SAME SCENE, DIFFERENT DECISIONS: MATERIAL SENSITIVITY IN EMBODIED MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Embodied models must connect visual appearance to task-relevant decisions. We study their material sensitivity: how those decisions change when scene geometry is retained but materials are converted to a different representation. In 99 paired navigation episodes, success falls from 52.5% to 48.5%, yet 30 episodes change success status: gains and losses largely cancel in the average. We investigate this hidden turnover through paired visual grounding and navigation experiments. A readout study across six vision-language model configurations separates localization from response-format and coordinate-scale effects. Controlled two-view tests then require models to follow a queried object across positions, and target-window crossovers reveal that changes to the target and its surroundings can shift grounding in opposite directions. In navigation, 17 previously successful episodes fail while 13 previously failed episodes succeed under converted materials. These results put the identity of successful decisions alongside mean accuracy as a central object of evaluation. Material representation is therefore more than an asset-format choice: it is an input variable through which we can study the visual dependence of embodied models.

1 INTRODUCTION

A scene's floor plan can stay fixed while an embodied model's decisions change. Converting materials alters color, contrast, and reflections without moving the objects that define the task. A navigation instruction still refers to the same destination, and a grounding query still refers to the same object. What changes is the evidence the model sees. Material sensitivity asks whether the model maintains the same task-relevant judgment across this appearance change. As simulated scenes move between rendering systems, this seemingly technical asset operation becomes an experiment on visual dependence.

The consequences are easy to miss in aggregate. In our paired navigation evaluation, success falls from 52.5% to 48.5%, but 30.3% of episodes change success status. Seventeen previously successful episodes fail; thirteen previously failed episodes succeed. A net loss of four thus hides thirty changed outcomes. Mean accuracy describes how many tasks succeed, but loses track of which tasks succeed.

Visual grounding brings this question into sharper focus. Given paired images, we can ask whether a model still points to the requested object, then change the query, the object's position, or only part of the image. A point prediction also introduces an important measurement problem: the same response can map to different locations under different coordinate conventions. We therefore separate the model's visual dependence from the interface used to express a location before interpreting its response to material conversion.

We start from USD scenes with NVIDIA MDL materials and construct UsdPreviewSurface counterparts while preserving scene composition and geometry. Figure 1 organizes these paired interventions; Figure 2 previews the resulting changes in model decisions. A readout experiment spans six vision-language model (VLM) configurations. Controlled two-view experiments then make the query determine the target: two objects share a canvas, their positions are swapped, and the image center lies between them. Finally, region crossovers replace the target window or its context to ask where appearance changes matter. In a separate navigation evaluation, the material intervention changes the observations available throughout an episode.

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.

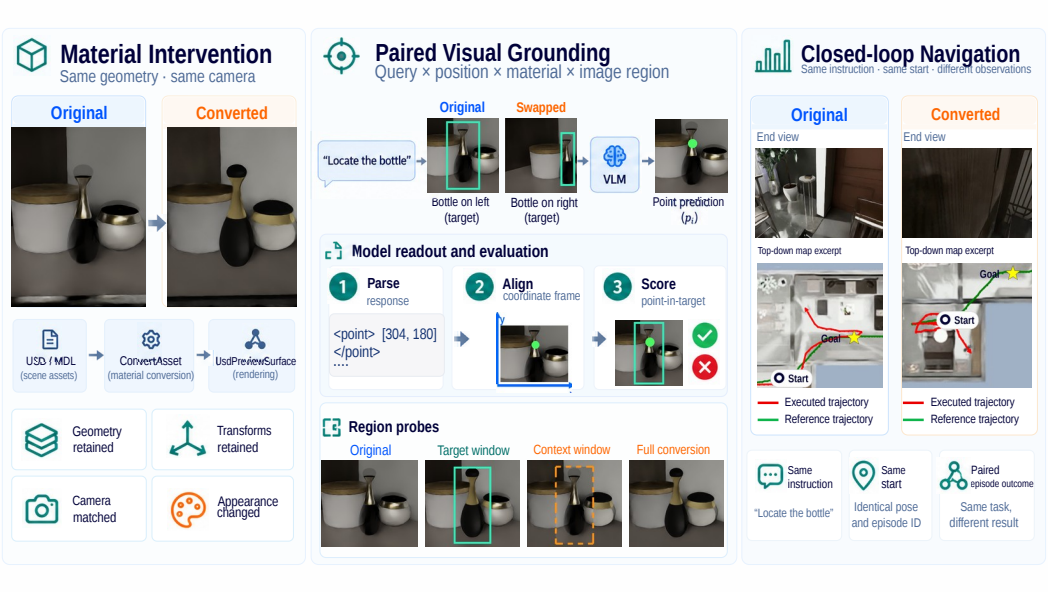

Figure 1: Probing material sensitivity in embodied models with ConvertAsset. *Left*: material conversion (USD/MDL → UsdPreviewSurface) retains scene geometry, transforms, and camera while changing rendered appearance. *Middle*: paired visual grounding varies the query, the target position, the material, and the image region; a parse–align–score readout turns the recorded VLM point prediction into a point-in-target decision, and region probes isolate where appearance changes matter. *Right*: closed-loop navigation pairs episodes that share the instruction, the start pose, and the episode ID but differ in observations. All image panels show recorded model inputs, recorded predictions, and recorded navigation runs.

The experiments establish three connected findings. Spatial readout can reshape the measured material effect; target and context changes can push grounding in opposite directions; and small net navigation changes can conceal substantial turnover in successful episodes. Together, they characterize material sensitivity at the level of paired decisions, from a single point to an episode outcome. This perspective makes the identity of successful queries and episodes part of the evaluation, alongside their total number. The accompanying experimental package preserves the inputs, responses, and pairing needed to trace those changes.

2 Related Work

Appearance and learned visual cues. Domain randomization varies simulated appearance to improve transfer (Tobin et al., 2017; Tremblay et al., 2018). Texture–shape cue conflicts reveal that models can rely on visual evidence different from that used by humans (Geirhos et al., 2019). Our intervention targets an operational source of appearance variation: changing a scene's material representation while retaining its geometry. This lets us examine localization and navigation decisions under a concrete transformation encountered when simulation assets are reused.

Grounding and model readout. Shikra and Ferret integrate referring and grounding with language generation (Chen et al., 2023; You et al., 2024). Molmo explicitly develops pointing data, Qwen2.5-VL supports object localization and dynamic image resolution, and InternVL3 explores multimodal training and inference recipes (Deitke et al., 2025; Bai et al., 2025b; Zhu et al., 2025). These capabilities make spatial outputs useful, but the interface through which a point is emitted also becomes part of the experiment. Prompt dependence is well established for language models (Zhao et al., 2021); we examine its interaction with a paired visual intervention and separate executable output from conditional localization accuracy.

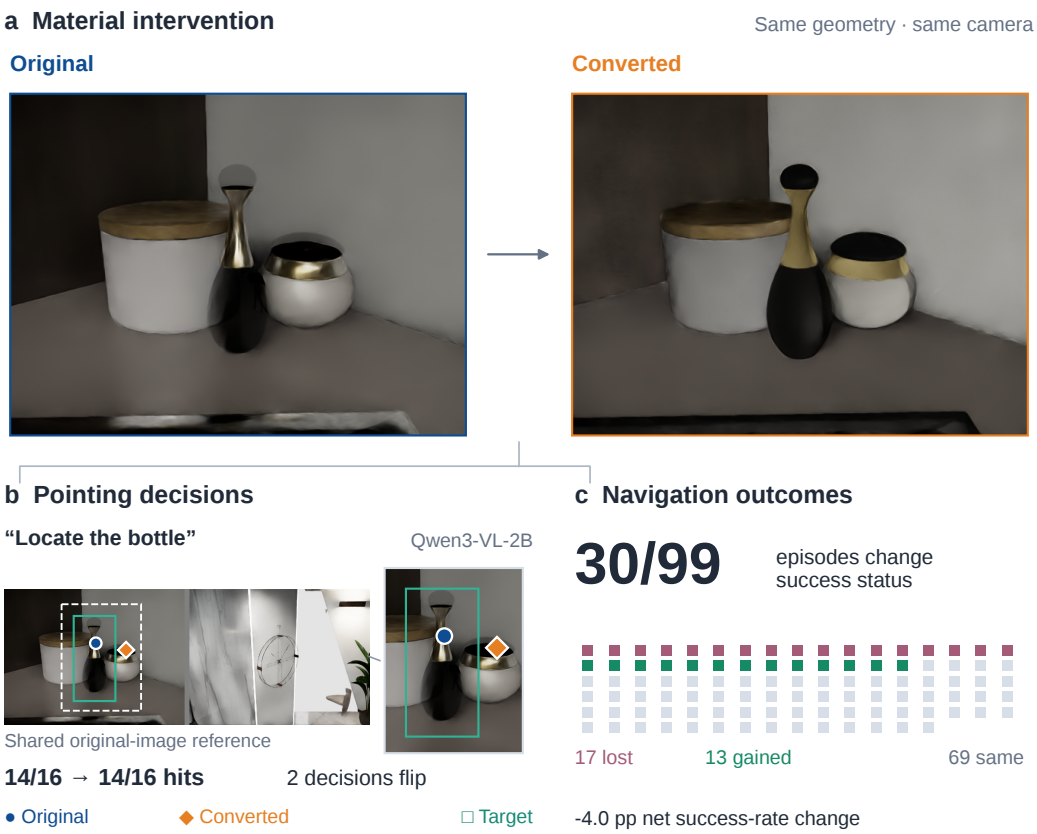

Figure 2: Same scene, different decisions. (a) Original MDL and converted UsdPreviewSurface views retain scene geometry. (b) Grounding points from both conditions share an original-image reference; the dashed outline identifies the enlargement and the green box marks the target. Neither outline is shown to the model. Equal hit totals hide changes in which queries succeed. (c) Navigation has 30 changed outcomes but a net loss of four successes.

Embodied evaluation. Vision-and-language navigation couples language understanding to sequential perception and action (Anderson et al., 2018). GRUtopia provides interactive scenes and tasks for embodied research (Wang et al., 2024); DualVLN combines grounding and control in a navigation system (Wei et al., 2026). We use GRScenes for point-level experiments and KuJiaLe scenes for paired navigation. The common experimental question is how a change to rendered appearance alters decisions, from one image to an entire episode.

3 Material Intervention and Model Readout

Changing the material representation. We construct two versions of each source scene. The original version retains its MDL shader networks; the converted version uses UsdPreviewSurface networks. The conversion traverses USD references, payloads, sublayers, variants, and texture dependencies, preserving composition rather than flattening the scene. Meshes and object transforms are retained. Matched camera settings expose the resulting appearance difference, including changes in color, contrast, and reflected light, to the model. Thus a material change can alter visual evidence even when the spatial answer remains the same.

A spatial output has two components. A model first generates a response to an image and a query. A readout rule then maps that response to a location. Let I_i^O and I_i^C denote paired images, q_i a query, and B_i the target region. For a model f and a prespecified readout r, the decoded point is $p_i = r(f(I_i, q_i))$. A readout includes both the response syntax and the coordinate frame. We

measure the complete pipeline:

$$H_i^v = \mathbf{1}\{p_i^v \text{ is a valid in-image point in } B_i\}, \qquad \Delta_H = \frac{1}{N}\sum_{i=1}^{N}(H_i^C - H_i^O). \tag{1}$$

A missing or malformed point has $H_i^v = 0$. Alongside H, we report numeric-output coverage and the hit rate among parsed numeric outputs. Coverage records whether a point can be decoded; the end-to-end hit rate additionally requires it to lie inside the image and target. All point displacements are converted to image pixels before distances are computed.

Fixed-factor comparisons. The readout experiment crosses two requested frames (pixels and normalized 0–1000) with two syntaxes (JSON and two-line text), using the same image queries. The six configurations are Gemma4-E4B in four-bit form (Google DeepMind, 2026), Qwen2.5-VL-7B (Bai et al., 2025b), Qwen3-VL-2B and 8B (Bai et al., 2025a), InternVL3-8B (Zhu et al., 2025), and Molmo-7B-D (Deitke et al., 2025). The analysis preserves the checkpoint, raw response, image digest, and prompt for every query. A common 0–1000 reinterpretation provides a second, uniformly applied diagnostic alongside scoring in the requested frame.

Decision turnover. Mean differences can cancel. We therefore also count O-success/C-failure and O-failure/C-success pairs, for both pointing and navigation. For multiple views of one object, we compare the actual sets of successful views rather than just their cardinalities. Uncertainty for point-hit differences is estimated by resampling target clusters, keeping repeated views together. Experimental details and paired render examples appear in Appendix A; Appendix B compares the retained material cues.

From a point to a trajectory. For navigation, the material intervention is applied to the KuJiaLe scenes used by the InternNav/DualVLN evaluation route. The language instruction, start state, episode definition, and agent configuration are matched across conditions. We record success rate (SR), success weighted by path length (SPL), navigation error (NE), and trajectory length (TL), retaining episode IDs to connect aggregate changes to individual outcomes.

4 Grounding: Following the Object Across Appearance Changes

Grounding tests should require the model to use both the image and the query. Our initial target-centered views reveal why this matters: a fixed image-center point hits every target box in the 30-pair close-view pool. We therefore change the task structure. We place two different rendered views on one canvas, ask about either category, and swap their positions. The center is now a neutral gutter between the views. The target location is determined jointly by the query and the panel arrangement.

Query and position swaps. Each board contains two different target categories. Both targets are queried, and both left/right arrangements are evaluated in each material condition. Annotations are translated with the corresponding view; boxes are never shown to the model. Four repaired source pairs form four boards, yielding 16 paired queries per intervention across two targets and two position permutations. Category queries ask for the named object. Action queries describe an interaction such as grasping a cup or turning on a faucet and ask for the corresponding object's location. The output is a single point on the complete canvas. Appendix C gives the source pairs and composition details.

Table 1 treats language dependence and appearance dependence as separate experimental axes. Swapping the query on an otherwise identical board asks whether the point follows the requested object. Swapping view positions asks whether the model can preserve that binding away from a preferred location. The paired material comparison then asks whether a different appearance changes the same binding. This structure makes a successful point more informative than success on a single large, centered target.

Material conversion changes which queries succeed. Qwen3-VL-8B solves 16 original category queries and 15 converted queries; the image-center baseline solves none. Qwen3-VL-2B has 14 and 14 hits in the two conditions but changes hit status on 2 paired queries. Gemma4 moves from 12 to 13 hits with 3 transitions. As in navigation, equal or nearly equal totals need not identify the same successful decisions. The action-framed query changes this pattern: Qwen3-VL-2B moves from 9

Model	Intervention	N	Original	Changed	Δ (pp)	Flips
Gemma4-E4B	Category query	16	75.0	81.2	+6.2	3/16
	Action query	16	75.0	75.0	+0.0	2/16
	Target window	16	75.0	68.8	-6.2	3/16
	Context window	16	75.0	87.5	+12.5	2/16
Qwen3-VL-2B	Category query	16	87.5	87.5	+0.0	2/16
	Action query	16	56.2	75.0	+18.8	3/16
	Target window	16	87.5	93.8	+6.2	3/16
	Context window	16	87.5	100.0	+12.5	2/16
Qwen3-VL-8B	Category query	16	100.0	93.8	-6.2	1/16
	Action query	16	93.8	93.8	+0.0	0/16
	Target window	16	100.0	93.8	-6.2	1/16
	Context window	16	100.0	100.0	+0.0	0/16

Table 1: Paired two-view interventions. Success rates are percentages over all queries; invalid outputs count as failures. Original and Changed refer to the image conditions for the listed intervention. Category and action queries use full converted views; window interventions exchange only the specified region. Flips count changes in hit status.

to 12 hits, whereas Qwen3-VL-8B retains 15 in both conditions. The gain for the smaller model is concentrated in action-framed queries, showing that the response to appearance depends on how the object is requested.

Scale and model behavior. The broader paired archive supplies a complementary readout experiment over six model configurations, including close and wider views. Its corrected tables report end-to-end outcomes and output coverage together (Appendix E). These data expose large differences between a generated number and a usable spatial prediction. For example, a low hit rate under one requested frame can coexist with much better spatial alignment under another interpretation of the same response. The two-view task tests the visual and language binding that remains after a frame and parser have been fixed.

5 Disentangling Visual Dependence from the Output Interface

Where in the image does the change matter? We use rectangular-window crossovers to localize dependence on the changed image. Let W be the projected target box in the composed canvas. The target-window condition replaces pixels in W with their converted counterparts while retaining the original context. The context-window condition makes the opposite exchange. The two conditions are exact pixel interventions with the same query, canvas, and target location. They separate sensitivity to the target window from sensitivity to the surrounding visual evidence.

A converted scene changes both the target and the visual context in which it appears. The crossover reveals whether changing one region reproduces the response to full conversion, or instead pushes the prediction in another direction.

Gemma4 provides a concrete example. Replacing the target window reduces success from 75.0% to 68.8%, whereas replacing the context increases it to 87.5%. The complete conversion yields 81.2%. For Qwen3-VL-8B, the target-window replacement introduces one miss and the context replacement retains all sixteen hits. Qwen3-VL-2B instead reaches sixteen hits with converted context, compared with fourteen in the original condition. These controlled outcomes show that the location of the appearance change matters: target-window and contextual changes can affect a model in different directions.

How is a location expressed? Output syntax and coordinate scale create a second mechanism for score changes. Figure 4 holds each set of raster inputs fixed across the readout variants. Several configurations respond differently to pixel and normalized requests. Molmo, for example, produces responses on a 0–100 scale in some JSON/pixel prompts, while its two-line normalized prompts

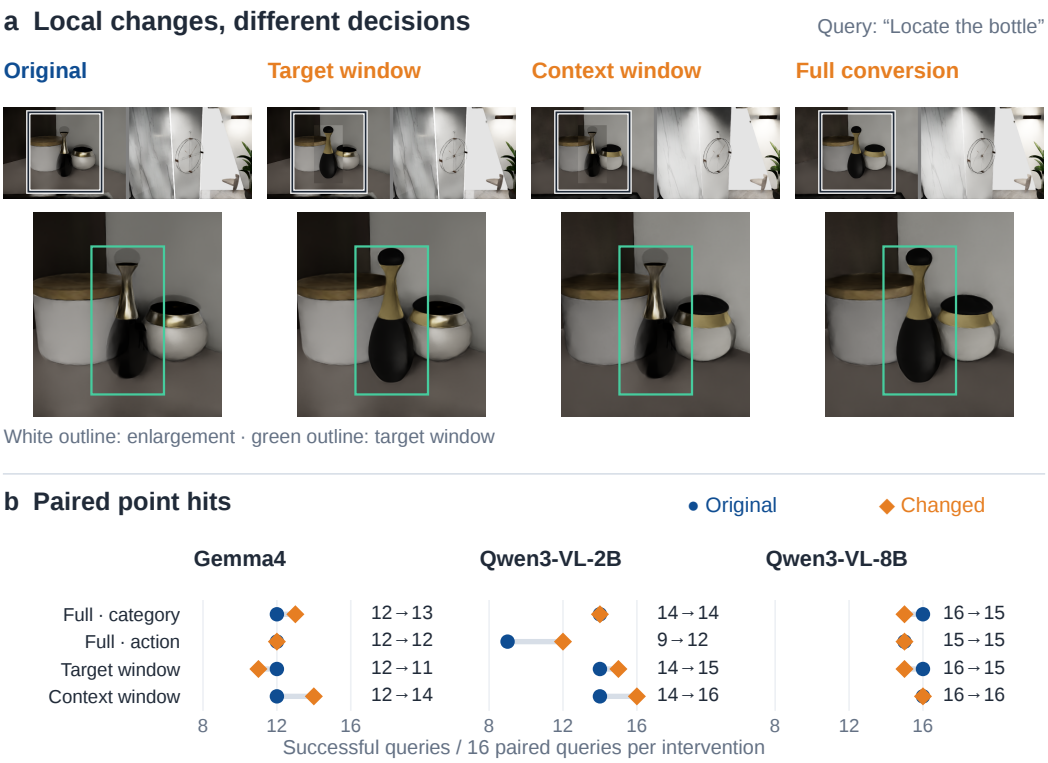

Figure 3: Region crossovers and their grounding outcomes. Full canvases are shown above matched enlargements. White outlines identify the enlargement; green boxes mark the target window. The four columns show original, target-window, context-window, and fully converted conditions. Annotations are absent from model inputs. Paired counts use all 16 queries per condition.

produce 0–1000-style points. Qwen3-VL-8B commonly retains a normalized-style output even when asked for pixels. The raw responses reveal these scale mismatches before target-box scoring.

The response format therefore changes the meaning of a measured material effect. A parser may discard a response, a coordinate mismatch may displace a correctly identified location, or a valid point may genuinely move on the image. Our analysis distinguishes these cases using raw text, explicit coordinate transforms, and pixel distances. A miss caused by a scale mismatch therefore has a different explanation from a point that moves off the object under a fixed readout.

Decoding variability. Three stochastic runs per image provide another view of spatial readout. On the recorded raster inputs, Qwen3-VL-8B has a mean pairwise seed distance of 7.66 pixels, compared with 24.34 for Qwen3-VL-2B and 27.07 for Gemma4. The corresponding hit-status variability is 0%, 16.7%, and 10.0%. Table 9 gives the complete-query counts. These differences motivate fixed decoding when comparing the two image conditions.

6 Navigation: Small Net Changes, Different Successful Episodes

The downstream experiment asks what happens when a language-conditioned agent observes converted scenes throughout an episode. We evaluate InternNav/DualVLN on the three KuJiaLe scenes in the official `val_unseen` package used here, with 33 episodes per scene. The paired run comprises 99 episode pairs and 198 rollouts. Both conditions use the same episode IDs and task definitions. The agent receives each material condition throughout closed-loop execution.

Outcome cancellation hides decision turnover. Across the 99 pairs, original materials yield 52 successful episodes and converted materials yield 48. The net loss of four consists of 17 success-to-failure transitions and 13 failure-to-success transitions. Consequently, 30 of 99 episodes change

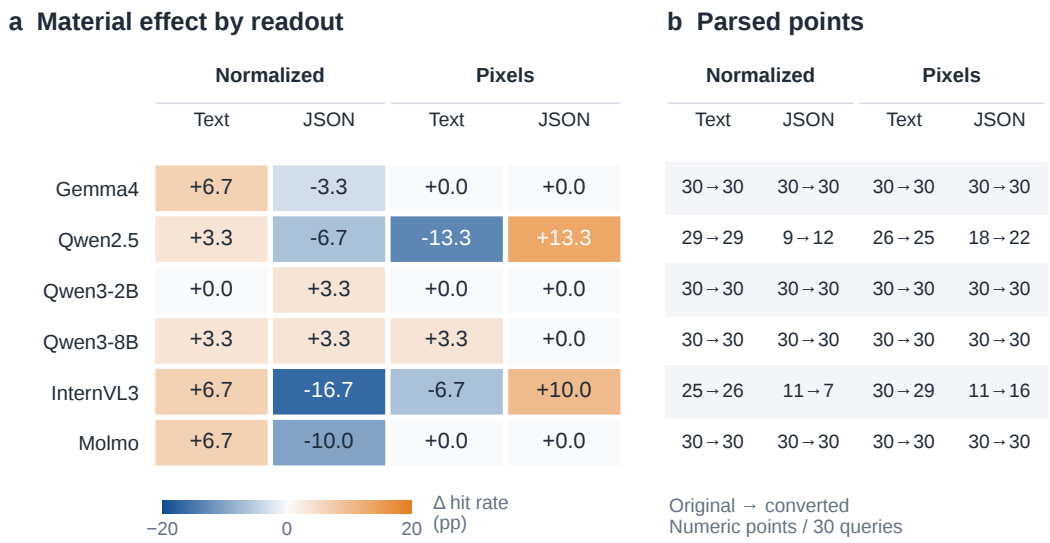

Figure 4: Readout-dependent paired score changes in the 30-pair raster archive. Each cell uses the requested coordinate frame and all 30 queries in each condition. The heatmap reports percentage-point differences and the accompanying coverage summary counts parsed numeric outputs. Full counts are in Table 5.

Scene	N	SR_O	SR_C	SPL_O	SPL_C	Flips
0031	33	51.5	57.6	47.9	49.6	8
0036	33	81.8	60.6	70.0	55.0	11
0066	33	24.2	27.3	24.2	24.3	11
All	99	52.5	48.5	47.4	43.0	30

Table 2: Paired navigation results by scene. SR and SPL are percentages; O and C denote original and converted materials. Flips count episodes whose success status changes between conditions.

success status: 30.3% turnover despite a 4.0 percentage-point aggregate SR change. Figure 5 exposes this cancellation directly. An unchanged mean would not imply that the same instructions still succeed.

Scene dependence. Scene 0036 contributes the largest net decrease, from 81.8% to 60.6% success, while scenes 0031 and 0066 improve in aggregate. SPL follows the largest decrease in scene 0036. Across all scenes, mean NE changes from 3.680 to 3.631, while mean TL changes from 6.975 to 7.060. These summaries describe different aspects of the trajectory: endpoint error, path efficiency, and binary completion need not move together. Retaining the episode-level transitions makes those differences visible instead of compressing them into a single stability score.

From outcomes to paths. Figure 6 pairs ending observations with map excerpts from recorded reruns of three episode IDs. The views show how the agent's surroundings and path differ across material conditions. Appendix F expands these comparisons to twelve recorded episode pairs.

7 Discussion and Conclusion

Material sensitivity in embodied models appears in which decisions succeed, as well as in aggregate performance. In navigation, a net loss of four successful episodes hides 30 changes in success status. In grounding, equal hit totals can likewise conceal different successful queries. Paired outcomes reveal this redistribution directly.

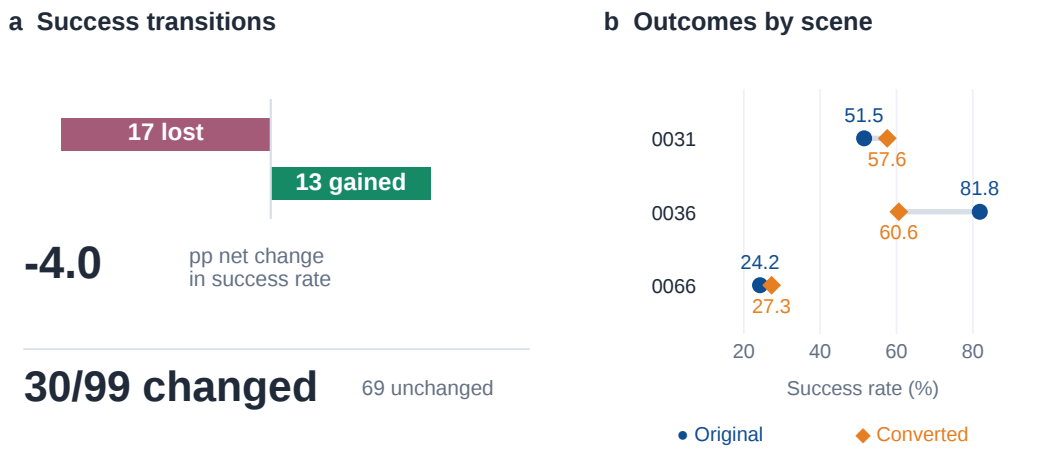

Figure 5: Paired navigation outcomes. Left: 17 losses and 13 gains produce 30 changes but a small net decrease. Right: scene-level success rates reveal opposing effects. The remaining 69 pairs retain their outcome.



Figure 6: Recorded ending views and map excerpts from separate qualitative reruns. Red view symbols indicate actions; red and green map lines show executed and reference paths. Table 2 reports the separate 99-pair metric runs.

The grounding experiments also show why interpreting that redistribution requires care with the measurement itself. Coordinate conventions determine how a response becomes a location; query-position swaps test whether that location follows the requested object; region crossovers distinguish

dependence on the target window from dependence on its surroundings. These comparisons turn a before/after score into a more informative account of model behavior.

As simulated scenes move between rendering systems, their geometry alone does not specify the evidence an embodied model will receive. Tracking which decisions change, alongside how much the mean changes, makes material representation an explicit and testable part of embodied evaluation.

AI Use Statement

Generative AI tools assisted with research planning, experiment implementation, result analysis, figure-generation code, illustrative figure components, and manuscript drafting and revision. Experimental scene images and navigation frames come from the recorded renders and rollouts. The accompanying artifact records raw model responses, input digests, execution configurations, and automated tests. These records document the AI-assisted experimental and writing workflow.

Reproducibility Statement

The supplement specifies the prompts, coordinate transforms, paired estimands, model environments, and image-composition procedures. Generated tables and figures are linked to machine-readable measurements. The experiment entry points support independent reanalysis of saved predictions and rerunning models with explicit configurations.

References

Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton van den Hengel. Vision-and-Language Navigation: Interpreting Visually-Grounded Navigation Instructions in Real Environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3674–3683, 2018. doi: 10.1109/CVPR.2018.00387. URL https://openaccess.thecvf.com/content_cvpr_2018/papers/Anderson_Vision-and-Language_Navigation_Interpreting_CVPR_2018_paper.pdf.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL Technical Report. *arXiv preprint arXiv:2511.21631*, 2025a. URL https://arxiv.org/abs/2511.21631.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL Technical Report. *arXiv preprint arXiv:2502.13923*, 2025b. URL https://arxiv.org/abs/2502.13923.

Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing Multimodal LLM's Referential Dialogue Magic. *arXiv preprint arXiv:2306.15195*, 2023. doi: 10.48550/arXiv.2306.15195. URL https://arxiv.org/abs/2306.15195.

Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, Jiasen Lu, Taira Anderson, Erin Bransom, Kiana Ehsani, Huong Ngo, YenSung Chen, Ajay Patel, Mark Yatskar, Chris Callison-Burch, Andrew Head, Rose Hendrix, Favyen Bastani, Eli VanderBilt, Nathan Lambert, Yvonne Chou, Arnavi Chheda, Jenna Sparks, Sam Skjonsberg, Michael Schmitz, Aaron Sarnat, Byron Bischoff, Pete Walsh, Chris Newell, Piper Wolters, Tanmay Gupta, Kuo-Hao Zeng, Jon Borchardt,

Dirk Groeneveld, Crystal Nam, Sophie Lebrecht, Caitlin Wittlif, Carissa Schoenick, Oscar Michel, Ranjay Krishna, Luca Weihs, Noah A. Smith, Hannaneh Hajishirzi, Ross Girshick, Ali Farhadi, and Aniruddha Kembhavi. Molmo and PixMo: Open Weights and Open Data for State-of-the-Art Vision-Language Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 91–104, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Deitke_Molmo_and_PixMo_Open_Weights_and_Open_Data_for_State-of-the-Art_CVPR_2025_paper.html.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1811.12231.

Google DeepMind. Gemma 4 model card. Google AI for Developers, 2026. URL https://ai.google.dev/gemma/docs/core/model_card_4. Accessed September 15, 2026.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning Robust Visual Features without Supervision. *Transactions on Machine Learning Research (TMLR)*, 2024. URL https://openreview.net/forum?id=a68SUt6zFt.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139 of *PMLR*, pp. 8748–8763, 2021. URL https://proceedings.mlr.press/v139/radford21a.html.

Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain Randomization for Transferring Deep Neural Networks from Simulation to the Real World. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 23–30, 2017. doi: 10.1109/IROS.2017.8202133. URL https://arxiv.org/abs/1703.06907.

Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon, and Stan Birchfield. Training Deep Networks with Synthetic Data: Bridging the Reality Gap by Domain Randomization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 969–977, 2018. doi: 10.1109/CVPRW.2018.00143. URL https://openaccess.thecvf.com/content_cvpr_2018_workshops/papers/w14/Tremblay_Training_Deep_Networks_CVPR_2018_paper.pdf.

Hanqing Wang, Jiahe Chen, Wensi Huang, Qingwei Ben, Tai Wang, Boyu Mi, Tao Huang, Siheng Zhao, Yilun Chen, Sizhe Yang, Peizhou Cao, Wenye Yu, Zichao Ye, Jialun Li, Junfeng Long, Zirui Wang, Huiling Wang, Ying Zhao, Zhongying Tu, Yu Qiao, Dahua Lin, and Jiangmiao Pang. GRUtopia: Dream General Robots in a City at Scale. *arXiv preprint arXiv:2407.10943*, 2024. doi: 10.48550/arXiv.2407.10943. URL https://arxiv.org/abs/2407.10943.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004. doi: 10.1109/TIP.2003.819861.

Meng Wei, Chenyang Wan, Jiaqi Peng, Xiqian Yu, Yuqiang Yang, Delin Feng, Wenzhe Cai, Chenming Zhu, Tai Wang, Jiangmiao Pang, and Xihui Liu. Ground Slow, Move Fast: A Dual-System Foundation Model for Generalizable Vision-and-Language Navigation. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=GK4rznYwhn.

Haoxuan You, Haotian Zhang, Zhe Gan, Xianzhi Du, Bowen Zhang, Zirui Wang, Liangliang Cao, Shih-Fu Chang, and Yinfei Yang. Ferret: Refer and Ground Anything Anywhere at Any Granularity. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=2msbbX3ydD.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 586–595, 2018. doi: 10.1109/CVPR.2018.00068. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Zhang_The_Unreasonable_Effectiveness_CVPR_2018_paper.html.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. Calibrate Before Use: Improving Few-shot Performance of Language Models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12697–12706, 2021. URL https://proceedings.mlr.press/v139/zhao21c.html.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, Zhangwei Gao, Erfei Cui, Xuehui Wang, Yue Cao, Yangzhou Liu, Xingguang Wei, Hongjie Zhang, Haomin Wang, Weiye Xu, Hao Li, Jiahao Wang, Nianchen Deng, Songze Li, Yinan He, Tan Jiang, Jiapeng Luo, Yi Wang, Conghui He, Botian Shi, Xingcheng Zhang, Wenqi Shao, Junjun He, Yingtong Xiong, Wenwen Qu, Peng Sun, Penglong Jiao, Han Lv, Lijun Wu, Kaipeng Zhang, Huipeng Deng, Jiaye Ge, Kai Chen, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. InternVL3: Exploring Advanced Training and Test-Time Recipes for Open-Source Multimodal Models. *arXiv preprint arXiv:2504.10479*, 2025. URL https://arxiv.org/abs/2504.10479.

A Conversion and the Appearance of the Input

What changes before a model sees the scene? Conversion replaces material networks while retaining meshes, object transforms, and scene composition. ConvertAsset follows USD references, payloads, sublayers, and variants and writes converted sibling layers; references in those layers point to their converted counterparts. The source scenes remain available as the original condition. Base color, roughness, metallic response, normal maps, and texture connections are mapped into UsdPreviewSurface. Richer material effects are examined separately in Appendix B.

A.1 Matched object views

The object comparison contains four chest-of-drawers assets, each rendered from six directions: front, back, left, right, and two elevated front views. Table 3 reports all 24 pairs. PSNR measures pixel error; SSIM measures local luminance and contrast structure (Wang et al., 2004); LPIPS uses learned perceptual features (Zhang et al., 2018). CLIP and DINOv2 cosine similarities compare image embeddings (Radford et al., 2021; Oquab et al., 2024).

Asset	Pairs	PSNR	SSIM	LPIPS	CLIP	DINOv2
Chest 0004	6	37.51	0.991	0.019	0.921	0.798
Chest 0011	6	36.30	0.993	0.017	0.956	0.922
Chest 0023	6	30.59	0.986	0.032	0.938	0.905
Chest 0029	6	37.68	0.991	0.014	0.885	0.864
All	24	35.52	0.990	0.020	0.925	0.872

Table 3: Matched object renders, averaged over six views per asset. PSNR is in dB; higher PSNR, SSIM, CLIP, and DINOv2 indicate greater similarity, while lower LPIPS indicates greater similarity.

The averages are high, but the per-asset results preserve useful differences. Chest 0023 has the lowest PSNR and highest LPIPS, whereas Chest 0004 has the lowest DINOv2 similarity. Pixel agreement and feature agreement thus emphasize different aspects of the same paired images. Figures 7 and 8 retain object context across camera directions so these measurements can be read alongside the visible surfaces.

For images scaled to $[0, 1]$, PSNR is $10 \log_{10}(1/\mathrm{MSE})$. The LPIPS implementation uses a VGG backbone. Feature measurements use CLIP ViT-B/32 and DINOv2 image embeddings, with cosine similarity $f(I^O)^\top f(I^C)/(\|f(I^O)\|_2 \|f(I^C)\|_2)$. Each reported mean weights the recorded views equally.

A.2 From objects to scene context

Figure 9 shows paired GRScenes views. Materials affect several parts of an observation at once: the target surface, the nearby background, and reflected illumination. These views motivate the full-image and window comparisons in the main paper. The object metrics describe the four-asset render set; the scene views show the visual inputs from which the grounding experiments were developed.

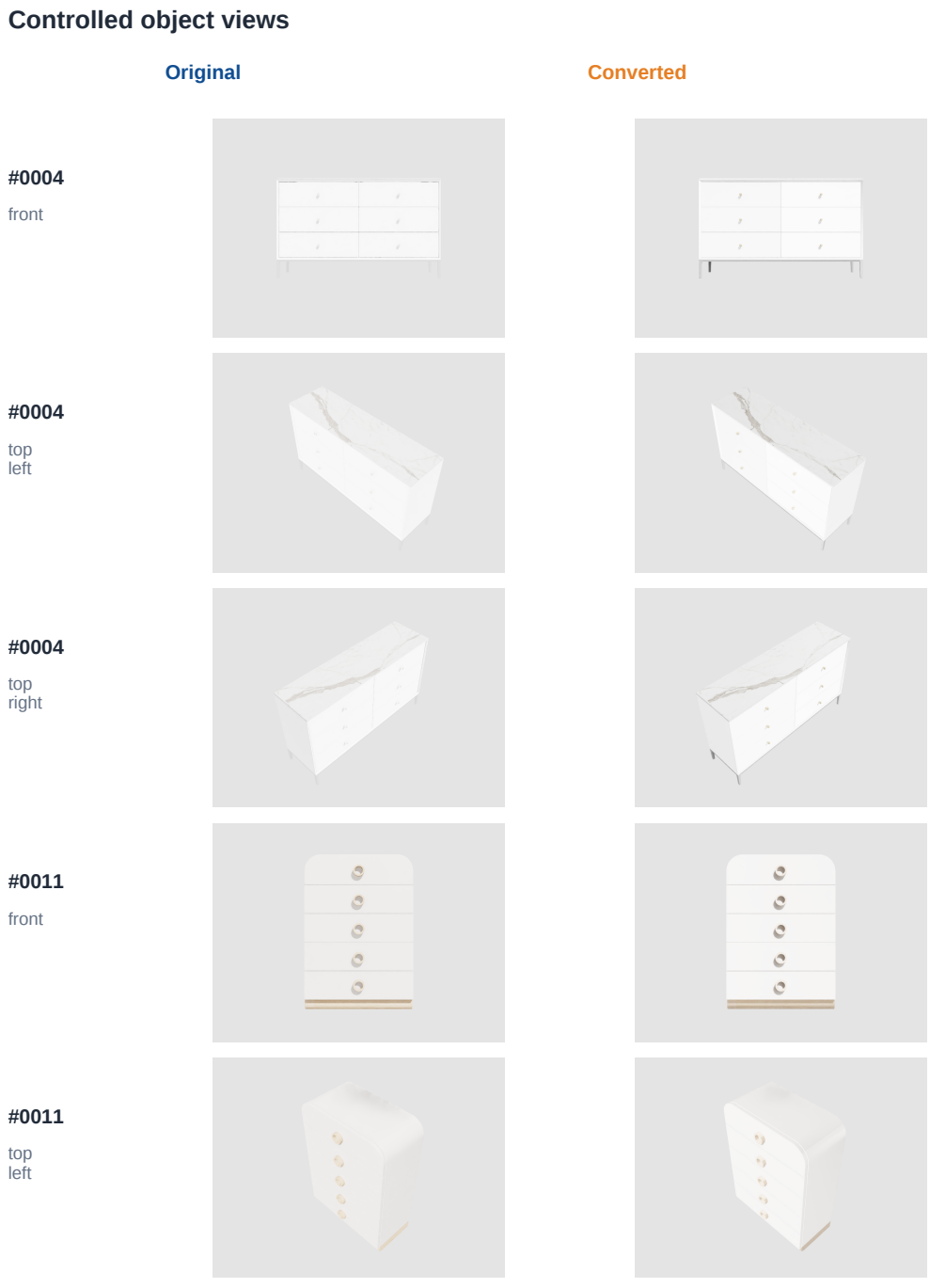

Figure 7: Original and converted object views from the four-asset render set. Each pair shows the same camera direction, retaining full object context.



Figure 8: Additional camera directions for the four-asset paired render set. The six-view measurements for every asset appear in Table 3.

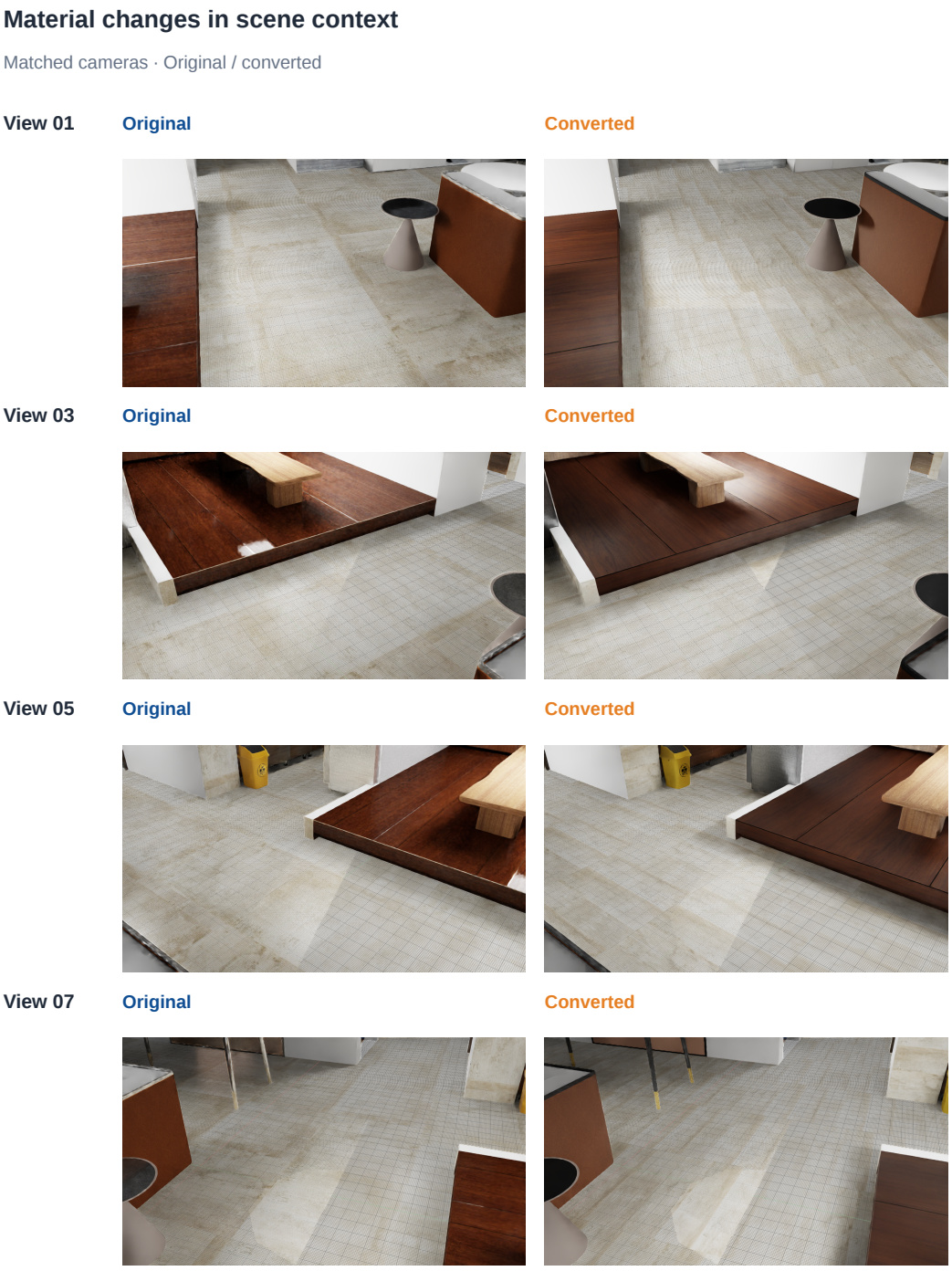

Figure 9: Paired GRScenes camera views under original and converted materials. Scene layout is shared while surface appearance and illumination cues change.

B WHICH MATERIAL CUES SURVIVE CONVERSION?

The three-condition comparison places original MDL, ConvertAsset, and the NVIDIA PreviewSurface conversion route side by side. The selected GRScenes cases cover opacity or transparency, emission, normal or bump mapping, and displacement or height. Figure 11 shows matched renders; Table 4 reports the corresponding static inventory. A sample may contain several effects, so the effect counts overlap.

Effect	N	Original	ConvertAsset	NVIDIA
Clearcoat fixture	1	1/1	1/1	0/1
Opacity / transparency	7	7/7	7/7	7/7
Emission	26	26/26	26/26	26/26
Procedural fixture	1	1/1	1/1	1/1
Normal / bump	30	30/30	30/30	30/30
Displacement / height	26	26/26	26/26	26/26

Table 4: Material-effect inventory. Entries count outputs passing the recorded static checks over selected samples. Converted outputs are checked for target presence and zero active MDL; visual appearance is examined in the paired panels. The two fixtures are listed separately from the GRScenes samples.

The four GRScenes effect groups produce statically usable outputs for both conversion routes. Their renders make the remaining appearance differences visible: loading a material successfully is only one part of preserving the cues a model receives. The shared columns let readers compare the target and its context directly, without interpreting the effect inventory as a perceptual score.

B.1 CLEARCOAT AND PROCEDURAL TEXTURE

The two fixtures in Figure 10 isolate concrete failure modes. In the clearcoat fixture, the NVIDIA-converted stage lacks the target prim. ConvertAsset retains the target but approximates its material; the recorded review flags the changed blue/clearcoat appearance. In the procedural-texture fixture, both converted stages retain the target, yet the authored checker pattern is lost. This second case is particularly instructive: static target and shader checks pass while a salient surface pattern disappears.

These fixtures explain why the grounding study uses rendered comparisons in addition to checking scene structure. They identify specific losses in the tested outputs; the selected examples do not estimate how often those losses occur across a material library.

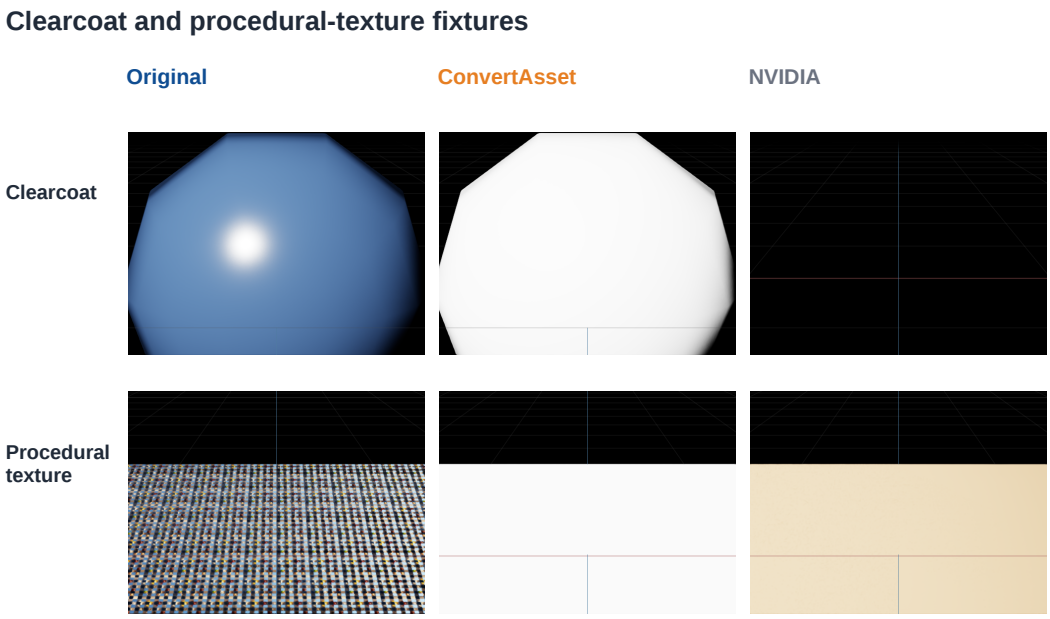

Figure 10: Clearcoat and procedural-texture fixtures under the three material conditions. The NVIDIA clearcoat output loses the target; both converted procedural outputs lose the checker pattern. ConvertAsset retains the clearcoat target with an altered appearance.

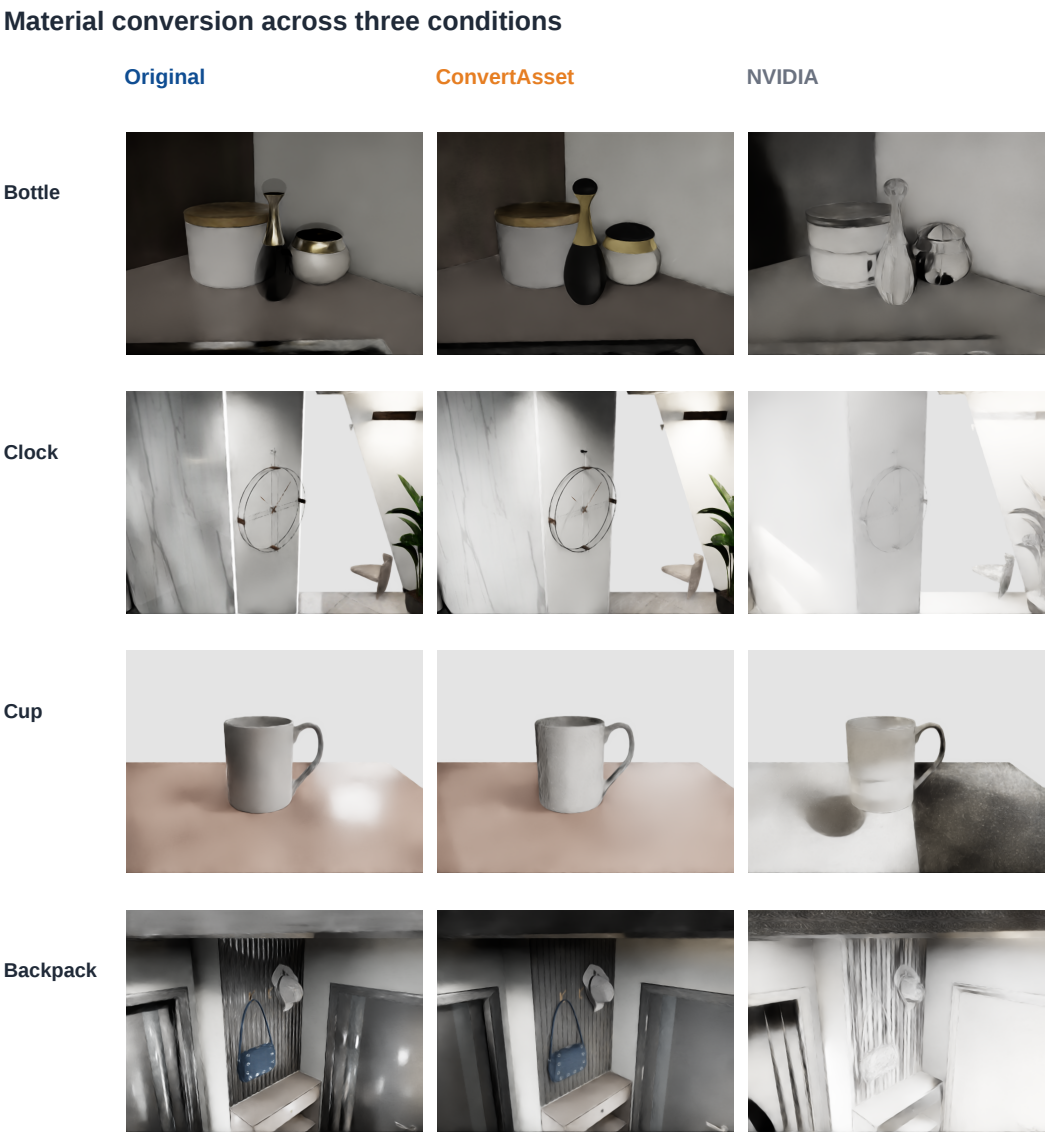

Figure 11: Matched material examples across original MDL, ConvertAsset, and NVIDIA conversion. Rows show a bottle, clock, cup, and backpack. These selected examples contain overlapping material effects; Table 4 reports the effect inventory.

C CONSTRUCTING THE CLEAN GROUNDING COMPARISONS

A point should follow the requested object when either the query or the object's position changes. The clean two-view construction makes this requirement explicit. Four repaired source pairs contain a bottle, clock, cup, and a bag annotated as backpack (Figure 12). We retain the annotation names in the queries. Cyclic pairing of distinct categories produces four boards. For each board, both categories are queried and both left/right arrangements are evaluated, giving $4 \times 2 \times 2 = 16$ paired queries per intervention and model.

Each source view keeps its original size. A 16-pixel gray gutter separates the two views on a canvas of size $(2W + 16) \times H$. The target box is translated by the panel offset; its shape is unchanged. Boxes are evaluation annotations and never appear in model inputs. The image center lies in the gutter, so a constant center response cannot solve this task.

C.1 QUERIES AND IMAGE REGIONS

Category prompts ask for the named object's location. Action prompts place the same category in an interaction description and request that object's location. Both ask for a single point in normalized 0–1000 coordinates over the entire canvas. The action variant tests action-framed object localization; it does not annotate a separate contact point on the object.

For target-window replacement, pixels inside the projected target rectangle come from the converted view, while the remaining target view and the distractor view remain original. Context replacement keeps that rectangle original and converts the rest of the canvas, including the distractor view. Full conversion uses converted versions of both views. The rectangle uses floor/ceiling integer bounds and includes any background within the bounding box. This makes the intervention an exact image-region crossover rather than a segmentation mask.

C.2 SOURCE CORRESPONDENCE

The source manifest records pair IDs, camera-derived boxes, image digests, and the composition recipe. The four clean pairs were selected from repaired captures with recorded material-resolution checks. The earlier raster archive contains 80 paired views of 16 targets, including a 30-pair close-view subset of 12 targets. Some earlier original-MDL images contain material-resolution errors. Those saved inputs support the readout analyses below; the clean window experiment uses the repaired four-pair set. Query and position permutations increase the tested comparisons without increasing the number of source objects.

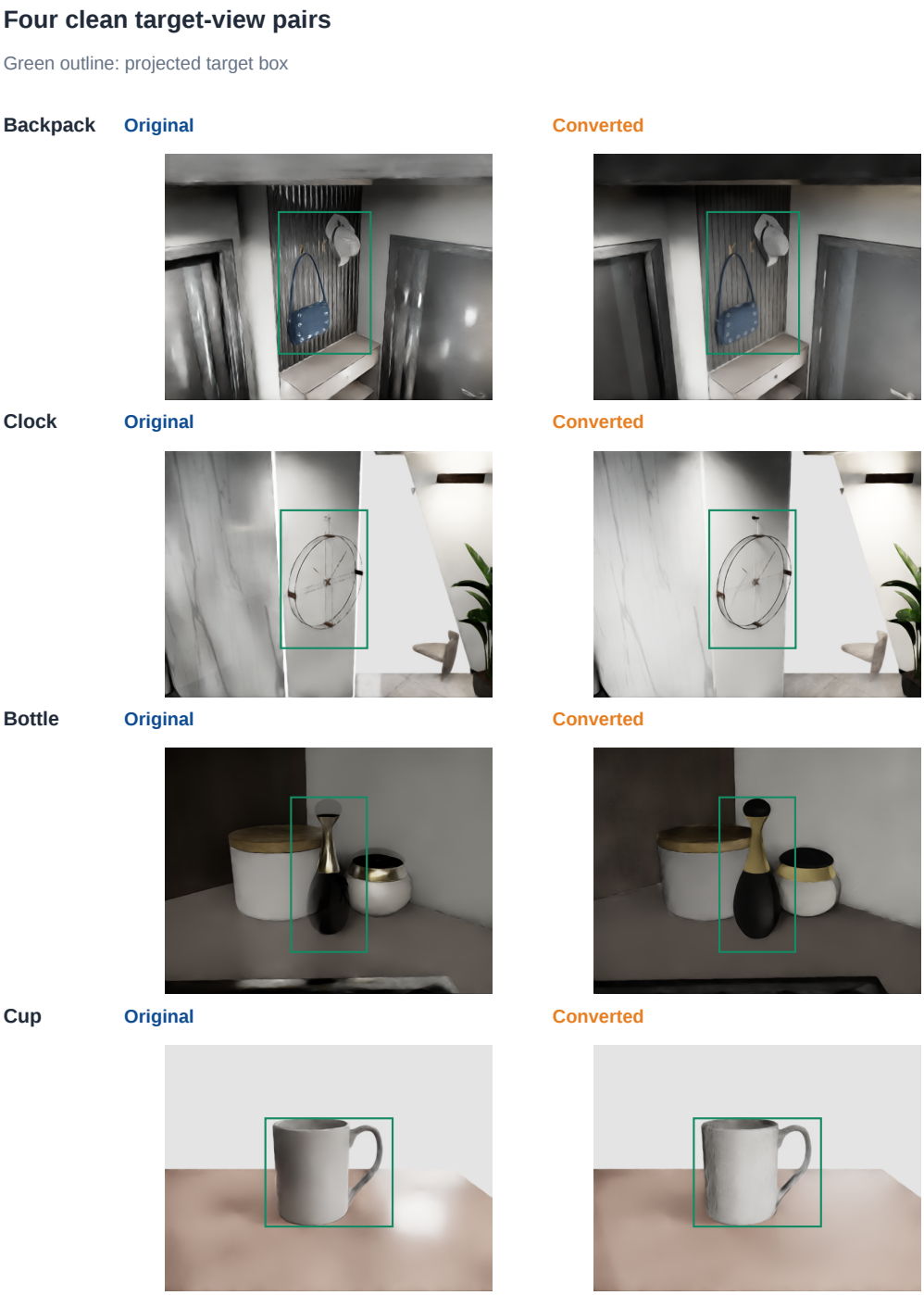

Figure 12: The four repaired source pairs used to construct the clean two-view experiment. Boxes show the projected evaluation regions and are absent from model inputs. Both targets and both panel positions are queried on each composed board.

D THE FULL SPATIAL READOUT COMPARISON

How much of a point-hit score comes from the way a location is requested and parsed? The readout experiment crosses pixel and normalized-1000 requests with JSON and two-line text. Table 5 retains every query in the denominator and reports numeric-output coverage beside end-to-end hits. A parsed point contributes to coverage even if it is outside the image; a hit must also lie inside the image and target box.

In the text condition, the prompt requests `Point: x, y` followed by `Answer: target category`. JSON requests `point_xy` and `answer`. Because the category is supplied in the prompt, repeating it measures answer adherence, not independent recognition. The clean two-view prompt requests only a point.

Model	Frame	Format	V_O	V_C	H_O (%)	H_C (%)
Gemma4-E4B	0–1000	text	30/30	30/30	90.0	96.7
Gemma4-E4B	0–1000	JSON	30/30	30/30	90.0	86.7
Gemma4-E4B	pixel	text	30/30	30/30	36.7	36.7
Gemma4-E4B	pixel	JSON	30/30	30/30	0.0	0.0
Qwen2.5-VL-7B	0–1000	text	29/30	29/30	10.0	13.3
Qwen2.5-VL-7B	0–1000	JSON	9/30	12/30	13.3	6.7
Qwen2.5-VL-7B	pixel	text	26/30	25/30	80.0	66.7
Qwen2.5-VL-7B	pixel	JSON	18/30	22/30	60.0	73.3
Qwen3-VL-2B	0–1000	text	30/30	30/30	86.7	86.7
Qwen3-VL-2B	0–1000	JSON	30/30	30/30	90.0	93.3
Qwen3-VL-2B	pixel	text	30/30	30/30	60.0	60.0
Qwen3-VL-2B	pixel	JSON	30/30	30/30	0.0	0.0
Qwen3-VL-8B	0–1000	text	30/30	30/30	93.3	96.7
Qwen3-VL-8B	0–1000	JSON	30/30	30/30	93.3	96.7
Qwen3-VL-8B	pixel	text	30/30	30/30	0.0	3.3
Qwen3-VL-8B	pixel	JSON	30/30	30/30	0.0	0.0
InternVL3-8B	0–1000	text	25/30	26/30	60.0	66.7
InternVL3-8B	0–1000	JSON	11/30	7/30	36.7	20.0
InternVL3-8B	pixel	text	30/30	29/30	16.7	10.0
InternVL3-8B	pixel	JSON	11/30	16/30	10.0	20.0
Molmo-7B-D	0–1000	text	30/30	30/30	90.0	96.7
Molmo-7B-D	0–1000	JSON	30/30	30/30	13.3	3.3
Molmo-7B-D	pixel	text	30/30	30/30	0.0	0.0
Molmo-7B-D	pixel	JSON	30/30	30/30	0.0	0.0

Table 5: Full readout matrix. V counts parsed numeric points among 30 queries; H is the end-to-end hit percentage over the same 30 queries. The normalized text rows are exact-ID subsets of the current 80-pair runs; the other rows use their recorded 30-pair runs.

For frame size K, (u, v) maps to image pixels $(uW/K, vH/K)$. Pixel requests use the point directly. Missing, malformed, and out-of-image points fail the hit metric. The parser also handles historical four-number text lines by taking the box center. Raw text remains available to distinguish an output-format failure from a coordinate mismatch. A uniformly applied alternative frame is reported as a diagnostic, rather than selected separately to fit each target.

The comparison reveals why both columns matter. A model can return numeric points consistently yet locate the target poorly under the requested frame. Conversely, low coverage can suppress end-to-end accuracy before localization is assessed. A material difference measured through either interface therefore needs the corresponding coverage and frame definition.

D.1 COORDINATE BASELINES

The 30-pair target-centered archive admits a simple center solution: the image center hits all 30 targets in both conditions when interpreted in pixels. An oracle box center also hits all targets when expressed and scored in the same frame. Interpreting either pixel baseline as normalized-1000 produces zero hits in this set; expressing the oracle in normalized-1000 restores all 30 hits. These checks explain the move to two-view boards and verify that the frame transformation is consequential even with

identical numeric outputs. A seeded random pixel baseline hits 5/30 original and 2/30 converted targets; its normalized reinterpretation yields 8/30 and 3/30. These are scorer checks, separate from model predictions. Figure 13 follows a response through parsing and coordinate interpretation before the point is scored.

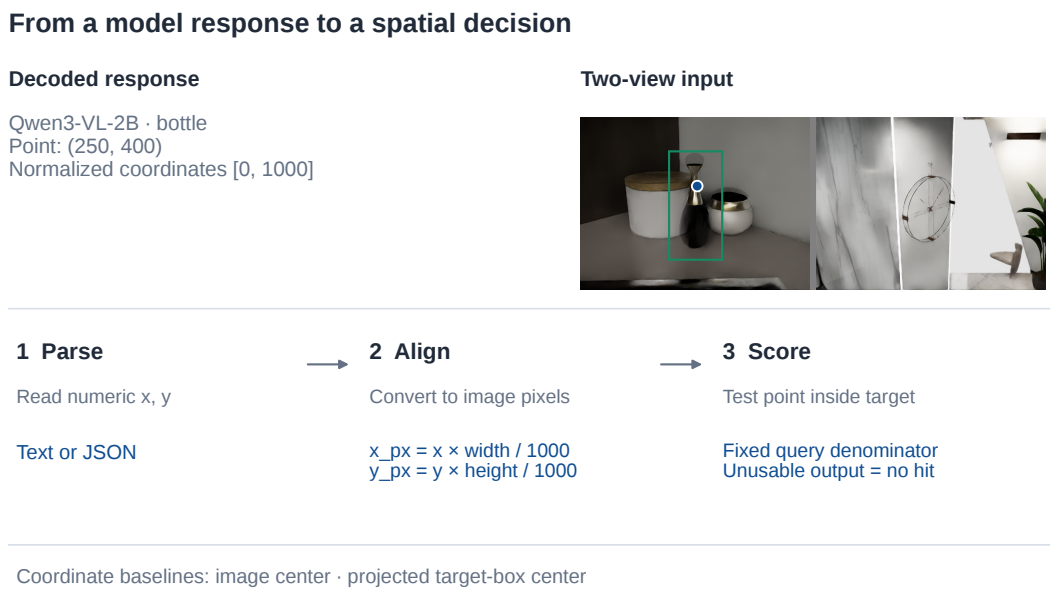

Figure 13: From a model response to a spatial score. Parsing establishes numeric coverage; the requested coordinate frame places the point on the image, and the target region determines a hit. Coordinate-only baselines expose the effect of changing the interpretation of identical output numbers.

E Additional Grounding Results and Decoding Variability

The clean two-view task tests query binding; the larger archive exposes behavior across repeated views and output conventions. Table 6 compares six configurations using normalized two-line output. Its 30-pair columns are exact-ID slices of the 80-pair inference runs, using the same saved responses and current image digests. The 80-pair set includes repeated views of 16 targets.

Model	Close views (30)		Mixed views (80)		Δ (pp)	Flips
Gemma4-E4B	90.0	96.7	70.0	65.0	-5.0	12/80
Qwen2.5-VL-7B	10.0	13.3	3.8	5.0	+1.2	5/80
Qwen3-VL-2B	86.7	86.7	53.8	50.0	-3.8	13/80
Qwen3-VL-8B	93.3	96.7	86.2	85.0	-1.2	7/80
InternVL3-8B	60.0	66.7	35.0	40.0	+5.0	16/80
Molmo-7B-D	90.0	96.7	92.5	93.8	+1.2	5/80

Table 6: Archived raster readout: end-to-end point hits (%). Each pair of columns reports Original then Converted. Differences and flips use all 80 pairs; all queries remain in the denominator.

The close views and mixed views give different pictures of performance. Gemma4 reaches 90.0/96.7% on close views but 70.0/65.0% over all 80 pairs. InternVL3 gains five percentage points in the larger set while changing hit status on 16 pairs. Qwen3-VL-2B changes status on 13 pairs despite a net decrease of three hits. The larger archive thus reinforces the distinction between the number of hits and the set of views on which those hits occur.

E.1 THE EARLIER CLEAN POOL AND DEVELOPMENT PROBES

Table 7 reports the 15-pair pool spanning nine targets. This pool precedes the repaired four-source construction. Gemma4 changes from 8 to 6 hits, Qwen3-VL-8B from 12 to 13, and Qwen2.5-VL records no hits under the normalized frame. These outcomes remain tied to their saved inputs.

Model	N	V_O	V_C	H_O	H_C	Flips
Gemma4-E4B	15	15	15	8	6	4
InternVL3-8B	15	13	11	3	3	2
Qwen2.5-VL-7B	15	14	15	0	0	0
Qwen3-VL-2B	15	15	15	3	3	2
Qwen3-VL-8B	15	15	15	12	13	1

Table 7: The 15-pair grounding pool under normalized two-line readout. V counts numeric outputs and H counts hits, both over $N = 15$ per condition. Flips count paired changes in hit status.

The four-pair pilot and 14-pair zoom study capture the development of the readout protocol (Table 8). The Gemma4 pilot requested pixels but its normalized reinterpretation yields four original and three converted hits. Qwen2.5's early JSON responses contain no parseable points; the two-line prompt produces points but still exposes a scale mismatch. In the zoom study, Gemma4 yields 11 and 13 normalized hits, while Qwen2.5 yields three in each condition. Its pixel reinterpretation instead gives nine and six. Recomputing these rows over every query preserves format failures in the same denominator as localization failures.

Run	N	V_O/V_C	Requested hits O/C	Other frame O/C
Gemma4 pilot	4	4/4	0/0	4/3
Qwen2.5 JSON pilot	4	0/0	0/0	0/0
Qwen2.5 text pilot	4	3/4	0/0	2/2
Gemma4 zoom	14	14/14	11/13	1/1
Qwen2.5 zoom	14	14/13	3/3	9/6

Table 8: Earlier pilot and zoom runs, rescored from saved predictions with fixed denominators. Entries separated by a slash are Original/Converted counts. The Gemma4 pilot requested pixels; all other rows requested normalized 0–1000. "Other frame" applies the alternative interpretation uniformly.

E.2 READING INDIVIDUAL RESPONSES

The recorded responses include several distinct cases: a Gemma4 faucet point changes from a normalized hit to a miss while the answer remains "faucet"; Qwen2.5 returns a truncated "fauc" in a faucet example; and a clock response changes from a pixel-frame miss to a hit. The examples separate point movement, truncated answers, and coordinate interpretation. Some archived inputs contain MDL compilation failures; their scores therefore reflect rendering faults as well as material differences. Figures 14 and 15 show matched qualitative rerenders after MDL dependency repair. Historical predictions and scores remain tied to their original inputs and are not transferred to these new images.

E.3 STOCHASTIC SPATIAL READOUT

Three runs at temperature 0.7 evaluate each of the 60 images in the 30-pair archive. Seed distance is the mean of the three pairwise pixel distances on one image; hit status is variable when the three binary outcomes differ. All queries in each model group have three numeric points.

Qwen3-VL-8B combines the smallest mean seed distance, 7.66 pixels, with zero hit-status variability. Qwen3-VL-2B and Gemma4 have larger distances, 24.34 and 27.07 pixels, and variability of 16.7% and 10.0%. A point can therefore move without crossing the target boundary, while another movement changes the binary score. The primary material comparisons use greedy decoding.

Model	Complete triples	Seed distance (px)	Hit variability (%)
Qwen3-VL-8B	60/60	7.66	0.0
Qwen3-VL-2B	60/60	24.34	16.7
Gemma4-E4B	60/60	27.07	10.0

Table 9: Stochastic readout on identical images. Complete triples count queries with three numeric points. Seed distance is measured in image pixels; hit-status variability counts queries whose three outcomes differ.

E.4 Pairing and uncertainty

Point-hit differences use all paired queries. Target-cluster bootstrap resamples an object's views together; clean two-view resampling groups query and position permutations by board. Boards share source targets, so those intervals describe board variability. Displacement uses only pairs with two numeric points and records that count separately. Coordinates are transformed to image pixels before distances are computed, and even-sized medians average the middle two values.

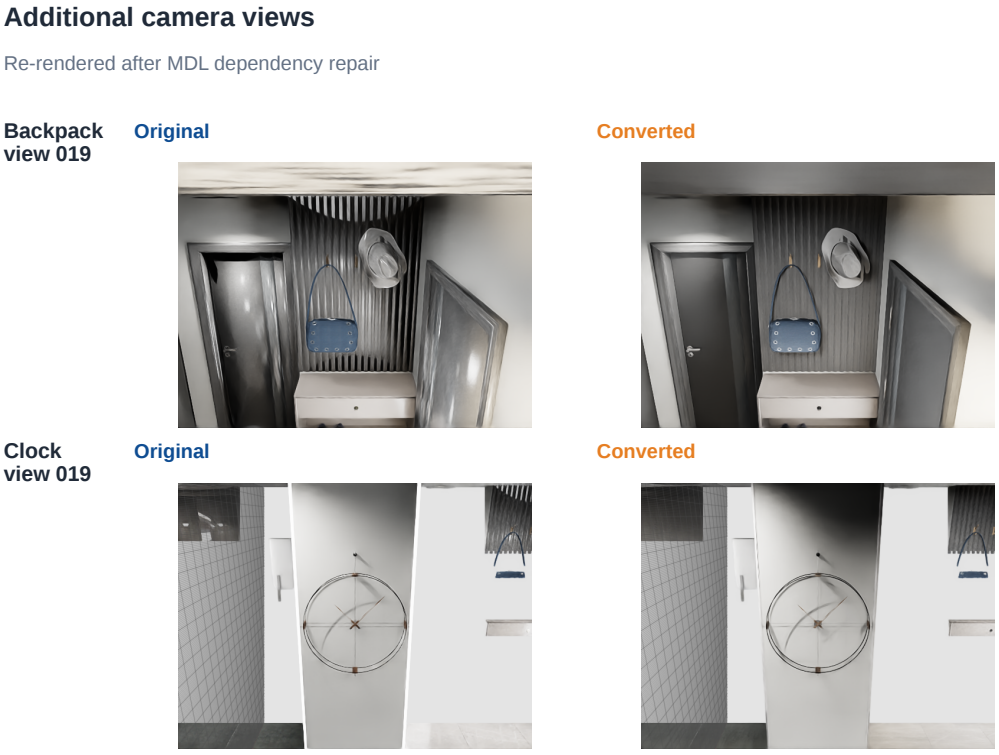

Figure 14: Additional target views after MDL dependency repair (1/2). Both material conditions are rerendered at the saved camera settings. These qualitative images are separate from the historical inference inputs and carry no prediction overlays.

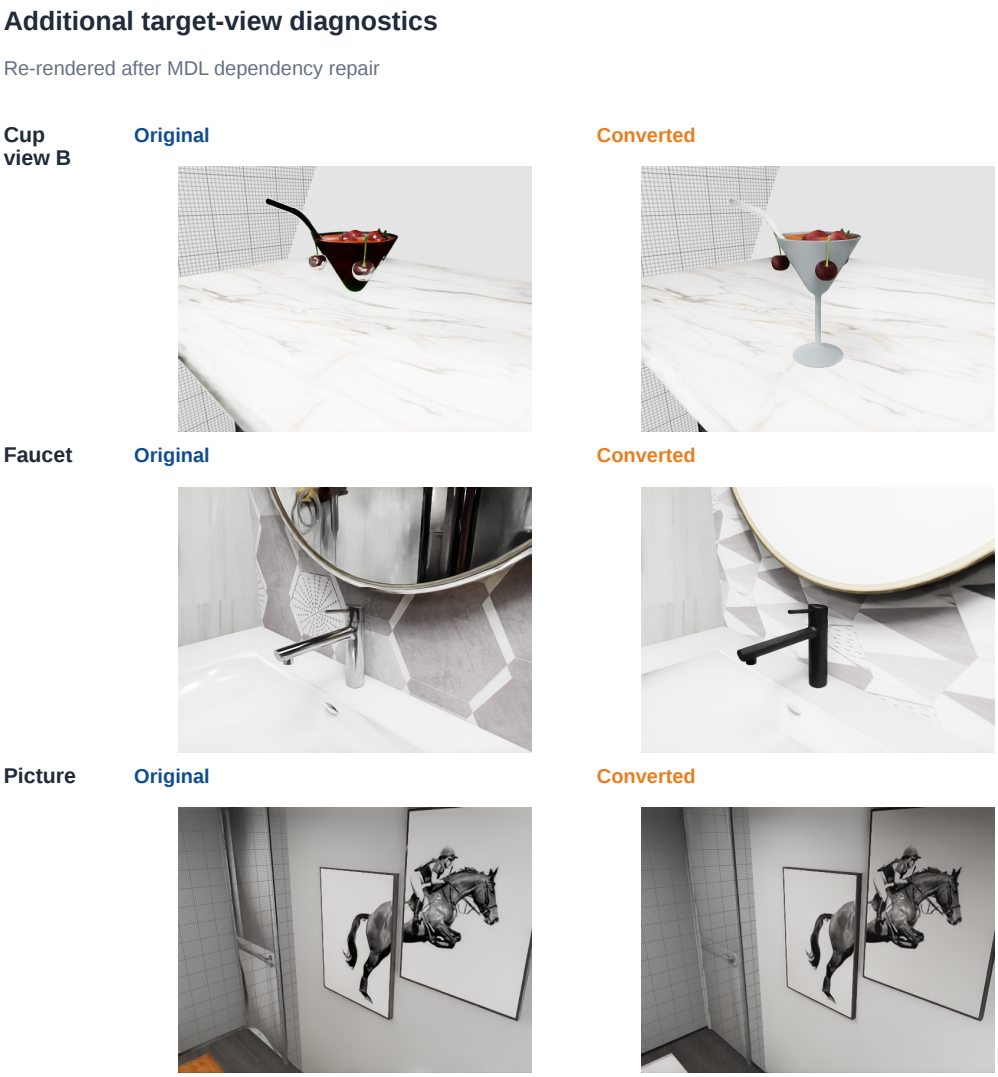

Figure 15: Additional target views after MDL dependency repair (2/2). Both material conditions are rerendered at the saved camera settings. These qualitative images are separate from the historical inference inputs and carry no prediction overlays.

F Navigation Outcomes, Runtime Checks, and Additional Cases

Which aspects of navigation change together? The full metric run contains 99 paired episode IDs across KuJiaLe scenes 0031, 0036, and 0066, with 33 per scene. Original and converted runs share instructions, start states, episode definitions, and agent settings. Table 10 adds oracle success and paired intervals to the main results.

Metric	Original	Converted	Δ	95% paired CI
SR (%)	52.530	48.480	-4.040	[-15.150, 7.070]
SPL (%)	47.390	42.980	-4.410	[-14.170, 5.140]
Oracle success (%)	66.670	61.620	-5.050	[-15.150, 5.050]
Navigation error	3.680	3.631	-0.049	[-0.646, 0.539]
Trajectory length	6.975	7.060	+0.084	[-0.737, 0.930]
Action steps	233.647	275.394	+41.748	—
Fall rate	0.020	0.020	+0.000	—
Stuck rate	0.000	0.000	+0.000	—

Table 10: Navigation metrics over all 99 episode pairs. Differences are Converted minus Original. SR, SPL, and oracle success use percentages; NE and TL retain the evaluator's distance units. Intervals are the recorded 95% paired bootstrap intervals; dashes denote metrics without a retained interval. Fall rate (FR) and stuck rate (StR) are fractions of episodes with the corresponding recorded failure reason.

SR records completion under the evaluator's success rule. NE is the final goal distance, TL is executed path length, and SPL weights success by path efficiency, $SL^*/\max(L, L^*)$, where L^* is reference path length. Oracle success records whether the episode reaches the success region along its path. SR and SPL decrease in aggregate, while mean NE decreases slightly and TL increases. The measures therefore capture different properties of the executed episodes. The paired intervals span zero; the 17 success-to-failure and 13 failure-to-success transitions describe the observed redistribution directly.

Each episode has one metric rollout per material condition. The observed turnover includes any runtime variability, for which these runs provide no same-condition repeat estimate. Grounding and navigation use different scenes and models, so they provide complementary observations rather than a measured causal chain from a particular pointing error to an episode outcome.

F.1 Loading and rendering the scenes

Separate fresh-launch checks exercise each of the three scenes three times per condition, totaling 18 launches. All complete successfully (Table 11). Ready time and warm-up frame rate are similar at this scale. These measurements concern initialization and rendering; they are distinct from the 198 navigation rollouts.

Condition	Loads	Ready time (s)	Warm-up FPS	GPU memory (MB)
Original	9/9	13.95 [11.64, 16.28]	2.36 [2.01, 2.70]	3807 [3748, 3862]
Converted	9/9	14.12 [12.31, 16.24]	2.42 [2.08, 2.73]	3829 [3689, 3971]

Table 11: Fresh-launch scene checks. Each condition has nine successful loads across three scenes. Values reproduce the recorded mean and interval summaries for ready time, warm-up frame rate, and GPU memory (bootstrap 95% CIs).

F.2 Twelve paired recorded examples

The following panels show selected episode reruns from all three scenes, including asymmetric outcomes, divergent routes with the same binary outcome, and similar-route examples. Each episode retains original and converted start, middle, and ending frames with path overlays. This temporal context shows how a final observation relates to the route that preceded it.

In episode 898, the converted route loops near the starting room while the original route extends farther through the interior. Episode 891 shows a wider converted detour, whereas episode 598 reaches

similar kitchen views in both recordings. These examples make the variety of paired trajectory changes visible beyond the final success indicator.

The full 99-pair metric runs were video-free. These recorded reruns share episode IDs with that evaluation but have their own trajectories and outcomes; the figures do not substitute rerun scores for the full-run measurements. Keeping both conditions visible also avoids treating a changed path as necessarily a changed success status.

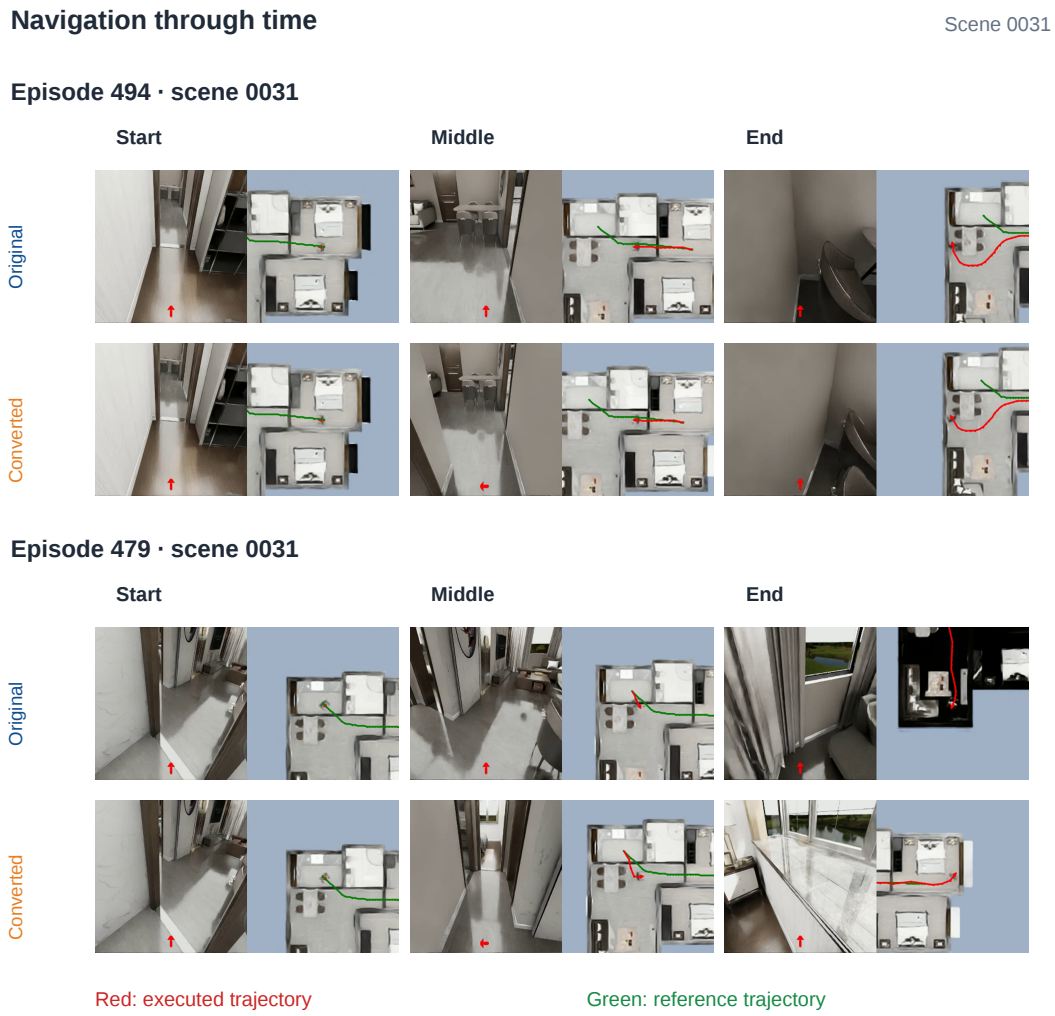

Figure 16: Paired navigation reruns: episodes 494 and 479 in scene 0031. Rows show Original and Converted; columns show the start, middle, and end. Red maps the executed path and green the reference path. These recordings are separate from the 99-pair metric runs.

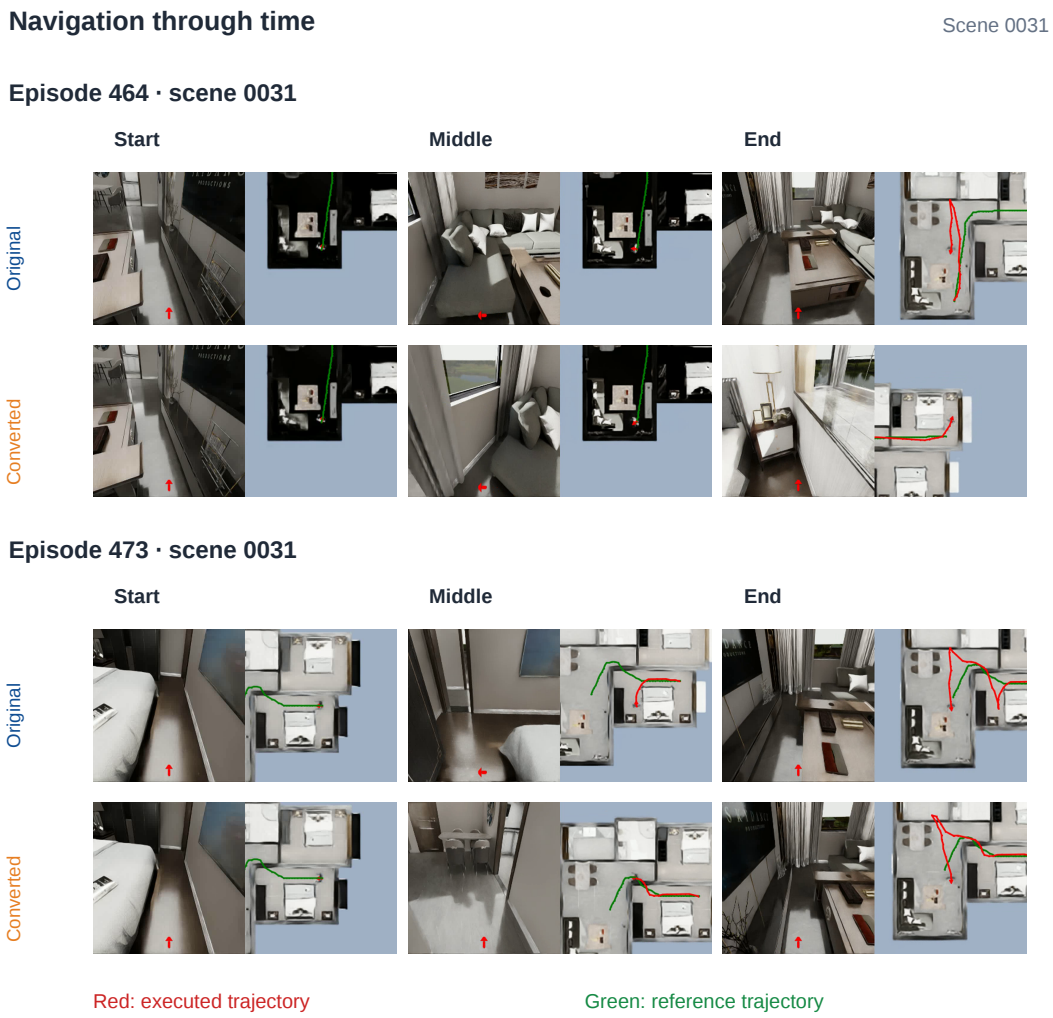

Figure 17: Paired navigation reruns: episodes 464 and 473 in scene 0031. Rows show Original and Converted; columns show the start, middle, and end. Red maps the executed path and green the reference path. These recordings are separate from the 99-pair metric runs.

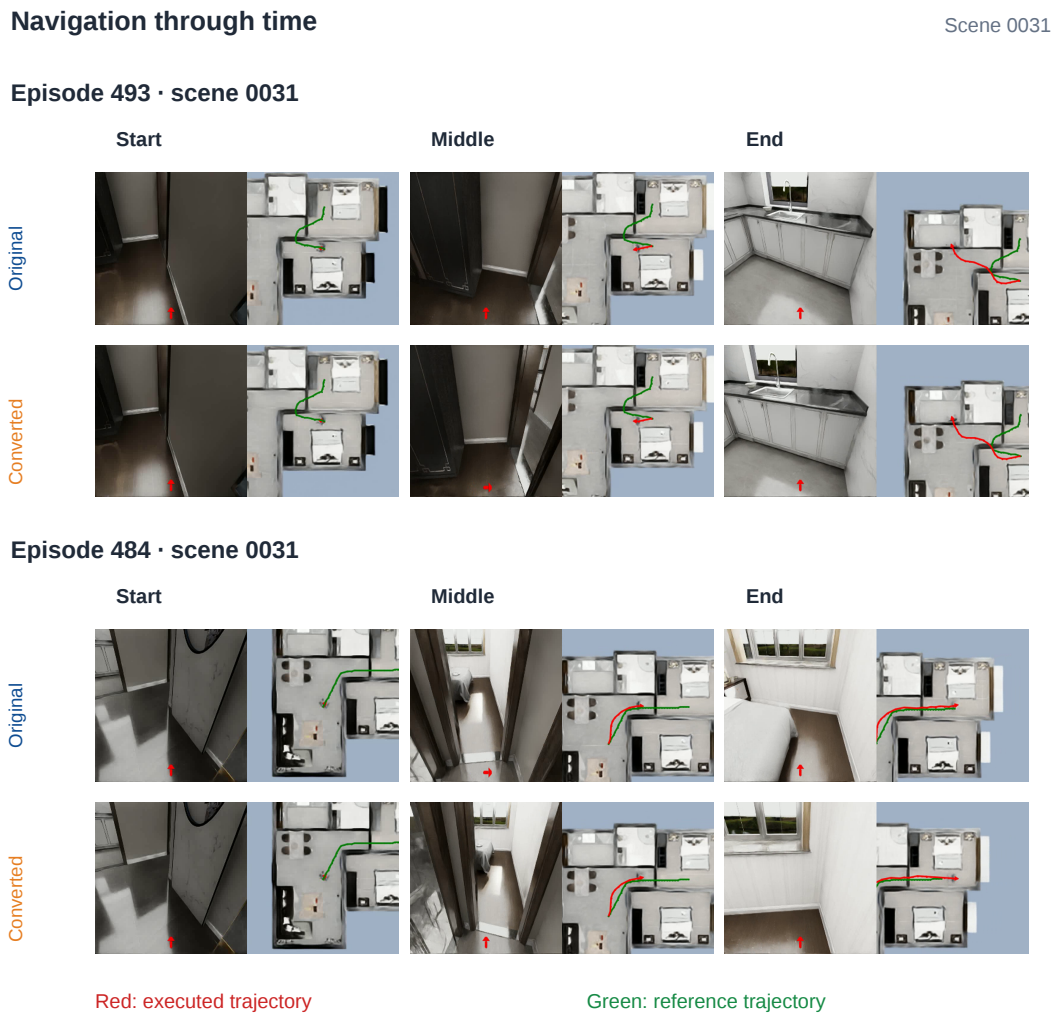

Figure 18: Paired navigation reruns: episodes 493 and 484 in scene 0031. Rows show Original and Converted; columns show the start, middle, and end. Red maps the executed path and green the reference path. These recordings are separate from the 99-pair metric runs.

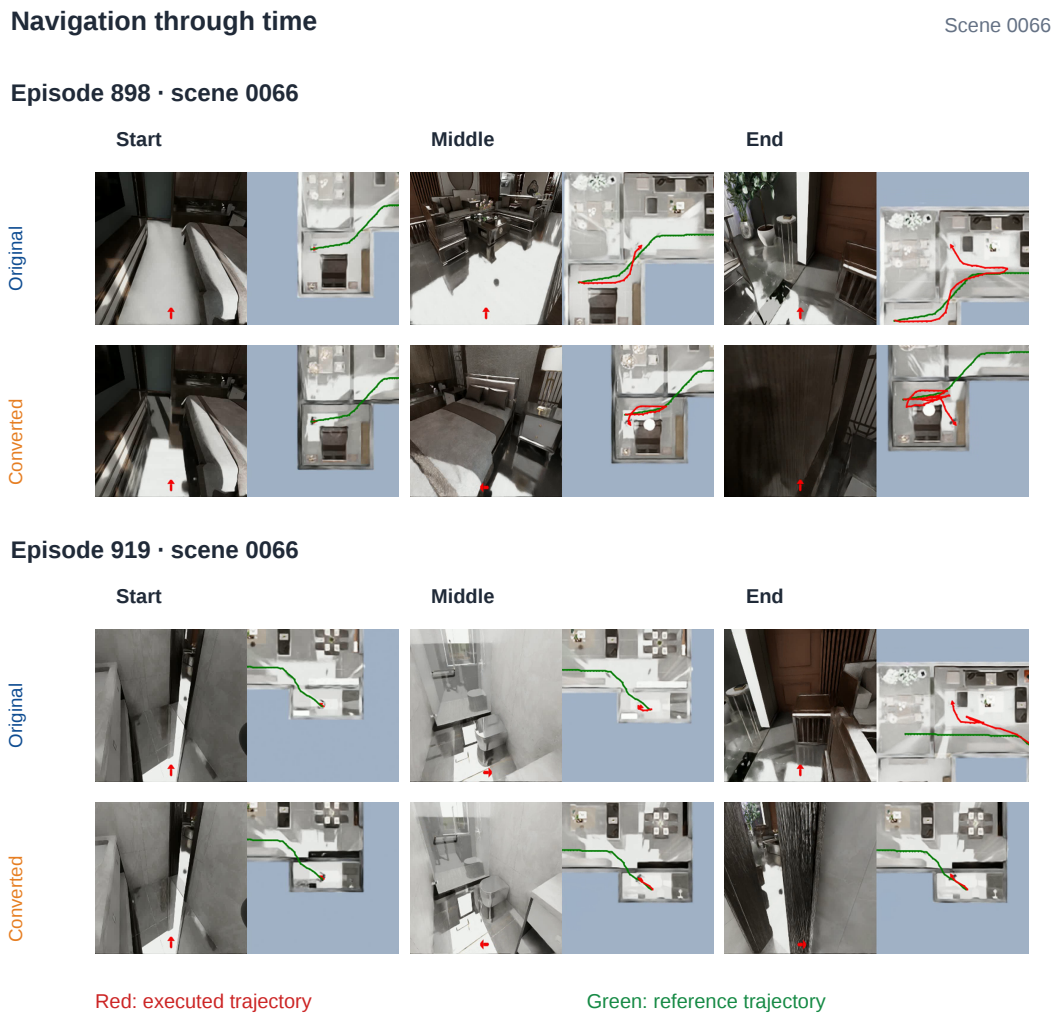

Figure 19: Paired navigation reruns: episodes 898 and 919 in scene 0066. Rows show Original and Converted; columns show the start, middle, and end. Red maps the executed path and green the reference path. These recordings are separate from the 99-pair metric runs.

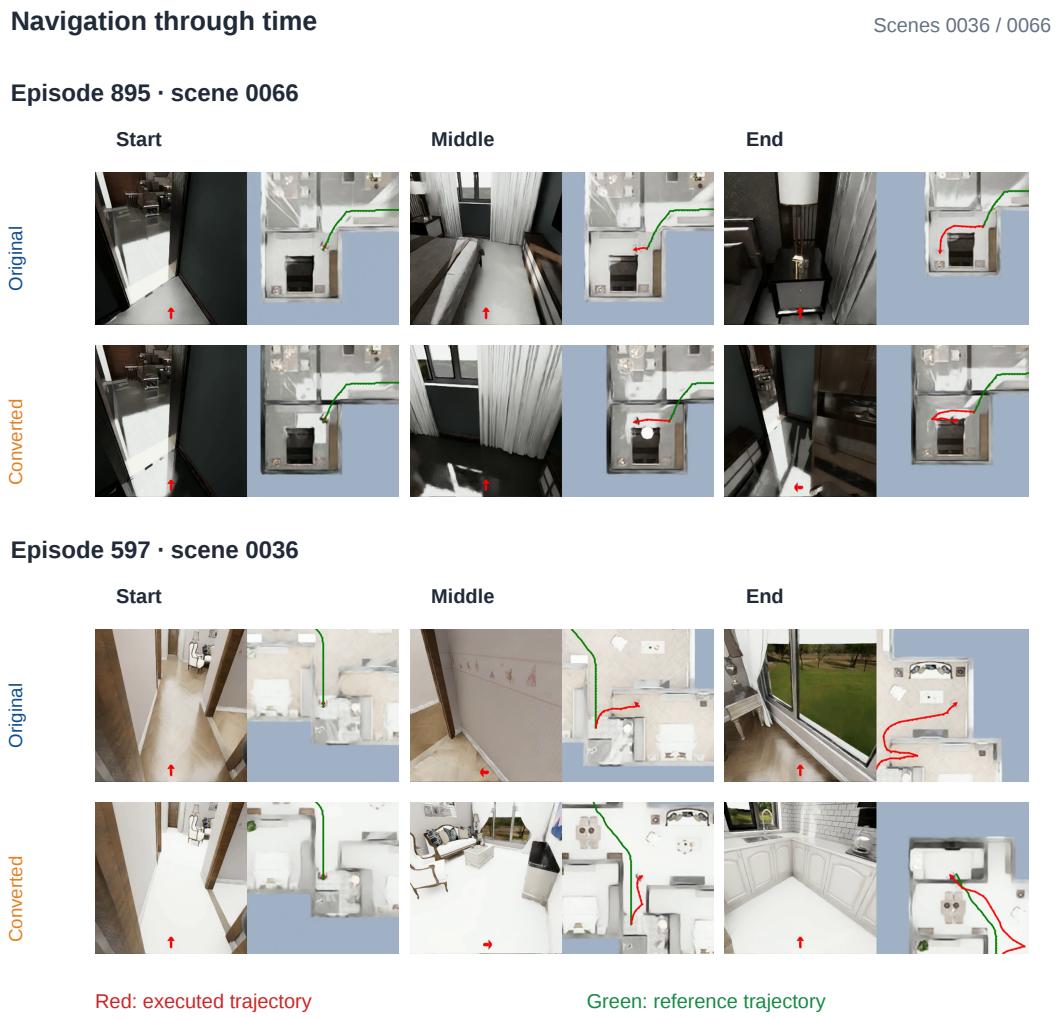

Figure 20: Paired navigation reruns: episodes 895 (scene 0066) and 597 (scene 0036). Rows show Original and Converted; columns show the start, middle, and end. Red maps the executed path and green the reference path. These recordings are separate from the 99-pair metric runs.

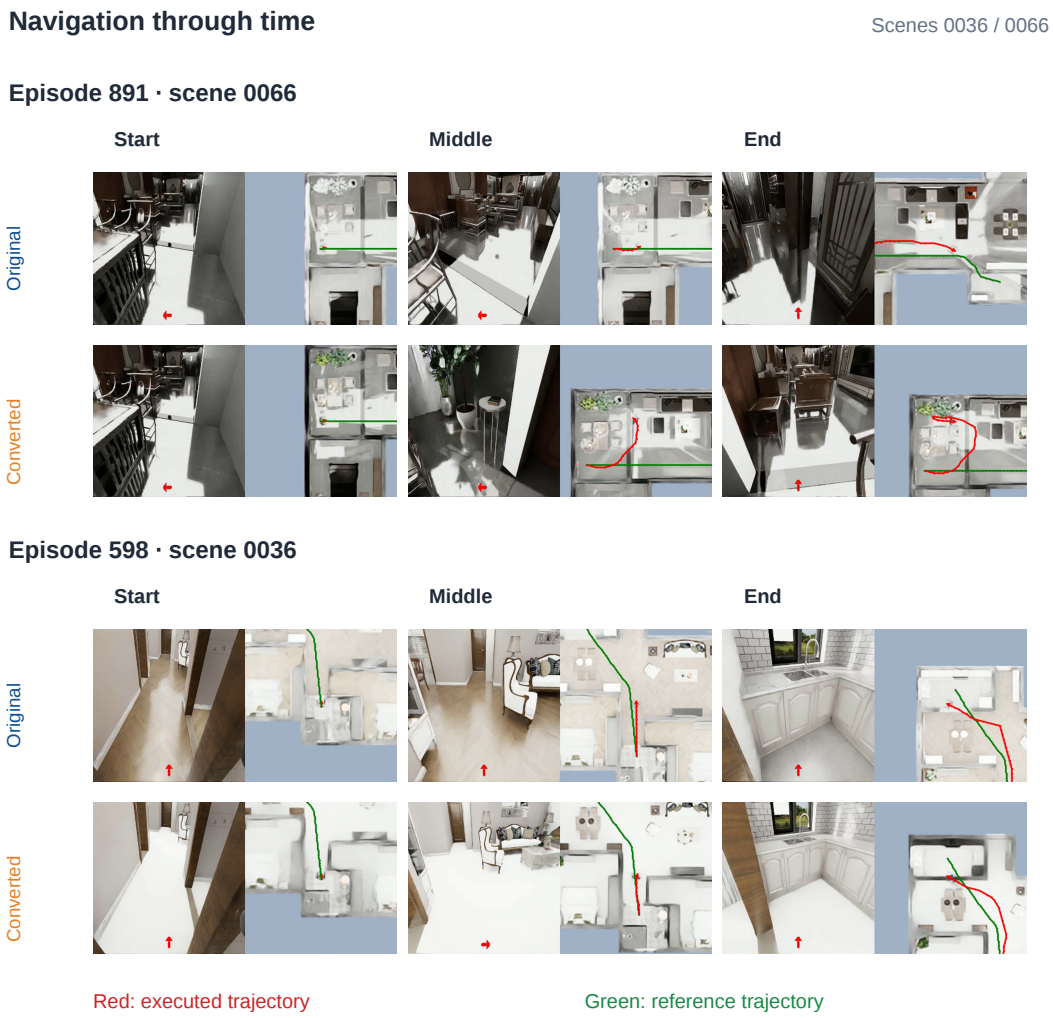

Figure 21: Paired navigation reruns: episodes 891 (scene 0066) and 598 (scene 0036). Rows show Original and Converted; columns show the start, middle, and end. Red maps the executed path and green the reference path. These recordings are separate from the 99-pair metric runs.

G REPRODUCING THE COMPARISONS

The analysis starts from saved predictions and paired measurements. Each prediction retains the raw response, input digest, prompt, model configuration, and target box. Image-composition manifests retain source identities and pixel operations. Navigation records preserve episode IDs separately from the selected recorded examples. The appendix tables are regenerated from these records, including fixed-denominator rescoring of the earlier pilot and zoom runs.

G.1 MODEL ENVIRONMENTS

Qwen and InternVL archive runs use Transformers 5.2 and PyTorch 2.10. The Gemma4 four-bit checkpoint uses its Unsloth-compatible environment. Molmo uses Transformers 4.46.3 and PyTorch 2.5.1 for its remote generation code. InternVL uses the remote chat model with one 448-pixel square image tile. Image preprocessors are retained with the model configurations. Greedy decoding uses a 64-token output budget in the historical grid and 128 tokens in the two-view probes. The stochastic study instead uses temperature 0.7 and three recorded seeds. Checkpoint revisions, prompts, and runtime patches accompany each run.

G.2 REBUILDING TABLES AND FIGURES

The ICLR build consumes corrected analysis results and figure specifications. The appendix table builder independently reads the saved proxy, grounding, material, and navigation records and writes a source-hash manifest. Rebuilding requires no new model inference or simulator execution:

```
python paper/venues/iclr27/appendix/build_tables.py
make -C paper iclr27-check
```

The source package separates raw measurements from generated figures and tables. Source manifests make it possible to verify input correspondence before recomputing paired metrics; figure manifests identify the render or video frame used in each panel.

G.3 ASSETS AND ACCESS

The experiments use third-party scenes and model checkpoints, whose source terms govern redistribution. The manuscript package includes analysis code, tables, and reproduction instructions. Private machine paths and author-identifying metadata are excluded from the anonymous PDF. Raw scene trees, model checkpoints, and videos remain separate from the manuscript artifacts.