GENErator: Supporting Design Style Exploration with Generative AI through Genetic Mixing

David Chuan-En Lin
Carnegie Mellon University
Pittsburgh, PA, USA
chuanenl@cs.cmu.edu

Yaqing Yang
Carnegie Mellon University
Pittsburgh, PA, USA
yaqingyy@cs.cmu.edu

Vikram Mohanty
Carnegie Mellon University
Pittsburgh, PA, USA
vikrammohanty@acm.org

Matthew K. Hong
Toyota Research Institute
Los Altos, CA, USA
matt.hong@tri.global

Yan-Ying Chen
Toyota Research Institute
Los Altos, CA, USA
yan-ying.chen@tri.global

Aniket Kittur
Carnegie Mellon University
Pittsburgh, PA, USA
nkittur@cs.cmu.edu

Nikolas Martelaro
Carnegie Mellon University
Pittsburgh, PA, USA
nikmart@cmu.edu

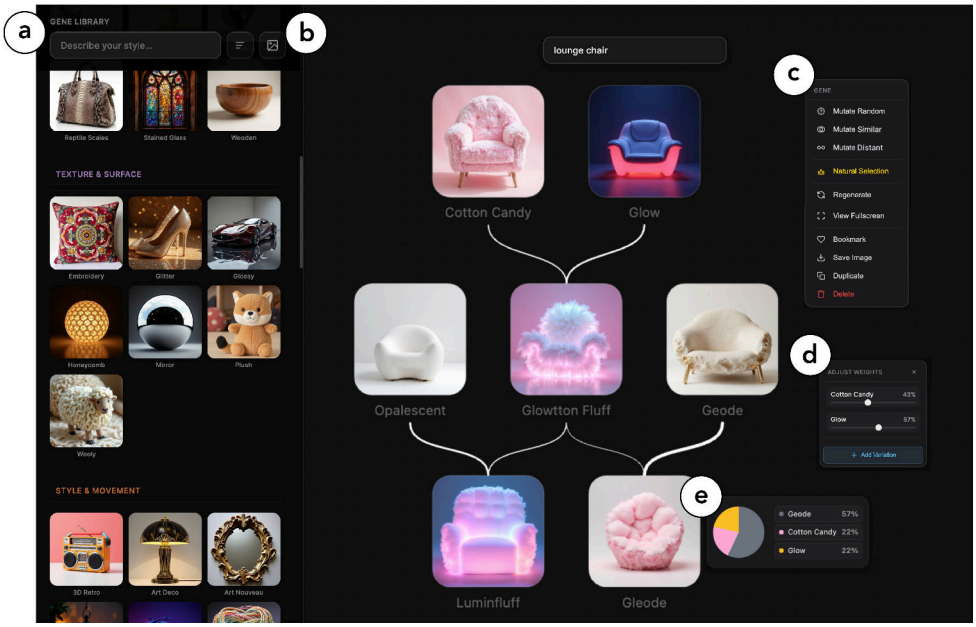


Figure 1: The GENErator interface. (a) The designer can browse and semantically search for style genes in the gene library. (b) Genes can be dragged onto the canvas and mixed by dragging one over another. For example, crossing `Cotton Candy` and `Glow` produces the offspring `Glowtton Fluff`. (c) Right-clicking a gene opens a context menu of available genetic operations. Similar mutation adds `Opalescent` (nearby aesthetic) and distant mutation adds `Geode` (unrelated aesthetic). (d) The designer can adjust the weight contribution of each parent, also visualized as branch thickness in the evolution tree. (e) Hovering over a hybrid gene reveals its genetic composition.

Abstract

Generative AI models can now be fine-tuned into thousands of visual styles, creating large design spaces for product designers. However, current tools offer little support for navigating and combining these styles: designers resort to keyword search and opaque weight sliders, often fixating on familiar styles rather than exploring the large combinatorial space. To address these challenges, we developed GENErator, a proof-of-concept tool that treats each style as a "gene" in a latent style space, enabling genetic operations (crossover, mutation, and natural selection) for guided style exploration with traceable evolution trees. In a within-subjects study comparing GENErator to a standard multi-style slider-based interface, we found that GENErator enabled significantly higher

exploration of unique styles, led designers to keep more of the designs they generated, improved their understanding of how styles relate, and supported a more intentional and additive exploration process.

CCS Concepts

• **Human-centered computing** → **Human computer interaction (HCI)**; **Interactive systems and tools**.

Keywords

generative AI, style composition, design fixation, design space exploration

1 Introduction

Generative AI models can now be fine-tuned into thousands of style-specific adapters, each encoding a distinct visual aesthetic. Low-Rank Adaptation (LoRA) models [31], in particular, have become a popular way for creators to capture and share visual styles (from Art Deco to Brutalist to Vaporwave), with platforms like CivitAI hosting over 100,000 such models. Recent advances in multi-LoRA composition have shown that these adapters can be smoothly blended [79] and combined [52], creating a combinatorial design space of significant scale. Product designers have begun experimenting with these adapters for concept ideation, surface material exploration, and moodboard creation [40, 65], applying visual styles to objects like sneakers, handbags, furniture, and vehicles.

Despite these possibilities, designers face a vast and largely opaque design space. With tens of thousands of style adapters available, and the combinatorial possibilities of blending them, the space of achievable styles far exceeds what any designer can explore manually. Yet current interfaces offer little support for navigation. The typical workflow involves either browsing large lists of style adapters or searching for style adapters by keyword and selecting them by name. Designers then adjust numerical weight sliders, generate an image, and repeat. The style adapter search process is vocabulary-bound: designers can only find styles they already know how to name. It is also opaque: slider values bear little meaningful relationship to visual outcomes, and it is challenging to see how styles relate to one another. Without visibility into the structure of the style space, exploration defaults to trial-and-error, a pattern well-documented to cause design fixation [33]. In our pilot sessions with professional designers from a major automotive company, we observed this firsthand: designers reported that with existing tools and interfaces, they '*can't remember how [they] got there*' when a good combination emerged, and that without structured operations, '*you're not going to use 99% of the library.*' Recent empirical findings confirm that text-to-image tools can actually increase rather than decrease fixation through prompt anchoring and output attachment [73], and that text-based prompting restricts creative exploration, especially for novices [15].

Prior work has explored ways to address AI design fixation through methods such as keyword recombination [12], sketch-based interaction [42], cross-domain analogy [34], and graphical style manipulation [15]. However, these tools address fixation at the level of input modality or ideation process. There remains limited support for exploring the *style space itself*: discovering which styles are available, understanding how they relate, and combining them into new directions. Interactive evolutionary computation (IEC) offered an early vision for this kind of exploration [64], allowing users to guide aesthetic search by selecting preferred candidates rather than describing them. But random mutation causes user fatigue, as most variations keep little meaningful relationship to the selected parent [68, 77]. Meanwhile, LoRA composition enabled multi-style blending [21, 50, 52, 79], but current interfaces, such as CivitAI or Automatic1111, still present designers with sliders and text fields, leaving the style space a black box with little visibility into how styles relate or where a combination sits in the space. Slider-based interfaces require designers to adjust opaque weights, which tends to keep results within familiar aesthetic territory, while evolutionary interfaces generate candidates through random mutation, mostly unrelated to the direction the designer intends. Hence, there remains an interesting, untapped area for exploration – supporting style exploration that is both broad and steerable. We refer to this middle ground as *guided divergence*: the tool should be able to let designers explore unfamiliar styles, but draw them from a region of the style space the designer intentionally chose (e.g., near the current design, far from it, or near a described target).

To address these challenges, we developed GENErator, a proof-of-concept tool that treats each visual style as a "gene" in a navigable latent style space. GENErator supports the early phase of product design, where designers generate diverse directions rather than refine a single concept. Designers discover styles by describing aesthetics, breed styles to produce hybrid offspring, mutate styles towards similar or distant regions of the space, and perform natural selection to automatically evolve a population toward a target aesthetic. An evolution tree makes the full lineage of every style combination transparent and traceable. These operations are made possible by style embeddings that isolates a style direction and gives every gene a location in a latent style space. Rather than adjusting opaque parameters, designers express intent through aesthetic relationships. For example, a designer may blend two genes, mutate toward a nearby aesthetic, mutate toward a distant one, or evolve a population toward a descriptive keyword, such as "luxurious." To the best of our knowledge, GENErator uniquely combines semantically guided operations, multi-style composition, and traceable lineage tracking (Table 1). In a within-subjects study (N=12), we compared GENErator to a standard multi-style slider-based interface where designers select styles by name and adjust numerical weight sliders. Participants using GENErator explored 2.2 times more unique styles, bookmarked 5.2 times more designs (meaning they liked more of the generated designs), reported significantly improved understanding of style relationships (d_z=1.66) (which they attributed to trying bolder combinations), and shifted from adjusting weight sliders to creating new styles with the genetic operations. Example designs created by participants are available at https://style-generator.github.io.

This research thus contributes:

- **GENErator**, a proof-of-concept design exploration tool that supports steerable style exploration with genetic operations. Designers can crossover, mutate, and automatically evolve styles in a shared latent style space, with an evolution tree that records traceable compositional lineages of designs.

- A **within-subjects evaluation** (N=12) comparing GENErator to a standard multi-style slider-based interface, showing significantly broader design space exploration (2.2 times more unique styles while exploring 1.7 times more styles within the region of the space they covered), more designs participants wanted to keep (5.2 times more bookmarked designs), and improved understanding of how styles relate to one another. Behavioral data point to a shift from trial-and-error tweaking of opaque parametric controls towards a seemingly more intentional and additive exploration process.

2 Related Work

GENErator builds on prior research in design fixation in AI-assisted design, style representation and composition in generative AI, and interactive evolutionary computation. Table 1 compares GENErator against the most closely related tools for AI-assisted style exploration.

2.1 Design Fixation in AI-Assisted Design

Design fixation, the tendency to remain stuck exploring a limited set of ideas [33], has become a growing concern as generative AI tools enter design practice. Wadinambiarachchi et al. [73] found that text-to-image (T2I) tools can increase fixation through anchoring to effective prompts and fixating on AI outputs rather than exploring alternatives. Doshi and Hauser [17] report that while AI improves individual creativity, it reduces collective diversity. Davis et al. [15] found that graphical interfaces outperform text prompting for creative exploration.

Several recent systems have proposed approaches for reducing fixation. Some change the input modality. For example, Inkspire [42] replaces text prompting with a sketch-to-design-to-sketch feedback loop and analogical inspiration. Others restructure the exploration process. Luminate [66] lets users vary LLM outputs along interpretable dimensions, CreativeConnect [12] supports visual reference recombination through keyword extraction, and DesignWeaver [69] scaffolds T2I product design with named dimensions extracted from generated designs. Concurrently, POET [27] addresses output homogeneity in text-to-image generation by automatically discovering and expanding dimensions of variation. Each of these tools addresses fixation at the level of input modality, ideation process, or output diversification. In this work, we aim to operate directly on the style space itself, making its structure visible and trackable, and allowing designers to compose, mutate, and evolve styles within it.

A complementary line of work has explored how making generative spaces visible and navigable supports creative exploration. Design Galleries [48] pioneered this approach for graphics parameter spaces, precomputing a dispersed set of renderings so that users could browse alternatives side by side rather than tune parameters blindly. Hartmann et al. [28] established that creating multiple alternatives in parallel supports design exploration. More recently, InterFaceGAN [61] identified interpretable semantic directions in GAN latent spaces, GANCollage [74] presented a mood board interface for composing GAN-generated elements, and Browsing the Latent Space [8] let users walk through a generative space to discover unexpected outputs. A growing body of systems have extended this approach to specific creative domains, including video editing [41], analogical design [34], art browsing [1], spatial ideation [13, 49], iterative concept exploration [60, 67, 76], visualizing prompt-editing history [25], and tracing prompts to the image regions they control [75].

These systems establish that visibility supports creative exploration, but most focus on navigation and selection within existing outputs rather than creating new points in the space. Prior work provides two further findings that inform the design of GENErator. First, designers who develop alternatives in parallel produce more divergent and better-rated designs than designers who refine one at a time [18, 28], suggesting that a tool which makes it easy to branch from any design may encourage designers to explore a broader range of styles. Second, prior work has examined how the distance between an example and the current design problem affects ideation: far examples raise the novelty of what designers produce but may reduce design quality when the examples are too distant or poorly presented [9, 22], and designers use near and far examples at different stages of the design process [29]. A tool should therefore let designers choose that distance rather than fix it. What this literature has not examined is whether designers understand how the styles they try relate to one another. Prior tools present styles as flat lists of names and thumbnails, and lineage views such as Artbreeder's genealogy [63] show which images were bred from which rather than where the styles themselves sit relative to each other. Keeping a visible record of how alternatives were derived supports reflection and reuse in creative work [23, 62, 71], and recent work has brought such records to image generation so that users can revisit past alternatives rather than regenerate them [4]. In our work, we apply genetic diversity operators directly to the style representation space. Where prior tools help designers express their intent more effectively or navigate existing outputs, GENErator lets designers create new points through crossover, move toward semantically relevant neighbors through mutation, and converge toward targets through natural selection.

2.2 Style Representation and Composition in Generative AI

Over the past year, the technical foundations for multi-style composition have advanced rapidly. B-LoRA [21] showed that style and content can be implicitly separated by training LoRA on specific transformer blocks, indicating that style is a manipulable parameter rather than a side effect of generation. K-LoRA [52] achieved training-free multi-LoRA fusion, demonstrating that multiple styles can be combined without additional optimization. DiffMorpher [79] achieved smooth image morphing by interpolating between LoRA parameters, showing that the LoRA parameter space supports smooth semantic transitions. CLoRA [50] introduced contrastive test-time attention masking to reduce semantic conflict when composing multiple LoRAs, Zhong et al. [80] proposed two training-free methods (LoRA Switch and LoRA Composite) for multi-LoRA composition with the ComposLoRA benchmark, and CMLoRA [56] used frequency-domain analysis to optimize LoRA integration order. Stylus [45] retrieves adapters from a 75K-adapter database by semantic similarity to the prompt and composes the retrieved adapters into a single generation, showing that semantic

Table 1: Comparison of tools for AI-assisted style exploration.

Tool	Exploration strategy	What is varied	Semantic guidance	Multi-style composition	Lineage tracking	Style search
GENErator	**Genetic operations**	**Style models (LoRA)**	**Yes (style embeddings)**	**Yes (crossover, mutation, evolution)**	**Yes (evolution tree)**	**Semantic + reverse image**
Picbreeder [59]	Collaborative IEC	CPPN genotypes	No	Yes (branching)	Yes (phylogenetic tree)	Keyword
Design Galleries [48]	Dispersed precomputation	Rendering parameters	No	No	No	Visual browsing
Artbreeder [63]	User-guided IEC	GAN latent vectors	No	Yes (weighted crossover)	Yes (genealogy view)	No
Hall & Yaman [26]	IEC + aesthetic scoring	CAN latent vectors	Partial (aesthetic fitness)	Yes (crossover)	No	No
DesignWeaver [69]	Dimension variation	Prompts	Partial (named dims)	Partial (multi-tag)	No	No
POET [27]	Auto-diversification	Model outputs	No (automatic)	No	No	No
Luminate [66]	Dimension variation	LLM outputs	Partial (interp. dims)	No	No	Keyword + filter
CreativeConnect [12]	Reference recombination	Text keywords	No	Partial (keyword merge)	No	No
StyleFactory [81]	Strength calibration	Single LoRA weight	No	No	No	No
CivitAI/Vizcom/A1111 [3, 14, 72]	Slider adjustment	LoRA weights	No	Yes (manual sliders)	No	Keyword
Flora/Figma Weave [19, 20]	Manual branching	Workflow nodes	No	Yes (weighted nodes)	Partial (workflow graph)	No

selection scales to large adapter libraries. However, it is a one-shot pipeline: designers cannot steer the selection or trace how a result was derived. In the text-embedding domain, EvoPrompt [24] applied genetic operators to discrete text prompts using LLMs, showing that evolutionary strategies can optimize prompts for downstream tasks, but for automatic prompt optimization rather than style exploration. WANDER [32] steers prompt evolution with CLIP embeddings to diversify outputs, again over prompts for a single model rather than over composable styles.

Many HCI systems focus on helping designers control individual styles. StyleFactory [81] supports style-strength ranking and alignment visualization for calibrating a single style's intensity. DesignPrompt [53] combines images, colors, and text into multimodal prompts for individual generations, and Generative.fashion [15] offers graphical style exploration using GANs. Misty [44] supports conceptual blending of UI design examples into prototypes. A related direction personalizes a single style model to a designer's preferences: DesignLoRA [38] fine-tunes a LoRA through designer-in-the-loop feedback, combining VLM-guided questioning with preferential Bayesian optimization. However, such approaches require the designer to already know the preference the model is being tuned toward. Our goal is to support designers at an early stage where designers discover what they want by exploring and combining styles across a library. While some of these systems support compositional mechanics, few provide ways to navigate or search across a large combinatorial style space. In current practice, platforms like CivitAI [14], Vizcom [72], and Automatic1111 [3] present multi-LoRA composition through weight sliders and text fields with no semantic search, relationship visibility, or lineage tracking. Commercial tools have since made multi-style blending mainstream: Scenario [57] merges up to five custom LoRAs with per-model weights, Recraft [55], Krea [36], and Leonardo [39] blend multiple style references or named style elements, and Midjourney [51] weights multiple style reference codes in a single generation. A separate group of tools supports branching exploration on node canvases: Flora [20] lets users branch variations from any image node, and Figma Weave [19] combines weighted style nodes with a remixable workflow graph. However, these two capabilities remain in separate tools, and in both groups the exploration operations are unguided: blending weights and branch points are chosen by hand, with no notion of where a style sits relative to others. Weight sliders allow designers to adjust the contribution of each style but do not indicate how an adjustment will change the generated design. Work on visual design parameters has found that people struggle to tune such parameters by hand [35] and do better with controls that name a quality of the result, such as learned semantic attributes for shapes and images [10, 78]. We therefore expect that providing genetic operations will reduce the need for designers to adjust style weights by hand. Designers can still adjust the weights of each offspring in GENErator with a slider, which allows us to examine how often they adjust the weights when the genetic operations are also available. In our work, we build on LoRA composition and semantic embeddings to create a style exploration layer where genetic operations guide designers across the style space through semantically meaningful composition rather than manual slider adjustment.

2.3 Interactive Evolutionary Computation

Interactive evolutionary computation (IEC) is a paradigm in which users guide an evolutionary process by selecting preferred candidates from a generated population, with the system producing new candidates through genetic operators such as crossover and mutation [68]. Dawkins [16] introduced the concept with biomorphs, simple shapes evolved through user selection. Sims [64] brought IEC to computer graphics at SIGGRAPH, establishing that selection is a natural interface for aesthetic preference: designers pick what they like rather than describing it. A persistent challenge with IEC is user fatigue: because mutations are random, most generated candidates bear little relationship to the user's aesthetic direction, requiring many evaluation rounds to make progress [68]. Picbreeder [59] extended this to collaborative online evolution where users could branch from each other's evolved images, and Artbreeder [63] brought IEC to millions of users through GAN-based image breeding. However, Artbreeder's mutations remain at the level of random perturbations without reference to named styles or aesthetic concepts, and it provides no lineage tracking across evolved images.

More recent systems have combined IEC with learned latent spaces. Bontrager et al. [6] placed GAN latent vectors under evolutionary control, showing that deep generative models provide compact representations for IEC. Diffusion Crossover [37] defines recombination inside a diffusion model by interpolating the noise sequences of two parent images. However, it operates within a single model, and the recombination is not semantically guided. This means that generations are largely random variations of the same learned model. Schrum et al. [58] extended this to game level generation, StyleIE [70] introduced crossover in StyleGAN's style space, and Hall and Yaman [26] combined IEC with creative adversarial networks, proposing an "intelligent mutation" operator based on aesthetic scoring. This last system comes closest to addressing user

fatigue but still lacks semantic guidance: mutations are drawn from a local neighborhood in latent space without reference to named concepts or user intent, which makes it difficult for designers to interpret and steer results.

Later systems have automated parts of this loop, defining computational fitness functions and running selection over many generations without user input [6, 26]. What has remained difficult is semantics: neither manual selection nor an automated fitness score gives designers a way to express *where* in the space new genetic material should come from [77]. In our work, we aim to address this by making mutation semantically guided: similar mutations select from the most similar styles in the latent style space, distant mutations from the most distant, and natural selection progressively evolves styles towards a user's natural language descriptions. Prior systems mutate the latent vector of a single generator at random. GENErator instead selects a second parent from an explicit region of a shared style space, over a library of named, composable style models, and records every operation in the design's lineage. Because the designer chooses whether new material comes from near or far, a surprising suggestion still has a known intent, and designers are able to better understand the suggested styles and identify those they wish to keep. This matches how serendipity has been described in creative search: an unplanned encounter that the person can recognize and act on [2]. To our knowledge, no prior system combines semantic guidance, multi-style composition, and lineage tracking in one tool (Table 1).

3 GENErator

GENErator supports style exploration through a small set of genetic operations over a shared latent style space (i.e., a learned latent space of design styles). Each operation makes a different kind of move. *Crossover* blends two styles the designer has chosen. *Similar* mutation refines within the current aesthetic family, *distant* mutation jumps to an unrelated region of the space, and *random* mutation draws blindly from the full library. *Natural selection* chains selection and mutation over multiple generations toward a described target. The operations differ along two axes: where new genetic material comes from in the style space, and whether the move spans a single step or multiple automated generations (Table 2). An evolution tree records every operation, so each design's full lineage stays visible. These operations respond to three needs we observed in pilot sessions with professional designers. Designers could not see how styles relate to one another, which motivated the shared style space. They also lacked structured ways to build on styles they already liked, motivating the genetic operations. Finally, they could not retrace how good combinations had emerged, which motivated the evolution tree.

In the rest of this section, we first illustrate how a designer would use GENErator through a lounge chair design example. We then describe the technical implementation, covering the latent style space, gene library and style search, genetic operations, natural selection, and the evolution tree. We detail technical validations of the system's core mechanisms in Appendix A.

3.1 System Walkthrough

Zaha is a furniture designer exploring styles for a new lounge chair concept. Figure 1, Figure 3, and Figure 4 show example interfaces of this walkthrough.

Zaha first types "fluffy" into the *semantic search* bar. GENErator ranks all genes in the database by similarity to her query and surfaces `Plush` (67% similarity score), `Wooly` (55%), and `Cotton Candy` (42%) (Figure 3). She drags `Cotton Candy` onto the canvas, types "lounge chair" as the prompt, and the system generates a playful, cloud-like chair in soft pastel hues.

Zaha also has a photo of a jellyfish that captures a luminous quality she wants. She uploads it using *reverse en-GENE-ering*, and the system identifies which genes best match: 42% `Glow`, 28% `Bioluminescent`, 18% `Ethereal`. She drags `Glow` onto the canvas, over `Cotton Candy` to create a crossover. GENErator blends the two style genes, producing an offspring that merges the softness of `Cotton Candy` with the luminescence of `Glow`. The offspring appears as a child node in the evolution tree (Figure 1b).

Wanting to push the hybrid further without losing its character, Zaha right-clicks the hybrid gene and selects *similar* mutation. GENErator randomly selects from the top 10% most semantically similar genes and picks `Opalescent`, creating a smoother glow. Curious about an unexpected direction, she tries *distant* mutation, which randomly draws from the bottom 10% most dissimilar genes. GENErator adds `Geode`, producing a crystalline offspring (Figure 1b).

Finally, Zaha wants to refine towards a specific aesthetic direction using *natural selection*. She types "luxurious" as a natural selection target. GENErator uses her hybrid gene as the initial seed and automatically evolves populations of styles toward this aesthetic over three generations, blending in genes like `Posh` and `Diamond` that score highest against the target (Figure 4). The evolution tree now shows her full exploration lineage: from initial search, through crossover and mutation, to automated evolution. She can click any node to backtrack and branch off in a new direction without losing her current path.

3.2 Latent Style Space

For genetic operations to be semantically meaningful, the system needs to know where each style lives in a shared space. The challenge is that styles are entangled with objects: encoding "a chair in Chrome" captures both the chrome material *and* the chair shape. Our approach isolates the style contribution by subtracting out the object. For each style gene s, we encode N=30 prompts with the style applied ("a chair in Chrome," "a sneaker in Chrome," etc.) and the same N prompts without ("a chair," "a sneaker," etc.), take the difference for each pair, and average:

$$e_s = \frac{1}{N}\sum_{i=1}^{N}\Big[\text{CLIP}(\text{“a } obj_i \text{ in } style_s\text{”}) - \text{CLIP}(\text{“a } obj_i\text{”})\Big] \quad (1)$$

Each subtraction removes the object's contribution, and averaging across N=30 object categories spanning furniture, apparel, vehicles, and everyday consumer products (chair, sneaker, lamp, handbag, car, etc.) cancels remaining object bias, so that no single object family dominates the averaged direction. We chose N=30

Table 2: GENErator's genetic operations.

Operation	New material drawn from	Scope
Crossover	a designer-chosen gene	single step
Similar mutation	nearest 10% of the style space	single step
Distant mutation	farthest 10% of the style space	single step
Random mutation	full library, uniformly	single step
Natural selection	genes nearest the target description	3–5 generations

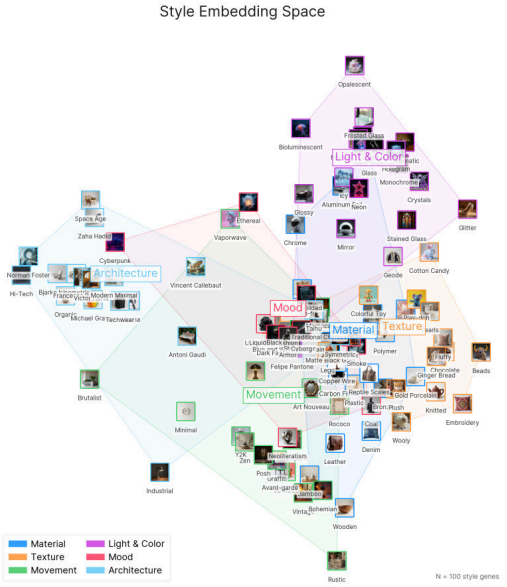


Figure 2: The 100-gene style space projected via PCA, with convex hulls drawn around six categories of related styles. Semantically related styles cluster without explicit supervision: e.g., architectural styles (upper left), luminous effects (top), reflective materials (center), and tactile textures (right).

heuristically, as a balance between averaging out object bias and keeping the embedding cheap to compute for new styles. The result is a 768-dimensional vector that captures what the style *adds* independent of any particular object. For example, `Chrome` and `Glossy` end up nearby because they add similar reflective qualities regardless of whether they are applied to a chair or a sneaker. Figure 2 shows the resulting 100-gene style space projected via PCA: semantically related styles cluster together without any explicit category supervision. This differenced averaging technique is what makes downstream operations work. Similar mutation draws from the nearest 10% of genes, so tighter clustering in the embedding space means more aesthetically coherent suggestions. We validate this in Section A.

We compute style embeddings from text prompts rather than from images generated with each adapter. Text encoding requires no generation passes, which keeps the embedding inexpensive to compute as libraries grow. The natural selection fitness function (Equation 2) instead operates on generated images, closing the loop between the text-derived space and the visual behavior of the composed adapters. An image-based style embedding (generating images with each adapter and subtracting an object baseline) is a natural alternative we have not compared against. We return to this in Section 6.1. This design also assumes that distances in CLIP's text space track how the adapters actually render. We test this assumption in Appendix A: gene-pair similarities in the text-derived space correlate with similarities between CLIP image embeddings of an example image of each gene (ρ=0.34, p<.001).

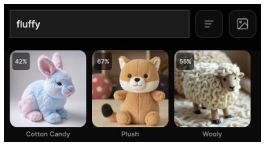


Figure 3: Semantic search in the gene library. Querying "fluffy" ranks all genes by semantic similarity, surfacing Plush (67%), Wooly (55%), and Cotton Candy (42%).

3.3 Gene Library and Style Search

GENErator's gene library contains 100 curated LoRA [31] style adapters sourced from CivitAI, HuggingFace, and our custom-trained models (Figure 2). We curated the library for coverage, selecting genes so that all six categories were well represented. Genes span six categories (material, texture, movement, light & color, mood, architecture) and are organized with filterable tags for era, fidelity, and color tone (Figure 3 and Table 3).

Designers find genes through two search modes. *Semantic search* lets designers type a natural language description (e.g., "dark and moody") and the system ranks all genes by cosine similarity between the query's CLIP embedding and each gene's style embedding. Because the embeddings capture semantic meaning rather than keyword overlap, "dark and moody" surfaces `Coal`, `Dark Fantasy`, `Grunge`, and `Brutalist` even though "moody" does not appear in any trigger word. *Reverse en-GENE-ering* lets designers upload any reference image and the system identifies which genes best match its aesthetic, using the CLIP image encoder in place of the text encoder. Results appear as a ranked list with match scores (e.g., "42% `Glow`, 28% `Bioluminescent`, 18% `Ethereal`"), and designers can drag any identified gene directly onto the canvas.

3.4 Genetic Operations

Crossover combines two parent genes into an offspring by blending their LoRA adapters. The offspring inherits both parents' adapter sets, each scaled by a blend ratio $\alpha \in [0, 1]$ (default 0.5, adjustable

via slider). Each offspring is added as a child node in the evolution tree, connected to both parents, building a branching exploration structure.

Mutation introduces new genetic material by blending a source gene with a second gene selected from the library. GENErator offers three mutation modes that control how far the generated offspring is from the source gene's style. *Similar* draws from the top 10% most similar genes in the style space (cosine similarity $\geq \tau_{90}$, approximately 0.66), yielding a pool of ~10 candidate genes that refine within the same aesthetic family. *Distant* draws from the bottom 10% most dissimilar genes (cosine similarity $\leq \tau_{10}$), introducing a different aesthetic direction. *Random* selects uniformly from the full library. The distinction between modes is not how blending occurs (all use the same weighted adapter combination as crossover) but *where* the second parent comes from in the style space (Figure 13).

3.5 Natural Selection

While crossover and mutation let designers manually guide exploration one step at a time, natural selection automates the process: the designer describes *where* they want to go, and the system evolves a population of styles toward that target.

The mechanism adapts the Simple Genetic Algorithm (SGA) [30] for semantic style exploration. In standard SGA, mutation is random and fitness is defined externally (e.g., manually by the user). GENErator makes two changes: mutation is semantically-guided (drawn from genes semantically related to the target rather than random noise), and fitness is computed automatically by comparing generated images to the designer's target description.

Given a target description t (e.g., "luxurious"), the system initializes a population of P individuals (optionally seeded from the canvas), then iterates. First, each individual is **evaluated** by generating an image and scoring how well it matches the target:

$$f(g, t) = \cos\big(\text{CLIP}_{\text{img}}(\text{Generate}(g)),\ \text{CLIP}_{\text{text}}(t)\big) \qquad (2)$$

where $\text{Generate}(g)$ renders a product image using gene g's LoRA adapters, and the cosine similarity between the image embedding and the target text embedding serves as the fitness score. Because raw image-text cosine similarities occupy a narrow band, the system linearly rescales them from this band (0.02 to 0.28 in practice) to a 0 to 1 fitness score before selection, and fitness values reported in this paper use this scale. The system then **selects** the top performers weighted by fitness, and **mutates** each by blending it with a gene drawn from the top 10% most similar to the *target description* (not the parent, as in similar mutation). The loop runs for 3–5 generations (defined by the user), with the interface offering population × generation presets (3×3, 4×4, and 5×5). Seeding from the canvas preserves continuity with the designer's current exploration: the target decides where the population is headed, while the seeds decide what aesthetic ancestry it carries along the way. Because mutation candidates are sampled randomly from within the target-nearest pool, successive runs vary even for identical targets. The result is a grid of evolved designs that progressively embody the target aesthetic (Figure 4).

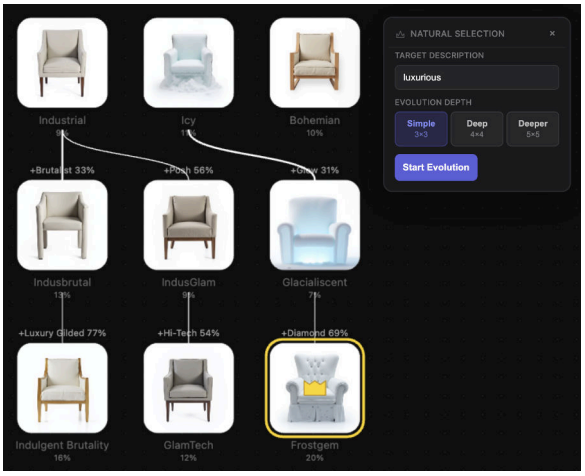


Figure 4: Natural selection evolving a population of three styles toward "luxurious" over three generations. In each generation, the system scores every candidate among the population by similarity to the target, selects the fittest, and mutates them using semantically related genes to produce the next generation. At the final generation, the winner is selected (gold crown).

3.6 Evolution Tree

As designers crossover, mutate, and evolve genes, GENErator builds an evolution tree that tracks the full lineage of every style explored. Parent genes appear above their offspring, with connection lines showing which operation produced each child (crossover, similar, distant, or evolution). Line thickness indicates the contribution of each parent: a thicker branch means that parent contributed more to the offspring's blend.

Each offspring receives a creative name generated by an LLM (e.g., crossing `Cotton Candy` with `Glow` might produce `Cotton Glow`). Hovering over any gene reveals a composition tooltip showing a pie chart of its genetic makeup, e.g., "45% `Cotton Candy`, 30% `Glow`, 25% `Geode`". Designers can click any node to backtrack, branching off from earlier points in the tree to explore alternative directions without losing their current branch. They can also bookmark favorite genes, save custom hybrids back to the library for future use, and export the full canvas as a visual record of their exploration. Figure 5 shows an example sequence: starting from two genes, a handbag design is refined across six iterations, with genes added and removed while each step builds on the previous offspring.

We implement GENErator as a web application using an infinite canvas interface with D3.js [7] visualizations of evolution trees. Image generation uses the FLUX.1-dev model [5] with multi-LoRA support, producing each image in approximately 2 seconds. An LLM recommends an appropriate camera angle for each product type to keep style comparisons visually consistent (e.g., a 3/4 front view for a sneaker). Style embeddings are encoded with the CLIP ViT-L/14 model [54]. All interaction events (generations, gene additions, crossovers, mutations, evolutions, searches, bookmarks, and weight adjustments) are logged in a database for post hoc analysis.

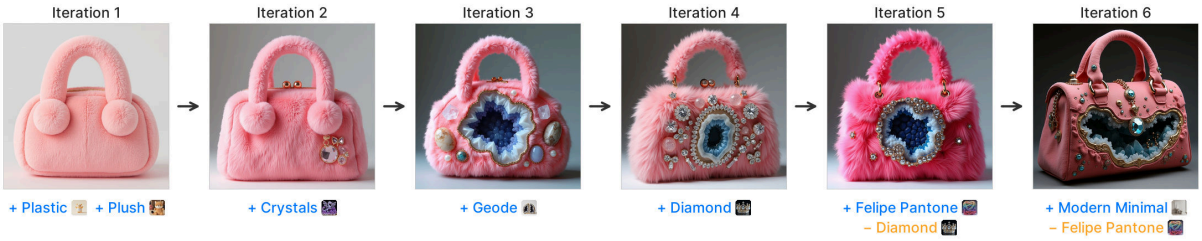


Figure 5: Example iteration sequence for a handbag. Starting from Plastic + Plush, the designer adds or removes genes across six iterations, each building on the previous offspring.

We validate three quantitative properties of the embedding and genetic operations in Appendix A. Our embedding method achieves 83% tighter within-group clustering than straightforward text encoding ($d_z = 2.32$). Similar-mutation targets land 24% closer to the source than random ($d_z = 3.22$), distant targets 27% farther ($d_z = 4.76$). Figure 13 shows the visual effect across six product types. Natural selection converges monotonically across all 10 tested aesthetics (Figure 12).

4 User Study

We conducted a within-subjects study to understand how GENErator could support designers in exploring visual styles, its potential to change how designers interact with style spaces, and to identify areas for improvement. We compared GENErator against a baseline condition representing a standard interface for multi-style AI image generation tools used by designers today. The following outlines our participants, baseline, procedure, and measures.

4.1 Participants

We invited twelve participants (P1–P12, 8 male and 4 female, aged 18–53, μ=27.8, σ=9.0) to participate in the study. Participants were recruited from design programs and professional practice through known contacts and word of mouth, with the majority coming from one of the top five industrial design schools in the United States. Design experience ranged from 1 to 26 years (μ=6.0, σ=6.8), spanning product design, automotive design, CMF (color, material, finish), furniture design, architecture, and interaction design. Student participants typically had professional-level experience through a combination of industry-sponsored internships and co-op programs. Participants reported moderate familiarity with AI image generation (μ=2.5, σ=1.1, 4-point scale) and low familiarity with LoRA-based style generation specifically (μ=2.0, σ=1.0, 4-point scale). They were not exposed to GENErator prior to the study. Participants accessed both tools through a web browser. We compensated participants $40 USD for approximately 90 minutes. This study was approved under our institution's IRB.

4.2 Baseline

The baseline represents a standard interaction model for multi-style AI image generation tools, similar to widely-used platforms such as Automatic1111 WebUI [3], CivitAI's online generator [14], and Vizcom [72]. The baseline shares the same gene library (all 100 styles) and the same generation pipeline (same model, resolution, and quality). The difference is how designers explore. In the baseline, designers select styles from a browsable grid, adjust individual weight sliders (0–100%) to control each style's influence, and click Generate. A generation history grid displays all previous outputs. Clicking any previous result restores its selected styles and weights, enabling one-click backtracking. This is stronger than most existing tools, which require designers to manually recreate previous configurations (Figure 16).

4.3 Procedure

4.3.1 Introduction (10 minutes). Participants provided informed consent, completed a background questionnaire, and were given an overview of both the baseline (Figure 16) and GENErator (Figure 1) tools.

4.3.2 Comparison Tasks (50 minutes). Participants completed a design task with GENErator and another design task with the baseline. The two design tasks are "design a futuristic sneaker" and "design a luxurious handbag." Participants received only these text briefs, with no reference images, and interpreted "futuristic" and "luxurious" in their own way. We chose the two adjectives because they can be interpreted in many aesthetic directions and apply naturally to both products. We counterbalanced both the order of the conditions and the assignment of design tasks to conditions. Each task lasted 20 minutes. Participants bookmarked their favorite outputs at the end of each session and completed questionnaires about the tool they just used. Throughout the comparison tasks, participants shared their screen and verbally explained their actions and thinking (think-aloud). The researcher observed and took notes.

4.3.3 Free Exploration and Post-Study (30 minutes). Participants used GENErator to design any product of their choice with full creative freedom (20 minutes). Participants designed water bottles, kitchen tables, flying cars, gaming mice, and cat food dispensers. Participants then gave feedback through open-ended questions asking them to compare the two tools, describe moments of surprise or discovery, and discuss how they might integrate each into their workflow (10 minutes).

4.4 Measures

For both conditions, we asked participants to complete questionnaires to capture their experience using GENErator and the baseline. We assess creativity and exploration using 5-point Likert questions adapted from the Creativity Support Index (CSI) [11]. We included

four custom questions targeting style-specific dimensions not covered by CSI: diversity of output, unexpected style discovery, understanding of style relationships, and sense of control. We developed these four items to assess the outcomes that the genetic operations are intended to affect. Diversity of output and unexpected discovery correspond to the problems of fixation on familiar styles [33] and the absence of serendipitous encounters with styles the designer would not have searched for [2]. Understanding of style relationships follows the distinction in exploratory search between finding an item and understanding the space it sits in [47], and targets the evolution tree and the semantic style space. Sense of control checks whether the guided operations and automated evolution take control away from the designer, a common concern for co-creative AI tools.

In addition, we log all interaction events to a database with timestamps, including every generation, crossover, mutation, evolution run, semantic search, bookmark, and weight adjustment. From these logs we analyze operation counts to see which operations participants relied on and whether they adjusted weights less often; operation sequences, to see how participants chained operations and whether they alternated between divergent and convergent moves; the lineage of bookmarked designs, to see whether the designs participants kept were bred through the operations or taken directly from the library; and event timing to see how long participants deliberated before each kind of operation.

We computed post-hoc semantic similarity metrics from user-generated designs, including style space coverage (convex hull area in PCA-projected embedding space, see Figure 8) and mean pairwise CLIP [54] distance between outputs, following prior work that measures design space coverage as convex hull area [43, 46] and diversity as mean pairwise distance in an embedding space [43]. Together with the number of unique styles a participant used, these give three views of how a participant covered the style space: how many styles they reached, how large a region those styles span, and how densely that region was explored (unique styles per unit of hull area). To describe how participants moved through the space, we group the operations by where they draw new material from. Divergent operations bring in a style from far away or from the whole library, such as distant and random mutation, natural selection, semantic search, and adding a gene from the library. Convergent operations stay near what is already on the canvas, such as crossover, similar mutation, and weight adjustment. We call a style unfamiliar to a participant when they have not used it in the session. Distant mutation reaches unfamiliar styles by design, since it draws from the farthest 10% of the space.

For each questionnaire item, we first tested whether the paired differences were approximately normally distributed (Shapiro-Wilk). We compared items that passed with paired t-tests, and items that did not (creativity, diversity, unexpected discovery, and understanding) with Wilcoxon signed-rank tests. Behavioral measures were compared with paired t-tests. For every comparison we report the mean and standard deviation for GENErator and for the baseline, the test statistic (t or W), the p-value, and the paired effect size. Because the questionnaire involves multiple comparisons, we also aggregate the eight CSI-adapted items into a composite score as the primary test of creativity support. Under a Holm-Bonferroni correction within item families, the composite and the discovery and understanding items remain significant. The remaining individual items do not survive this correction. We reviewed qualitative data from open-ended responses, think-aloud transcripts, and researcher observation notes to support the quantitative results.

5 Results

All twelve participants completed both comparison tasks and the free exploration session. GENErator improved exploration breadth, style understanding, and creative ownership across both subjective (Figure 6) and behavioral measures (Figures 8, 10). Example designs created by participants are available at https://style-generator.github.io.

5.1 Creativity and Exploration

GENErator enables broader exploration and stronger creative ownership. Participants reported significantly higher ease of exploration when using GENErator (GENErator: μ=4.25, σ=0.75; baseline: μ=3.42, σ=0.90; $t(11)$=2.59, p=.025, d_z=0.75). Participants' sense of creativity showed the largest effect among the CSI items (GENErator: μ=4.08, σ=0.79; baseline: μ=3.25, σ=0.62; W=0, p=.016, d_z=1.00). We also aggregated the eight CSI-adapted items into a single score. On this composite, participants rated GENErator higher overall (GENErator: μ=3.90, σ=0.34; baseline: μ=3.41, σ=0.34; $t(11)$=3.31, p=.007, d_z=0.96). Participants also reported higher enjoyment (GENErator: μ=4.25, σ=0.75; baseline: μ=3.58, σ=0.79; $t(11)$=2.60, p=.025, d_z=0.75) and satisfaction with their designs (GENErator: μ=4.00, σ=0.85; baseline: μ=3.08, σ=0.51; $t(11)$=2.93, p=.014, d_z=0.85). As P4 put it, using GENErator felt like '*I made this*' while the baseline felt like '*the AI made this*' (P4).

This was supported by behavioral data: participants explored 2.2 times more unique base styles with GENErator (GENErator: μ=16.8, σ=4.5; baseline: μ=7.5, σ=2.1; $t(11)$=6.26, p<.001, d_z=1.81), and bookmarked 5.2 times more outputs (GENErator: μ=9.2, σ=2.2; baseline: μ=1.8, σ=1.4; $t(11)$=8.16, p<.001, d_z=2.36). Bookmarking happened during the task, so it reflects which designs participants valued as they worked. Bookmarked outputs also grew faster than exploration itself (5.2 times vs. 2.2 times), suggesting that exploring more styles also led to more designs participants wanted to keep. Participants also generated more often with GENErator (GENErator: μ=48.3, σ=6.1; baseline: μ=28.0, σ=7.9; $t(11)$=10.26, p<.001, d_z=2.96; Figure 7), and exploration grew faster than generation volume: 2.2 times more unique styles from 1.7 times more generations (0.35 vs. 0.27 unique styles per generation). Participants also rated their outputs as significantly more diverse (GENErator: μ=4.00, σ=0.95; baseline: μ=3.33, σ=0.89; W=0, p=.031, d_z=0.86), and several found that hybrid offspring surpassed their parents: '*This is actually looking better than previous ones*' (P11).

Figure 8 shows where the explored styles sit in the style space. The convex hull spanned by a participant's styles was larger with GENErator for 9 of 12 participants (GENErator: μ=0.21, σ=0.05; baseline: μ=0.17, σ=0.06; $t(11)$=1.71, p=.116, d_z=0.49). The mean pairwise CLIP distance between explored styles was comparable (GENErator: μ=0.54, σ=0.01; baseline: μ=0.53, σ=0.03; $t(11)$=1.63, p=.131, d_z=0.47). Additional styles filled in the region participants covered, where unique styles per unit of hull area were 1.7 times higher with

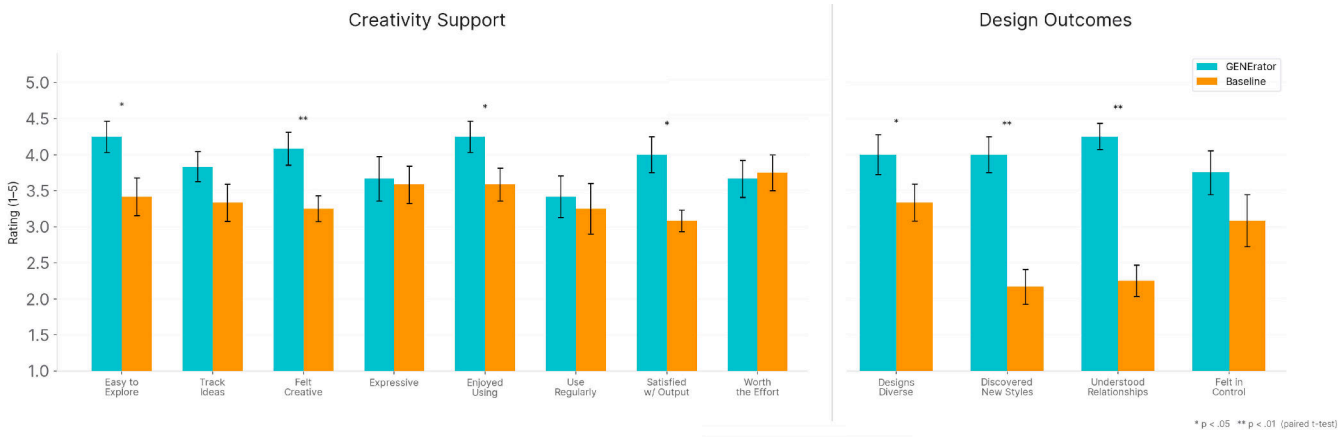


Figure 6: Likert results across all subjective measures. *Left:* Creativity and exploration (adapted from the Creativity Support Index [11]). GENErator rated significantly higher on ease of exploration, creativity, enjoyment, and satisfaction. *Right:* Style-specific design outcomes. GENErator rated significantly higher on diversity, unexpected discovery, and understanding of style relationships. Error bars show standard error.

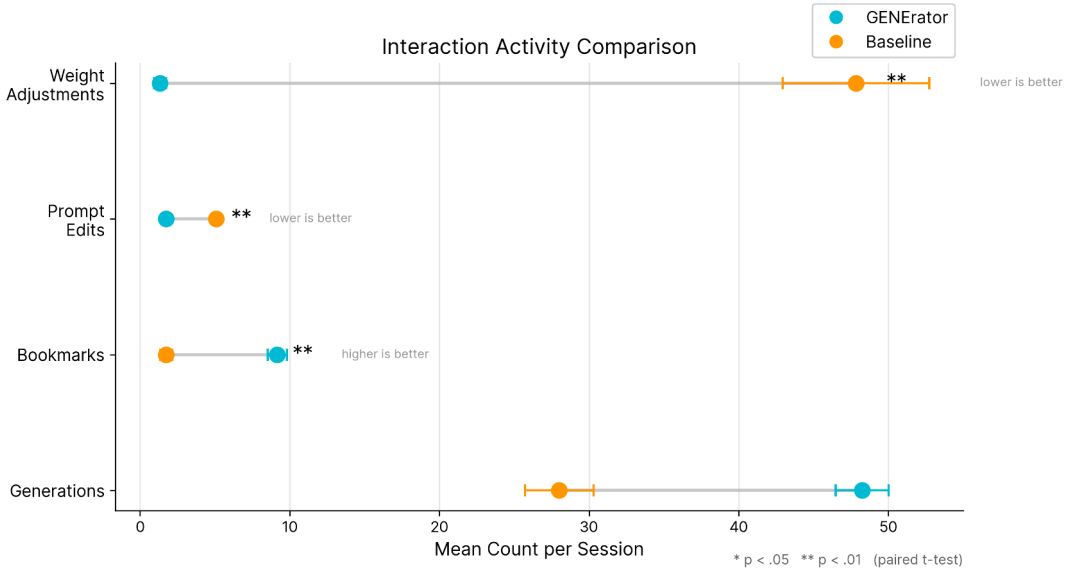


Figure 7: Mean interaction counts per session for shared operations between GENErator and baseline (weight adjustments, prompt edits, bookmarks, generations). GENErator significantly reduced weight adjustments and prompt edits while increasing bookmarks (designs participants liked). Error bars show standard error.

GENErator (GENErator: μ=81.1, σ=16.2; baseline: μ=48.0, σ=15.5; $t(11)$=4.67, p<.001, d_z=1.35), and higher for 11 of 12 participants. GENErator's advantage comes from supporting designers to visit more styles within an intended region of the design space rather than from wide sampling of different individual styles. Figure 15 shows P12's trajectories as a representative example.

Because the tree records every operation, each bookmarked design can be traced back to the library genes it came from. Of the 54 designs we could trace, 46% were base genes taken directly from the library, 19% were first-generation offspring, and 35% were at least two operations deep, with the deepest lineages reaching six operations.

5.2 Style Understanding and Discovery

GENErator makes style relationships understandable and leads to more serendipitous discoveries. The ability to understand the relationships between styles showed the largest subjective effect (d_z=1.66). Participants rated significantly higher understanding when using GENErator (GENErator: μ=4.25, σ=0.62; baseline: μ=2.25, σ=0.75; W=0, p<.001, d_z=1.66). The evolution tree interface

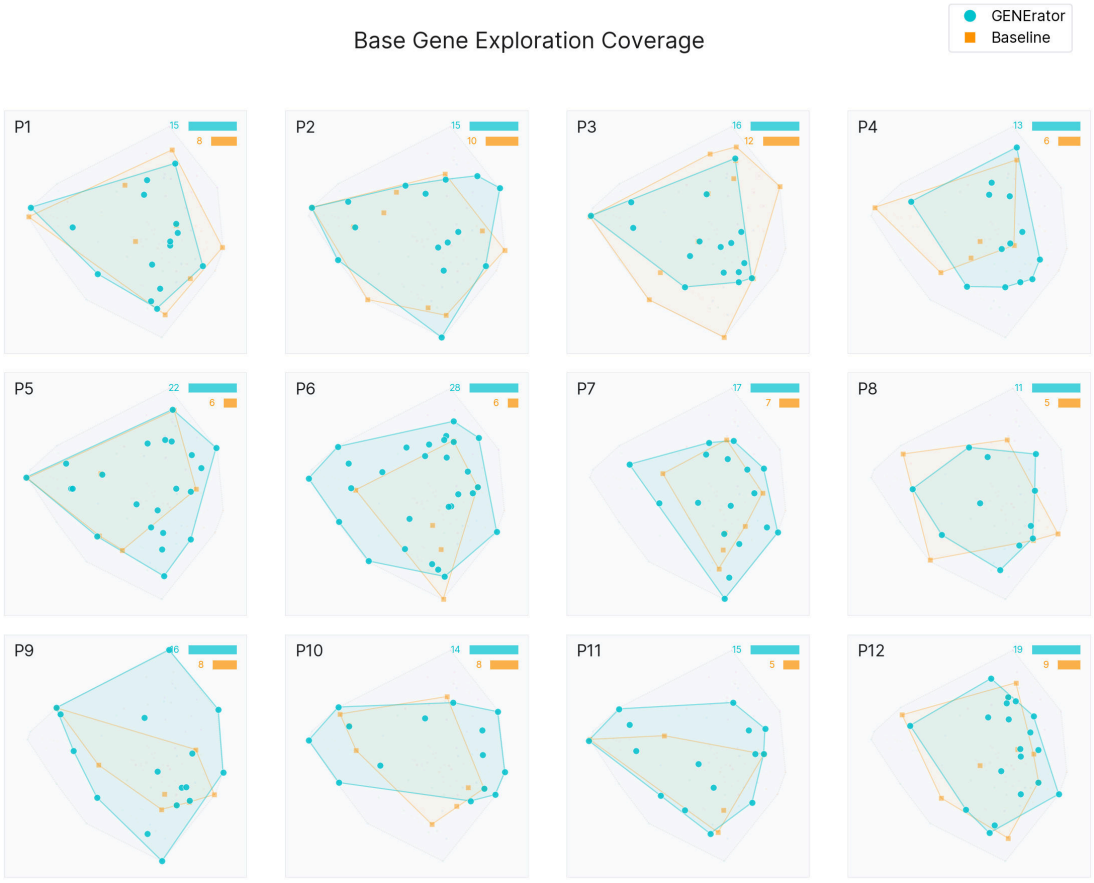


Figure 8: Per-participant style space coverage in PCA-projected CLIP space. Convex hulls show GENErator (teal) vs. baseline (orange). The faded gray convex hull shows the complete style space (the same one as Figure 2). GENErator participants generally covered more of the style space.

was noted as a key mechanism: P6 commented '*you can see the whole tree of styles I explored and eventually see how the end product was created*' (P6), and P3 described '*the canvas tool (GENErator) feels more reliable because I can understand the generation process and adjust it with clearer direction*' (P3). In the baseline, the same participants called it '*all blind box behind me*' (P3), and P5 noted '*the name of the style doesn't mean anything because the outcome is very different from the style*' (P5). Several participants connected seeing the lineage to taking bigger steps. P6 wrote that the canvas '*gives more freedom to merge different styles because of the tree format*' (P6), and P12 described their process as first combining two options that fit the brief, then '*throw[ing] in some bolder options to see how far I could push the design*' (P12), with the earlier branch still on the canvas to return to.

Participants also reported significantly more *unexpected discoveries* with GENErator (GENErator: μ=4.00, σ=0.85; baseline: μ=2.17, σ=0.83; W=0, p<.001, d_z=2.20). Distant mutation was the main mechanism for surfacing unfamiliar styles. It draws from the farthest 10% of the style space, and its offspring landed 1.7 times farther from the source than similar-mutation offspring (Appendix A). Of the 85 mutations participants ran, 29 were similar, 22 distant, and 34 random, so most drew from outside the aesthetic family the participant was working in. P7 applied a distant mutation to a mirrored metal gene, expecting something dull. Instead, GENErator surfaced a neon light gene: '*I expected something more bland... like the opposite of mirror*' (P7). The result was not a literal opposite but an unexpected aesthetic direction. Later, exploring a flying car concept, P7 observed: '*Guess it associates flying with translucency. That's interesting*' (P7). Not all distant suggestions were immediately accepted, but participants valued that surprises were interpretable: P6 described an unexpected glass texture from crossover as '*not exactly like the [parent] styles but also something that came out of merging them*' (P6). For P12, '*combining two completely different options and getting a result that was unexpectedly interesting was a discovery that could've only been quickly generated with this tool*' (P12). All twelve participants used at least one mutation during their session, all twelve used the similar mode alongside a divergent mode (distant or random), and nine used all three. Figure 14 visualizes this pattern across all participants: similar mutations (blue) generally stay closer to the source gene and distant mutations (orange) jump to farther regions.

5.3 Interaction Patterns

GENErator shifts designers from repeated parameter tweaking to structured, orchestrated exploration. Interaction logs revealed a clear shift in how designers explored the style space (Figures 9 and 10). With the baseline, participants followed a loop through the sliders. The most common path ran from adjusting a weight to generating (33% of moves out of a weight adjustment), from generating to viewing the result full screen (52%), and from viewing back to the sliders (24%). Before generating, participants usually adjusted more than one slider: half of all weight adjustments (51%) were followed by another adjustment, so adjustments came in short bursts between generations. Each generation largely started from a new slider configuration rather than building on previous results. The dominant action was weight adjustment (μ=47.8, σ=17.0 per session), accounting for nearly half of all interactions – yet adjusting a slider from 70% to 80% gives little indication of how the output will change. Participants also deselected styles 15 times per session on average, and 63% of the times they used the history grid to return to an earlier result, their next action was another restore. P9 commented '*I just keep changing sliders randomly. It's hard to understand what's happening.*' (P9).

With GENErator, participants built on previous outputs through crossover, mutation, and evolution rather than reconfiguring each generation independently. Weight adjustments dropped significantly (GENErator: μ=1.3, σ=1.4; baseline: μ=47.8, σ=17.0; $t(11)$=−10.01, p<.001, d_z=−2.89), with participants leveraging crossover (μ=12.3, σ=5.9), mutation (μ=7.1, σ=1.1), and evolution (μ=1.6, σ=1.3). Prompt editing also dropped significantly (GENErator: μ=1.8, σ=0.8; baseline: μ=5.1, σ=1.0; $t(11)$=−8.86, p<.001, d_z=−2.56), suggesting that genetic operations substantially reduced the role of prompt engineering (Figure 7). Adding these up, participants made about 53 parameter adjustments per session in the baseline (weight changes plus prompt edits) and about 3 with GENErator. In their place, GENErator sessions averaged about 21 genetic operations (crossover, mutation, and evolution combined). The operations also followed a consistent order (Figure 9). For example, adding a gene was followed by a crossover 57% of the time, a crossover by a generation 72% of the time, and a mutation by a generation 71% of the time (81% for similar mutations). After a generation, participants most often added another gene (20%) or mutated the result. Weight adjustment was rarely part of the process, despite being clearly available, where a generation was followed by a slider adjustment 9% of the time.

Participants' use of the operations also shifted within a session. In the first third of a session, divergent operations (distant and random mutation, evolution, search, and adding genes) accounted for 73% of exploration actions. This share fell to 63% in the middle third and 59% in the final third (first vs. final third, p=.05), indicating that participants gathered new material early and refined it later. The two kinds of operation were not separated into phases: all twelve participants used both, and on average 45% of consecutive exploration actions switched between them. Participants also paused for different lengths of time before each operation. The median pause was 3.7 seconds before a crossover, 10 to 14 seconds before a mutation depending on the mode, and 57 seconds before an evolution run (typing the description). Participants also ran evolution sparingly (μ=1.6 per session) and combined it with manual operations rather than letting it explore on its own: three quarters of evolution runs were followed by manual operations on the results within the next five actions. P10, with over 25 years of design experience, described GENErator like orchestrating a design team: '*Your team of designers is constantly generating ideas... and you are combining and you say, how about a little bit of this one with this part of that one*' (P10). Several participants placed GENErator within their broader workflow: P10 saw it as '*intended for the mood board phase*' (P10), and P7 as '*a great "pop" moment for exploring CMF, inspirations, or just overall visualization of what a product would look like*' (P7).

In the free exploration session, participants designed products from their own practice, from a car interior (P10) to a gaming mouse and a cat food dispenser (P5), and most of these designs combined three or more genes (Figure 11). P9's lounge chair carried forward the Lego gene they had evolved earlier in the session, and crossed it with five others. When participants ran natural selection, the targets they typed were often their own briefs rather than a single adjective: '*comfort, sectional, surrounded*' (P9), '*simplified organic*' (P10), and '*minimalist table with wooden angular legs*' (P1).

6 Discussion

A core challenge we grappled with in this work was helping designers explore diverse styles without resorting to random trial-and-error. Current tools mainly sit at two extremes: fixation, where vocabulary-bounded search keeps designers circling the same aesthetic territory, or noise, where random mutation in traditional IEC produces user fatigue [68, 77]. Our results suggest GENErator enables a middle ground we call *guided divergence*: exploration that is simultaneously broader and more intentional than either extreme. The system introduces styles the designer has not tried, but each one is drawn from a chosen region of the style space (near the current design, distant from it, or near a described target) rather than at random, and the evolution tree records which operation introduced it. Each genetic operation plays a distinct role: similar mutation refines within an aesthetic family, distant mutation leaps to unfamiliar territory, crossover blends two directions, and natural selection converges toward a target. That most participants used both similar and distant mutation within a single session (Section 5.2) suggests they valued surprise as much as refinement. The mean pairwise distance between explored styles was comparable across conditions (Section 5.1). However, the additional breadth when using GENErator came from visiting *more regions of the intended design direction*. With GENErator, participants were able to converge confidently toward a solution. The share of divergent operations fell from 73% in the first third of a session to 59% in the final third (Section 5.3). In the free exploration session, participants were able to leverage genetic operations to evolve design styles for a diverse range of products of their own choosing (water bottles, watches, houses). The more general principle may generalize to other domains where AI models can be embedded in a meaningful space (e.g., fonts, music, or motion styles): once each style model has a position in a shared style space, exploration can be defined as intentional moves in that space rather than as edits to parameters.

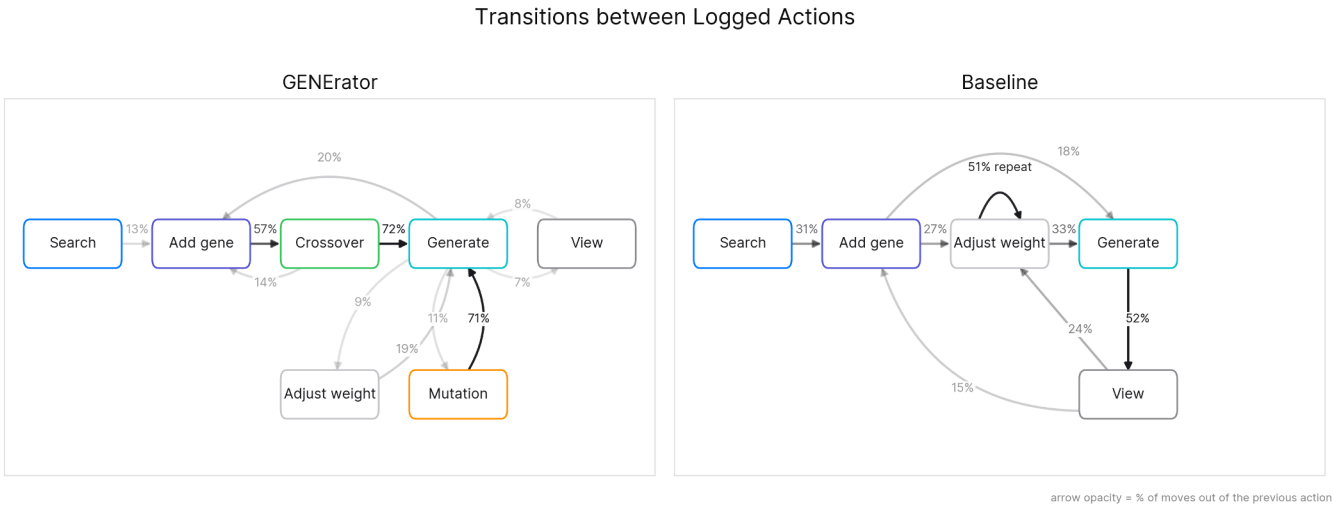


Figure 9: Transitions between logged actions across participants. Arrows show the share of moves from one action to the next (darker is greater share). GENErator moves involve cycles of genetic operations: crossover or mutation, generate, and repeat. Baseline moves involve frequent manual slider tweaks: half of all weight adjustments were followed by another adjustment before seeing any generation.

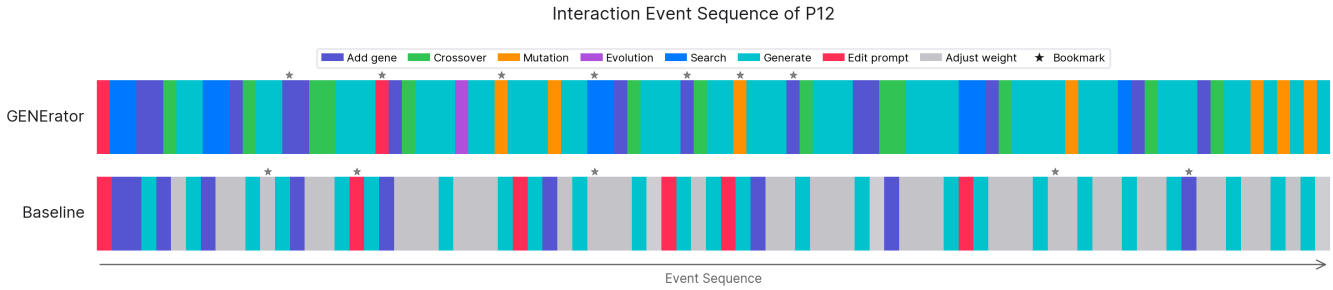
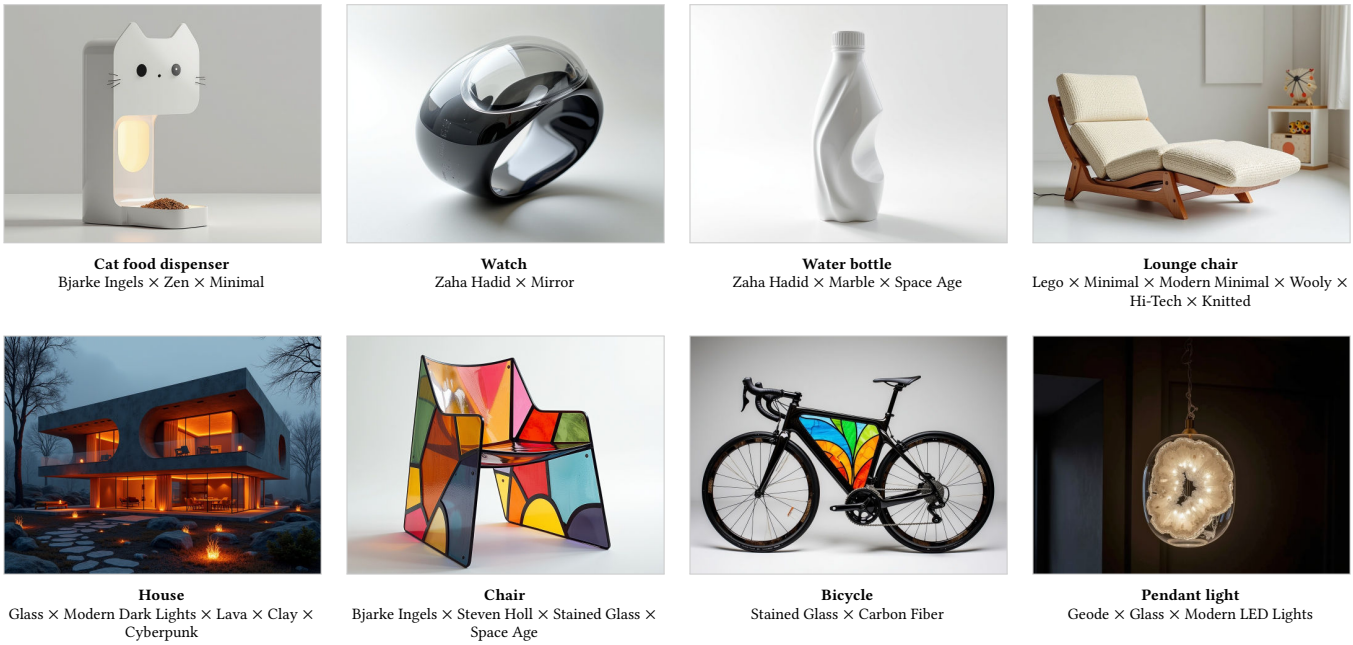


Figure 10: Interaction event sequence for P12. The baseline (bottom) is dominated by weight adjustments (gray). GENErator (top) shows crossover (green), mutation (orange), evolution (purple), and search (blue), with more bookmarks (stars). See Figure 17 for all 12 participants.

Our results also suggest a structural limitation of slider-based interfaces: weight sliders offer parametric control with little semantic meaning. Genetic operations can replace opaque parameter tuning with semantically interpretable composition. GENErator also provides weight sliders: the designer can adjust each parent's contribution to any offspring (Figure 1d). However, participants chose to steer their generations with GENErator through genetic operations rather than weight adjustments (Section 5.3). Understanding style relationships and how styles can be combined was the strongest subjective finding for GENErator, and the evolution tree was frequently mentioned as a primary mechanism. Participants who could see how designs derived from parent styles reported feeling more confident branching into unfamiliar territory, suggesting that transparency may enable bolder exploration. Where the baseline's history grid shows *what was generated*, the evolution tree shows *how generated designs relate to each other*. This was supported by the pairing in our results, where understanding of style relationships and unexpected discovery were the two largest subjective effects (d_z=1.66 and 2.20).

6.1 Limitations and Future Work

There are several avenues for future work collected from participants' feedback and our observations. First, GENErator operates at the level of overall visual style. Several participants expressed interest in controlling specific visual elements, such as mutating only the color palette while keeping the form, or crossing over the texture of one design with the silhouette of another. Second, GENErator currently exists as a standalone exploration tool. P10 described it as '*for the mood board phase*' (P10), suggesting it fits early in the design process. Exporting material parameters or color palettes to downstream tools such as Figma, Rhino, or KeyShot could bridge the gap between exploration and production. Third, our study compares interaction paradigms as wholes: the full GENErator system against a slider-based workflow. This establishes that



Cat food dispenser
Bjarke Ingels × Zen × Minimal

Watch
Zaha Hadid × Mirror

Water bottle
Zaha Hadid × Marble × Space Age

Lounge chair
Lego × Minimal × Modern Minimal × Wooly × Hi-Tech × Knitted

House
Glass × Modern Dark Lights × Lava × Clay × Cyberpunk

Chair
Bjarke Ingels × Steven Holl × Stained Glass × Space Age

Bicycle
Stained Glass × Carbon Fiber

Pendant light
Geode × Glass × Modern LED Lights

Figure 11: Designs from the free exploration session, where participants chose their own products. Each label lists the genes mixed to produce the design.

the paradigm helps, but it cannot attribute the improvement to any single component. Isolating the contribution of each genetic operation, of semantic search, and of the evolution tree would require ablation studies that hold the rest of the interface fixed. Fourth, we measured exploration breadth and subjective experience but did not independently evaluate output quality. Compared to the baseline, participants bookmarked 5.2 times more outputs (p<.001), a behavioral indicator of perceived design value taken during the task rather than a post-hoc rating. Participants also rated significantly higher satisfaction (p=.014, d_z=0.85). Additional expert quality ratings could complement these self-report measures. Fifth, natural selection uses CLIP similarity as the fitness function. While convergence is consistent across 10 tested aesthetics (Figure 12), CLIP was not trained for aesthetic judgment specifically. Each generation also keeps the candidates with the highest CLIP score, so the score is expected to rise as evolution proceeds. Rising fitness shows that selection and mutation steer the population toward the target as CLIP measures it, but not whether the evolved designs would look better to a person. Future work could explore aesthetic-specific metrics. The style embeddings are also computed from text prompts. The embedding could instead be computed from images generated with each adapter, which would capture visual similarity directly. Sixth, while no participant reported fatigue in our sessions, longer-horizon studies with explicit fatigue measures would test whether guided divergence sustains its advantage as exploration scales up. Finally, our library of 100 genes was curated by the research team. The clustering in Figure 2 also partly reflects that curation: an uncurated collection would contain near-duplicate and low-quality adapters, which could crowd the space and weaken the separation between similar and distant pools. The embedding technique itself scales (computing additional embeddings is inexpensive), but deployment with larger ecosystems such as CivitAI's [14] 100K+ styles would test whether the experience remains effective at scale.

7 Conclusion

This research identifies the challenges designers face when exploring visual styles with AI generation tools and proposes a way to support exploration that is both broad and steerable. The proof-of-concept research prototype, GENErator, treats each visual style as a gene in a shared style space, enabling designers to crossover, mutate, and evolve styles through genetic operations grounded in a content-independent style embedding. In a within-subjects study, participants using GENErator explored 2.2 times more styles, filled the region of the style space they covered 1.7 times more densely, and bookmarked 5.2 times more designs. They also reported improved understanding of how styles relate to one another, shifted from adjusting weight sliders to genetic operations, and adopted more intentional, branching exploration patterns rather than opaque trial-and-error.

References

[1] Shm Garanganao Almeda, John Joon Young Chung, Bjoern Hartmann, Sophia Liu, Brett Halperin, Yuwen Lu, and Max Kreminski. 2025. Artographer: a Curatorial Interface for Art Space Exploration. *arXiv preprint arXiv:2512.02288* (2025).

[2] Paul André, m.c. schraefel, Jaime Teevan, and Susan T. Dumais. 2009. Discovery is never by chance: designing for (un)serendipity. In *Proceedings of the Seventh ACM Conference on Creativity and Cognition*. ACM, 305–314. doi:10.1145/1640233.1640279

[3] AUTOMATIC1111. 2022. Stable Diffusion Web UI. https://github.com/AUTOMATIC1111/stable-diffusion-webui. Open-source web interface for Stable Diffusion with multi-LoRA support.

[4] Karim Benharrak and Amy Pavel. 2026. HistoryPalette: Supporting Exploration and Reuse of Past Alternatives in Image Generation and Editing. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems*. ACM. doi:10.1145/3772318.3790742

[5] Black Forest Labs. 2024. FLUX.1: Open-Weight Text-to-Image Models. https://github.com/black-forest-labs/flux. Open-weight diffusion model with native multi-LoRA support.

[6] Philip Bontrager, Wending Lin, Julian Togelius, and Sebastian Risi. 2018. Deep interactive evolution. In *International Conference on Computational Intelligence in Music, Sound, Art and Design*. Springer, 267–282.

[7] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. 2011. D^3: Data-Driven Documents. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (2011), 2301–2309. doi:10.1109/TVCG.2011.185

[8] Tomas Cabezon Pedroso and Daragh Byrne. 2023. Browsing the latent space: A new approach to interactive design exploration for volumetric generative systems. In *Proceedings of the 15th Conference on Creativity and Cognition*. 330–333.

[9] Joel Chan, Katherine Fu, Christian Schunn, Jonathan Cagan, Kristin Wood, and Kenneth Kotovsky. 2011. On the Benefits and Pitfalls of Analogies for Innovative Design: Ideation Performance Based on Analogical Distance, Commonness, and Modality of Examples. *Journal of Mechanical Design* 133, 8 (2011). doi:10.1115/1.4004396

[10] Siddhartha Chaudhuri, Evangelos Kalogerakis, Stephen Giguere, and Thomas Funkhouser. 2013. AttribIt: content creation with semantic attributes. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*. ACM, 193–202. doi:10.1145/2501988.2502008

[11] Erin Cherry and Celine Latulipe. 2014. Quantifying the creativity support of digital tools through the creativity support index. *ACM Transactions on Computer-Human Interaction (TOCHI)* 21, 4 (2014), 1–25.

[12] DaEun Choi, Sumin Hong, Jeongeon Park, John Joon Young Chung, and Juho Kim. 2024. Creativeconnect: Supporting reference recombination for graphic design ideation with generative ai. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–25.

[13] John Joon Young Chung and Max Kreminski. 2024. Patchview: Llm-powered worldbuilding with generative dust and magnet visualization. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology*. 1–19.

[14] CivitAI. 2023. CivitAI: Open-Source Generative AI Platform. https://civitai.com. Community platform hosting 100,000+ LoRA style models.

[15] Richard Lee Davis, Thiemo Wambsganss, Wei Jiang, Kevin Gonyop Kim, Tanja Käser, and Pierre Dillenbourg. 2024. Fashioning creative expertise with generative AI: Graphical interfaces for design space exploration better support ideation than text prompts. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–26.

[16] Richard Dawkins. 1996. *The blind watchmaker: Why the evidence of evolution reveals a universe without design*. WW Norton & Company.

[17] Anil R Doshi and Oliver P Hauser. 2024. Generative AI enhances individual creativity but reduces the collective diversity of novel content. *Science advances* 10, 28 (2024), eadn5290.

[18] Steven P. Dow, Alana Glassco, Jonathan Kass, Melissa Schwarz, Daniel L. Schwartz, and Scott R. Klemmer. 2010. Parallel prototyping leads to better design results, more divergence, and increased self-efficacy. *ACM Transactions on Computer-Human Interaction* 17, 4 (2010), 1–24. doi:10.1145/1879831.1879836

[19] Figma, Inc. 2026. Figma Weave. https://figma.com/solutions/figma-ai-tool-weave. Accessed August 2026.

[20] Flora, Inc. 2026. Flora: An Intelligent Canvas for Creative Professionals. https://flora.ai. Accessed August 2026.

[21] Yarden Frenkel, Yael Vinker, Ariel Shamir, and Daniel Cohen-Or. 2024. Implicit style-content separation using b-lora. In *European Conference on Computer Vision*. Springer, 181–198.

[22] Katherine Fu, Joel Chan, Jonathan Cagan, Kenneth Kotovsky, Christian Schunn, and Kristin Wood. 2013. The Meaning of "Near" and "Far": The Impact of Structuring Design Databases and the Effect of Distance of Analogy on Design Output. *Journal of Mechanical Design* 135, 2 (2013). doi:10.1115/1.4023158

[23] Tovi Grossman, Justin Matejka, and George Fitzmaurice. 2010. Chronicle: capture, exploration, and playback of document workflow histories. In *Proceedings of the 23rd Annual ACM Symposium on User Interface Software and Technology*. ACM, 143–152. doi:10.1145/1866029.1866054

[24] Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. 2023. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. *arXiv preprint arXiv:2309.08532* (2023).

[25] Yuhan Guo, Hanning Shao, Can Liu, Kai Xu, and Xiaoru Yuan. 2024. PrompTHis: Visualizing the Process and Influence of Prompt Editing during Text-to-Image Creation. *IEEE Transactions on Visualization and Computer Graphics* (2024). doi:10.1109/TVCG.2024.3408255

[26] Ole Hall and Anil Yaman. 2024. Collaborative interactive evolution of art in the latent space of deep generative models. In *International Conference on Computational Intelligence in Music, Sound, Art and Design (Part of EvoStar)*. Springer, 194–210.

[27] Evans Xu Han, Alice Qian Zhang, Haiyi Zhu, Hong Shen, Paul Pu Liang, and Jane Hsieh. 2025. POET: Supporting Prompting Creativity and Personalization with Automated Expansion of Text-to-Image Generation. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*. 1–18.

[28] Björn Hartmann, Loren Yu, Abel Allison, Yeonsoo Yang, and Scott R Klemmer. 2008. Design as exploration: creating interface alternatives through parallel authoring and runtime tuning. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*. 91–100.

[29] Scarlett R. Herring, Chia-Chen Chang, Jesse Krantzler, and Brian P. Bailey. 2009. Getting inspired!: understanding how and why examples are used in creative design practice. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 87–96. doi:10.1145/1518701.1518717

[30] John H Holland. 1992. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press.

[31] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. 2022. Lora: Low-rank adaptation of large language models. *Iclr* 1, 2 (2022), 3.

[32] Alex Inch, Passawis Chaiyapattanaporn, Yuchen Zhu, Yuan Lu, Ting-Wen Ko, and Davide Paglieri. 2025. Evolve to Inspire: Novelty Search for Diverse Image Generation. In *NeurIPS 2025 Workshop on Generative and Protective AI for Content Creation (GenProCC)*. arXiv:2511.00686.

[33] David G Jansson and Steven M Smith. 1991. Design fixation. *Design studies* 12, 1 (1991), 3–11.

[34] Hyeonsu B Kang, David Chuan-En Lin, Yan-Ying Chen, Matthew K Hong, Nikolas Martelaro, and Aniket Kittur. 2025. Biospark: Beyond analogical inspiration to llm-augmented transfer. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–29.

[35] Yuki Koyama, Issei Sato, Daisuke Sakamoto, and Takeo Igarashi. 2017. Sequential line search for efficient visual design optimization by crowds. *ACM Transactions on Graphics* 36, 4 (2017), 1–11. doi:10.1145/3072959.3073598

[36] Krea, Inc. 2026. Krea 2: Style References. https://krea.ai. Accessed August 2026.

[37] Chisato Kumada, Satoru Hiwa, and Tomoyuki Hiroyasu. 2026. Diffusion Crossover: Defining Evolutionary Recombination in Diffusion Models via Noise Sequence Interpolation. arXiv:2604.14790. arXiv:2604.14790

[38] Seung Won Lee, Yejin Yun, and Kyung Hoon Hyun. 2026. DesignLoRA: Interactive Fine-Tuning of Text-to-Image Models with Designer-in-the-Loop Feedback. *Advanced Engineering Informatics* 76 (2026), 105053. doi:10.1016/j.aei.2026.105053

[39] Leonardo Interactive Pty Ltd. 2026. Leonardo.Ai: Elements. https://leonardo.ai. Accessed August 2026.

[40] Jianye Li, Sheng Zhang, and Li Sun. 2025. Enhancing Product Concept Image Generation through Semantic Feature Prompts and LoRA Training. *Scientific Reports* (2025). doi:10.1038/s41598-025-24600-9

[41] David Chuan-En Lin, Fabian Caba Heilbron, Joon-Young Lee, Oliver Wang, and Nikolas Martelaro. 2024. VideoMap: Supporting Video Exploration, Brainstorming, and Prototyping in the Latent Space. In *Proceedings of the 16th Conference on Creativity & Cognition*. 311–327.

[42] David Chuan-En Lin, Hyeonsu B Kang, Nikolas Martelaro, Aniket Kittur, Yan-Ying Chen, and Matthew K Hong. 2025. Inkspire: Supporting design exploration with generative ai through analogical sketching. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–18.

[43] Songkai Liu, Yanqing Shen, Yilun Zhang, Zhangli Hou, Xin Wang, Jianxi Luo, and Zhinan Zhang. 2026. iDesignGPT enhances conceptual design via large language model agentic workflows. *Nature Communications* 17 (2026). doi:10.1038/s41467-026-68672-1

[44] Yuwen Lu, Alan Leung, Amanda Swearngin, Jeffrey Nichols, and Titus Barik. 2025. Misty: Ui prototyping through interactive conceptual blending. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–17.

[45] Michael Luo, Justin Wong, Brandon Trabucco, Yanping Huang, Joseph E. Gonzalez, Zhifeng Chen, Ruslan Salakhutdinov, and Ion Stoica. 2024. Stylus: Automatic Adapter Selection for Diffusion Models. In *Advances in Neural Information Processing Systems (NeurIPS '24)*.

[46] Kevin Ma, Daniele Grandi, Christopher McComb, and Kosa Goucher-Lambert. 2024. Exploring the Capabilities of Large Language Models for Generating Diverse Design Solutions. arXiv:2405.02345 [cs.HC] http://arxiv.org/abs/2405.02345

[47] Gary Marchionini. 2006. Exploratory search: from finding to understanding. *Commun. ACM* 49, 4 (2006), 41–46. doi:10.1145/1121949.1121979

[48] Joe Marks, Brad Andalman, Paul A. Beardsley, William Freeman, Sarah Gibson, Jessica Hodgins, Thomas Kang, Brian Mirtich, Hanspeter Pfister, Wheeler Ruml, Kathy Ryall, Joshua Seims, and Stuart Shieber. 1997. Design Galleries: A General Approach to Setting Parameters for Computer Graphics and Animation. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '97)*. ACM Press/Addison-Wesley Publishing Co., 389–400. doi:10.1145/258734.258887

[49] Nicolai Marquardt, Asta Roseway, Hugo Romat, Payod Panda, Michel Pahud, Gonzalo Ramos, Steven M Drucker, Andrew D Wilson, Ken Hinckley, and Nathalie Riche. 2025. ImaginationVellum: Generative-AI Ideation Canvas with Spatial Prompts, Generative Strokes, and Ideation History. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*. 1–19.

[50] Tuna Han Salih Meral, Enis Simsar, Federico Tombari, and Pinar Yanardag. 2025. Contrastive test-time composition of multiple LoRA models for image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 18090–18100.

[51] Midjourney. 2024. Midjourney. https://www.midjourney.com. AI image generation platform.

[52] Ziheng Ouyang, Zhen Li, and Qibin Hou. 2025. K-lora: Unlocking training-free fusion of any subject and style loras. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 13041–13050.

[53] Xiaohan Peng, Janin Koch, and Wendy E. Mackay. 2024. DesignPrompt: Using Multimodal Interaction for Design Exploration with Generative AI. In *Proceedings of the 2024 ACM Designing Interactive Systems Conference (DIS '24)*. ACM. doi:10.1145/3643834.3661588

[54] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML '21) (PMLR, Vol. 139)*.

[55] Recraft, Inc. 2026. Recraft: Brand Style Consistency and Control. https://recraft.ai. Accessed August 2026.

[56] Guanyu Ren, Yuxiang Li, Zhenguo Liu, and Chen Change Loy. 2025. Cached Multi-LoRA Composition for Multi-Concept Image Generation. *arXiv preprint arXiv:2502.04923* (2025).

[57] Scenario, Inc. 2026. Scenario: Compose Models. https://scenario.com/features/compose-models. Accessed August 2026.

[58] Jacob Schrum, Jake Gutierrez, Vanessa Volz, Jialin Liu, Simon Lucas, and Sebastian Risi. 2020. Interactive Evolution and Exploration Within Latent Level-Design Space of Generative Adversarial Networks. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO '20)*. ACM, 148–156. doi:10.1145/3377930.3389821

[59] Jimmy Secretan, Nicholas Beato, David B. D'Ambrosio, Adelein Rodriguez, Adam Campbell, Jeremiah T. Folsom-Kovarik, and Kenneth O. Stanley. 2011. Picbreeder: A Case Study in Collaborative Evolutionary Exploration of Design Space. *Evolutionary Computation* 19, 3 (2011), 373–403. doi:10.1162/EVCO_a_00030

[60] Hanshu Shen, Lyukesheng Shen, Wenqi Wu, and Kejun Zhang. 2025. IdeationWeb: Tracking the Evolution of Design Ideas in Human-AI Co-Creation. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. ACM. doi:10.1145/3706598.3713375

[61] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. 2020. Interpreting the Latent Space of GANs for Semantic Face Editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR '20)*.

[62] Ben Shneiderman. 2007. Creativity support tools: accelerating discovery and innovation. *Commun. ACM* 50, 12 (2007), 20–32. doi:10.1145/1323688.1323689

[63] Joel Simon. 2018. Artbreeder. https://www.artbreeder.com.

[64] Karl Sims. 1991. Artificial Evolution for Computer Graphics. In *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '91)*. ACM, 319–328. doi:10.1145/122718.122752

[65] Jochen Suessmuth, Florian Fick, and Stan van der Vossen. 2023. Generative AI for Concept Creation in Footwear Design. In *ACM SIGGRAPH 2023 Talks*. ACM. doi:10.1145/3587421.3595416

[66] Sangho Suh, Meng Chen, Bryan Min, Toby Jia-Jun Li, and Haijun Xia. 2024. Luminate: Structured Generation and Exploration of Design Space with Large Language Models for Human-AI Co-Creation. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (CHI '24)*. ACM. doi:10.1145/3613904.3642400

[67] Sangho Suh, Michael Lai, Kevin Pu, Steven P Dow, and Tovi Grossman. 2025. StoryEnsemble: Enabling dynamic exploration & iteration in the design process with AI and forward-backward propagation. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology*. 1–36.

[68] Hideyuki Takagi. 2001. Interactive Evolutionary Computation: Fusion of the Capabilities of EC Optimization and Human Evaluation. *Proc. IEEE* 89, 9 (2001), 1275–1296. doi:10.1109/5.949485

[69] Sirui Tao, Ivan Liang, Cindy Peng, Zhiqing Wang, Srishti Palani, and Steven P Dow. 2025. DesignWeaver: dimensional scaffolding for text-to-image product design. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–26.

[70] Carlos Tejeda-Ocampo, Armando López-Cuevas, and Hugo Terashima-Marín. 2021. Improving Deep Interactive Evolution with a Style-Based Generator for Artistic Expression and Creative Exploration. *Entropy* 23, 1 (2021), 11. doi:10.3390/e23010011

[71] Michael Terry and Elizabeth D. Mynatt. 2002. Recognizing creative needs in user interface design. In *Proceedings of the Fourth Conference on Creativity & Cognition*. ACM, 38–44. doi:10.1145/581710.581718

[72] Vizcom. 2024. Vizcom: AI-Powered Industrial Design Tool. https://www.vizcom.ai. AI rendering tool for industrial designers.

[73] Samangi Wadinambiarachchi, Ryan M. Kelly, Saumya Pareek, Qiushi Zhou, and Eduardo Velloso. 2024. The Effects of Generative AI on Design Fixation and Divergent Thinking. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems (CHI '24)*. ACM. doi:10.1145/3613904.3642919

[74] Qian Wan and Zhicong Lu. 2023. GANCollage: A GAN-Driven Digital Mood Board to Facilitate Ideation in Creativity Support. In *Proceedings of the 2023 ACM Designing Interactive Systems Conference (DIS '23)*. ACM. doi:10.1145/3563657.3596072

[75] Wen-Fan Wang, Ting-Ying Lee, Chien-Ting Lu, Che-Wei Hsu, Nil Ponsa i Campanyà, Yu Chen, Mike Y. Chen, and Bing-Yu Chen. 2025. GenTune: Toward Traceable Prompts to Improve Controllability of Image Refinement in Environment Design. In *Proceedings of the 38th Annual ACM Symposium on User Interface Software and Technology (UIST '25)*. ACM. doi:10.1145/3746059.3747774

[76] Wen-Fan Wang, Chien-Ting Lu, Nil Ponsa Campanyà, Bing-Yu Chen, and Mike Y. Chen. 2025. AIdeation: Designing a Human-AI Collaborative Ideation System for Concept Designers. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems (CHI '25)*. ACM. doi:10.1145/3706598.3714148

[77] Yanan Wang and Yan Pei. 2024. A Comprehensive Survey on Interactive Evolutionary Computation in the First Two Decades of the 21st Century. *Applied Soft Computing* 164 (2024), 111950. doi:10.1016/j.asoc.2024.111950

[78] Mehmet Ersin Yumer, Siddhartha Chaudhuri, Jessica K. Hodgins, and Levent Burak Kara. 2015. Semantic shape editing using deformation handles. *ACM Transactions on Graphics* 34, 4 (2015), 1–12. doi:10.1145/2766908

[79] Kaiwen Zhang, Yifan Zhou, Xudong Xu, Bo Dai, and Xingang Pan. 2024. DiffMorpher: Unleashing the Capability of Diffusion Models for Image Morphing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR '24)*.

[80] Ming Zhong, Yelong Shen, Shuohang Wang, Yadong Lu, Yizhu Jiao, Siru Ouyang, Donghan Yu, Jiawei Han, and Weizhu Chen. 2024. Multi-lora composition for image generation. *arXiv preprint arXiv:2402.16843* (2024).

[81] Mingxu Zhou, Dengming Zhang, Weitao You, Ziqi Yu, Yifei Wu, Chenghao Pan, Huiting Liu, Tianyu Lao, and Pei Chen. 2024. StyleFactory: Towards Better Style Alignment in Image Creation through Style-Strength-Based Control and Evaluation. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology (UIST '24)*. ACM. doi:10.1145/3654777.3676370

A Technical Validation

GENErator's genetic operations are only useful if the underlying style space is well-structured and the operations perform as intended. In this section, we validate three properties: that the embedding groups similar styles together, that similar and distant mutations move in the intended directions, and that natural selection converges reliably toward a target aesthetic.

A.1 Style Embedding Quality

All genetic operations depend on the underlying embedding space placing similar styles nearby and dissimilar styles far apart. Without this, similar mutation would draw from unrelated styles and distant mutation would be indistinguishable from random. A more straightforward alternative to our content-independent style embedding approach is to encode each style's trigger words directly as an embedding using CLIP [54]. However, trigger words are often ambiguous: "chrome" conflates the material with the color and the web browser, and "glow" and "opalescent" share few linguistic traits despite producing similar luminous visual effects.

To test whether content-independent style embedding improves this, we identified 10 groups of aesthetically similar genes (e.g., {`Chrome`, `Glossy`, `Mirror`, `Aluminum Foil`}, {`Glow`, `Neon`, `Bioluminescent`, `Opalescent`}) and measured average pairwise cosine similarity within each group. Our content-independent style embedding achieves $M = 0.66$ ($SD = 0.06$) within-group similarity as compared to $M = 0.36$ ($SD = 0.13$) for text encoding, an 83% improvement in the tightness of clusters ($t(9) = 7.33$, $p < .01$, $d_z = 2.32$). Our technique produces tighter clusters in 9 of 10 groups. This means that when similar mutation draws from the nearest 10% of genes, it can more reliably return styles in the same aesthetic family. The similar-mutation threshold (τ_{90}, Section 3.4) is also approximately 0.66, so genes in the similar pool are about as close to each other as genes within these groups.

We also test whether the text-derived space tracks visual similarity. For each pair of genes, we compare cosine similarity in the style embedding space against cosine similarity between CLIP image embeddings of an example image of each gene. The two agree ($\rho = 0.34$, $p < .001$ across 4,560 gene pairs), even though each example image depicts a different object, which adds object-level noise to the image similarities. The most similar pairs in the text space are also among the most visually similar (e.g., `Frosted Glass` and `Glass`, `Art Deco` and `Art Nouveau`).

A.2 Semantic Directionality of Mutations

For similar and distant mutations to serve different roles, similar-mutation targets should land closer to the source than random, and distant targets should land farther. We verified this by measuring cosine distance from each of the 100 genes to its similar pool (nearest 10%), distant pool (farthest 10%), and a random selection.

The similar pool stays closest ($M = 0.31, SD = 0.04$), 24% closer as compared to random ($t(99) = 32.2$, $p < .01$, $d_z = 3.22$). The distant pool lies farthest ($M = 0.52$, $SD = 0.04$), 27% farther as compared to random ($t(99) = 47.6$, $p < .01$, $d_z = 4.76$). Random falls in the middle ($M = 0.41$, $SD = 0.04$). Overall, distant targets land 1.7 times farther from the source than similar targets ($p < .01$). For example, `Chrome`'s similar pool yields `Metallic` (similar reflective surface), while its distant pool yields `Denim` (completely different texture). This separation is what lets designers control how far to push from their current direction rather than relying only on random mutation [68, 77].

A.3 Natural Selection Convergence

Natural selection is only useful if successive generations actually move toward the designer's target rather than drifting. We ran 10 evolution sessions targeting diverse aesthetics ("luxurious," "futuristic," "minimal," "organic," "brutalist," etc.) with a population of 5 and 5 generations each. The fitness score (semantic similarity between the generated image and the target description) increases monotonically from $M = 0.14$ ($SE = 0.01$) at generation 0 to $M = 0.87$ ($SE = 0.02$) at generation 5, with no session showing fitness regression (Figure 12). This consistency comes from drawing each generation's mutation candidates from genes semantically related to the target description rather than selecting randomly from the full library. Convergence held across all 10 aesthetics (e.g., from "minimal" to "brutalist"), suggesting the mechanism generalizes well across the style space.

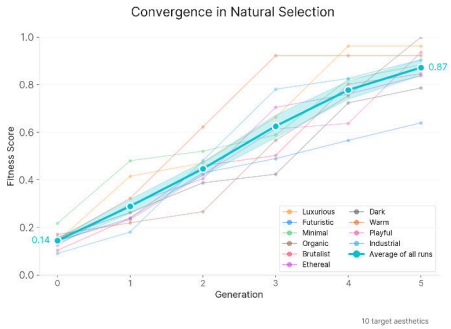


Figure 12: Natural selection convergence across 10 target aesthetics. Average fitness score increases monotonically from 0.14 to 0.87 over 5 generations, with no session showing regression. The bold teal line shows the average across all runs.

B Additional Figures

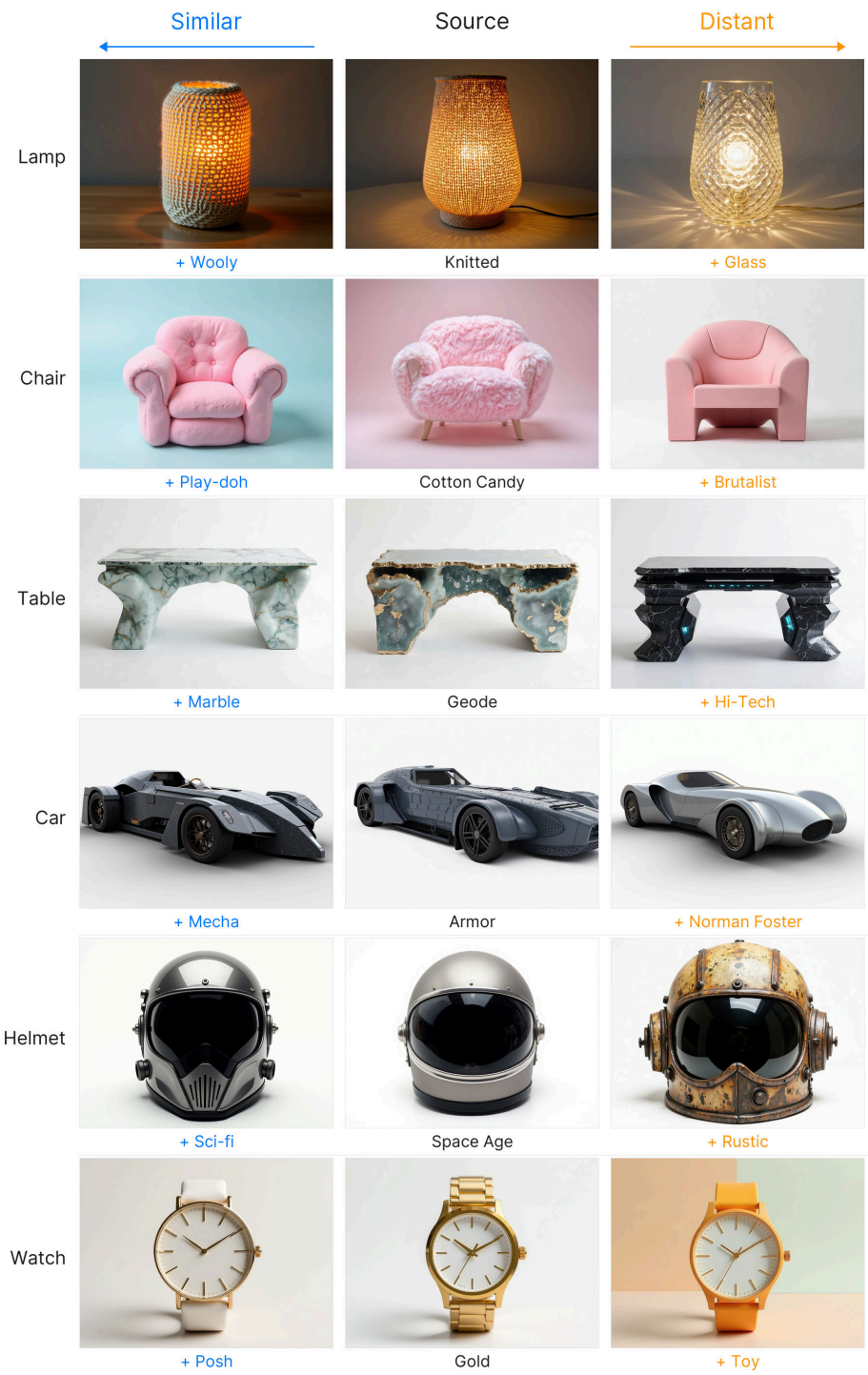


Figure 13: Similar vs. distant mutation across six products. Each row shows a source gene (center) with its similar-mutation offspring (left) and distant-mutation offspring (right).

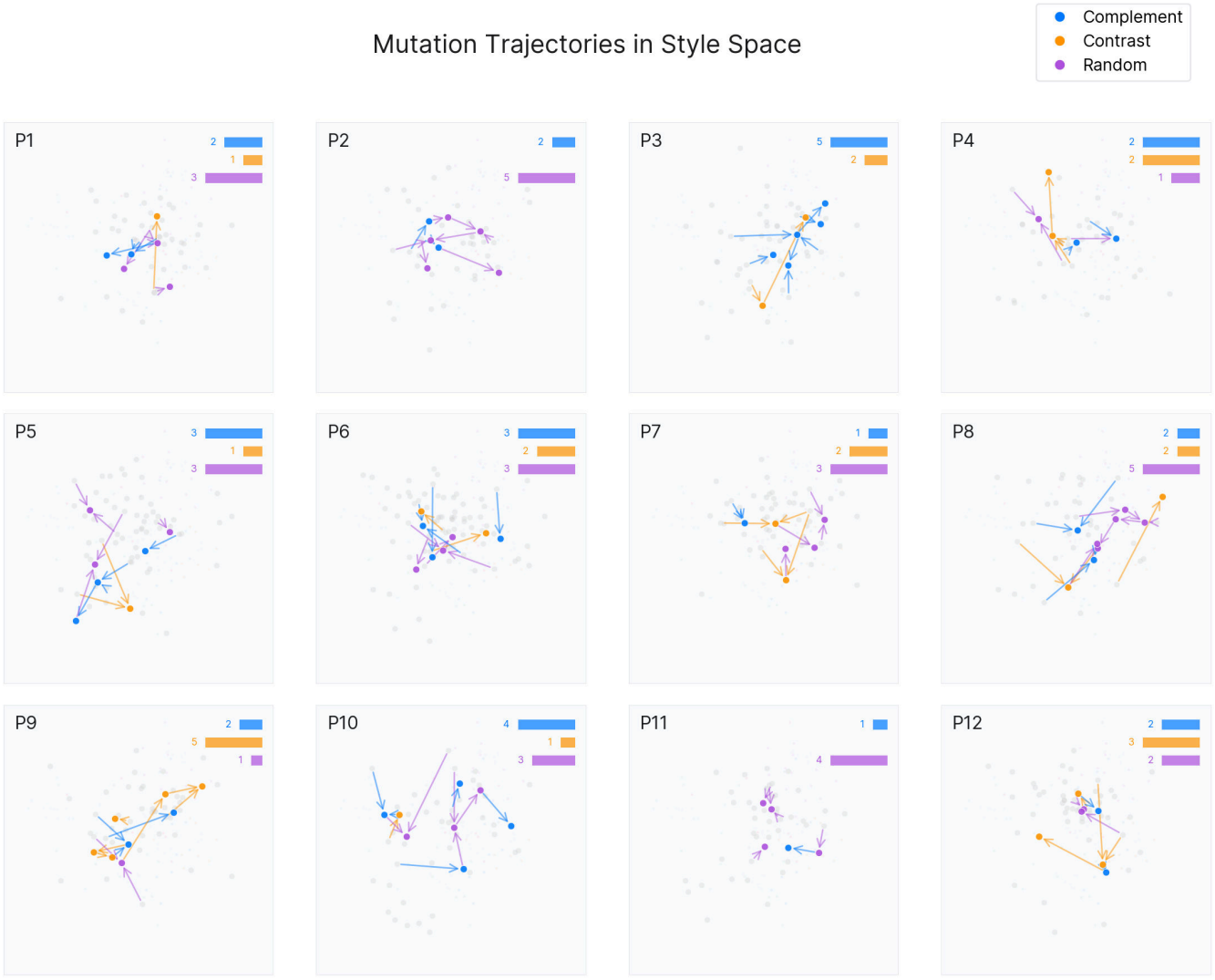


Figure 14: Mutation trajectories per participant in PCA-projected style space. Each arrow connects a source gene to its mutation offspring. Similar mutations (blue) generally produce shorter arrows that stay near the source and distant mutations (orange) generally produce longer arrows that jump to farther regions. Stacked bars (top right of each subplot) show each participant's mutation mode distribution, indicating that most participants used all three modes.

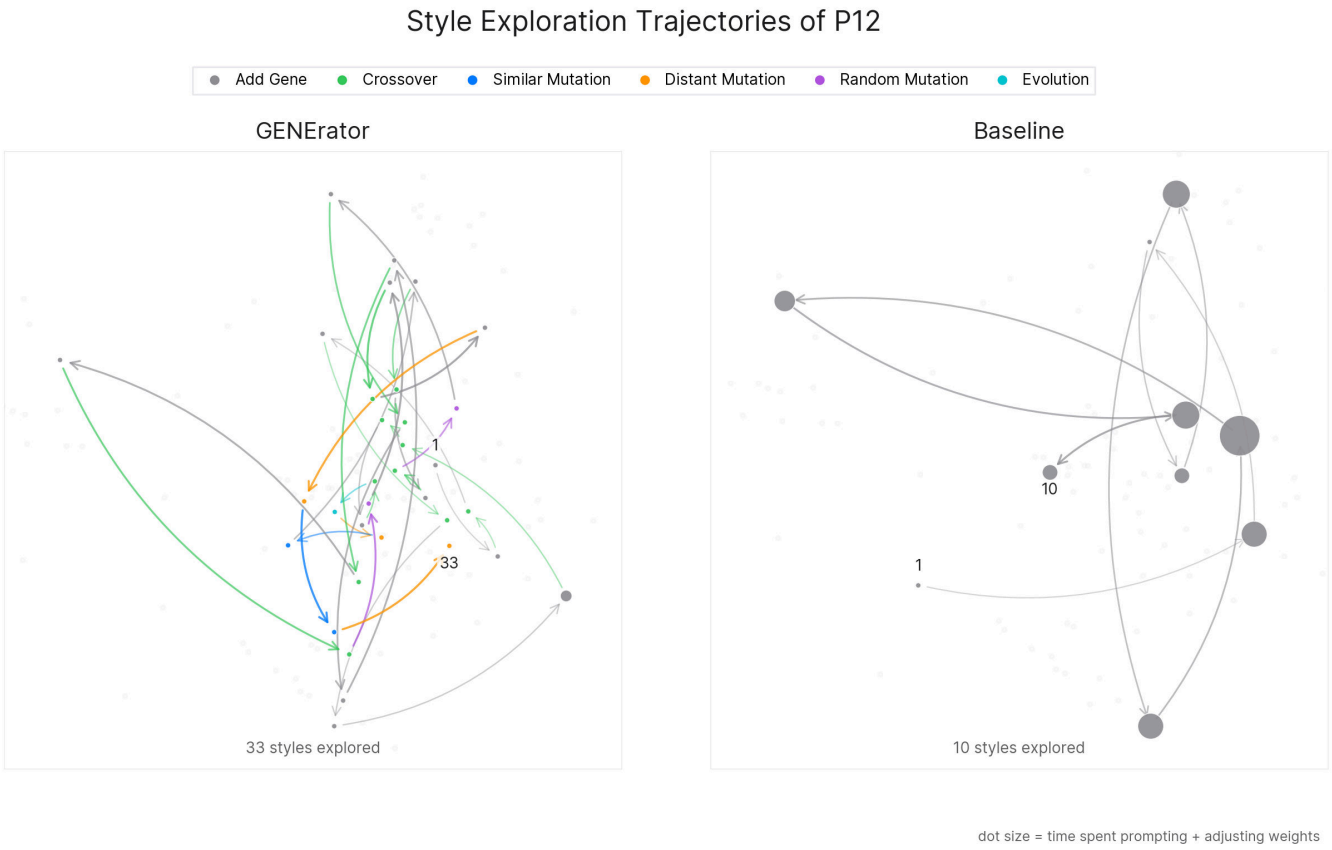


Figure 15: Style exploration trajectories for P12 in PCA-projected style space. Dot size encodes time spent prompting and adjusting weights with the same gene combination. GENErator (left, 33 styles) shows rapid movement across the space through diverse operations (crossover, similar, distant, evolution), with small dots indicating frequent transitions between styles. Baseline (right, 10 styles) shows fewer, larger dots concentrated in a small region, with repeated loops back to the same gene combinations, a characteristic of the fixation pattern where designers spend more time tuning familiar styles than discovering new ones.

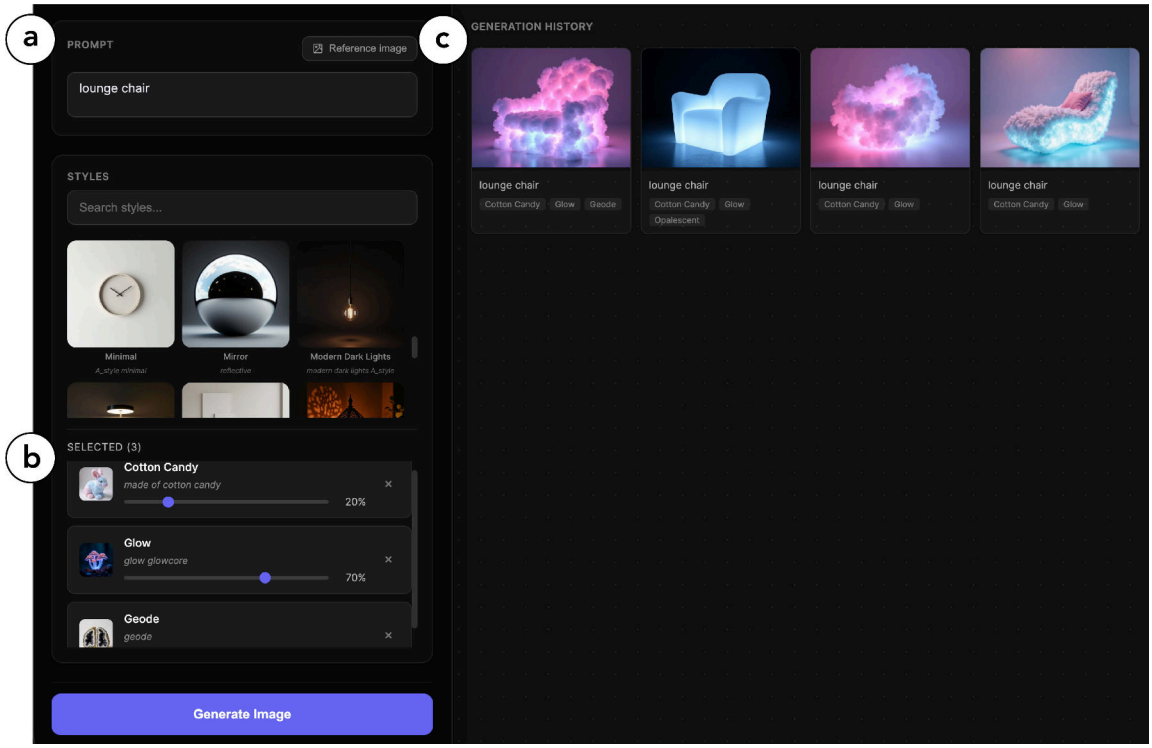


Figure 16: The baseline interface. Designers select styles from the gene library (a), adjust weight sliders to control each style's influence (b), and generate images. A generation history grid (c) displays all previous outputs. Clicking any previous result restores its selected styles and weights, enabling one-click backtracking. This provides stronger backtracking than most real-world tools, which require designers to manually recreate previous configurations.

Category	Genes
Material (19)	Aluminum Foil, Bamboo, Bronze, Carbon Fiber, Chrome, Copper Wire, Frosted Glass, Glass, Glossy, Gold Porcelain, Leather, Marble, Matte Black Gold, Mirror, Plastic, Polymer, Reptile Scales, Satin, Wooden
Texture (18)	Beads, Blue and White Porcelain, Chocolate, Cotton Candy, Denim, Embroidery, Fluffy, Geode, Ginger Bread, Knitted, Lego, Moroccan, Pearls, Play-doh, Plush, Taihu, Traditional Chinese, Wooly
Movement (14)	Art Deco, Art Nouveau, Avant-garde, Bohemian, Felipe Pantone, Neoliteratism, Posh, Rococo, Rustic, Steampunk, Streetwear, Techwear, Upcycling Fashion, Vintage
Light & Color (13)	Bioluminescent, Colorful Toy, Crystals, Glitter, Glow, Hard Shadows, Hologram, Icy, Modern LED Lights, Monochrome, Neon, Opalescent, Stained Glass
Mood (18)	Armor, Cinematic, Coal, Cyberpunk, Cyborg, Dark Fantasy, Ethereal, Glitch, Graffiti, Isometric, Lava, Liquid Black, Mecha, Sci-fi, Smoke, Space Age, Vaporwave, Y2K
Architecture (18)	3D Retro, Antoni Gaudi, Bjarke Ingels, Brutalist, Francesco Borromini, Hi-Tech, Industrial, Luxury Gilded, Michael Graves, Minimal, Modern Minimal, Norman Foster, Organic, Symmetric, Victor Horta, Vincent Callebaut, Zaha Hadid, Zen

Table 3: The 100 style genes in the library grouped by category.

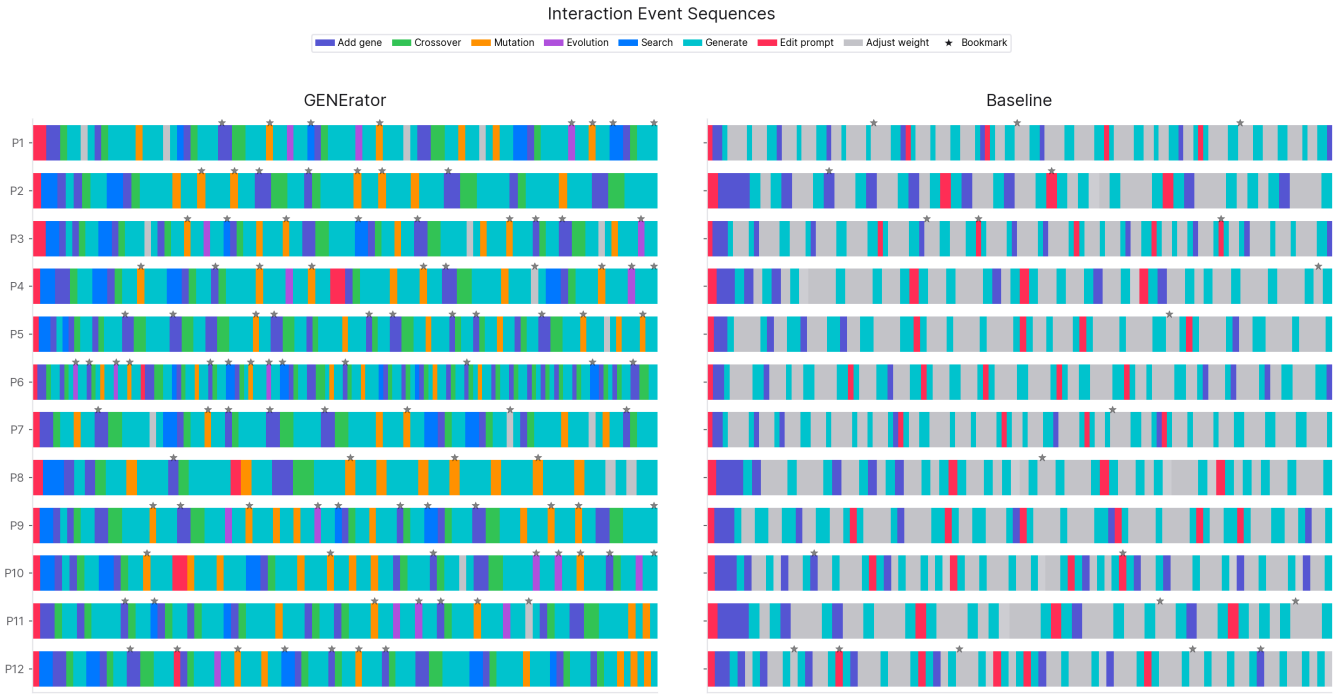


Figure 17: Interaction event sequences for all 12 participants. Baseline (right) is dominated by trial-and-error weight adjustments and manual prompting. GENErator (left) shows a diverse mix of crossover, mutation, evolution, and semantic search operations across all participants. More designs are bookmarked (favorited) when using GENErator.