THE GEOMETRY OF CONSISTENT INTERVENTION IN LAYERED 3D WORLDS

Anonymous authors
Paper under double-blind review

ABSTRACT

Object-level edits to a 3D scene can break support and containment relations, while propagating an edit through the full dependency graph can move objects unnecessarily. We formulate translational edit consistency in the object-position space $W = \prod_i \mathbb{R}^3$, where structural relations define a dependency-consistent feasible set. This shifts 3D editing from a collection of local transformation rules to a structured intervention problem over the state of the world. For a fixed support assignment, this set is a convex polyhedron defined by vertical support equalities, support-surface inequalities, and containment bounds. We characterize intervention-induced constraint violation and its relation to distance from feasibility. With a unique active supporter held at the edited position, horizontal support repair requires a dependent object to move precisely when its center of gravity leaves that support surface. A feasible, closed convex target then admits a unique minimum-displacement response within the fixed stratum. On InteriorGS, local geometry predicts repair requirements with AUROC 0.877, compared with 0.694 for graph fan-out. On a separate 340-edit benchmark, the minimal-intervention strategy moves 59% fewer objects than closure propagation and achieves zero incremental residual in 334 cases. Across 142 translation, deletion, and container-constrained editing cases, the practical solvers meet the respective consistency criteria in all cases. The aggregate constraint residual $R_{\mathcal{F}}$ separates intact from synthetically corrupted states with AUROC 0.998. Cross-dataset evaluations and an intervention-margin sweep further examine transfer and the balance between moved-object count and geometric residual. The framework provides provides a step toward world models that do not merely generate editable objects, but maintain coherent structured worlds under intervention.

1 INTRODUCTION

As 3D world models move from one-shot generation toward persistent interaction, the central challenge is no longer only how to generate or edit an object, but how the surrounding world should respond to that intervention. Moving a table, for example, may require the objects resting on it to move, while nearby but unrelated objects should remain unchanged. Existing editing systems typically address such cases through local transforms, dependency-graph propagation, or post-hoc physical correction. These strategies are useful, but they do not provide a general account of what constitutes a valid edited world state, when an intervention should propagate beyond the edited object, or how far that propagation should extend. As interactive world models increasingly support object-level manipulation, this missing notion of consistency under intervention becomes a fundamental problem rather than a purely editing-specific implementation detail.

This work studies that problem through the geometry of structured world states. A scene is represented on an object-position manifold $W = \prod_i \mathbb{R}^3$, while support and containment relations define a dependency-consistent feasible set $\mathcal{F} \subset W$. Within this view, object dependencies induce geometric constraints on admissible world states, and an edit becomes an intervention that may move the state outside this feasible set. The resulting formulation separates three questions that are often conflated in heuristic editing pipelines: what geometric structure characterizes valid world states, how the consequences of an intervention can be predicted from local constraint geometry, and which dependent objects must actually respond to restore consistency. A central implication is that dependency does not imply propagation: dependencies determine where an intervention may

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.

have consequences, whereas the geometry of the current state and the direction of the intervention determine which constraints are actually violated and which objects therefore need to move.

Contributions. We develop a dependency-aware editing framework in which the geometry of a scene's support and containment structure turns three practical questions— *when* an edit breaks the scene's rules, *what* must move to restore consistency, and *how* to restore it with minimal intervention—into computable, predictive quantities rather than heuristics. Concretely:

(1) Formalization with predictive content. World states are points on the object-position manifold W; support and containment rules define the dependency-consistent feasible set $\mathcal{F}$ (a stratified space of support equalities and containment inequalities). The definitional statement—consistent editing is projection of the naive edit onto $\mathcal{F}$ (T2.2)—is a reformulation of classical constrained optimization; its value is that it makes scene failure computable, which the items below exploit.

(2) When: an analytic violation bound . We prove a lower bound on the violation of unconstrained edits: vertical displacement of a supporter breaks its support edges by at least $|v_z|$, and horizontal displacement that carries a top's center of gravity off the support surface violates by the overhang distance. The bound depends only on the edit geometry and predicts, before any repair, whether an edit will break rules and how large the repair must be—AUROC 0.877 for "needs repair" across 680 edits, vs. 0.694 for graph fan-out.

(3) What: the exact propagation set. The set of supported objects that actually move under an edit is exactly the tops whose center of gravity leaves the support surface—not the full dependency closure. This characterization is exact (F1 1.00 vs. 0.43 for the closure) and corrects the over-approximation inherent in graph propagation.

(4) How: minimal-intervention propagation. Practical $\mathcal{F}$-projection solvers (convex QP projection for translation, iterative settling for deletion, container clamping), built on a cleaned dependency graph with validated cleaning rules. They move exactly the propagation set above, reducing moved objects by 59% at near-zero incremental violation (98% of edits; residual 0.014 m vs. 0.001 m for the closure).

(5) A verification diagnostic. $\mathrm{dist}(w, \mathcal{F})$—the computable distance to the feasible set—is a task-agnostic consistency check for generated or edited scenes, separating intact from corrupted states at AUROC 0.998.

(6) A subdifferentiable training signal. $\mathrm{dist}(w, \mathcal{F})$ is piecewise linear in the world state (differentiable almost everywhere, subdifferentiable everywhere) and usable as a plug-in regularizer: on a partially supervised repair task it improves test consistency by 20% and accuracy by 27% .

(7) Empirical validation. 142 editing cases across real indoor 3DGS scenes (translate/delete/container): projected edits achieve 100% zero-violation; naive and follow baselines violate at the predicted magnitudes.

2 Related Work

Geometric views of generation and editing. Recent work has increasingly interpreted generation and editing through geometric structure. Karczewski et al. (2026) study diffusion models through an information-geometric view of latent spacetime and derive geodesic notions of edit distance. Lu et al. (2026) use information geometry to characterize adaptive sampling in graph diffusion, while GeoLatent (Yang et al., 2023) constructs a Riemannian metric over the latent space of deformable 3D shape generators. Related analysis studies how score models recover geometric structure under the manifold hypothesis (Li et al., 2026b), and RemEdit (Adhikarla & Davison, 2026) applies Riemannian geometry to diffusion-based image editing. These works concern the geometry of latent spaces, data manifolds, or generative trajectories. In contrast, we study the geometry of the world state itself: a multi-object position space whose feasible configurations are restricted by explicit structural dependencies.

Controllable 3D generation and world models. Recent 3D generative systems provide increasingly explicit control over geometry and spatial configuration. SpaceControl (Fedele et al., 2026) introduces test-time spatial control for 3D generative modeling, allowing spatial constraints to steer

generated 3D content without retraining. HunyuanWorld 1.0 (HunyuanWorld Team, 2025) generates immersive and explorable 3D worlds from text or image inputs and represents a broader shift toward interactive world generation. These systems improve controllability and world construction, but object-level control alone does not determine how a local edit should propagate through structural dependencies among multiple objects. Our focus is therefore not only on transforming a target object, but on characterizing the consequences of that intervention for the surrounding structured world.

Constraint- and physics-aware scene generation and editing. A complementary line of work incorporates geometric, relational, or physical constraints directly into scene generation and optimization. SPREAD (Li et al., 2026a) introduces geometry-aware diffusion with differentiable guidance for spatial and physical reasoning. HOG-Layout (Jiang et al., 2026) combines hierarchical 3D scene generation with constraint-based optimization and interactive editing, while LiteReality (Huang et al., 2025) reconstructs graphics-ready structured 3D scenes from RGB-D scans. DiffuScene (Tang et al., 2024) models indoor scene synthesis with diffusion over structured scene representations, and Scenethesis (Ling et al., 2026) uses a language-and-vision agentic pipeline to generate and refine 3D scenes under geometric and physical plausibility constraints. These methods demonstrate the importance of structure and physical plausibility in scene generation and editing, but their primary objective is to produce or optimize a consistent configuration. Our problem is different: given an existing structured world and a local intervention, we ask which constraints become violated, which dependent objects must respond, and how consistency can be restored with minimal displacement.

Constraint projection and intervention propagation. Projection methods provide classical tools for enforcing constraints once a feasible set has been specified. Dykstra's algorithm (Dykstra, 1983), convex-set projection methods (Combettes, 1997), and metric projection theory (Moreau, 1965) provide standard machinery for constrained correction. Our contribution is not a new generic projection algorithm. Instead, we study the structure that precedes projection: how support and containment dependencies define the feasible world-state geometry, how a local intervention induces predictable constraint violation, and how that geometry determines which dependencies must actually propagate. This distinction between *dependency* and *intervention-dependent propagation* is the central gap addressed by this work.

3 Formulation

3.1 Object-Position Manifold W

A scene has N objects with fixed shapes and orientations. For translational editing, object i has position $p_i \in \mathbb{R}^3$, and the world state is $w = (p_1, \dots, p_N)$ in

$$W = \prod_{i=1}^{N} \mathbb{R}^3, \qquad \dim W = 3N. \tag{1}$$

We equip W with the standard Euclidean product distance

$$d_W(w, w') = \left(\sum_{i=1}^{N} \|p_i - p_i'\|_2^2 \right)^{1/2}. \tag{2}$$

Rigid motions additionally require orientation-dependent contact geometry.

3.2 Dependency-Consistent Feasible Set $\mathcal{F}$

A cleaned dependency graph encodes support and containment relations. Let $\mathcal{A}(w)$ denote the selected active support assignment at state w; it is held fixed during a within-stratum edit and abbreviated as $\mathcal{A}$. For each support edge $(j \to i) \in \mathcal{A}$, we impose a vertical equality $E_{ji} : z_i^{\mathrm{bot}} = z_j^{\mathrm{top}}$ and a support-surface condition S_{ji}: the horizontal center of gravity of i lies inside the top projection of j. Containment constraints C_o require the contained object's bounding box to lie within its container. Shapes and orientations remain fixed, so these support-surface and containment conditions are linear inequalities in the object positions.

Let $\mathcal{B}$ be the set of objects whose ground contact is enforced and write $H_i : z_i^{\text{bot}} = 0$ for $i \in \mathcal{B}$ (take $\mathcal{B} = \varnothing$ when no grounding constraint is imposed). For a fixed assignment, define

$$\begin{aligned}\mathcal{F}_{\mathcal{A}} &= \bigcap_{(j,i)\in\mathcal{A}} \{E_{ji} \wedge S_{ji}\} \cap \bigcap_{o} \{C_o\} \cap \bigcap_{i\in\mathcal{B}} \{H_i\} \\ &= \{w : A_{\mathcal{A}} w = b_{\mathcal{A}},\ G_{\mathcal{A}} w \le h_{\mathcal{A}}\}.\end{aligned} \tag{3}$$

Here $A_{\mathcal{A}}$ collects the vertical support equalities, and $G_{\mathcal{A}}$ collects both support-surface and containment inequalities. When grounding is enforced, its equalities are included in $A_{\mathcal{A}}$ using the fixed ground plane $z = 0$. Thus $\mathcal{F}_{\mathcal{A}}$ is a closed convex polyhedron, assumed nonempty whenever projection is used.

For fixed object inventory, the full feasible space is $\mathcal{F} = \bigcup_{\mathcal{A}} \mathcal{F}_{\mathcal{A}}$ over admissible support assignments, and need not be convex. We use stratum to denote a fixed-assignment polyhedral component; its boundary faces may have lower dimension. Support creation and reassignment switch these components. Deletion additionally changes N and hence the ambient world-state space, and is handled as a discrete transition.

3.3 Editing as Constraint Propagation

A user translates object k to $p'_k = p_k + v$, with $v \in \mathbb{R}^3$. The naive edited state is $(p_1, \ldots, p_k + v, \ldots, p_N)$. For a hard target, $T = \{w : p_k = p'_k\}$ fixes the edited object at its requested position. A consistent response lies in $T \cap \mathcal{F}_{\mathcal{A}}$ when this intersection is nonempty. Dependencies identify potentially affected objects, while constraint violations determine the local need for propagation. If the requested position is infeasible, it must be relaxed or the support assignment changed; container clamping is an example of a relaxed target rather than an exact hard-target response.

4 Theory

4.1 Feasible-State Geometry

Theorem 1 (Feasible-State Geometry)**.** *Fix an assignment $\mathcal{A}$ with nonempty feasible set as in Eq.* (3)*. Its component $\mathcal{F}_{\mathcal{A}}$ is a closed convex polyhedron. If the inequalities leave a nonempty relatively open subset of the affine space $w : A_{\mathcal{A}} w = b_{\mathcal{A}}$ feasible, then*

$$\dim \operatorname{aff}(\mathcal{F}_{\mathcal{A}}) = 3N - \operatorname{rank}(A_{\mathcal{A}}). \tag{4}$$

Without this condition, the right-hand side is an upper bound; inequalities can impose additional affine restrictions. Boundary faces can have lower dimension.

The equalities specify vertical support and any grounding constraints; the inequalities specify both horizontal support-surface membership and containment. Their intersection is polyhedral. The full feasible set is the union over admissible assignments and need not be convex. Changing support assignment is a discrete transition; deletion also changes the ambient dimension.

Theorem 1 gives a geometric interpretation of editability. The polyhedral description itself relies on standard convex geometry. Here, layered 3D world structure is represented as a stratified feasible-state geometry, turning support and containment relations into explicit constraints on world-state degrees of freedom. Editability therefore becomes a geometric property of the current dependency stratum.

4.2 Intervention Geometry

Consider a feasible state $w_0 \in \mathcal{F}_{\mathcal{A}}$ and a local intervention that translates supporter j by v, while initially leaving all other objects fixed. Let $w_1 = (p_1, \ldots, p_j + v, \ldots, p_N)$, and write the current stratum as $\mathcal{F}_{\mathcal{A}} = \{w : A_{\mathcal{A}} w = b_{\mathcal{A}},\ G_{\mathcal{A}} w \le h_{\mathcal{A}}\}$, where $A_{\mathcal{A}}$ collects support and grounding equalities, and $G_{\mathcal{A}}$ collects support-surface and containment inequalities.

All vector norms in the distance bound are Euclidean; matrix norms are the corresponding operator norms. Define the constraint residual

$$V(w_1) = \left\| [A_{\mathcal{A}} w_1 - b_{\mathcal{A}};\ [G_{\mathcal{A}} w_1 - h_{\mathcal{A}}]_+] \right\|. \tag{5}$$

Theorem 2 (Intervention Geometry). *Let $(j \to i) \in \mathcal{A}$ be an active support relation and let supporter j be translated by $v \in \mathbb{R}^3$. Keep the assignment $\mathcal{A}$ fixed. For the repair decision in (ii), hold j at its edited position and assume j is the unique active supporter of i, no support reassignment is allowed, and consider repair of this support relation alone. Then:*

(1) if $v_z \neq 0$, keeping i fixed produces vertical support violation exactly $|v_z|$;

(2) under horizontal translation, restoring this support relation requires moving i if and only if its center of gravity leaves the translated support surface of j. The violation is the terminal horizontal overhang (in the ℓ_∞ norm); other scene constraints may independently require i to move;

(3) the constraint residual is equivalent, up to constants determined only by the local constraint geometry, to the distance from the edited state to the feasible stratum:

$$c_1 \operatorname{dist}(w_1, \mathcal{F}_\mathcal{A}) \leq V(w_1) \leq c_2 \operatorname{dist}(w_1, \mathcal{F}_\mathcal{A}), \tag{6}$$

where $c_1 = 1/\kappa$, $c_2 = \|[A_\mathcal{A}; G_\mathcal{A}]\|$, and κ is a Hoffman constant of the constraint system.

Part (3) measures distance to the structural feasible set, without fixing the edited object in the distance minimization. For an isolated support equality with no other constraints, $V(w_1) = |v_z|, \quad \operatorname{dist}(w_1, \mathcal{F}_\mathcal{A}) = \frac{|v_z|}{\sqrt{2}}$, so the relation is tight with $c_1 = c_2 = \sqrt{2}$. Hoffman's lemma is classical; here it links intervention-induced violation to distance from the feasible stratum.

Theorem 2 separates *dependency* from *propagation*. A dependency edge specifies where an intervention may have consequences, but whether the dependent object must move is determined by the intervention direction and the local constraint geometry. Vertical motion necessarily breaks unchanged active supports, whereas the edge-local horizontal criterion selects violated support surfaces. Applying it through a hierarchy requires re-evaluating downstream constraints after each response. It is not a global ℓ_0-minimality guarantee in the presence of coupled containment, multiple supports, or support reassignment.

4.3 Minimal Consistent Response

Theorem 3 (Minimal Consistent Response). *Let $w_0 \in \mathcal{F}_\mathcal{A}$ and let T denote the target set imposed by the user edit. Assume T is closed and convex and $T \cap \mathcal{F}_\mathcal{A} \neq \varnothing$. Within this fixed stratum, the minimum-displacement consistent response is*

$$w^* = \arg \min_{w \in T \cap \mathcal{F}_\mathcal{A}} \frac{1}{2} \|w - w_0\|_2^2. \tag{7}$$

The squared Euclidean objective is strictly convex and coercive, so projection onto the nonempty closed convex intersection exists and is unique. No rank or regular-position assumption is needed for uniqueness. For the affine hard target $T = \{w : p_k = p'_k\}$ (and any affine constraints fixing unselected objects), the problem is a convex quadratic program, provided that the restricted feasible set remains nonempty. Edits that change the active dependency structure, including deletion and support reassignment, are instead treated as discrete transitions between strata.

Theorem 3 minimizes displacement over the chosen feasible response set. Fixing unselected objects adds affine constraints to T and requires a fresh feasibility check. The ℓ_2 objective does not minimize the number of moved objects, so the practical propagation strategy is evaluated empirically rather than claimed to solve a global ℓ_0 problem. The key distinction is that projection computes the correction only after the intervention geometry has identified which dependencies must respond.

4.4 Stability of Projection-Consistent Editing

The projection view provides more than existence and uniqueness: within a fixed feasible stratum, it also yields a stability guarantee for consistent editing.

Proposition 1 (Stability of projection-consistent editing). *Let $\mathcal{F}_A \subset \mathcal{W}$ be a nonempty closed convex stratum and let $P_A(w) = \arg\min_{w' \in \mathcal{F}_A} \|w' - w\|_2$ denote its metric projection. Then:*

(1) $P_A(w)$ exists and is unique for every $w \in \mathcal{W}$, and $\|P_A(w) - w\|_2 = \operatorname{dist}(w, \mathcal{F}_A)$.

(2) P_A is non-expansive: $\|P_A(w) - P_A(w')\|_2 \leq \|w - w'\|_2 \qquad \forall\, w, w' \in \mathcal{W}$.

(3) For a naive intervention $w_1 = w_0 + v, \quad w_0 \in \mathcal{F}_A$, *the projected response satisfies* $\|P_A(w_1) - w_0\|_2 \le \|v\|_2$, *and* $\|P_A(w_1) - w_1\|_2 = \mathrm{dist}(w_1, \mathcal{F}_A)$.

The first two statements are standard properties of metric projection onto a closed convex set. Their consequence here is operational: perturbations in the estimated world state are not amplified by the consistency projection, as long as the active dependency structure is unchanged. Thus, within a fixed stratum, projection-consistent editing is stable to small state-estimation errors.

The guarantee is local to a fixed stratum. Over the full stratified feasible space $\mathcal{F} = \bigcup_A \mathcal{F}_A$, the nearest feasible state may be non-unique because the union is generally non-convex. However, whenever one stratum is uniquely nearest and separated from the alternatives by a positive margin, the selected projection locally coincides with P_A and inherits its 1-Lipschitz stability. Discontinuities can arise only when an intervention crosses a stratum boundary, such as a support break, support reassignment, or deletion. In this sense, editability is governed not only by the constraints themselves, but by the geometry of the current dependency stratum and its transitions. These events are therefore treated separately as discrete structural transitions.

4.5 Consistency Residual

For a world state w evaluated against a fixed assignment $\mathcal{A}$, define

$$R_{\mathcal{F}}(w) = \sum_{(j,i)\in\mathcal{A}} \left[s_{ji}(w) + s^h_{ji}(w)\right] + \sum_{i\in\mathcal{B}} |z_i^{\mathrm{bot}}| + \sum_o \text{containment exceedance}(w). \quad (8)$$

A dependency-consistent state satisfies $R_{\mathcal{F}}(w) \approx 0$, whereas broken support, grounding, or containment relations produce positive residual.

Importantly, $R_{\mathcal{F}}(w)$ is an aggregate constraint residual rather than the exact metric distance $d_{\mathcal{F}}(w) = \inf_{u\in\mathcal{F}} \|w - u\|$. However, within a fixed stratum, Theorem 2 implies $c_1 d_{\mathcal{F}_\mathcal{A}}(w) \le R_{\mathcal{F}}(w) \le c_2 d_{\mathcal{F}_\mathcal{A}}(w)$, with adjusted norm-equivalence constants, provided the residual and feasible set encode exactly the same support, grounding, and containment constraints without tolerance truncation. Thus the residual provides a computable consistency diagnostic while remaining geometrically tied to distance from feasibility.

4.6 Practical Projection

For translational edits within a fixed stratum, the consistent response is obtained by solving the convex quadratic program in Eq. (7) over the support and grounding equalities, support-surface inequalities, and containment bounds. Before projection, the propagation set is restricted using the geometric criterion in Theorem 2: an object inside its translated support surface needs no repair for that edge alone. Coupled constraints and downstream edges must still be checked; a selection yielding an empty restricted feasible set cannot use the projection guarantee.

Container-constrained translations reduce to projection onto the feasible containment box. Deletion changes the dependency structure and is therefore treated as a discrete stratum transition rather than as projection onto a fixed $\mathcal{F}_\mathcal{A}$.

Violation and repair as two views of the same geometry. Intervention-induced violation and consistency restoration can be understood as complementary views of the same feasible-state geometry. The former characterizes how an unconstrained edit departs from structural feasibility, while the latter computes the smallest admissible response that returns the world to the feasible set. This unifies prediction and correction within a single geometric framework: the same constraints that determine when an intervention becomes inconsistent also determine how that inconsistency should be repaired.

5 Experiments

The framework is evaluated primarily on InteriorGS, using real indoor 3DGS scenes with object-level labels and bounding boxes. Baselines include *naive* editing, dependency-following and scene-graph closure, generic alternating projection (POCS), and physics settling.

The primary consistency measure is the edit-induced residual $R_{\mathcal{F}}(w)$ from Eq. (**??**). The main benchmark contains 142 translation, deletion, and container-constrained editing cases, with separate edit sets used for intervention prediction, propagation-set analysis, and minimal-response evaluation.

While InteriorGS provides the primary theory-linked editing benchmark; ARKitScenes evaluates transfer to real RGB-D scenes; a synthetic benchmark stresses deeper and denser support hierarchies; and HunyuanWorld evaluates the same consistency formulation on generated worlds.

5.1 Predicting Intervention Consequences

Theorem 2 predicts the consequence of an intervention before any repair is performed. The prediction benchmark contains 680 edits across nine scenes, with 20 random displacement directions and magnitudes sampled per supporter. For each edit, the local geometric predictor is $\mathrm{pred}(v) = |v_z|\, n_{\mathrm{supp}} + \mathrm{overhang}$, where $\mathrm{overhang}$ denotes the aggregate horizontal exceedance over the support surfaces of the affected dependents, and is compared against displacement magnitude $|v|$, vertical displacement $|v_z|$, and graph fan-out. The prediction target is whether the naive state w_1 requires consistency restoration.

Table 1: Prediction of intervention-induced inconsistency over 680 edits. AUROC is evaluated leave-one-scene-out.

predictor	AUROC	Spearman
geometric predictor	**0.877**	**0.882**
$\lvert v_z\rvert$	0.815	0.576
$\lvert v\rvert$	0.808	0.427
fan-out	0.694	0.611

Local constraint geometry predicts intervention failure substantially better than graph fan-out ($\Delta\mathrm{AUROC} = 0.18$) and displacement magnitude. The relationship also remains strong within fixed fan-out strata, with Spearman correlation ranging from 0.74 to 0.94. A scene-clustered bootstrap gives a 95% confidence interval of $[0.76, 0.95]$ for the leave-one-scene-out AUROC.

These results establish the predictive content of Theorem 2: the geometry does not merely describe an inconsistent state after the edit, but predicts in advance whether a local intervention will leave the feasible set and how strongly consistency restoration will be required.

5.2 Propagation Set and Minimal Response

Under its fixed-assignment, unique-support assumptions, Theorem 2 identifies horizontal support violations through the center-of-gravity condition. Across 408 translational edits on nine scenes, the criterion achieves propagation-set F1 of 1.00, compared with 0.43 for dependency closure.

Because the geometric criterion and the solver use the same center-of-gravity condition, the perfect agreement should be interpreted as an internal-consistency check rather than independent validation. The comparison with dependency closure nevertheless shows that the distinction is nontrivial: closure substantially overestimates the set of objects that must respond to a particular intervention.

An independent 340-edit benchmark evaluates the resulting minimal-response strategy. Restricting propagation to the geometrically necessary set reduces the average number of moved objects from 7.9 to 3.3, a 59% reduction, while retaining zero incremental residual in 334/340 edits (98%). The remaining mean residual is 0.014 m, compared with 0.001 m for full closure.

The reduction is direction-dependent, as predicted by the theory. Under vertical translation, every active support relation is violated if its dependent object remains fixed, so the required propagation set coincides with the upward closure. Under horizontal translation, only supports whose center-of-gravity condition is crossed require response; in the 68-edit benchmark, the projected method moves 4.6 objects on average, compared with 7.9 for closure and 35.5 for cyclic POCS.

Together, these results support the distinction between dependency and propagation: a dependency edge identifies a possible route of influence, while the local intervention geometry determines whether that influence must actually propagate.

5.3 Consistency Restoration and Physical Validation

The practical consistency of the response characterized by Theorem 3 is evaluated on 68 translational edits. Naive editing produces an edit-induced residual above 0.1 in 88% of cases, whereas the projected response achieves zero edit-induced residual in all 68 cases.

Additional comparisons include scene-graph closure, POCS, and physics settling. For an external physical check, every edited output is evaluated by a PyBullet rollout with the user-edited object fixed. Total physical displacement and the number of hanging objects therefore provide signals that are independent of $R_{\mathcal{F}}$.

Table 2: Strong baseline comparison on 68 translational edits across InteriorGS scenes. Physical displacement and hanging count are measured by an independent PyBullet rollout.

method	zero-resid.	moved	phys. disp.	hanging
naive	13%	1.0	9.15	5.16
follow (upward closure)	100%	7.9	0.22	0.06
scene-graph closure	99%	7.9	0.22	0.06
POCS (cyclic)	100%	35.7	2.06	1.37
ours	**100%**	**6.3**	0.79	0.59
physics settling	20%	7.6	—	—

POCS reaches zero residual but moves $5.7\times$ more objects than the proposed response (35.7 vs. 6.3) and fails to converge within the iteration budget in 12/68 cases. The independent physical rollout confirms the main consistency trend: naive edits leave 5.16 supported objects hanging and induce 9.15 m of physical displacement, whereas the projected outputs reduce these values to 0.59 and 0.79 m.

Full closure is physically more conservative, leaving 0.06 hanging objects on average, but moves more objects. The comparison therefore exposes the intended trade-off: the proposed method preserves the target dependency structure with substantially less intervention, whereas unconstrained physics settling is allowed to slide, restack, or otherwise rewrite that structure.

Intervention-margin trade-off. The propagation policy is tunable through a support-residual margin τ. Table 3 summarizes 5,360 trials on 536 ARKitScenes scenes, with two random supporter displacements per scene and per setting. Increasing τ from 0.05 to 0.30 m reduces the average moved-object count from 1.17 to 0.48 (59%), while the mean geometric residual remains at 3.7–6.3 mm. This exposes a controllable intervention–violation balance. The sweep measures geometric residual rather than margin-wise PyBullet stability, so it does not by itself explain away the higher hanging count in Table 2.

Table 3: Intervention–violation trade-off on 5,360 ARKitScenes trials. Larger margins reduce propagation while mean residuals remain at millimeter scale.

margin τ (m)	zero-resid.	avg. moved	avg. residual (m)
0.05	97.5%	1.17	0.0063
0.10	98.2%	0.97	0.0058
0.15	98.8%	0.87	0.0041
0.20	98.5%	0.76	0.0053
0.30	**99.1%**	**0.48**	**0.0037**

The same framework also handles the two additional edit types. For 34 deletion cases, iterative re-support eliminates floating tops in all 34 cases without introducing new collisions. For 40 container-constrained translations, projection by clamping reduces the average containment residual from 0.764 to 0 and achieves zero residual in 40/40 cases.

The consistency residual itself separates intact from synthetically corrupted InteriorGS states with AUROC 0.998. This result is treated as a diagnostic check rather than independent physical validation, since the corruption process directly perturbs the constraints measured by $R_{\mathcal{F}}$.

5.4 Cross-Dataset and Generated-World Transfer

Transfer beyond InteriorGS is evaluated on ARKitScenes, using 536 real RGB-D indoor scenes with support structure, and on a synthetic benchmark of 1,000 deep-support scenes.

For randomly displaced supporters, the repair restores zero residual in 97.3% of ARKitScenes trials and reduces total violation by 94.7%. On the substantially denser synthetic benchmark, zero residual is achieved in 86.3% of trials and the residual decreases by 37.0%.

Horizontal editing again provides the nontrivial regime for minimal intervention. On the synthetic benchmark, the geometric propagation rule moves 24.7% fewer objects than full closure (72.82 vs. 96.75 objects per edit) while producing the same post-repair residual of 0.326. On ARKitScenes, it moves 7.4% fewer objects (2.13 vs. 2.30) while reducing the post-repair residual from 0.502 to 0.465. The consistency diagnostic also transfers across datasets, separating intact from corrupted scenes with AUROC 0.997 on the synthetic benchmark and 0.871 on ARKitScenes.

The formulation is further evaluated on HunyuanWorld-generated scenes. Across 50 generated panoramas, the pipeline detects 1,023 support relations. Before geometric correction, 40.0% of these relations violate the detected support surface; applying the consistency projection reduces the breach rate to 0.0%.

A separate object-level inpainting study contains six supporter edits. None of the six preserves the original support relation, showing that object-level editing alone does not automatically propagate structural dependencies.

Dependency-graph robustness. Graph cleaning reduces geometric edit violation from 0.018 to 0.009 over 28 cases. To test sensitivity to missing relations, we randomly remove support edges on InteriorGS and evaluate the repaired outputs against the original graph. The zero-residual rate falls from 100% to 42% as edge removal increases to 30%, where mean violation reaches 0.53 m. Missing edges prevent the solver from enforcing relations absent from its representation; spurious edges can instead induce unnecessary propagation. These results quantify the dependence on graph quality rather than assuming graph extraction is error-free. Results scored against a perturbed graph are not directly comparable to this original-graph check.

Overall, structural dependencies provide a basis for diagnosing inconsistency and predicting intervention consequences in both reconstructed and generated worlds, with performance depending on the accuracy of those dependencies.

6 Discussion

The fixed-stratum translational setting is the simplest regime in which intervention consistency admits a clean geometric characterization, but the formulation extends beyond it. Changes in support assignment lead to a mixed discrete–continuous problem, while full rigid motion replaces the present polyhedral geometry with orientation-dependent constraints over $\mathrm{SE}(3)$. Oriented boxes, mesh contacts, signed-distance representations, and robust or probabilistic feasible sets provide natural extensions for richer geometry and uncertain dependencies. Across these settings, the central principle remains unchanged: structural relations define admissible world states, interventions perturb them, and consistency restoration is governed by feasible-state geometry.

More broadly, intervention consistency depends not on dependency structure alone, but on how that structure is realized geometrically in the current world state and how an intervention acts upon it. This geometry can serve as a structural-consistency layer for interactive and generative world models. The residual $R_{\mathcal{F}}(w)$ can support verification, learning-time regularization, rewards, rejection, or re-generation, and the partially supervised experiment already indicates benefits for both support consistency and prediction. This does not replace richer physics or generative modeling, nor does membership in $\mathcal{F}$ imply complete physical, semantic, dynamic, or perceptual plausibility. Rather, it isolates a complementary notion of consistency that is explicit, predictable under intervention, and independently verifiable, pointing toward world models that maintain coherent structured worlds throughout interaction and editing.

AI USE STATEMENT

Generative AI tools were used to assist with language editing, clarity, and structural refinement of the manuscript. All mathematical arguments, theoretical claims, experimental design, numerical results, and cited sources were reviewed and verified by the authors. The authors take full responsibility for the content of the paper.

REPRODUCIBILITY STATEMENT

The paper is designed to make both the theoretical analysis and the empirical evaluation reproducible. All geometric operators, consistency conditions, intervention definitions, and evaluation criteria are stated explicitly in the main text and appendices. The construction of layered scenes, the admissible intervention families, and the procedures used to measure geometric and cross-layer consistency are specified sufficiently to reconstruct the experiments independently. For each reported experiment, the scene configuration, intervention protocol, baselines, and evaluation metrics are documented, together with the relevant implementation and optimization details. Additional derivations, proofs, ablations, and qualitative examples are provided in the supplementary material. The experiments use deterministic scene transformations wherever applicable, and random components can be reproduced by fixing the reported experimental seeds.

REFERENCES

Eashan Adhikarla and Brian D. Davison. Remedit: Efficient diffusion editing with riemannian geometry. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 5037–5046, 2026.

Patrick L. Combettes. Convex set theoretic image recovery by extrapolated iterations of parallel subgradient projections. *IEEE Transactions on Image Processing*, 6(4):493–506, 1997.

Richard L. Dykstra. An algorithm for restricted least squares regression. *Journal of the American Statistical Association*, 78(384):837–842, 1983.

Elisabetta Fedele, Francis Engelmann, Ian Huang, Or Litany, Marc Pollefeys, and Leonidas J. Guibas. Spacecontrol: Introducing test-time spatial control to 3d generative modeling. In *International Conference on Learning Representations*, 2026.

Zhening Huang, Xiaoyang Wu, Fangcheng Zhong, Hengshuang Zhao, Matthias Nießner, and Joan Lasenby. LiteReality: Graphics-ready 3d scene reconstruction from rgb-d scans. In *Advances in Neural Information Processing Systems*, 2025.

HunyuanWorld Team. Hunyuanworld 1.0: Generating immersive, explorable, and interactive 3d worlds from words or pixels, 2025.

Haiyan Jiang, Deyu Zhang, Dongdong Weng, Weitao Song, and Henry Been-Lirn Duh. HOG-Layout: Hierarchical 3d scene generation, optimization and editing via vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 31586–31596, 2026.

Rafal Karczewski, Markus Heinonen, Alison Pouplin, Soren Hauberg, and Vikas Garg. The space-time of diffusion models: An information geometry perspective. In *International Conference on Learning Representations*, 2026.

Minzhang Li, Kuixiang Shao, Xuebing Li, Yuyang Jiao, Yinuo Bai, Hengan Zhou, Sixian Shen, Jiayuan Gu, and Jingyi Yu. SPREAD: Spatial-physical reasoning via geometry aware diffusion. *arXiv preprint arXiv:2603.27573*, 2026a.

Xiang Li, Zebang Shen, Ya-Ping Hsieh, and Niao He. When scores learn geometry: Rate separations under the manifold hypothesis. In *International Conference on Learning Representations*, 2026b.

Lu Ling, Chen-Hsuan Lin, Tsung-Yi Lin, Yifan Ding, Yu Zeng, Yichen Sheng, Yunhao Ge, Ming-Yu Liu, Aniket Bera, and Zhaoshuo Li. Scenethesis: A language and vision agentic framework for 3d scene generation. In *International Conference on Learning Representations*, 2026.

Yuhui Lu, Wenjing Liu, and Kun Zhan. Information-geometric adaptive sampling for graph diffusion. *arXiv preprint arXiv:2605.00250*, 2026.

Jean-Jacques Moreau. Proximité et dualité dans un espace hilbertien. *Bulletin de la Société Mathématique de France*, 93:273–299, 1965. doi: 10.24033/bsmf.1625.

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*, pp. 20507–20518, 2024. doi: 10.1109/CVPR52733.2024.01938.

Haitao Yang, Bo Sun, Liyan Chen, Amy Pavel, and Qixing Huang. Geolatent: A geometric approach to latent space design for deformable shape generators. *ACM Transactions on Graphics*, 42(6): 242:1–242:20, 2023. doi: 10.1145/3618371.

A Additional Theory

A.1 Proof of Theorem 1: Feasible-State Geometry

For a fixed support assignment $\mathcal{A}$, Eq. (3) includes both E_{ji} and S_{ji} for every selected support edge, as well as containment and any grounding constraints. With fixed shapes and orientations, these are affine equalities and linear inequalities, so their intersection is a closed convex polyhedron.

The equality space $H = \{w : A_{\mathcal{A}} w = b_{\mathcal{A}}\}$ has dimension $3N - \mathrm{rank}(A_{\mathcal{A}})$ whenever it is nonempty. If the feasible polyhedron contains a nonempty relatively open subset of H, its affine hull is H, which proves the stated dimension formula. Otherwise, inequalities can reduce its affine hull further, and the formula gives only an upper bound. An individual boundary face can also have smaller dimension than the whole component.

The union over admissible support assignments need not be convex. Adding or removing an independent equality changes the equality-space dimension by one; the feasible dimension has the same change only if the relative-interior condition continues to hold. Deletion also changes the number of position variables. Conditioning of distance bounds depends on the full constraint matrices; a boundary contact alone does not imply that the fixed linear constraint gradients become degenerate.

A.2 Empirical Grounding of the Dependency Structure

Across the 10 InteriorGS scenes, the raw dependency extraction produces 71,039 candidate edges. After geometric filtering and graph cleaning, these reduce to 275 support edges and 4,904 containment edges.

Support relations are sparse—approximately 0.49% of all object pairs—but are geometrically reliable after cleaning. Approximately 91% of the retained support edges are geometrically valid. The principal false-positive modes in the raw graph are wall-hung objects, carpet-mediated support chains, and dense same-class stacks.

Containment relations are substantially denser and noisier. Approximately 63% of the candidate containment edges remain usable after removing dense-overlap false positives.

A.3 Proof of Theorem 3: Minimal Consistent Response

Let $C = T \cap \mathcal{F}_{\mathcal{A}}$ be nonempty, closed, and convex. The function $\frac{1}{2}\|w - w_0\|_2^2$ is continuous and coercive, so it attains its minimum on C. Strict convexity implies uniqueness: two distinct minimizers would have a feasible midpoint with strictly smaller objective. This proof does not require full row rank of $A_{\mathcal{A}}$. For affine target and object-fixing constraints, C is polyhedral and the problem is a convex quadratic program.

The edge-local propagation criterion in Theorem 2(ii) and this projection solve different questions. Selecting objects and fixing the rest adds affine constraints, whose joint feasibility must be checked. Projection minimizes squared displacement on the resulting feasible set; it does not establish a globally smallest set of moving objects.

A.4 Stability of Projection-Consistent Editing

Proposition 2 (Stability of projection-consistent editing). *Let $P_{\mathcal{A}} : W \to \mathcal{F}_{\mathcal{A}}$ be the metric projection onto a nonempty fixed stratum,*

$$P_{\mathcal{A}}(w) = \arg\min_{w' \in \mathcal{F}_{\mathcal{A}}} \|w' - w\|.$$

Then:

1. *$P_{\mathcal{A}}(w)$ exists and is unique for every w, and*
$$\|P_{\mathcal{A}}(w) - w\| = \operatorname{dist}(w, \mathcal{F}_{\mathcal{A}}).$$
2. *$P_{\mathcal{A}}$ is non-expansive:*
$$\|P_{\mathcal{A}}(w) - P_{\mathcal{A}}(w')\| \le \|w - w'\|,$$
for all w, w' (Moreau, 1965). Consequently, an estimation error δ in the input world state perturbs the projection by at most $\|\delta\|$, provided the constraint set is unchanged. Perturbing the estimated graph or boxes changes that set and is not covered by this bound.
3. *For a naive edit $w_1 = (p_1, \dots, p_j + v, \dots, p_N)$ with $w_0 \in \mathcal{F}_{\mathcal{A}}$,*
$$\|P_{\mathcal{A}}(w_1) - w_0\| \le \|v\|,$$
while
$$\|P_{\mathcal{A}}(w_1) - w_1\| = \operatorname{dist}(w_1, \mathcal{F}_{\mathcal{A}}).$$
4. *Over a finite union of fixed-inventory components*
$$\mathcal{F} = \bigcup_{\mathcal{A}} \mathcal{F}_{\mathcal{A}},$$
the nearest point may be non-unique because the union is non-convex. However, whenever one stratum is uniquely nearest and separated from all alternatives by a positive margin, the selected projection coincides locally with $P_{\mathcal{A}}$ and is therefore 1-Lipschitz. Changes of the nearest component can cause discontinuities at these decision boundaries.

Justification. The first statement follows from the classical projection theorem for closed convex sets. For the second, Moreau's non-expansiveness result gives

$$\langle P_{\mathcal{A}}(w) - P_{\mathcal{A}}(w'), w - w' \rangle \ge \|P_{\mathcal{A}}(w) - P_{\mathcal{A}}(w')\|^2,$$

and Cauchy–Schwarz yields the Lipschitz inequality.

For the third statement, apply non-expansiveness with $w' = w_0$, noting that $P_{\mathcal{A}}(w_0) = w_0$ because $w_0 \in \mathcal{F}_{\mathcal{A}}$.

For the final statement, each individual $\mathcal{F}_{\mathcal{A}}$ is closed and convex and therefore admits a unique projection. Non-uniqueness can occur only where two or more strata are equidistant from the input. Away from such decision boundaries, projection reduces to the single-stratum operator and inherits its stability.

A.5 Proof of Theorem 2: Intervention Geometry

Let $w_0 \in \mathcal{F}_{\mathcal{A}}$ and consider a support edge $(j \to i)$. A naive edit moves supporter j by

$$v = (v_x, v_y, v_z)$$

while keeping the remaining objects fixed.

Vertical displacement. At the initial feasible state,

$$z_i^{\mathrm{bot}}(0) = z_j^{\mathrm{top}}(0).$$

Under the edit,

$$z_j^{\mathrm{top}}(t) = z_j^{\mathrm{top}}(0) + t v_z, \qquad z_i^{\mathrm{bot}}(t) = z_i^{\mathrm{bot}}(0).$$

Hence the support residual is

$$s_{ji}(t) = |t v_z|,$$

and therefore

$$s_{ji}(1) = |v_z|.$$

Horizontal displacement. Let

$$v_h = (v_x, v_y)$$

and let $r(t)$ denote the center of gravity of the supported object relative to the support surface. Then

$$r(t) = r(0) - tv_h.$$

Initially,

$$|r(0)_x| \le \frac{w_j}{2}, \qquad |r(0)_y| \le \frac{d_j}{2}.$$

Write the rectangular support footprint as $Q_j = [-w_j/2, w_j/2] \times [-d_j/2, d_j/2]$, where w_j, d_j are its width and depth. When $r(1) \notin Q_j$, define the exit time

$$t_{\text{exit}} = \sup\{t \in [0,1] : r(t) \in Q_j\}.$$

Since r is affine and Q_j is closed and convex, the feasible times form a closed interval $[0, t_{\text{exit}}]$, with $t_{\text{exit}} < 1$ and $r(t_{\text{exit}}) \in \partial Q_j$. This also covers a boundary start: an outward motion has exit time zero, whereas an inward motion can exit later. The terminal horizontal violation is therefore

$$s^h_{ji}(1) = \max\left(0, |r(1)_x| - \frac{w_j}{2}, |r(1)_y| - \frac{d_j}{2}\right).$$

If the edited supporter is fixed and remains the unique active supporter, a positive terminal overhang cannot be removed while i stays fixed. Conversely, if the overhang is zero, that support-surface constraint needs no repair; convexity of the rectangular surface also makes the straight relative path feasible when both endpoints are inside. Moving i to the nearest point of the surface repairs this edge when necessary, but may conflict with other constraints. Thus this is an edge-local equivalence, not a proof of global movement sparsity or feasibility under additional constraints.

Relation to distance from the feasible set. Write

$$\mathcal{F}_{\mathcal{A}} = \{w : A_{\mathcal{A}} w = b_{\mathcal{A}},\ G_{\mathcal{A}} w \le h_{\mathcal{A}}\},$$

and define

$$V(w) = \left\| [A_{\mathcal{A}} w - b_{\mathcal{A}};\ [G_{\mathcal{A}} w - h_{\mathcal{A}}]_+] \right\|.$$

The residual map is Lipschitz with constant

$$L = \|[A_{\mathcal{A}}; G_{\mathcal{A}}]\|$$

and vanishes on $\mathcal{F}_{\mathcal{A}}$. Let w^* be the metric projection of w_1 onto $\mathcal{F}_{\mathcal{A}}$. Then

$$V(w_1) \le L\|w_1 - w^*\| = c_2 \operatorname{dist}(w_1, \mathcal{F}_{\mathcal{A}}).$$

For completeness, the reverse inequality can be proved directly in finite dimensions. Let $q = P_{\mathcal{A}}(w)$ and $d = w - q$. If $d = 0$, the claim is immediate. Otherwise, the projection optimality condition places d in the normal cone at q. For a polyhedron, this cone is generated by the signed rows of $A_{\mathcal{A}}$ and the rows of $G_{\mathcal{A}}$ active at q. Hence

$$d = B^\top \alpha, \qquad \alpha \ge 0,$$

for a matrix B of linearly independent selected normal rows. To obtain independence, start with any conic representation and remove a generator whenever the selected normals are linearly dependent: move the coefficients along a dependence until one becomes zero, keeping all coefficients nonnegative. Repeating terminates with independent rows.

Every selected equality normal has scalar product with d equal to a signed equality residual at w. Every selected active inequality normal has scalar product with d equal to its inequality residual at w, which is bounded above by its positive part. Independence prevents selecting both signs of the same equality row. Cauchy–Schwarz therefore gives

$$\|d\|_2^2 = \alpha^\top B d \le \|\alpha\|_2 V(w) \le \|(B^\top)^\dagger\|_2 \, \|d\|_2 V(w),$$

where $\dagger$ is the pseudoinverse; $B^\top$ has full column rank. There are only finitely many independent row selections from the signed equality and inequality normals. Taking κ to be the maximum

of $\|(B^\top)^\dagger\|_2$ over these nonempty selections gives a finite positive constant depending only on $A_\mathcal{A}, G_\mathcal{A}$. Dividing by $\|d\|_2$ proves

$$\operatorname{dist}(w, \mathcal{F}_\mathcal{A}) \le \kappa V(w).$$

This is the required Hoffman bound, with $c_1 = 1/\kappa$. The support edge in the theorem ensures at least one nonzero constraint normal; for an unconstrained world both distance and residual are identically zero.

For an isolated support equality with no other constraints

$$E_{ji} : z_i - z_j = 0,$$

the constraint normal is

$$(0, 0, 1, 0, 0, -1).$$

Moving only j vertically gives

$$V(w_1) = |v_z|.$$

The Euclidean distance to the support hyperplane is

$$\operatorname{dist}(w_1, \mathcal{F}_\mathcal{A}) = \frac{|v_z|}{\sqrt{2}},$$

and therefore

$$V(w_1) = \sqrt{2}\operatorname{dist}(w_1, \mathcal{F}_\mathcal{A}).$$

Thus the bound is tight with

$$c_1 = c_2 = \sqrt{2}.$$

A.6 Corollary: Naive, Follow, and Fan-Out Editing

A follow edit keeps

$$z_i^{\text{bot}}$$

tracking

$$z_j^{\text{top}},$$

so the corresponding support residual remains zero. Consequently, the difference between naive and follow editing is exactly $|v_z|$ for a vertical support violation and the overhang distance for a horizontal violation.

For a supporter with m dependent tops, the naive fan-out violation is

$$\sum_k \left(|v_z| + \text{overhang}_k\right).$$

Similarly, moving an object outside its container incurs violation equal to its geometric exceedance beyond the containment box.

A.7 Residual Score and Metric Distance

The quantity

$$R_\mathcal{F}(w)$$

used in the main text is a sum of per-constraint residuals rather than the exact metric distance

$$d_\mathcal{F}(w) = \inf_{u \in \mathcal{F}} \|w - u\|.$$

Within a fixed stratum, when both quantities encode the same complete constraint system and no tolerance truncation is applied, Theorem 2 and finite-dimensional norm equivalence give, for adjusted positive constants,

$$c_1 d_{\mathcal{F}_\mathcal{A}}(w) \le R_\mathcal{F}(w) \le c_2 d_{\mathcal{F}_\mathcal{A}}(w).$$

The four terms used in the residual—support breakage, horizontal overhang, ground breakage, and containment exceedance—are geometric residuals measured in metres (horizontal overhang uses the ℓ_∞ norm). They are therefore summed without additional weighting or normalization.

Each term is continuous and piecewise linear and vanishes exactly when the corresponding constraint is satisfied. The residual is thus zero precisely when all encoded support, grounding, and containment constraints hold.

B Dependency Graph Construction and Projection Details

B.1 Dependency-Graph Cleaning

Raw bounding-box-derived dependency graphs contain systematic false positives. We apply three cleaning rules.

1. **False-support filtering.** Thin or wall-hung objects such as carpets, mirrors, and wall racks are removed as supporters. Wall-hung objects instead inherit support from floor, wall, or ceiling when appropriate.
2. **Multi-support deduplication.** When a top object is assigned multiple candidate supporters, the highest geometrically valid support surface is retained.
3. **Dense-overlap removal.** Same-label containment pairs with near-identical bounding boxes, particularly dense stacks such as cups, are removed from the containment graph.

The cleaning reduces the support graph from 329 to 275 edges and the containment graph from 5,568 to 4,904 edges. All rules were checked numerically and visually.

B.2 $\mathcal{F}$-Projection Solvers

Translation. For pure translation, the active support set is fixed and the optimization remains inside a single stratum $\mathcal{F}_{\mathcal{A}}$. For an affine target T, the target-set projection is

$$w^* = \arg\min_{w \in T} \frac{1}{2}\|w - w_0\|_2^2 \quad \text{subject to} \quad A_{\mathcal{A}} w = b_{\mathcal{A}}, \quad G_{\mathcal{A}} w \leq h_{\mathcal{A}}.$$

The inequality matrix includes support-surface and containment conditions, not merely independent per-object bounds. Provided the intersection is nonempty, this is a convex quadratic program, which can be solved by a QP solver or Dykstra-style projection onto its constituent convex constraints.

Deletion. Deleting a supporter changes the dependency graph and therefore cannot be represented as projection onto the same fixed stratum. We use iterative settling: unsupported tops are processed by height and placed on the highest live support surface below them with overlapping XY projection, or on the ground when no such supporter exists. The procedure is repeated until no support assignment changes.

Container-constrained translation. When a contained object is translated to a target outside its container, its position is clamped to the nearest point inside the containment box. This is the minimum-displacement correction satisfying the containment constraint.

Minimal-intervention propagation. Propagation and displacement are treated separately. Theorem 2(ii) identifies violated horizontal support edges under its fixed-assignment assumptions. The practical rule re-evaluates support conditions as responses propagate through the hierarchy. If unselected objects are fixed, the resulting QP requires a nonempty feasible set; the local criterion alone does not guarantee feasibility with coupled constraints or global ℓ_0 minimality.

B.3 Grounding Convention

InteriorGS contains no explicit floor object. We therefore treat the plane $z = 0$ as the implicit ground. Objects with no supporter and with their lower bounding-box face close to $z = 0$ are marked as grounded. Furniture spuriously assigned to carpets is lifted to the ground support relation after graph cleaning.

B.4 Center-of-Gravity Approximation

The support-surface condition and Theorem 2(ii) require the horizontal center of gravity of each object. We approximate it using the center of the object's axis-aligned bounding box in the XY plane. This convention requires no mesh-level metadata and is applied uniformly to every scene.

For elongated, concave, or rotated objects, the AABB center may differ from the true geometric centroid. Uniform XY perturbations of the estimated center quantify the sensitivity of the propagation decision to this approximation.

C Additional Editing Experiments

C.1 Translation Edits

Across the 10 InteriorGS scenes, supporter objects are translated vertically by $\Delta z = +0.3$ and horizontally by $\Delta x = +0.5$.

Table 4: Translation edits (68 cases): edit-induced violation.

	naive	follow	projected
Δ-violation > 0.1	88%	0%	**0%**
Δ-naive $\geq$ follow $\geq$ projected	—	—	**68/68 (100%)**
Δ-projected $= 0$	—	—	**68/68 (100%)**

C.2 Deletion Edits

Deleting a supporter requires the affected objects to acquire new support rather than remaining suspended.

Table 5: Deletion edits (34 cases): floating tops and collisions.

	naive	follow	projected
floating tops (avg.)	5.0	0 (drops content)	**0.0**
zero-floating cases	0/34	34/34 (drops content)	**34/34 (100%)**
new collisions	0	—	**0.0**

C.3 Container-Constrained Translation

When a target position lies outside the object's container, the projected edit clamps the object to the feasible boundary.

Table 6: Container-constrained translation (40 cases).

	follow	projected (clamp)
avg. containment violation	0.764	**0.000**
zero-violation cases	—	**40/40 (100%)**

D Strong-Baseline Details

The POCS baseline uses cyclic alternating projection onto the active support constraints with an iteration budget of 400 and residual tolerance 10^{-10}. A run is counted as non-converged when it does not satisfy all active constraints within this budget.

The high moved-object count of cyclic POCS is a consequence of repeatedly updating the entire connected component. A block-coordinate schedule would reduce this overhead, but would still lack the geometric propagation criterion used by our method.

Across the 68 translational edits, POCS fails to converge in 12 cases (18%), requiring 161 iterations on average in the remaining runs. Failures concentrate in scenes with deep multi-level support chains, where alternating projections cycle among competing branches.

The independent PyBullet evaluation should be interpreted as a complementary signal rather than as the definition of consistency. A rigid-body engine is allowed to rewrite the scene's support graph by sliding or re-stacking objects, whereas $\mathcal{F}$ encodes the intended dependency structure of the edited scene.

E Intervention–Violation Trade-off

The intervention margin τ controls how aggressively support relations are enforced. Support edges whose residual is below τ are accepted without propagation.

We sweep

$$\tau \in \{0.05, 0.10, 0.15, 0.20, 0.30\}\,\mathrm{m}$$

on 536 ARKitScenes scenes, using two random supporter displacements per scene and per setting, for 5,360 trials in total.

Table 7: Pareto sweep of the intervention margin τ on ARKitScenes.

τ (m)	zero-viol. rate	avg. moved	avg. violation (m)
0.05	97.5%	1.17	0.0063
0.10	98.2%	0.97	0.0058
0.15	98.8%	0.87	0.0041
0.20	98.5%	0.76	0.0053
0.30	**99.1%**	**0.48**	**0.0037**

As τ increases from 0.05 to 0.30 m, the average number of moved objects decreases from 1.17 to 0.48, a 59% reduction, while the residual violation ranges from 3.7 to 6.3 mm across the tested settings. These are aggregate geometric results, not a strict-feasibility guarantee for every trial. No margin-wise PyBullet stability evaluation is available.

F Additional Corruption-Detection Results

For each of nine InteriorGS scenes, we construct corrupted variants by randomly displacing supporters.

Table 8: Corruption detection using $R_{\mathcal{F}}(w)$.

	intact	corrupted
$R_{\mathcal{F}}(w)$ mean	0.080	3.416
AUROC	—	**0.998**

Every scene's six corrupted states have larger residual than the corresponding intact state. A scene-clustered bootstrap with 2,000 replications gives AUROC 0.998 with 95% confidence interval

$$[0.992, 1.000].$$

This evaluation measures separability under the synthetic corruption family used by the editing benchmark. It should therefore be interpreted as a consistency diagnostic with respect to the scene's encoded support and containment relations, rather than as a general perceptual realism score.

G Additional Prediction and Propagation Analysis

G.1 Per-Scene Stability of the Geometric Predictor

The leave-one-scene-out AUROC of the geometric predictor from Theorem 2 averages 0.877. The per-scene range is $[0.45, 1.00]$; the weakest scene, 0007, contains only 20 evaluated edits. A scene-clustered bootstrap yields a 95% confidence interval of

$$[0.76, 0.95].$$

Within fixed graph-fan-out strata, the geometric predictor retains strong association with the realized violation, with Spearman correlation ranging from 0.74 to 0.94.

G.2 Interpretation of the Propagation-Set Result

Table 9: Propagation-set agreement over 408 translational edits.

	propagation-set F1	Jaccard
geometric criterion (CoG outside support)	**1.00**	**1.00**
dependency closure	0.43	0.39

The geometric predictor and the projection solver both use the same center-of-gravity criterion. Therefore the observed F1 of 1.00 should be interpreted as an internal consistency check rather than as fully independent validation.

Two additional observations show that the criterion is nevertheless nontrivial.

First, full dependency closure is substantially larger than the propagated set: the closure contains approximately $2.76\times$ as many objects and achieves only Jaccard 0.38 against the minimal propagation set.

Second, the independent PyBullet rollout does not use the geometric criterion, yet the restricted projected edits leave only 0.59 hanging objects on average, compared with 5.16 after naive editing.

On the independent 340-edit benchmark, the minimal-intervention projector reduces the average moved-object count from 7.9 to 3.3 while producing zero incremental violation in 334/340 cases. The remaining average residual is 0.014 m, compared with 0.001 m under full closure.

H Cross-Dataset Evaluation Details

ARKitScenes contributes 536 real RGB-D indoor scenes with support structure. These scenes contain 2.8 support edges on average and have maximum support-chain depth 8.

The synthetic benchmark contains 1,000 procedurally generated deep-support scenes with approximately 1,199 support edges per scene. Chain depths range from 3 to 17, with mean 8.1.

H.1 Vertical Damage Benchmark

For each scene, one supporter is displaced vertically using

$$\Delta z \in [-0.5, 0.5].$$

On ARKitScenes, across three trials per scene, repair restores zero violation in 97.3% of cases and reduces violation by 94.7%.

On the synthetic suite, zero violation is achieved in 86.3% of cases and average violation falls by 37.0%.

In this benchmark, vertical supporter displacement breaks every direct active support relation, so the minimal propagation set collapses to the full closure. This setting therefore tests repair reliability rather than the sparsity advantage of Theorem 2(ii).

H.2 Horizontal Edit Benchmark

To test the nontrivial propagation regime, one randomly selected supporter is translated horizontally by

$$\Delta x \in [0.5, 1.5]\,\mathrm{m}.$$

On the synthetic suite, minimal propagation moves 72.82 objects per edit compared with 96.75 for full closure, a reduction of 24.7%, while both methods obtain the same post-repair residual of 0.326.

On ARKitScenes, minimal propagation moves 2.13 objects compared with 2.30 for closure, a reduction of 7.4%. The post-repair residual decreases from 0.502 to 0.465, and the zero-violation rate increases from 37.9% to 41.0%.

The diagnostic also transfers across datasets, reaching AUROC 0.997 on the synthetic suite and 0.871 on ARKitScenes.

I Generated-Scene Validation Details

I.1 Support Structure on Generated Scenes

We apply the pipeline to HunyuanWorld-1.0-lite scenes generated at 2560×1280 resolution and use Grounding-DINO for object detection.

In a representative living-room scene, 27 objects are detected and the geometric procedure identifies 23 support pairs. For objects on the coffee table, cup–table gaps are 38–43 pixels, corresponding to approximately 4.0–4.5% of image height, and book–table gaps are 24–36 pixels, or approximately 2.5–3.8%.

These measurements indicate that the support relation required by the framework is recoverable from structured generated scenes.

I.2 Editing Does Not Propagate Without Explicit Structure

We move the coffee table using object-level inpainting and re-detect the supported objects. Across six supporter edits—two cups, two books, and two cushions—the table moves 710–1225 pixels while dependent objects move only 0–206 pixels.

Their relative displacement with respect to the table changes by 584–1033 pixels, and none of the six edits preserves the original support relation. Thus naive object-level inpainting preserves non-target content but does not automatically propagate scene dependencies.

I.3 Layered 3D Reconstruction

The generated panorama is decomposed into layered depth estimates and reconstructed as layered meshes. A representative scene contains:

- layer 0: a foreground mesh with approximately 464K vertices;
- layer 1: a smaller object layer containing the book, with approximately 6K vertices;
- layer 2: a background mesh of approximately 3.3M vertices representing floor, walls, and ceiling.

The decomposition yields geometrically separated object components on structured indoor scenes.

I.4 Scene Dependence of Object-Layer Identification

Layer decomposition is substantially less reliable on texture-heavy scenes. Ancient walls with strong mottling or erosion may be segmented into multiple small attachment-like regions, which are subsequently reconstructed as floating fragments.

The framework therefore assumes sufficiently stable object-layer extraction. This assumption holds well for structured interiors but can fail on highly textured surfaces.

I.5 Hard-Constraint Post-Processing

We further evaluate $\mathcal{F}$-projection on 1,023 support edges extracted from 50 HunyuanWorld-generated panoramas. The dataset contains 20.5 support edges per scene on average, with median 16.5 and range 1–72.

For each edge, we measure support-surface overhang

$$\mathrm{hov}_{px},$$

defined as the number of pixels by which the dependent object's center exceeds the supporter's top projection. We count an edge as violated when

$$\mathrm{hov}_{px} > 5.$$

Before post-processing, 409/1,023 edges (40.0%) violate the support surface, with mean overhang 63.78 pixels. After projection, the breach rate falls to 0.0% and mean overhang to 0.

Because the correction is a closed-form 2D shift of each violating dependent object, the post-processing step is linear in the number of detected support relations.

J Robustness

J.1 Dependency-Graph Noise

Table 10: Sensitivity to missing support edges on InteriorGS. Projected edits are evaluated against the original dependency graph.

removed support edges	zero-residual rate
0%	100%
5%	83%
10%	66%
20%	58%
30%	42%

Missing or spurious graph edges can alter the propagation set. Cleaning alone reduces the violation of geometric edits from 0.018 to 0.009 over 28 cases.

We further delete a random fraction p of support edges and evaluate the projected edit against the original graph. On InteriorGS, the zero-violation rate decreases from 100% at $p = 0$ to

$$83\%,\ 66\%,\ 58\%,\ 42\%$$

for

$$p = 5\%, 10\%, 20\%, 30\%,$$

respectively, while mean violation increases to 0.53 at the largest perturbation.

Missing edges are the more damaging failure mode because the solver cannot propagate a support relation it does not observe. Spurious edges instead tend to cause unnecessary propagation and are partly removed by graph cleaning.

Applying the same edge-deletion sweep to ARKitScenes and the synthetic benchmark yields zero-violation rates of approximately 97.7–98.8% on ARKitScenes and 100% on the synthetic suite when evaluated against the perturbed graph.

J.2 Center-of-Gravity Robustness

We perturb the AABB-center estimate uniformly in the XY plane and measure how often the move/don't-move propagation decision changes.

For small perturbations of 1–5 cm, the propagation decision changes only rarely. At ± 0.05 m, the flip rate is 1.15% on the synthetic benchmark and 2.35% on ARKitScenes. At ± 0.10 m, the corresponding rates are 15.71% and 4.58%.

The denser synthetic hierarchy is more sensitive because many small objects lie close to support boundaries. For such cases, mesh centroids or oriented bounding boxes provide a natural extension beyond the AABB-center approximation.

Table 11: Sensitivity of the propagation decision to perturbations of the AABB-center proxy.

perturbation d (m)	synthetic		ARKitScenes	
	flip rate	n edges	flip rate	n edges
± 0.01	0.00%	492,678	0.43%	4,431
± 0.03	0.15%	492,678	1.56%	4,431
± 0.05	1.15%	492,678	2.35%	4,431
± 0.10	15.71%	492,678	4.58%	4,431
± 0.15	53.21%	492,678	7.20%	4,431
± 0.20	72.41%	492,678	10.72%	4,431

K $R_{\mathcal{F}}(w)$ AS A TRAINING REGULARIZER

The residual $R_{\mathcal{F}}(w)$ is piecewise linear in object positions and therefore differentiable almost everywhere and subdifferentiable everywhere.

To test whether it can act as a training signal, we construct a partially supervised repair task from InteriorGS. Each support edge $(b \to t)$ becomes a sample by vertically displacing supporter b using

$$\Delta z \in \{\pm 0.2, \ldots, \pm 0.5\}.$$

The dependent object is left floating.

A small MLP maps

$$[\mathrm{bbox}(b), \mathrm{bbox}(t)] \in \mathbb{R}^{12}$$

to the repaired vertical position of the top object. Only 50% of the repair targets are observed, and the observed labels are corrupted with Gaussian noise

$$\sigma = 0.1.$$

We optimize

$$L_\lambda = \frac{1}{|\mathcal{L}|} \sum_{i \in \mathcal{L}} (\hat{z}_i - z_i)^2 + \lambda \frac{1}{n} \sum_{i=1}^{n} \mathrm{relu}\left(|\hat{z}_i - z_{b_i}^{\mathrm{top}}| - \mathrm{tol}\right), \tag{9}$$

where $\mathcal{L}$ denotes the labeled subset and the second term is the differentiable support-breakage component of the consistency residual.

Table 12: Partial-supervision repair with differentiable consistency regularization. Results are leave-one-scene-out, mean±std over four seeds.

λ	support breakage	MSE (true target)
0	0.115 ± 0.035	0.052 ± 0.024
1	0.100 ± 0.023	0.043 ± 0.018
2	$\mathbf{0.092 \pm 0.022}$	$\mathbf{0.039 \pm 0.014}$
5	0.096 ± 0.028	0.041 ± 0.025

At $\lambda = 2$, test support breakage decreases from 0.115 to 0.092, a reduction of 20%, while MSE against the noise-free repair target decreases from 0.052 to 0.039, a reduction of approximately 25% calculated from the reported means.

All tested nonzero regularization strengths outperform the unregularized baseline on both metrics. The result suggests that the same geometric residual used for verification can also provide a useful consistency prior under noisy or partial supervision.