DOVETAIL3D: UNIFYING LAYOUT AND STYLING FOR VLM-BASED 3D INDOOR SCENE SYNTHESIS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language models (VLMs) play an increasingly crucial role in 3D indoor scene synthesis, yet they rarely acquire the ability to construct scenes directly. Current systems broadly follow two design paradigms: tool-based agentic pipelines utilize explicit optimization tools but can impose rigid or mutually incompatible constraints, whereas model-driven generation preserves flexibility but lacks reliable physical control. Beyond the spatial optimization problems, style coherence is largely absent across both paradigms, despite its importance to overall scene aesthetics. We close this gap by presenting Dovetail3D, which post-trains one VLM model to jointly retrieve, select and place 3D assets, optimizing functional layout and stylistic coherence as a whole. Driven by the spatial properties of indoor scenes, we formulate scene synthesis as a sequential decision-making process. We first design a compact set of general-purpose, physically grounded rules to prune the agent's vast coordinate action space into feasible regions, while leaving the final placement decision to the agent. To train the model, we introduce a hierarchical RL post-training algorithm that integrates coarse-to-fine credit assignment for joint layout and styling. Experiments on 3D-FRONT scenes across diverse room types demonstrate that our post-training paradigm successfully internalizes layout and styling as core model capabilities, combining the reliability of tool-based pipelines with the flexibility of model-driven generation.

1 INTRODUCTION

3D indoor scene synthesis supports applications in interior design (Merrell et al., 2011; Zhang et al., 2019), interactive scene authoring and editing (Paschalidou et al., 2021; Tang et al., 2024), and language-guided virtual environment construction (Feng et al., 2023; Yang et al., 2024b). The rise of embodied AI has broadened this role: synthesized scenes provide scalable virtual environments for training and evaluating embodied agents (Deitke et al., 2022; Yang et al., 2024a; 2025a). Across these applications, the central challenge is to compose 3D assets into scenes that are visually coherent, physically and functionally viable, and aligned with task intent.

Current language-guided scene synthesis systems broadly follow two design paradigms. Tool-based pipelines improve physical controllability through spatial optimization (Sun et al., 2025a), tool orchestration (Yang et al., 2025a; Pfaff et al., 2026; Xia et al., 2026), or iterative refinement (Yang et al., 2026). However, tools often optimize placements locally without global scene feedback, so individually valid constraints can become mutually incompatible as the scene evolves. An alternative is to retain placement decisions directly through LLM/VLM generation, preserving greater flexibility (Feng et al., 2023; Yang et al., 2025b; Bucher & Armeni, 2025; Pan & Liu, 2026). However, reliably grounding these predictions in physical constraints remains challenging, as sampling alone cannot recover when the model's proposal distribution is biased toward invalid configurations. Beyond these limitations, both paradigms leave scene-level style consistency underexplored. Style emerges from interactions among all assets in a scene and is therefore not well captured by optimizing them independently. Existing methods lack an explicit objective for holistic style coordination.

To bridge this gap, we present Dovetail3D, a unified agentic VLM that coordinates asset retrieval, selection, and placement and is post-trained with hierarchical reinforcement learning (RL) for joint layout and styling. As shown in Figure 1, Dovetail3D couples physically grounded control with model-driven flexibility: general-purpose rules prune invalid actions, exposing a physically

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.

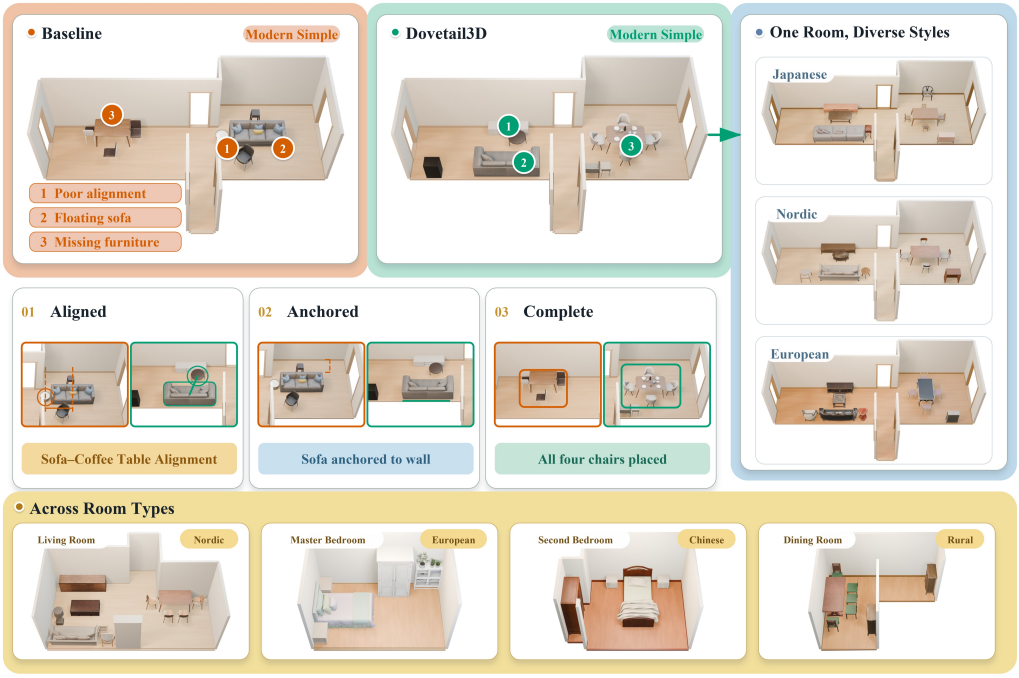

Figure 1: **Dovetail3D jointly improves layout and styling.** Given a room and a style request, the baseline produces an incomplete and spatially misaligned arrangement, whereas Dovetail3D selects coherent assets and organizes a complete inventory into a functional layout. The bottom insets highlight aligned focal relations, anchored activity zones, and complete functional groups. Dovetail3D also synthesizes diverse styles for the same room while preserving its geometry (right).

grounded yet flexible action space to the agent for subsequent decisions, while RL post-training improves global coordination and overall scene quality. This unified policy further supports diverse yet consistent furnishing schemes both within the scene and across different room types. Together, these capabilities integrate the reliability of tool-based pipelines and the flexibility of model-driven generation within a single scene-construction policy, leading to the following contributions.

- We formulate indoor scene synthesis as a sequential decision process and instantiate it with a unified workflow that coordinates asset composition, spatial layout, and style in an evolving context.
- We develop a decision-aligned agentic RL post-training algorithm tailored to 3D indoor scene synthesis. It combines general-purpose action pruning, hierarchical layout credit and asset-specific style attribution, enabling direct and globally coherent scene construction.
- Extensive experiments demonstrate that Dovetail3D achieves state-of-the-art performance in both layout quality and stylistic coherence across diverse indoor scenes.

2 Related Work

3D indoor scene synthesis. 3D indoor scene synthesis spans several complementary representations. Learned scene distributions generate object inventories and poses through autoregressive or diffusion-based models (Paschalidou et al., 2021; Tang et al., 2024; Yang et al., 2024a), whereas visual-prior methods lift the knowledge of 2D generative models into textured 3D environments (Höllein et al., 2023; Schult et al., 2024). Programmatic and procedural systems instead construct editable, simulation-ready scenes through executable rules or code (Deitke et al., 2022; Raistrick et al., 2024; Lu et al., 2026; Wang et al., 2026). These families provide complementary strengths in learned realism, visual appearance, and explicit control, while orthogonal work improves scene-aware retrieval and stylistic compatibility (Liu et al., 2023; Zhao et al., 2024; Marshall

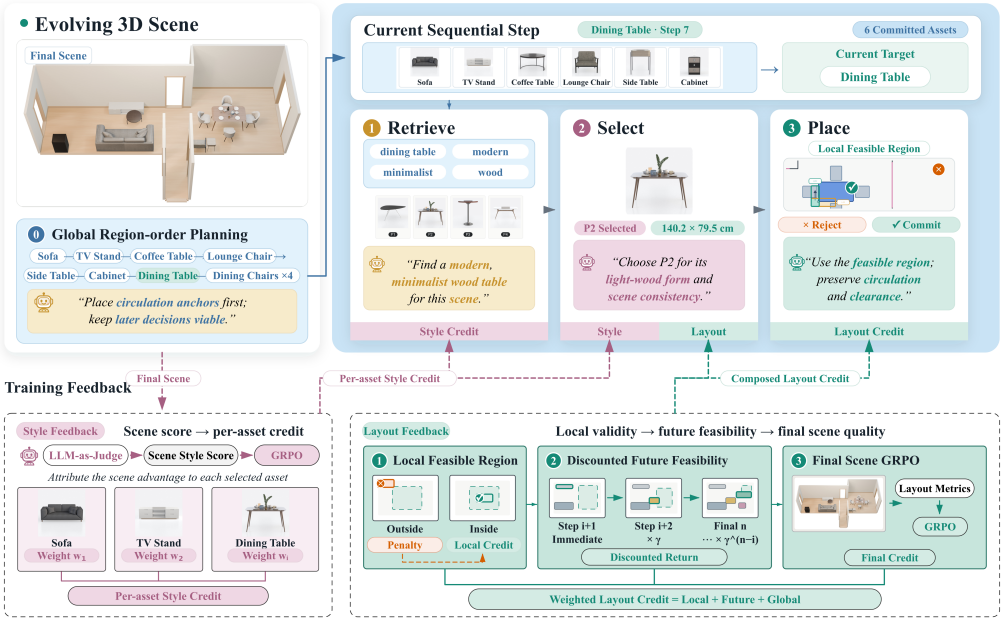

Figure 2: **Unified agentic workflow and decision-aligned credit assignment.** The upper panel shows global order planning followed by iterative retrieval, selection, and placement in an evolving scene. The lower panel decomposes scene-level style and layout feedback into asset- and action-level credits and routes them back to the decisions that produced the scene.

et al., 2025; Pan et al., 2025). Nevertheless, asset selection, styling, and spatial arrangement are usually handled by separate models or stages rather than optimized as coupled decisions.

Agentic scene synthesis and policy learning. Language- and vision-guided methods add semantic flexibility by using foundation models to directly propose scene plans and layouts (Feng et al., 2023; Çelen et al., 2024; Littlefair et al., 2025), to combine model predictions with geometric optimization (Sun et al., 2025a; Yang et al., 2024b; Bucher & Armeni, 2025; Deng et al., 2025), or to coordinate specialized tools through iterative agents (Ling et al., 2025; Yang et al., 2025a; Sun et al., 2025b; Pfaff et al., 2026; Xia et al., 2026). Recent systems further use preference optimization, verification, or reinforcement learning to improve spatial decisions (Yang et al., 2025b; 2026; Zhao et al., 2026). Across these variants, stronger rule grounding commonly comes from external optimizers or tools that can rigidify intermediate decisions, whereas model-driven generation preserves flexibility but offers weaker guarantees of physical validity and global consistency.

3 DOVETAIL3D: A UNIFIED SCENE SYNTHESIS WORKFLOW

This section first formulates 3D indoor scene synthesis as a constrained optimization problem and realizes it through a retrieve–select–place workflow in Section 3.1. We then present the design of physically grounded feasible regions in Section 3.2.

3.1 SEQUENTIAL SCENE SYNTHESIS

Problem formulation. Let R denote the room specification, including its floor boundary and openings. $\mathcal{G} = \{g_n\}_{n=1}^{N}$ denotes the requested furnishing units, σ an optional target style request, and $D(g_n)$ the admissible asset set for g_n. The output scene $S = \{(z_n, p_n)\}_{n=1}^{N}$ pairs each selected asset z_n with a placement $p_n = (x_n, y_n, o_n)$ in a shared pose space $\mathcal{P}$, where o_n is its discrete

orientation. Given general constraints $\{c_m\}_{m=1}^{M}$, scene synthesis solves

$$S^\star = \arg\max_{S} \ \lambda_{\text{lay}} U_{\text{lay}}(S; R, \mathcal{G}) + \lambda_{\text{sty}} U_{\text{sty}}(S; R, \sigma)$$
$$\text{s.t.} \quad z_n \in D(g_n),\ p_n \in \mathcal{P},\ \forall n, \qquad c_m(R, \mathcal{G}, S) = 1,\ \forall m, \tag{1}$$

where $c_m(R, \mathcal{G}, S) \in \{0, 1\}$ indicates whether the m-th constraint is satisfied, U_{lay} measures spatial quality, U_{sty} measures style alignment, and λ_{lay} and λ_{sty} are their trade-off weights. Style consistency is optimized when σ is omitted.

Retrieve–select–place workflow. Following the above formulation, Dovetail3D takes as input the room specification R, the furnishing requests $\mathcal{G}$, and an optional style request σ. At step 0 of Figure 2, the VLM uses these inputs to produce a permutation $(g_{i_1}, \ldots, g_{i_N})$ of the requested units; for clarity, we relabel this ordered sequence as $(g_1, \ldots, g_N)$ below. It then processes one request at a time through the retrieve–select–place cycle. Before decision n, the current-step context identifies the target g_n and provides the remaining requests $\mathcal{G}_{n:N} = (g_n, \ldots, g_N)$ together with a contact sheet of the assets committed to S_{n-1}.

Retrieve. In step 1 of Figure 2, the VLM receives g_n, σ, the accepted-asset contact sheet, and the complete product-attribute vocabulary. It selects scene-relevant attributes to form comprehensive queries covering size, style, material, and other properties. The local retriever ranks assets by the fraction of query terms matched in their metadata and returns the top-K candidate product images.

Select. In the step-2 Select block, the policy receives candidate dimensions and a composite image juxtaposing the candidate products with the accepted assets, then selects z_n according to request relevance, ensemble compatibility, and spatial feasibility.

Place. In step 3, the selected asset and current room geometry induce the local feasible region described below. Conditioned on this region and the evolving scene, the policy predicts p_n; the check gate either commits the valid asset–pose pair to form S_n or rejects it with localized violation feedback while leaving S_{n-1} unchanged. The updated asset contact sheet and scene geometry then condition the next cycle. After the final cycle, the completed scene produces the layout and style feedback routed through the workflow as described in Section 4.3 and the lower panel of Figure 2.

3.2 Physically Grounded Action-Space Pruning

Off-the-shelf VLMs can reason about coarse spatial relations, but their coordinate predictions remain brittle. To ground the predicted pose under physical constraints for a specific asset, a lightweight compiler retrieves the relevant local constraint set $\mathcal{C}_n(z_n) \subseteq \{c_m\}_{m=1}^{M}$ and constructs valid placements in the configuration space of the asset. Since indoor furniture is predominantly aligned with room boundaries, we discretize the asset orientation as $o \in \mathcal{O} = \{\text{up}, \text{down}, \text{left}, \text{right}\}$. For each orientation o, the construction proceeds in three steps:

Step 1. Obstacle map construction. We construct an orientation-dependent service footprint containing both its physical body and required usage clearances. The room boundary, existing furniture, fixed regions, and doorway clearances define where this footprint cannot be placed.

Step 2. Anchor-space sweep. We represent each placement by the oriented asset's lower-left corner, and define the anchor space as the set of all candidate coordinates for this reference point. We partition the anchor space using a 5-cm base grid cell. For a cell, its sweep is the union of all regions occupied by the service footprint as the anchor moves within the cell. We retain this cell only if its entire sweep remains inside the room and has no positive-area overlap with any obstacle in step 1, and the union of all retained cells forms the geometric safe domain $\Omega^{\text{geo}}_{n,o}(z_n)$.

Step 3. Relational filtering. Relations such as wall attachment, alignment, and inter-object distance are converted into a relational domain $\Omega^{\text{rel}}_{n,o}(z_n)$. The valid cells in $\Omega^{\text{geo}}_{n,o}(z_n) \cap \Omega^{\text{rel}}_{n,o}(z_n)$ are then merged for orientation o. Taking the union across orientations yields the feasible pose region $\Omega_n(z_n) \subseteq \mathcal{P}$. The agent is then permitted to generate any coordinate within the domain.

Our rule set consists of general-purpose constraints designed to cover most indoor layouts, with the complete rule inventory provided in Appendix A.1. For atypical or out-of-distribution room geometries, specific constraints can be added to, removed from, or modified within this general rule set without retraining the model, meanwhile the agent continues to apply its general spatial reasoning within the resulting action space. By contrast, both agent-generated rules and unconstrained coordinate prediction remain brittle on cases not encountered during training.

4 DECISION-ALIGNED CREDIT ASSIGNMENT

Effective credit assignment is critical for long-horizon agentic RL, as required by our sequential workflow. This need is underscored by terminal-score-only algorithms, which we find leads to pronounced training collapse in practice, motivating our hierarchical credit assignment scheme. We therefore extract intermediate feedback signals from the generation process to support effective RL optimization. We introduce hierarchical layout credit at different temporal scales in Section 4.1, attributed style credit in Section 4.2, and their joint action-level optimization in Section 4.3.

4.1 HIERARCHICAL LAYOUT CREDIT ASSIGNMENT

Layout feedback operates at three causal scales. Rejected placement proposals require local failure credit because the check gate leaves the scene unchanged. Accepted placements affect the feasible space available to later assets, while a terminal score evaluates the completed scene. We align one credit signal with each scale.

Local potential-difference credit. For the q-th proposal of one placement, let $v_q \in [0,1]$ be the normalized violation cost returned by the feasible-region compiler. It combines the invalid-footprint fraction with the distance to the nearest legal pose, as detailed in Appendix A.1. We denote the proposed pose by $\tilde{p}_q$, define $\Phi^{\mathrm{loc}}(\tilde{p}_q) = -v_q$, reset $v_0 = 0$ at the start of each placement, and set $v_q = 0$ for a valid proposal. The local credit is

$$r_q^{\mathrm{pot}} = \Phi^{\mathrm{loc}}(\tilde{p}_q) - \Phi^{\mathrm{loc}}(\tilde{p}_{q-1}) = v_{q-1} - v_q. \tag{2}$$

Thus, the first invalid attempt is penalized, retries are credited according to whether they reduce the violation, and successful recovery receives the correction potential.

Feasible-domain return credit. Complex combinatorial optimization requires identifying the action that creates a bottleneck in the whole solution, which terminal-only RLVR cannot reveal until the trajectory is complete. Our action pruning mechanism naturally exposes it during generation by measuring how much optimization space each action preserves for subsequent decisions.

For furnishing decision n in rollout b, let $\mathcal{A}_{b,n}$ contain all distinct legal (o,x,y) actions for the selected asset under the previously accepted placements, and let $\mathcal{A}_{b,n}^{\max}$ be the corresponding in-bound action set in the empty room. We define the normalized feasible-domain score for asset z_n and its discounted future return as

$$F_{b,n} = \frac{\log(1+|\mathcal{A}_{b,n}|)}{\log\left(1+|\mathcal{A}_{b,n}^{\max}|\right)}, \qquad G_{b,n}^{\mathrm{lay}} = \sum_{j=n+1}^{N} \gamma^{j-n-1} F_{b,j}, \tag{3}$$

where $\gamma \in (0,1]$ is the discount factor. This Monte Carlo return is computed retrospectively: each $F_{b,j}$ is evaluated when the future asset is actually selected and then accumulated backward.

Group-relative advantage credit. Let s_b^{lay} be the completion-aware terminal outcome for rollout b in a same-case rollout group indexed by B. It combines furnishing completion with the final evaluator of functional relationships, contextual layout quality, and navigability defined in Appendix A.2. With $\mathbf{s}^{\mathrm{lay}} = \{s_b^{\mathrm{lay}} : b \in B\}$, the group-relative advantage is

$$A_b^{\mathrm{lay}} = \frac{s_b^{\mathrm{lay}} - \mathrm{mean}(\mathbf{s}^{\mathrm{lay}})}{\mathrm{std}(\mathbf{s}^{\mathrm{lay}})}, \tag{4}$$

We use GRPO to optimize this outcome reward. Therefore, we define the resulting action-level layout credit alongside the joint credit as

$$c_{b,n}^{\mathrm{lay}} = \lambda_{\mathrm{loc}} r_q^{\mathrm{pot}} + \lambda_{\mathrm{seq}} G_{b,n}^{\mathrm{lay}} + \lambda_{\mathrm{final}} A_b^{\mathrm{lay}}, \tag{5}$$

where the coefficients λ_{loc}, λ_{seq}, and λ_{final} weight the three layout scales.

4.2 ATTRIBUTED STYLE CREDIT ASSIGNMENT

To align scene-level stylistic coherence with asset-level decisions, we use an LLM-as-Judge to score the complete asset set and then attribute the resulting advantage to the responsible assets.

Scene-level style score. For rollout b, let Z_b denote its final selected-asset set. The judge evaluates six dimensions: style consistency, color harmony, material and texture, shape and form, visual atmosphere, and alignment with the user-requested style. To improve consistency, it selects one of five semantic ratings rather than assigning a number directly: *excellent*, *minor mismatch*, *clear mismatch*, *major conflict*, or *multiple major conflicts*, mapped to $10, 8, 6, 4, 2$, respectively. Their weighted aggregate gives $s_b^{\text{sty}} \in [0, 10]$. With $\mathbf{s}^{\text{sty}} = \{s_b^{\text{sty}} : b \in B\}$, the style advantage is

$$A_b^{\text{sty}} = \frac{s_b^{\text{sty}} - \text{mean}(\mathbf{s}^{\text{sty}})}{\text{std}(\mathbf{s}^{\text{sty}})}. \tag{6}$$

Asset-level attribution. For each asset n in rollout b, the judge outputs *support*, *conflict*, *neutral*, or *not observable*, together with a contribution strength $w_{b,n} \in \{1, 2, 3\}$ of *minor*, *clear*, or *major* when applicable. Let $\mathcal{T}_b$ contain only supporting assets when $A_b^{\text{sty}} > 0$ and only conflicting assets when $A_b^{\text{sty}} < 0$; the other labels receive no credit. For a nonempty $\mathcal{T}_b$, we distribute the advantage proportionally while preserving its total magnitude:

$$c_{b,n}^{\text{sty}} = A_b^{\text{sty}} \frac{w_{b,n}}{\sum_{j \in \mathcal{T}_b} w_{b,j}}, \qquad n \in \mathcal{T}_b, \qquad \sum_{n \in \mathcal{T}_b} c_{b,n}^{\text{sty}} = A_b^{\text{sty}}. \tag{7}$$

Appendix A.3 provides the full scoring and attribution protocols.

4.3 Joint Policy Optimization

The lower panel of Figure 2 closes the loop with the sequential workflow by routing each feedback signal to the decisions that produced it. For asset n, the attributed style credit $c_{b,n}^{\text{sty}}$ is assigned to *retrieve* and *select* actions. Layout credit $c_{b,n}^{\text{lay}}$ follows the spatial decision path. Local potential-difference credit is assigned to every placement proposal, the feasible-domain return to the accepted *select* and *place* actions, and the terminal advantage to the step-0 ordering action and all accepted *select* and *place* actions. Actions outside a signal's causal path receive zero from that component.

$$C_{b,t} = \eta_{\text{lay}} C_{b,t}^{\text{lay}} + \eta_{\text{sty}} C_{b,t}^{\text{sty}}, \tag{8}$$

where η_{lay} and η_{sty} mix the layout and style credits. For a training case $x \sim \mathcal{D}_{\text{train}}$, let B be the index set of a same-case rollout group $\{\tau_b\}_{b \in B}$ sampled from the behavior policy π_{old}. Let $a_{b,t}$ and $h_{b,t}$ denote action t and its history in rollout b, respectively, and define $\rho_{b,t}(\theta) = \pi_\theta(a_{b,t} \mid h_{b,t}) / \pi_{\text{old}}(a_{b,t} \mid h_{b,t})$. We optimize the VLM policy π_θ with

$$\begin{aligned} J(\theta) = \mathbb{E}\Bigg[\frac{1}{|B|} \sum_{b \in B} \frac{1}{T_b} \sum_{t=1}^{T_b} \Big(& \min[\rho_{b,t}(\theta) C_{b,n}, \text{clip}(\rho_{b,n}(\theta), 1-\varepsilon, 1+\varepsilon) C_{b,t}] \\ & - \beta D_{\text{KL}}(\pi_\theta(\cdot \mid h_{b,t}) \,\|\, \pi_{\text{ref}}(\cdot \mid h_{b,t})) \Big) \Bigg], \end{aligned} \tag{9}$$

where T_b is the length of rollout b, π_{ref} is the reference policy, ε is the clipping range, and β is the KL weight. The objective preserves temporal and object-level attribution while jointly optimizing the shared policy that controls the complete workflow.

5 Experiments

5.1 Experimental Setup

Dataset and baselines. We evaluate on 3D-FRONT (Fu et al., 2021), representing each room by its 2D boundary and openings while using its furniture instances as input requests without exposing the ground-truth layout. Training uses 250 bedrooms and 192 living rooms. The test set includes 100 bedroom-family, 53 living-room, and 47 dining-room cases, each paired with a style request. We compare seven baselines spanning diffusion-based generation (DiffuScene (Tang et al., 2024)), LLM planning (LayoutGPT (Feng et al., 2023), I-Design (Çelen et al., 2024)), optimization-grounded synthesis (LayoutVLM (Sun et al., 2025a), Holodeck (Yang et al., 2024b)), and iterative construction (SceneWeaver (Yang et al., 2025a), ReSpace (Bucher & Armeni, 2025)). All GPT-dependent scene-generation methods use GPT-5.4. Full data and baseline details appear in Appendix B.

Table 1: Layout results across room types. All metrics are percentages (%). Cell backgrounds encode relative performance within each room type, with darker shades indicating better performance.

Method	Bedroom							Master Bedroom						
	CNT↑	NAV↑	ACN↑	OOB↓	COL↓	OOR↑	OAR↑	CNT↑	NAV↑	ACN↑	OOB↓	COL↓	OOR↑	OAR↑
LayoutGPT (*NeurIPS'23*)	76.4	90.9	11.5	17.0	53.1	79.5	62.9	77.1	92.1	13.7	27.3	51.3	75.9	64.1
LayoutVLM (*CVPR'25*)	94.4	91.8	27.7	19.2	87.6	62.4	34.6	95.9	92.5	15.9	16.0	92.3	60.1	40.7
I-Design (*ECCVW'24*)	72.3	98.4	50.3	28.4	18.8	48.9	22.9	81.1	98.3	45.4	31.1	20.0	42.7	17.4
DiffuScene (*CVPR'24*)	67.5	95.1	18.8	51.3	34.0	16.1	33.4	62.9	95.8	26.9	36.8	46.2	15.3	31.4
ReSpace (*ICLRW'26*)	97.3	96.7	26.6	7.3	4.8	16.6	47.1	98.3	95.3	22.3	6.7	10.3	11.8	45.3
SceneWeaver (*NeurIPS'25*)	58.6	95.6	53.8	0.0	0.0	54.7	43.5	65.1	95.1	52.1	2.3	2.7	56.5	43.3
Holodeck (*CVPR'24*)	86.0	97.7	82.1	0.3	0.0	24.0	83.6	90.9	97.0	71.8	2.3	0.0	25.5	88.8
Dovetail3D	96.5	100.0	94.9	0.0	0.0	100.0	100.0	95.4	100.0	90.5	0.0	0.0	100.0	100.0

Method	Living Room							Dining Room						
	CNT↑	NAV↑	ACN↑	OOB↓	COL↓	OOR↑	OAR↑	CNT↑	NAV↑	ACN↑	OOB↓	COL↓	OOR↑	OAR↑
LayoutGPT (*NeurIPS'23*)	55.9	98.2	34.8	20.8	47.2	47.6	88.6	65.9	98.1	50.9	12.9	27.7	63.6	91.0
LayoutVLM (*CVPR'25*)	88.4	85.7	39.7	6.2	84.9	34.0	80.8	82.5	81.8	32.3	6.7	57.4	40.0	85.5
I-Design (*ECCVW'24*)	27.1	98.8	26.4	30.3	0.0	22.0	79.5	70.7	94.3	48.7	20.3	12.5	39.7	74.3
DiffuScene (*CVPR'24*)	63.9	98.8	36.2	53.8	49.1	51.4	81.6	67.0	95.2	36.6	53.4	57.4	53.8	78.8
ReSpace (*ICLRW'26*)	99.8	99.4	44.0	1.0	15.1	40.6	88.2	99.4	97.6	46.3	2.4	27.7	41.2	90.9
SceneWeaver (*NeurIPS'25*)	68.6	96.4	54.9	0.3	3.8	27.2	83.1	62.8	93.5	44.6	1.2	0.0	33.4	72.2
Holodeck (*CVPR'24*)	80.9	98.6	64.1	0.5	0.0	67.8	93.0	61.5	96.8	45.9	0.8	0.0	38.4	41.2
Dovetail3D	88.4	99.8	74.6	0.0	0.0	86.4	100.0	86.8	97.2	73.7	0.0	0.0	82.3	91.2

Training and Evaluation. We use Qwen3.5-9B as our base model and full hyperparameters are provided in Appendix B. Following SceneEval (Tam et al., 2026), we report required-object recall (CNT), navigability (NAV), out-of-bound rate (OOB), and collision rate (COL), together with activity-centric connectivity (ACN) and rule-based object–object (OOR) and object–room (OAR) relations. During training, Qwen3.5-Flash provides efficient style scores and asset-level attribution feedback. To ensure feedback quality in evaluation, we use GPT-5.5 to perform the final style and scene-quality scoring following SceneWeaver (Yang et al., 2025a). For human evaluation, we randomly sample 20 test cases shared across all eight methods, yielding 160 scenes. Five annotators rate each scene for style, realism (REAL), functionality (FUNC), and layout quality (LAY), and we report the mean ratings across annotators.

5.2 Main Results

Table 1 shows that Dovetail3D achieves the strongest joint layout performance. It produces zero out-of-bound and collision violations across all room types, maintains near-perfect navigability, and improves ACN and OOR over the strongest external baseline by 10.5–22.8 and 18.6–24.1 percentage points, respectively. ReSpace and LayoutVLM illustrate that high object recall does not guarantee scene usability: both achieve strong CNT but remain substantially lower in ACN and relation scores. Holodeck and SceneWeaver exhibit the complementary trade-off. Their low OOB and COL rates show that explicit constraints and local repair can enforce basic geometric validity, yet they underperform Dovetail3D on metrics that more directly reflect layout quality, including ACN and relational correctness. Table 2 extends the comparison to style and overall scene quality. Dovetail3D obtains the highest style score in every room type, exceeding the strongest baseline by 0.4–0.7 points, while improving aggregate scene quality by 0.2–0.7 points. Its functionality score is also consistently strongest, indicating that the relational gains in Table 1 remain visible in rendered-scene evaluation. Human ratings provide independent confirmation. Together, these metrics show that Dovetail3D jointly improves layout quality and stylistic coherence in indoor scene synthesis.

5.3 Ablation Studies

Table 3 compares Dovetail3D with the same backbone and workflow without RL. Post-training reduces invalid proposals and rollout length while improving future feasibility, layout, and perceptual

Table 2: Style and scene-quality evaluation across room types. REAL, FUNC, and LAY denote realism, functionality, and layout quality; the adjacent Human columns report human ratings for Style and Scene Quality on the sampled subset. All scores range from 0 to 10; higher is better.

Method	Bedroom							Master Bedroom						
	Style↑	Hum.↑	REAL↑	FUNC↑	LAY↑	Judge↑	Hum.↑	Style↑	Hum.↑	REAL↑	FUNC↑	LAY↑	Judge↑	Hum.↑
LayoutGPT *(NeurIPS'23)*	7.5	8.0	6.1	5.5	5.4	5.7	4.9	7.5	4.4	6.3	6.1	5.3	5.9	4.5
LayoutVLM *(CVPR'25)*	7.5	7.2	4.8	4.1	3.6	4.2	3.7	7.3	6.0	4.6	3.9	3.5	4.0	2.9
I-Design *(ECCVW'24)*	7.4	6.5	4.8	5.4	4.4	4.8	4.1	7.8	6.0	5.4	6.2	4.8	5.5	4.9
DiffuScene *(CVPR'24)*	5.9	7.2	5.3	4.6	4.4	4.8	5.5	6.2	6.4	4.8	4.2	4.3	4.4	4.1
ReSpace *(ICLRW'26)*	8.1	8.4	6.0	6.0	5.2	5.7	6.7	7.7	8.0	6.2	5.8	5.2	5.7	5.1
SceneWeaver *(NeurIPS'25)*	7.5	8.4	4.5	5.3	4.0	4.6	4.7	8.5	5.6	5.2	5.8	4.7	5.2	3.5
Holodeck *(CVPR'24)*	7.5	8.8	5.8	6.8	4.9	5.8	8.4	8.1	7.2	6.2	7.1	5.1	6.1	8.0
Dovetail3D	8.8	9.2	6.1	7.3	5.7	6.4	9.6	8.9	8.8	6.3	7.4	5.7	6.5	9.6

Method	Living Room							Dining Room						
	Style↑	Hum.↑	REAL↑	FUNC↑	LAY↑	Judge↑	Hum.↑	Style↑	Hum.↑	REAL↑	FUNC↑	LAY↑	Judge↑	Hum.↑
LayoutGPT *(NeurIPS'23)*	5.3	5.1	5.5	4.1	4.8	4.8	5.6	8.6	7.3	5.2	4.7	4.8	4.9	6.7
LayoutVLM *(CVPR'25)*	6.1	5.1	4.8	3.9	3.6	4.1	3.3	8.3	6.7	4.9	4.3	3.9	4.4	3.1
I-Design *(ECCVW'24)*	3.6	3.0	2.2	2.5	1.7	2.1	5.5	7.5	6.7	4.3	4.8	3.8	4.3	5.2
DiffuScene *(CVPR'24)*	6.6	8.0	3.9	3.6	3.1	3.6	3.6	7.8	5.3	4.0	4.2	3.4	3.9	2.9
ReSpace *(ICLRW'26)*	6.3	7.4	5.7	5.2	4.8	5.2	5.7	8.2	8.0	5.0	4.1	3.9	4.3	5.9
SceneWeaver *(NeurIPS'25)*	8.2	6.0	5.0	5.2	4.2	4.8	4.8	7.7	7.0	4.5	4.6	3.9	4.3	5.2
Holodeck *(CVPR'24)*	6.1	6.4	4.7	5.9	3.8	4.8	6.9	5.6	8.0	3.4	3.8	2.5	3.2	6.7
Dovetail3D	8.8	9.4	5.3	6.3	4.6	5.4	8.0	9.1	8.0	5.7	6.3	4.9	5.6	8.8

quality, showing that RL strengthens spatial reasoning without additional test-time tools. The component ablations further reveal how the credits interact. Removing the local potential-difference credit r^{pot} raises PIR from 0.3% to 15.9% and the average rollout length from 23.4 to 27.7 steps, showing that invalid-action feedback improves both validity and decision efficiency. Without the feasible-domain return G^{lay}, the policy preserves less space for subsequent placements and loses quality across several layout metrics. Removing the terminal credit A^{lay} causes a broader decline across feasible space, geometric quality, and perceptual scores, supporting its role as global scene-level feedback. The objectives are also coupled: removing the attributed style credit C^{sty} lowers not only style quality but also ACN, because style supervision affects asset retrieval and selection, whose geometry in turn shapes later placements. Thus, each credit contributes a distinct signal, while their combination supports coordinated layout and style decisions.

Table 3: Credit-component ablation on Living Room. C^{sty}, r^{pot}, G^{lay}, and A^{lay} denote attributed style credit, local potential-difference credit, feasible-domain return, and terminal GRPO credit, respectively. PIR is the invalid-placement rate, Steps is the average rollout length, and ΣF is the cumulative normalized feasible-domain score across a rollout. Darker cell shades indicate better within-column performance after rounding.

Variant	Rollout			Layout (%)					Scene Quality (0–10)				
	PIR↓	Steps↓	ΣF↑	CNT↑	NAV↑	ACN↑	OOR↑	OAR↑	Style↑	REAL↑	FUNC↑	LAY↑	Judge↑
w/o C^{sty}	5.4	25.1	147.3	88.4	99.4	69.1	86.2	100	8.2	5.2	6.2	4.5	5.3
w/o r^{pot}	15.9	27.7	147.3	87.7	98.9	71.5	85.8	99.0	8.7	5.2	6.0	4.4	5.2
w/o G^{lay}	1.5	23.4	151.5	86.6	98.1	71.0	88.1	99.4	8.5	5.2	6.2	4.3	5.2
w/o A^{lay}	1.5	23.6	150.0	88.5	98.3	73.9	87.8	99.1	8.2	5.3	6.2	4.4	5.3
w/o RL	16.2	27.9	141.7	79.0	99.8	73.2	82.0	100	8.3	5.2	5.7	4.5	5.1
Dovetail3D	0.3	23.4	157.9	88.4	99.8	74.6	86.4	100	8.8	5.3	6.3	4.6	5.4

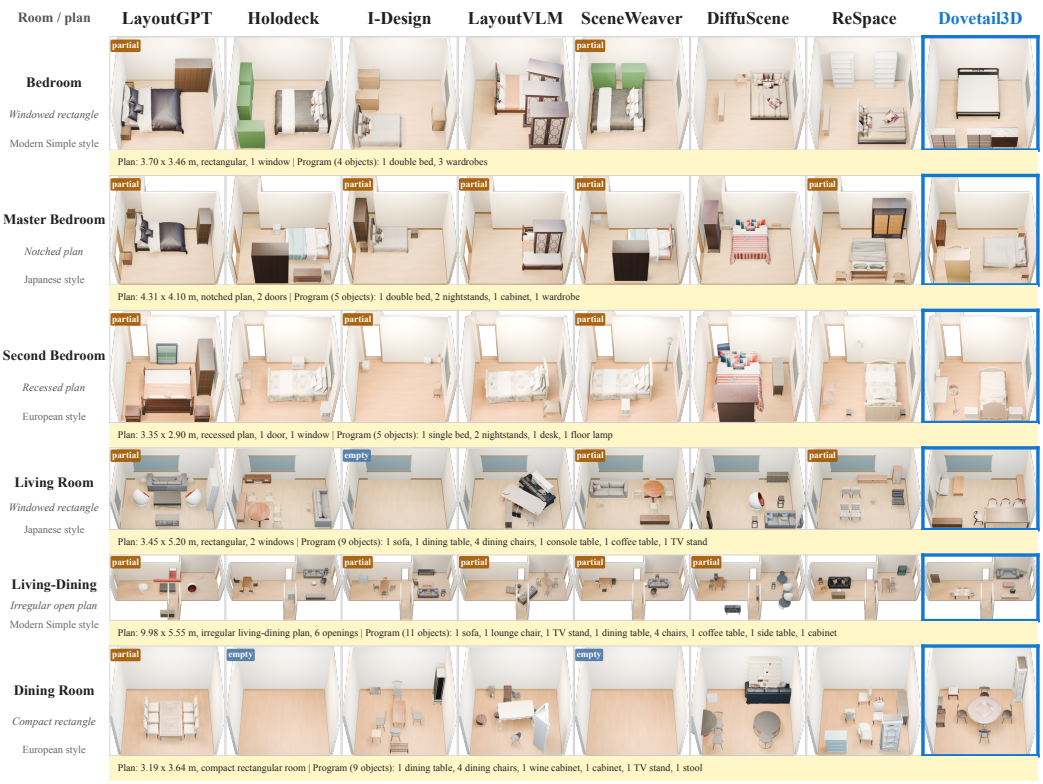

Figure 3: Qualitative comparison on six challenging cases. Rows specify the room, furnishing program, and target style; columns compare seven baselines with Dovetail3D under a common per-scene viewpoint. *Partial* and *empty* mark incomplete and failed scene construction, respectively.

5.4 Case Study

Figure 3 presents six challenging cases that stress furnishing completeness and geometric feasibility under the same room and target style. Additional cases are provided in Appendix C.1. Optimization-based pipelines gain control by compiling model decisions into explicit constraints, but locally specified relations can conflict as the scene evolves and lack global feedback. Model-driven pipelines retain flexibility, yet weakly constrained proposals cannot be repaired when sampling yields no valid layout. The cases expose both failures. In the Living-Dining room case, Holodeck discards a relation that refers to a later object and optimizes an incomplete plan. I-Design similarly produces a cyclic relational graph, while SceneWeaver deletes objects when local repair fails. Among model-driven methods, LayoutGPT preserves one-shot inventory omissions, DiffuScene replaces requested wardrobes with nightstands in the Bedroom case, and ReSpace retains an overlap in the LivingRoom case because every sampled trajectory is flawed. Dovetail3D instead uses feasible regions for local control and RL-derived global feedback, preserving flexibility while learning scene-level feasibility.

6 Conclusions

We present Dovetail3D, a unified agentic VLM for 3D indoor scene synthesis that jointly coordinates asset retrieval, selection, placement, and styling throughout an evolving scene. Instead of delegating placement to a disconnected optimizer, Dovetail3D uses physically grounded rules to expose structured feasible action regions while retaining the model's control over final decisions. Its decision-aligned RL post-training further connects local violation feedback, future optimization space, terminal layout quality, and fine-grained asset-level style attribution to the actions responsible for them. Experiments on 3D-FRONT show that Dovetail3D consistently improves layout quality and aesthetic coherence over strong generative, optimization-based, and agentic baselines.

AI USE STATEMENT

In this work, we have not used generative AI tools for any other tasks with required disclosure, and the rest of the required disclosure tasks are not applicable to this work. Additionally, we used generative AI tools to edit a research paper to improve readability, format references, and create or modify scientific figures or images. We have reviewed all AI-assisted work. All AI-assisted edits and artifacts were manually reviewed and verified by the authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Section 5.1 summarizes the datasets, baselines, training setup, and evaluation metrics used in our experiments. Appendix B provides the data preprocessing procedure and implementation details for each baseline, while Appendix B.3 specifies the dataset splits, training and rollout hyperparameters, and evaluation protocol. We provide our source code as supplementary material.

REFERENCES

Martin JJ Bucher and Iro Armeni. Respace: Text-driven autoregressive 3d indoor scene synthesis and editing. *arXiv preprint arXiv:2506.02459*, 2025.

Ata Çelen, Guo Han, Konrad Schindler, Luc Van Gool, Iro Armeni, Anton Obukhov, and Xi Wang. I-Design: Personalized llm interior designer. *arXiv preprint arXiv:2404.02838*, 2024.

Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Jordi Salvador, Kiana Ehsani, Winson Han, Eric Kolve, Ali Farhadi, Aniruddha Kembhavi, and Roozbeh Mottaghi. ProcTHOR: Large-scale embodied ai using procedural generation. In *Advances in Neural Information Processing Systems*, 2022.

Wei Deng, Mengshi Qi, and Huadong Ma. Global-local tree search in VLMs for 3d indoor scene generation. *arXiv preprint arXiv:2503.18476*, 2025.

Weixi Feng, Wanrong Zhu, Tsu-jui Fu, Varun Jampani, Arjun Akula, Xuehai He, Sugato Basu, Xin Eric Wang, and William Yang Wang. LayoutGPT: Compositional visual planning and generation with large language models. *Advances in Neural Information Processing Systems*, 2023.

Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, and Hao Zhang. 3D-FRONT: 3d furnished rooms with layouts and semantics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

Lukas Höllein, Ang Cao, Andrew Owens, Justin Johnson, and Matthias Nießner. Text2Room: Extracting textured 3d meshes from 2d text-to-image models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

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. *arXiv preprint arXiv:2505.02836*, 2025.

Gabrielle Littlefair, Niladri Shekhar Dutt, and Niloy J. Mitra. FlairGPT: Repurposing LLMs for interior designs. *arXiv preprint arXiv:2501.04648*, 2025.

Jingyu Liu, Wenhan Xiong, Ian Jones, Yixin Nie, Anchit Gupta, and Barlas Oğuz. CLIP-Layout: Style-consistent indoor scene synthesis with semantic furniture embedding. *arXiv preprint arXiv:2303.03565*, 2023.

Yuhao Lu, Weichen Zhang, Wenyi Xiao, Haohui Chen, and Yiyun Fei. Learning an interior layout policy in a domain specific language action space. *arXiv preprint arXiv:2608.07547*, 2026.

Kelly O. Marshall, Omid Poursaeed, Sergiu Oprea, Amit Kumar, Anushrut Jignasu, Chinmay Hegde, Yilei Li, and Rakesh Ranjan. Decorum: A language-based approach for style-conditioned synthesis of indoor 3d scenes. *arXiv preprint arXiv:2503.18155*, 2025.

Paul Merrell, Eric Schkufza, Zeyang Li, Maneesh Agrawala, and Vladlen Koltun. Interactive furniture layout using interior design guidelines. In *ACM SIGGRAPH 2011 Papers*, 2011.

Zhenyu Pan and Han Liu. Metaspatial: Reinforcing 3d spatial reasoning in vlms for the metaverse. In *International Conference on Learning Representations*, volume 2026, pp. 117974–117993, 2026.

Zhenyu Pan, Yucheng Lu, and Han Liu. MetaFind: Scene-aware 3d asset retrieval for coherent metaverse scene generation. In *Advances in Neural Information Processing Systems*, 2025.

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

Nicholas Pfaff, Thomas Cohn, Sergey Zakharov, Rick Cory, and Russ Tedrake. SceneSmith: Agentic generation of simulation-ready indoor scenes. In *International Conference on Machine Learning*, 2026.

Alexander Raistrick, Lingjie Mei, Karhan Kayan, David Yan, Yiming Zuo, Beining Han, Hongyu Wen, Meenal Parakh, Stamatis Alexandropoulos, Lahav Lipson, Zeyu Ma, and Jia Deng. Infinigen Indoors: Photorealistic indoor scenes using procedural generation. *arXiv preprint arXiv:2406.11824*, 2024.

Jonas Schult, Sam Tsai, Lukas Höllein, Bichen Wu, Jialiang Wang, Chih-Yao Ma, Kunpeng Li, Xiaofang Wang, Felix Wimbauer, Zijian He, Peizhao Zhang, Bastian Leibe, Peter Vajda, and Ji Hou. ControlRoom3D: Room generation using semantic proxy rooms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

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

Fan-Yun Sun, Shengguang Wu, Christian Jacobsen, Thomas Yim, Haoming Zou, Alex Zook, Shangru Li, Yu-Hsin Chou, Ethem Can, Xunlei Wu, Clemens Eppner, Valts Blukis, Jonathan Tremblay, Jiajun Wu, Stan Birchfield, and Nick Haber. 3D-Generalist: Self-improving vision-language-action models for crafting 3d worlds. *arXiv preprint arXiv:2507.06484*, 2025b.

Hou In Ivan Tam, Hou In Derek Pun, Austin T. Wang, Angel X. Chang, and Manolis Savva. SceneEval: Evaluating semantic coherence in text-conditioned 3d indoor scene synthesis, 2026. URL https://arxiv.org/abs/2503.14756.

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.

Puyi Wang, Yuhao Wang, Linjie Li, Zhengyuan Yang, Kevin Qinghong Lin, Yangguang Li, and Yu Cheng. SceneCode: Executable world programs for editable indoor scenes with articulated objects. *arXiv preprint arXiv:2605.19587*, 2026.

Hongchi Xia, Xuan Li, Zhaoshuo Li, Qianli Ma, Jiashu Xu, Ming-Yu Liu, Yin Cui, Tsung-Yi Lin, Wei-Chiu Ma, Shenlong Wang, Shuran Song, and Fangyin Wei. SAGE: Scalable agentic 3d scene generation for embodied ai. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22358–22368, 2026.

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

Yandan Yang, Baoxiong Jia, Shujie Zhang, and Siyuan Huang. SceneWeaver: All-in-one 3d scene synthesis with an extensible and self-reflective agent. In *Advances in Neural Information Processing Systems*, 2025a.

Yixuan Yang, Zhen Luo, Tongsheng Ding, Junru Lu, Mingqi Gao, Jinyu Yang, Victor Sanchez, and Feng Zheng. OptiScene: Llm-driven indoor scene layout generation via scaled human-aligned data synthesis and multi-stage preference optimization. *arXiv preprint arXiv:2506.07570*, 2025b.

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

Zhijing Yang, Haocheng Lin, Zhihua Xu, Haojie Li, Keze Wang, Liang Lin, and Tianshui Chen. Agentic Designer: Progressive multi-agent collaboration for structure-aware interior layout generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026. doi: 10.1109/TPAMI.2026.3711762.

Song-Hai Zhang, Shao-Kui Zhang, Yuan Liang, and Peter Hall. A survey of 3d indoor scene synthesis. *Journal of Computer Science and Technology*, 34(3):594–608, 2019.

Yang Zhao, Shizhao Sun, Meisheng Zhang, Yingdong Shi, Xubo Yang, and Jiang Bian. SceneReVis: A self-reflective vision-grounded framework for 3d indoor scene synthesis via multi-turn rl. *arXiv preprint arXiv:2602.09432*, 2026.

Yiqun Zhao, Zibo Zhao, Jing Li, Sixun Dong, and Shenghua Gao. RoomDesigner: Encoding anchor-latents for style-consistent and shape-compatible indoor scene generation. In *Proceedings of the International Conference on 3D Vision*, pp. 1413–1423, 2024.

Appendix

A Additional Method Details

This section records the geometric rule design and the final layout and style evaluation used by the workflow and credit assignment.

A.1 Feasible-Region Rule Design

The local rule set is organized into physical, use-space, relational, and look-ahead predicates. Physical predicates enforce room containment, non-overlap with fixed structures and accepted furniture, and clearance around openings. Use-space predicates reserve the free space required to approach or operate an asset. Relational predicates encode wall support, facing, alignment, and anchor-relative placement, while look-ahead predicates reserve the joint space needed by dependent furniture. Each rule is instantiated from the room geometry, selected asset footprint, and accepted scene state; Table 4 summarizes the operational inventory.

Table 4: Rules used for local feasible-region construction. Distances are measured in centimeters.

Category	**Representative Rule**
Physical validity	The oriented footprint must remain inside the floor polygon and be disjoint from fixed structures, accepted furniture, and door-clearance regions.
Use-space clearance	Front clearances are 55 for beds, 70 for desks, 60 for vanities and tall cabinets, and 30 for low cabinets; a bed additionally preserves a 55-wide passage on at least one side.
Wall support	Beds attach by the headboard; wardrobes, cabinets, desks, vanities, and TV units attach by the back. The wall-contact tolerance is 10, and wall-mounted TVs are limited to a depth of 5.
Paired relations	Desk–chair and vanity–chair contact use a tolerance of 3, dining table–chair and bed–nightstand contact use 10, and a side table must lie within 3 of its assigned seat. Facing and lateral-overlap tests are applied jointly.
Group composition	A dining-table action is legal only if the complete requested chair set can be assigned to symmetric slots, with a 5 seat gap and 20 rear clearance. Sofa, coffee-table, side-table, stool, and TV relations are compiled in the same anchor-relative manner.

The policy emits an orientation and integer-centimeter coordinates. For expensive two-dimensional relational tests, the compiler conservatively evaluates a 5-cm lattice while retaining exact relation boundaries; line and point anchors remain at centimeter resolution. The resulting rectangles, segments, and points are unioned per orientation, so an overlapping legal region is counted only once.

For an invalid proposal, the physical violation is the square root of the fraction of its footprint lying outside the room or intersecting furniture or door-clearance geometry. Relational deviation is normalized as $1 - \exp(-d/100)$, where d is the distance in centimeters to the nearest legal pose, with an additional orientation-mismatch distance when needed. For retry q, their union cost is $v_q = 1-(1-v_q^{\mathrm{geo}})(1-v_q^{\mathrm{rel}}) \in [0,1]$, which defines the potential used by the local difference credit. Thus the implementation does not tune an independent weight for every rule: physical overlaps are unioned geometrically, and all remaining rule deviations are summarized by distance to the compiled feasible domain.

Feasible-region evolution. Figure 4 visualizes how the local rule system becomes a state-dependent action mask. Once an asset and its dimensions are fixed, the environment computes orientation-specific legal anchor regions from the room geometry and the accepted scene state. After each accepted placement, the next asset receives a newly computed feasible region; the policy therefore chooses coordinates from a valid but continually evolving action space.

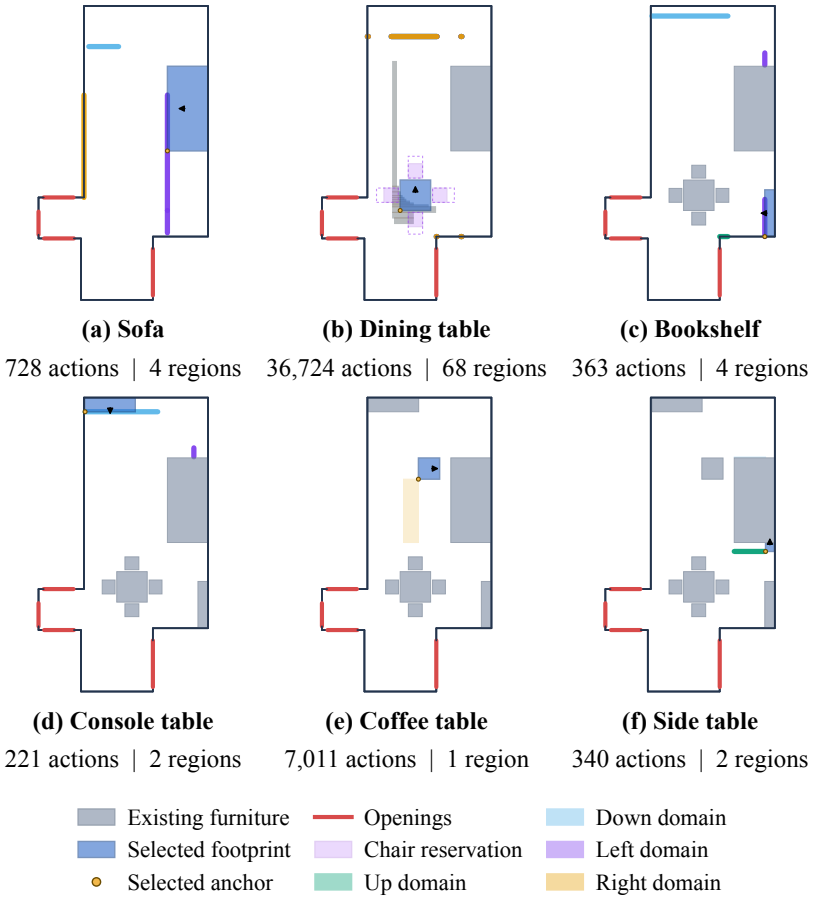

Figure 4: Evolution of the feasible coordinate domain during one rollout. Previously accepted furniture is shown in gray, colored regions denote legal anchors for different orientations, and the selected placement is highlighted in blue. The domain is recomputed after every accepted action, exposing how an early asset decision reshapes the action space available to subsequent furniture.

A.2 Final Layout Score

The global evaluator measures properties that emerge only from the joint arrangement. Let u_b^{func}, u_b^{ctx}, and u_b^{nav} denote functional relationships, contextual layout quality, and navigability for rollout b, each normalized to $[0, 10]$. The final layout-quality score is

$$\ell_b^{\text{lay}} = \frac{5}{14} u_b^{\text{func}} + \frac{1}{7} u_b^{\text{ctx}} + \frac{1}{2} u_b^{\text{nav}}. \tag{10}$$

Functional relationships evaluate the relative pose of furniture that forms an activity group, whereas contextual quality covers scene-level architectural and compositional compatibility not already guaranteed by the local gate. If navigability cannot be computed, the two available terms are renormalized to weights $5/7$ and $2/7$; an empty layout receives zero.

Connectivity metric. Local validity alone does not ensure that a completed scene remains navigable. We rasterize collision-free configuration space at 5-cm resolution and represent the occupant by a disc of radius 20 cm. From an entrance, or from a primary activity zone when no entrance is available, the evaluator computes the widest path to every active furniture-use zone. Bottleneck width is mapped linearly between 40 cm (feasible) and 60 cm (comfortable), and the target scores are averaged to obtain u_b^{nav}. Figure 5 illustrates the distinction between locally valid furniture and globally connected use zones.

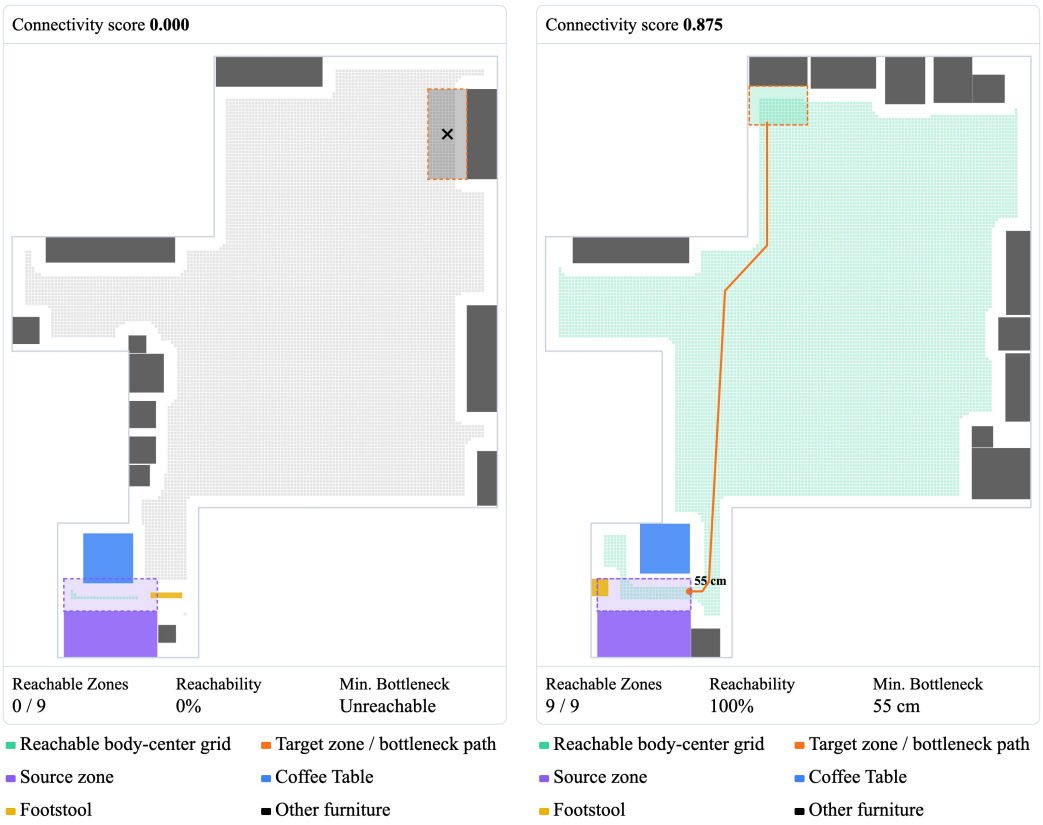

Figure 5: A before–after connectivity case. Furniture footprints and required use regions can each be locally valid while jointly fragmenting navigable free space. The global connectivity objective distinguishes these cases by evaluating reachability between openings and furniture interaction regions in the completed scene.

Table 5: Style-scoring rubric used during training and evaluation.

Dimension	Weight	Criterion
Style consistency	0.32	Agreement with the requested style, or internal design-language consistency when no style is given.
Color harmony	0.24	Compatibility of dominant and accent colors.
Material and texture	0.14	Compatibility of visible materials, finishes, and surface character.
Shape and form	0.12	Compatibility of silhouettes, line language, and visual weight.
Visual atmosphere	0.13	A conservative derived judgment determined by style, color, and form.
User-request alignment	0.05	Satisfaction of aesthetic requirements that are directly verifiable from the images.

A.3 Final Style Score and Attribution

This subsection covers the multi-dimensional prompt-based style score and its attribution to individual assets.

Multi-dimensional style scoring. The rubric evaluates the complete selected-asset ensemble against the requested style rather than scoring each item independently. Table 5 gives the six dimensions and fixed aggregation weights.

To improve calibration, the judge selects one semantic level per dimension rather than producing a number directly: *excellent*, *minor mismatch*, *clear mismatch*, *major conflict*, or *multiple major conflicts*, mapped deterministically to $10, 8, 6, 4, 2$. A dimension marked *not observable* is removed from both numerator and denominator. For observable dimensions $\mathcal{D}_b^{\text{obs}}$, the score is

$$s_b^{\text{sty}} = \frac{\sum_{d \in \mathcal{D}_b^{\text{obs}}} \omega_d e_{b,d}}{\sum_{d \in \mathcal{D}_b^{\text{obs}}} \omega_d}, \tag{11}$$

where $e_{b,d}$ is the mapped semantic score and ω_d is the weight in Table 5. Visual atmosphere is fixed to the worst observable level among style consistency, color harmony, and shape and form. User-request alignment is unobservable when the request is absent or cannot be verified visually.

Object-level style attribution. For every visible tile $P_{b,n}$ representing asset n in rollout b, the judge returns *support*, *conflict*, *neutral*, or *not observable*. Support indicates that the asset reinforces the evaluated combination, conflict identifies an asset participating in an explicit inconsistency, and the latter two labels receive no asset-level credit. The contribution strengths *minor*, *clear*, and *major* map to $w_{b,n} = 1, 2, 3$. For a positive group-relative style advantage, only supporting assets form $\mathcal{T}_b$; for a negative advantage, only conflicting assets do. Equation 7 then distributes the advantage in proportion to $w_{b,n}$ and preserves its sum.

Attribution is accepted only when every visible tile appears exactly once and maps to a valid product and region identifier in the rollout. The parser enforces a closed schema, requires conflict records and product attributions to reference each other consistently, and checks the deterministic visual-atmosphere rule. A malformed response, an unknown identifier, or an attribution that cannot be bound to the corresponding retrieval or selection action yields no asset-level style credit.

Judge prompts. The system prompt states that the input is a contact sheet of isolated product images rather than a room render, and forbids judging layout, circulation, room-scale proportion, or lighting. The user message provides the target style, any visually verifiable aesthetic request, a manifest that binds each printed tile label to a product and furnishing region, and the contact sheet. It asks the judge to identify explicit conflicts before rating the six style dimensions and assigning per-product attributions. The response is restricted to the following logical schema; all keys are required and additional keys are rejected.

```
{
  "style_conflict_inventory": [{
    "conflict_id": "C1",
    "product_tile_ids": ["P1", "P3"],
    "conflict_type": "...",
    "severity": "minor_mismatch|clear_mismatch|major_conflict|multiple_major_conflicts",
    "visual_axes": ["..."],
    "is_direct_conflict": true,
    "evidence": "..."
  }],
  "product_attributions": [{
    "tile_id": "P1",
    "attribution": "support|neutral|conflict|not_observable",
    "strength": "minor|clear|major|none",
    "affected_dimensions": ["..."],
    "conflict_ids": ["C1"],
    "evidence": "..."
  }],
  "dimension_levels": {
    "style_consistency": "...",
    "color_harmony": "...",
    "material_texture": "...",
    "shape_and_form": "...",
    "visual_atmosphere": "...",
    "user_request_alignment": "..."
  }
}
```

During training, Qwen3.5-Flash produces one accepted evaluation with at most three parsing attempts. Final Style and LLM-as-Judge evaluation uses GPT-5.5.

Table 6: **Comparison of scene construction mechanisms.** *Direct* denotes model-decided final placement; *Tool*, an external layout solver, optimizer, or tool; *Refine*, iterative scene-level refinement; *Gate*, feasibility checking before an action is committed; *A–L–S*, joint asset–layout–styling coordination; and *RL*, an RL-trained spatial policy. ✓, ○, and – indicate full, partial, and no support, respectively.

	Spatial Execution				Unified Learning	
Method	**Direct**	**Tool**	**Refine**	**Gate**	**A–L–S**	**RL**
LayoutGPT	✓	–	–	–	–	–
LayoutVLM	○	✓	–	–	–	–
I-Design	–	✓	○	–	–	–
Holodeck	–	✓	–	–	–	–
DiffuScene	✓	–	–	–	–	–
ReSpace	✓	–	○	–	–	✓
SceneWeaver	–	✓	✓	–	○	–
Dovetail3D	✓	–	–	✓	✓	✓

B Experimental Setup

B.1 Data Preprocessing

We convert each 3D-FRONT room into a centimeter-scale two-dimensional specification. The largest closed boundary extracted from the triangulated floor mesh defines the floor polygon; consecutive collinear segments are merged, and high-confidence door and window meshes are projected onto matching boundary spans. Malformed or unmatched openings are retained only as provenance and are not promoted to hard clearance constraints.

Valid floor-supported furniture instances are mapped to a normalized category vocabulary, with scene roles preserved when they are more informative than the underlying asset label. We create one furnishing request per instance, thereby retaining multiplicity, but remove the ground-truth product identity, pose, orientation, visual tags, and exact dimensions. The source footprint supplies only a one-sided retrieval bound of 110% of its width and depth; compact replacements remain eligible. Ceiling and pendant lamps are excluded because the workflow models ground-plane placement. The bedroom-family labels (Bedroom, MasterBedroom, and SecondBedroom) are retained for reporting but share the same training and evaluation interface.

The nine target styles are American, Chinese, European, Japanese, light luxury, modern simple, Nordic, retro, and rural. They are assigned after the room-disjoint split is fixed, so style assignment does not expose held-out geometry or ground-truth assets.

B.2 Baselines and Adaptations

Table 6 summarizes the scene-construction mechanisms represented by the evaluated systems.

Each baseline adapter begins from the canonical case and exposes the fields accepted by the original method. Their predictions are normalized to centimeter-scale catalog boxes with lower-left positions and yaw for common geometry and rendering evaluation.

LayoutGPT. We retain the 3D CSS prompting and parsing protocol, selecting train-only room exemplars by room dimensions ($K = 8$ for bedrooms and $K = 4$ for living rooms). The adapter supplies room geometry and the request without ground-truth furniture, then parses the predicted seven-field records and grounds them in 3D-FUTURE for evaluation.

LayoutVLM. We convert the canonical floor polygon from centimeters to its native meter-scale scene representation and instantiate distinct variables for repeated categories. Candidate assets are selected deterministically by category and footprint compatibility, including 90° rotations, be-

fore LayoutVLM predicts their spatial constraints. Its center-based poses are converted back to centimeter-scale lower-left boxes and yaw angles.

I-Design. We provide the room dimensions, furnishing request, and target style in the method's native text interface, without ground-truth furniture. Rectangular rooms retain their exact dimensions, while non-rectangular rooms use the axis-aligned bounding-box interface required by the method. Floor-supported objects from the resulting scene graph are converted from meter-scale centers and sizes to the canonical box representation.

Holodeck. We initialize Holodeck from the canonical 3D-FRONT shell rather than regenerating the floor plan. The adapter preserves the ordered room polygon and openings, converts the horizontal axes to Holodeck's coordinate convention, and keeps the geometry locked while its native object-selection and placement stages operate inside the fixed room.

DiffuScene. We run the released room-category model in its native unconditional setting; the target polygon, furnishing request, and target style are not injected because the released generator does not accept them. We retain its predicted semantic boxes for floor-supported categories and retrieve 3D-FUTURE assets by predicted category and footprint, allowing a 90° rotation. Non-floor categories are omitted, and the generated layout is evaluated in the corresponding benchmark room without semantic repair.

SceneWeaver. We lock the canonical floor shell and openings, seed the requested inventory with assets from the shared 3D-FUTURE catalog, and expose the method's native scene-editing tools for relation, layout, rotation, size, addition, and removal. The final native iteration is converted to lower-left centimeter boxes; removed or otherwise absent requested objects remain absent rather than being restored by the adapter.

ReSpace. We provide the canonical room and requested category sequence to the released `respace-sg-llm-1.5b` policy. Following its default protocol, we use eight request-order shuffles with four right-angle room rotations, generate autoregressive box layouts, and apply its native prompt-match and physical-box-loss reranking. The selected semantic boxes are grounded in the shared 3D-FUTURE catalog by category and footprint only after generation; the adapter does not move previously generated boxes or repair the chosen trajectory.

B.3 Implementation and Evaluation Details

For the bedroom and living-room subsets, we inherit the room partitions used by LayoutGPT (Feng et al., 2023) and sample within rectangular and non-rectangular buckets using a fixed seed. The training manifest contains 250 bedrooms (125 per bucket) and all 192 eligible living rooms (76 rectangular and 116 non-rectangular), excluding combined living–dining rooms. The room-disjoint test manifest contains 200 cases: 100 bedroom-family cases spanning bedrooms, master bedrooms, and secondary bedrooms; 53 living rooms; and 47 dining rooms. One of nine fixed style requests is assigned to each case.

Training and credit hyperparameters. We post-train Qwen3.5-9B for 500 updates in bfloat16 with a disaggregated eight-GPU configuration (four GPUs for training and four for rollout inference). Each update contains two case groups with four rollouts per case, for a rollout batch size of eight. We use the Megatron distributed optimizer with a learning rate of 1×10^{-6} and maximum gradient norm 0.5. The policy clip is $\varepsilon = 0.1$, and the active KL-loss coefficient is $\beta = 0.02$.

Rollouts use temperature 0.9, top-$p = 1.0$, no top-k truncation, at most 512 generated tokens per action, and a 16,384-token context. We set the local potential-difference, feasible-domain-return, and terminal-advantage weights to $\lambda_{\text{loc}} = 0.3$, $\lambda_{\text{seq}} = 0.2$, and $\lambda_{\text{final}} = 1.0$, respectively, and use $\eta_{\text{lay}} = \eta_{\text{sty}} = 1.0$ to mix the layout and attributed style token credits. The future-domain discount is $\gamma = 0.95$. The completion-aware terminal outcome s_b^{lay} combines furniture completion with the completion-weighted layout score ℓ_b^{lay} using weights 1.0 and 0.2, respectively; $\epsilon_{\text{std}} = 0.1$, and the resulting advantage is clipped to $[-1, 1]$.

For $|\mathcal{A}_{b,n}|$, we count distinct legal (o, x, y) actions after unioning overlapping regions for each of the four orientations. The normalizer $|\mathcal{A}_{b,n}^{\max}|$ counts all in-bound actions for the same selected SKU in the empty room bounding box, yielding $F_{b,n}$ in Equation 3. A missing future-domain measurement suppresses only the corresponding return term for earlier actions; valid commits in an incomplete rollout retain their local recovery and commit credit. Terminal layout credit remains completion-aware through CNT. When valid style evidence exists for a partial rollout, its style advantage is multiplied by the completion fraction; an unavailable or malformed style outcome receives no attributed asset credit.

Evaluation protocol. All 200 test cases are converted to the same canonical centimeter representation before automatic evaluation. Layout metrics are computed from the resulting boxes and openings, while GPT-5.5 evaluates the rendered scene for realism, functionality, and layout quality and evaluates the selected-asset contact sheet with the style rubric above. Human evaluation uses a fixed subset of 20 cases—10 bedroom-family, seven living-room, and three dining-room cases—and reports 0–10 ratings for style and overall scene quality in Table 2.

C Additional Experimental Results

This section provides additional qualitative examples and detailed quantitative results across room types.

C.1 Qualitative Results

Figure 9 illustrates the diversity of Dovetail3D within the Living Room category across varied room geometries and target styles.

Figure 3 compares Dovetail3D with seven representative baselines across six challenging room specifications and target styles. A common viewpoint for each scene makes differences in asset selection, furnishing completeness, style consistency, and spatial arrangement directly visible.

Figures 6–9 further present 33 scenes synthesized by Dovetail3D across varied room geometries and target styles. Each panel is labeled with its room identifier and requested style.

C.2 Detailed Quantitative Results

Tables 1 and 2 already report the complete layout, aggregate style, LLM-as-Judge, and human-evaluation results. Table 7 therefore provides only the complementary style subcriteria: style consistency (SC), color harmony (CH), material and texture compatibility (MT), shape and form compatibility (SF), and visual atmosphere (VA).

D Workflow Demonstration

Each furniture requirement specifies the category and optional size constraints of an object to be placed. The abbreviated JSONC schema below summarizes the workflow input.

```
{
  "room_info": {
    "coordinate_unit": "cm",
    // Room boundary represented by 2D line segments
    "wall": [
      { "start": [x1, y1], "end": [x2, y2] },
      ...
    ],
    // Architectural openings
    "door": [
      { "start": [x1, y1], "end": [x2, y2] },
      ...
    ],
    "window": [
```

Table 7: Style subcriteria across the five original room categories. All scores use a 0–10 scale, and higher is better. Bedroom and secondary-bedroom results are reported separately here to expose the breakdown underlying the combined Bedroom block in Table 2.

Room Type	Method	SC	CH	MT	SF	VA
Bedroom	LayoutGPT	6.0000	9.5714	6.7143	6.0000	6.0000
	LayoutVLM	5.4286	9.2857	7.3571	5.5714	5.3571
	I-Design	7.1429	7.7143	7.7857	7.1429	7.1429
	DiffuScene	4.2857	7.0714	7.5000	5.5714	4.2857
	ReSpace	7.0714	8.2857	9.0000	7.0714	6.9286
	SceneWeaver	6.7143	7.3571	8.0000	6.7143	6.7143
	Holodeck	6.9286	7.9286	8.5000	6.9286	6.7857
	Dovetail3D	8.1429	9.0000	9.5000	8.0714	8.0000
Master Bedroom	LayoutGPT	6.6154	9.4359	7.2308	6.6667	6.5641
	LayoutVLM	6.3077	9.5897	7.0256	6.3590	6.2564
	I-Design	7.3846	8.6667	8.1026	7.3333	7.3333
	DiffuScene	5.0769	7.0256	8.2564	6.3077	4.9231
	ReSpace	7.0256	8.8205	8.4103	6.9744	6.9231
	SceneWeaver	8.3077	8.7692	9.0256	8.3077	8.1026
	Holodeck	7.7436	8.5641	8.7179	7.6923	7.5897
	Dovetail3D	8.5128	9.5897	9.0769	8.5641	8.4615
Secondary Bedroom	LayoutGPT	7.2727	9.5758	8.0606	7.1515	7.1515
	LayoutVLM	7.6364	9.5152	8.6061	7.6364	7.6364
	I-Design	7.0909	8.1212	7.8182	6.9697	6.9697
	DiffuScene	4.9091	7.3333	8.3636	5.8788	4.8485
	ReSpace	8.1212	9.3333	9.0909	8.1212	8.1212
	SceneWeaver	7.8182	8.1818	8.4848	7.8182	7.8182
	Holodeck	7.3333	7.8182	8.4848	7.3333	7.2121
	Dovetail3D	8.6061	9.6364	9.4545	8.6061	8.6061
Living Room	LayoutGPT	4.0755	7.8113	5.2830	4.2264	3.9245
	LayoutVLM	5.0189	8.4528	6.1509	5.0566	5.0189
	I-Design	3.2830	4.0755	3.6226	3.2830	3.2830
	DiffuScene	5.5849	9.4340	6.0377	5.7358	5.5849
	ReSpace	5.1698	9.1321	5.8491	5.3208	5.1698
	SceneWeaver	7.6604	9.5094	8.2264	7.6226	7.5849
	Holodeck	5.3585	7.5472	6.2264	5.7358	5.2830
	Dovetail3D	8.4151	9.7358	8.7170	8.3774	8.3774
Dining Room	LayoutGPT	8.0426	9.8723	8.3830	8.0426	8.0000
	LayoutVLM	7.7447	9.4043	8.4681	7.7872	7.7447
	I-Design	7.1489	8.2128	7.5745	7.2340	7.1489
	DiffuScene	7.0638	9.8298	7.4894	7.1064	7.0638
	ReSpace	7.6170	9.4468	8.2553	7.7021	7.6170
	SceneWeaver	7.4043	8.5106	7.6170	7.4468	7.4043
	Holodeck	5.2340	6.4255	5.5319	5.2340	5.2340
	Dovetail3D	8.8636	9.9574	9.2727	8.8636	8.8636

```
      { "start": [x1, y1], "end": [x2, y2] },
      ...
    ]
  },
  // One entry for each furniture item to be placed
  "furniture_requirements": [
    {
      "id": "region_1",
      "category": "sofa",
      "required": true,
      "size_constraint": {
        "max_width": 220,
        "max_depth": 100
      }
    },
    {
      "id": "region_2",
      "category": "coffee_table",
      "required": true
    },
    ...
  ],
  // Optional natural-language preference
  "style_request": "modern, minimalist, and warm",
  // Optional furniture already present in the room
  "fixed_furniture": [...]
}
```

Listing 1: Illustrative workflow input schema. Ellipses denote omitted or repeated entries; `style_request` and `fixed_furniture` are optional.



Figure 6: Additional Dovetail3D results for bedrooms. The examples cover six room specifications and multiple target styles.



Figure 7: Additional Dovetail3D results for master bedrooms. The examples cover nine room specifications and multiple target styles.



Figure 8: Additional Dovetail3D results for secondary bedrooms. The examples cover six room specifications and multiple target styles.

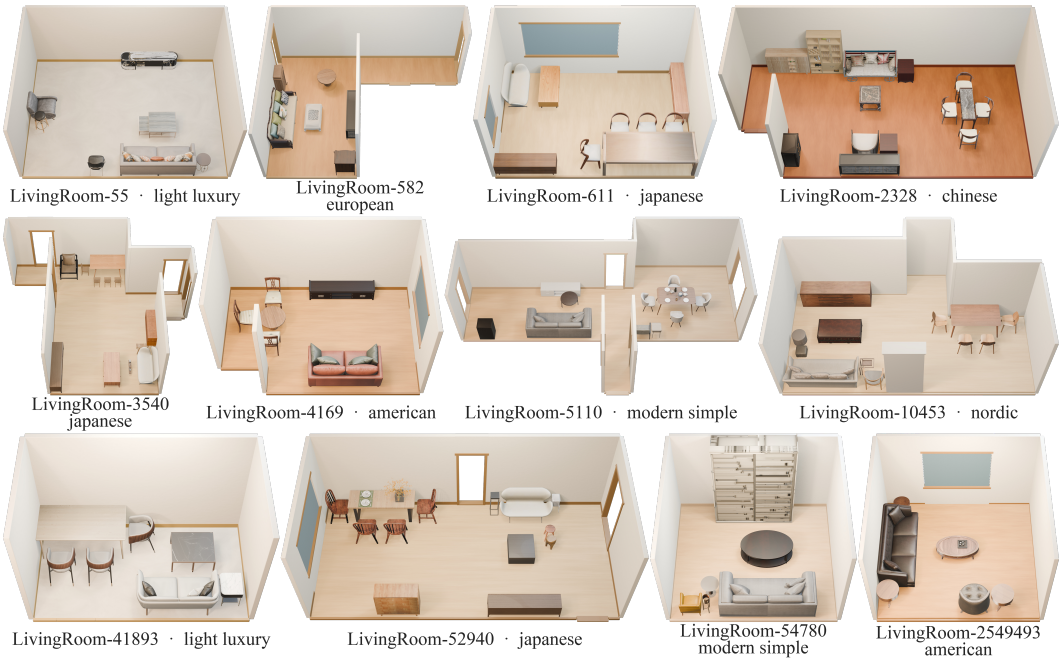

Figure 9: Additional Dovetail3D results for living rooms. The examples cover twelve room specifications and multiple target styles.