERIC VISION REWRITE uses the same model, temperature, and per-call token limit, with the source image supplied to both calls and no additional crops. Methods differ in call count and task design: Direct (0), Text-only Rewrite (1 text-only call), and GENERIC VISION REWRITE, ablations, and SOLVEEDIT-PLAN (2 vision-enabled calls each; SOLVEEDIT-PLAN additionally uses up to six inspected crops).

The appendix reports the main results, regime-wise gains, planner failure rate, and token use. It also reports representative cases in which planning resolves scene-dependent transitions. The release artifact contains the full prompts, schemas, crops, and intermediate records for inspecting individual planning decisions and the visual evidence behind each selected transition.

Table 14: SOLVEEDIT-PLAN relative gain over Direct by information-dependence regime on GPT-Image-2 (2,728-case benchmark).

Regime	Relative gain	Description
IS	$+12.0\%$	Transition specified by the grounded request
SD	$+23.3\%$	Missing variable determined by the current scene
RD	$+38.0\%$	Valid arrangement derived from an in-image rule

On the 2,728-case benchmark, SOLVEEDIT-PLAN improves GPT-Image-2 SOLVESCORE by $+14.6$ points, winning roughly two-thirds of cases whose scores change (68% win rate excluding ties). The planner compiles a valid instruction for every case (0% failure rate). Each case requires two LLM calls (Inspect and Resolve), averaging 3,207 input tokens and 635 output tokens (3,842 total) with mean end-to-end latency of 20.9 s. At official GPT-5.6 Sol list pricing (short-context rates of \$4.00/M input and \$20.00/M output), this costs approximately \$0.026 per case. No cached-input tokens were recorded. These figures cover Inspect and Resolve before the single final generator call.

Table 14 reports the relative gain of SOLVEEDIT-PLAN over Direct by information-dependence regime on GPT-Image-2. Gains increase monotonically from IS to RD, consistent with the RQ2 diagnosis that scene- and rule-dependent transitions present a substantial difficulty.

F LIMITATIONS

We note three limitations of this work. First, the failure attribution in our diagnosis is not fully causal. SOLVEEDIT locates the output condition at which an edit fails, but it cannot tell whether the failure came from perception, transition determination, or rendering. A model may fail because it did not see a small or occluded object, did not work out the required transition, or could not render a transition it had worked out. The regime axis and the six diagnostic cells show where the failure appears, not which internal mechanism caused it. The RQ2 findings should therefore be read as descriptive evidence about output conditions, not as causal claims about model internals.

Second, the automatic evaluation is validated on only a finite sample. The evaluator is a multimodal judge augmented by specialized checkers, and we compare it with human consensus on a stratified sample of 360 model–case pairs, or 3,832 atomic judgments, rather than the full manifest. The 95.3% question-level agreement is thus an estimate with sampling uncertainty, and the evaluator may still fail on conditions that are rare in the sample or under occlusion, strong stylization, and large generative changes. We report criterion coverage and abstentions separately so that conditions without a verdict remain visible in the audit record rather than being treated as evidence of either success or failure.

Third, video outputs are scored only from their terminal frames. The video results therefore test whether the same transition contracts transfer across modalities, but they say nothing about temporal coherence, motion validity, or temporal reasoning, all of which we leave outside this work.



Figure 9: Multi-model comparisons (rows 1–2). Columns: input, GPT-Image-2, Seedream 5.0 Pro, Gemini 3.1 Flash Image. Row 1 (RD, retail text-symbol): scores 100.0, 0.0, 87.5. Row 2 (SD, data-chart object-zone): scores 0.0, 100.0, 0.0. Models can disagree completely on the same request; visual coherence alone does not establish a valid transition.



Figure 10: Multi-model comparisons (rows 3–4). Row 3 (RD, map status-update): scores 0.0, 16.7, 100.0. Row 4 (IS, poster cleanup): scores 50.0, 100.0, 0.0. Each model succeeds on cases where others fail, indicating that no single editor dominates across information-dependence regimes.

G Exhibition Board

Following the exhibition-board style of prior visual benchmarks, this section collects model outputs at the end of the appendix. Captions identify the dependency regime and per-model SolveScore. The gallery is not used to select or alter quantitative results.

G.1 Multi-Model Comparisons

Figures 9 and 10 show representative cases with divergent model outcomes. Each row shows one case with input, GPT-Image-2, Seedream 5.0 Pro, and Gemini 3.1 Flash Image outputs. Cases are selected from the full 2,728-case benchmark to illustrate large inter-model disagreement.

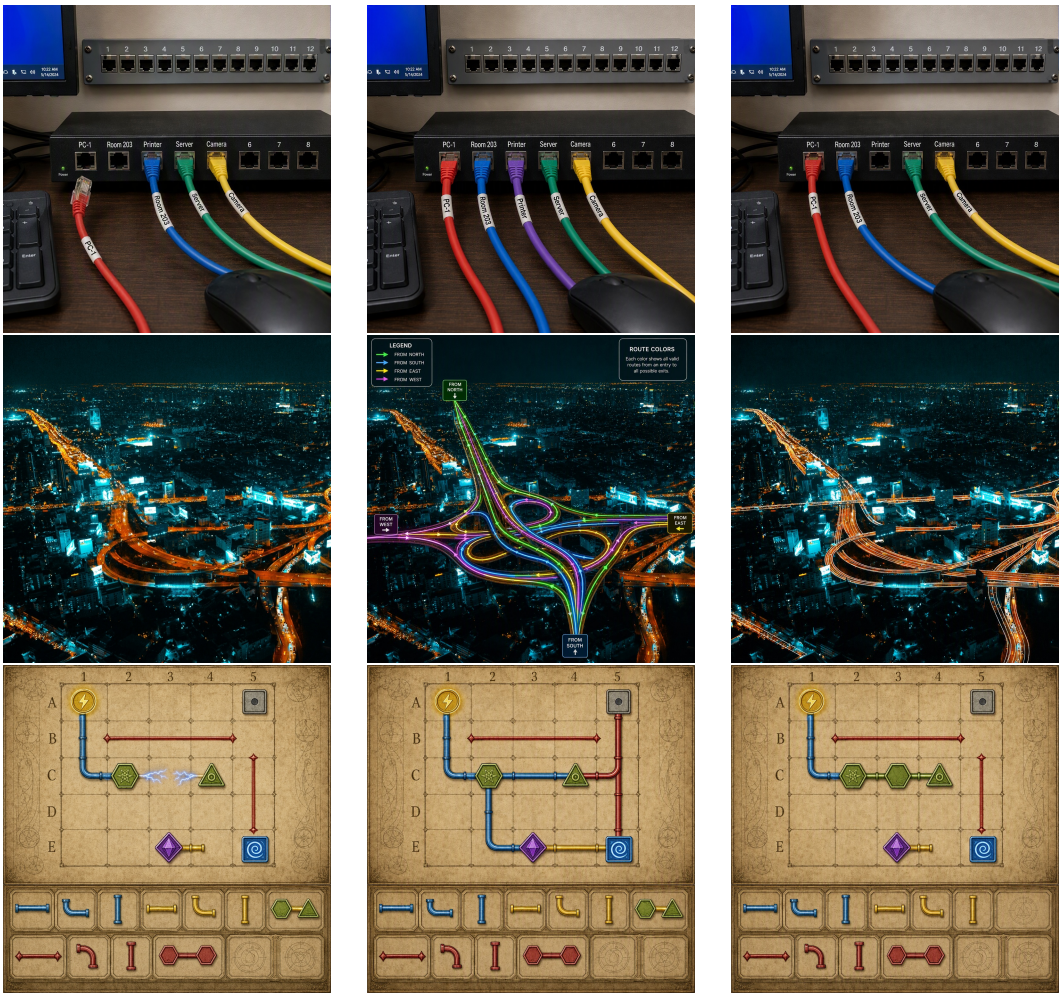


Figure 11: Direct versus SOLVEEDIT-PLAN on GPT-Image-2. Columns: input, Direct, SOLVEEDIT-PLAN. Row 1 (SD, architecture connection-repair): 75.0 → 100.0. Row 2 (SD, navigation route-path): 33.3 → 100.0. Row 3 (RD, maze connection-repair): 25.0 → 66.7. Explicit transition compilation corrects target selection, destination choice, and rule application that Direct editing misses.

G.2 DIRECT VERSUS SOLVEEDIT-PLAN

Figure 11 compares GPT-Image-2 Direct and SOLVEEDIT-PLAN on three cases where explicit transition compilation yields large gains. Each row shows input, Direct output, and Plan output.