

Research Article



AI-Enhanced Home Design with Augmented Reality

Ali Hassan Kadri,*^{ORCID} Arfa Shaikh,^{ORCID} Aafiya Shaikh,^{ORCID} Isar Fatima Syed^{ORCID} and Farhana Siddiqui^{ORCID}

Department of Computer Engineering, M. H. Saboo Siddik College of Engineering, Mumbai, Maharashtra, 400008, India

*Corresponding Author

Ali Hassan Kadri

✉ hasanqadri1990@gmail.com

Received: 02 August 2026

Revised: 12 September 2026

Accepted: 19 September 2026

Published Online: 19 September 2026

Citation

A. H. Kadri, A. Shaikh, A. Shaikh, I. Fatima Syed, F. Siddiqui, AI-enhanced home design with augmented reality, *Journal of Visual Artificial Intelligence*, 2026, 2(2), 26106, doi: <https://doi.org/10.64189/vai.26106>.

Open Access

This article is licensed under a [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/), which permits the non-commercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as appropriate credit to the original author(s) and the source is given by providing a link to the Creative Commons License and changes need to be indicated if there are any. The images or other third-party material in this article are included in the article's Creative Commons License, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons License and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this License, visit: <https://creativecommons.org/licenses/by-nc/4.0/>

© The Author(s) 2026

Abstract

Most people planning a redesign choose furniture, colour and materials from catalogues, floor plans or imagination. AR interior tools typically overlay individual catalogue objects, and general-purpose image generators can restyle a room but often move its walls or perspective in the process. This study presents Home Gennie, a web-based system that restyles a photograph of the user's own room while keeping its geometry intact, reconstructs the restyled room in three dimensions from that same photograph, and displays it in AR without any app installation. Straight-line and monocular depth maps extracted from the photograph condition a ControlNet diffusion model, so furnishings, materials and lighting are regenerated while the architectural lines stay put; a four-layer cascade keeps the service available when an endpoint fails. The redesigned image is then converted into a colour-textured, full-room mesh by back-projecting a monocular depth map through a pinhole camera model and placed in AR through the browser. Tested on 15 room photographs submitted by users of the deployed system, line-conditioned generation preserved room structure substantially better than unconditioned text-only generation (structural similarity 0.720 ± 0.125 vs. 0.453 ± 0.148 ; Mann-Whitney $p = 0.0059$). Full-room reconstruction produced meshes of 262,144 vertices (about 10 MB) in a mean of 4.47 s over 30 production requests, and 65.5% of requests resolved to a permanent image. Pretrained models can be combined into an accessible preview tool, though the reconstructed scale is approximate and geometry is preserved only when the first cascade layer answers the request.

Keywords: Interior design; Augmented reality; Controllable image generation; Diffusion models; Monocular depth estimation; Three-dimensional reconstruction; Web-based visualization.

1. Introduction

Interior design decisions about furniture, colour, materials and lighting are normally made before their effect can be seen in the actual space. Conventional communication between homeowners and designers relies on two-dimensional drawings, static renderings and product catalogues, which many clients find difficult to translate into an expectation of the finished room; the resulting mismatch between expectation and outcome can lead to revisions, delays and added cost.^[1,2] Artificial intelligence (AI) and augmented reality (AR) have therefore been proposed as complementary tools for making design intent visible earlier in the process.^[3-5]

AR superimposes virtual content on a live camera view, allowing users to test furniture and layouts in their own room rather than on paper,^[2,6,7] and AR-based presentation has been reported to influence engagement and purchase decisions in housing and retail contexts.^[1,8] In parallel, AI techniques have been applied to automatic furniture layout and room configuration,^[9,10] interactive decoration,^[11] furniture design,^[12] design graphics and modelling,^[13] and smart-home planning.^[14] More recently, latent diffusion models,^[15] extended with spatial conditioning,^[16] have made it possible to generate photorealistic images whose composition is constrained by structural maps such as edges, line segments or depth, which opens the possibility of restyling a photograph of an existing room rather than composing a new scene.

Despite this progress, homeowners who want to preview a redesign of the room they already have still run into three practical gaps. Most AR interior tools overlay individual pre-modelled catalogue objects on the camera view,^[2,6,17] rather than restyling the whole existing room; unconstrained text-to-image generators,^[18] can produce attractive interiors, but nothing stops them from moving a wall or a window, so the output is a picture of a different room rather than a preview of the user's own. And getting a 3D model of the complete room for AR inspection usually means depth sensors, multi-view capture or manual modelling, plus a native app to install. Evaluations in this space also tend to lean on qualitative feedback rather than defined metrics and baselines.

To address these gaps, this paper presents Home Gennie, a web-based system that turns a single photograph of an existing room into a restyled design image, a textured 3D model of the complete room and an AR preview, without specialised sensors or app installation. The methodology combines pretrained models rather than training a new network: straight-line and monocular depth maps are extracted from the photograph as structural features and used to condition a ControlNet-guided diffusion model; the redesigned image is converted into a full-room mesh by pinhole back-projection of a monocular depth estimate; and the mesh is delivered to the browser's AR runtime. The specific

contributions are:

- (1) A geometry-preserving redesign pipeline in which M-LSD line maps^[19] and Depth-Anything-V2 depth maps^[20] serve as structural conditions for ControlNet,^[16] embedded in a four-layer provider cascade that maintains availability on free-tier inference infrastructure (Algorithm 1).
- (2) A single-image, full-room reconstruction procedure that converts a depth map into a colour-textured, artefact-filtered GLB mesh (Algorithm 2), with closed-form expressions for the mesh size and storage footprint (Eqs. (3) – (9)).
- (3) An installation-free delivery path from photograph to AR through WebXR, Android Scene Viewer and iOS Quick Look, backed by row-level data isolation between users.
- (4) An evaluation on real user requests comprising structure-preservation, style-adherence (with a per-class classification report), latency, reliability and mesh metrics, including a comparison against unconditioned text-only generation.

The contribution is mainly in system design and integration. Every model in the pipeline is an existing pretrained component; what is new is how the models are combined, constrained and delivered for the specific job of previewing a redesign of an existing room. Section 2 reviews related work; Section 3 covers the system overview, working principle, software, implementation, model configuration and testing protocol; Section 4 reports the results, a competitive analysis, a comparison of methods, an interpretation and the limitations; and Section 5 concludes.

2. Literature review

2.1 Augmented reality in interior design

Revathy et al. reviewed the use of AR in interior design and its capacity to let users try furniture arrangements in real environments.^[6] Samant and Vartak^[2] and Juned *et al.*^[7] developed mobile AR applications in which users place virtual furniture models in their room, the latter using ARCore with a cloud-hosted model catalogue, and Huang and Ni presented ARID, an AR interior design system.^[17] Dave *et al.* examined how AR, virtual reality (VR) and AI influence consumer decisions in housing design.^[1] Outside the home-design domain, Zimmermann et al. showed in an online experiment that an AR shopping assistant with personalised, explainable recommendations improved perceived in-store shopping experience,^[8] and Kim et al. studied how uncertainty in AI-based object detection should be exposed to users in AR interaction design.^[21] These systems demonstrate the value of in-situ visualisation, but they generally operate on individual catalogue assets rather than on a restyled rendering of the entire existing room.

2.2 AI for interior design and layout

Cuevas et al. presented LayOut Loud, an AI-powered AR and mobile application for room interior design and layout optimisation.^[9] K n *et al.* proposed an algorithm based on a hierarchical tree of procedural rules that generates personalised furniture configurations and presents them in mobile AR and evaluated it in three user studies.^[10] Wu and Han evaluated the combination of AI and VR for interactive interior decoration;^[11] Gong^[5] applied AI-based visual feature modelling to interior design schemes; Jiang^[12] built a generative adversarial network for personalised furniture design within an Internet of Things environment; Samuel *et al.*^[13] discussed computational technology for design graphics and modelling; and Almusaed *et al.* reviewed AI models for smart-home living spaces.^[14] Broader reviews cover AI, VR and AR in product design^[3] and the convergence of AI and AR.^[4] The common output of these approaches is an arrangement, selection or design of discrete furniture objects; photorealistic restyling of the user's own photograph under explicit geometric constraints is not their focus.

2.3 Controllable image generation

Latent diffusion models perform the denoising process in the latent space of an autoencoder, which makes high-resolution text-to-image synthesis tractable;^[15] SDXL scales this approach to higher resolutions.^[18] InstructPix2Pix edits an input image according to a natural-language instruction^[22] but does not explicitly enforce structural constraints. ControlNet attaches a trainable copy of the diffusion encoder to a frozen pretrained model and injects features computed from a control image, such as edges, line segments or depth, into the denoising network.^[16] M-LSD is a lightweight line-segment detector designed for real-time use on constrained devices;^[19] because interior architecture is dominated by straight lines, its output is a natural control signal for preserving walls and openings

2.4 Monocular depth estimation and single-image 3D

Dense prediction transformers (DPT)^[23] and the Depth Anything family^[20] estimate relative depth from a single image and generalise across indoor and outdoor scenes. Image-to-3D generators such as TRELLIS^[24] and InstantMesh^[25] produce high-quality meshes of individual objects, but they typically isolate a foreground object and are therefore not suited to reconstructing an entire room with its walls, floor and ceiling.

2.5 Research gap

No existing category of approach simultaneously (i) restyles the user's own room photograph, (ii) preserves its architectural geometry by construction, (iii) produces a 3D model of the complete room and (iv) delivers that

model to AR without a native application; the categories are compared in Section 4.6. Home Gennie is designed to fill this combination of requirements. It differs from LayOut Loud,^[9] ARID^[17] and the procedural-rule approach of K n *et al.*^[10] in that it does not place catalogue objects but regenerates the appearance of the entire room under line and depth constraints, and it differs from generic generators,^[18,22] in that geometry preservation is enforced through conditioning rather than requested only in the prompt.

3. Materials and methods

Miscommunication between homeowners and designers leads to delays and costly redesigns. The proposed system addresses this by combining AI-based design generation with 3D and AR visualisation in a single workflow: the user's photograph and style choice are processed by a geometry-preserving generator, the result is reconstructed in 3D, and the model is shown in the user's physical environment before any physical change is made. The following subsections describe the system overview, working principle, software, implementation, model configuration and testing protocol.

3.1 System Overview

Home Gennie follows a decoupled four-tier architecture (Fig. 1). The client tier is a React single-page application that handles authentication, photo upload, design browsing, 3D viewing and AR launch. The application tier is a FastAPI service that orchestrates AI inference and exposes four endpoints: POST /generate (asynchronous 2D redesign), POST /generate-3d-room (full-room reconstruction), POST /generate-3d (single-object reconstruction) and GET /health. The data tier is Supabase, providing authentication, a PostgreSQL database with row-level security (RLS) and object storage. The AI model tier comprises pretrained models executed either locally on the backend processor (M-LSD, fallback depth models, mesh construction) or through hosted inference services. Design records persist across sessions, and all data access is restricted to the owning user.

3.2 Working principle

The end-to-end flow of the system is shown in Fig. 2. After signing in, the user uploads a room photograph, which the client stores directly in the uploads bucket under the user's identifier, and selects a room type and interior style from the options in Table 1, together with an optional colour palette. The client then calls POST /generate. The backend inserts a design record with status 'pending', returns its identifier immediately and runs the generation cascade (Section 3.4.3) as a background task, so that the browser never waits on a long-running request. The Gallery page polls the

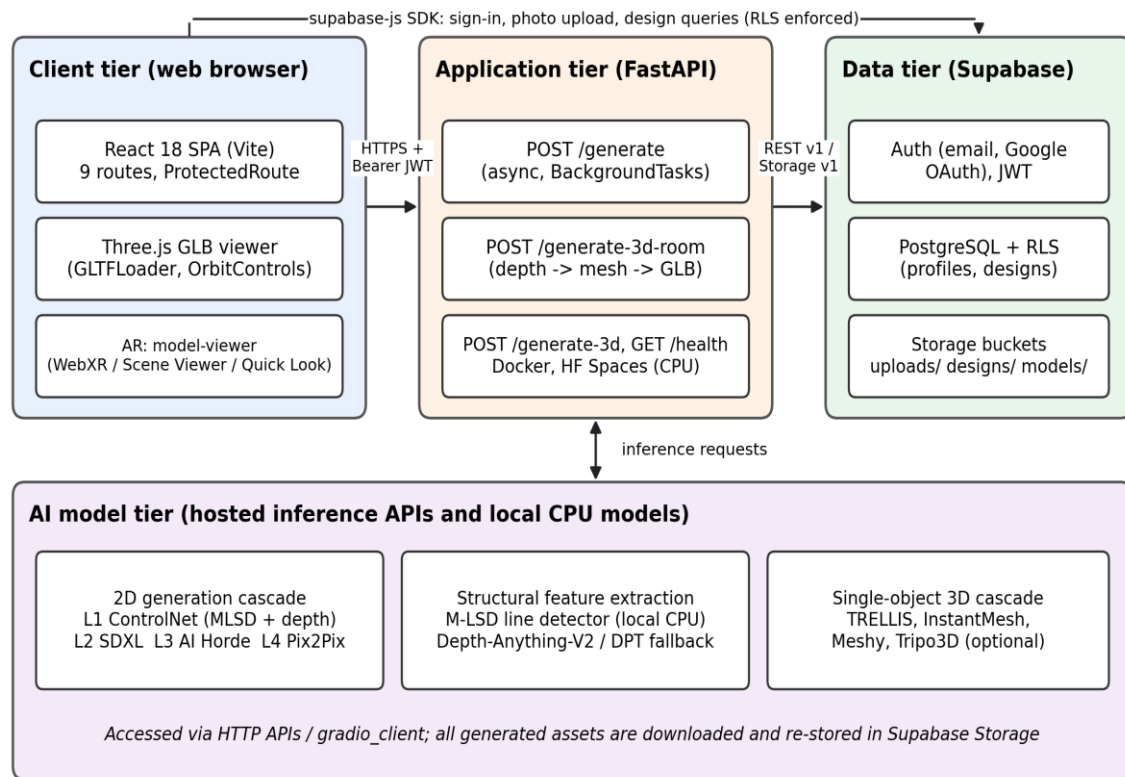


Fig. 1: System architecture of Home Gennie, showing the client, application, data and AI model tiers and their communication paths.

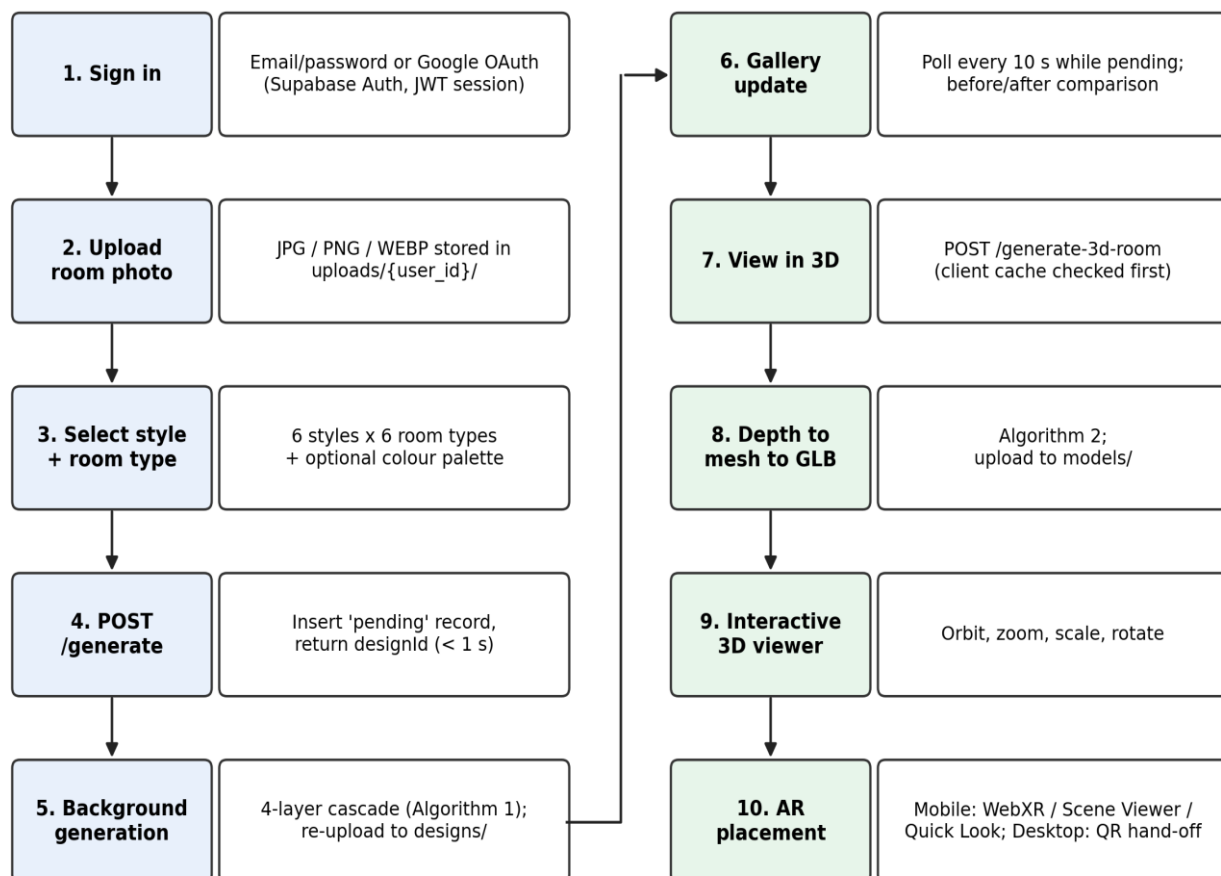


Fig. 2: Overall workflow of the system, from sign-in and photo upload to 3D viewing and AR placement.

Table 1: Design options offered by the interface and the style vocabulary used for evaluation

Dimension	Categories	Count
Interior style (interface)	Modern, Minimalist, Classic, Industrial, Bohemian, Scandinavian	6
Room type (interface)	Living Room, Bedroom, Kitchen, Dining, Bathroom, Office	6
Style vocabulary for CLIP evaluation (Section 3.6)	Modern, Minimalist, Scandinavian, Industrial, Bohemian, Japandi, Mid-Century Modern, Farmhouse, Art Deco, Coastal, Traditional, Contemporary	12

database every 10 s while any design is pending and presents completed results with a before/after comparison slider. From any completed design the user can request a 3D model; POST /generate-3d-room reconstructs the room (Section 3.4.4), and the resulting GLB model is shown in an interactive viewer and can be launched in AR (Section 3.4.5).

The evaluation vocabulary contains five of the six interface styles and seven additional common interior styles, so that a generated image can be assigned to a related but different style when it drifts from the request. 'Classic' has no label in this vocabulary; its nearest equivalent is 'Traditional' (Section 4.5).

3.3 Software

The technologies used in each architectural tier are listed in Table 2. Development and the local processor measurements reported in Section 4 were carried out on a laptop with an Intel Core i5-1155G7 processor (4 cores, 8 threads, 2.50 GHz), 8 GB DDR4 memory and integrated Intel Iris Xe graphics, running Windows 11 Home. The production backend runs in a CPU-only Docker container (python:3.11-slim) on the Hugging Face Spaces free tier, and the frontend is served by Vercel. Hosted models run on the providers' infrastructure, whose hardware is not under the authors' control; timings that involve these services therefore include network and queueing delays.

3.4 Implementation

3.4.1 Image acquisition

Users upload a single photograph in JPG, PNG or WEBP format through a drag-and-drop component; the file is stored unmodified at uploads/{user_id}/{timestamp}.{ext}, and its URL is passed to the backend. Server-side, each pipeline downloads the image and resizes it to 512 × 512 pixels, the native resolution of the Stable Diffusion 1.5 backbone used by the ControlNet model. No further normalisation is applied by the system itself; each pretrained model applies its own input normalisation internally.

3.4.2 Feature extraction

Because the objective is to change the appearance of a room while retaining its structure, the system extracts two structural representations and one semantic

representation from each photograph. All extractors are pretrained and used without modification.

- *Line-segment map.* M-LSD^[19] (weights from llyasviel/Annotators, executed on the backend processor through controlnet_aux) detects straight line segments and renders them as a binary image. In interior photographs these segments correspond predominantly to wall junctions, window and door frames, skirting and ceiling lines, so the map acts as an architectural blueprint of the room.
- *Relative depth map.* Depth-Anything-V2^[20] is requested through the Hugging Face Inference API; if the request fails, the DPT-Hybrid MiDaS model^[23] is run locally. The resulting single-channel map encodes the relative distance of surfaces from the camera and is used as the control image only when line extraction fails.
- *Semantic prompt features.* The selected style s and room type r are inserted into a fixed prompt template (Section 3.4.3), which the diffusion model's text encoder converts into conditioning embeddings.

Within ControlNet, the chosen control image is encoded by a trainable copy of the diffusion U-Net encoder, and its multi-scale feature maps are added to the corresponding layers of the frozen denoising network through zero-initialised convolutions.^[16] The structural features therefore constrain where edges and surfaces appear in the generated image, while the prompt determines their appearance. One control image is supplied per request, selected in the priority order line map, depth map, original photograph (Fig. 3).

3.4.3 Design generation and recommendation

The first-generation layer (Fig. 3) calls the public ControlNet v1.1 Gradio Space with the selected control image and a prompt that explicitly restates the geometric constraints: "Redesign this exact room as a {style} style {room_type}. IMPORTANT: Keep the exact same room layout, wall positions, windows, doors, ceiling height, and floor geometry. Only change the furniture, decor, colors, materials and lighting. Same perspective and camera angle. Photorealistic, interior design magazine quality, 8k, highly detailed." The negative prompt is "deformed walls, wrong perspective, different room layout, low quality, blurry, distorted, extra doors, extra windows". If

Layer 1 of the generation cascade. If Layer 1 returns no image, Layers 2-4 (SDXL, AI Horde, InstructPix2Pix) are tried in order.

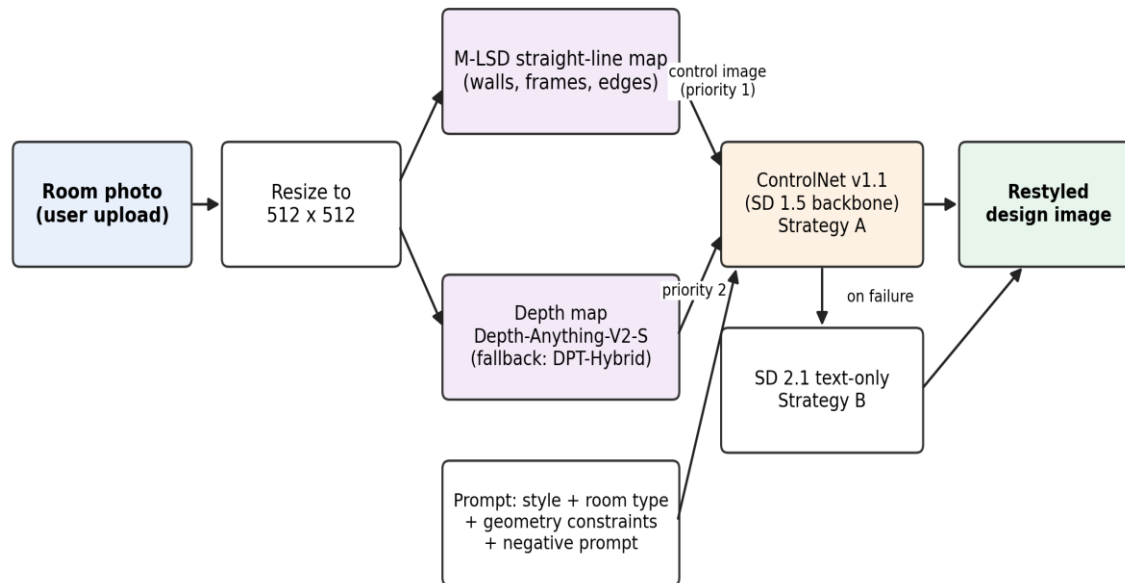


Fig. 3: First layer of the generation cascade: structural feature extraction and ControlNet-conditioned generation with a text-only fallback.

Table 2: Technology stack by architectural tier

Tier	Technologies (versions)	Function
Client (presentation)	React 18.3.1, Vite 8.0, React Router 7.13 (9 routes), Tailwind CSS 4.2, Framer Motion 12.38, Three.js 0.183, Google model-viewer, qrcode 1.5	User interface, routing, 3D viewing, AR launch, QR hand-off
Application	Python 3.11, FastAPI, Uvicorn, Pydantic v2, Pillow ≥ 10, requests, gradio_client ≥ 2.3, Replicate SDK	REST endpoints, asynchronous orchestration, provider cascades
Local models and geometry	controlnet_aux ≥ 0.0.7 (M-LSD), transformers ≥ 4.40 with CPU-only PyTorch (Depth-Anything-V2-Small, DPT-Hybrid), NumPy ≥ 1.24, trimesh ≥ 4.0	Line and depth extraction, mesh construction, GLB export
Hosted inference	ControlNet v1.1 Gradio Space; Hugging Face Inference API (Depth-Anything-V2-Large, Stable Diffusion 2.1, InstructPix2Pix); Replicate (SDXL, TRELIS); AI Horde; TRELIS and InstantMesh Spaces; Meshy and Tripo3D (optional)	2D generation, depth estimation, single-object 3D
Data and security	Supabase Auth (email/password, Google OAuth), PostgreSQL with RLS, Supabase Storage (uploads/, designs/, models/)	Identity, persistence, per-user isolation
Deployment	Vercel (frontend), Hugging Face Spaces with Docker (CPU-only, port 7860), GitHub	Hosting and continuous deployment

the Space is unavailable (Strategy A fails), Stable Diffusion 2.1 is called with the same prompt but no control image (Strategy B).

Hosted inference on free-tier infrastructure is subject to cold starts, queueing and intermittent failures. Generation is therefore wrapped in a four-layer cascade (Algorithm 1): ControlNet (Layer 1), SDXL^[18] via Replicate (Layer 2), the AI Horde community GPU network (Layer 3) and InstructPix2Pix^[22] via the Hugging Face Inference API (Layer 4). Only Layer 1 enforces geometric

conditioning; the remaining layers trade geometric fidelity for availability, and every output is stored with a label identifying the layer that produced it. Because Replicate and AI Horde return temporary URLs (valid for roughly seven days and 30 minutes, respectively), each result is intended to be downloaded immediately and re-uploaded to permanent storage before the database record is updated; Section 4.5 reports a case where this step was not applied.

Algorithm 1: Fault-tolerant geometry-preserving generation (background task of POST /generate)

Input : photo URL x , style s , room type r , design id k , user token j
Output: permanent URL of the design image, or status 'failed'

- 1 $p \leftarrow \text{Prompt}(s, r)$; $n \leftarrow \text{NegativePrompt}$
- 2 for layer ℓ in [ControlNet, SDXL, AIHorde, InstructPix2Pix] do
- 3 if credentials for ℓ are unavailable then continue
- 4 if $\ell = \text{ControlNet}$ then
- 5 $I \leftarrow \text{Resize}(\text{Download}(x), 512 \times 512)$
- 6 $L \leftarrow \text{MLSD}(I)$; $D \leftarrow \text{Depth}(I)$ // either may be null
- 7 $C \leftarrow \text{first non-null of } (L, D, I)$
- 8 $B \leftarrow \text{ControlNetSpace}(C, p, n)$ or else $\text{SD21TextOnly}(p, n)$
- 9 else $B \leftarrow \ell.\text{Generate}(x, p, n)$
- 10 if B is a temporary URL then $B \leftarrow \text{DownloadBytes}(B)$
- 11 if $B \neq \text{null}$ then
- 12 $u \leftarrow \text{Upload}(B, \text{designs}/\{\text{user}\}/\{\text{time}\}_{\ell.\text{label}}.\text{jpg}, j)$
- 13 $\text{UpdateRecord}(k, u)$; return u
- 14 $\text{UpdateRecord}(k, \text{'failed'})$; return 'failed'

If layer i succeeds with probability p_i and the layers fail independently, the probability that a request yields an image and the expected processing time are

$$P_{\text{succ}} = 1 - \prod_{i=1}^L (1 - p_i) \quad (1)$$

$$E[T] = \sum_{i=1}^L [\prod_{j=1}^{i-1} (1 - p_j)] \bar{t}_i \quad (2)$$

where $L = 4$ and $\bar{t}_i$ is the mean duration of an attempt at layer i . Eq. (1) shows that the cascade raises availability whenever any single layer is unreliable, while Eq. (2) shows the latency cost of reaching lower layers. The independence assumption is optimistic when providers share failure causes, such as loss of network connectivity at the backend.

3.4.4 Three-dimensional room reconstruction

Image-to-3D object generators isolate a single foreground object and discard the walls, floor and ceiling.^[24,25] For whole-room reconstruction, POST /generate-3d-room therefore converts a monocular depth map of the design image directly into a mesh (Fig. 4, Algorithm 2). The image is resized to $W \times H = 512 \times 512$ and passed through a three-layer depth cascade: Depth-Anything-V2-Large through the Hugging Face Inference API, the Depth-Anything-V2 Gradio Space, and Depth-Anything-V2-Small (94 MB) on the local processor.^[20] The raw depth prediction d is normalised and mapped to an approximate metric range, with larger normalised values corresponding to closer surfaces:

$$\hat{d}(u, v) = \frac{d(u, v) - \min d}{\max d - \min d} \quad (3)$$

$$Z(u, v) = Z_{\max} - \hat{d}(u, v)(Z_{\max} - Z_{\min}) \quad (4)$$

where $Z_{\min} = 0.1$ m and $Z_{\max} = 8.0$ m. Each pixel (u, v) is back-projected with a pinhole camera model whose intrinsics are approximated from the image size, because the true focal length of the user's camera is unknown:

$$X(u, v) = \frac{(u - c_x)Z(u, v)}{f_x}, \quad Y(u, v) = \frac{(v - c_y)Z(u, v)}{f_y} \quad (5)$$

$$f_x = f_y = \kappa W, \quad c_x = \frac{W}{2}, \quad c_y = \frac{H}{2} \quad (6)$$

with $\kappa = 0.8$. Neighbouring pixels are connected into a regular grid with two triangles per 2×2 pixel quad, giving

$$N_v = WH, \quad N_f \leq 2(W - 1)(H - 1) \quad (7)$$

At depth discontinuities, such as the silhouette of furniture in front of a wall, grid triangles become long and thin and would appear as 'curtains' joining unrelated surfaces. A triangle t is therefore retained only if

$$\max_{e \in t} \|e\| \leq \tau \cdot \text{median}_{e \in E} \|e\| \quad (8)$$

where E is the set of all edges in the grid and $\tau = 4.0$. Each vertex is assigned the RGBA colour of its source pixel, and the mesh is exported in the binary glTF (GLB) format with trimesh. Assuming 32-bit floating-point positions (12 bytes per vertex), 8-bit RGBA colours (4 bytes per vertex) and 32-bit vertex indices (12 bytes per triangle), the uncompressed geometry size in bytes is

$$S_{\text{GLB}} \approx 16 N_v + 12 N_f \quad (9)$$

The processing constants are summarised in Table 3. The GLB file is uploaded to `models/{user_id}/3d_models/`; if the upload fails, it is served from a static route of the backend instead. The separate POST /generate-3d endpoint, which chains TRELLIS,^[24] InstantMesh,^[25] Meshy and Tripo3D, is retained for reconstructing a single dominant object but is not used for rooms.

3.4.5 Three-dimensional visualization, AR delivery and security

The Viewer page first checks a client-side cache keyed by the image URL, so that a previously reconstructed room

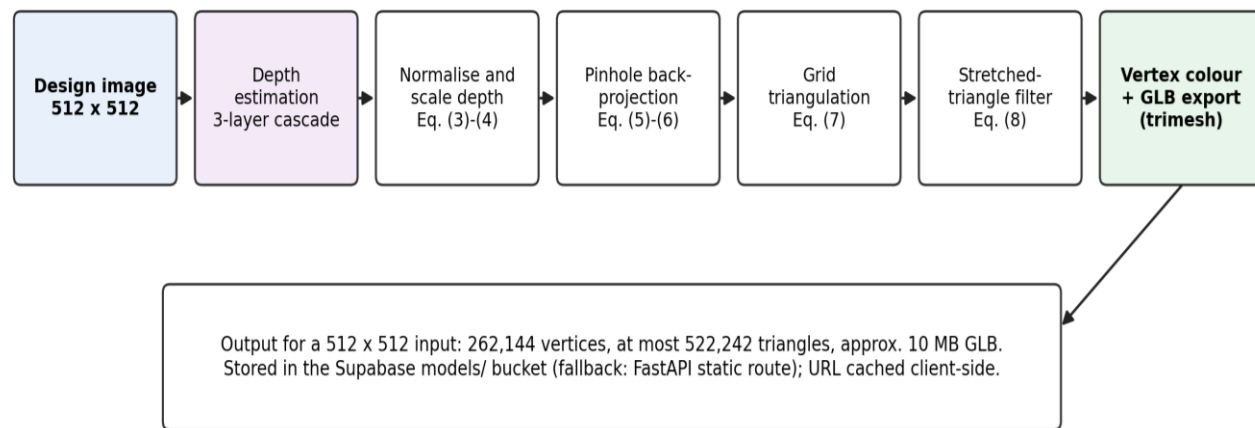


Fig. 4: Full-room reconstruction pipeline: depth estimation, scaling, back-projection, triangulation, artefact filtering and GLB export.

Table 3: Parameters of the reconstruction pipeline

Symbol	Implementation constant	Value	Role
W, H	TARGET_SIZE	512 px	Processing resolution
$Z_{\min}, Z_{\max}$	ROOM_MAX_DEPTH	0.1 m, 8.0 m	Approximate metric depth range (Eq. (4))
κ	FOCAL_LENGTH_FACTOR	0.8	Focal length as a fraction of image width (Eq. (6))
τ	EDGE_STRETCH_MULTIPLIER	4.0	Stretched-triangle rejection threshold (Eq. (8))

Algorithm 2: Single-image full-room reconstruction (POST /generate-3d-room)

Input : design image URL y ; $W = H = 512$; $Z_{\min}, Z_{\max}, \kappa, \tau$ (Table 3)

Output: permanent URL of a GLB mesh

```

1  $I \leftarrow \text{Resize}(\text{Download}(y), W \times H)$ 
2  $d \leftarrow \text{first success of [API DA-V2-Large, Gradio DA-V2, local DA-V2-Small]}(I)$ 
3  $\hat{d} \leftarrow \text{Normalise}(d)$  // Eq. (3)
4  $Z \leftarrow Z_{\max} - \hat{d} \cdot (Z_{\max} - Z_{\min})$  // Eq. (4), vectorised
5  $P[u, v] \leftarrow ((u - c_x)Z/f_x, (v - c_y)Z/f_y, Z)$  // Eqs. (5)–(6)
6  $F \leftarrow \text{two triangles per pixel quad}$  // Eq. (7)
7  $F \leftarrow \{t \in F : \max \text{edge}(t) \leq \tau \cdot \text{median edge}\}$  // Eq. (8)
8  $c[u, v] \leftarrow \text{RGBA}(I[u, v]); M \leftarrow \text{Mesh}(P, F, c)$ 
9  $g \leftarrow \text{ExportGLB}(M)$ 
10 try  $u \leftarrow \text{Upload}(g, \text{models}/\{\text{user}\}/3d\_models/)$  catch  $u \leftarrow \text{StaticRoute}(g)$ 
11 return  $u$ 

```

loads without recomputation; a regenerate control clears the entry. The GLB is rendered with Three.js with user-adjustable scale and rotation. AR is provided by Google's model-viewer web component with the modes webxr, scene-viewer and quick-look: on Android the model is placed through the WebXR hit-test API (ARCore) or the Scene Viewer application, and on iOS through AR Quick Look (ARKit). On desktop browsers, which lack AR support, the page displays a QR code that opens the same model on a phone.

The database contains a profiles table, populated by a trigger on sign-up, and a designs table storing the original and generated image URLs, style, room type, prompt, status and optional GLB URL of each design. RLS policies restrict every read and write on designs to rows where

the authenticated user identifier equals `user_id`. The frontend attaches the user's JSON Web Token (JWT) to every backend request, and the backend forwards it on its database and storage calls, so that RLS remains the authority on data access even for server-side operations. Cross-origin requests are restricted to an explicit allowlist of frontend origins.

3.5 Model configuration

The system does not train or fine-tune any model, and therefore involves no training dataset, data split, augmentation or optimiser. All generators and feature extractors are used with their published pretrained weights; the configurable elements are the prompt template and negative prompt (Section 3.4.3), the order

of the generation and depth cascades, the processing resolution and the reconstruction constants (Table 3). This keeps the system reproducible from public model identifiers and allows it to run on free-tier, CPU-only hosting, at the cost of no domain-specific adaptation to particular room types or regional styles (Section 4.8).

3.6 Testing of the system

Benchmark configuration: The evaluation was retrospective and used requests recorded by the deployed system between March and August 2026. The structure-preservation and style-adherence evaluation used 15 distinct room photographs uploaded by end users, of which 8 were processed by the default line-conditioned ControlNet path (Layer 1) and 7 by the unconditioned text-only AI Horde path (Layer 3). The reliability evaluation covers 31 design requests in the production database, of which 2 early records predate the four-layer cascade and are reported separately (Section 4.5). The reconstruction evaluation covers 7 independently generated room meshes, and the latency evaluation covers 30 consecutive requests to the deployed POST /generate-3d-room endpoint on a single date. Room type, lighting conditions and phone models were not logged systematically.

Baseline: Outputs of the text-only AI Horde layer, which receives the same prompt template but no control image, serve as an unconditioned baseline for the default line-conditioned configuration. Because the evaluation was retrospective, each photograph was processed by only one of the two paths, so the two groups are independent samples rather than paired comparisons. A depth-only ControlNet configuration was not evaluated separately.

Structure preservation: Let $\Lambda(\cdot)$ denote the M-LSD line map of an image. Structure preservation is measured as the structural similarity index (SSIM)^[26] between the line maps of the input and generated images, after light Gaussian smoothing to tolerate displacements of one or two pixels:

$$S_{struct} = SSIM(\Lambda(I_{in}), \Lambda(I_{out})) \quad (10)$$

Style adherence and classification report: Because the requested style is known, each generated image x can be classified into one of the 12 styles of the evaluation vocabulary (Table 1) by zero-shot CLIP^[27] and compared with the request, where φ_I and φ_T are the CLIP image and text encoders and π_k is the prompt 'a photo of a {style_k} style interior':

$$\hat{s}(x) = \arg \max_{k \in \{1, \dots, 12\}} \cos(\varphi_I(x), \varphi_T(\pi_k)) \quad (11)$$

$$P_k = \frac{TP_k}{TP_k + FP_k}, \quad R_k = \frac{TP_k}{TP_k + FN_k}, \quad F1_k = \frac{2P_k R_k}{P_k + R_k} \quad (12)$$

Per-style precision, recall and F1-score, their macro

averages over the 12-label vocabulary, overall accuracy and the confusion matrix are reported. This measures whether the generated image is recognisable as the requested style; it evaluates the generator, since the system itself does not classify styles.

Reliability, latency and reconstruction: For each cascade layer, the proportion of requests whose final output it produced is reported with a Wilson 95% confidence interval.^[28] Latency is reported as the mean and standard deviation over n runs with a t-based 95% confidence interval,

$$\bar{x} \pm t_{0.975, n-1} \cdot \frac{s}{\sqrt{n}} \quad (13)$$

and, for reconstruction, the proportion of triangles removed by Eq. (8) and the GLB file size are reported.

Statistical analysis: Because the baseline and default configurations were evaluated on disjoint sets of photographs, differences in S_{struct} are tested with the two-sided Mann-Whitney U test for independent samples ($\alpha = 0.05$), with rank-biserial effect size $r_{rb} = 1 - 2U/(n_1 n_2)$; confidence intervals for style accuracy are obtained by bootstrap resampling (2,000 replicates). No user study involving human participants was conducted; all metrics are computed programmatically from generated images, GLB files and system logs, using the evaluation script provided as supporting information.

4. Results and analysis

4.1 Functional verification

The complete system is deployed and publicly accessible at <https://home-gennie.vercel.app>, with the backend hosted on Hugging Face Spaces. All nine client routes, the authentication flows (email and password, Google OAuth and password reset) and the four backend endpoints were exercised end to end by the developers. Because generation runs as a background task, POST /generate returned a design identifier in under one second, independently of the subsequent generation time. The user interface is shown in Fig. 5. No automated test suite exists at present, so functional verification was manual.

4.2 Reconstruction output characteristics

For the 512×512 processing resolution, Eq. (7) gives $N_v = 262,144$ vertices and at most $N_f = 522,242$ triangles before filtering. With these values, Eq. (9) gives $S_{GLB} \approx 16 \times 262,144 + 12 \times 522,242 = 10,461,208$ bytes (≈ 10.5 MB), consistent with the file sizes of approximately 10 MB observed. Mesh size is therefore fixed by the processing resolution rather than by scene content, which makes transfer and rendering cost predictable; doubling the resolution would roughly quadruple both. Across the 7 reconstructed rooms, the discontinuity filter of Eq. (8) removed $2.61\% \pm 1.97\%$ of triangles (median 2.54%, 95% CI [0.79%, 4.43%], range 0.12–6.46%), so the filter suppresses stretched triangles at object boundaries while leaving more than 93% of the surface intact in every case.

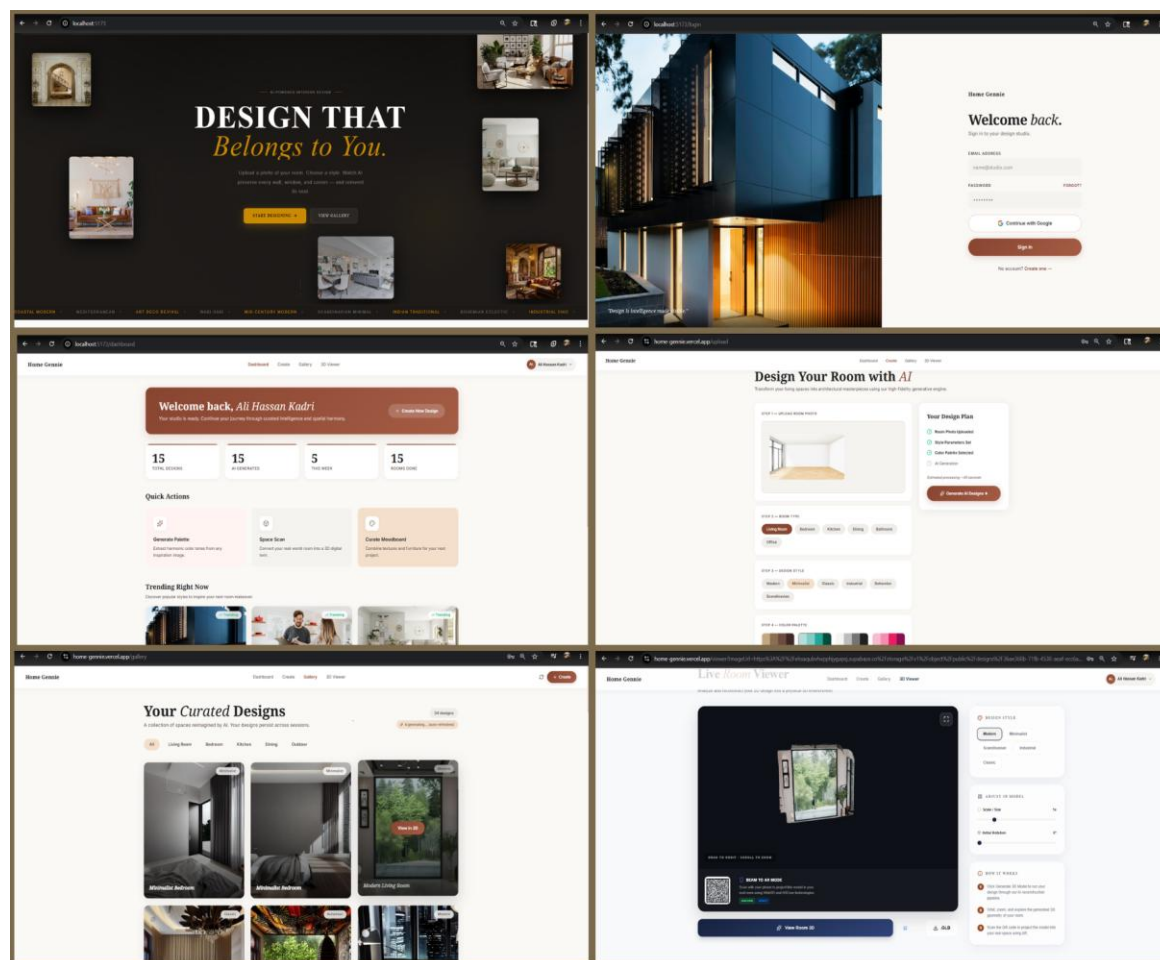


Fig. 5: User interface of the deployed system: (a) landing page, (b) sign-in with email or Google, (c) dashboard, (d) design creation with room type, style and colour palette, (e) gallery of generated designs, (f) 3D viewer with AR hand-off by QR code.

4.3 Processing latency

Table 4 lists the per-stage processing times. The 'observed range' column contains informal measurements made during development on the machine described in Section 3.3; the controlled measurement covers the full-room reconstruction endpoint, which was benchmarked over 30 consecutive production requests.

Full-room reconstruction completed in a mean of 4.47 s end to end in production, so the 3D and AR stages add only seconds to the workflow and need no GPU. 2D generation dominates the end-to-end time and varies by an order of magnitude, mostly because of queueing and cold starts on shared hosted GPUs rather than the computation itself. The asynchronous design keeps the interface responsive during this wait, although polling adds a display delay of up to 10 s.

4.4 Qualitative results

Fig. 6 shows four rooms processed by the pipeline. In the first three rows, the line-conditioned output retains the input's wall junctions, openings and camera viewpoint while changing surfaces and furnishings, whereas the

text-only outputs in column (d) present rooms with different layouts, window positions and viewpoints. Line conditioning constrains edges rather than content: in the third row the blank rear wall is regenerated as a full-height window, which alters the room's appearance without violating its line structure. The fourth row is a failure case. The cluttered office produces a dense line map dominated by furniture, cables and glass partitions rather than architectural edges, and the generated image reinterprets the partitions as a high-rise window view ($S_{\text{struct}} = 0.466$, the lowest among the eight line-conditioned outputs). The reconstructed meshes in column (f) show the single-view character of the method: surfaces facing the camera are recovered, but hidden surfaces are absent.

4.5 Competitive analysis

Table 5 compares the text-only baseline with the default line-conditioned configuration. Line-map conditioning gives a substantially higher S_{struct} than the baseline (0.720 vs. 0.453, a 59% relative increase), a difference that is statistically significant despite the modest sample (Mann-Whitney $U = 5.0$, $p = 0.0059$, $r_{\text{fb}} = 0.82$). Style

Table 4: Processing time per stage (seconds)

Stage	Executed on	Observed range (development)	Measured mean $\pm$ SD [95% CI], n
POST /generate response (record created)	Backend	< 1	—
M-LSD line extraction	Local processor	1–3	—
Depth estimation, Layer 1 (DA-V2-Large)	HF Inference API	2–5	—
Depth estimation, Layer 2 (DA-V2)	HF Gradio Space (queued)	10–30	—
Depth estimation, Layer 3 (DA-V2-Small)	Local processor (i5-1155G7)	3–8	—
Full-room reconstruction (depth to GLB)	Local processor (i5-1155G7)	5–12	—
POST /generate-3d-room, end to end	Production backend	5–15	4.47 $\pm$ 0.78 [4.18, 4.76], n = 30
2D redesign, end to end (cascade)	Hosted GPU services	30–300	—
Single-object 3D (POST /generate-3d)	Hosted GPU services	60–180	—
Gallery refresh delay after completion	Client polling	≤ 10 (by design)	—

— : not instrumented separately in this evaluation

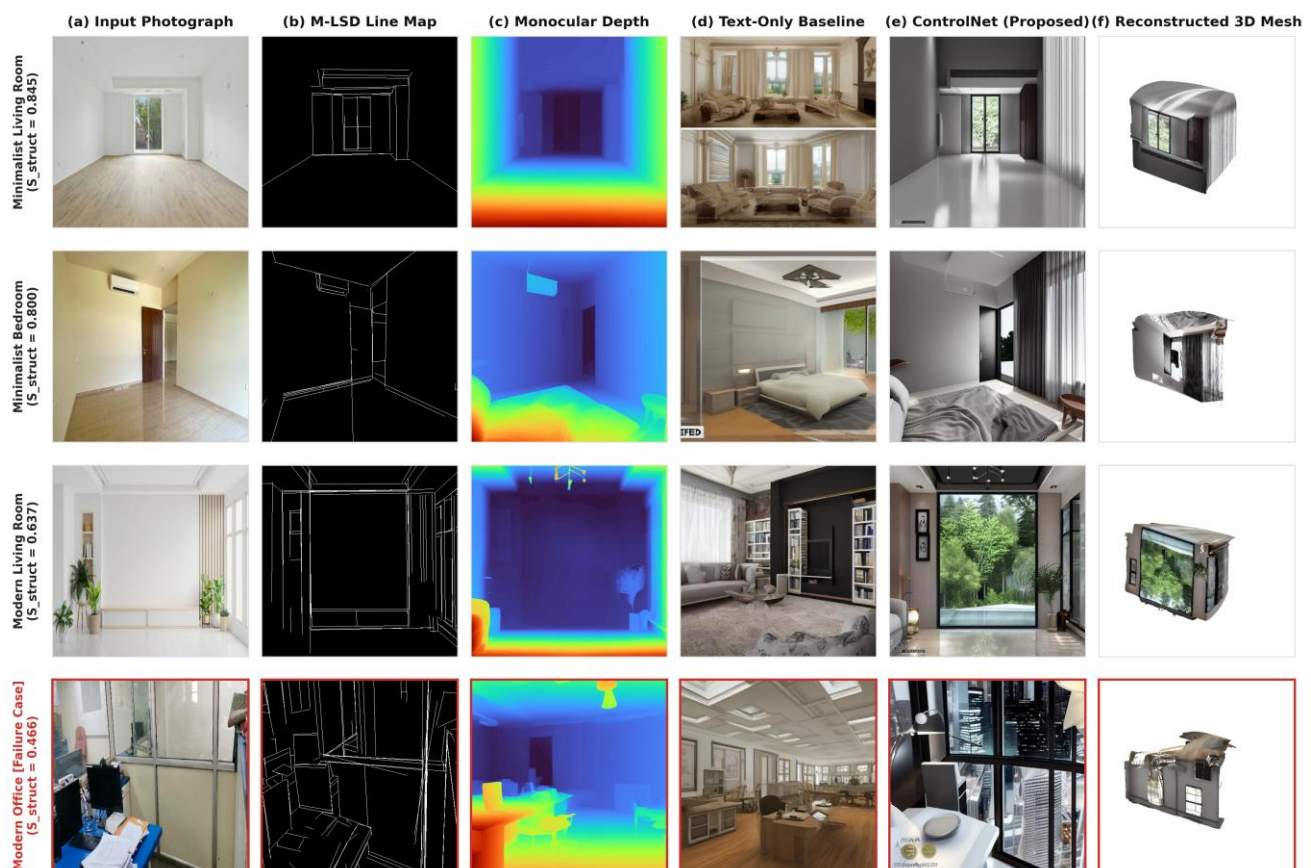


Fig. 6: Qualitative examples: (a) input photograph, (b) M-LSD line map, (c) monocular depth map, (d) text-only output for the same photograph (illustrative; the quantitative comparison in Table 5 uses independent samples), (e) line-conditioned ControlNet output, (f) reconstructed 3D mesh. The fourth row (outlined) is a failure case.

Table 5: Structure preservation and style adherence of the default configuration and the text-only baseline

Configuration	S_{struct} mean $\pm$ SD [95% CI]	Style accuracy [95% CI]	Macro-F1	Mann-Whitney vs. baseline
Text-only baseline (AI Horde, Layer 3), n = 7	0.453 $\pm$ 0.148 [0.316, 0.590]	14.3% [0.0%, 42.9%]	0.056	—
Line-conditioned ControlNet (Layer 1, default), n = 8	0.720 $\pm$ 0.125 [0.615, 0.825]	50.0% [12.5%, 87.5%]	0.146	U = 5.0, p = 0.0059, r_{rb} = 0.82

Table 6: Per-style classification report for the default configuration (zero-shot CLIP, Eqs. (11) – (12))

Requested / predicted style	Precision	Recall	F1-score	Support
Modern	—	0.00	0.00	3
Minimalist	0.60	1.00	0.75	3
Bohemian	1.00	1.00	1.00	1
Contemporary	0.00	— (no support)	0.00	0
Other 8 vocabulary styles	—	—	—	0
Classic (outside vocabulary)	—	—	—	1
Macro average (12 labels)	0.133	0.167	0.146	8
Accuracy			50.0% (4/8)	8

accuracy is also higher under line-map conditioning (50.0% vs. 14.3%), but the bootstrap confidence intervals overlap substantially, so this difference should be read as descriptive rather than as a confirmed effect at the present sample size.

Table 6 gives the per-style classification report for the default configuration, and Fig. 7 the corresponding confusion matrix. Every Minimalist- and Bohemian-requested image was classified correctly. All three Modern-requested images were assigned to

neighbouring styles, Minimalist (2 images) or Contemporary (1 image), consistent with the visual overlap between these categories. The one Classic-requested image, which has no label in the evaluation vocabulary, was assigned to Contemporary and counts as an error in the accuracy of 50.0% (4/8). No image was assigned to any of the remaining eight vocabulary styles, so their precision and recall are undefined at this sample size and enter the macro averages as zero; the macro averages are therefore conservative.

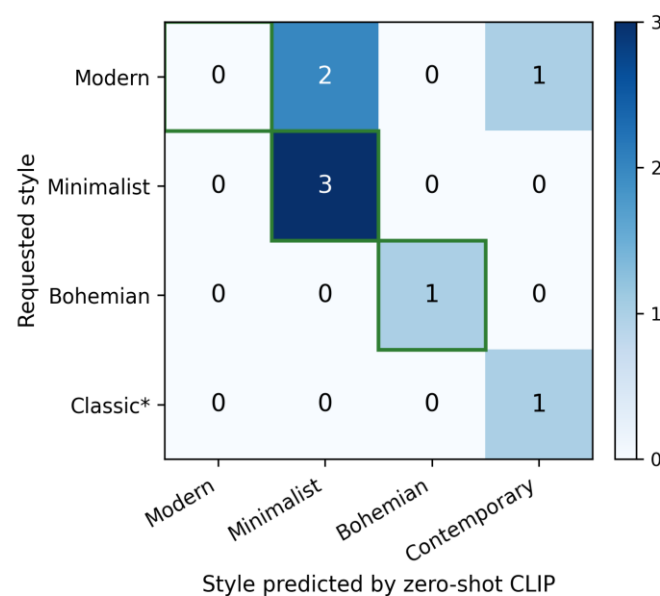


Fig. 7: Confusion matrix of requested versus CLIP-predicted style for line-conditioned generation (n = 8). Diagonal cells outlined in green are correct. *'Classic' has no label in the 12-style evaluation vocabulary; vocabulary labels that received no predictions are omitted.

Table 7: Reliability of the four-layer generation cascade over the evaluation period (29 cascade-era requests)

Layer	Geometry conditioning	Final outputs	Share of requests [95% Wilson CI]
1. ControlNet (Space / SD 2.1 fallback)	Yes (Space) / No (fallback)	8	27.6% [14.7%, 45.7%]
2. SDXL (Replicate)	No	0	0.0% [0.0%, 11.7%]
3. AI Horde	No	11	37.9% [22.7%, 56.0%]
4. InstructPix2Pix	No	0	0.0% [0.0%, 11.7%]
End to end (resolvable permanent image)	—	19	65.5% [47.3%, 80.1%]

Table 7 reports how often each layer of the generation cascade produced the final output, and the end-to-end rate of requests that resolve to a permanent image. Of the 19 completed designs with a resolvable output image, 8 (42.1% of resolvable outputs; 27.6% of all 29 cascade-era requests) came from Layer 1 and therefore carry the geometry-preservation property; the remaining 11 came from Layer 3 (AI Horde), which does not condition on the room's structure.

Table 7 needs three caveats. Layers 2 and 4 produced no outputs at all, because Replicate credits were inactive for the whole evaluation period, so anything that failed at Layer 1 fell straight through to Layer 3. The backend also only records the layer that produced the final output, not failed attempts along the way, so the per-layer attempt rates p_i in Eq. (1) cannot be recovered from these logs. And 9 of the 29 cascade-era requests (31.0% [17.3%, 49.2%]) stored a raw AI Horde URL with a 30-minute expiry instead of a re-uploaded copy, contrary to how Section 3.4.3 describes the pipeline; generation at Layer

3 probably succeeded for these, but the images are gone now, so we count them as not resolvable rather than successful. One further request was logged as failed outright. Two earlier records from 5 March 2026, generated through a prototype endpoint (pollinations.ai) that predates the cascade and has since been removed, are excluded from these figures; folding them back in gives an all-time rate of $19/31 = 61.3\%$ [43.8%, 76.3%].

4.6 Comparison of methods

Table 8 compares the capabilities of the approach categories reviewed in Section 2 with the proposed system. The comparison is made at the level of capabilities rather than headline accuracy figures, because no common benchmark exists across these categories.

An earlier version of this manuscript reported a 'design accuracy' of 95% for the proposed system against 70% for traditional methods, and design completion times of 3–5 days against 12–15 days. Those

Table 8: Capability comparison between approach categories and the proposed system

Approach category	Representative works	Input	Restyles user's own photo	Preserves room geometry	Full-room 3D	AR delivery
Catalogue-based AR furniture placement	[2,6,7,17]	Live camera + asset library	No	Yes (real room is backdrop)	No	Yes
Rule- or optimisation-based automatic layout	[9,10]	Room data + catalogue	No (places objects)	Yes	Object layout only	Yes
Unconstrained text-to-image generation	[15,18]	Text prompt	No	No	No	No
Instruction-based image editing	[22]	Photo + instruction	Yes	Not enforced	No	No
Single-object image-to-3D	[24,25]	Single image	No	No (background removed)	Object only	Not inherent
Proposed (Home Gennie)	This work	One room photo + style + room type	Yes	Yes (line and depth conditioning)	Yes (monocular depth)	Yes (browser, no install)

figures were not derived from a defined metric, a documented sample or a statistical test, and the system contains no trained classifier to which such an accuracy could refer; they have therefore been withdrawn and replaced by the metrics of Section 3.6.

4.7 Interpretation

The design of the system follows from the observation, stated in Section 1, that a useful preview must show the user's own room. Structural conditioning is the response to this: constraining generation with the room's line structure takes the architectural shell from the photograph, so only surfaces, furnishings and lighting are synthesised. Line-conditioned outputs retained substantially more of the input's line structure than unconditioned outputs, with a large effect size (Table 5), and in the qualitative examples the difference shows up as preserved walls, openings and viewpoint (Fig. 6). The failure case shows where the mechanism breaks down. When clutter dominates the line map, the conditioning signal no longer describes the architecture, and structure preservation falls into the range of the unconditioned outputs. The reconstruction stage carries the same principle into 3D, because the mesh is built from a depth map of the generated image, which already retains the original geometry.

The reliability results expose the main practical trade-off of the cascade. Lower layers raise the probability of returning an image (Eq. (1)) but do not enforce geometry, and in the evaluation period only 27.6% of requests were served by the geometry-preserving layer. The cascade therefore protects availability at the cost of the property the system is designed to provide, which is why every output records the layer that produced it. Compared with catalogue-based AR tools^[2,6,17] and automatic layout systems,^[9,10] the proposed system gives up object-level control (individual furniture items cannot yet be moved or swapped) in exchange for a photorealistic restyling of the whole room from one photograph. The two kinds of tool address different parts of the same problem.

4.8 Limitations

- *Dependence on third-party services.* 2D generation and the primary depth layer rely on free public inference endpoints that are subject to cold starts, queueing, rate limits and changes in availability or terms of use; the backend runs as a single process without a task queue, rate limiting or horizontal scaling.
- *Partial geometric guarantees.* Only Layer 1 applies structural conditioning, only one control image is used per request, and conditioning constrains rather than guarantees geometry; in the evaluation period most outputs came from an unconditioned layer (Table 7).

- *Approximate metric scale.* Depth-Anything produces relative rather than metric depth, and the linear mapping of Eq. (4) and the focal-length approximation of Eq. (6) mean that the reconstructed room is not metrically accurate; objects placed in AR should not be used for precise measurements.
- *Single-view reconstruction.* Surfaces hidden from the camera are missing, the discontinuity filter leaves holes behind foreground objects, and the 512×512 resolution limits detail.
- *Privacy.* Room photographs reveal the interior of users' homes. They are currently stored in public storage buckets and are sent to third-party inference providers; private buckets with signed URLs and documented provider data-retention policies are needed before wider deployment.
- *Scope and design of the evaluation.* The evaluation covers 15 end-user photographs, without controlled variation of room type or lighting, and the two configurations were compared on independent rather than paired samples, so differences in the photographs themselves may contribute to the observed gap. The pretrained models were not adapted to Indian homes, and results may not generalise to all room types, cultural styles or capture conditions. The depth-only configuration and per-layer attempt rates were not measured, and AR requires an ARCore- or ARKit-capable device.
- *Data and engineering gaps.* One evaluated request used a style label ('Classic') outside the evaluation vocabulary; 9 of 29 cascade-era designs point to expired AI Horde URLs because the re-upload step was not applied to that layer in every case; there is no automated test suite; and the gallery relies on polling rather than real-time subscriptions.

5. Conclusion

This paper set out to restyle a photograph of an existing room without losing the room itself. Home Gennie, the result, is a web-based system that restyles a room while preserving its geometry, turns the result into a 3D mesh from the same single photograph, and shows it in augmented reality in an ordinary browser. Geometry preservation comes from extracting straight-line and monocular depth maps and using them to condition a ControlNet diffusion model, inside a four-layer provider cascade that keeps the service running on free-tier infrastructure. Full-room reconstruction back-projects a monocular depth estimate through a pinhole model, filters out stretched triangles, and exports a colour-textured GLB mesh whose size follows directly from the processing resolution: 262,144 vertices, at most 522,242 triangles and about 10 MB at 512×512 . On real user requests, line-map conditioning preserved room

structure far better than unconditioned generation ($S_{\text{struct}} = 0.720$ vs. 0.453 , Mann-Whitney $p = 0.0059$); reconstruction averaged 4.47 s across 30 production requests, and 65.5% of cascade-era requests resolved to a permanent image. These results show that pretrained components can be combined into a preview tool that homeowners can use without special hardware. The tool has clear limits: its scale is approximate, it depends on services outside the authors' control, and it preserves geometry only when the first cascade layer answers the request. Future work includes making every cascade output persist reliably, logging per-layer attempts, letting users edit individual objects through segmentation, moving to private storage with signed URLs, trying metric-scale depth or multi-view capture, and running a paired evaluation with real users.

CRedit Author Contribution Statement

Ali Hassan Kadri: Conceptualization, Methodology, Project administration, Software, Visualization, Writing-original draft, Writing-review & editing. **Arfa Shaikh:** Formal analysis, Writing-original draft, Resources. **Aafiya Shaikh:** Data curation, Resources, Writing-original draft. **Isar Fatima Syed:** Data curation, Resources, Visualization. **Farhana Siddiqui:** Supervision, Validation. All authors have read and agreed to the published version of the manuscript.

Ethics Statement

The evaluation used room photographs submitted to the deployed application between March and August 2026. Before analysis, all account identifiers, storage paths and metadata were removed, and results are reported only in aggregate or under anonymized sample identifiers. The evaluated images depict room interiors and contain no people.

Funding Declaration

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Data Availability Statement

The source code of the generation pipeline, the 3D reconstruction algorithm and the evaluation scripts is available at <https://github.com/Akhassan12/Home-Gennie>, and the deployed application is accessible at <https://home-gennie.vercel.app>. The latency log ($n = 30$), the mesh-geometry measurements and the anonymised per-sample structure-preservation and style scores are provided as supporting information. To protect the privacy of private residences, the raw user-submitted photographs are not distributed beyond the examples shown in Fig. 6; anonymised evaluation features are available from the corresponding author on reasonable

request.

Conflict of Interest

The authors received no specific financial support for the research, authorship, or publication of this work.

Artificial Intelligence (AI) Use Disclosure

The authors declare that large language model tools were used to assist with code development and language editing during manuscript preparation. The authors reviewed and verified all content, methods, results and conclusions, and take full responsibility for the published work. No images were manipulated using AI; the AI-generated room designs in Figs. 5 and 6 are outputs of the system under study and are identified as such.

References

- [1] M. Dave, P. Gupta, A. Gandhi, B. Sejal, From visualization to purchase: How augmented reality (AR), virtual reality (VR) and artificial intelligence (AI) influence consumer purchase decisions in housing design decisions, *International Journal of Innovative Science and Research Technology*, 2025, **10**, 1295-1317, doi: 10.38124/ijisrt/25apr1332.
- [2] T. Samant, S. Vartak, Interior design using augmented reality, *International Research Journal of Engineering and Technology*, 2019, **6**, 1003-1007.
- [3] N. Rane, S. Choudhary, J. Rane, Enhanced product design and development using artificial intelligence (AI), virtual reality (VR), augmented reality (AR), 4D/5D/6D printing, Internet of Things (IoT), and blockchain: A review, *SSRN Electronic Journal*, 2023, doi: 10.2139/ssrn.4644059.
- [4] L. Chitra, Artificial intelligence meets augmented reality: Redefining regular reality, BPB Publications, New Delhi, 2024.
- [5] M. Gong, Application and practice of artificial intelligence technology in interior design, *Applied Mathematics and Nonlinear Sciences*, 2023, **8**, doi: 10.2478/amns.2023.1.00020.
- [6] R. S. P. Revathy, A. Harini, S. Sruthika, Augmented reality in interior design, *Journal of Innovation in Image Processing*, 2024, **6**, 305-313, doi: 10.36548/jiip.2024.3.005.
- [7] M. Juned, H. Mishra, M. A. Nursumar, R. Chondekar, S. Jiwani, Interior designing with augmented reality: A practical implementation using Java and ARCore, *International Journal of Creative Research Thoughts*, 2025, **13**, h709-h724.
- [8] R. Zimmermann, D. Mora, D. Cirqueira, M. Helfert, M. Bezbradica, D. Werth, W. J. Weitzl, R.

- Riedl, A. Auinger, Enhancing brick-and-mortar store shopping experience with an augmented reality shopping assistant application using personalized recommendations and explainable artificial intelligence, *Journal of Research in Interactive Marketing*, 2023, **17**, 273-298, doi: 10.1108/JRIM-09-2021-0237.
- [9] J. R. G. Cuevas, J. K. R. Rili, A. N. G. Santillan, V. A. Agustin, LayOut Loud: An AI-powered augmented reality and mobile application for room interior design and layout optimization, Proceedings of the International Conference on Intelligent Cybernetics Technology and Applications (ICICyTA), 2024, 888-894, doi: 10.1109/ICICYTA64807.2024.10912928.
- [10] P. K n, A. Kurtic, M. Radwan, J. M. Lo iciga Rodríguez, Automatic interior design in augmented reality based on hierarchical tree of procedural rules, *Electronics*, 2021, **10**, 245, doi: <http://dx.doi.org/10.3390/electronics10030245>.
- [11] S. Wu, S. Han, System evaluation of artificial intelligence and virtual reality technology in the interactive design of interior decoration, *Applied Sciences*, 2023, **13**, 6272, doi: 10.3390/app13106272.
- [12] X. Jiang, The analysis of interactive furniture design system based on artificial intelligence, Scientific Reports, 2025, **15**, 28961, doi: 10.1038/s41598-025-14886-0.
- [13] A. Samuel, N. R. Mahanta, A. C. Vitug, Computational technology and artificial intelligence (AI) revolutionizing interior design graphics and modelling, 2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kharagpur, India, 2022, 1-6, doi: <http://dx.doi.org/10.1109/ICCCNT54827.2022.9984232>.
- [14] A. Almusaed, I. Yitmen, A. Almssad, Enhancing smart home design with AI models: A case study of living spaces implementation review, *Energies*, 2023, **16**, 2636, doi: 10.3390/en16062636.
- [15] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, B. Ommer, High-resolution image synthesis with latent diffusion models, 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 2022, 10674-10685, doi: 10.1109/CVPR52688.2022.01042.
- [16] L. Zhang, A. Rao, M. Agrawala, Adding conditional control to text-to-image diffusion models, Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 2023, 3813-3824, doi: 10.1109/ICCV51070.2023.00355.
- [17] Y.-C. Huang, Z. Ni, ARID: Augmented Reality Interior Design. In: C. Stephanidis, M. Antona, S. Ntoa, G. Salvendy, G. (eds), HCI International 2025 Posters. HCII 2025. Communications in Computer and Information Science, Springer, Cham, 2025, 2527, doi: 10.1007/978-3-031-94165-8_27.
- [18] D. Podell, Z. English, K. Lacey, A. Blattmann, T. Dockhorn, J. Müller, J. Penna, R. Rombach, SDXL: Improving latent diffusion models for high-resolution image synthesis, International Conference on Learning Representations (ICLR), 2024.
- [19] G. Gu, B. Ko, S. Go, S.-H. Lee, J. Lee, M. Shin, Towards light-weight and real-time line segment detection, Proceedings of the AAAI Conference on Artificial Intelligence, 2022, **36**, 726-734, doi: 10.1609/aaai.v36i1.19953.
- [20] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, H. Zhao, Depth Anything V2, Advances in Neural Information Processing Systems (NeurIPS), 2024, **37**.
- [21] M. Kim, K. Lee, R. K. Balan, Y. Lee, Bubbleu: Exploring augmented reality game design with uncertain AI-based interaction, Proceedings of the ACM CHI Conference on Human Factors in Computing Systems (CHI '23), ACM, 2023, 1-18, doi: 10.1145/3544548.3581270.
- [22] T. Brooks, A. Holynski, A. A. Efros, InstructPix2Pix: Learning to follow image editing instructions, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 2023, 18392-18402, doi: 10.1109/CVPR52729.2023.01764.
- [23] R. Ranftl, A. Bochkovskiy, V. Koltun, Vision transformers for dense prediction, 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 2021, 12159-12168, doi: 10.1109/ICCV48922.2021.01196.
- [24] J. Xiang, Z. Lv, S. Xu, Y. Deng, R. Wang, B. Zhang, D. Chen, X. Tong, J. Yang, Structured 3D latents for scalable and versatile 3D generation, 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 2025, 21469-21480, doi: 10.1109/CVPR52734.2025.02000.
- [25] J. Xu, W. Cheng, Y. Gao, X. Wang, S. Gao, Y. Shan, InstantMesh: Efficient 3D mesh generation from a single image with sparse-view large reconstruction models, arXiv preprint, 2024,

arXiv:2404.07191, doi:
10.48550/arXiv.2404.07191.

- [26] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, Image quality assessment: From error visibility to structural similarity, *IEEE Transactions on Image Processing*, 2004, 13, 600-612, doi: 10.1109/TIP.2003.819861.
- [27] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, I. Sutskever, Learning transferable visual models from natural language supervision, *Proceedings of the 38th International Conference on Machine Learning (ICML)*, PMLR, 2021, 139, 8748-8763.
- [28] E. B. Wilson, Probable inference, the law of succession, and statistical inference, *Journal of the American Statistical Association*, 1927, **22**, 209-212, doi: 10.1080/01621459.1927.10502953.

Publisher Note: The views, statements, and data in all publications solely belong to the authors and contributors. G R Scholastic is not responsible for any injury resulting from the ideas, methods, or products mentioned. G R Scholastic remains neutral regarding jurisdictional claims in published maps and institutional affiliations.