GREP: GROUNDING GENERATED ROOM EVALUATION IN PRINCIPLES

Mohan Cao[1,2], **Yongsen Mao**[3], **Chucheng Xiang**[2,4], **Zhongyuan Liu**[2], **Zhihang Zhong**[5], **Haiping Wang**[3*] **& Yuan Liu**[3*]
[1]University of Washington [2]Tencent AI Lab
[3]The Hong Kong University of Science and Technology
[4]University of Science and Technology of China [5]Shanghai Jiao Tong University

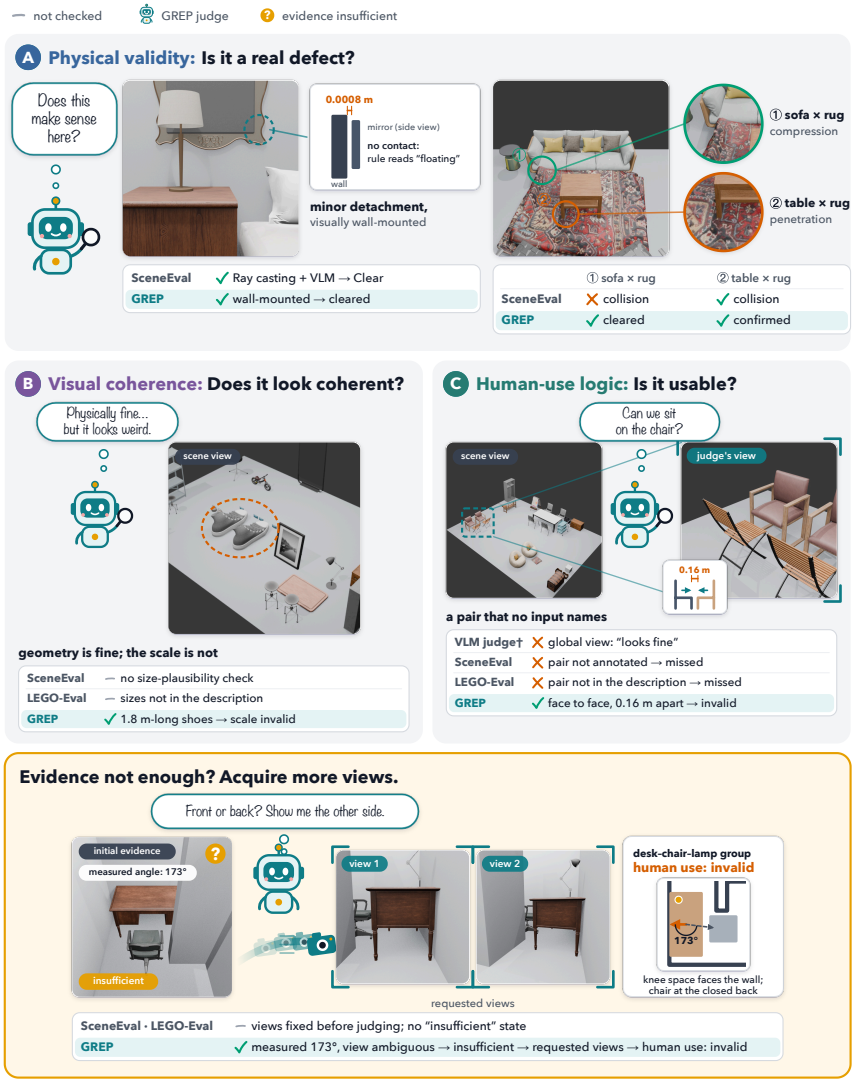


Figure 1: GREP is a scoring framework for generated scenes, grounded in three progressive *principles*: (A) physical validity, (B) visual coherence, and (C) human-use logic. In contrast, existing metrics rely on narrow criteria or hard metrics, failing to assess the scene quality accurately. Moreover, GREP lets the evaluation agent *autonomously* identify issues, gather evidence, and reach comprehensive judgments. Check and cross marks indicate whether each evaluator's output matches the actual state of the scene, not whether the scene passes. In (A), a pure contact rule would read the mirror 0.0008 m from the wall as floating; both VLM-assisted evaluators clear it, whereas only GREP separates the intended sofa–rug contact from the table–rug penetration. †A holistic VLM judge given only the global view.

*Corresponding authors.

ABSTRACT

Recent LLM-centric advances can generate attractive furnished rooms by selecting and arranging assets. Yet evaluation remains limited to specific perspectives or predefined questions, failing to reflect overall scene quality. To address this pressing need, we introduce **GREP**. Its core is a focus shift from defining specific questions to establishing a three-tier evaluation philosophy: basic *physical validity* checks whether a scene obeys physical constraints; *visual coherence* concerns compatibility in object scale, style, and composition; and, at the highest level, *human-use logic* requires object arrangements to align with everyday human habits. Guided by these principles, an evaluation agent autonomously and adaptively identifies issues from scene context, gathers evidence, and produces scores and reports. This supports scene-specific issue discovery and reliable judgments. Experiments show that (1) GREP achieves 87% agreement with human majority preferences, substantially outperforming baseline evaluators. (2) GREP reports improve scene quality through refinement, whereas score-only and no-feedback baselines show slight declines. (3) GREP suggests that as planning ability improves, recent models such as GPT-5.6 Sol can move from relying on pipelines and harnesses to challenging their guidance and constraints, but still lack the ability to resolve the resulting conflicts. Stronger models are therefore expected not only to plan well, but also to adapt to external guidance and constraints more flexibly. Project page: `han-808.github.io/GREP`.

1 INTRODUCTION

Given a room description, recent generation systems use language models (Feng et al., 2023; Ran et al., 2025), specialized pipelines (Yang et al., 2024b; Sun et al., 2025), or harnessed agents (Yang et al., 2025) to select three-dimensional (3D) assets and arrange them into a furnished scene. The output specifies each object's size, position, and orientation. Evaluation must answer a broader question: does this arrangement make sense as a room?

As generated scenes grow more diverse, evaluation has not kept pace. Existing methods assess particular aspects of scene quality, including instruction-derived constraints (Kang et al., 2026), text fidelity and physical plausibility (Tam et al., 2026), or activity-specific functional feasibility (Maillard et al., 2026). Scene-specific problems outside these evaluation scopes may therefore be overlooked, limiting assessment of the scene as a whole from a human perspective (Fig. 1).

We introduce **GREP** (**G**rounding generated **R**oom **E**valuation in **P**rinciples), an evaluation framework that addresses this gap. Our core insight lies in *establishing a principled three-tier evaluation philosophy, rather than in defining specific checks.* First, at the most basic level, a scene must follow *physical validity*: objects must have adequate support and avoid unintended collisions or interpenetration. Next, *visual coherence* requires objects to have compatible styles and sensible relative scales. Finally, at the highest level, real-world usability requires obeying *human-use logic*, spanning open problems; for example, facing chairs need room for people to sit, and a desk chair should face the desk's usable side, as illustrated in Fig. 1.

This philosophy provides consistent principles, while the specific questions to ask vary from scene to scene. Our second contribution puts it into practice through *scene-specific inspection with adaptive evidence collection*. An evaluation agent uses room context to identify objects and relations that warrant inspection, including those absent from the input description. It then autonomously uses geometric measurement, existing-view selection, and additional rendering to gather the evidence each check requires. Finally, GREP aggregates these judgments into metric and scene scores, but its output goes beyond numbers. A detailed report identifies affected objects, explains the judgments, and records supporting evidence and unresolved gaps. Encouragingly, we further show that *such diagnostic reports help generators largely improve scene quality* by pinpointing specific problems and guiding targeted revisions.

Our experiments have two parts. First, we use GREP to evaluate current scene generation approaches. The results suggest that, as planning improves, models such as GPT-5.6 Sol can move from relying on pipelines and harnesses to challenging their guidance and constraints. Yet they still struggle to resolve

the resulting conflicts, compromising scene quality. This calls for future models and systems that combine strong planning with the flexibility to adapt to external guidance and constraints. Second, we assess GREP itself. It achieves 87% agreement with human preferences, substantially outperforming baseline metrics. In a five-scene study with GPT-4.1, report-guided refinement increases the mean score from a shared initial value of 0.585 to 0.676 after three revision rounds, compared with 0.543 without feedback, using the refinement scoring protocol described in Appendix E.1.

2 Related Work

LLM-driven Scene Generation. Earlier learned generators synthesize layouts from data priors with autoregressive or diffusion models (Paschalidou et al., 2021; Tang et al., 2024; Lin & Mu, 2024), and PhyScene adds physical interactability (Yang et al., 2024a). Language-guided scene generation systems differ in representation and control. LayoutGPT and DirectLayout use language-model reasoning to predict numerical placements (Feng et al., 2023; Ran et al., 2025); Holodeck and LayoutVLM incorporate spatial constraints or optimization into scene construction (Yang et al., 2024b; Sun et al., 2025); SceneWeaver coordinates scene-generation tools through an iterative agent (Yang et al., 2025); and recent layout agents and co-optimization methods extend these designs (Wan et al., 2026; Xiang et al., 2026). This diversity calls for evaluation independent of the generation mechanism. GREP accepts scenes, assets, transforms, and spatial context through a common interface.

Structured Scene Evaluation. Structured evaluators decompose tasks into explicit requirements or verification programs. Davidsonian Scene Graph (DSG) uses atomic questions and dependencies to assess text-to-image faithfulness (Cho et al., 2024); weak-verifier approaches generate task-specific programs for spatial layouts (Zhang et al., 2026); and SceneEval, LEGO-Eval, and SceneTeract verify physical, instructional, or activity-specific properties in generated 3D scenes (Tam et al., 2026; Kang et al., 2026; Maillard et al., 2026). These methods provide valuable checks, but their check vocabularies are primarily fixed by input requirements, annotations, or supplied activities. Model-based judges instead score outputs directly, from reference-free and question-answering metrics (Hessel et al., 2021; Hu et al., 2023) to LLM and VLM judges (Zheng et al., 2023; Lee et al., 2024); GREP also uses a VLM, but conditions each judgment on a scene-specific check and its retained evidence. GREP organizes evaluation around physical validity, visual coherence, and human-use logic. This three-tier philosophy guides scene-specific checks, including relations that emerge from the generated arrangement.

Evidence-Grounded Visual Reasoning. Task-directed observation selection reveals information missing from a fixed view. AffordBot, Agentic Affordance Reasoning (A3R), and Chain-of-View (CoV) acquire complementary visual or geometric evidence for affordance and embodied reasoning tasks (Wang et al., 2025; Li et al., 2026; Zhao et al., 2026). GREP applies this principle to scene-specific inspection: each check gathers the measurements and views needed for judgment, and each verdict retains its supporting evidence. Adaptive evidence collection implements the inspection guided by our evaluation philosophy.

3 GREP Evaluation Framework

GREP evaluates generated rooms from their assets, positions, orientations, scales, and available boundaries. We first outline the *evaluation workflow*, from scene analysis to evidence collection and reporting. We then describe the *evaluation dimensions*, which make the three principles operational through eight metrics, the *evidence operations* that support adaptive inspection, and the resulting *scores and reports*.

3.1 GREP Workflow

Figure 2 summarizes the evaluation in three stages: formulating checks, inspecting and judging them, and scoring with diagnoses; the evaluation procedure depends on the metric (Appendix A). (1) *Step 1: Analyze the room to raise questions.* The evaluation agent uses a vision-language model (VLM) to inspect room-wide views, object information, and intended use. It identifies objects and relationships requiring inspection under the three principles and formulates scene-specific *checks*. Screening resolves conclusive cases at this stage: geometric predicates settle clear physical cases, and

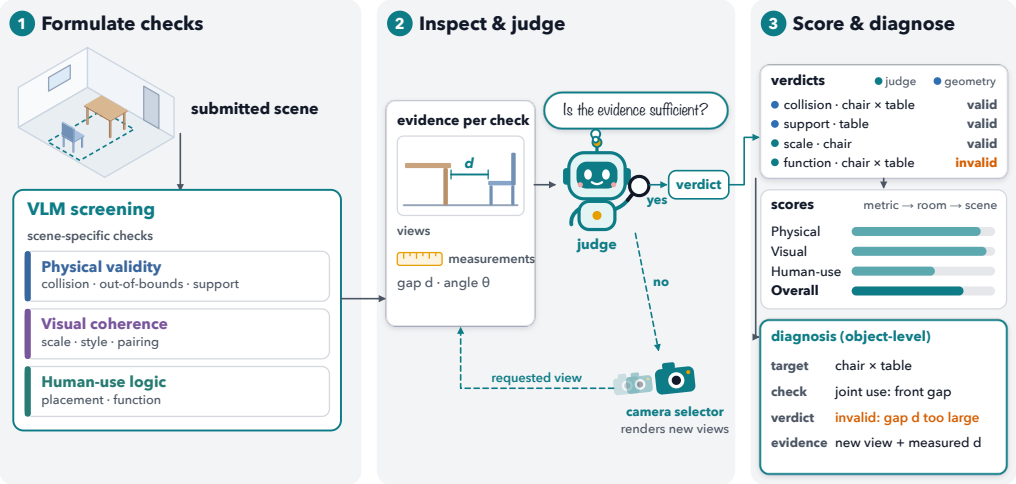


Figure 2: GREP evaluation framework. (1) Analyze the room globally to raise questions under the three principles. (2) Adaptively gather evidence to resolve remaining checks. (3) Aggregate scores and compile a diagnostic report.

VLM screens clear evidently valid Scale and Pairing cases (Appendix A). (2) *Step 2: Complete the remaining checks.* For each unresolved check, the agent gathers relevant evidence for judgment. It records a verdict with its reasons and supporting evidence. (3) *Step 3: Aggregate scores and compile the report.* GREP aggregates the judgments into metric, room, and scene scores (Section 3.4). The accompanying report records affected objects, reasons, evidence, and unresolved gaps, providing specific guidance for scene revision.

3.2 Evaluation Dimensions

The three principles provide conceptual guidance for evaluation. Eight metrics make these principles operational. Each metric specifies a quality dimension; each check applies it to particular objects or relationships, and each check type specifies the property to inspect.

Checks for the physical validity. A room must first obey physical constraints. *Collision* checks for unintended intersections, distinguishing them from intended attachments. *Out-of-bounds* checks whether objects cross the supplied spatial boundary. *Support* checks whether objects have physical support. Together, these checks assess basic physical validity; they do not determine whether the arrangement looks coherent or supports its intended use.

Checks for the visual coherence. Physically valid objects can still form a visually inconsistent room. *Scale* checks whether sizes are reasonable for the object categories, *Style* checks visual compatibility, and *Pairing* checks whether object choices are compatible or unnecessarily repetitive. These checks assess how the selected assets fit together in size, style, and composition.

Checks for the human-use logic. A visually coherent room must also make sense for human use. *Placement* asks whether people would place an object in that location, considering its properties and scene context. *Function* then asks whether objects can be used as intended in that arrangement, checking access, orientation, and spacing for individual objects and objects used together.

3.3 Evidence Operations

Guided by the check types above, the evaluation agent formulates scene-specific questions and gathers evidence to answer them. We allow three types of operations that the agent selects and combines as needed. (1) *Measure spatial relations.* Compute distances, relative angles, overlap, and clearance for the target objects from their geometry and poses. (2) *Retrieve existing views.* Each check starts from the room-wide views and the views already acquired for its scope; Function checks in the same object group share one evidence bank. This tool returns views of the target objects and their surroundings without additional rendering. (3) *Render additional views.* When existing views do not reveal a

required feature, acquire another view by changing the camera while keeping the scene fixed. The evaluator checks whether the image supplies the missing evidence, and the view is kept for later checks in the same scope.

These tools support all three principles. If geometry cannot settle a physical check, for example, the agent can inspect a view of the relevant region or request a new one if the region is hidden. Visual coherence and human-use checks likewise obtain evidence as needed. Evidence collection ends when it is sufficient or the acquisition budget is exhausted; unresolved gaps remain in the report. Appendix A details measurement procedures, camera selection, and budget controls.

3.4 Scores and Reports

Scores. GREP computes scores from confirmed defects identified by its checks. For each metric, repeated observations of the same defect are deduplicated into defect events. Each event receives a severity-based penalty, which is attributed to the responsible objects. The metric score combines the total object penalty B_m, normalized by the room's object count N, with the largest event penalty P_m:

$$S_m = 100(1 - \alpha P_m)(1 - B_m/N)^{\gamma}, \qquad \alpha = 0.4, \quad \gamma = 4.$$

Higher scores indicate fewer or less severe defects. Appendix B specifies the event penalties, object attribution, and aggregation rules.

Metric scores are combined using predefined nominal weights to obtain the room score. We also report principle scores as weight-normalized averages of the metrics belonging to physical validity, visual coherence, and human-use logic. The FrozenAssets* configuration excludes visual coherence metrics from the overall score and renormalizes the remaining weights. For multi-room scenes, room scores are aggregated using net floor area as the weight. Model and method results then average the included scene scores equally.

Reports. The report pairs these score summaries with the underlying inspection records. Each record corresponds to a check, identifying the check type, target objects or relations, verdict, reason, and supporting measurements or views. Evidence gaps and decision sources thus remain explicit. This links the overall assessment to specific issues and provides the generator with context for scene revision, as we will show by experiments.

4 Experiments

We first use GREP and specially designed benchmarks to evaluate current scene generation methods, identify performance trends, and derive design insights. We then focus on GREP itself, examining its value for scene refinement and its reliability as an evaluator.

4.1 Benchmark Observations

4.1.1 Benchmark Protocol

GREP uses GPT-5.6 Sol as the evaluation agent. We organize baselines into three generation settings and describe their protocols and task suites below. We report scores for each principle and the overall scene score, abbreviating physical validity, visual coherence, and human-use logic as Physical, Visual, and Human-use, respectively. Further details appear in Appendices C.1 and C.2.

(1) **Model-native planning.** This setting focuses on native reasoning and planning abilities of language models. Given an open space or a floorplan with walls, a model first describes the objects it intends to include. An external matcher retrieves assets matching these descriptions from the Imaginarium asset library (Zhu et al., 2025). The model then outputs their scales, positions, and orientations in a single call. We evaluate GPT-6 Astra, GPT-5.6 Sol, Claude Opus 5, Claude Fable 5, Claude Sonnet 5, Kimi K3, GLM-5.3, and HY4 Preview. The suite contains 30 scenes: 20 open spaces and 10 floorplans with relatively simple rectangular wall layouts splitting the space into multiple rooms.

(2) **Specialized pipelines.** This setting evaluates specialized pipelines that use handcrafted rules to guide and adjust model-generated asset arrangements. We compare LayoutGPT (Feng et al.,

Table 1: Model-native results in open spaces (without walls) and floorplans with rectangular wall layouts. Bold marks the highest value in each row. HY4 Preview's with-wall average covers nine of the ten floorplans (Appendix C.3).

	GPT-6 Astra	GPT-5.6 Sol	Claude Opus 5	Claude Fable 5	Claude Sonnet 5	Kimi K3	GLM-5.3	HY4 Preview
Open space								
Overall	**84.5**	73.7	81.5	77.9	65.3	73.2	70.8	61.8
Physical	85.0	81.0	**90.3**	77.8	72.4	70.2	67.2	55.9
Visual	88.4	84.8	**95.3**	95.1	87.0	91.2	90.2	86.8
Human-use	**83.4**	66.4	72.4	74.8	56.0	72.0	69.7	61.4
With wall layouts								
Overall	**81.2**	73.4	69.5	72.5	65.7	66.6	68.7	54.2
Physical	**80.2**	73.8	79.4	79.2	73.5	75.4	72.0	54.0
Visual	86.9	89.9	94.0	91.5	**96.7**	92.9	89.5	87.5
Human-use	**80.9**	69.9	57.6	64.0	54.0	55.1	62.4	47.9

Table 2: Results of specialized pipelines across model backbones and asset placement methods. Visual coherence metrics are excluded under FrozenAssets*. "None" means raising the asset arrangement plans with the native planning abilities of models without requiring a specialized pipeline. Bold marks the highest value in each row.

	None	LayoutGPT	LayoutVLM	DirectLayout	SceneWeaver
GPT-5.6 Sol backbone					
Overall	**78.3**	74.3	73.6	68.5	62.1
Physical	**83.4**	81.1	81.3	81.3	77.5
Human-use	**74.5**	69.4	68.0	59.2	50.9
GPT-4.1 backbone					
Overall	40.2	31.0	**54.5**	44.1	40.0
Physical	36.6	19.0	**57.9**	38.1	29.9
Human-use	42.8	39.7	**52.1**	48.5	47.4

2023), DirectLayout (Ran et al., 2025), LayoutVLM (Sun et al., 2025), and SceneWeaver (Yang et al., 2025) against direct placement (*None*), using both the stronger GPT-5.6 Sol and the weaker GPT-4.1 as backbones. The suite contains ten open-space scenes. Following common practice, we use FrozenAssets*, providing each scene with a fixed set of preselected assets that every method must use and place. Visual coherence checks (Scale, Style, and Pairing) are excluded from Overall because asset choices and scales are predetermined.

(3) **Harnessed agents.** This setting evaluates language models within a shared agent harness. We use GPT-5.6 Sol, Kimi K3, GLM-5.3, and HY4 Preview as backbones. Models have access to information about the full Imaginarium database (2,043 3D assets across 385 categories) and callable tools: `get-task`, `search-assets`, `inspect-asset`, `validate-submission`, and `finalize-submission`. The harness supports tool use, self-correction, and iterative refinement without exposing GREP scores. However, the harness also imposes *strict constraints*. Its suite contains ten scenes with irregular wall boundaries, predefined room uses, and prescribed asset counts, which may restrict the model's planning flexibility. For evaluating these advanced systems, we use ten highly complex floorplans with irregular wall layouts.

4.1.2 Benchmark Results

GREP supports three observations on key questions in scene generation: the impact of native ability of models, the effectiveness of specialized pipelines, and the value of agent harnesses.

(1) Model-native planning

Results. Stronger models, like GPT-5.6 Sol, already produce largely sensible layouts with relatively few conflicts, achieving an overall score above 70 in both settings (Table 1). Weaker models such as

Table 3: Results of harnessed agents under default requirements and a stricter condition requiring 20% more assets. Bold marks the highest value in each row.

	GPT-5.6 Sol	Kimi K3	GLM-5.3	HY4 Preview
Default constraints				
Overall	83.0	**88.5**	84.7	**88.5**
Physical	75.8	78.5	84.5	**89.6**
Visual	**98.8**	97.3	98.0	97.8
Human-use	85.2	**94.0**	82.3	86.0
20% more assets				
Overall	80.3	87.4	88.3	**89.3**
Physical	66.8	73.2	84.3	**89.4**
Visual	**98.8**	98.3	96.7	96.0
Human-use	86.6	**95.7**	89.7	87.9

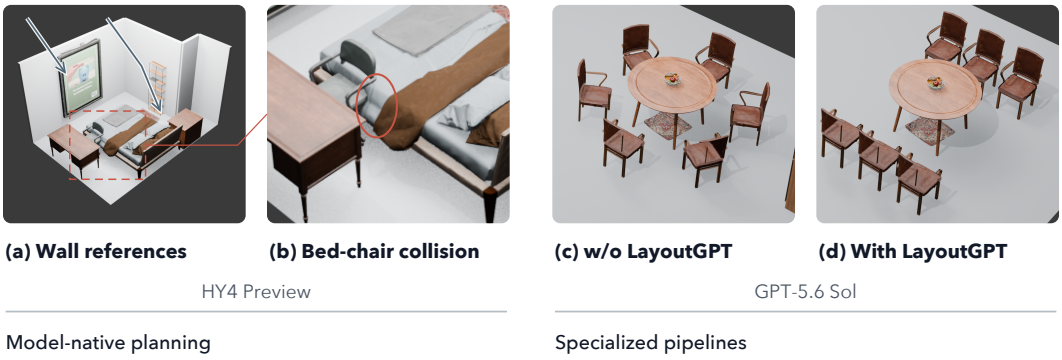


Figure 3: Qualitative examples of Model-native planning and pipeline-dependent generation. (a) HY4 Preview uses walls as placement references, (b) yet its layout contains a bed-chair collision, shown in the magnified view. With the same dining assets and GPT-5.6 Sol backbone, (c) Model-native arranges chairs around the round table, while (d) LayoutGPT places them in two straight rows.

HY4 Preview remain less capable despite using spatial cues from walls. Generation traces show HY4 using wall positions and inward normals to place and orient the picture and bookshelf (Figure 3(a)), yet its layout still contains a bed-chair collision (Figure 3(b)).

Insights. With few constraints, recent advances in planning abilities already enable stronger models to generate sensible room layouts.

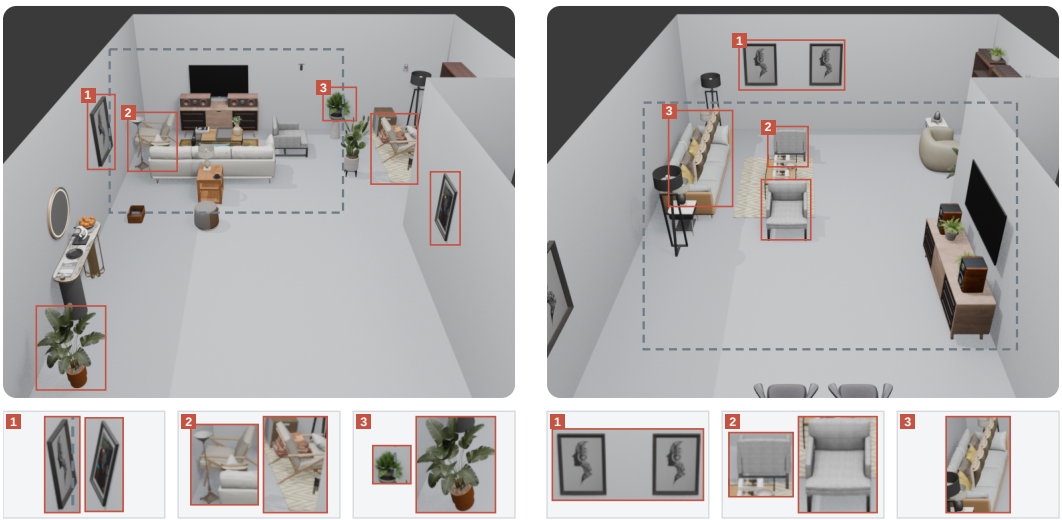
(2) Specialized pipelines

Results. However, strong planning does not ensure that a model can recognize unsuitable guidance. Handcrafted pipelines can improve weaker models' plans: LayoutVLM raises GPT-4.1's Overall from 40.2 to 54.5 (Table 2). For GPT-5.6 Sol, such guidance may instead disrupt an otherwise sensible plan. Figure 3(c–d) shows Sol arranging chairs around a round table on its own, but placing them in two straight rows under LayoutGPT. All four pipelines lower Sol's Overall by 4.0–16.2 points, suggesting that the model does not adequately assess whether their guidance suits the scene.

Insights. Better planning alone is not enough. Models still *cannot effectively assess the suitability of external guidance*, which undermines the coherence of the overall scene.

(3) Harnessed agents

Results. The agent setting reveals a striking ranking reversal: HY4 Preview outperforms GPT-5.6 Sol, reversing their order in the Model-native setting (Tables 1 and 3). One possible explanation is that harness compensates for weaker planning, while requirements such as mandatory asset counts interfere with stronger models' own plans. To probe sensitivity to these requirements, we increase the required asset count by 20% within the agent suite. HY4's Overall improves slightly, whereas Sol's deficit relative to HY4 widens from 5.5 to 9.0 points and its Physical falls from 75.8 to 66.8,



(a) HY4 Preview

Diverse assets

(b) GPT-5.6 Sol

Repeated assets

Figure 4: Qualitative examples under the agent harness. (a) HY4 Preview uses a more diverse set of assets. (b) GPT-5.6 Sol repeats several assets, including wall pictures, chairs, and sofa cushions. Numbered boxes mark the enlarged insets below each layout: (1) wall pictures, (2) chairs, and (3) plants in (a) or sofa cushions in (b).

revealing the strong harness hurts the strong model performances. Figure 4 illustrates differences in asset selection: HY4 Preview uses a more diverse set of assets, whereas GPT-5.6 Sol uses more repeated assets.

Insights. Stronger planning now comes with *poorer adaptation to external constraints*: models may adhere to existing plans rather than effectively revise them to meet new requirements.

Summary

These observations suggest a progression from *reliance*, through *conflict*, toward *integration*. Weaker models can benefit from external guidance that compensates for limited planning. Stronger models can form better plans of their own, yet struggle to reconcile them with supplied guidance and constraints. The next stage should unite these strengths: models should judge when guidance is useful, adapt it to the scene, and incorporate constraints into a coherent overall plan. This calls for training and system design that develop planning ability and flexibility together.

4.2 Evaluating GREP

We examine three aspects of GREP itself: the value of its diagnostic reports, its agreement with human judgments, and its repeatability. Further details and analyses are provided in the appendix.

4.2.1 Diagnostic Feedback for Revision

We show that GREP's diagnostic reports provide more useful guidance for scene revision than scores alone. Specifically, we test their value on five scenes using GPT-4.1. Each branch revises the generated scene three times. At each revision, the model receives the current scene and GREP's scene score or report as context. We have the following observations.

(1) Reports improve revision beyond scores. After three revisions, the Report branch raises the mean score from 0.585 to 0.676, while score-only and no-feedback baselines show slight declines (Figure 5(a)). This study uses the refinement-specific scoring protocol specified in Appendix E.1.

(2) Reports prompt concrete changes. Across the three revision rounds and five scenes, the diagnostic branch changes the layout in all 15 revisions, compared with 5/15 for scores alone and 6/15 without

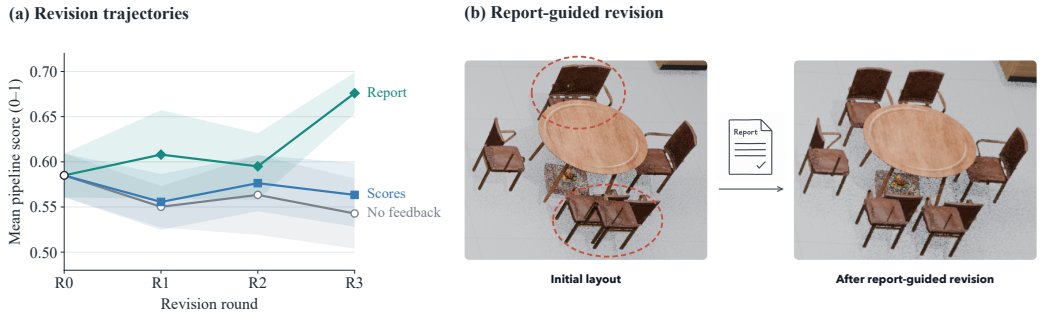
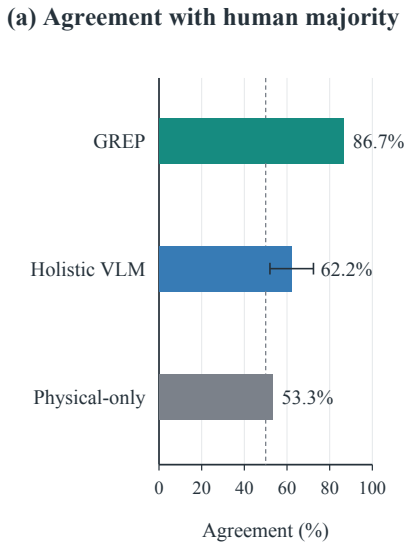
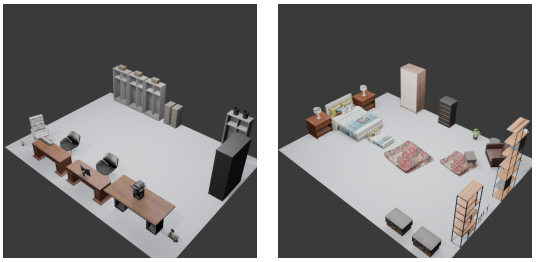


Figure 5: Report-guided scene refinement with GPT-4.1. (a) Mean score across five scenes from the shared initial layout (R0) through three revisions (R1–R3), using the 0–1 pipeline scoring protocol in Appendix E.1; shaded bands show ± 1 standard error across the five scenes. (b) The dining layout of S101 before and after report-guided revision: dashed circles mark overlapping chairs, which are separated in the revised layout. These views were rendered again from the saved layouts.



(b) P03: the baselines diverge from humans



A · Claude Opus 5 S112 | B · Claude Fable 5 S113

Evaluator	Preference
Human annotators	A · 5/5 votes
GREP	A
Holistic VLM	B · 3/3 runs
Physical-only	Tie · 100 vs 100

Figure 6: Agreement with human overall preferences across 15 pairs of six scenes. (a) Agreement with the human majority; the error bar shows the between-run standard deviation of the Holistic VLM, and the dashed line marks chance (50%). (b) Pair P03: all five annotators and GREP prefer A, all three Holistic VLM runs prefer B, and Physical-only assigns equal scores.

feedback. These results suggest that numerical scores provide little direction for revision, whereas diagnoses identify specific issues that the model can act on.

The example in Figure 5(b) shows a dining layout before and after report-guided revision. Several chairs overlap in the initial layout; the revised layout separates them around the table.

4.2.2 Agreement with Human Judgments

Figure 6 compares GREP, Holistic VLM judgments, and Physical-only scores with human overall preferences. GREP agrees with the human majority on 86.7% of pairs, compared with 62.2% for Holistic VLM averaged over three runs and 53.3% for Physical-only. Here, we reuse GREP's Physical score as the Physical-only baseline, as it already covers typical physical checks. Appendices D.2 and D.3 provide the study procedure and baseline details.

4.2.3 REPEATABILITY

We evaluated 12 selected open-space scenes five times each with fixed inputs, configuration, and judge. The per-scene standard deviation of the overall score has a median of 2.9 points and a maximum of 7.3. Differences between scenes are much larger: repeat-to-repeat variation accounts for 8% of the score variance across these scenes (Appendix D.1). A smaller-scale analysis with a second judge, Claude Fable 5, on six of these scenes preserves the ordering of the scenes (Spearman $\rho = 0.94$) but scores all six higher, by 4.8 points on average (paired t-test, $p = 0.12$, $n = 6$).

5 LIMITATIONS

Single judge model. All reported evaluations use GPT-5.6 Sol as the judge, and GPT-5.6 Sol is also one of the evaluated generators. LLM evaluators can recognize and favor their own generations (Panickssery et al., 2024). Our judge-sensitivity analysis with Claude Fable 5 covers only six scenes and cannot establish whether either judge favors its own model family (Appendix D.1).

Coverage of discovered checks. Discovery instantiates checks from fixed check types over the submitted objects. Judgment coverage refers to these instantiated checks; it does not show that discovery found every relevant object or relationship in a room (Appendix A.3).

Forced verdicts under a bounded budget. Evidence acquisition is bounded. Checks still unresolved when the budget is exhausted receive a forced binary verdict from the retained evidence, and on some scene-quality procedures non-hard failures resolve to a labeled default-valid outcome (Appendix A.5). Each record identifies these decision sources, but such verdicts rest on incomplete evidence and can be wrong.

6 CONCLUSION

GREP grounds generated room evaluation in three progressive principles: physical validity, visual coherence, and human-use logic. These principles guide autonomous inspection and adaptive evidence collection, yielding multidimensional scores and diagnostic reports. GREP achieves 87% agreement with human majority preferences. Under the refinement-specific scoring protocol, report feedback raises the mean score from 0.585 to 0.676 after three revisions, while score-only and no-feedback baselines show slight declines. Our benchmark observations suggest that stronger planning alone does not ensure effective use of guidance or constraints. This requires models and systems that combine strong autonomous planning with the flexibility to incorporate guidance and constraints into coherent scene plans.

AI USE STATEMENT

We used AI tools to assist with writing, literature search, research ideation, and execution. The authors reviewed and verified all content, analyses, and claims.

ETHICS STATEMENT

We identify no significant ethical risks specific to this work beyond those generally associated with generative AI and automated evaluation.

REPRODUCIBILITY STATEMENT

The paper and appendix provide the evaluation procedure, experimental settings, and implementation details needed to reproduce our results.

REFERENCES

Blender Online Community. Blender – a 3D modelling and rendering package. Blender Foundation, `https://www.blender.org`, 2026.

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 *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=ITq4ZRUT4a.

Weixi Feng, Wanrong Zhu, Tsu-Jui Fu, Varun Jampani, Arjun Akula, Xuehai He, Sugato Basu, Xin Eric Wang, and William Yang Wang. LayoutGPT: Compositional visual planning and generation with large language models. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2305.15393.

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. doi: 10.18653/v1/2021.emnlp-main.595.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 20406–20417, 2023.

Minseok Kang, Dongwook Choi, Gyeom Hwangbo, Seungwon Lim, Kai Tzu-iunn Ong, and Jinyoung Yeo. Apples on the table? Evaluating text-guided 3D scene synthesis via fine-grained constraint verification. In *Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing*, 2026. URL https://arxiv.org/abs/2511.03001.

Seongyun Lee, Seungone Kim, Sue Park, Geewook Kim, and Minjoon Seo. Prometheus-Vision: Vision-language model as a judge for fine-grained evaluation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 11286–11315, 2024. doi: 10.18653/v1/2024.findings-acl.672.

Di Li, Jie Feng, Guanbin Li, Ronghua Shang, Yuhui Zheng, Weisheng Dong, and Guangming Shi. A3R: Agentic affordance reasoning via cross-dimensional evidence in 3D Gaussian scenes. *arXiv preprint arXiv:2604.01882*, 2026.

Chenguo Lin and Yadong Mu. InstructScene: Instruction-driven 3D indoor scene synthesis with semantic graph prior. In *The Twelfth International Conference on Learning Representations*, 2024.

Léopold Maillard, Francis Engelmann, Tom Durand, Boxiao Pan, Yang You, Or Litany, Leonidas Guibas, and Maks Ovsjanikov. SceneTeract: Agentic functional affordances and VLM grounding in 3D scenes. *arXiv preprint arXiv:2603.29798*, 2026.

Yongsen Mao, Junhao Zhong, Chuan Fang, Jia Zheng, Rui Tang, Hao Zhu, Ping Tan, and Zihan Zhou. SpatialLM: Training large language models for structured indoor modeling. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Arjun Panickssery, Samuel R. Bowman, and Shi Feng. LLM evaluators recognize and favor their own generations. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Despoina Paschalidou, Amlan Kar, Maria Shugrina, Karsten Kreis, Andreas Geiger, and Sanja Fidler. ATISS: Autoregressive transformers for indoor scene synthesis. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Xingjian Ran, Yixuan Li, Linning Xu, Mulin Yu, and Bo Dai. Direct numerical layout generation for 3D indoor scene synthesis via spatial reasoning. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://arxiv.org/abs/2506.05341.

Fan-Yun Sun, Weiyu Liu, Siyi Gu, Dylan Lim, Goutam Bhat, Federico Tombari, Manling Li, Nick Haber, and Jiajun Wu. LayoutVLM: Differentiable optimization of 3D layout via vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 29469–29478, 2025.

Hou In Ivan Tam, Hou In Derek Pun, Austin T. Wang, Angel X. Chang, and Manolis Savva. SceneEval: Evaluating semantic coherence in text-conditioned 3D indoor scene synthesis. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 7355–7365, 2026.

Jiapeng Tang, Yinyu Nie, Lev Markhasin, Angela Dai, Justus Thies, and Matthias Nießner. DiffuScene: Denoising diffusion models for generative indoor scene synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Cheng Wan, Yongsen Mao, Wenzheng Wu, Yuxuan Xie, Chucheng Xiang, Runze Wang, Xiang Zhang, Zhongyuan Liu, Rushi Dai, and Yuan Liu. NaLA: A 3D native LLM layout agent for high-quality 3D scene generation. In *European Conference on Computer Vision*, 2026. URL https://arxiv.org/abs/2606.29395.

Xinyi Wang, Xun Yang, Yanlong Xu, Yuchen Wu, Zhen Li, and Na Zhao. AffordBot: 3D fine-grained embodied reasoning via multimodal large language models. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-4691.

Chucheng Xiang, Ruchao Bao, Biyin Feng, Wenzheng Wu, Zhongyuan Liu, Yirui Guan, and Ligang Liu. Co-layout: Llm-driven co-optimization for interior layout. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 14371–14379, 2026.

Yandan Yang, Baoxiong Jia, Peiyuan Zhi, and Siyuan Huang. PhyScene: Physically interactable 3D scene synthesis for embodied AI. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16262–16272, 2024a.

Yandan Yang, Baoxiong Jia, Shujie Zhang, and Siyuan Huang. SceneWeaver: All-in-one 3D scene synthesis with an extensible and self-reflective agent. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://arxiv.org/abs/2509.20414.

Yue Yang, Fan-Yun Sun, Luca Weihs, Eli VanderBilt, Alvaro Herrasti, Winson Han, Jiajun Wu, Nick Haber, Ranjay Krishna, Lingjie Liu, Chris Callison-Burch, Mark Yatskar, Aniruddha Kembhavi, and Christopher Clark. Holodeck: Language guided generation of 3D embodied AI environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16227–16237, 2024b.

Mario Zechner and Earendil Works. Pi Coding Agent. https://github.com/earendil-works/pi, 2026. npm package @earendil-works/pi-coding-agent.

Sharon Zhang, R. Kenny Jones, Jiajun Wu, and Maneesh Agrawala. Aggregating LLM-based weak verifiers for spatial layout generation. *arXiv preprint arXiv:2606.05268*, 2026. doi: 10.48550/arXiv.2606.05268. URL https://arxiv.org/abs/2606.05268.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 Embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025.

Haoyu Zhao, Akide Liu, Zeyu Zhang, Weijie Wang, Feng Chen, Ruihan Zhu, Gholamreza Haffari, and Bohan Zhuang. CoV: Chain-of-view prompting for spatial reasoning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 32427–32440, 2026. doi: 10.18653/v1/2026.findings-acl.1623.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Xiaoming Zhu, Xu Huang, Qinghongbing Xie, Zhi Deng, Junsheng Yu, Yirui Guan, Zhongyuan Liu, Lin Zhu, Qijun Zhao, Ligang Liu, and Long Zeng. Imaginarium: Vision-guided high-quality 3D scene layout generation. *arXiv preprint arXiv:2510.15564*, 2025.

A Evaluation Details

This appendix expands the three stages of Section 3: formulating checks, inspecting and judging them, and producing scores with diagnoses. A *metric* is one of the eight evaluation dimensions under the physical, visual coherence, and human-use principles. A *check* is a scene-specific question about particular objects or relationships. A *check type* specifies the property to inspect, and an *evaluation procedure* specifies how a metric formulates and resolves its checks. The agent's work is divided among VLM discovery, deterministic procedures, a camera selector, and a metric evaluator that issues the verdicts for each metric. Appendix B gives the score computation, and Appendix C lists the evaluation configuration.

A.1 End-to-End Evaluation Overview

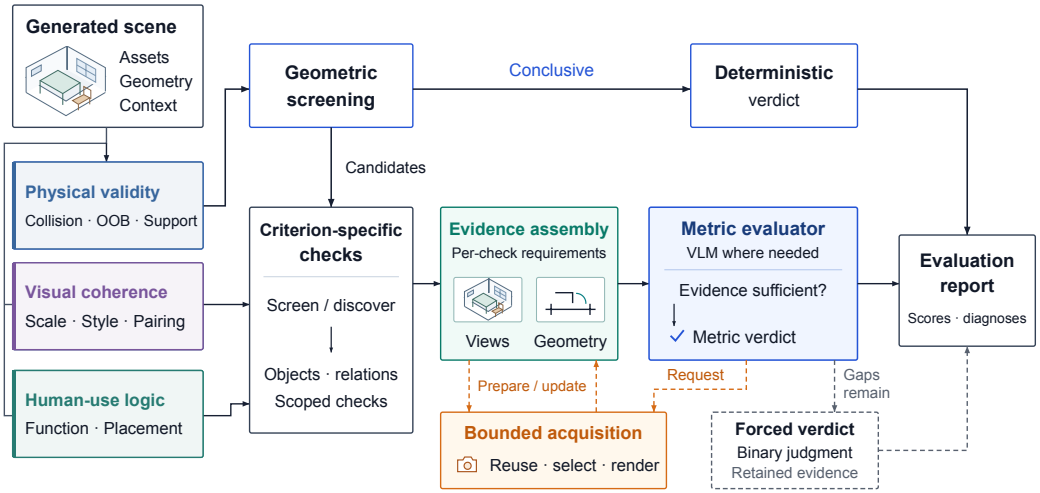


Figure 7: Per-metric evaluation procedures. Criterion-specific checks organize evidence for adjudication; conclusive geometric predicates are resolved directly. Bounded acquisition supports check preparation and requests for more evidence. The procedure depends on the metric (Table 4). OOB denotes out-of-bounds.

GREP evaluates each room from its objects, asset geometry, positions, orientations, scales, and available scene context. Relationships and explanations authored by the generator are treated as claims, not as evidence that the described relations were realized. Before evaluation, input checks verify that the scene is parseable, that coordinates and units are consistent, and that the required asset data are available. A room that fails these checks is not evaluated. Boundary checks use the spatial boundary defined for the evaluated setting.

Table 4 summarizes how each metric formulates, scopes, and resolves its checks, and Fig. 7 shows the corresponding procedures. Physical metrics begin with geometric screening, which resolves conclusive cases directly and passes the remaining candidates on for inspection. Scale and Pairing begin with a VLM screen of structured object information that resolves clearly valid cases and sends suspected defects to visual confirmation. Style begins with a global visual screen, followed by local confirmation when the global screen identifies a suspected defect. Function and Placement use VLM discovery to select, from the submitted objects, the objects and relationships to inspect within fixed check types.

Checks that need further inspection receive measurements and views. The metric evaluator interprets this evidence and may request additional views when a required property remains unclear, and the camera selector supplies them while the scene stays fixed. Verdicts come from deterministic procedures or from the metric evaluator, never from the camera selector, and a check may be resolved without passing through every component. Each check ends in an inspection record that retains its verdict, reason, supporting evidence, and decision source. These records feed both score aggregation and object-level diagnoses.

Table 4: Per-metric evaluation procedures. A group is an object group from the room partition (Appendix A.2), and the evaluator is the metric's VLM evaluator.

Metric	Check formulation	Evidence scope	Resolution
Collision	Mesh and oriented bounding box (OBB) screening of object pairs	Object pair	Deterministic for certified separation or a bounded interface overlap, else evaluator
Out-of-bounds	Screening against the room envelope	Object and boundary	Deterministic for containment within tolerance, else evaluator
Support	Support-chain screening along gravity	Object and its supports	Deterministic for grounded contact, else evaluator
Scale	VLM screen of structured object information	Group, after the screen	Screen for clearly valid cases, else evaluator
Style	Global visual screen	Room, then group	Screen, or evaluator after local confirmation
Pairing	VLM screen of the object inventory	Room inventory, at most one global view	Screen for clearly valid cases, else evaluator
Function	Two-stage VLM discovery of affordances and joint-use relations	Group with a shared evidence bank, plus own scopes for cross-group relations	Evaluator
Placement	VLM discovery of location checks	Group, with the global view	Evaluator

A.2 Check Instantiation and Evidence Scope

A metric names the quality to assess, while a check names the objects, relationship, and property to inspect in a particular room. In the chair–table example of Fig. 2, Function checks whether the chair is oriented toward the table and whether the space between them permits their joint use.

Function discovery runs in two VLM stages over the global view, object identities and categories, and supplied scene and architectural context. The first stage records the affordances of individual objects, and the second audits joint-use relationships between them. Discovery selects among the submitted objects and does not add objects. Accepted proposals take one of five check types: architecture orientation, clearance, local affordance confirmation, directional correspondence, and relative-use geometry. Placement discovery proposes checks of three types (support surface and height, scene zone, and contextual anchor) from the global view, object information, and the spatial context produced by Function check discovery. This context directs attention and does not carry Function verdicts.

Deterministic procedures validate the object references and check type of each proposal, remove duplicates, and convert accepted proposals into checks. Each check fixes the inspection target and its evidence requirements and leaves the verdict to the metric evaluator. Discovery therefore instantiates checks from a fixed vocabulary rather than generating new evaluation procedures.

Object groups define local inspection scopes. A VLM proposes the partition of the room, with a geometric-topology fallback when the proposal fails. The partition considers support and attachment, local interaction, and shared anchors before proximity alone, keeps unusual objects, and gives functional relationships that span groups their own scopes. Table 4 lists the scope each metric uses, and all scopes stay within the evaluated room.

A.3 Check-Specific Evidence Requirements

Each check specifies the observations and measurements its judgment needs. For Function and Placement, each check type carries predefined requirements, supplemented by observation goals from discovery and by later requests from the metric evaluator. A chair–table orientation check requires identifiable targets, a reference for the chair's usable side, and the relative orientation and distance between the two objects; a clearance check additionally requires evidence of the surrounding region.

The usable-side reference comes from the asset catalog where available and otherwise from VLM interpretation of asset-side previews. Combined with the object transforms, it gives the world-space directions used for measurement.

An evidence packet may contain object identities, asset geometry, global and local views, spatial measurements, and scene context, and it records unavailable measurements and unmet observation requirements. Function checks in the same object group draw on a shared evidence bank, so a view acquired for one check can serve the others in that group.

Evidence availability and sufficiency are assessed separately. A deterministic gate confirms only that an observation can be consumed: the file exists, decodes, and is not blank. Whether the observation reveals the orientation, contact region, or obstruction relevant to the check is judged by the metric evaluator. Evidence coverage refers to the instantiated checks and does not show that discovery found every relevant relationship in the room.

A.4 Bounded Evidence Acquisition

Evidence is acquired at two points: while Function checks are prepared, and when a metric evaluator requests more evidence during adjudication. For Function, deterministic measurements are attached to accepted checks before any camera scheduling, so the spatial facts remain available even when no view is added. These measurements use the asset geometry and the resolved usable-side reference. Clearance is measured as the obstruction of an approach corridor in front of the usable side.

Function check preparation first reserves views for relationships that span object groups and then allocates the remaining probe units by the additional observation coverage each view provides. The resulting views enter the group's shared evidence bank. The other metrics acquire views for their own scopes (Table 4).

Camera candidates are generated deterministically and filtered for geometric feasibility, visibility, and framing. The camera selector returns views rather than verdicts, and the metric evaluator decides whether a new view resolves the question.

Acquisition limits apply per evaluation episode to evidence rounds, images, camera-selection calls, and camera actions, and they count usage carried into the episode. Function check preparation has its own budget, so no single rendering budget applies to a whole room. Budget exhaustion and infeasible views are recorded in the inspection record. Table 7 in Appendix C lists the limits.

A.5 Adjudication and Reporting

Conclusive geometric predicates are resolved deterministically. They include certified separation between two objects, containment within the evaluated boundary up to a floor-contact tolerance, grounded support contact, and two bounded interface overlaps: a millimeter-scale overlap at a support interface, and a vertical-only overlap between an explicit floor covering and an object resting on it. An overlap between coarse bounding volumes identifies a collision candidate rather than a verdict, and relationship claims authored by the generator never exempt a pair.

When a physical check remains inconclusive, the metric evaluator adjudicates it from scoped geometric and visual evidence and treats the measurements as fixed facts. For human-use checks, measurements establish spatial facts that the evaluator interprets in context: a chair–table distance informs the judgment, but usability also depends on orientation and access. Checks still unresolved when the evidence budget is exhausted receive a forced binary verdict from the retained evidence, and on some scene-quality procedures non-hard failures resolve to a labeled default-valid outcome.

Each inspection record identifies the affected objects or relationship, the check, the verdict and its reason, the supporting views or measurements, and the decision source, which distinguishes deterministic verdicts, evaluator verdicts, forced verdicts, and policy defaults. Execution failures are recorded separately and are not treated as verdicts. Appendix B specifies how the records contribute to scores, and the same records provide the object-level diagnoses used for scene revision.

B SCORING DETAILS

We compute quality scores in three stages: assigning penalties to confirmed defects, combining these penalties for each object and metric, and aggregating the resulting metric scores.

Let m denote an evaluation metric, $\mathcal{O}$ the set of objects in the evaluated room, and $N = |\mathcal{O}|$ the number of objects, with $N > 0$. Each object instance is counted once, and all metrics use the same N.

A confirmed defect constitutes an event e with penalty $p(e)$. We define p_0 as the minimum penalty and p_1 as the maximum penalty for a confirmed defect. The penalty range is $\Delta p = p_1 - p_0$. We use $p_0 = 0.4$ and $p_1 = 1.0$, giving $\Delta p = 0.6$.

For Collision, Support, and Out-of-bounds, let $M(e) \in [0, 1]$ denote the normalized geometric severity of event e. The event penalty is

$$p(e) = p_0 + \Delta p\, M(e). \tag{1}$$

$M(e)$ reflects the extent of penetration, the support gap, or the extent of boundary crossing. When a defect is confirmed but its magnitude cannot be reliably measured, we assign p_0 and record the missing measurement. The remaining metrics use the severity levels in Table 5.

Table 5: Severity levels and their event penalties.

Metric	Severity and penalty
Scale	Noticeable: p_0; gross: p_1
Style	Noticeable: p_0; gross: p_1
Pairing	Invalid: p_1
Function	Impaired: p_0; blocked: p_1
Placement	Atypical: p_0; implausible: p_1

Valid judgments contribute no penalty. Repeated observations of the same defect are merged, retaining the strongest confirmed severity. Let $\mathcal{E}_m$ denote the resulting set of defect events for metric m.

Each event penalty is assigned to the responsible objects. Let $a(e, o)$ denote the portion of event e's penalty assigned to object o. An event involving one scoring object assigns its entire penalty to that object; a shared penalty is divided among the scoring objects. The allocation preserves the event's total penalty:

$$\sum_{o \in \mathcal{O}} a(e, o) = p(e), \qquad a(e, o) \geq 0. \tag{2}$$

Objects receiving no part of the event's penalty have $a(e, o) = 0$. For example, a Collision event divides its penalty equally between its two endpoints.

We define $b(m, o)$ as the combined penalty assigned to object o under metric m. For Collision, Support, Out-of-bounds, Scale, Style, and Pairing, assigned penalties are added and capped at one:

$$b(m, o) = \min\left(1, \sum_{e \in \mathcal{E}_m} a(e, o)\right). \tag{3}$$

For Function and Placement, we retain the largest assigned penalty:

$$b(m, o) = \max_{e \in \mathcal{E}_m} a(e, o). \tag{4}$$

Thus, two impaired Function defects assigned entirely to the same object contribute p_0 in total. If either defect is blocked, the object's contribution becomes p_1. An object with no assigned defects has $b(m, o) = 0$.

We then compute two quantities for each metric. The total object penalty B_m measures the accumulated defects across the room:

$$B_m = \sum_{o \in \mathcal{O}} b(m, o). \tag{5}$$

The maximum event penalty P_m measures the severity of the single worst defect before its penalty is divided among objects:

$$P_m = \max_{e \in \mathcal{E}_m} p(e). \tag{6}$$

For example, an event with penalty one contributes $P_m = 1$ even when its penalty is divided between two objects. When $\mathcal{E}_m$ is empty, we define all corresponding object penalties and both B_m and P_m to be zero.

Two hyperparameters control the score mapping: $\alpha \in [0, 1]$ controls the influence of the worst defect, and $\gamma > 0$ controls sensitivity to the overall penalty proportion B_m/N. We use $\alpha = 0.40$ and $\gamma = 4$. The metric score combines these two quantities:

$$S_m = 100\,(1 - \alpha P_m)\left(1 - \frac{B_m}{N}\right)^{\gamma}. \tag{7}$$

Scores lie between zero and 100, with a fully evaluated metric receiving 100 when no defects are found. Placement uses its primary checks for this calculation; the additional global review does not contribute.

For example, a room containing 20 objects with one impaired Function defect assigned to one object has $B_m = p_0$ and $P_m = p_0$. Its Function score is

$$S_{\text{Function}} = 100\,(1 - \alpha p_0)\left(1 - \frac{p_0}{20}\right)^{\gamma} \approx 77.48. \tag{8}$$

The room score is a weighted average of its metric scores. Let w_m denote the weight of metric m, with $\sum_m w_m = 1$:

$$Q_{\text{room}} = \sum_m w_m S_m. \tag{9}$$

The metric weights are listed in Table 6.

Table 6: Metric weights used to aggregate room scores.

Metric	w_m
Collision	0.15
Support	0.15
Out-of-bounds	0.08
Scale	0.03
Style	0.03
Pairing	0.04
Function	0.32
Placement	0.20

The result tables also report one score per principle, the weight-normalized average of its metric scores:

$$S_p = \frac{\sum_{m \in p} w_m S_m}{\sum_{m \in p} w_m}. \tag{10}$$

The physical principle comprises Collision, Support, and Out-of-bounds; the visual coherence principle comprises Scale, Style, and Pairing; and the human-use principle comprises Function and Placement. The room score therefore equals $0.38\,S_{\text{phys}} + 0.10\,S_{\text{vis}} + 0.52\,S_{\text{use}}$. The FrozenAssets* configuration fixes asset identities and scales before placement. Since object selection, style, and scale are predetermined in these experiments, their overall score excludes the visual coherence metrics. We mark these experiments with an asterisk ($*$), giving an overall score of $(0.38\,S_{\text{phys}} + 0.52\,S_{\text{use}})/0.90$. Multi-room scene scores are weighted averages of room scores using net floor area. Model and method scores are equal averages across included scenes.

Unresolved judgments and evaluation failures are recorded separately from valid and invalid judgments. Partial results retain their judgment-coverage information and enter the aggregate only when they meet the experiment's scoring eligibility criteria. Missing results are never assigned zero or a perfect score.

C Experimental Setup

C.1 Benchmark Composition

The benchmark comprises three generation settings, with task counts specified separately for each setting.

Model-native (Table 1). Each model is evaluated on 30 tasks: 20 open-space scenes (S100–S119) and 10 relatively simple floorplans (`layout_01`–`layout_10`). The open-space setting (WF, wall-free) has one room per scene and no enclosing walls. The floorplan tasks form the rectangular (RC) setting, comprising 31 rectangular rooms across the ten floorplans.

Specialized pipelines (Table 2). Each model–pipeline configuration is evaluated on ten open-space scenes (S100–S109), comprising ten single rooms under FrozenAssets*. These tasks share their room briefs and boundaries with the corresponding WF subset of the Model-native setting, while their asset inventories are fixed before generation.

Harnessed agents (Table 3). Each backbone is evaluated on 20 cases: ten under the default condition and ten under the *more complicated condition* ($+20\%$ assets). The two conditions reuse the same ten irregular floorplans, with 42 rooms in each condition. We refer to this architectural setting as irregular (IR). The increased furnishing requirements define the second set of cases; the floorplans and required room programs are shared between the two conditions.

WF task descriptions specify the intended room use, dimensions, target instance-count range, and furnishing instruction. RC additionally provides room rectangles, local-to-global offsets, active walls, openings, and the order in which rooms are generated. Each RC room has a prescribed use and object-count range. Counts refer to expanded object instances rather than object categories or rows in an object plan.

The IR floorplans are adapted from the SpatialLM dataset (Mao et al., 2025). Their horizontal coordinates are uniformly scaled by a factor of 1.25, preserving the polygon structure while increasing floor area by a factor of 1.5625. The benchmark supplies the required room programs and activities, together with total and per-room instance-count ranges. It does not supply the source dataset's room-to-use assignment: the generator assigns each required program to a room. The architecture, program requirements, count ranges, and asset-scale constraints remain fixed within a comparison.

The shared Imaginarium asset library (Zhu et al., 2025) contains 2,043 distinct assets and 385 distinct labels in its `category` field. Assets are accompanied by descriptions, dimensions, and identity metadata. FrozenAssets* fixes a human-curated inventory for each of ten WF tasks (S100–S109), comprising 269 object instances drawn from 168 distinct assets in total. Every pipeline receives the same inventory for a task, including exact asset identities and native scales. This condition isolates placement from object and asset selection.

C.2 Generation Settings

Model-native. We use *Plan–Retrieve–Place* for ordinary Model-native generation. It uses two model calls: one to plan the objects and one to place them, with deterministic asset retrieval between the calls. The retrieval profile uses Qwen3-Embedding-0.6B (Zhang et al., 2025) and a deterministic Top-1 policy combining semantic similarity with size compatibility and a fixed tie-breaking order. The model has no interactive tool loop or evaluator feedback during these two generation stages. Task descriptions, retrieval policy, and output contracts are held fixed across the backbones compared under this generation setting.

RC is generated sequentially, one room at a time. Each room receives one object-planning call and one placement call, with retrieval between them. Assembly prefixes instance identifiers with the room identity and translates room-local coordinates into the floorplan frame; it does not reposition, rotate, rescale, or repair the objects.

GPT-6 Astra, GPT-5.6 Sol, and HY4 Preview use `high` reasoning effort. Claude Opus 5, Sonnet 5, and Fable 5 use the provider-default `high` setting. Kimi K3 and GLM-5.3 use `max`, with a

lower-effort fallback when reasoning exceeds the token limit. The Model-native setting supplies the model-controlled planning and placement reference.

Specialized pipelines. We pair LayoutGPT, DirectLayout, LayoutVLM, and SceneWeaver with GPT-5.6 Sol and GPT-4.1 under FrozenAssets*. Each method receives one WF task, its room boundary, fixed object inventory, asset identities, and physical dimensions. Adapters preserve these bindings while translating between the benchmark scene representation and each method's native inputs and outputs.

LayoutGPT uses its numerical Cascading Style Sheets (CSS) representation with fixed object dimensions and an in-context example set. DirectLayout receives the frozen assets through its native generation and refinement pipeline. LayoutVLM receives fixed asset sizes and identities and runs its constraint-generation and layout-solving procedure. SceneWeaver initializes the supplied meshes and retains its iterative reflection loop, while object insertion, removal, replacement, and resizing are disabled. These adaptations constrain the inventory while preserving the methods' placement procedures.

The within-backbone baseline, labeled None in Table 2, uses the placement stage of Plan–Retrieve–Place. Because the inventory is already fixed, it makes one placement call and skips object planning and retrieval, with no iterative refinement. Methods retain their own call and optimization schedules, so inference calls, token use, and wall-clock computation are not equalized. Since asset selection and scale are fixed, the overall FrozenAssets* score excludes the visual coherence metrics and follows the renormalization in Appendix B.

Harnessed agents. All backbones operate through a common harness built on the open-source Pi coding agent (Zechner & Earendil Works, 2026) (version 0.85.0), with an isolated workspace, immutable architecture, required room programs, count ranges, and access to the same asset database. The agent chooses its object composition, searches for assets, inspects their metadata, and writes a complete submission. The tool interface provides `get-task`, `search-assets`, `inspect-asset`, `validate-submission`, and `finalize-submission`. Asset inspection returns structured metadata and bounding-box dimensions. The harness provides no rendering or image observations to the generation agent.

The public validator checks the submission schema, layout identity, room-program assignment, total and per-room instance counts, asset membership, binding and placement cardinality, and unit asset scale. It supplies neither GREP quality scores nor defect diagnoses or suggested repairs. A completed generation requires a finalized submission that passes this contract. The same scene definitions, asset access, and episode limits apply across the agent backbones.

Each episode permits at most 120 model requests and 160 benchmark-tool calls, with sublimits of 80 asset searches, 80 asset inspections, and 30 submission validations. A search returns at most 12 candidates. The episode wall-clock limit is 14,400 seconds. These are generation limits, separate from the evaluation budgets in Section C.3.

The default furnishing condition uses the benchmark's prescribed integer ranges. The *more complicated condition* increases the required furnishing density: each per-room lower and upper bound is multiplied by 1.2 and rounded down, and scene-level bounds are the sums of the resulting room bounds. The architecture, room programs, asset scales, and tool limits remain unchanged. The default condition provides the reference for harnessed agents, while the more complicated condition tests increased furnishing demands.

C.3 Evaluation Configuration

Evaluation and rendering. GREP evaluates generated scenes room by room, using GPT-5.6 Sol for VLM judgments. Evidence rendering uses the supplied asset meshes and scene poses, with the scene fixed throughout inspection. The reference rendering configuration uses the EEVEE rendering engine of Blender (Blender Online Community, 2026) with 768×768 final views and 256×256 candidate previews. Global room views initialize inspection, and targeted views provide additional evidence when requested. The metric procedures are described in Appendix A, and Table 7 gives the evidence-acquisition limits. Score parameters, metric weights, FrozenAssets* renormalization, and room-to-scene aggregation follow Appendix B.

Table 7: Evidence-acquisition limits. Episode limits apply to each scoped evaluation episode, and Function check preparation has a separate budget.

Limit	Value
Evidence rounds per episode	3
Images per episode	8
Views per round	2
Camera-selection calls per episode	4
Camera actions per episode	3
Deterministic camera candidates per request	6
VLM repair proposals per request	3
Function check preparation: probe units	32
Function check preparation: candidates per probe	4
Function check preparation: views kept per probe	1
Function shared evidence bank: images per active window	6

Completion and inclusion. Generation completion, report completion, and judgment coverage are tracked separately. A failed generation has no quality score, including when it exhausts its allowed attempts without producing an accepted scene. Missing or excluded results are not assigned zero or a perfect score. Only scenes with a scoring-eligible report for every required room enter the aggregate. Room scores are weighted by their net floor areas within each scene, and eligible scenes receive equal weight.

For Table 1, all eight models contribute 20 scoring-eligible WF scenes. In RC, seven models contribute all ten floorplans (31 rooms); HY4 Preview contributes nine floorplans (27 rooms). Its generated `layout_09` is excluded because one required room lacks a scoring-eligible report. For Table 2, all ten scenes are generated and included for each model–pipeline configuration. For Table 3, every backbone contributes all ten IR floorplans (42 rooms) under both conditions, and all 168 rooms in each condition have scoring-eligible reports.

D Evaluator Validation Details

This appendix gives the experimental procedures and results behind Section 4.2: repeatability under a fixed configuration and sensitivity to the judge model (Appendix D.1), agreement with human judgments (Appendix D.2), and a comparison with alternative evaluators (Appendix D.3). GREP scene scores use the 0–100 scale defined in Appendix B; agreement rates below are percentages of pairwise preferences.

D.1 Repeatability and Judge Sensitivity

Experimental procedure. We analyze 12 selected scenes generated by eight Model-native backbones on the open-space tasks S110–S119. Each scene was evaluated five times with GPT-5.6 Sol as the judge, with the generation inputs, evaluator configuration, and judge fixed. The inputs were re-hashed before every repeat and were identical across repeats, and all 60 evaluations completed with full judgment coverage, so the differences reported below arise from evaluation alone.

Scene scores. Table 8 lists the mean, standard deviation (SD), and range of each scene over the five repeats. The per-scene standard deviation has a median of 2.9 points (from 0.4 to 7.3), and the median difference between the highest and lowest repeat is 7.2 points (maximum 17.2). Differences between scenes are larger: repeat-to-repeat variation accounts for 8% of the score variance across the 12 scenes (intraclass correlation 0.92), and any two repeats order the scenes with a Spearman correlation between 0.70 and 0.98 (median 0.90). In every scene, at least one metric's number of confirmed defects changed across repeats.

Implications for comparisons. Assuming independent evaluations, the variation observed here gives a standard error of about 1.3 points for a mean over ten scenes and about 1.9 points for the difference between two such means. Differences between model means below about 4 points are

Table 8: Repeatability of GREP scores. Each scene was evaluated five times with fixed inputs, configuration, and judge (GPT-5.6 Sol).

Generator	Task	Mean	SD	Min–Max
GPT-6 Astra	S112	97.7	2.0	94.4–99.3
Kimi K3	S118	95.8	5.2	86.6–98.7
GPT-5.6 Sol	S118	94.4	0.4	93.9–94.8
Claude Fable 5	S110	92.1	0.7	90.9–92.4
Claude Opus 5	S114	89.1	2.8	84.1–90.7
Claude Sonnet 5	S116	88.1	1.0	87.1–89.6
GPT-6 Astra	S113	87.6	7.1	74.9–91.1
GLM-5.3	S118	83.5	2.2	81.0–85.7
HY4 Preview	S113	80.8	6.0	73.3–88.3
GPT-5.6 Sol	S116	64.4	4.6	58.5–70.9
Claude Opus 5	S111	63.4	7.3	54.5–71.7
Claude Fable 5	S114	54.9	2.9	51.1–58.9

therefore within what a single evaluation per scene can produce. The refinement study reports paired differences with their standard errors for the same reason (Appendix E).

Judge sensitivity. We use 26 room evaluations across six scenes to assess judge sensitivity, with Claude Fable 5 as the alternative judge. The two judges order the six scenes almost identically (Spearman $\rho = 0.94$; only the two lowest-scoring scenes swap). Fable 5 scores every scene higher, by 4.8 points on average (paired $t = 1.88$ with five degrees of freedom, $p = 0.12$). The largest gap, 16.8 points, occurs on a scene generated by Claude Fable 5; the other five differ by 0.3 to 6.7 points. Its repeated scores also vary less (median per-scene standard deviation 0.1 versus 2.9 points for GPT-5.6 Sol on the same scenes). With six scenes, the analysis indicates that a second judge preserves relative comparisons while shifting absolute scores; it cannot establish whether a judge favors scenes generated by its own model family.

D.2 Human Agreement

Experimental procedure. Five computer graphics researchers, including graduate students and professors, independently compared six selected open-space scenes. The study contains all 15 unordered scene pairs, with each scene appearing in five pairs. Each pair presents one global perspective image per scene, its intended room use and dimensions, and descriptions of physical validity, visual coherence, and human-use logic. Annotators selected A or B for the scene with better overall quality in its intended room context. They did not explore the scenes or receive local views, metric scores, or per-check judgments. All five annotators completed all 15 pairs, yielding 75 preferences.

Results. All five annotators agreed on 11 of the 15 pairs; the other four pairs split three to two. Across the ten annotator pairs for each scene pair, agreement was 126/150, or 84.0% (Fleiss' $\kappa = 0.677$). GREP's Overall scene scores agreed with the human majority on 13/15 pairs (86.7%) and with 61/75 individual preferences (81.3%). These comparisons use the same six scenes, whose 15 pairwise decisions are not independent scene samples.

Disagreements. GREP differed from the human majority on two pairs. For P07, all five annotators preferred GLM-5.3 S115 over Claude Opus 5 S111, whereas GREP's Overall scores preferred the latter. For P09, three annotators preferred GLM-5.3 S115 and two preferred Claude Fable 5 S113, whereas GREP preferred Fable S113. Thus, P07 is a unanimous human disagreement and P09 a divided human judgment.

D.3 Comparison with Alternative Evaluators

Fig. 6 compares GREP with a holistic VLM and a Physical-only score baseline using the same 15 pairs. The holistic VLM uses GPT-5.6 Sol with the same model configuration as GREP's judge.

In each of three separate runs, it receives the same six global images, room context, and overall-preference instructions as the human annotators, and returns one A/B choice for every pair. It does not receive scene geometry, GREP scores, additional views, or other annotators' choices. Across the three runs, its agreement with the human majority was 62.2% $\pm$ 10.2%, and agreement with individual preferences was 60.4% $\pm$ 12.7% (mean $\pm$ between-run SD). Its choice was identical across runs on 10/15 pairs.

The Physical-only baseline orders each pair by its existing GREP Physical score, which combines Collision, Support, and Out-of-bounds. It agreed with the human majority on 8/15 pairs (53.3%) and with 36/75 individual preferences (48.0%). Its scores tied on P03; because human annotators made a forced A/B choice, a tied prediction receives no agreement credit. These are post-adjudication GREP Physical scores, not the output of an independent geometry-only detector. Humans and the holistic VLM used the same global-image evidence bank, whereas GREP and the Physical-only baseline reuse the original GREP evaluations and their evidence.

E Refinement Study Details*

This appendix documents the refinement study of Section 4.2.1: the experimental procedure and feedback content (Appendix E.1), per-scene results (Appendix E.2), layout changes (Appendix E.3), an example of diagnosed defects and their revision (Appendix E.4), and the evaluation records (Appendix E.5).

E.1 Experimental Procedure and Feedback Content

The study uses five open-space scenes (S101–S105) under FrozenAssets*, with GPT-4.1 as the generator. The generator places the fixed inventory of each task at native scale, using the placement stage of Plan–Retrieve–Place (Appendix C.2). The initial placement (R0) fixes instance identities and counts; revisions change only positions and yaw rotations, and every response contains all instances.

Each scene has one R0 layout, shared by three branches. In every round, the generator receives the task, the frozen object plan and asset selection, the previous layout, and a branch-specific evaluation report of that layout:

- **No feedback.** The report is empty, and the generator is asked to inspect the layout itself.
- **Scores.** The report contains the room score and the scores of the five metrics that enter the FrozenAssets* overall score: Collision, Support, Out-of-bounds, Function, and Placement.
- **Report.** The report contains the same scores and every confirmed defect of the previous layout. Each defect record gives the metric, the check type, the affected objects, a severity, the evaluator's reason, and the measurements attached to the check (Table 9). Unresolved items are listed separately as neither defects nor passes.

The branches share one system prompt. For the Report branch, it adds that the report lists confirmed defects with their evidence and omits passing checks, that unlisted objects and relations are not certified as correct, and that the generator should correct the confirmed defects while preserving the rest of the layout where possible. In all branches, the report is supplied as data and cannot override the placement contract.

Pipeline scoring protocol. For this pipeline-based refinement study, we report the evaluator's recorded benchmark score on a 0–1 scale, retaining its original metric scores and layer aggregation. The motivation for using a separate protocol is to preserve discrimination among low-quality layouts: large accumulated defect penalties can compress scores near the lower bound and obscure differences between revisions. The reported score is

$$s_{\text{pipeline}} = 0.3 s_{\text{physical}} + 0.7 s_{\text{quality}},$$

where the two terms are the physical and scene-quality layer scores recorded by the evaluator. The scene-quality term retains the original evaluator's metric contributions, excluding its visual metrics. All three branches use the same reported score definition. The feedback delivered to the Scores and Report branches contains only the five retained FrozenAssets* metrics and their renormalized total,

so it differs from the full benchmark score reported here. This study's reporting protocol differs from the main 0–100 GREP protocol in Appendix B; its scores are not obtained by dividing those scores by 100. Each branch ran three revisions, therefore we analyze R1–R3.

Table 9: A defect record from the diagnostic report* on the R0 layout of S104, shortened to the fields listed. Records also carry the verdict, a defect category, the check scope, an attribution field, and check identifiers, omitted here.

Field	Content
Metric	Function
Check type	Relative-use geometry
Objects	Piano bench, with the digital piano
Severity	Blocked
Reason	"The bench is positioned about 2.0 m from the piano rather than within seated operating reach, so it cannot function as seating for the piano in the authored arrangement."
Measurements	Center distance 2.0 m; the seating side of the bench and the control side of the piano each at 90° to the direction of the other object

E.2 Per-Scene Results

Table 10 lists every score under the pipeline scoring protocol. Averaged over scenes and R1–R3, No feedback scores 0.552, Scores 0.565, and Report 0.626, from a shared R0 mean of 0.585. Paired by scene and round (15 pairs), Scores differs from No feedback by $+0.013$ (standard error 0.012), Report by $+0.074$ (standard error 0.027; positive in 11 pairs), and Report from Scores by $+0.061$ (standard error 0.025; positive in 12 pairs). These pairs contain repeated observations from the same five scenes. The Report branch has the highest mean in four of the five scenes. S103 is the exception: its Scores branch, which never changed the layout, has the highest mean.

Table 10: Pipeline scores of the refinement study (0–1). R0 is shared by the three branches of a scene. A dagger marks a revision whose layout is identical to that of the previous round.

Scene	Branch	R0	R1	R2	R3	Mean R1–R3
S101	No feedback	0.534	0.489	0.456†	0.526	0.490
	Scores		0.494	0.517	0.499	0.503
	Report		0.435	0.526	0.670	0.544
S102	No feedback	0.608	0.542†	0.458†	0.403†	0.468
	Scores		0.468†	0.493†	0.458†	0.473
	Report		0.569	0.612	0.712	0.631
S103	No feedback	0.525	0.566	0.618†	0.628†	0.604
	Scores		0.591†	0.659†	0.615†	0.622
	Report		0.656	0.508	0.590	0.585
S104	No feedback	0.605	0.627†	0.620	0.594	0.613
	Scores		0.615†	0.615†	0.629†	0.619
	Report		0.665	0.616	0.706	0.662
S105	No feedback	0.653	0.527†	0.665	0.563†	0.585
	Scores		0.610	0.598	0.616†	0.608
	Report		0.715	0.713	0.702	0.710

E.3 Layout Changes

A revision counts as changed when the hash of its placement file differs from that of the previous round; R1 is compared with R0. The layout changed in 6 of 15 revisions without feedback, 5 of 15 with scores, and 15 of 15 with reports. The score branch returned the R0 layout in every round for S102, S103, and S104, and the no-feedback branch did so for S102. Unchanged revisions are evaluated again rather than copied, so their scores differ from R0 only through evaluation. For S103,

the three unchanged layouts of the score branch score 0.591 to 0.659 against 0.525 for R0, because the evaluation of R0 confirmed one Placement defect that the later evaluations did not.

E.4 Example: Function in S104

S104 asks for a media, music, game, and recreation room of 10.4 by 7.6 m with 30 objects. The evaluation of its R0 layout confirmed 12 Function defects. Eight objects, among them the television, soundbar, media console, both studio monitors, and the guitar amplifier, had their functional side facing the room boundary. Three access regions were blocked, for example the access face of a bookcase 0.22 m from the boundary. The piano bench stood 2.0 m from the digital piano (Table 9). The Report branch received these records together with the defects of the other metrics.

In R1, none of the eleven orientation and clearance defects was confirmed again, and the recorded Function score rose from 0.447 to 0.750, remaining at 0.750 in R2. Over the same rounds, Function stayed between 0.400 and 0.480 in the other two branches, whose evaluations confirmed 11 to 13 Function defects in every round. The revision was incomplete. The piano bench moved next to the piano but across its control side, with its back toward the keyboard, and was reported in R1, R2, and R3 without being resolved. R1 also introduced a new defect, a soundbar placed at the position of the television, which was cleared in R2. In R3, the recorded Function score remained 0.750 despite two further defects: the television and the sofa were each turned 63° away from the other, so viewing required turning the head, and the soundbar blocked the front of the media console. The reported pipeline score was 0.605 at R0, followed by 0.665, 0.616, and 0.706. Physical defects remained: ten Collision defects were recorded in every round, and Support defects rose from two in R0 to five in R1 and R2.

E.5 Evaluation Records

The study comprises 50 evaluations: five R0 layouts and 45 revisions. Every retained evaluation is complete with full judgment coverage. Nine were retained from a second or third attempt after the earlier attempts did not produce a complete evaluation; incomplete attempts were discarded and were neither scored nor fed back. The feedback for each round was produced from the retained evaluation of the previous layout.