TO-AGENTS: A MULTI-AGENT AI FRAMEWORK FOR SUBJECTIVE PREFERENCE-GUIDED TOPOLOGY OPTIMIZATION

Isabella A. Stewart[1] **Hongrui Chen**[2] **Faez Ahmed**[2*]
[1]Department of Civil and Environmental Engineering
Massachusetts Institute of Technology
Cambridge, MA, 02139 USA
[2]Department of Mechanical Engineering
Massachusetts Institute of Technology
Cambridge, MA, 02139 USA

ABSTRACT

Topology optimization can generate efficient structures, but designers often must manually translate qualitative intent, such as desired visual style, product experience, or manufacturability into solver settings that are not directly tied to those preferences. We present TO-Agents, a multi-agent AI framework that connects natural-language design intent with iterative topology optimization. The framework converts a human-provided problem description into validated solver inputs, runs a topology optimization solver, renders the resulting 3D topology, and uses multiview vision-language reasoning with an independent judge agent to critique each result and revise solver parameters. We evaluate the framework on two long-horizon design tasks: a cantilever beam benchmark and a phone-stand product design. In both tasks, the designer specifies an aesthetic preference for hierarchically branched structures inspired by natural tree morphologies, and the system performs four revision cycles across ten independent replicates. TO-Agents produces at least one preference-aligned design in 60% of trials for each case study, corresponding to up to 6× more successful trials than an ablated pipeline without visual or historical feedback. Judge scores and human evaluations show that the pipeline can identify effective parameter levers, recover from poor revisions, and expand design exploration. A manufacturing agent further post-processes top-ranked designs for additive manufacturing, enabling end-to-end intent-to-prototype design. We also identify failure modes, including overshooting, selective memory, misplaced tools, and incorrect parameter reasoning. These results suggest that agentic topology optimization can shift designers from low-level parameter tuning toward higher-level specification of form and function, while highlighting safeguards needed for reliable autonomous engineering design. Code is available at `https://github.com/bellastewart/TO-Agents`.

1 INTRODUCTION

Engineering design is shaped by preferences for performance, cost, manufacturability, and appearance, as well as the tradeoffs among them. Translating these preferences into a viable product often requires repeated negotiation and revision. Computational design can accelerate this process by exploring many possible candidate solutions, but it also creates a critical challenge where the engineer must express those goals in terms the computational method can act upon [1, 2, 3, 4, 5].

1.1 When Topology Optimization Meets Human Preferences

This challenge is especially apparent in topology optimization (TO), which distributes material within a design domain to optimize an objective such as stiffness under a volume constraint. To formulate a TO problem, the engineer must specify not only the objective but also the loads, boundary conditions, material behavior, geometric and manufacturing constraints, and solver parameters [6, 7, 8]. These choices strongly influence the resulting form. Consequently, reaching an intended design often requires the engineer to inspect the result, revise the formulation, and rerun the optimization. The revision process becomes particularly demanding when the intended outcome is expressed as a semantic or subjective preference. An aesthetic quality, for

*Address all correspondences to istewart@mit.edu

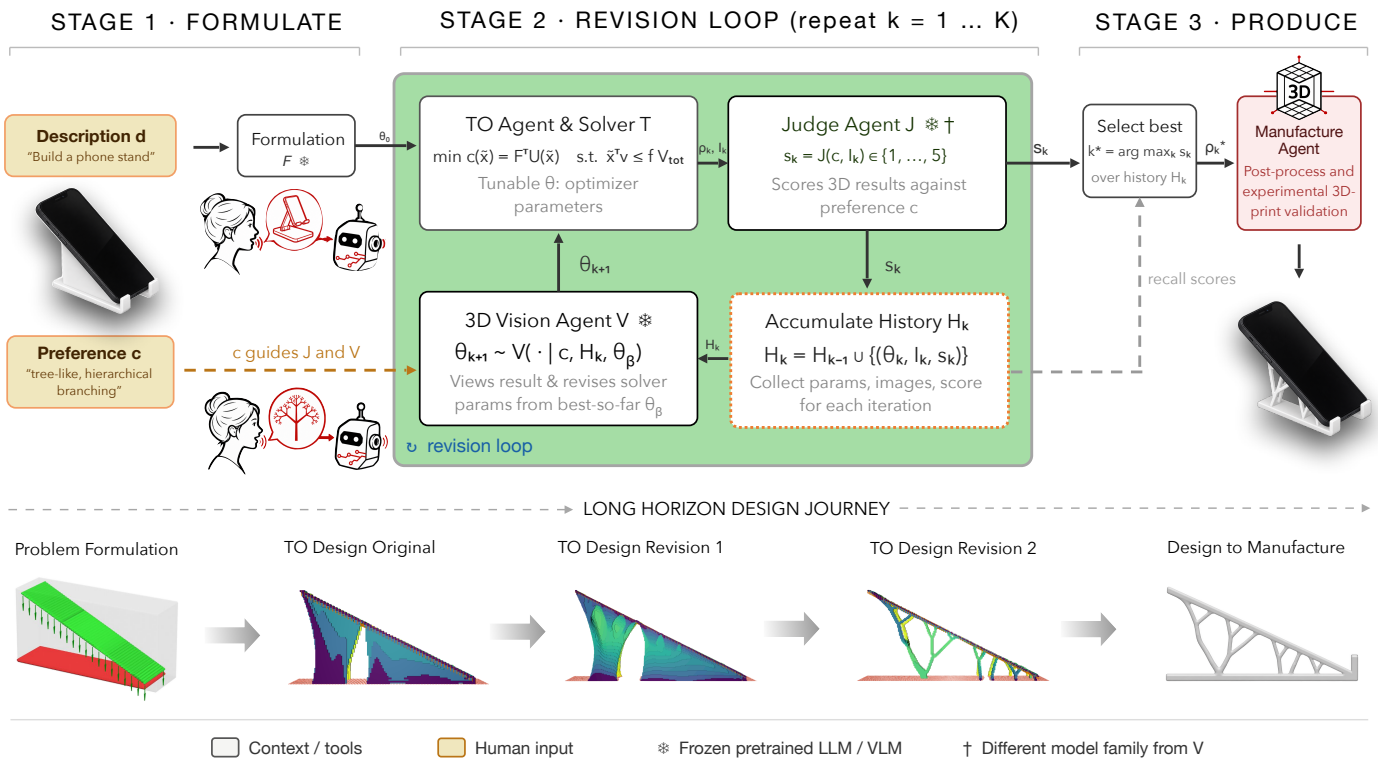


FIGURE 1: The TO-Agents pipeline iteratively revises a topology-optimized design toward a designer's qualitative goal. In this example, feedback from the judge agent guides revisions toward a bioinspired, dendriform structure with hierarchical branching.

example, may arise from the interaction of several design features rather than correspond to any single constraint or parameter. The engineer must infer which changes to the optimization setup might produce that quality, then evaluate the next result and repeat [9, 10]. Thus, even when candidate generation is automated, repeated human interpretation and evaluation can limit the scale and consistency of design exploration.

This limitation matters increasingly as TO extends to products in which ergonomics, aesthetics, and personalization are integral to success, including medical devices, high-performance equipment, and consumer products. Affordable 3D printing has made topology-optimized products more accessible to non-experts, while platforms such as MakerWorld and Thingiverse allow users to share designs and leave comments requesting changes to their appearance, fit, or use. At the scale of these platforms, manually reviewing and incorporating such feedback into each design becomes impractical. Even a brief comment must be interpreted and translated into changes to the TO formulation. Reducing this burden could make it easier for designers to respond to user preferences and for non-experts to participate in TO-based design.

1.2 Translating Human Preferences into Optimization with LLMs and Agentic AI

Addressing this burden requires a way to relate a designer's language to both the TO formulation and the resulting form. Multimodal large language models (LLMs) are a promising candidate because they can interpret natural-language goals and visual results while supporting reasoning across scientific and engineering tasks [11, 12, 13, 14, 15, 16, 17, 18, 19]. In TO, such reasoning could help identify how changes to coupled problem parameters affect performance and appearance, reducing the expert effort required to set up, steer, and evaluate computational design workflows [20, 21, 22, 23, 24]. Because satisfying a qualitative preference may require several rounds of formulation, optimization, and evaluation, a single LLM response is insufficient as a workflow. Agentic systems provide a structure for this process: they can plan actions, invoke tools, assess intermediate results, and use those assessments to guide subsequent steps [25, 26, 27, 28]. Prior multi-agent systems and engineering applications have paired specialized agents with simulation and code [29, 30, 31, 32, 33, 34, 35, 36]. Building on this direction, we investigate agent-driven TO in which the formulation and subse-

quent revisions are guided by a designer's qualitative preference.

1.3 Our Contributions

We introduce **TO-Agents**, a multi-agent AI framework that autonomously formulates and tunes topology optimization problems to produce optimized designs aligned with human preferences. The designer describes the product design and a desired qualitative quality in natural language, and the system then proceeds without further human input: specialized agents translate that intent into solver settings, run the optimization, visually inspect the resulting structure, evaluate how well it satisfies the request, and revise their parameter choices over successive cycles before preparing the best design for manufacture. We examine three research dimensions:

- The translation of qualitative natural-language requirements into TO objectives and parameter selections, including tradeoffs among performance, appearance, and other design criteria;
- The improvement of agent decision-making through experimentation and agentic collaboration, drawing on prior outcomes and knowledge shared by other agents to better align generated designs with human preferences;
- The use of vision-language models' visual reasoning capabilities to evaluate TO outcomes, relate geometric features to design intent, and provide feedback for iterative learning and refinement.

This work positions agents as an interface between designers and topology optimization, bringing engineering performance into closer alignment with human judgment and design intent.

2 RELATED WORKS

Prior methods incorporating human preferences into topology optimization are categorized by intervention timing: before, after, or during execution (Figure 2).

Pre-processing approach These methods encode preferences into the problem statement. Yan et al. [37] extend BESO using regional filters (MR-BESO), local volume fractions (MV-BESO), and element weighting (MW-BESO). Rather than adjusting solver levers, Meng et al. [38] directly specify target geometries, embedding prescribed patterns into free-form shell optimization to control opening placement to control lighting aesthetics. Reviewing this broader category, Li et al. [39] identify a shared obstacle: explicit design inputs must be finalized before any optimization results are generated. Tian et al. [40] attempt to alleviate this constraint via DPDNTO, utilizing a unified interaction parameter and mask-based loss for neural optimization, though the method still necessitates pre-defined geometric inputs.

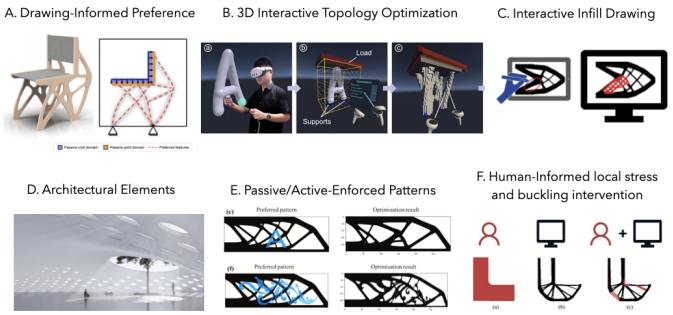


FIGURE 2: State-of-the-art human preference integration: (A) drawing-informed constraints [39]; (B) VR 3D specification [41]; (C) user infill patterns [42]; (D) prescribed features [38]; (E) active/passive regions [40]; (F) local feature-size interventions [1].

Post-processing approach These methods run the solver unconstrained and correct the output. Loos et al. [20] apply Gestalt perception cues (similarity, closure, continuity) to manually refine contours, balancing aesthetic unity and variety [43]. However, post-optimization editing risks degrading structural performance.

Interactive approach These methods place designers inside the optimization loop. Building on HiTop [44], which enables users to locally merge, thicken, or eliminate members, Schiffer et al. [1] introduce HiTop 2.0 to incorporate multi-phase minimum and maximum size controls for both solid and void features within regions of interest. To bypass numerical tuning entirely, researchers have also exploited sketching and drawing as an intuitive medium [45, 46, 47]. Schiffer et al. [42] use hand-drawn infill patterns. SP-BESO [48] converts user scores and sketches into weights. Moving further toward automated and immersive paradigms, SBFE-BESO [49] derives geometry from images via quadtree/octree decomposition, while ISP-BESO [41] combines VR sculpting with similarity constraints for 3D designs.

3 METHOD

TO-Agents is a multi-agent framework that turns a natural-language product description and a single qualitative preference into an optimized, manufacturable design. Given a description d and a preference c, TO-Agents formulates a topology optimization problem, solves it, revises the solver settings over K rounds of visual critique, and prepares the best design for 3D printing. The complete procedure is summarized in Fig. 1 and Algorithm 1.

Problem Formulation. Let $\theta \in \Theta$ denote the full set of solver settings: the design domain and mesh, the supports and loads, the volume fraction f, the filter radius $r_{\min}$, the penalization and the optimizer settings. The solver $\mathscr{T}: \Theta \to [0,1]^n$

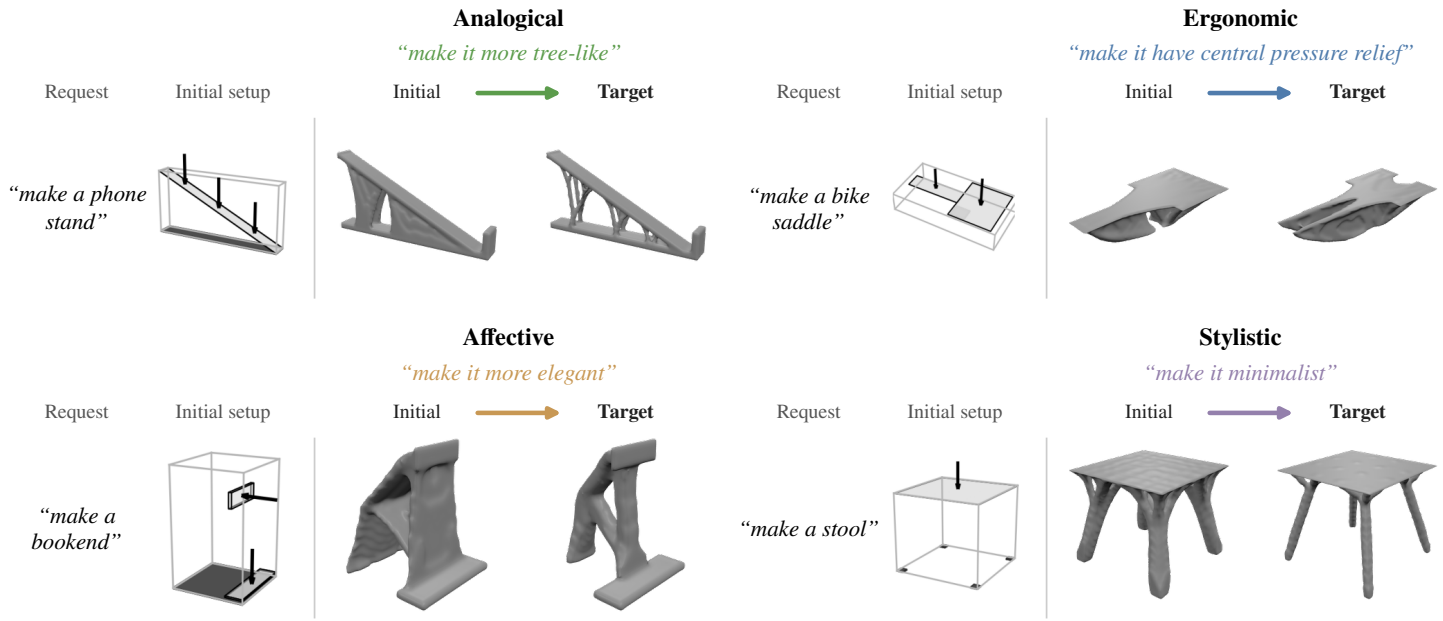


FIGURE 3: Holistic preference categories: analogical, ergonomic, affective and stylistic. In each panel a one-line product request is turned by the formulation agent into a topology optimization set-up with design domain, loads distributed over a node set and supports, and optimized once to give the initial design. A one-line preference in the category then guides the agents' revisions toward the target design. The target designs are revision 4 of one attempt per category.

returns the density field $\rho = \mathscr{T}(\theta)$ of minimum compliance for those settings (Eq. 2), and the renderer $\mathscr{R}$ maps ρ to six orthographic views $I = \mathscr{R}(\rho)$. A holistic preference c cannot be written as an objective or a constraint of $\mathscr{T}$. We therefore seek the settings whose optimized design best satisfies c as judged from its images,

$$\theta^\star = \arg\max_{\theta\in\Theta} \; \mathscr{J}\left(c, \mathscr{R}(\mathscr{T}(\theta))\right), \tag{1}$$

where the judge $\mathscr{J}$ scores agreement with c on a scale of 1 to 5. The search is black-box: no gradient of $\mathscr{J}$ with respect to θ is available, every evaluation requires a full optimization, and the budget is an initial design and K revisions. Compliance does not enter Eq. 1. It is minimized inside every solve, and the stiffness given up to satisfy c is measured afterwards as the compliance ratio C/C_0 (Section 4.1).

Topology Optimization Solver. We use pyFANTOM [50], an open-source, GPU-ready package for finite-element topology optimization. Mesh, physics, boundary conditions and optimizer are separate composable components, so a problem can be specified programmatically and every solver setting exposed to an agent, rather than fixed in a script or hidden behind a proprietary interface. pyFANTOM solves the SIMP minimum-compliance problem

$$\begin{aligned} \min_{\mathbf{x}\in[0,1]^n} \quad & C(\tilde{\mathbf{x}}) = \mathbf{F}^{\mathrm{T}}\mathbf{U}(\tilde{\mathbf{x}}) \\ \text{s.t.} \quad & \tilde{\mathbf{x}}^{\mathrm{T}}\mathbf{v} \le \bar{v} = fV_{\mathrm{tot}}, \\ & \mathbf{K}(\tilde{\mathbf{x}})\,\mathbf{U}(\tilde{\mathbf{x}}) = \mathbf{F}, \end{aligned} \tag{2}$$

over element densities $\mathbf{x}$, where the compliance C is minimized subject to a volume budget set by the volume fraction f and to equilibrium under the load $\mathbf{F}$, with $\mathbf{v}$ the element volumes. A linear density filter of radius $r_{\min}$ maps $\mathbf{x}$ to filtered densities $\tilde{\mathbf{x}}$, whose stiffness follows the SIMP interpolation $E(\tilde{x}_i) = E_{\min} + \tilde{x}_i^p(E_0 - E_{\min})$ between the void and solid moduli $E_{\min}$ and E_0 at penalty exponent p, with Poisson's ratio ν fixed. The gradient-based optimizer uses the standard compliance sensitivity

$$\frac{\partial C}{\partial \tilde{x}_i} = \mathbf{u}_{di}^{\mathrm{T}}\left[p\tilde{x}_i^{p-1}(E_0 - E_{\min})\mathbf{k}_i^0\right]\mathbf{u}_i, \tag{3}$$

with $\mathbf{k}_i^0$ the unit-modulus element stiffness matrix and $\mathbf{u}_i$, $\mathbf{u}_{di}$ the displacement and adjoint vectors on element i. The agents may change f, $r_{\min}$ and p, as well as the mesh resolution, the Heaviside projection, the optimizer with its tolerance and iteration budget, and the loads and supports.

Algorithm 1 TO-Agents: Closed-Loop Visual Optimization

Require: Design description d, user preference c, refinement budget K
Require: Formulator $\mathscr{F}$, reviser $\mathscr{V}$, judge $\mathscr{J}$, producer $\mathscr{M}$
Require: Solver $\mathscr{T}\colon \Theta \to [0,1]^n$, renderer $\mathscr{R}\colon [0,1]^n \to \mathscr{I}^6$

Stage 1: Initial formulation and evaluation
$\theta_0 \leftarrow \mathscr{F}(d)$ ▷ Formulate and validate $\theta_0 \in \Theta$
$\rho_0 \leftarrow \mathscr{T}(\theta_0)$
$I_0 \leftarrow \mathscr{R}(\rho_0)$ ▷ Render six canonical views
$s_0 \leftarrow \mathscr{J}(c, I_0)$ ▷ $s_0 \in \{1,\dots,5\}$
$\mathscr{H}_0 \leftarrow \{(\theta_0, I_0, s_0)\}$

Stage 2: Closed-loop visual refinement
for $k = 1,\dots,K$ **do**
  $b_k \leftarrow \arg\max_{0 \le j < k} s_j$ ▷ Select the incumbent
  $\theta_k \sim \mathscr{V}(\cdot \mid c, \mathscr{H}_{k-1}, \theta_{b_k})$ ▷ Propose an incumbent revision
  $\rho_k \leftarrow \mathscr{T}(\theta_k)$
  $I_k \leftarrow \mathscr{R}(\rho_k)$
  $s_k \leftarrow \mathscr{J}(c, I_k)$ ▷ Evaluate with an independent judge
  $\mathscr{H}_k \leftarrow \mathscr{H}_{k-1} \cup \{(\theta_k, I_k, s_k)\}$
end for

Stage 3: Final production
$k^\star \leftarrow \arg\max_{0 \le k \le K} s_k$
return $\mathscr{M}(\rho_{k^\star})$ ▷ Post-process, mesh, and export

The reviser $\mathscr{V}$ is not provided with documentation of the parameter space Θ; it must infer parameter semantics and sensitivities from pretrained knowledge and from the designs, rendered images, and scores recorded in $\mathscr{H}_{k-1}$. To reduce evaluator–generator coupling, $\mathscr{J}$ and $\mathscr{V}$ are instantiated from different model families.

Pipeline Overview. TO-Agents decomposes the search of Eq. 1 into three stages: (1) problem formulation, (2) a revision loop driven by 3D visual reasoning, and (3) production. Four agents drive the pipeline, the formulation agent $\mathscr{F}$, the vision agent $\mathscr{V}$, the judge $\mathscr{J}$ and the manufacturing agent $\mathscr{M}$, and they act on two tools, the solver $\mathscr{T}$ and the renderer $\mathscr{R}$. The agents speak in a fixed turn order, each only when its role is required. At revision k the state consists of the solver settings θ_k, the density field ρ_k, its rendered views I_k and the judge's score s_k, and the history $\mathscr{H}_k$ accumulates all of them, so that every later decision is conditioned on every earlier design.

Formulation Agent. The designer provides a natural-language description d. It may specify the topology optimization set-up in technical detail or describe the product broadly, as in "make a phone stand". In the latter case $\mathscr{F}$ first writes a design brief that defines the set-up from scratch: the design domain, supports placed and loads sized as the product is used, and functional regions, voids kept clear for the user or neighbouring objects and solids locked where a minimum-compliance solve would not place material (Fig. 3; Section 4.3). A structured-output step converts the brief into a JSON configuration, which is validated with Pydantic [51] to give $\theta_0 \in \Theta$. The solver and renderer then produce ρ_0 and I_0, the judge scores the initial design, and the triple (θ_0, I_0, s_0) seeds the history $\mathscr{H}_0$.

Renderer. Each density field is thresholded at $\rho \ge 0.5$ and rendered with pyFANTOM's visualizer, which extracts the surface with VTK, draws it with k3d and captures it in a headless browser with Playwright, from six orthographic directions with a depth-based colour encoding that improves the spatial interpretability of the images for vision-language models (Appendix A). The vision agent receives the six views of each design as one sheet, together with a front view of the von Mises stress; the judge receives each design as one 3×2 sheet of the six views.

Vision Agent and Reviser. After the initial design, the designer states a single preference c, the only human input beyond d; all later stages proceed autonomously. At revision k, $\mathscr{V}$ inspects the history $\mathscr{H}_{k-1}$, with the images, settings and scores of every earlier design, and proposes revised settings θ_k in natural language. It is given no definition of the parameters in Θ and no guidance on how each affects the topology, so it must infer their meaning and sensitivity from pretrained knowledge and from the evidence in $\mathscr{H}_{k-1}$ (supplementary prompt document). Rather than revising its most recent design, $\mathscr{V}$ anchors each proposal on the incumbent θ_b, the highest-scoring design so far, which lets the loop recover from a poor revision. A reviser converts the proposal into a valid configuration; it sees only the current configuration and the proposal, not the product or the preference.

Judge Agent. The judge $\mathscr{J}$ is instantiated from a different model family than $\mathscr{V}$ to limit self-preference bias [52, 53]. In each call it scores every design of the attempt so far against c, from their image sheets alone, as $s_k \in \{1,\dots,5\}$ with a written justification (supplementary prompt document). It sees neither the solver settings nor the vision agent's reasoning. The scores and the justification are appended to $\mathscr{H}_k$ and reach $\mathscr{V}$ at the next revision. Because the judge compares the designs it is shown, its scores act as ranks within an attempt (Section 4.2).

Holistic Preferences. We consider four categories of holistic preference that capture how people interpret and experience a design as a whole: analogical, ergonomic, affective and stylistic (Table 1, Fig. 4). They concern what the design resembles, how it feels to use, the impression it creates, and the style people recognize. The supporting literature and rationale for each category are given in Appendix F. For each category we evaluate one product with a positive preference and its implied negative.

Manufacturing Agent. The highest-scoring design $k^\star$ is selected, ties going to the later revision, and its density field $\rho_{k^\star}$ is passed to $\mathscr{M}$, which post-processes it into a manufacturable geometry with solid padding of its functional surfaces, converts it to a polygonal mesh and exports it for slicing and 3D printing (Fig. 5; Appendix B). We verified the pipeline end to end with a printed phone stand.

Models and Implementation. The agents are orchestrated with AutoGen [29] in a group chat with a fixed turn order, and models are assigned by role (Table 2). The formulation agent

TABLE 1: Holistic preference categories with example words, and the experiment run in each: one product with a positive preference and its implied negative.

Category	Example words	Benchmark Product	Positive	Negative
Analogical	tree-like, like the Eiffel tower	Phone stand	tree-like	block-like
Ergonomic	relieves strain, pressure, fits the hand	Bike saddle	pressure relief	firm, flat
Affective	sturdy-looking, delicate, elegant	Bookend	elegant	unrefined
Stylistic	modern, brutalist, clean, balanced, novel	Stool	minimalist	ornate

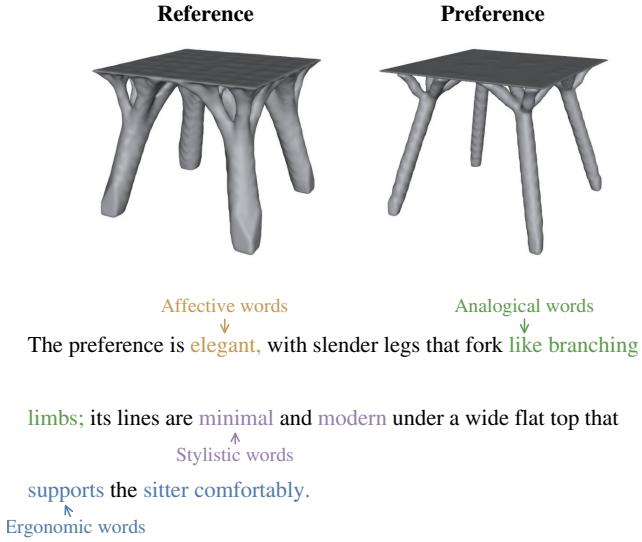


FIGURE 4: The four holistic preference categories in the words of one illustrative sentence, written about two stool designs (an earlier revision and a later one of the same attempt). Each colour marks the words of one category, and the same colours stand for the categories in every figure.

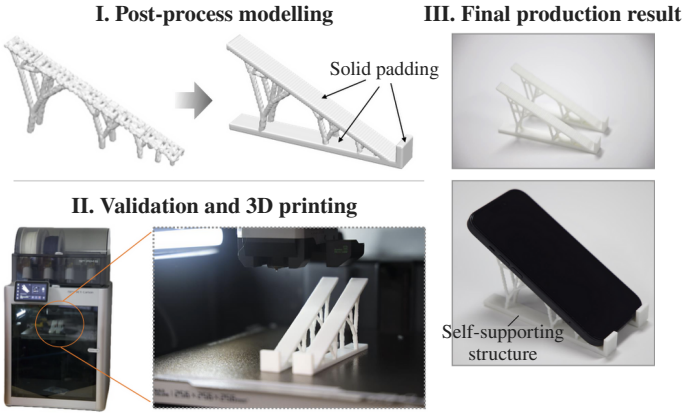


FIGURE 5: Manufacturing agent prepares the best design made by the agentic system for additive manufacturing which includes post-processing the raw TO structure with solid padding followed by meshing of dense regions. This structure can be directly imported and sliced in a 3D-printing slicer. We verify this pipeline end-to-end with the final print shown in the result.

writes its brief with Claude Opus 5.5; the vision agent and the reviser are Claude Opus 5.5, run through the Claude command-line interface; and the judge is GPT-6-Astra, a different model family, run through the Codex command-line interface. The solver runs on NVIDIA A100 GPUs.

4 EXPERIMENTS, RESULTS, AND DISCUSSION

We evaluated whether TO-Agents turned a one-line product request and a single holistic preference into an optimized, manufacturable design, and how the agents reasoned while doing so. Section 4.1 describes the four experiments and the metrics. Section 4.2 reports how far each preference was realized and at what structural cost. Section 4.3 examines how the agents reasoned about a single design, and Section 4.4 how the vision agent used the history of earlier designs and judge feedback. Section 4.5 reports the printed prototypes. All quotes are verbatim from the agents' transcripts.

4.1 Experimental Setup

Tasks. Each task paired a casual request with a whole-design preference. The four tasks were a phone stand that should be "more branching tree-like" (analogical), a bike saddle "relieving central pressure on nerves" (ergonomic), a bookend that should be "more elegant" (affective), and a stool that should be "minimalist" (stylistic). The initial design of a product came from a fixed seed configuration, produced once by the formulation stage and shared by every attempt, so that the attempts differed only in what the vision agent changed afterwards.

Agents and protocol. The agents used the models of Table 2 and saw the inputs described in Section 3. The vision agent received every earlier design with its settings and the judge's latest scores and justification, but not its compliance, and the judge scored all designs of an attempt in one call. Following Algo-

TABLE 2: Agents and tools of TO-Agents.

Component	Role	Model / tool
Formulation agent $\mathscr{F}$	Specifies the initial TO problem.	Claude Opus 5.5
Vision agent $\mathscr{V}$	Interprets rendered designs and proposes revisions.	Claude Opus 5.5
Reviser	Applies revisions to the configuration.	Claude Opus 5.5
Judge agent $\mathscr{J}$	Scores preference satisfaction.	GPT-6-Astra
Manufacturing agent $\mathscr{M}$	Prepares the best design for printing.	Claude Opus 5.5
TO solver $\mathscr{T}$	Produces a density field from solver parameters.	pyFANTOM
Renderer $\mathscr{R}$	Produces six orthogonal views of the design.	VTK and k3d, Playwright

rithm 1, each attempt had an initial design and $K = 4$ revisions, and each product was run for ten independent attempts, giving 50 designs, 40 revisions and 40 judge calls per product and 200 designs in total.

Evaluation metrics. We report four quantities per design.

- **Judge score** $s \in \{1,\dots,5\}$ was the in-loop judge's score from the last call of the attempt. The judge compared the designs it was shown.
- **Compliance ratio** C/C_0 was the final compliance relative to the initial design of the same attempt.
- **Volume fraction** f was the share of the design domain filled with material.
- **Shape change** GD was the change in the overall form of the design, measured as a symmetric Chamfer distance to the initial design (Eq. 5).

Shape change is computed in three dimensions on the centroids $\mathbf{p}_i$ of all solid elements of a design. Each design is first normalized on its own. The centroids are centred on the midpoint $\mathbf{m}$ of the design's bounding box and scaled so that the farthest one lies on the unit sphere,

$$\hat{\mathbf{p}}_i = \frac{\mathbf{p}_i - \mathbf{m}}{\max_j \|\mathbf{p}_j - \mathbf{m}\|}, \tag{4}$$

so that GD compares form independently of size and position. With A and B the normalized centroids of the initial design and of the revision,

$$\mathrm{GD}(A,B) = \frac{1000}{2}\Big(\frac{1}{|A|}\sum_{\mathbf{a}\in A}\min_{\mathbf{b}\in B}\|\mathbf{a}-\mathbf{b}\|^2 + \frac{1}{|B|}\sum_{\mathbf{b}\in B}\min_{\mathbf{a}\in A}\|\mathbf{b}-\mathbf{a}\|^2\Big). \tag{5}$$

The first term measures how far the material of the initial design lies from the revision, and the second how far the material of the revision lies from the initial design. GD therefore counts both removed and added material, and the two designs need not have the same number of elements or the same volume fraction f. The two terms are weighted equally whatever the number of elements in each design.

We deliberately placed no constraint on the vision agent's choices. It could change parameters as it saw fit, leaving the design as open-ended as it would be for a topology-optimization engineer. C/C_0 therefore measures the stiffness given up through all of these choices together, including any material removed, rather than a comparison at equal volume fraction f.

We also report whether an attempt's last revision was its best (*final was best*).

4.2 Preference Attainment and Structural Cost

Preference attainment. The central test of TO-Agents was whether the loop could move a design toward a preference that cannot be written as an objective, using only a judge's reading of rendered images as feedback, and keep improving it from one revision to the next. Table 3 summarizes the four benchmark experiments. In every one of the 40 attempts at least one revision reached the judge's top score, and the best design was always a revision, never the initial design. The phone stand and the stool rose by exactly one point per revision in all twenty attempts, the saddle in seven of ten and the bookend in five of ten. The last revision was the best in 35 of 40 attempts meaning agents benefitted from progressive feedback loops. Figure 7a shows the same pattern as mean scores.

Figure 6 shows one attempt per category. The changes were the ones each preference asked for and differed sharply between products. Material was removed until the phone stand's wedge separated into trunks and branches, a relief channel opened down the center of the saddle, the bookend's thin X-brace gave way to a few rounded members, and the stool's four vaulted arches became four straight legs. These sweeps show that each category reached its top score through its own change in form.

The cost of preference. A preference changed the problem the solver was given, and with it the stiffness the design could reach. We left stiffness and material unconstrained, so that the experiments measured what each preference drove the vision agent to change rather than what a budget allowed. Compliance was still minimized inside every solve, so each design was the stiffest the optimizer found for the settings the vision agent chose. The volume fraction f, like every other solver setting, was left to the agent, because some preferences concern the amount of material as well as its arrangement. The compliance ratios below therefore reflect the price of the preference.

Figure 7b shows that median compliance roughly doubled at every revision of the phone stand and the stool, reaching $12.01\times$

TABLE 3: Preference experiments, ten attempts per product. Judge scores are means over attempts for the initial design, each attempt's best design and its final revision. C/C_0, GD and f are medians at the best design. Every product reached the top in-loop score in every attempt, but the stiffness paid for it ranges from 1.66× to 22.32×.

Category	Preference	Judge s: initial / best / final	Final was best	$C/C_0 \downarrow$	GD	f: $f_0 \rightarrow$ best
Analogical	phone stand, tree-like	1.0 / 5.0 / 5.0	10/10	12.01×	1.43	0.150 → 0.044
Ergonomic	saddle, pressure relief	1.2 / 5.0 / 4.9	9/10	1.66×	1.09	0.150 → 0.120
Affective	bookend, more elegant	1.0 / 5.0 / 4.1	6/10	2.08×	4.63	0.120 → 0.090
Stylistic	stool, minimalist	1.0 / 5.0 / 5.0	10/10	22.32×	0.92	0.120 → 0.050

and 22.32× at the best design and 72.24× for stool attempt 1, while the saddle and the bookend stayed near 1.7× and 2.1×. Figure 8 relates this cost to the change in shape as a Pareto front, which gives for each category the largest shape change any design reached at or below a given compliance ratio. The bookend changed shape the most (median GD 4.63) at a moderate cost, because the filter rounded and merged members rather than removing them, whereas the phone stand and the stool paid the most stiffness for modest changes in overall shape.

4.3 How the Agents Reason About a Design

Formulation agent. The optimizer knew nothing of how a product would be used, so before any optimization ran the formulation agent had to make that use explicit in the problem it posed. The formulation agent's briefs read the product's use into supports, loads and keep-out regions, and named in advance what a minimum-compliance optimizer would get wrong (Table 5). Thus, the brief encoded how the product would be used before any optimization ran. The agent had to infer the form that the optimization would produce from the supports and loads it chooses, ensuring the result meets the user's product goal.

Vision agent. The vision agent views the rendered orthogonal images of the topology optimized structure and translates user preferences into changes to the solver settings. Its reasoning followed the same structure in every turn, from observation to diagnosis to proposal.

Before it changed anything, the vision agent described the design, stated what the preference required of it, named why the design fell short, and proposed concrete changes with an expected result and a fallback. Table 4 follows this sequence through the first turn of attempt 1 in each category, where the vision agent saw only the initial design and had no feedback yet, together with the judge's first verdict on the same design. The paragraphs below give the counts over all vision-agent revisions after this initial design per category and the judge's verdict.

Recovering 3D spatial reasoning from multiple views. The vision agent saw each design only as a set of multi-view two-dimensional renders, so understanding its three-dimensional form required it to combine what the separate views showed. In all 160 vision-agent revisions the vision agent described the current design in checkable structural terms, members, openings, walls and where the load went, and in at least 146 of them it named the view it read them from (34 turns in the phone stand, 36 in the saddle, 40 in the bookend and 36 in the stool), in at least 111 drawing on views from two or more directions. The vision agent at times located features by combining evidence from multiple views to recover the third dimension.

> *The left, right and top views show a full-thickness slab, so each trunk is a flat plate extruded through the depth rather than a round limb. (Phone stand A8, R3)*

> *There is a tiny hole in the middle of the web (the white dot in the front, back and stress views). (Bookend A5, R4)*

> *In the top and bottom views they form a smooth U shape, and the openings in the front and side views line up with each other. (Bookend A4, R3)*

The judge cited views as well in at least 41 of its 160 calls, mostly to say where a quality showed or failed to show.

> *The wider central opening and more distinct tapered supports make the trunk-to-branch relationship clearer in the front and back views. (Phone stand A1, R1)*

> *Slimmer legs and narrower connections consistently reduce visual mass across the side and bottom views. (Stool A6, R1)*

> *. . . broad, heavy flaring in the side and bottom views undermines its graceful profile. (Bookend A6, R2)*

Together, these citations show that both agents read each design as a three-dimensional object assembled from its views rather than from a single image.

From preference to structural requirements. A holistic preference named a quality rather than a structure, so before it could change any setting the vision agent had to decide what the preference required of the material layout. The vision agent translated each preference into a structural requirement. Table 6

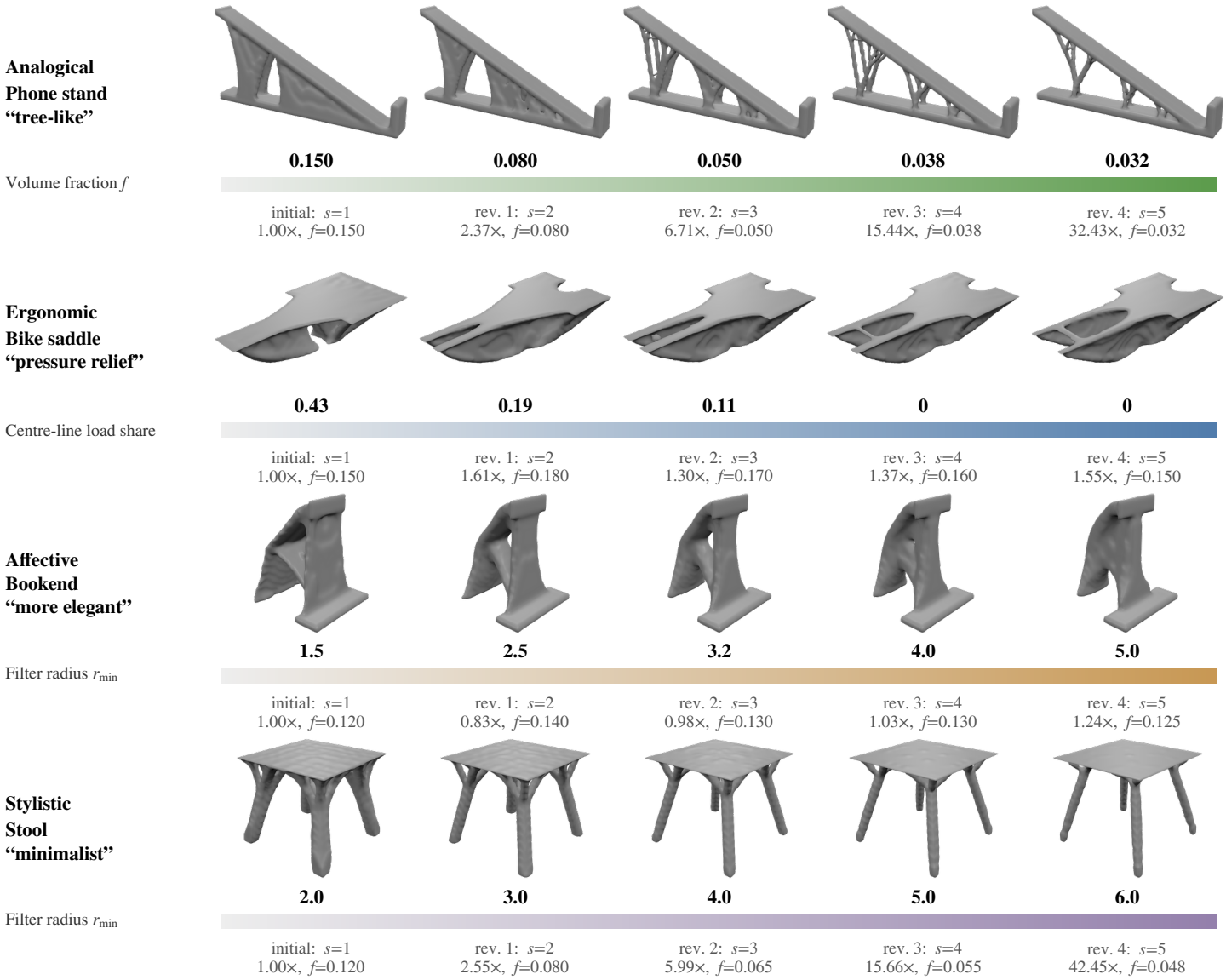


FIGURE 6: The vision agent's main lever in each category, as a sweep of the designs of one attempt placed along that lever's value (the bar runs from the initial to the final value), with the in-loop judge score s, compliance ratio and achieved volume fraction f of each revision. Less material turns the phone stand's wedge into branches, unloading the centre line opens the saddle's relief channel, a larger filter rounds and merges the bookend's members, and a larger filter with less material pares the stool to four legs.

gives the reading of each preference shared by all attempts.

The vision agent's reading was consistent within each category where it explained each preference by how the optimizer distributed material, not by appearance alone. In the phone stand, for example, it tied the branching it wanted to the thinnest member the filter allowed.

> *A smaller filter allows thinner members, so the plate can split into many slender branches instead of blurring into one sheet. This is the biggest change for finer, multi-level branching. (Phone stand A1, R1)*

The vision agent also recognized when a preference worked against the physics of its set-up.

> *For that setup, a continuous shear wall is a very stiff answer. You get discrete branches only if material is scarce or the penalty makes in-between densities costly. (Phone stand A3, R2)*

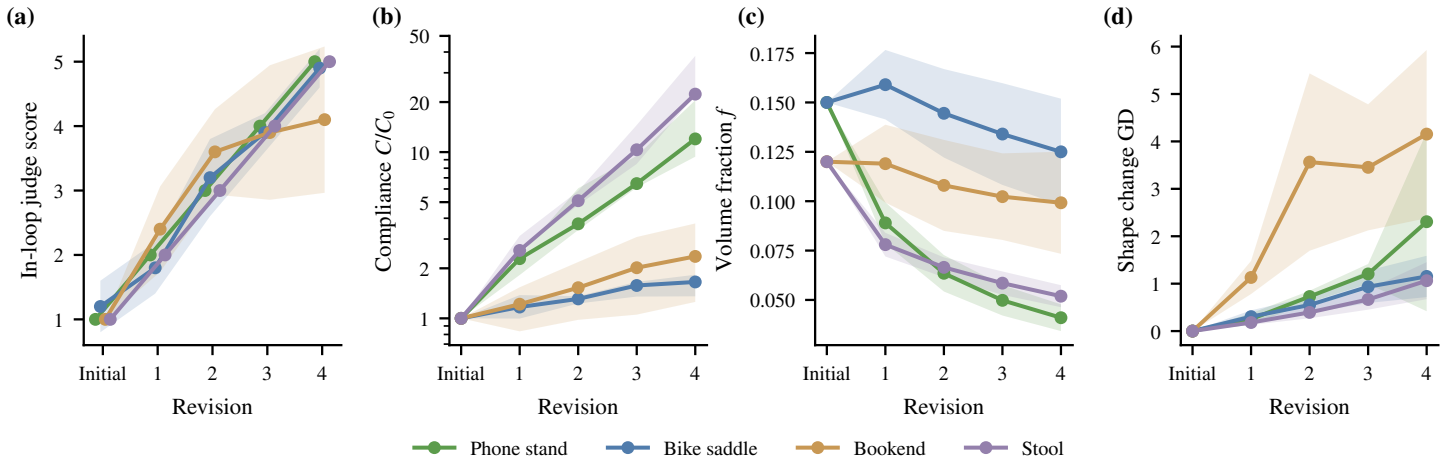


FIGURE 7: Mean in-loop judge score (a), median compliance ratio on a log scale (b), mean volume fraction f (c) and mean shape change GD (d) by revision. Bands are ± 1 standard deviation in (a), (c) and (d) and the interquartile range in (b), over ten attempts. In (a) the curves are offset slightly along the revision axis, because the phone stand and the stool score exactly 1 to 5 in all ten attempts and would coincide. Scores rise in every category, while compliance grows by more than an order of magnitude only for the phone stand and the stool, the two products whose recipe removes material at every step.

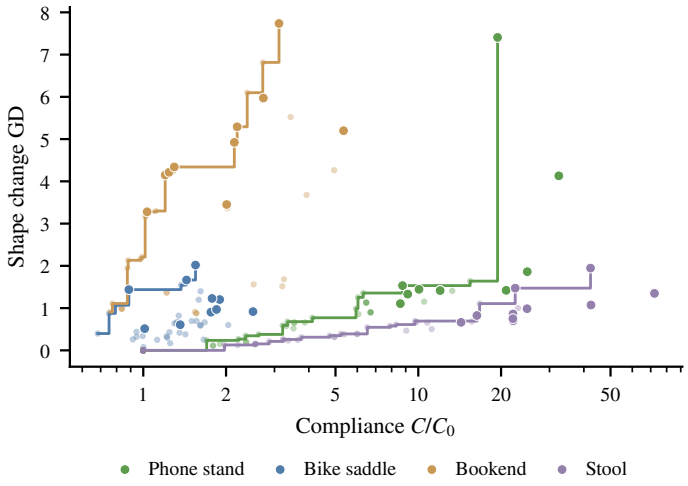


FIGURE 8: Stiffness cost against shape change GD for every design of every attempt (small dots), the best design of each attempt (large dots) and each category's Pareto front (line), the largest shape change reached at or below a given compliance ratio. The bookend's front rises steeply, reaching a GD of 3.28 at about the initial stiffness (1.03×), while the phone stand and the stool first exceed a GD of 1 at 6.04× and 16.71×.

Defining a preference by what it is not. In every category the vision agent defined the preference first by what it was not. It named what in the current design violated the preference, and only then said what the design should become. In the phone stand, it ruled out walls and plates as not tree-like in all ten attempts, describing the initial design as a plate with "no branch points, and no separate members". In the saddle, it called the unbroken top over the centre line the opposite of relief in seven attempts. In the bookend, it named the thin secondary braces as clutter in all ten attempts. In attempt 1 it first listed what made the design inelegant, namely small secondary braces, stair-stepping and members of uniform thickness, and then described the target as a slender, tapered column with an hourglass waist. In the stool it named the vault under the seat as the main obstacle to minimalism. The negative thus fixed what the vision agent removed first, and the target followed from it.

Diagnosing the cause of a shortfall. Similar to the negative used to understand the preference, every one of the 160 turns named a specific structural cause for the design's remaining distance from the preference, not only that it fell short. In the phone stand, 11 turns traced the solid wall to the fully fixed base, which let every load point go straight down. In the saddle, the causes were load on the centre line, the clamp directly below it and spare material. In the stool, the obstacle moved from the vault under the seat in all ten first turns to the forked tops of the legs in eight or nine of the ten attempts at each later turn. Thus, each diagnosis pointed at a setting the vision agent could change.

Reasoning about the use-case. A preference concerned a product in use, so meeting it required attention to the people and objects the product would serve. We find that the vision agent reasoned about the user or objects that mattered to how the product would be used. In the saddle it reasoned about the rider's body throughout, where it discussed the sit bones in all 40 turns,

TABLE 4: How the vision agent reasons about the initial design, step by step, in the first turn of attempt 1 of each category, and the judge's first verdict on the same design. Quotes are verbatim, and "…" marks a cut.

Step	Quote
Phone stand, "more branching tree-like architecture"	
Describes	"Front/back: It is one solid triangular plate. It runs from the lower left, along the loaded diagonal, up to the upper right, and drops to the ground through one wide wall on the right."
Diagnoses	"A tree needs a few trunk members that split into smaller branches reaching the load points. Here there is one hole, no branch points, and no separate members. The spread-out diagonal load is carried by a continuous plate instead."
Proposes	"With less material, a filled plate costs too much. The optimizer has to open holes in the plate and keep only the load paths along the stress bands, which gives separate trunks and branches."
Expects	"…the solid triangle should break into several diagonal members that run from the loaded edge toward the supports and meet at branch points."
Judge	"A central fork suggests a tree-like support, but broad, continuous wedge-shaped walls dominate. Branching is limited rather than expressed throughout the structure." (2/5)
Bike saddle, "relieving central pressure on nerves"	
Describes	"Top shell: the top view shows a solid, unbroken T-shaped plate. It is wide at the rear (high x) and narrow at the nose, and there is no relief channel or cutout on the centreline."
Diagnoses	"Because we apply force on the centre strip, the optimizer puts material there. That is the pressure the user wants removed."
Proposes	"1. Split the rear load into two sit-bone pads and leave the centre unloaded." … "2. Move the nose load off the perineal zone and onto the tip." … "3. Add two thin edge rails so the seating surface stays continuous."
Expects	"Top view: a clear gap or slot along the centreline from about x 0.22 to 0.9. It should be flanked by the rails and the two sit-bone pads, with a solid nose tip and solid outer rear edges."
Judge	"The broad upper surface remains continuous across the center. Openings beneath it provide little visible relief from direct central contact." (2/5)
Bookend, "more elegant"	
Describes	"Left and right views: the buttress is a solid, tapered, flat wall across the full depth (z). It reads as a slab, not a sculpted form."
Diagnoses	"In short, what makes it look inelegant is (a) the tiny secondary braces, (b) the stair-step roughness from the coarse mesh and small filter, and (c) straight, uniform members with no flowing change in thickness."
Proposes	"This is the main lever. A bigger filter removes members thinner than about 2–3 elements, so the X-brace and thin tie disappear."
Expects	"The upright should become a slender, slightly tapered column that flows into one curved diagonal buttress with a clear open window between them, and no small cross-bracing."
Judge	"The sweeping profile and triangular openings suggest elegance, but the broad, solid surfaces visible from the left and right make the overall structure feel heavy and less consistently graceful." (3/5)
Stool, "minimalist"	
Describes	"Legs: four diagonal legs splay out to the corner feet, so it works as a stool. But the legs are thick and wedge-shaped, widening toward the feet into heavy 'boots'."
Diagnoses	"Centre: the legs don't meet the seat cleanly. They merge into a large vaulted dome under the middle of the seat …This is the main thing working against a minimalist look."
Proposes	"This is the biggest lever. It thins the legs and shrinks the dome. With a third less material, the optimizer has to drop the low-stress vault and focus on the direct load paths…"
Expects	"The central vault should shrink to a shallow haunch just under the plate, or disappear. That leaves the space under the seat clearly open and cuts the number of visible features, which is what makes it read as minimalist."
Judge	"Clearly minimalist, with a plain square top and repeated open arches. Broader supports and more pronounced shoulder transitions make the overall structure somewhat less pared down." (4/5)

and moving weight from the nose to the sit bones in 20 turns across all ten attempts.

> *This matches the goal physically, since more weight on the sit bones means less on the soft tissue in the centre. (Saddle A1, R4)*

In the bookend the books appeared in all 40 turns, and in 12 they set a value or a limit, such as a minimum volume fraction f to hold back leaning books.

TABLE 5: How the formulation agent read each request, with each decision in its brief and its reason, verbatim. The phone-stand experiments start from the designer's set-up, whereas the phone-stand rows come from two briefs written from the bare request "make a phone stand", with the phone-stand example removed from the formulation agent's prompt.

Decision in the brief	Formulation agent
Phone stand	
backrest loaded from behind	"…the phone's upper back presses back and slightly down on a small patch just behind the phone-body void; this patch is what pulls up a backrest."
front lip locked solid	"…a short upright wall stopping the phone's bottom sliding forward off the ledge; it costs stiffness, so compliance would never grow it"
Bike saddle	
rear load 0.8, nose 0.2	"Most of the weight goes through the sit bones on the wide rear platform, and a lighter share goes through the narrow nose."
empty blocks beside the nose	"…the inner thigh swings past the nose on every pedal stroke, so without this the optimizer widens the nose with side bracing…"
Bookend	
foot plate locked solid	"…the books sit on this plate, and their weight is the only thing stopping the bookend from tipping or sliding; minimum compliance with a fixed base would never grow it"
Stool	
four feet, floor not fixed	"Fixing the whole floor would grow a central column."

> *The foot plate under the books stays locked solid, so the books' weight still pins the bookend down. (Bookend A3, R1)*

Thus, the vision agent kept the product's use in view while it pursued the preference.

Translating preferences into solver parameters. The final step of each turn translated a description of the desired structure into numerical solver settings, whose effects the vision agent had to know or infer. Every one of the 160 turns turned the requirements for the design into specific parameter values with the expected effect of each, and 156 added a conditional fallback. The following paragraphs describe the levers the vision agent chose and how it reasoned.

A typical turn set a value, said what it should do to the design and named the result that would make it back off, as in this saddle proposal with respect to volume fraction f.

> *With less material available, the optimizer is more likely to empty the now-unloaded centre strip than to keep a thin skin over it. 0.13 still leaves enough for the rear platform and the support down to the clamp. If the rails look thin or broken in the next result, go back to 0.14. (Saddle A1, R1)*

A conditional fallback named the visible failure that would trigger it and the setting to return to as shown in the stool.

> *If the result at a volume fraction of 0.045 comes out with holes in the seat, gaps in the plate edge, or a leg that doesn't connect all the way to the seat, go back to a volume fraction of 0.05 and keep the filter radius at 4.5. (Stool A6, R4)*

The vision agent received no documentation of the parameters, so the choice of levers was its main act of translation. Each category settled on its own recipe. For the analogical "tree-like" preference the vision agent cut the volume fraction f in all 40 revisions, by a median of 24.5% per step, raised the SIMP penalty p in all 30 revisions that changed it, and dropped the filter radius $r_{\min}$ to its floor of 1.5 in all ten first revisions. For "elegant" it raised the filter radius $r_{\min}$ in all 40 revisions, from 1.5 to a mean of 4.0, and called it "the main lever".

> *This is the main lever. A bigger filter removes members thinner than about 2–3 elements, so the X-brace and thin tie disappear. (Bookend A1, R1)*

For "minimalist" it cut material in all 40 revisions and raised the filter radius $r_{\min}$ in 39, from 2.0 to 5.1 on average (Figs. 7c and 6). The reviser carried these proposals into the solver configuration faithfully. Across all four categories, the vision agent changed only a handful of settings each turn. The ten most adjusted parameters of each category, with their initial and final values, are listed in Appendix F.2. Overall, each category settled on a small recipe of levers that the vision agent named and justified before it saw their effect.

Reshaping the problem. Some preferences could not be met by redistributing material within a fixed problem, because a minimum-compliance optimizer always places material where the load demands it. In such cases the vision agent changed the problem itself. The saddle was the one category the vision agent solved through its loads rather than its material. The vision agent stated the mechanism explicitly.

> *Minimum compliance therefore builds the stiffest material exactly where the nerves need relief. (Saddle A3, R1)*

It moved load patches in all 40 revisions, split the full-width rear load into two sit-bone pads at revision 1 in all ten attempts, and never moved the support or changed the penalty p. As a result, the share of the rider's load on the centre line fell from 0.425 in the initial design to a mean of 0.041 at the best design, and three attempts unloaded it completely. This shows that, when a preference concerned where load was carried, the vision agent changed the problem itself rather than the material.

4.4 How the Vision Agent Uses the History

We examined across our experimental replicate the vision agent using the growing history of prior designs, parameter settings and AI-judge feedback to learn progressively how to ma-

TABLE 6: What each preference requires as reasoned by the agent across replicates of each category, and the judge's criterion is visual. Quotes are verbatim, with the attempt in parentheses, and more are given in Appendix F.1.

Category	Reading shared by all ten attempts	Vision agent	Judge
Analogical	a tree gathers a spread load into a few paths to the ground	"The cheapest way to feed many support points from few ground contacts is a trunk that forks, which is exactly a tree." (6)	"…branching remains mostly a single tier rather than a fully developed hierarchy." (1)
Ergonomic	relief means unloading the centre line, because material follows load	"Minimum-compliance optimization always makes the loaded area stiffest. As long as the centreline is loaded, it will get a hard, fully supported ridge, which is the opposite of what you want." (2)	"The broadest central opening provides the most consistent visible clearance over the nerve-sensitive center…" (2)
Affective	elegance means fewer, smoother members and no clutter	"More elegant here means fewer, thicker, smoother members with flowing transitions into the post and the foot plate." (7)	"…an uninterrupted curved body, slender waist, and balanced flared transitions give it the most coherent elegance…" (3)
Stylistic	minimalism means a plate on four legs, one strut per leg	"That second set of members is the main thing keeping it from reading as 'seat plate + four legs'." (4)	"The plain square top and repeated open arches create a clearly minimalist object…" (10, initial design)

nipulate the TO parameters toward the human designer's objective. At every revision the vision agent received, for each earlier revision, its solver parameters, judge scoring with the judge's written feedback, and its rendered views.

Using judge feedback for credit assignment. An iterative loop could improve a design only if the vision agent could tell which of its changes had raised or lowered the score, which is the problem of credit assignment. In every category the vision agent named the parameter that realized the preference, with a mechanism, and the judge's next ranking evaluates the effect. The lever was chosen in the first turn, from prior knowledge, rather than discovered over the trajectory. The accumulated history then added credit assignment.

With reward from the judge, after a score rise the vision agent repeated the same move in nearly every turn and cited the history as the reason, in all 30 such turns in the phone stand and the stool, 24 of 27 in the saddle and 21 of 27 in the bookend. The vision agent credited the judge's rising scores to the moves that preceded them and pushed the same strategy further. In the phone stand, after revision 2 scored 4, it traced every gain to one combined move.

> *Revision 2 changed four parameters together, all in the same direction: less material, a stronger penalty, more iterations and a tighter tolerance. Each step up in score followed that move. (Phone stand A1, R3)*

In the saddle it credited the split of each load into left and right patches with the rise from 2 to 4 and planned more of the same.

> *The next step is more of the same: wider unloaded midline gaps, a smaller share of load on the nose, and less spare material that could fill the gap. (Saddle A7, R2)*

In the bookend it raised the filter radius $r_{\min}$ again, from 2.5 to 3.2, because the previous increase had raised the score.

> *This is the main change, continuing the step that worked. (Bookend A3, R2)*

The vision agent also cited renders in its accumulated history as evidence to point out features. In the phone stand, for example, the score rose from 3 to 4 when the final penalty p went from 4.0 to 4.5, and the vision agent raised it again to 5.0, aiming the step at a feature in the latest render.

> *The higher cap of 5.0 penalises in-between densities more, so partly formed slots like the one in the right column should open into real forks instead of filling back in. (Phone stand A6, R3)*

The history set the size of the next step more often than its direction. In 32 turns the vision agent sized a step from the trajectory or stated a stop rule ("smaller steps", "half the previous step", "the gain is getting smaller"), whereas no phone-stand or stool turn reversed a parameter.

Recovery from low-scoring revisions. Another behavior observed in the loop was recovery from low-scoring revisions. When a decision moved the design away from the preference, the vision agent used the history of prior revisions and the judge's feedback to reason backward, identified the most effective earlier revision, and continued from it with the responsible parameter corrected. Figure 9 shows the in-loop scores of all 40 attempts. When a revision scored below the best design so far, the incumbent rule of Algorithm 1 sent the vision agent back to the best design, and it then had to decide which of its own changes had caused the drop. One saddle attempt and one bookend attempt showed this most clearly.

In the saddle, the first revision scored 2, one point below the initial design. The vision agent blamed the larger filter radius

$r_{\min}$ it had just introduced, went back to the initial design and set the filter radius $r_{\min}$ back to 2.0.

> *A larger filter blurs small voids and joins the two side load paths into one thick web. That's the most likely reason the Revision 0 opening disappeared. Go back to 2.0. (Saddle A8, R2)*

Its next revision scored 4 and overtook the initial design.

In the bookend, the third revision grew an extra leg and fell from 5 to 3. The vision agent built its next proposal on revision 2 rather than on the latest design, and it kept the filter radius $r_{\min}$ at revision 2's value of 3.0 instead of the 3.6 it associated with the drop.

> *Setting 3.6 went with the worse result. At 3.0 the curves are already smooth, and the minimum member thickness stays sensible at the lower volume. (Bookend A1, R4)*

The revision built on this rollback scored 5 and was the best design of the attempt. In both cases the vision agent used its history to trace the score drop to one change and undo it, and by building on the best design in that history rather than on the latest one, it reached a better design.

Taking up the judge's feedback. Another rewarding behavior occurs when the vision agent pays careful attention to judge feedback. In many instances, the vision agent turns the judge's verdict into a located geometric target, mapping a phrase onto a region of the render and making it the aim of the next revision. In the phone stand the judge wrote that "branching remains mostly a single tier rather than a fully developed hierarchy", and in its next turn the vision agent made that phrase the target of the revision.

> *The arch openings should multiply and become more evenly sized, which is the 'fully developed hierarchy' the judge said was missing. (Phone stand A1, R3)*

In the saddle the judge noted that "Relief does not extend along the entire centerline", and the vision agent quoted the remark back when it planned a relief channel running down the centre line from near the nose to the rear.

> *This addresses the judge's remark that relief 'does not extend along the entire centerline'. (Saddle A8, R3)*

In the bookend the judge found that "The abrupt transitions into the feet and top block keep it from feeling fully resolved", and the vision agent aimed the next revision at that quality.

> *Together that should give one continuous flowing shape, which is the 'fully resolved' quality the judge felt was missing. (Bookend A4, R3)*

Together, these cases show that the judge's feedback served the vision agent as a located target, not only as a score.

Designing readable experiments. The vision agent strategically made use of the accumulated history as a tool to learn from optimizer behavior. In every category some turns planned the next revision so that its result could be read from the history in a future revision.

The vision agent held a lever fixed, changed a single setting, bracketed the best value between two revisions, or noticed diminishing returns and moved to another lever. This demonstrates the vision agent planning ahead to trace cause and effect similar to how engineers systematically explore the TO design space. In the phone stand it set up an experiment where it held the penalty p at 4.0 while it cut the volume fraction f, so that the next design would isolate the effect of one lever.

> *Keeping it fixed means the result shows the effect of the volume change alone. (Phone stand A7, R2)*

In the saddle, after its latest revision scored lower than the one before, it returned to the best design and changed only one setting, in order to test that change on its own.

> *Changing only the rear gap tests Revision 3's one harmless change on its own. (Saddle A7, R4)*

In the bookend it used two neighbouring results to locate the best setting between them, and it returned to revision 2 with only the filter radius $r_{\min}$ changed, from 3.0 to 3.25.

> *Revision 2 scored 5/5. Revision 3 moved further in the same direction and dropped to 3/5, so the best settings lie close to Revision 2, just short of Revision 3. (Bookend A4, R4)*

Thus, the vision agent used its history as a record of controlled experiments, not only of outcomes.

4.5 Prototyping and Production

The final test of a design was whether it could be built and used as a physical product. The manufacturing agent selected the highest-scoring design of each attempt, taking the most recent revision on ties. The manufacturing agent then post-processed and meshed the design and exported an object file that was imported directly into Bambu Studio. We scaled the exported phone stands so that the longest edge was 120 mm and printed several of the best-scoring designs in standard PLA on a Bambu Lab X1C printer at the default 0.2 mm layer height, confirming that they were structurally sound. One printed prototype is shown in Figure 5.

5 HUMAN BASELINE

To assess how closely the AI judge's scores align with a human baseline, we conducted a study with four participant human reviewers. Each participant was tasked with independently evaluating each replicate. All the designs evaluated are shown in Figures 15, 16, 17, and 18.

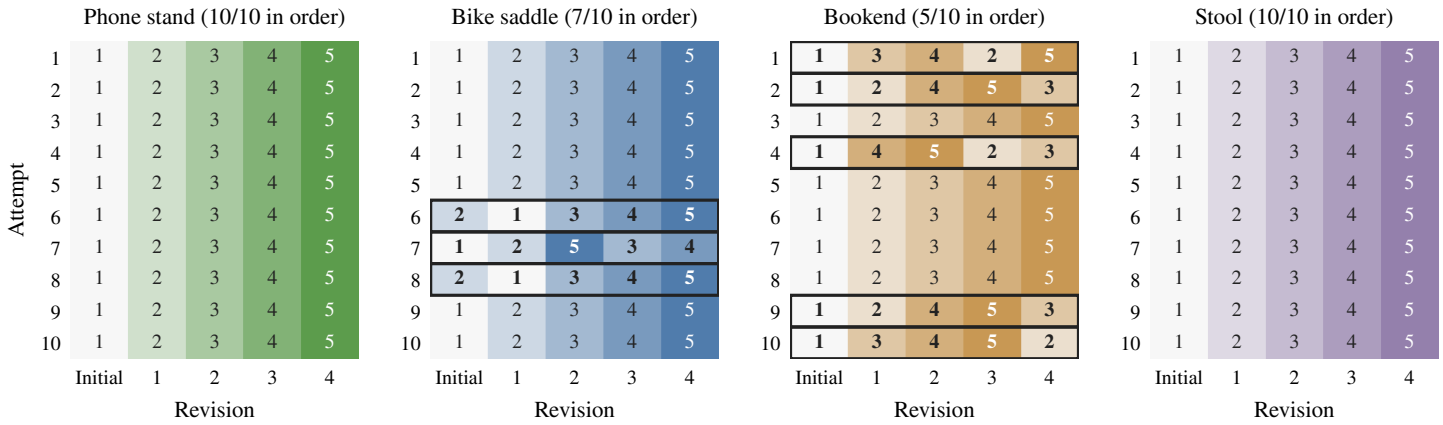


FIGURE 9: In-loop judge scores of every attempt by revision. Outlined rows depart from the one-point-per-revision pattern. All twenty phone-stand and stool attempts rise in order, and departures occur only in the saddle and the bookend, where the judge ranked a continuation of the vision agent's trend below an earlier design.

5.0.1 Protocol Each replicate, consisting of one original design and four revisions, was treated as a separate question. Reviewers received the same instructions as the AI judge and evaluated the images without additional context. Designs were rated from 1 (poor) to 5 (excellent) based on overall visual complexity, particularly hierarchical, tree-like branching relative to the original structure. Reviewers were instructed to evaluate designs only within the same sequence.

This baseline has several limitations. Human reviewers may experience fatigue or develop implicit biases from previously evaluated designs, despite instructions to assess each sequence independently. In contrast, the AI judge evaluates each replicate in isolation without carryover between questions. Accordingly, our goal is not a perfect one-to-one comparison, but rather to determine whether the AI judge exhibits reasonably reliable judgment overall.

5.0.2 Correlation The correlation between the human reviewers and the AI judge is shown in Figure 10. At the aggregate level, the AI judge (Gemma-3-27B) and the human raters yielded comparable mean scores, with $s = 3.04$ and $s = 3.15$, respectively, for Set A, and $s = 3.00$ and $s = 2.97$, respectively, for Set B. Both capture the general upward trend across revisions. An exception appears in Set B, where there is a larger discrepancy between human and AI judge ratings of the original design.

5.0.3 Human-AI Inter-Rater Agreement We further looked at the inter-rater agreement between all human rater scores compared against the AI judge's score. Since each of the four human raters independently evaluated every design using the same 1–5 integer scale as the AI judge, we obtain 200 paired scores per set (4 raters × 10 replicates × 5 revisions). For Set A, shown in Figure 10 B, the confusion matrix reveals a visible diagonal ridge. Overall, 32% of pairs are in exact agreement and 72% fall within ± 1 (Cohen's $\kappa = 0.138$), indicating modest but structured alignment.

For Set B in Figure 10 D, the matrix does not show the same diagonal concentration. Instead, a single off-diagonal cell at (Human=2, AI=3) dominates with 31 of 200 pairs, meaning that when human raters judged a cantilever design as poor, the AI judge assigned it a middling score ($s = 3$). A key factor underlying this discrepancy is the AI judge's compressed scoring confidence: 89% of its scores fall in the 2–4 range, compared to human raters who use the full 1–5 scale more liberally. Humans are more willing to commit to extreme judgments, confidently giving extremely low scores ($s = 1$) when something looks bad and extremely high scores ($s = 5$) when something looks great. In contrast, the AI judge hedges toward middling scores. The cantilever beam case study is inherently more difficult to evaluate, even for human raters, in terms of whether it satisfies the designer's intent regarding aesthetic branching thus judges hedge toward safer scoring.

5.0.4 Human-Human Variability Comparison We further examined whether AI–human disagreement falls within the natural variability observed among human raters. In Set A (Figure 10B), human–human agreement was relatively strong, with exact agreement of 40–66%, agreement within ± 1 of 84–96%, and positive κ values (0.24–0.58). Human–AI agreement was lower but followed a similar pattern, with exact agreement of 30–32%, agreement within ± 1 of 66–76%, and κ values of 0.12–0.16. Although weaker than human–human agreement, the AI remained within a comparable range, suggesting potential for AI to function as an additional fifth rater.

In Set B (Figure 10D), human–human agreement again remained strong, with exact agreement of 40–46%, agreement within ± 1 of 80–98%, and κ values of 0.22–0.32. However, human–AI agreement declined: exact agreement dropped to 10–25%, agreement within ± 1 to 58–75%, and κ values approached zero or became negative (-0.11 to 0.02). Unlike Set A, the AI appeared misaligned with human judgment, tending to favor mid-range scores when uncertain. Despite weaker agreement at the individual level in Set B, the AI judge maintained strong aggregate-level agreement across both studies and multiple replicates, supporting its effectiveness as a directional feedback signal for guiding the revision process toward improved designs.

6 ABLATION

We next evaluate which of the vision agent's inputs the loop depends on. The agent receives four channels at every turn: the rendered views of every design, the history of earlier revisions, the judge's scores and feedback, and the solver configurations. We remove them one at a time, and all together, and measure the effect on the designs.

Task. We run the ablation on the phone stand with the preference "more branching tree-like architecture". An analogical preference names an external referent, so whether a design is tree-like is a clear visual signal, a matter of branching that can be checked, rather than a matter of taste on which reasonable people disagree. This makes it the category in which a change in the agent's inputs can be attributed to the agent rather than to the judge's opinion. The phone stand is also the product of our earlier open-model ablation, which removed all channels at once.

Conditions. All five conditions use the same seed configuration, agents, judge, solver and reviser as Section 4; only what the vision agent receives changes (Table 7).

- **Full pipeline (ours)**: the configuration of Section 4; its ten attempts are the phone-stand runs reported there.
- **No images**: no renders or stress plots; history, judge feedback and configurations remain.
- **No history**: only the current design, its score and its configuration.
- **No judge feedback**: renders, history and configurations, but no scores or judge comments.
- **Blind**: no renders, history, scores or configurations, only the text of the original problem; this is our earlier ablation.

Without history or judge feedback there is no best-scored revision to build on, so in those conditions the reviser edits the latest configuration. Each vision call is audited to confirm what the agent received. Every condition ran until ten complete attempts of five designs were obtained; an attempt whose solve diverged, typically because the structure disconnected, was discarded and restarted under the same rule in every condition, and we report the number of restarts as an outcome. Before the runs we recorded a prediction: no images and blind lowest, no history in between, and no judge feedback close to the full pipeline.

TABLE 7: Ablation conditions, from the full pipeline down to the blind agent, ordered by the final blind judge score. Each row strips channels from the vision agent's input: • kept for all designs; ○ kept for the current design only; × removed. Without a best-scored design to build on, the reviser edits the latest one. Every channel removed lowers the score, and removing all of them returns the initial design's score of 2.

Condition	Renders	Judge feedback	Builds on best	Earlier revisions	Solver configs.	Blind score ↑
Full pipeline (ours)	•	•	•	•	•	**3.90**
No images	×	•	•	•	•	3.60
No judge feedback	•	×	×	•	•	3.35
No history	○	○	×	×	○	3.10
Blind	×	×	×	×	×	2.00

Blind pooled judge. The in-loop judge ranks the designs of one attempt (Section 4.1), so its scores cannot be compared across conditions. We therefore score every unique design of every condition alone, without its condition, revision or history, with the same model, rubric and 1–5 anchors as the in-loop judge, twice and in shuffled order, and average the two calls; each call returns a whole-number score, so a design scores a half point when its two calls differ by one. The protocol is repeatable: across the 287 designs scored, the two calls agreed exactly on 94% and within one point on all. We compare each condition with the full pipeline using a two-sided, tie-corrected Mann–Whitney test on the ten final-design scores, with a Bonferroni threshold of 0.0125 for four comparisons.

Results. Table 8 reports the outcome. The full pipeline produced the best designs, with a mean blind score of 3.90 and 9 of 10 attempts ending at 4. Removing all channels, the blind condition of our earlier ablation, lowered the score to 2.00, the same as the initial design, in every attempt ($p_{\mathrm{MW}} = 0.00002$). Of the single channels, the history matters most: without it the mean fell to 3.10 ($p_{\mathrm{MW}} = 0.0006$) and only one attempt reached 4. Removing judge feedback (-0.55, $p_{\mathrm{MW}} = 0.055$) or images (-0.30, $p_{\mathrm{MW}} = 0.50$) lowered the mean, but neither difference is significant with ten attempts. Figure 11a shows how the conditions diverge over revisions. The full pipeline reached 4 by revision 3 in every attempt and held it in nine, the no-images condition started more slowly but was still rising at revision 4, the no-history condition gained for two revisions and then stalled near 3, and the blind condition never left 2. Figure 12 shows every design.

How the designs differ. Without images, the agent applied the same recipe as the full pipeline and reached an almost identical design, at a compliance ratio of 12.03× against 12.01×.

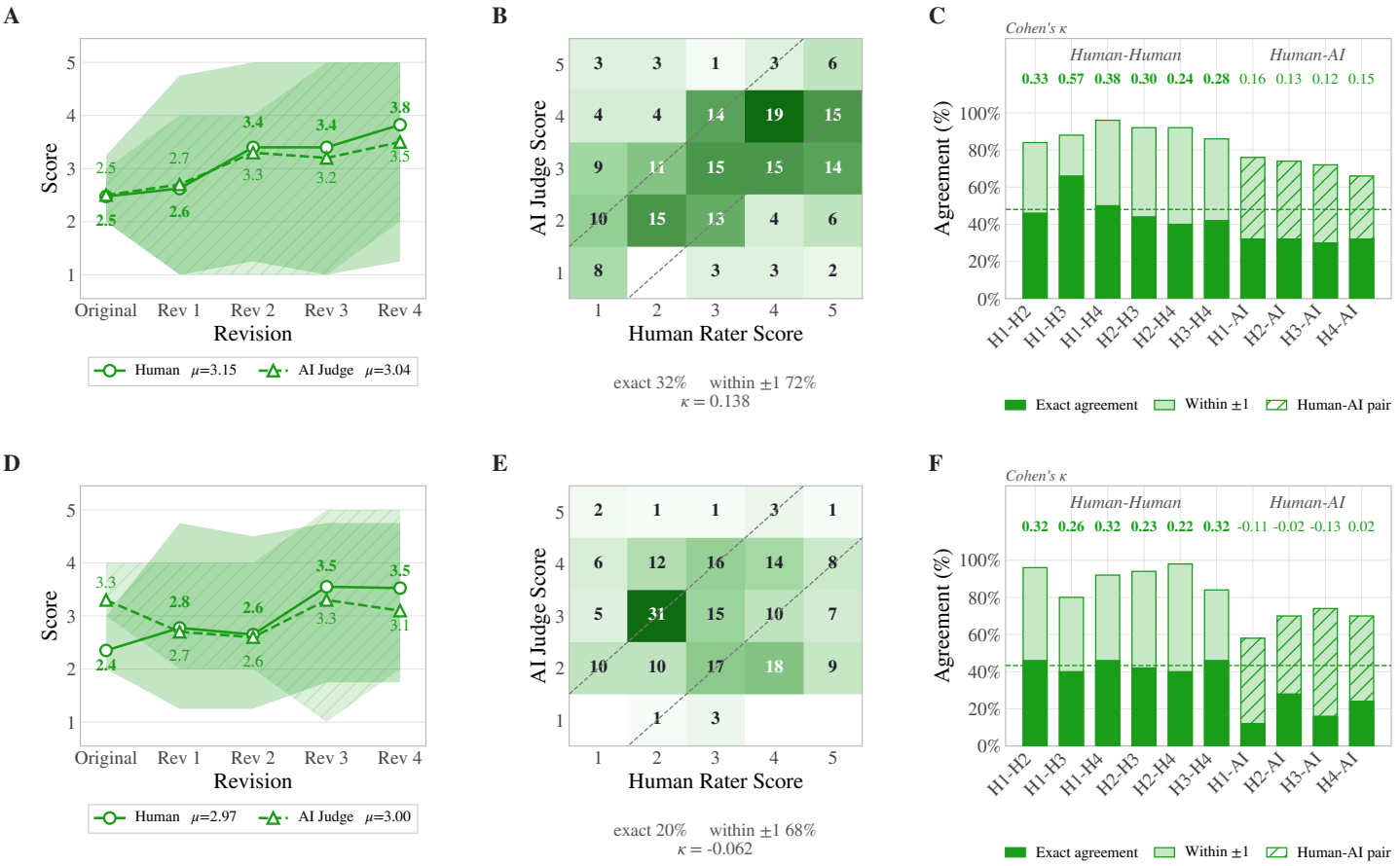


FIGURE 10: (A) Mean scores across revisions in Set A (the phone stand case study), showing similar upward trends for both human raters and the AI judge. (B) Confusion matrix for Set A ratings, showing human–AI agreement (32% exact agreement, 72% within ± 1, $\kappa = 0.138$). (C) Pairwise agreement comparisons for the Set A task, where human–AI agreement is positive but lower than human–human agreement. (D) Mean scores across revisions in Set B (the cantilever beam study), where human scores improve while AI scores remain more stable but follow the overall trend. (E) Confusion matrix for Set B ratings, showing weaker human–AI alignment (20% exact agreement, 68% within ± 1, $\kappa = -0.062$). (F) Pairwise agreement comparisons for the Set B task, where human–AI agreement falls near the lower end of the human inter-rater range.

Without history or judge feedback, the agent also changed the problem: it narrowed the fixed support from the whole base to a central strip, in attempt 9 of the no-judge condition to 3% of the base, and refined the mesh, so that the stand balances on a single trunk and its ledge warps. These conditions also took more than twice as long per attempt. In attempt 1 the blind agent produced the same design at every revision after the first.

Agent behaviour under each ablation. Table 9 counts what the vision agent did in each condition. The full pipeline is the only condition that cut material in every revision and never reversed direction. Each removed channel produced its own failure.

- *No images.* The agent acknowledged the missing renders in 36 of 40 calls and reasoned from the judge's prose instead: "Rev 0's settings did produce a trunk splitting into ribs. Its remaining problem is the broad continuous surfaces, not a lack of branching." Its two failed attempts were the two that pushed the volume fraction f back up to 0.15 or more.
- *No history.* Unable to see which way it had been moving, the agent raised the volume fraction f in 13 of 40 revisions, 11 of them right after a high in-loop score, and reversed direction 15 times:

> *The judge scored this 5/5, so the direction is right.*

- *No judge feedback.* With nothing to tell it when to stop, the agent kept cutting material in 33 of 40 revisions, refined the

TABLE 8: Ablation results, ten attempts per condition. The full pipeline (ours) produces the best designs on every preference measure: the highest blind score, the most attempts reaching 4, and the only steady climb. The blind score is the pooled judge's mean for the final designs. Gain is the full pipeline's score minus the condition's, and p_{MW} tests it (two-sided Mann–Whitney p-value, tie-corrected; significant below 0.0125). Volume fraction f is a mean and C/C_0 the median over final designs. Restarts are the attempts discarded to obtain ten complete ones; the no-images count comes from an earlier, shorter queue window. Improvement is relative to the strongest ablation.

Condition	Blind score ↑	Gain of ours	p_{MW}	Attempts at 4 ↑	f	C/C_0	Restarts
No images	3.60	+0.30	0.50	8/10	0.072	6.0×	3
No history	3.10	+0.80	**0.0006**	1/10	0.069	9.5×	79
No judge feedback	3.35	+0.55	0.055	5/10	0.047	10.0×	162
Blind	2.00	+1.90	**0.00002**	0/10	0.106	1.8×	12
Full pipeline (ours)	**3.90**	–	–	**9/10**	0.041	12.0×	33
Improvement (%)	**↑8.3**			**↑12.5**			

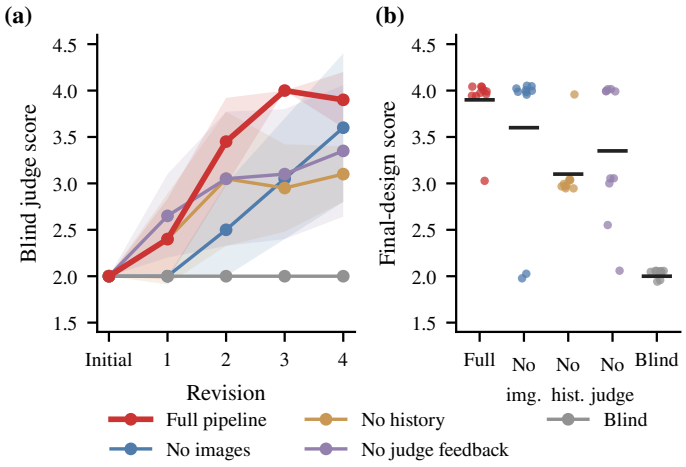


FIGURE 11: (a) Mean blind judge score by revision for each condition; bands are ±1 standard deviation over ten attempts. (b) Final-design scores of the ten attempts, with the mean as a bar. Only the full pipeline climbs steadily and reaches 4; the blind agent never improves on the initial design.

mesh in 22 and narrowed the supports in 14. Solves came apart five times as often as in the full pipeline, 162 restarts against 33.

- *Blind.* The agent never saw its own result, treated the volume fraction f as 0.15 in all 30 of its later calls, and re-proposed the same first step: "I can't confirm what the current design looks like. I've assumed the usual result for this setup". Four of its 40 revisions copied the previous configuration exactly.

In-loop versus blind scores. The in-loop judge's final-design means were 5.0, 5.0, 3.7, 4.7 and 2.9 for the five conditions, against blind means of 3.90, 3.60, 3.10, 3.35 and 2.00. Because it ranks the designs of an attempt against one another, the in-loop judge rewarded any sequence of distinct designs: no-images attempts 4 and 6 ended at an in-loop 5 while every one of their designs scored 2 blind, and the blind condition, whose designs were nearly identical, still received in-loop scores of 3 to 5. The in-loop judge is therefore adequate for choosing the best revision within an attempt, which is how the pipeline uses it, but an independent, context-free score is required to compare conditions.

Discussion. The prediction was partly right. Blind was the lowest and no history fell in between, but removing images

TABLE 9: Vision-agent behaviour by condition, counted over the 40 vision calls and revisions of each. Only the full pipeline (ours) cuts material in every revision, never reverses direction and grounds its reasoning in what it sees. Visual observations are calls that describe what the renders show; prior-based calls reason from what a setup "usually" produces.

Per 40 revisions	No img.	No hist.	No judge	Blind	**Ours**
Material cut ↑	35	27	33	16	**40**
Volume-fraction reversals ↓	4	15	12	11	**0**
Mesh refined	4	15	22	0	1
Supports edited ↓	0	13	14	0	**0**
Config. identical to previous ↓	0	0	0	4	**0**
Visual observations ↑	1	14	22	0	**23**
Prior-based reasoning ↓	13	6	5	19	**2**

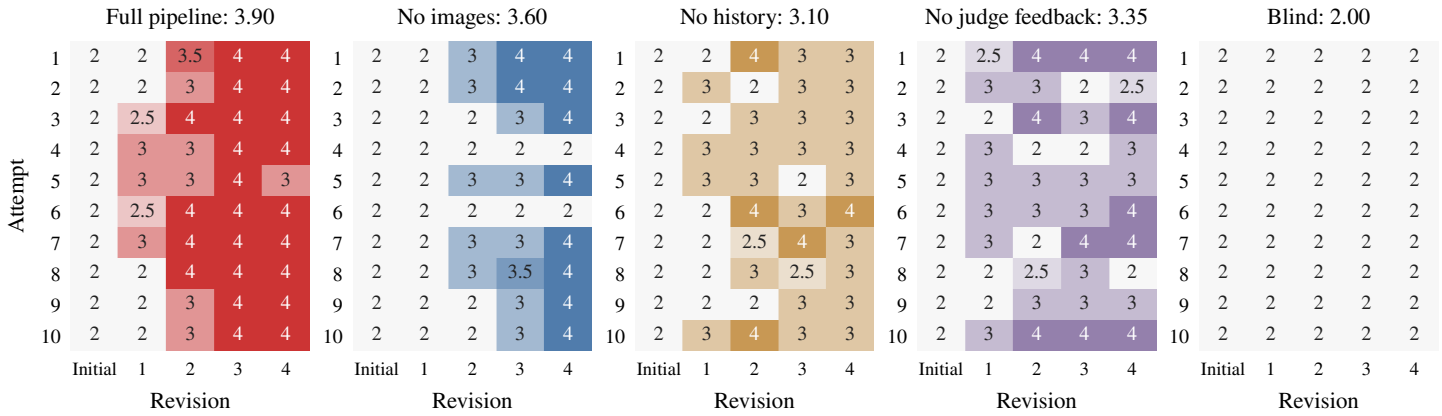


FIGURE 12: Blind judge score of every design in the ablation, by attempt and revision, with the mean final score in each title. Each call gives a whole-number score; a value such as 2.5 is the mean of two calls one point apart. The full pipeline turns uniformly dark by revision 3, no history and no judge feedback scatter, and every blind design scores 2.

cost less than expected, because the judge's written feedback describes the design in enough detail for the agent to infer it, and removing judge feedback cost little in score but much in stability. The two channels play different roles. The history tells the agent which direction it has been moving, and without it the agent undoes its own progress. The judge's feedback tells it when to stop, and without it the agent pursues the look of a tree until the structure disconnects. Together with the blind result, which reproduces our earlier finding that an agent deprived of all feedback cannot improve on the initial design, these results show that the gains of the full pipeline come from the feedback loop and not from the agent's prior knowledge of topology optimization. With ten attempts per condition the smaller effects are not significant, and a larger study, or a judge whose scores are absolute in the loop, would be needed to resolve them.

7 LIMITATIONS AND CONCLUSIONS

We present a framework for studying how agents tune topology optimization parameters toward broad qualitative design goals specified by a human designer, enabling iterative parameter exploration, revision, and cross-agent feedback. Across our case studies, agents demonstrate a strong understanding of parameter effects, occasionally discovering creative multi-parameter strategies that better align with human design preferences, while consistently respecting design constraints so that resulting geometries can be post-processed and manufactured by the downstream manufacturing agent. Score trends and the diversity of final designs further suggest that agents become increasingly exploratory as context accumulates across revisions.

These results should be interpreted as a preliminary investigation rather than a comprehensive evaluation. We do not conduct a controlled human-user study comparing designer performance, effort, or decision-making on the same task, and our evaluation relies on an AI judge whose scores may be sensitive to model choice and prompting configuration; although we adopted a deliberately broad and minimally constrained prompting strategy, prompting is inherently biased and may systematically influence evaluations. System behavior likewise depends on the underlying models, and we evaluate only a single state-of-the-art open-source vision–language model for both the vision and judge agents, leaving systematic comparison with other open- and closed-source alternatives to future work.

Despite these limitations, the framework offers a foundation for deeper study of agentic behavior, improved trust in agent decision-making during autonomous multistep workflows, and insight into how agents integrate multimodal inputs (images of 3D structures for spatial reasoning and long-context conversational history) into engineering decisions. This raises the possibility of human designers operating one level removed from direct parameter control, with natural-language descriptions interpreted by agents and translated into actionable design outcomes. Beyond the present case studies, the approach extends naturally to domains where topology optimization is established but qualitative or human-preference objectives are hard to encode directly: automotive and consumer product design (form, packaging, brand language, manufacturability), architecture and the built environment (lightweight structural systems balancing efficiency with spatial and aesthetic goals), and healthcare (medical devices, assistive technologies, implants, and rehabilitation products, where performance must be weighed against comfort, usability, safety, and patient experience).

Future work includes new feedback-loop architectures that improve inter-agent reasoning and collaboration, strategies for managing growing conversational context to support more reliable retrieval-augmented reasoning across revisions, and inves-

tigation of agents' spatial reasoning together with more efficient 3D representations that vision-based agents can readily interpret. Finally, while our study focuses on specific examples, the framework is built for scalability and can incorporate a broader array of case studies spanning more diverse human-designer descriptions and natural-language instructions.

AUTHOR CONTRIBUTIONS

I.A.S developed the methodology and conducted the experiments. H.C. and I.A.S wrote the manuscript. H.C. contributed to development of experimental design and assisted with prototype production. F.A. supervised and directed the research.

FUNDING

I.A.S. acknowledges that this material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, under Award Number DE-SC0026073. H.C. acknowledges support by Tata Steel.

CODE AND SUPPLEMENTARY FILES

All code, analysis notebooks, and computational protocols developed in this study are publicly available at `https://github.com/bellastewart/TO-Agents`. The repository also contains supplementary materials, including the full agent outputs referenced throughout this manuscript.

COMPETING INTERESTS

The authors declare no competing interests.

REFERENCES

[1] Schiffer, G., Ha, D. Q., and Carstensen, J. V., 2023. "HiTop 2.0: combining topology optimisation with multiple feature size controls and human preferences". *Virtual and Physical Prototyping,* ***18***(1), Dec., p. e2268603.

[2] Mueller, C. T., and Ochsendorf, J. A., 2015. "Combining structural performance and designer preferences in evolutionary design space exploration". *Automation in Construction,* ***52***, Apr., pp. 70–82.

[3] Krish, S., 2011. "A practical generative design method". *Computer-Aided Design,* ***43***(1), Jan., pp. 88–100.

[4] Lazarov, B. S., Wang, F., and Sigmund, O., 2016. "Length scale and manufacturability in density-based topology optimization". *Archive of Applied Mechanics,* ***86***(1-2), Jan., pp. 189–218.

[5] Regenwetter, L., Abu Obaideh, Y., and Ahmed, F., 2025. "Multi-Objective Counterfactuals for Design: A Model-Agnostic Counterfactual Search Method for Multi-Modal Design Modifications". *Journal of Mechanical Design,* ***147***(2), Feb., p. 021401.

[6] Sigmund, O., 2001. "A 99 line topology optimization code written in Matlab". *Structural and Multidisciplinary Optimization,* ***21***(2), Apr., pp. 120–127.

[7] Andreassen, E., Clausen, A., Schevenels, M., Lazarov, B. S., and Sigmund, O., 2011. "Efficient topology optimization in MATLAB using 88 lines of code". *Structural and Multidisciplinary Optimization,* ***43***(1), Jan., pp. 1–16.

[8] Sigmund, O., 2022. "On benchmarking and good scientific practise in topology optimization". *Structural and Multidisciplinary Optimization,* ***65***(11), Nov., p. 315.

[9] Oduguwa, V., Tiwari, A., and Roy, R., 2003. "Handling integrated quantitative and qualitative search space in a real world optimisation problem". In The 2003 Congress on Evolutionary Computation, 2003. CEC '03., Vol. 2, IEEE, pp. 1222–1229.

[10] Brintrup, A. M., Ramsden, J., and Tiwari, A., 2007. "An interactive genetic algorithm-based framework for handling qualitative criteria in design optimization". *Computers in Industry,* ***58***(3), Apr., pp. 279–291.

[11] Zhou, Y., Muresanu, A. I., Han, Z., Paster, K., Pitis, S., Chan, H., and Ba, J., 2022. Large Language Models Are Human-Level Prompt Engineers. Version Number: 2.

[12] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I., 2017. Attention Is All You Need. Version Number: 7.

[13] Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., Liu, P., Nie, J.-Y., and Wen, J.-R., 2023. A Survey of Large Language Models. Version Number: 18.

[14] Wei, J., Tay, Y., Bommasani, R., Raffel, C., Zoph, B., Borgeaud, S., Yogatama, D., Bosma, M., Zhou, D., Metzler, D., Chi, E. H., Hashimoto, T., Vinyals, O., Liang, P., Dean, J., and Fedus, W., 2022. Emergent Abilities of Large Language Models. Version Number: 2.

[15] Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P. S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X. E., Tang, B., Taylor, R., Williams, A., Kuan, J. X.,

Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., and Scialom, T., 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. Version Number: 2.

[16] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., and Zhou, D., 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. Version Number: 6.

[17] Song, Z., Lu, J., Du, Y., Yu, B., Pruyn, T. M., Huang, Y., Guo, K., Luo, X., Qu, Y., Qu, Y., Wang, Y., Wang, H., Guo, J., Gan, J., Shojaee, P., Luo, D., Bran, A. M., Li, G., Zhao, Q., Luo, S.-X. L., Zhang, Y., Zou, X., Zhao, W., Zhang, Y. F., Zhang, W., Zheng, S., Zhang, S., Khan, S. T., Rajabi-Kochi, M., Paradi-Maropakis, S., Baltoiu, T., Xie, F., Chen, T., Huang, K., Luo, W., Fang, M., Yang, X., Cheng, L., He, J., Hassoun, S., Zhang, X., Wang, W., Reddy, C. K., Zhang, C., Zheng, Z., Wang, M., Cong, L., Gomes, C. P., Hsieh, C.-Y., Nandy, A., Schwaller, P., Kulik, H. J., Jia, H., Sun, H., Moosavi, S. M., and Duan, C., 2025. Evaluating Large Language Models in Scientific Discovery. Version Number: 1.

[18] Zhou, Y., Liu, H., Srivastava, T., Mei, H., and Tan, C., 2024. "Hypothesis Generation with Large Language Models". In Proceedings of the 1st Workshop on NLP for Science (NLP4Science), Association for Computational Linguistics, pp. 117–139.

[19] Zhang, Y., Khan, S. A., Mahmud, A., Yang, H., Lavin, A., Levin, M., Frey, J., Dunnmon, J., Evans, J., Bundy, A., Dzeroski, S., Tegner, J., and Zenil, H., 2025. "Exploring the role of large language models in the scientific method: from hypothesis to discovery". *npj Artificial Intelligence,* ***1***(1), Aug., p. 14.

[20] Loos, S., Wolk, S. V. D., Graaf, N. D., Hekkert, P., and Wu, J., 2022. "Towards intentional aesthetics within topology optimization by applying the principle of unity-in-variety". *Structural and Multidisciplinary Optimization,* ***65***(7), July, p. 185.

[21] Saadi, J. I., Chong, L., and Yang, M. C., 2024. "The effect of targeting both quantitative and qualitative objectives in generative design tools on the design outcomes". *Research in Engineering Design,* ***35***(4), Oct., pp. 409–425.

[22] Saadi, J., and Yang, M., 2023. "OBSERVATIONS ON THE IMPLICATIONS OF GENERATIVE DESIGN TOOLS ON DESIGN PROCESS AND DESIGNER BEHAVIOUR". *Proceedings of the Design Society,* ***3***, July, pp. 2805–2814.

[23] Elrefaie, M., Dai, A., and Ahmed, F., 2025. "DrivAerNet: A Parametric Car Dataset for Data-Driven Aerodynamic Design and Prediction". *Journal of Mechanical Design,* ***147***(4), Apr., p. 041712.

[24] Sung, N., Spreizer, S., Elrefaie, M., Samuel, K., Jones, M. C., and Ahmed, F., 2025. "BlendedNet: A Blended Wing Body Aircraft Dataset and Surrogate Model for Aerodynamic Predictions". In Volume 3B: 51st Design Automation Conference (DAC), American Society of Mechanical Engineers, p. V03BT03A049.

[25] Stewart, I. A., and Buehler, M. J., 2026. Higher-Order Knowledge Representations for Agentic Scientific Reasoning, Jan. arXiv:2601.04878 [cs].

[26] Nisa, U., Shirazi, M., Saip, M. A., and Pozi, M. S. M., 2025. "Agentic AI: The age of reasoning—A review". *Journal of Automation and Intelligence*, Aug., p. S2949855425000516.

[27] Chowa, S. S., Alvi, R., Rahman, S. S., Rahman, M. A., Raiaan, M. A. K., Islam, M. R., Hussain, M., and Azam, S., 2025. From Language to Action: A Review of Large Language Models as Autonomous Agents and Tool Users. Version Number: 2.

[28] Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y., 2022. ReAct: Synergizing Reasoning and Acting in Language Models. Version Number: 3.

[29] Wu, Q., Bansal, G., Zhang, J., Wu, Y., Li, B., Zhu, E., Jiang, L., Zhang, X., Zhang, S., Liu, J., Awadallah, A. H., White, R. W., Burger, D., and Wang, C., 2023. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation. Version Number: 2.

[30] Duan, Z., and Wang, J., 2024. Exploration of LLM Multi-Agent Application Implementation Based on LangGraph+CrewAI. Version Number: 1.

[31] Wang, J., and Duan, Z., 2024. Agent AI with LangGraph: A Modular Framework for Enhancing Machine Translation Using Large Language Models. Version Number: 1.

[32] Stewart, I. A., Hage, T. P., Hsu, Y.-C., and Buehler, M. J., 2026. GraphAgents: Knowledge Graph-Guided Agentic AI for Cross-Domain Materials Design. Version Number: 1.

[33] Ghareeb, A. E., Chang, B., Mitchener, L., Yiu, A., Szostkiewicz, C. J., Laurent, J. M., Razzak, M. T., White, A. D., Hinks, M. M., and Rodriques, S. G., 2025. Robin: A multi-agent system for automating scientific discovery. Version Number: 1.

[34] Elrefaie, M., Qian, J., Wu, R., Chen, Q., Dai, A., and Ahmed, F., 2025. "AI Agents in Engineering Design: A Multi-Agent Framework for Aesthetic and Aerodynamic Car Design". In Volume 3B: 51st Design Automation Conference (DAC), American Society of Mechanical Engineers, p. V03BT03A048.

[35] Ni, B., and Buehler, M. J., 2023. MechAgents: Large language model multi-agent collaborations can solve mechanics problems, generate new data, and integrate knowledge. Version Number: 1.

[36] Massoudi, S., and Fuge, M., 2025. "Agentic Large Language Models for Conceptual Systems Engineering and Design". In Volume 3B: 51st Design Automation Conference (DAC), American Society of Mechanical Engineers,

p. V03BT03A045.
[37] Yan, X., Bao, D., Zhou, Y., Xie, Y., and Cui, T., 2022. "Detail control strategies for topology optimization in architectural design and development". *Frontiers of Architectural Research,* ***11***(2), Apr., pp. 340–356.
[38] Meng, X., Zhang, L.-Y., Zhao, Z.-L., and Xie, Y. M., 2023. "A direct approach to achieving efficient free-form shells with embedded geometrical patterns". *Thin-Walled Structures,* ***185***, Apr., p. 110559.
[39] Li, Z., Lee, T.-U., and Xie, Y., 2023. "Topology optimisation considering subjective preferences: current progress and challenges".
[40] Tian, B., Qian, W., and Yang, Y., 2026. "Designer preference-driven topology optimization using a human-in-the-loop neural network". *Computers & Structures,* ***321***, Jan., p. 108070.
[41] Li, Z., Lee, T.-U., and Xie, Y. M., 2025. "Interactive 3D structural design in virtual reality using preference-based topology optimization". *Computer-Aided Design,* ***180***, Mar., p. 103826.
[42] Schiffer, G., Schmidt, M.-P., Pedersen, C. B. W., and Carstensen, J. V., 2024. "Interactive infill topology optimisation guided by user drawn patterns". *Virtual and Physical Prototyping,* ***19***(1), Dec., p. e2361864.
[43] Hekkert, P., et al., 2006. "Design aesthetics: principles of pleasure in design". *Psychology science,* ***48***(2), p. 157.
[44] Ha, D. Q., and Carstensen, J. V., 2023. "Human-Informed Topology Optimization: interactive application of feature size controls". *Structural and Multidisciplinary Optimization,* ***66***(3), Mar., p. 59.
[45] Zhang, W., Wang, Y., Youn, S.-K., and Guo, X., 2024. "Machine learning powered sketch aided design via topology optimization". *Computer Methods in Applied Mechanics and Engineering,* ***419***, Feb., p. 116651.
[46] Feng, S., Caremel, C., and Kawahara, Y., 2026. "Sketch2Topo: Using Hand-Drawn Inputs for Diffusion-Based Topology Optimization". In Proceedings of the Extended Abstracts of the 2026 CHI Conference on Human Factors in Computing Systems, pp. 1–5. arXiv:2603.18960 [cs.HC].
[47] Zhu, S., Hu, J., Qi, J., Wang, L., Guo, J., Ma, J., and Zhu, G., 2025. "Sketch-Guided Topology Optimization with Enhanced Diversity for Innovative Structural Design". *Applied Sciences,* ***15***(5), Mar., p. 2753.
[48] Li, Z., Lee, T.-U., and Xie, Y. M., 2023. "Interactive Structural Topology Optimization with Subjective Scoring and Drawing Systems". *Computer-Aided Design,* ***160***, July, p. 103532.
[49] Su, R., Tangaramvong, S., and Song, C., 2024. "Automatic Image-Based SBFE-BESO Approach for Topology Structural Optimization". *International Journal of Mechanical Sciences,* ***263***, Feb., p. 108773.
[50] Ahnobari, A., 2024. pyfantom. GitHub repository, `https://github.com/ahnobari/pyFANTOM`. Accessed: 2026-05-18.
[51] Colvin, S., Jolibois, E., Ramezani, H., Badaracco, A. G., Dorsey, T., Montague, D., Matveenko, S., Trylesinski, M., Runkle, S., Hewitt, D., and Hall, A., 2024. Pydantic (version 2.9.0). GitHub repository, `https://github.com/pydantic/pydantic`. Accessed: 2026-05-18.
[52] Wataoka, K., Takahashi, T., and Ri, R., 2024. Self-Preference Bias in LLM-as-a-Judge. Version Number: 2.
[53] Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E. P., Zhang, H., Gonzalez, J. E., and Stoica, I., 2023. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. Version Number: 4.
[54] Schroeder, W., Martin, K., and Lorensen, B., 2006. *The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics*. Kitware.
[55] K3d Project, n.d.. K3d: Lightweight wrapper to run k3s in docker. `https://k3d.io/stable/`. Accessed: 2026-05-18.
[56] Bambu Lab, 2026. *Bambu Studio (version 2.5.0)*. Computer software. Available at `https://bambulab.com/en-us/download/studio`, accessed 2026-05-18.
[57] Creusen, M. E. H., and Schoormans, J. P. L., 2005. "The different roles of product appearance in consumer choice". *Journal of Product Innovation Management,* ***22***(1), pp. 63–81.
[58] Crilly, N., Moultrie, J., and Clarkson, P. J., 2004. "Seeing things: consumer response to the visual domain in product design". *Design Studies,* ***25***(6), pp. 547–577.
[59] Desmet, P. M. A., and Hekkert, P., 2007. "Framework of product experience". *International Journal of Design,* ***1***(1), pp. 57–66.
[60] Bloch, P. H., 1995. "Seeking the ideal form: Product design and consumer response". *Journal of Marketing,* ***59***(3), pp. 16–29.
[61] Norman, D. A., 2004. *Emotional Design: Why We Love (or Hate) Everyday Things*. Basic Books, New York.
[62] Blijlevens, J., Creusen, M. E. H., and Schoormans, J. P. L., 2009. "How consumers perceive product appearance: The identification of three product appearance attributes". *International Journal of Design,* ***3***(3), pp. 27–35.
[63] Hsiao, K.-A., and Chen, L.-L., 2006. "Fundamental dimensions of affective responses to product shapes". *International Journal of Industrial Ergonomics,* ***36***(6), pp. 553–564.
[64] Homburg, C., Schwemmle, M., and Kuehnl, C., 2015. "New product design: Concept, measurement, and consequences". *Journal of Marketing,* ***79***(3), pp. 41–56.
[65] Chen, K., and Owen, C. L., 1997. "Form language and style

description". *Design Studies,* ***18***(3), pp. 249–274.
[66] Hekkert, P., Snelders, D., and van Wieringen, P. C. W., 2003. "'Most advanced, yet acceptable': Typicality and novelty as joint predictors of aesthetic preference in industrial design". *British Journal of Psychology,* ***94***(1), pp. 111–124.
[67] Nagamachi, M., 1995. "Kansei engineering: A new ergonomic consumer-oriented technology for product development". *International Journal of Industrial Ergonomics,* ***15***(1), pp. 3–11.
[68] Govers, P. C. M., and Schoormans, J. P. L., 2005. "Product personality and its influence on consumer preference". *Journal of Consumer Marketing,* ***22***(4), pp. 189–197.
[69] Mugge, R., Govers, P. C. M., and Schoormans, J. P. L., 2009. "The development and testing of a product personality scale". *Design Studies,* ***30***(3), pp. 287–302.
[70] Hekkert, P., and Cila, N., 2015. "Handle with care! Why and how designers make use of product metaphors". *Design Studies,* ***40***, pp. 196–217.
[71] DiSalvo, C., and Gemperle, F., 2003. "From seduction to fulfillment: the use of anthropomorphic form in design". In Proceedings of the 2003 International Conference on Designing Pleasurable Products and Interfaces, ACM, pp. 67–72.
[72] Cila, N., Borsboom, F., and Hekkert, P., 2014. "Determinants of aesthetic preference for product metaphors". *Empirical Studies of the Arts,* ***32***(2), pp. 183–203.
[73] Helander, M. G., and Zhang, L., 1997. "Field studies of comfort and discomfort in sitting". *Ergonomics,* ***40***(9), pp. 895–915.
[74] De Looze, M. P., Kuijt-Evers, L. F. M., and Van Dieën, J., 2003. "Sitting comfort and discomfort and the relationships with objective measures". *Ergonomics,* ***46***(10), pp. 985–997.
[75] Vink, P., and Hallbeck, S., 2012. "Editorial: Comfort and discomfort studies demonstrate the need for a new model". *Applied Ergonomics,* ***43***(2), pp. 271–276.
[76] Hassenzahl, M., 2004. "The interplay of beauty, goodness, and usability in interactive products". *Human–Computer Interaction,* ***19***(4), pp. 319–349.

Table of Contents for Appendices

A Spatial Perception Pipeline **23**

B Post-Processing Geometry Rules **24**

C Agentic Pipeline Implementation Details **24**

D Human Designer Requests **24**

E Full Set of Replicate Runs **25**

F Holistic Preference Experiments **30**
F.1 How the agents read the brief and each preference 30
F.2 Parameters adjusted in each category 30

A Spatial Perception Pipeline

An automated pipeline was developed and added to pyFANTOM [50] to generate consistent three-dimensional visualizations of finite element solutions and export high-resolution images from multiple viewpoints. The implementation integrates the Visualization Tool Kit (VTK) [54] for mesh processing, K3D [55] for WebGL-based rendering, and Playwright for automated screenshot capture within an asynchronous Python workflow.

Input data consist of nodal coordinates, element connectivity, nodal forces, boundary constraints, and element density variables from topology optimization. When density variables are provided, elements with density $\rho < 0.5$ are excluded from visualization. For multi-material cases, elements are separated into material subsets and rendered as independent meshes.

Finite element geometry is first converted to a VTK unstructured grid, where elements are represented as tetrahedral or hexahedral cells. A surface extraction filter (vtkDataSetSurfaceFilter) is applied to obtain a triangular surface mesh suitable for real-time rendering. The resulting vertices and connectivity are transferred to K3D mesh objects.

External forces and boundary conditions are visualized using glyphs generated with vtkGlyph3D. Force vectors are represented as arrow glyphs scaled relative to the approximate element size, while boundary conditions are visualized using directional constraint markers indicating fixed or free degrees of freedom.

To enhance spatial perception in perspective views, a depth-based color encoding is applied. For each camera configuration, vertex depths are computed along the camera viewing direction and normalized. These values are mapped using the Viridis color map so that surfaces closer to the camera appear brighter than distant regions.

Six orthogonal camera views (top, bottom, left, right, front, and back) are automatically defined based on the model bounding box. For each view, the camera position and orientation are set relative to the model center and the scene is exported as a standalone HTML visualization. High-resolution figures are generated using the Playwright browser automation framework: a headless Chromium instance loads each HTML file, renders the WebGL canvas, and captures a screenshot of the visualization at a resolution of 1920×1080 pixels. This automated workflow ensures consistent rendering parameters and reproducible visualization outputs across all generated images which are fed as context to the vision agent.

B Post-Processing Geometry Rules

Following topology optimization, the optimized density volume x_{Phys} is modified through a set of deterministic geometric operations that enforce manufacturable support structures for the phone-stand case study. First, remaining voxels located above a prescribed diagonal plane are removed to generate the characteristic inclined surface used to support the device. A diagonal reinforcement band of fixed thickness is then inserted along this plane. To prevent sliding, an anti-slip lip is added at the lower front edge of the model by enforcing solid material within a rectangular region defined by the lip thickness and height parameters. Additionally, a solid base layer of fixed thickness is imposed along the bottom of the volume to increase stability and provide a flat contact surface for printing. Support thickness is prescribed relative to the mesh resolution selected by the agents.

For visualization and fabrication, the processed density field is converted to a polygonal surface mesh using the marching cubes algorithm. The binary volume is first padded and reoriented to ensure consistent surface extraction. An isosurface is then generated at a density threshold, producing vertices, faces, and surface normals. The resulting triangular mesh is rendered. During production, this mesh is exported in OBJ format for CAD inspection and downstream additive manufacturing.

C Agentic Pipeline Implementation Details

This section expands on the pipeline summarized in Section 3 and Algorithm 1.

Case Studies. We evaluate the pipeline through two problem formulations. The first is the classic cantilever beam problem, commonly used as a benchmark for evaluating new TO methods [8]. The second is a phone stand designed to hold a smartphone on a desk, representing a real-world product for which human preferences and aesthetic qualities play a prominent role. Both use simple boundary conditions and loading scenarios; their complete problem formulations appear in Appendix D, Figures 13 and 14. We assume the designer provides a verbose technical natural-language description so that all inputs required for the topology optimization are specified. Future work will investigate how simplifying this description impacts the process.

Formulation Agent. Pydantic [51] is a Python-based validation framework linked to the agentic framework. The formulation agent uses it to transform the designer's description into validated, structured JSON matching the solver's expected input format, providing a reliable interface for passing the problem formulation to the solver.

Solver Choice. We selected pyFANTOM [50] because it avoids the black-box constraints of proprietary software, giving full visibility into the parameters the agent can access and adjust to influence internal algorithmic behavior. The agent can tune parameters throughout a Python-native, scalable, multi-constraint topology optimization framework that supports both structured and unstructured meshes as well as a wide range of topology optimization objective functions.

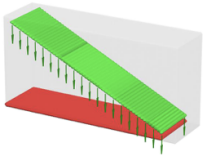
Vision Agent Reasoning. The vision agent implements the designer's feedback through a three-stage process. First, it analyzes the full chat history, including the previous solver parameters and the rendered 3D images of prior structures. Second, it evaluates the designer's request alongside the solver parameters available for modification. Third, it determines which parameters should be adjusted and to what extent. The agent may change any and all parameters used to run the optimizer. It outputs detailed natural-language reasoning identifying which parameters should change and why those modifications are expected to better satisfy the designer's intent. Because the agent receives no explicit parameter definitions or prior guidance on parameter effects, it relies entirely on knowledge acquired during foundational training to infer the meaning and relative sensitivity of these parameters, making the multi-objective reasoning process autonomous. Its suggestions are converted into structured output and passed back to the topology optimization agent for a revised solver run, after which the resulting topology is again rendered and saved from multiple viewpoints. The prompt that drives this workflow is reproduced verbatim in the supplementary prompt document.

Judge Agent and History Conditioning. The AI judge evaluates both the initial and revised designs, scoring whether the designer's criterion of a more branching, intricate structure has been achieved. Scoring criteria and the prompt are reproduced verbatim in the supplementary prompt document. The judge assigns a score from 1 to 5 to each design after all revisions complete. We intentionally power the judge with a larger language model trained independently from the vision–language model used by the vision agent, which promotes a more neutral and unbiased evaluation. The judge's feedback is added to the conversation history and provided to the vision agent for the next revision. At that point the vision agent receives a richer context: the judge's evaluation, the images of all topology optimization results, and their corresponding solver parameters. As the revision history expands, the vision agent is instructed to use the most recent highest-scoring design as the basis for its next revision while continuing to align with the designer's request. The maximum number of iterations can be specified by the engineer; for interpretability we limit the process to five total iterations, one original optimization followed by four revision cycles, replicated over 10 trials per case study under identical settings.

Production Hardware. The final geometry is sent to Bambu Studio [56] for slicing, then 3D printed in standard PLA using a Bambu Lab X1C printer.

D Human Designer Requests

The two case studies are initiated by a natural-language problem description authored by the human designer and sup-



Case Study: Phone-Stand Problem

Human Designer Request
Phone-Stand Problem

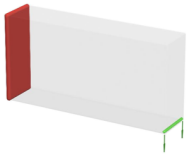
"In this setup, the finite element analysis (FEA) uses linear elasticity physics with a Young's modulus $E = 1.0$ and a Poisson's ratio $\nu = 0.3$. A three-dimensional structured mesh is defined with $n_x = 128$, $n_y = 64$, $n_z = 16$ elements and physical dimensions $l_x = 1.0$, $l_y = 0.5$, $l_z = 0.125$. The mesh is linked to the linear elastic physics so the material properties apply uniformly throughout the domain. A multigrid solver is configured with tol $= 1e{-}4$, maxiter $=$ 50, and n_level $= 5$. **One Dirichlet boundary condition is imposed: the bottom face ($y = 0$), where all nodes are fixed in the u_x, u_y, and u_z directions. One force is applied along a diagonal: with a force of -1.0 in the y-direction distributed among all nodes.** Use a density filter with radius 1.5. Set up a minimum compliance topology optimization problem with a volume fraction of 15%, SIMP penalty of 3.0, void material stiffness of $1e{-}9$, and enable Heaviside projection. Use the PGD optimizer with a function tolerance of $1e{-}4$ and no change tolerance."

FIGURE 13: Phone-stand case study: the human designer specifies the FEA setup, boundary conditions (red), applied loads (green), and TO parameters as a single natural-language request.

plied to the agentic pipeline as the starting context. We render these requests in a distinct blue style to visually separate human-authored input from agent-generated prompts elsewhere in the appendix. Within each request, boundary conditions are highlighted in **red** and applied loads in **green**, matching the corresponding setup figure.

E Full Set of Replicate Runs

Figures 15 and 16 report the phone-stand replicates, separated into successful and failure cases. Figures 17 and 18 report the corresponding cantilever-beam replicates.



Case Study: Cantilever Beam Problem

Human Designer Request
Cantilever Beam Problem

"In this setup, the finite element analysis (FEA) uses linear elasticity physics with a Young's modulus $E = 1.0$ and a Poisson's ratio $\nu = 0.3$. A three-dimensional structured mesh is defined with $n_x = 128$, $n_y = 64$, $n_z = 64$ elements and physical dimensions $l_x = 1.0$, $l_y = 0.5$, $l_z = 0.5$. The mesh is linked to the linear elastic physics so the material properties apply uniformly throughout the domain. A multigrid solver is configured with tol $= 1e{-}4$, maxiter $=$ 200, and n_level $= 5$. **Two Dirichlet boundary conditions are imposed: (1) Left face: nodes where $x = 0$ are fixed in the u_x and u_z directions while u_y remains free, with zero displacement. (2) Right-bottom corner: nodes where $x = 1.0$ and $y = 0$ are fixed in the u_y and u_z directions while u_x remains free, with zero displacement. One point force is applied: nodes where $x = 0$ and $y = 0.5$ have a force of -1.0 in the y-direction distributed among all nodes.** Use a density filter with radius 1.5. Set up a minimum compliance topology optimization problem with a volume fraction of 40%, SIMP penalty of 3.0, void material stiffness of $1e{-}9$, and enable Heaviside projection. Use the PGD optimizer with a function tolerance of $1e{-}4$ and no change tolerance."

FIGURE 14: Cantilever-beam case study: the human designer specifies the FEA setup, boundary conditions (red), applied loads (green), and TO parameters as a single natural-language request.



FIGURE 15: Successful Replicates. The set of replicates in which at least one design in the sequence aligns with the human designer's request for a branching, tree-like skeletal structure. Replicates A through F correspond to separate runs of the same pipeline in the phone-stand case study, where each sequence of revisions reflects the outcomes produced by passing through the agentic pipeline. The score from the AI judge is presented below each design where scores are made relative to the other revisions within the same replicate.

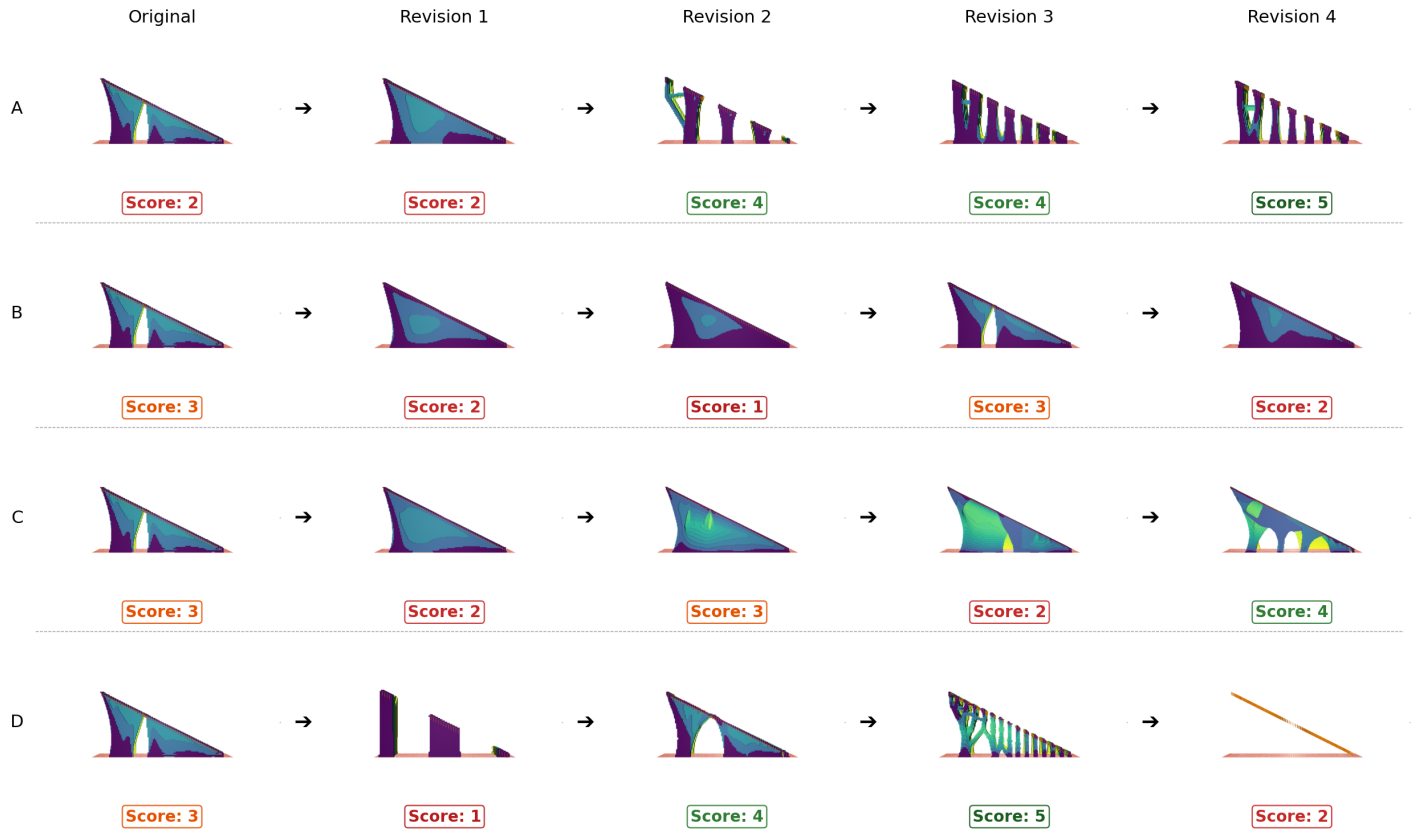


FIGURE 16: Failure Replicates. The set of replicates in which none of the designs in the sequence align with the human designer's request for a branching, tree-like skeletal structure. Replicates A through D correspond to separate failure cases in the phone-stand case study, where each sequence of revisions reflects the outcomes produced by passing through the agentic pipeline. The score from the AI judge is presented below each design where scores are made relative to the other revisions within the same replicate.



FIGURE 17: Successful Replicates. In the cantilever beam case study, we focus on the set of replicates in which at least one design in the sequence aligns with the human designer's request for a branching, tree-like skeletal structure. Replicates A through F correspond to separate runs of the process, where each sequence of revisions reflects the outcomes produced by passing through the agentic pipeline. The score from the AI judge is presented below each design where scores are made relative to the other revisions within the same replicate.



FIGURE 18: Failure Replicates. In the cantilever beam case study, we focus on the set of replicates in which none of the designs in the sequence align with the human designer's request for a branching, tree-like skeletal structure. Replicates A through D correspond to separate failure cases of the process, where each sequence of revisions reflects the outcomes produced by passing through the agentic pipeline. The score from the AI judge is presented below each design where scores are made relative to the other revisions within the same replicate.

F Holistic Preference Experiments

To develop concrete preference categories, we drew on widely cited and accepted research frameworks in consumer behavior, design, ergonomics, and Kansei engineering concerning how people perceive and use physical products.

These fields distinguish responses to a product's appearance, what its form communicates, how it feels in use, and the emotions it evokes. For example, in interviews with 142 consumers, Creusen and Schoormans [57] found that participants read a durable product's appearance for aesthetic, symbolic, ergonomic, and functional information. Crilly et al. [58] distinguish aesthetic impression, semantic interpretation, and symbolic association; Desmet and Hekkert [59] distinguish sensory pleasure, meaning, and emotion. Bloch [60] separates aesthetic from utilitarian value and identifies ergonomic requirements as a distinct constraint, while Norman [61] distinguishes responses to appearance, use and physical feel, and what a form means or evokes.

These recurring distinctions motivate our categorization of design preferences into four groupings: stylistic, affective, analogical and ergonomic categories. The grouping also draws on empirical work. Blijlevens et al. [62] studied how consumers sort and rate product appearances; Hsiao and Chen [63] analyzed semantic-differential ratings; and Homburg et al. [64] validated a consumer response scale.

Stylistic is the aesthetic role: where a form stands relative to the conventions of its kind, in composition as well as in idiom. It covers period and cultural labels such as *modern* or *brutalist*, compositional words such as *clean*, *balanced* or *ornate*, and novelty. Consumers place forms spontaneously on modernity, simplicity and playfulness [62], a style can be described as a form language on polar opposite adjective scales independent of what the form expresses [65], aesthetic preference is predicted jointly by typicality and novelty relative to the category [66], and unity in variety is its compositional principle [43]. It is separable from expressed character: Hsiao and Chen's trendiness factor (traditional–modern) loads apart from their emotion factor and tracks prototypicality rather than curvature [63], and Crilly et al. treat style as a cultural moderator of response rather than as expression [58].

Affective is the semantic role that describes the character a product conveys through its form. Kansei engineering studies such impressions using adjective pairs [67]. Hsiao and Chen [63], for example, identify an emotion factor associated with contrasts such as soft–hard and feminine–masculine, which they relate to curved and straight geometry. Consumers also infer personality traits from product appearance and may prefer products whose perceived personality matches their own [68, 69]. Desmet and Hekkert [59] classify preferences like *elegant* as a judgment of meaning, distinct from sensory pleasure or a felt emotion. Thus, this category concerns the impression a form creates, without prescribing one particular shape or ornament.

Analogical is the symbolic role that asks a product to resemble something outside its own category. Design research calls this a *product metaphor*: a form that refers to properties of another entity [70]. Such references are often anthropomorphic [71], and their success depends in part on whether people can recognize and understand the reference [72]. Consumer frameworks discuss these references under symbolism, categorization, or meaning [58, 57, 59]. But, we distinguish analogical preferences because they specify an external referent that the design should evoke. Although that resemblance contributes to the product's meaning, evaluating the preference requires judging whether the particular referent remains recognizable.

Ergonomic is the only role tied to the body. This is a distinct role or motive in Creusen and Schoormans and Blijlevens et al., Norman's behavioral level. Comfort and discomfort are independent constructs [73], with discomfort tied to pressure and posture and comfort to well-being and appearance [74, 75], and pragmatic quality is independent of hedonic quality and beauty [76]. A request such as pressure relief therefore has a physical component the other categories lack and a perceived component they share.

We establish the framework on these four as the categories of preference with functional preference as the performance objective.

F.1 How the agents read the brief and each preference

Tables 10 and 11 extend Tables 5 and 6. Table 10 groups the decisions in the formulation agent's briefs for the bike saddle, the bookend and the stool, and in two briefs it wrote for a phone stand from the bare request, by habit. Table 11 lists every reading of a preference that recurs across attempts, with the number of the ten attempts that share it, and the judge's criterion for each category. Quotes are verbatim. The load magnitudes in the briefs were not always carried into the configuration (Section 4.2). For the stool, four legs, a flat seat and an open space under the seat are already asked for by the brief, so they are not readings of "minimalist".

F.2 Parameters adjusted in each category

Tables 12 to 16 list what the reviser changed in each category. The first summarizes the outcomes of all four categories; the remaining four list the most adjusted parameters of each category, up to ten, with how often each was changed, in how many attempts, the mean relative change per adjustment, and its mean value in the initial design and at the last revision. The stool and the bookend changed only five parameters in all; the saddle changed far more, almost all of them bounds of load patches.

TABLE 10: Habits of the formulation agent in its briefs for the three experiments it formulated and in two briefs for a phone stand written from the bare request, with examples. Quotes are verbatim.

Habit	Product	Formulation agent
Turns the use case into supports	Stool	"Supports: four separate foot patches at the corners of y = 0, each 0.08 x 0.08 and fixed in all directions. That puts the load into four legs."
	Bike saddle	"This is where the seatpost clamp grips the rails. Fixing all three directions stops rigid-body motion, and the patch sits slightly behind centre because most of the load is at the rear."
Keeps space for the user and neighbouring objects	Bookend	"…the book space is locked void, so the optimizer cannot put a buttress where the books go."
	Bike saddle	"Minimum compliance would otherwise fill these spaces with side bracing, and the saddle would rub the thighs."
	Phone stand	"…the volume the tilted phone and the user's line of sight occupy in front of the backrest; it must stay open for the phone to sit in the stand"
Sizes the problem to the real product	Bike saddle	"This matches a real saddle of about 275 x 50 x 140 mm, so there is little empty air in the box."
	Bookend	"Domain 0.5 x 0.8 x 0.5 fits a typical bookend, about 15 cm tall and 12 cm wide and deep."
	Bookend	"The load goes on a small contact patch 0.55-0.65 high on the face at x $\approx$ 0.36, so the forced-solid block it creates stays the size of a real contact area."
Predicts what minimum compliance gets wrong	Stool	"Without the void, minimum compliance would likely put a pedestal or central cone straight under the load. With the middle cleared, the result has to be a seat on corner legs, which is a stool."

TABLE 11: Recurring readings of each preference by the vision agent, with the number of the ten attempts that share each, an example (attempt in parentheses), and the judge's criterion over its 40 calls.

Reading	Attempts	Example
Phone stand, "more branching tree-like architecture"		
A tree gathers a spread load into a few paths to the ground	10	"…has to be gathered, step by step, into a few thin members going down to the support." (2)
One fork is not a tree	10	"…a trunk with two branches, not a tree with branches of its own." (5)
The clamped base works against the tree	9	"Known limitation: the whole bottom surface is fixed, so the optimizer tends to prefer many separate columns over a few trunks that fork." (9)
Struts merge into fewer, thicker trunks	5	"The optimizer then collects the spread-out diagonal load through many small struts that merge into fewer, thicker trunks toward the fixed base. That merging pattern is the tree." (8)
Judge	40 calls	slender trunks dividing into spreading branches across the span, on a coherent object; hierarchy named in 3 calls
Bike saddle, "relieving central pressure on nerves"		
Material follows load; the initial design is stiffest under the soft tissue	10 / 7	"So the stiffest, most load-bearing line on the saddle runs straight under the soft tissue that should be protected." (6)
Move the weight onto the sit bones	8	"This matches the goal physically, since more weight on the sit bones means less on the soft tissue in the centre." (1)
Openings the rider never touches do not count	8	"The only openings are small side windows in the support underneath, and a rider never touches those." (4)
A compliant centre instead of a cut-out, later abandoned	4	"A small load keeps a thin, flexible layer across the channel. The surface stays continuous, but it is compliant exactly where the nerves are." (3)
The clamp explains the last solid bridge	9	"The optimiser keeps it because the surface directly above the clamp is the cheapest place for that tie." (10)
Judge	40 calls	visible removal of contact area along the centre line with the sides retained; never load or stiffness
Bookend, "more elegant"		
Fewer members; idle braces are clutter	10	"The mid-height horizontal rod and the thin X-diagonals look like scaffolding." (2)
Members of consistent weight	4	"Elegant structures usually have a few members of consistent weight, and this one doesn't." (9)
The opening is part of the elegance	5	"The open space is part of what makes it look elegant." (2)
Nothing that looks accidental	6	"It looks like a leftover artifact rather than a deliberate design feature." (3)
The filter radius $r_{\min}$ is the main lever	10	"This is where elegance came from." (5, on the larger filter)
Judge	40 calls	slender, flowing and consistent across the views while still usable; penalizes competing braces; never stress
Stool, "minimalist"		
Plate plus four legs, one strut per leg	10	"It is thin, but it is a second structural idea alongside 'plate plus four legs'." (5)
Idle material is not minimalist	10	"Much of the material is doing very little, which is the opposite of minimalist." (10)
Extra members add complexity, not strength	10	"The struts are all working and not overbuilt, so the extra members are adding visual complexity, not strength." (5)
Fewer, thicker members are acceptable	4	"…the design settled into a few thick, clean struts rather than many thin ones." (1)
Too thin becomes a lattice	3	"I would stop there, because below that the thin members start to look like a lattice rather than a minimal stool." (8)
Judge	40 calls	less bulk at legs and joints on a recognizable, usable object; praised the initial design's open arches in eight attempts

TABLE 12: Outcomes per category over the same attempts: judge scores on the 1 to 5 scale, how often the final revision was the best and how often the judge's scores rose one step per revision, the change in compliance from the initial design to the final one, how many parameters the reviser changed per turn, how much of each solve's iteration budget was used and how often the solver stopped on its convergence test.

	Stylistic stool	Affective bookend	Ergonomic bike saddle	Analogical phone stand
Attempts	10	10	10	10
Judge score of the initial design, mean	1.0	1.0	1.2	1.0
Best judge score, mean	5.0	5.0	5.0	5.0
Judge score at the final revision, mean	5.0	4.1	4.9	5.0
Attempts whose final revision scored best	10/10	6/10	9/10	10/10
Attempts scored 1, 2, 3, 4, 5 in order	10/10	5/10	7/10	10/10
Compliance, final over initial, median	22.32x	2.36x	1.66x	12.01x
Parameters changed per reviser turn, mean	3.3	3.7	15.6	4.2
Solver iterations used, share of budget, mean	60%	93%	59%	94%
Solves stopped by convergence	70%	30%	94%	28%

TABLE 13: Stylistic: stool, "minimalist". The 5 most adjusted parameters over the first 10 attempts, 10 attempts by 4 revisions, so 40 reviser turns; 5 parameters were changed at least once. Times adjusted counts revision-by-attempt pairs; mean change is the mean relative change per adjustment; initial and final value are the mean across attempts in the initial design and at the last revision.

Parameter	Times adjusted	Attempts	Mean change	Initial value	Final value
iteration budget	40	10/10	+33%	100	305
volume fraction f	40	10/10	-18%	0.12	0.05
filter radius $r_{\min}$	39	10/10	+27%	2	5.1
SIMP penalty p	7	3/10	+17%	3	3.4
function tolerance	6	6/10	-90%	1.0e-4	4.6e-5

TABLE 14: Affective: bookend, "more elegant". The 5 most adjusted parameters over the first 10 attempts, 10 attempts by 4 revisions, so 40 reviser turns; 5 parameters were changed at least once. Times adjusted counts revision-by-attempt pairs; mean change is the mean relative change per adjustment; initial and final value are the mean across attempts in the initial design and at the last revision.

Parameter	Times adjusted	Attempts	Mean change	Initial value	Final value
filter radius $r_{\min}$	40	10/10	+30%	1.5	4
iteration budget	40	10/10	+32%	100	290
volume fraction f	38	10/10	-5%	0.12	0.1
function tolerance	24	10/10	-41%	1.0e-4	2.2e-5
penalty p continuation	6	2/10	–	–	–

TABLE 15: Ergonomic: bike saddle, "relieving central pressure on nerves". The 10 most adjusted parameters over the first 10 attempts, 10 attempts by 4 revisions, so 40 reviser turns; 90 parameters were changed at least once. Times adjusted counts revision-by-attempt pairs; mean change is the mean relative change per adjustment; initial and final value are the mean across attempts in the initial design and at the last revision. Loads are numbered in the order they appear in the configuration; the reviser split the original loads into several patches, so a load with no initial value did not exist in the initial design.

Parameter	Times adjusted	Attempts	Mean change	Initial value	Final value
volume fraction f	34	10/10	-5%	0.15	0.12
load 1 region, width upper limit	33	10/10	-23%	0.48	0.16
load 2 region, width lower limit	33	10/10	+24%	0.18	0.34
load 3 region, width upper limit	32	10/10	-11%	–	0.21
load 4 region, width lower limit	30	10/10	+15%	–	0.29
load 4 region, length upper limit	24	10/10	+15%	–	0.56
load 4 region, width upper limit	23	10/10	+16%	–	0.33
load 3 region, length upper limit	21	10/10	-9%	–	0.51
filter radius $r_{\min}$	20	10/10	-10%	2	1.6
load 3, magnitude along the height	20	10/10	-30%	–	-0.29

TABLE 16: Analogical: phone stand, "more branching tree-like architecture". The 10 most adjusted parameters over the first 10 attempts, 10 attempts by 4 revisions, so 40 reviser turns; 10 parameters were changed at least once. Times adjusted counts revision-by-attempt pairs; mean change is the mean relative change per adjustment; initial and final value are the mean across attempts in the initial design and at the last revision.

Parameter	Times adjusted	Attempts	Mean change	Initial value	Final value
iteration budget	40	10/10	+64%	100	659
volume fraction f	40	10/10	-27%	0.15	0.04
function tolerance	34	10/10	-87%	1.0e-4	3.2e-7
SIMP penalty p	30	10/10	+21%	3	5.4
filter radius $r_{\min}$	11	10/10	-23%	2	1.5
penalty p continuation	11	3/10	–	–	–
mesh elements along the length	1	1/10	+50%	128	134.4
mesh elements along the height	1	1/10	+50%	64	67.2
mesh elements along the width	1	1/10	+50%	16	16.8
multigrid levels	1	1/10	-20%	5	4.9