Axes-and-Tags: LLM-Driven Design Galleries for Generative Content

Asanshay Gupta* Vishnu Sarukkai* Kayvon Fatahalian
Stanford University
{asanshay, sarukkai}@stanford.edu, kayvonf@cs.stanford.edu

Abstract

We introduce Axes-and-Tags, a domain-agnostic framework for structured exploration of LLM-driven generative design spaces. Current interfaces for generative content typically trap users in unstructured trial-and-error cycles with complex prompts, offering little insight into how variations relate to underlying dimensions. Our approach transforms unstructured generation into systematic exploration by dynamically identifying interpretable axes of variation and guiding users through one dimension at a time using semantic tags rather than complex prompts. The key innovation is a tag-first generation strategy where semantic descriptors precede content creation, ensuring variations occur along intended dimensions while maintaining consistency across others. Our architecture separates domain-agnostic exploration logic from domain-specific rendering, allowing Axes-and-Tags to generalize across diverse tasks—diffusion-generated images, HTML/CSS websites, and text composition—with minimal adaptation. User feedback indicates this method of structured exploration eases the cognitive load of exploring from a blank canvas.

Figure 1. **Structured exploration of generating an image of the letters "HIGEN" using Axes-and-Tags.** The user begins with the request 'The letters "HIGEN" made of objects' and progressively refines the design by exploring one axis at a time. First, they explore 'Object Material' options, selecting 'letters made of stone objects' (green border). With material fixed, they explore 'Background Setting,' choosing 'a natural forest background with dense green foliage.' Next, they explore 'Lighting Type' variations while maintaining consistency in the stone material and forest background, selecting 'low-key dramatic lighting.' Finally, they explore 'Color Scheme' options, selecting 'pastel colors' for the final design.

1. Introduction

The emergence of large language models (LLMs) has fundamentally transformed the landscape of computational design. Unlike traditional parameterized tools where designers navigate fixed parameter spaces [22, 31], generative models now enable the creation of diverse content—images, websites, text—through dynamic, text-based representations that implicitly encode complex design spaces spanning style, structure, color, semantics, and composition [2, 4, 10, 35]. These systems have demonstrated remarkable capabilities in domains such as text-to-image synthesis [27, 28], web design [8], and text generation [1, 6], yet the exploration of their generative potential remains challenging—particularly in ways that generalize across different design domains.

Design processes are inherently iterative [11, 31], requiring designers to generate alternatives against evolving objectives. Effective interfaces must support rapid generation and refinement of options [32, 33]. Yet current generative interfaces predominantly offer single-input/single-output interactions, forcing inefficient trial-and-error cycles. While some systems present multiple outputs [4, 10, 36], they force users to interact with prompts themselves, rather than building abstractions exposing higher-level dimensions of variation.

We introduce Axes-and-Tags, a framework that structures generative model output into an interpretable exploration process. Our key insight is that LLMs can dynami-

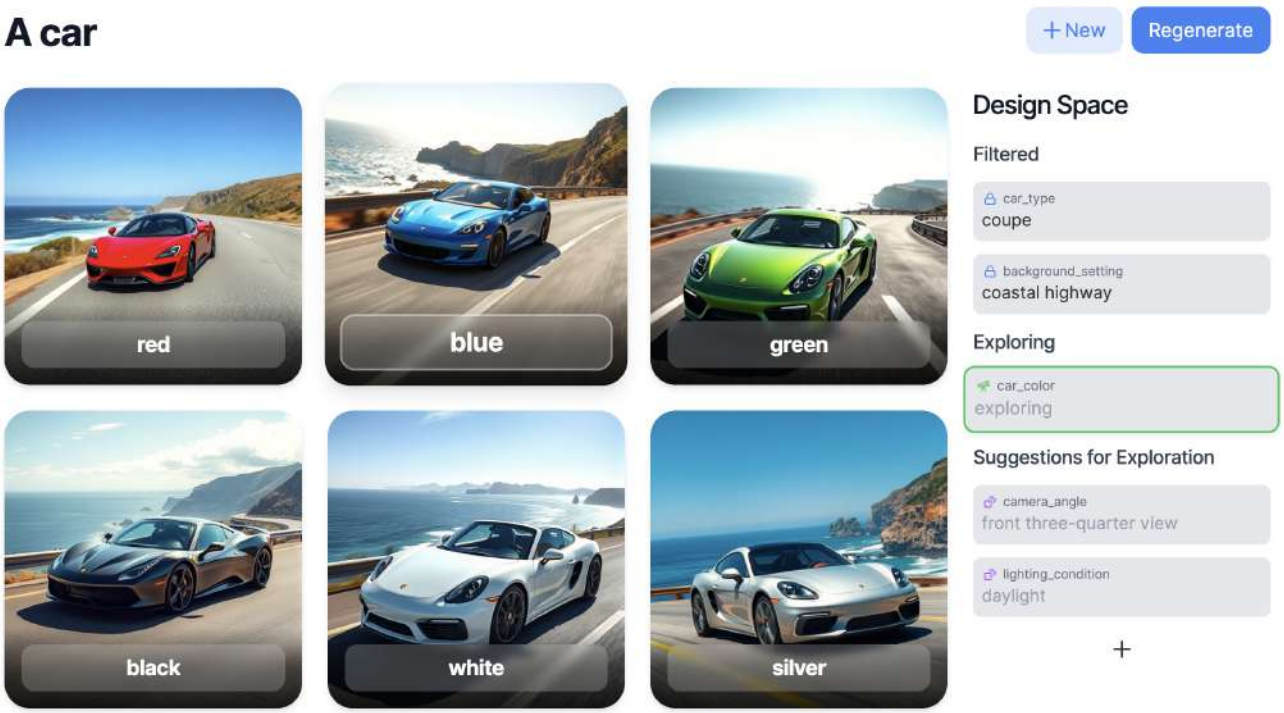

Figure 2. **Axes-and-Tags, facilitating text-to-image generation given the user request 'A car'.** The system has already identified meaningful axes of variation and the user has made selections for two dimensions: 'car type' (coupe) and 'background setting' (coastal highway), shown in the 'Filtered' section of the right panel. The left panel displays a gallery of six images with variation along the currently explored axis 'car color' (e.g., 'red,' 'blue,' 'green,' 'black,' 'white,' 'silver'), each labeled with its corresponding tag. The right panel shows the system state: currently selected preferences, the axis being explored (car_color), and suggestions for future exploration including 'camera angle' and 'lighting condition' with default values. Users can select any image to choose that color preference and advance to exploring another axis, or directly manipulate the exploration state through the right panel. The 'New' and 'Regenerate' buttons at the top allow starting a new design task or refreshing the current gallery. This structured approach enables systematic exploration of the design space while maintaining consistency in previously selected dimensions.

cally identify meaningful dimensions of variation and guide users through systematic exploration using concise semantic tags rather than complex prompts. Unlike predetermined parameters in traditional interfaces [19, 22], Axes-and-Tags leverages domain-agnostic prompting for core exploration functions (axis identification, tag generation, preference tracking), with only a simple domain-specific layer to map semantic specifications to appropriate outputs.

Given an initial user request, Axes-and-Tags employs LLMs to: (1) identify interpretable axes of variation, (2) select one axis to explore while holding others fixed, (3) generate diverse semantic tags along this axis, and (4) produce content samples from these tags while respecting user preferences from previous iterations. Users provide feedback by selecting preferred tags, prompting the system to fix that dimension and explore another axis.

Our contributions include:

1. **Axes-and-Tags, structured exploration framework with domain-agnostic core components.** We present a novel approach that transforms unstructured generative capabilities into structured design exploration using a unified interaction model, bridging the gap between natural language flexibility and systematic exploration of traditional design tools.
2. **LLM-powered mechanisms for domain-independent dimension discovery and tag-first generation.** We develop techniques that leverage LLMs to extract meaningful dimensions, generate diverse semantic tags, and produce content samples while respecting user preferences—using identical prompt templates regardless of design domain.
3. **Multi-domain implementation of Axes-and-Tags.** We implement our framework across three distinct generative tasks—image generation, HTML/CSS website creation, and text composition—where the only domain-specific component is a simple prompt template mapping semantic specifications to appropriate outputs. We demonstrate that Axes-and-Tags allows users to explore in a more structured way, increasing breadth and depth concurrently.

Axes-and-Tags demonstrates how LLMs can guide structured design exploration, combining natural language flexibility and semantic tag simplicity with systematic exploration capabilities of parameter-based interfaces—while maintaining separation between domain-agnostic exploration logic and domain-specific output generation. Here's a more concise version that preserves the meaning:

2. Related Work

Design processes typically progress through concept creation, iteration, and fine-tuning [11, 31]. Axes-and-Tags supports this by enabling structured exploration through interpretable axes of variation.

Design Galleries and Parameterized Exploration Design Gallery [22] pioneered gallery-based interfaces by sampling parameter spaces and arranging results for comparison. Later approaches improved efficiency using preference models [5, 20], active learning [16, 18], or genetic algorithms [34]. However, these systems restrict users to visible designs and often require tool-switching. Axes-and-Tags dynamically generates galleries based on user feedback, enabling continuous exploration without precomputation or fixed parameter spaces.

Parameter Manipulation and Re-parameterization While sliders offer precise control [7, 33], they become tedious in high-dimensional spaces. Re-parameterization transforms complex spaces into meaningful dimensions. Semantic attributes enable manipulation through human-understandable labels in faces [3], materials [29], shapes [9, 38], and typography [24]. Recent approaches use deep learning to discover semantic directions in generative models' latent spaces [15, 26, 30], but require offline training on large datasets. Axes-and-Tags leverages LLMs to identify interpretable axes on-the-fly, eliminating pre-computation while providing structured exploration.

Structured Exploration in Creative Design Tools Design Adjectives [31] learns personalized attributes from interactions, while ShadowDraw [21] and Dream Lens [23] guide exploration through visual feedback. Axes-and-Tags differs by explicitly exposing semantic dimensions rather than implicitly learning preferences or requiring textual refinement. This transparency helps users build accurate mental models [13, 14] and supports intentional creative direction.

Design Interfaces for Generative Models Generative models have transformed design interfaces. Text-to-image tools [4, 36] help craft prompts for diffusion models, while text composition tools like Wordcraft [37] support interactive writing. Most systems treat prompts as opaque objects [4, 36] or require manual specification of parts of prompts to modify [10]. Axes-and-Tags automatically identifies semantic axes across domains and supports exploration through named axes with explicit tag-based generation rather than direct prompt manipulation. By integrating design galleries with modern generative capabilities, Axes-and-Tags transforms unstructured content generation into structured exploration, bridging natural language flexibility and systematic exploration regardless of output type.

Algorithm 1 LLM-Guided Exploration Framework

Require: Initial user request r

$A \leftarrow \textsc{IdentifyAxes}(r)$; $T \leftarrow \emptyset$
while design task continues **do**
  $A_{free} \leftarrow \{a \in A \mid \nexists t \in T : t \text{ is associated with } a\}$
  $a_{sel} \leftarrow \textsc{SelectAxis}(A_{free})$
  $T_{var} \leftarrow \textsc{GenTags}(a_{sel}, T)$
  $T_{temp} \leftarrow \textsc{GenTempTags}(A_{free} \setminus \{a_{sel}\})$
  $S \leftarrow \emptyset$
  for each tag $t \in T_{var}$ **do**
    $T_{complete} \leftarrow T \cup \{t\} \cup T_{temp}$
    $s \leftarrow \textsc{TagsToContent}(r, T_{complete})$
    $S \leftarrow S \cup \{(s, t)\}$
  $u \leftarrow \textsc{GetUserAction}()$
  if u is tag selection **then**
    $T \leftarrow T \cup \{u\}$
  else if u is axis selection **then**
    $a_{sel} \leftarrow u$
  else if u is state edit **then**
    Update A and/or T according to edit

3. Method

3.1. System Overview

Axes-and-Tags maintains two core state components: (1) a set of interpretable axes $A = \{a_1, a_2, ..., a_n\}$ with each axis being a concise phrase (e.g., 'color palette,' 'room style') representing a semantic dimension, and (2) a set of preferred tags $T = \{t_1, t_2, ..., t_m\}$, where each tag t_i is a brief descriptor (e.g., 'minimalist,' 'warm sunset') associated with a specific axis. This state evolves through user interactions and LLM-guided exploration as outlined in Alg. 1.

3.2. Working with Tags Instead of Prompts

A fundamental design decision in Axes-and-Tags is the use of semantic tags rather than direct prompt manipulation. Modern generative AI prompts are complex and verbose, creating significant cognitive load for users trying to understand how specific changes affect outputs.

Figure 3. **Tags streamline interaction with prompts.** The user is exploring the 'background type' axis with three different options. On the bottom, we show the complete text-to-image prompts where the words corresponding to each tag are highlighted in bold. Note how a simple tag like 'lush garden' expands to influence multiple phrases across the prompt ('outdoors in a lush garden', 'vibrant green foliage...', 'serene and peaceful atmosphere'). Our interface allows users to interact with these concise semantic tags rather than editing complex prompts directly.

Our tag-based approach encapsulates this complexity behind simple, human-readable descriptors. Fig. 3 illustrates this by comparing raw text-to-image prompts to corresponding tags. The tag 'lush garden' expands to influence multiple phrases across the prompt, but users interact only with the semantic concept. The tradeoff is that users cannot manipulate prompts directly themselves–trading controllability for ease of use. In this paper, we make this tradeoff given the prior work on LLMs themselves being human-level prompt engineers [17, 39] and on the effectiveness of automatic prompt expansion for text-to-image generation [12].

3.3. Dynamic Axis Identification and Selection

Unlike traditional systems with fixed dimensions, Axes-and-Tags dynamically identifies relevant axes specific to each design task. Given a user request, we query the LLM to analyze the design and identify semantically meaningful dimensions (Alg. 1, L1). For example, given 'a logo for a coffee shop,' Axes-and-Tags might identify axes such as 'color palette,' 'iconography,' 'typography,' and 'composition style.'

At each step, Axes-and-Tags selects an untagged axis a_{sel} from A_{free} for exploration (Alg. 1, L4). This selection can be automatic or user-directed. The axis identification

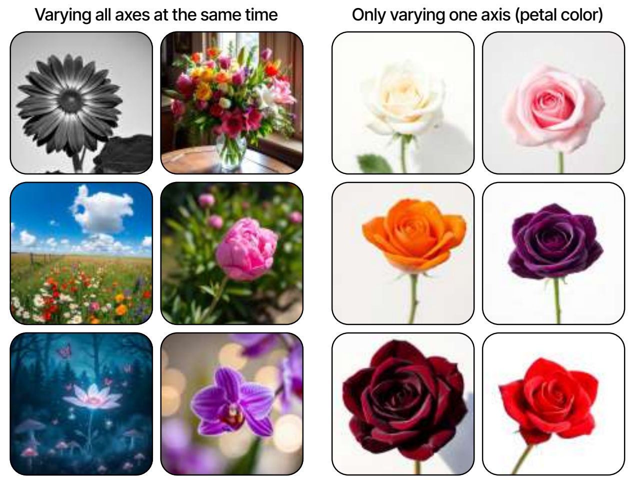

Figure 4. **Varying a single axis at a time helps users navigate the design space.** Here, we take a look at a user using Axes-and-Tags to generate an image of a flower. Left: When multiple dimensions vary simultaneously (flower type, background, lighting, style, composition), users cannot easily attribute visual differences to specific dimensions. Right: Our approach varies only one dimension ('petal color'') while keeping all other dimensions fixed. This focused variation enables users to clearly understand the impact of the selected dimension and make informed decisions.

and selection prompts are entirely domain-agnostic, functioning identically whether the task involves image generation, website creation, or text composition.

3.4. Tag-First Generation Approach

A key insight in Axes-and-Tags is generating semantic tags before generating content. This ensures variation occurs primarily along the intended dimension while respecting previous preferences.

3.4.1. Diverse Tag Generation

For the selected axis a_{sel}, Axes-and-Tags generates diverse tags T_{var} representing different positions along this dimension (Alg. 1, L5). These tags capture meaningful variation along a_{sel} while being compatible with existing preference tags T. For example, if a_{sel} represents 'art style' for a mountain landscape, T_{var} might include 'photorealistic,' 'impressionist,' and 'minimalist vector art.'

The tag generation process uses domain-agnostic prompting that encourages diversity while maintaining coherence with the design request. The same prompt templates are used regardless of output type.

3.4.2. Tag-Conditioned Content Generation

Once tags T_{var} are generated, Axes-and-Tags produces specifications for each sample, consisting of:

1. The initial user request r

2. The current set of preferred tags T for dimensions with established preferences
3. One tag from T_{var} for the currently explored axis
4. Temporary tags for all remaining untagged dimensions

By generating temporary tags for dimensions that are neither being explored nor have established preferences, we ensure variation occurs only along the intended axis.

The fully-tagged specification is then passed to the only domain-specific component: a 'rendering' prompt that translates the semantic tags into the appropriate output format—a text-to-image prompt, HTML/CSS code, or formatted text (Alg. 1, L10). This separation between domain-agnostic exploration logic and domain-specific output rendering allows Axes-and-Tags to generalize across diverse tasks with minimal adaptation.

3.5. Interface Design and User Interaction

Our interface (Fig. 2) employs a two-panel design. The left panel displays a gallery of six samples showing variation along the current dimension, with corresponding tags prominently displayed. The right panel provides a complete view of the current state, with axes organized by status (selected preferences, current exploration, future options).

Axes-and-Tags supports several interaction modes:

1. **Tag Selection**: Users can select any sample to add its tag as a preference, fixing that dimension and advancing to another axis.
2. **Axis Navigation**: Users can select which axis to explore next, edit axis names, or add new axes.
3. **Preference Management**: Users can modify, add or remove both axes and tags to adjust preferences.

Each interaction updates the state, triggering a new generation cycle that respects updated preferences. This creates a feedback loop where users iteratively refine their design one dimension at a time while maintaining control over the exploration process. This interaction model remains consistent across all generative domains.

4. Experimental Setup

We evaluate by implementing Axes-and-Tags for three generative design domains, and examine the results both via sample walkthroughs in Sec. 5 (with additional examples in Apps. A to C) and via user feedback in Sec. 6. We implemented Axes-and-Tags for the following three domains:

1. **Image Generation:** A system for exploring text-to-image diffusion models. Since effective prompts typically require complex, multi-paragraph instructions with specific stylistic directives [12, 25], Axes-and-Tags abstracts this complexity through semantic tags. We use FLUX.1 Schnell, a 12B parameter diffusion model, which provides an approximately 2 second latency for image generation.
2. **Website Design:** A system for exploring LLM-generated web interfaces. Users can create landing pages, payment portals, portfolios, etc. by navigating dimensions such as layout structure, color scheme, and component style. The system generates HTML with Tailwind CSS via Claude Sonnet 3.7, achieving a latency of 10 seconds per generation.
3. **Text Composition:** A system for exploring written content generation. Users can create emails, product descriptions, marketing copy, etc. by exploring dimensions like tone, formality, and structure. Content is generated using GPT-4.1-mini, resulting in a latency of 4 seconds per generation.

These domains represent fundamentally different output modalities and user objectives, yet our framework applies a consistent interaction model across all three, with only the final tags-to-content stage adapted to each medium.

4.1. Implementation Details

We use GPT-4.1-mini for all core functions (axis identification, tag generation, semantic processing) with temperature 0.7 to balance deterministic reasoning with creative diversity. The interface was implemented as a simple HTML and JS app with a Python FastAPI backend to orchestrate LLM calls and maintain state.

Our implementation follows a modular architecture where domain-agnostic components (axis identification, tag generation, state management) remain identical across domains, while domain-specific modules handle only the final translation from semantic specifications to output format. This separation enables Axes-and-Tags to generalize across different generative tasks with minimal adaptation. Complete prompt templates are provided in App. F.1.

5. Results

We demonstrate Axes-and-Tags through examples across three domains: image generation (Figs. 5 and 6, more in App. A), website design (Fig. 7, more in App. B), and text composition (Fig. 8, more in App. C). Each figure shows the step-by-step exploration process, highlighting our key contributions: explicit parameterization through semantic axes, focused single-dimension variation, and tag-based preference specification.

Explicit Parameterization through Semantic Axes Axes-and-Tags automatically identifies meaningful dimensions specific to each domain and design task. In Fig. 5, the system identifies dimensions relevant to truck imagery: 'truck type,' 'truck color,' 'background environment,' and 'time of day.' For bedroom design in Fig. 6, it extracts 'bed style,' 'wall color,' 'bedding color,' and 'decor theme.' The website design in Fig. 7 is parameterized along 'layout

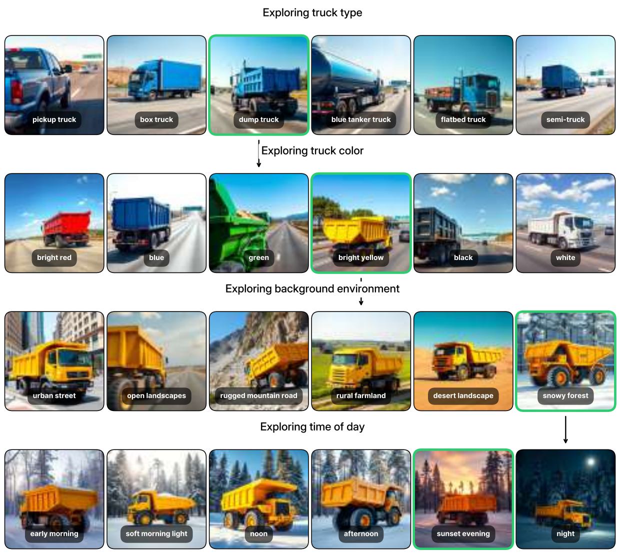

Figure 5. **Generating a truck image via Axes-and-Tags** Starting with 'truck type,' the user selects 'dump truck' (highlighted in green), fixing this preference for subsequent explorations. The system then presents variations along 'truck color,' where the user selects 'bright yellow.' With type and color established, exploration continues to 'background environment,' where 'snowy forest' is selected. Finally, the system offers variations along 'time of day,' with the user selecting 'night.' Green borders indicate the user's selections at each step.

Figure 6. **Progressive refinement of generating a bedroom image through our axis-and-tag approach.** The user first explores 'bed style' options and selects 'minimalist platform bed' (green border). With the bed style fixed, the system presents 'wall color' variations while maintaining the selected bed, and the user chooses 'slate blue.' Exploration continues to 'bedding color,' where 'warm terracotta accents' is selected, and finally to 'decor theme,' where 'minimalist' is chosen. This demonstrates our system's ability to enable focused, interpretable exploration along individual dimensions while preserving user preferences across iterations.

style,' 'color palette,' 'typography style,' and 'navigation placement,' while the email composition in Fig. 8 is structured along 'formality level,' 'reason for extension,' 'length of email,' and 'tone of request.'

This explicit parameterization helps users develop a mental model of the design space and enables systematic exploration. Rather than treating generation as a black box, users can see which dimensions are available for exploration and how they relate to their design goals. For example, in Fig. 6, the separation of 'wall color' from 'bedding color' allows the user to make targeted decisions about each aspect independently. As these axes are generated using LLMs, there are sporadic failure cases involving extraneous axes being generated. For example, when generating text for the prompt 'A name for an AI agent startup', the LLM assumes we are creating a full brand identity and suggests 'brand color' as an axis.

Focused Variation Along Single Dimensions By varying only one dimension at a time while holding others constant, Axes-and-Tags enables clear understanding of each dimension's impact. This focused variation is evident across all examples. In Fig. 5, when exploring 'background environment,' the dump truck maintains its yellow color while only the surroundings change from construction site to desert road to snowy forest. Similarly, in Fig. 7, when exploring 'navigation placement,' the previously selected monospace typography and complementary color palette remain consistent across all samples.

The tag-first generation approach is essential for achieving this focused variation. By generating temporary tags for dimensions not yet explored, we ensure consistency where needed. In Fig. 8, this allows exploring 'tone of request' (varying from 'apologetic' to 'confident' to 'humble') while maintaining the previously selected formal style, family emergency reason, and moderate length. Without this approach, changing one aspect might inadvertently affect others, making it difficult for users to understand cause and effect. Occasionally, the tag of a value constrained is not strict enough, which may lead to unexpected variation among an axis, like in the case of 'dramatic lighting' which could mean a variety of lighting techniques.

Tag-Based Preference Specification Our tag-based approach enables interpretable feedback and consistent application of user choices throughout the design process. In each figure, green highlights indicate selected tags that become fixed preferences for subsequent explorations. For ex-

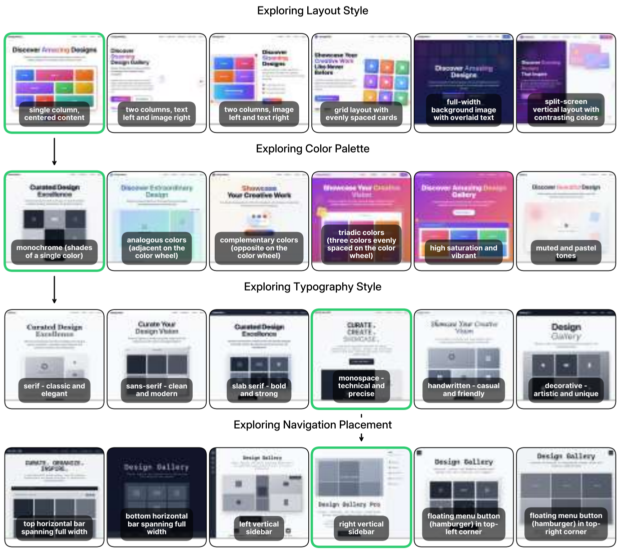

Figure 7. **Progressive refinement of a design gallery website through axis-by-axis exploration.** Each row shows the system exploring a different dimension while maintaining consistency with previous selections (highlighted in green). The exploration begins with 'Layout Style,' offering options from single-column to grid layouts. After selecting 'monospace - technical and precise' from the 'Typography Style' options in the third row, the system explores 'Navigation Placement' in the bottom row. This sequential exploration enables users to understand how each dimension affects the design independently, while the tag-based approach ensures that only the intended dimension varies in each step.

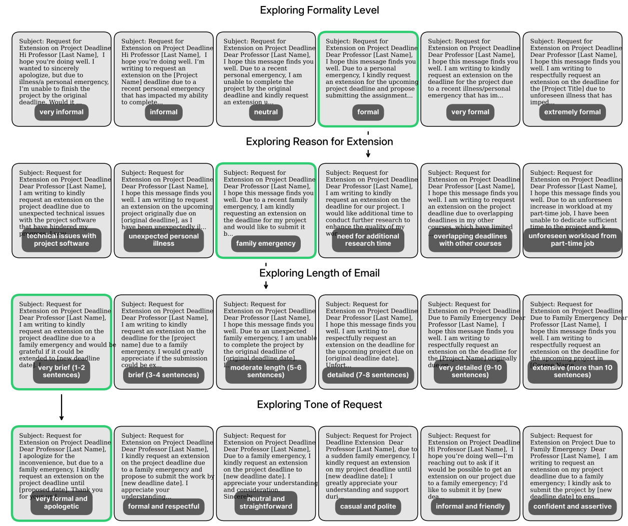

Figure 8. **Structured exploration of an email requesting a deadline extension across multiple axes.** Each row shows the system exploring a different semantic dimension while preserving preferences from previous selections (highlighted in green). The top row explores 'Formality Level' with options ranging from 'very informal' to 'extremely formal.' After the user selects 'formal,' the system explores 'Reason for Extension' in the second row, presenting options like 'technical issues,' 'unexpected illness,' and 'family emergency.' With 'family emergency' selected, the third row varies 'Length of Email' from 'very brief' to 'extensive.' Finally, the bottom row explores 'Tone of Request' while maintaining all previous selections.

ample, in Fig. 6, after selecting 'minimalist platform bed' for the bed style, this preference persists through all subsequent explorations of wall color, bedding color, and decor theme.

This approach provides transparency not available in traditional generative interfaces. Users can clearly see which decisions they've made and how those decisions constrain the remaining design space. The ability to revisit earlier decisions is also valuable. For instance, if after exploring several dimensions in Fig. 5, the user wanted to change from a dump truck to a pickup truck, they could simply modify that tag and the system would regenerate samples that maintain other preferences (yellow color, snowy background) while adopting the new truck type.

Through these examples across multiple domains, we demonstrate that Axes-and-Tags enables structured, interpretable exploration of generative design spaces by explicitly parameterizing dimensions, focusing variation along one axis at a time, and providing a transparent mechanism for preference specification.

6. User Feedback

We invited 6 participants (4 female, 2 male) to explore design spaces using Axes-and-Tags for text-to-image generation. The participants varied in design background and experience with AI, with one participant having no experience with AI altogether. All other participants had prior experience using prompt-based generative AI tools, but none had used systems with explicit dimension-based exploration. Participants were given a brief introduction (approximately 5 minutes) to the system, focusing on the concepts of axes, tags, and structured exploration. All participants were able to understand the interface and start using the tool without further questions.

Participants were asked to complete 12 design tasks in image generation. For comparison, they also completed similar tasks using OpenAI ChatGPT image generation, which offers 2 options per prompt from the user. Figs. 9 and 10 illustrate how participants explored design spaces using our system, more traces are available in App. D. We captured all interaction logs, final designs, and took detailed notes documenting participants' comments and thought processes. We then analyzed this qualitative data

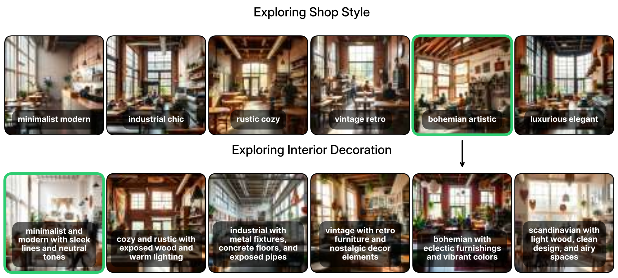

Figure 9. **A user exploring coffee shop design with Axes-and-Tags.** Having manually set the tag for the 'color scheme' axis, the user explores 'Shop Style' with options ranging from minimalist modern to luxurious elegant, selecting 'bohemian artistic' (green border). Then, with style fixed, exploration continues to 'Interior Decoration,' where the system generates variations that maintain the bohemian artistic style while exploring different decoration approaches. The user selects 'minimalist and modern with sleek lines and neutral tones' (green border), demonstrating how our approach enables exploring seemingly contradictory combinations (bohemian style with minimalist decoration) that might not be discovered through unstructured exploration.

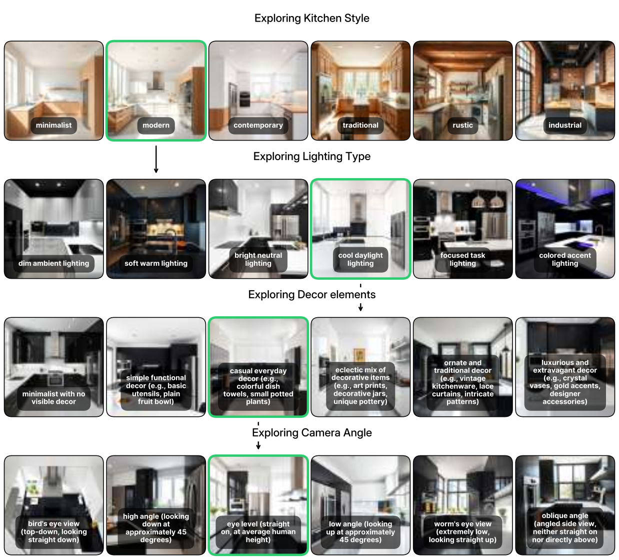

Figure 10. **A user exploring kitchen design using Axes-and-Tags.** The user progressively refines a kitchen design by exploring four axes sequentially. First, they select 'modern' kitchen style (green border) from options including minimalist, contemporary, and industrial. With style fixed, they explore 'lighting type,' choosing 'cool daylight lighting.' They then select 'casual everyday decor' from various decor options, and finally choose 'eye level' camera angle.

to extract insights into the user experience.

In the early stages of exploration, participants found the explicit axis representation valuable for understanding the available design dimensions. When participants had specific design preferences in mind, manually editing the axes helped quickly narrow the space of possibilities—see Fig. 9, where Participant 2 had specific notions about the color scheme for a kitchen and instantly locked them in. On the other hand, Participant 4 used the system in a much more experimental way, actually ending their journey early as they unexpectedly found an image they liked (see Fig. 10).

After exploring initial dimensions, participants typically focused on refining specific aspects of their designs. The tag-based preference specification allowed them to lock in certain decisions while continuing to explore others. For instance, Participant 4 selected a preferred typography style and then extensively explored color variations while maintaining the chosen typography—see Fig. 10.

Across domains, participants expressed appreciation for the variations presented. As one participant noted: 'It's really nice to see all the layouts laid out in front of me, as I often choose things that I didn't even think of before.' Many users particularly valued the structured approach, with one commenting: 'It feels like creating a character in a video game, where I can focus in on one stat at a time.'

6.1. Limitations

While Axes-and-Tags enables structured exploration, it relies on the model's ability to surface semantically meaningful and discrete axes, which can be inconsistent across domains or prompts. In some cases, generated axes may be redundant, overly narrow, or omit relevant concepts entirely. For example, the model will commonly create a 'camera angle axis', which is suboptimal for exploring the design of concrete objects like bedrooms. The system also assumes axis independence during traversal, which can lead to incoherent combinations when inherent dependencies exist between tags, like in the case of 'bedroom style' and 'interior decorations'. We allow users to directly edit axes and tags, which helps mitigate these limitations.

7. Conclusion

We presented Axes-and-Tags, a framework that offers structured exploration of generative design spaces via interpretable semantic dimensions. Our approach suggests that LLMs can serve not only as generative engines but also as guides for systematic design exploration across diverse domains. By identifying axes, enabling focused variation along one axis at a time, and providing a tag-based preference mechanism, our system addresses key usability challenges in generative interfaces. Future work could explore extending this approach to additional modalities like audio and 3D content, integrating more granular control, and developing automated evaluation metrics for tag-to-content fidelity. We believe that structured exploration frameworks like Axes-and-Tags represent an important step toward making powerful generative systems more accessible and useful for both novice and expert users.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1

[2] Shm Garanganao Almeda, JD Zamfirescu-Pereira, Kyu Won Kim, Pradeep Mani Rathnam, and Bjoern Hartmann. Prompting for discovery: Flexible sense-making for ai art-making with dreamsheets. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pages 1–17, 2024. 1

[3] Volker Blanz, Curzio Basso, Tomaso Poggio, and Thomas Vetter. Reanimating faces in images and video. In *Computer graphics forum*, pages 641–650. Wiley Online Library, 2003. 3

[4] Stephen Brade, Bryan Wang, Mauricio Sousa, Sageev Oore, and Tovi Grossman. Promptify: Text-to-image generation through interactive prompt exploration with large language models. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–14, 2023. 1, 3

[5] Eric Brochu, Tyson Brochu, and Nando De Freitas. A bayesian interactive optimization approach to procedural animation design. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 103–112, 2010. 3

[6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 1

[7] Stefan Bruckner and Torsten Möller. Result-driven exploration of simulation parameter spaces for visual effects design. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1468–1476, 2010. 3

[8] Tommaso Calò and Luigi De Russis. Leveraging large language models for end-user website generation. In *International Symposium on End User Development*, pages 52–61. Springer, 2023. 1

[9] Siddhartha Chaudhuri, Evangelos Kalogerakis, Stephen Giguere, and Thomas Funkhouser. Attribit: content creation with semantic attributes. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*, pages 193–202, 2013. 3

[10] DaEun Choi, Kihoon Son, HyunJoon Jung, and Juho Kim. Expandora: Broadening design exploration with text-to-image model. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pages 1–10, 2025. 1, 3

[11] Nigel Cross. *Design thinking: Understanding how designers think and work*. Bloomsbury Publishing, 2023. 1, 3

[12] Siddhartha Datta, Alexander Ku, Deepak Ramachandran, and Peter Anderson. Prompt expansion for adaptive text-to-image generation. *arXiv preprint arXiv:2312.16720*, 2023. 4, 5

[13] Steven P Dow, Alana Glassco, Jonathan Kass, Melissa Schwarz, Daniel L Schwartz, and Scott R Klemmer. Parallel prototyping leads to better design results, more divergence, and increased self-efficacy. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 17(4):1–24, 2010. 3

[14] John S Gero and Julie Milovanovic. A framework for studying design thinking through measuring designers' minds, bodies and brains. *Design Science*, 6:e19, 2020. 3

[15] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. Ganspace: Discovering interpretable gan controls. *Advances in neural information processing systems*, 33:9841–9850, 2020. 3

[16] Faisal Khan, Bilge Mutlu, and Jerry Zhu. How do humans teach: On curriculum learning and teaching dimension. *Advances in neural information processing systems*, 24, 2011. 3

[17] Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T Joshi, Hanna Moazam, et al. Dspy: Compiling declarative language model calls into self-improving pipelines. *arXiv preprint arXiv:2310.03714*, 2023. 4

[18] Vladlen Koltun. Exploratory modeling with collaborative design spaces. 2009. 3

[19] Yuki Koyama, Daisuke Sakamoto, and Takeo Igarashi. Crowd-powered parameter analysis for visual design exploration. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 65–74, 2014. 2

[20] Yuki Koyama, Issei Sato, and Masataka Goto. Sequential gallery for interactive visual design optimization. *ACM Transactions on Graphics (TOG)*, 39(4):88–1, 2020. 3

[21] Yong Jae Lee, C Lawrence Zitnick, and Michael F Cohen. Shadowdraw: real-time user guidance for freehand drawing. *ACM Transactions on Graphics (ToG)*, 30(4):1–10, 2011. 3

[22] Joe Marks, Brad Andalman, Paul A Beardsley, William Freeman, Sarah Gibson, Jessica Hodgins, Thomas Kang, Brian Mirtich, Hanspeter Pfister, Wheeler Ruml, et al. Design galleries: A general approach to setting parameters for computer graphics and animation. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 73–84. 2023. 1, 2, 3

[23] Justin Matejka, Michael Glueck, Erin Bradner, Ali Hashemi, Tovi Grossman, and George Fitzmaurice. Dream lens: Exploration and visualization of large-scale generative design datasets. In *Proceedings of the 2018 CHI conference on human factors in computing systems*, pages 1–12, 2018. 3

[24] Peter O'Donovan, Jānis Lībeks, Aseem Agarwala, and Aaron Hertzmann. Exploratory font selection using crowd-sourced attributes. *ACM transactions on graphics (TOG)*, 33 (4):1–9, 2014. 3

[25] Jonas Oppenlaender. A taxonomy of prompt modifiers for text-to-image generation. *Behaviour & Information Technology*, 43(15):3763–3776, 2024. 5

[26] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *Proceedings of the IEEE/CVF inter-*

national conference on computer vision, pages 2085–2094, 2021. 3

[27] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1 (2):3, 2022. 1

[28] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1

[29] Ana Serrano, Diego Gutierrez, Karol Myszkowski, Hans-Peter Seidel, and Belen Masia. An intuitive control space for material appearance. *arXiv preprint arXiv:1806.04950*, 2018. 3

[30] Yujun Shen, Jinjin Gu, Xiaoou Tang, and Bolei Zhou. Interpreting the latent space of gans for semantic face editing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9243–9252, 2020. 3

[31] Evan Shimizu, Matthew Fisher, Sylvain Paris, James McCann, and Kayvon Fatahalian. Design adjectives: a framework for interactive model-guided exploration of parameterized design spaces. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, pages 261–278, 2020. 1, 3

[32] Ben Shneiderman. Creativity support tools: accelerating discovery and innovation. *Communications of the ACM*, 50(12): 20–32, 2007. 1

[33] Michael Terry, Elizabeth D Mynatt, Kumiyo Nakakoji, and Yasuhiro Yamamoto. Variation in element and action: supporting simultaneous development of alternative solutions. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 711–718, 2004. 1, 3

[34] Luigi Troiano and Cosimo Birtolo. Genetic algorithms supporting generative design of user interfaces: Examples. *Information Sciences*, 259:433–451, 2014. 3

[35] Yael Vinker, Yuval Alaluf, Daniel Cohen-Or, and Ariel Shamir. Clipascene: Scene sketching with different types and levels of abstraction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4146–4156, 2023. 1

[36] Zhijie Wang, Yuheng Huang, Da Song, Lei Ma, and Tianyi Zhang. Promptcharm: Text-to-image generation through multi-modal prompting and refinement. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, pages 1–21, 2024. 1, 3

[37] Ann Yuan, Andy Coenen, Emily Reif, and Daphne Ippolito. Wordcraft: story writing with large language models. In *Proceedings of the 27th International Conference on Intelligent User Interfaces*, pages 841–852, 2022. 3

[38] Mehmet Ersin Yumer, Siddhartha Chaudhuri, Jessica K Hodgins, and Levent Burak Kara. Semantic shape editing using deformation handles. *ACM Transactions on Graphics (TOG)*, 34(4):1–12, 2015. 3

[39] Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers. In *The Eleventh International Conference on Learning Representations*, 2022. 4

A. Image generation examples

Prompts used for each generation are highlighted in **bold** at the beginning of each caption.

Figure 11. **Lunch.** Step-by-step exploration of a lunch image using Axes-and-Tags. The user begins by selecting a meal type (e.g., rice bowl), then progressively varies cuisine, protein, and plate style across separate gallery rows. Each axis is presented independently, and prior selections are preserved throughout, enabling controlled, interpretable variation.

Figure 12. **A house.** Starting with house style, the user continues to refine the image through roof type, number of stories, exterior color, house size, and landscape setting. The user then explores house style again, while retaining all other constrained tags

B. Website design examples

Figure 13. **A login form.** Design of a login UI using semantic axes. The system surfaces input field styles first, followed by button design, color scheme, and layout orientation. Users select options in each gallery while retaining earlier choices, enabling compositional control over the interface through visually grounded tags.

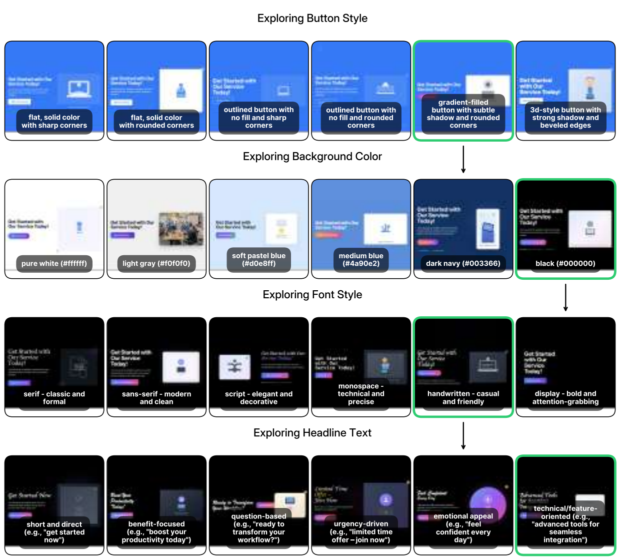

Figure 14. **A Call-to-Action section.** The user injects a brand-new axis—Headline Text—and the gallery instantly responds while preserving earlier choices for Button Style, Background Color, and Font Style. Axes-and-Tags is instantly extensible beyond the initial set of axes

C. Text generation examples

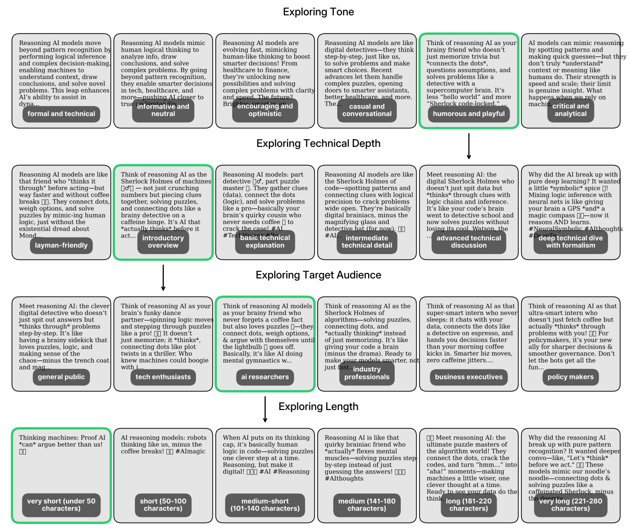

Figure 15. **A tweet about reasoning AI models.** The user begins by selecting a tone, and then explores axes for technical depth, target audience, and length. At each step, the gallery reflects changes along a single axis while holding others constant, supporting precise stylistic iteration in text generation.

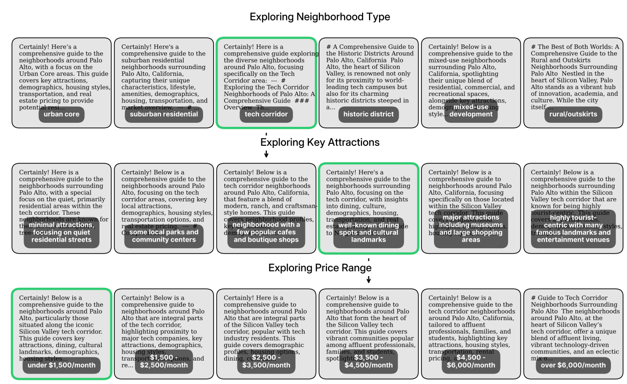

Figure 16. **A guide to all neighborhoods around Palo Alto.** The user selects a neighborhood type (e.g., residential) and then proceeds to refine the text through key attractions and price range. Each axis introduces content-relevant tags that scaffold the output, with prior tags retained across steps.

D. User feedback examples

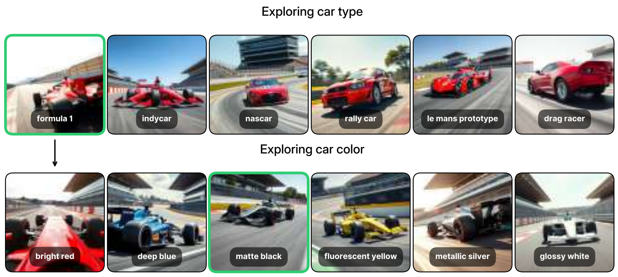

Figure 17. **A racecar.** Initial exploration of a car image. The user first selects a car type then explores variations in car color while keeping the body style fixed. The broad exploration range leads to a very quick convergence to a satisfactory image.

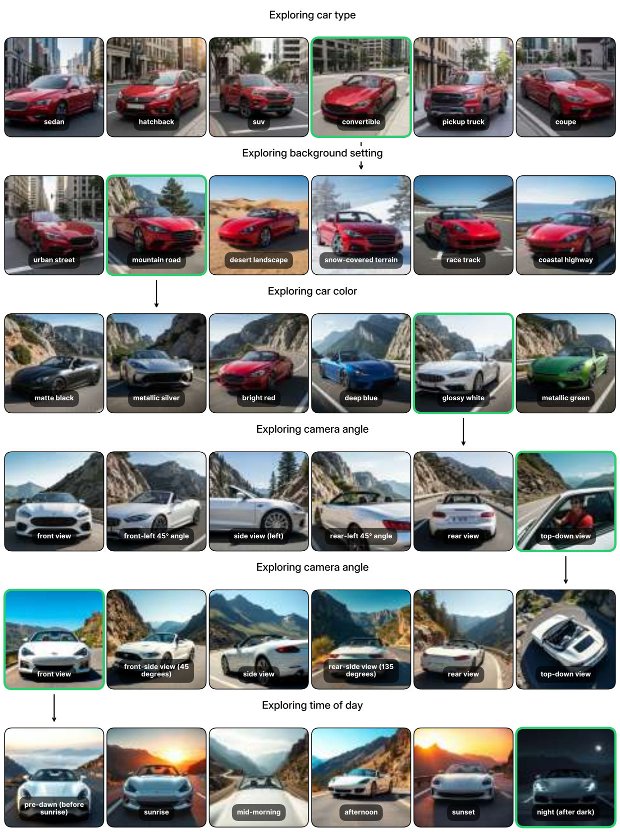

Figure 18. **A car.** The user iteratively varies a wide variety of axes, including background setting, car color, camera angle, and time of day. Axes-and-Tags can scale the number of axes indefinitely.

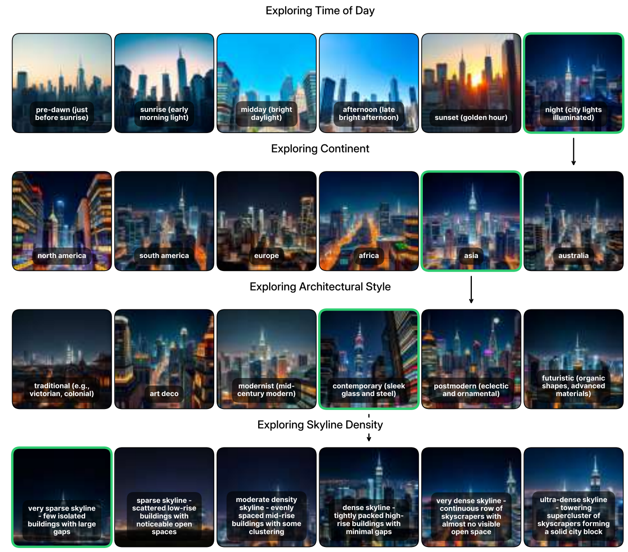

Figure 19. **A city skyline.** Starting with time of day, the user proceeds through continent, architectural style, and skyline density.

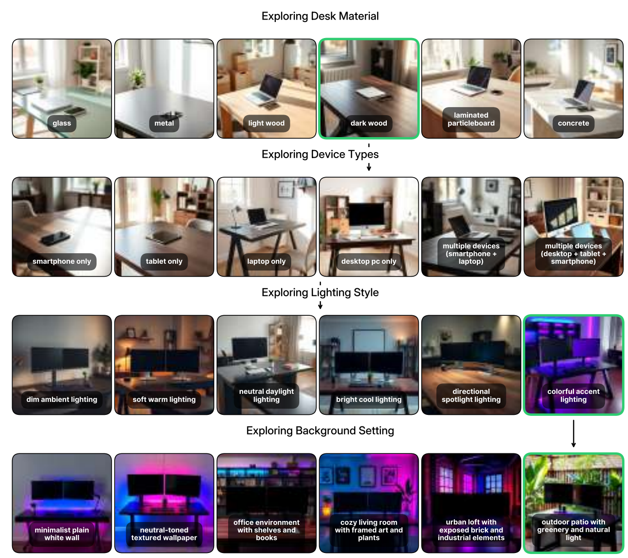

Figure 20. **A desk setup.** The selection of axes in this image mimic real design decisions made when setting up a desk - desk material, choice of tech, lighting, etc.

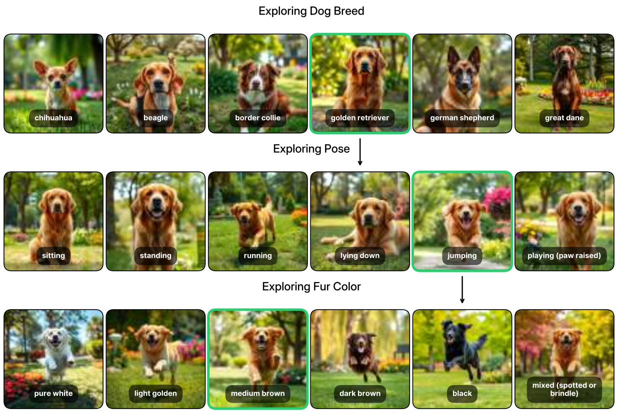

Figure 21. **A dog.** Even while the main axis is varied, the environment in which the dog is placed is kept consistent between generations.

E. User Feedback Details

Participants had the opportunity to generate images both via our interface and via ChatGPT. For using our interface, all users were provided with the following text as a tutorial before running experiments:

> This short guide walks you through Design Galleries - an interface for rapidly exploring large creative spaces. In just a few clicks you'll learn how to move through ideas, control axes, and regenerate fresh variations.
>
> **1: Enter a prompt & pick a domain**
> On the home page, describe what you'd like to explore (for example 'a futuristic city at dusk') and select a generation domain such as Images, UIs, or Text. Click Generate to create the first gallery of possibilities.
>
> **2: Pick a value for the current axis**
> Hover any preview to reveal a tag representing the value being explored along the current design axis. Click the tag (or the preview itself) to constrain the design space to that value.
>
> **3: Choose the next axis to explore**
> In the Design Space panel you can set any axis to Exploring, Unconstrained, or Constrained. Pick a new axis to explore next by clicking the telescope icon. Want to explore something that isn't listed yet? Click the + Add Axis button to define a brand-new dimension and provide a few example values.
>
> **4: Regenerate for fresh results**
> Hit the Regenerate button to sample new designs that honor your constraints while introducing variation along the axis you've chosen to explore.

Participants were provided with 12 randomly sampled initial prompts from this list of common objects. A car, A dog, A helmet, A house, A living room, A kitchen, A bedroom, A desk setup. A flower, A beach, A mountain, A city skyline, A cat, A coffee shop, A jacket, A race car, A floating island, A bird.

F. Implementation details

F.1. Domain-agnostic prompt templates

Creating axes from a prompt User prompt provided as *concept*, the generation domain defined via *domain* string ('image', 'text', or 'UI').

```
create_design_space_prompt = f'''
You are tasked with creating a design space for a {domain} of a {concept}.
Generate a list of axes of the design space that is relevant to the concept and domain. For example, for an image of a car, the design space could be `car_type', `car_color', `background', `camera_angle', etc... There should be between 4-6 concrete axes. Each axis should be 1-4 words and not duplicate the others.
Return the list of axes in a <axes></axes> XML tag, like this:
<axes>
<axis>AXIS HERE</axis>
<axis>AXIS HERE</axis>
<axis>AXIS HERE</axis>
</axes>
'''
```

Generating default tags for each axis

```
fill_design_space_prompt = '''
Here is set of axes in the design space for a {domain} of a {concept}:
<axes>
{axes}
</axes>
Come up with the most likely value for each axis.
Return the design space in a <axes></axes> XML tag, like this:
<axes>
<axis name='AXIS NAME HERE'>AXIS VALUE HERE</axis>
<axis name='AXIS NAME HERE'>AXIS VALUE HERE</axis>
<axis name='AXIS NAME HERE'>AXIS VALUE HERE</axis>
```

```
</axes>
'''
```

Exploring variations along an axis

```
explore_axis_prompt = f'''
Here is an axis in the design space of a {domain} of a {concept}:
{axis}
Create {n} possible values for the axis. They should be meaningfully different and vary along only this axis. If the axis is continuous in any way, organize your options along that (like shortest to tallest, darkest to lightest, etc...)
Return the axis in a <options></options> XML tag, like this:
<options>
<option>OPTION HERE</option>
<option>OPTION HERE</option>
<option>OPTION HERE</option>
</options>
'''
```

F.2. Domain-specific mapping from tags to visual content

Image generation Given the current state of the design space, map to a detailed prompt given to a text-to-image diffusion model.

```
image_gen_expand_system_prompt = '''
You are a helpful assistant that expands prompts for image generation.
You will be given a concept and a list of examples.
You will need to expand the concept into a more detailed prompt that will be used to generate an image.
The expanded prompt should be more specific and detailed than the original concept.
'''
image_gen_expand_user_prompt = f'''
Expand the following concept:

<concept>
{concept}
</concept>

Here is a precise description of what needs to be constrained in your expanded prompt:
{design_space}
'''
```

UI Generation Given the current state of the design space, we aim to generate a functional HTML+CSS UI. We do this in two steps: 1) first map the state of the design space to a detailed prompt, then 2) map the detailed prompt to a HTML+CSS UI.

Design space to detailed prompt:

```
ui_gen_expand_system_prompt = '''
You are a helpful assistant that expands prompts that will be sent to a UI generation model.
You will be given a concept and a list of examples.
You will need to expand the concept into a more detailed prompt that will be used to generate UI designs.
The expanded prompt should be more specific and detailed than the original concept.
'''

ui_gen_expand_user_prompt = f'''
Expand the following concept:

<concept>
{concept}
</concept>

Here is a precise description of what needs to be constrained in your expanded prompt:
{design_space}
'''
```

System prompt for mapping UI prompt to full UI:

```
ui_gen_system_prompt = '''
You must generate a UI element based on a given prompt.

The UI must be wrapped in a <ui></ui> XML tag, like this:

<ui>
<!-- UI HERE -->
</ui>

```

```
The UI must be a valid HTML div element, and you should use Tailwind CSS classes to style it.

The outer level of the UI must be a div element, with the w-full and h-full classes, so it takes the full width and height of the container.

Make sure to follow every single instruction provided in the prompt carefully, and comment your code extensively.
'''
```

Text generation Similar to UI generation, we set up a two-stage process–given a user prompt and the current state of the design space, an LLM is first asked to generate a prompt, then actually runs the prompt. However, while UI generation requires a custom system prompt to guide the LLM, the text generation process does not require this.

```
text_gen_expand_system_prompt = '''
You are a helpful assistant that expands prompts for text generation.
You will be given a concept and a list of examples.
You will need to expand the concept into a more detailed prompt that will be used to generate text.
The expanded prompt should be more specific and detailed than the original concept.
'''

text_gen_expand_user_prompt = f'''
Expand the following concept:

<concept>
{concept}
</concept>

Here is a precise description of what needs to be constrained in your expanded prompt:
{design_space}
'''
```