Unboxing Diffusion Models for the Arts: Interactive Model Bending and Practice-Based Explainability

Ahmed M. Abuzuraiq and Philippe Pasquier

Abstract Explainable AI (XAI) in creative practice can be less about technocentric explanation and more about enabling artists to inspect, modify, and debug models as part of making. Yet large-scale text-to-image diffusion systems are typically presented as opaque end-to-end tools, limiting this kind of material engagement. We argue that even large models can function as creative materials when their internal structure is made visible and manipulable. To support this, we propose a hands-on approach to explainability centred on experimentation and intervention. We instantiate this approach with a model bending and an interactive (inspection) interface integrated into ComfyUI's node-based workflow, including interactive layer selection and intervention controls. Through qualitative and quantitative analysis of bending interventions in Stable Diffusion 1.5, we show how manipulating specific components of a diffusion pipeline produces relatively consistent families of visual effects, allowing artists to build practical, layer-level intuition about how different parts of the model shape generated images.

1 Introduction

Explainable AI (XAI) traditionally focuses on demystifying machine learning systems for auditing, transparency, or safety [1]. However, explainability in creative contexts can serve different roles, including making models modifiable and debuggable for a meaningful artistic engagement [2], and creating the conditions for a sustained artistic practice that goes beyond mere amusement [3]. The last workshop on XAIxArts [4] culminated in a manifesto that called artists and researchers to ex-

Ahmed M. Abuzuraiq
School of Interactive Arts and Technology, Surrey, Canada e-mail: aabuzura@sfu.ca

Philippe Pasquier
School of Interactive Arts and Technology, Surrey, Canada e-mail: pasquier@sfu.ca

Under review for "Explainable AI for the Arts" (N. Bryan-Kinns, Ed.), Springer.

plore alternatives to the technocentric explanations of AI that value artistic practices and use intentional hacking, glitches, and imperfections as creative tools. This work continues along those lines.

It is argued that working with curated (small) datasets and human-scale models can enhance artists' agency [5]. Broadening the scope of the artistic process with AI to include both model training and inference can also reinforce artists' trust in AI models [3]. Both approaches afford artists more control over AI models. However, artists' ability to craft with AI models as materials for creation [6, 7, 8] is diminished when working with large-scale generative models [9]. If small-data and model training present an alternate explainable way for artists working with human-scale generative models, what options are available for artists wanting to work with large-scale models such as text-to-image diffusion models? Although this material relationship is easier to build with small models, where data and model structure/training are accessible and malleable, it becomes more challenging at scale. However, we argue that large models can also be treated as material, provided their structure is exposed and can be manipulated.

We propose that treating AI models as materials, especially within interfaces that expose the model's components like the node-based interface of ComfyUI, can foster a craft-based relationship between artists and generative systems. In particular, through long-term engagement and hands-on manipulation of a model's components, artists can develop a form of tacit understanding and familiarity with their AI materials akin to that found in traditional crafts, i.e. a form of explainability that is rooted in doing. We explore the application of these ideas for large-scale models through a plugin and interface for model bending [10] that is implemented into the node-based interface of ComfyUI [11].

We argue that interactive model bending can expose the material structure of large generative models to artists, enabling creative exploration and practice-based explainability through direct intervention. To support this claim, we present a model-bending plugin integrated into ComfyUI, an interactive inspection interface for navigating and manipulating diffusion model components, and a systematic qualitative and quantitative exploration of bending outputs in Stable Diffusion 1.5. Together, these contributions point toward how large-scale diffusion models can be engaged with creative materials.

2 Background

2.1 Image Generation Models

Generative models for image generation aim to learn the underlying distribution of visual data so that new, realistic images can be synthesized. Among the various approaches, diffusion-based generative models have emerged as a powerful and flexible framework, particularly for high-fidelity and text-conditioned image synthesis.

2.1.1 Latent Diffusion

Diffusion models gradually add noise to real data using a fixed forward process, then train a neural network to reverse that process step by step so it can turn random noise into coherent samples at generation time. Latent diffusion does the same thing in a compact latent space, e.g., learned by a variational autoencoder (VAE) model, where the encoder maps images into low-dimensional latents for diffusion, and the decoder maps the denoised latents back to pixels, making generation more efficient. The outputs of the text encoder of a contrastive model (e.g., CLIP or T5) can be fed into the diffusion model via cross-attention, where they are used to guide the generation to adhere to a user prompt. In addition to text conditioning, each denoising step is associated with a discrete time index that indicates the current noise level. These timesteps are embedded , via the *time_embed* layers in SD1.5, into continuous representations and provided to the model alongside the noisy latents. This allows the network to condition its denoising behaviour on the current noise level, ensuring a coherent progression from high-noise initial states to low-noise, structured outputs.

2.1.2 Transformers and Attention

Transformers are a neural network architecture built around attention mechanisms, originally introduced for sequence modelling and now widely used across text, image, audio, and multimodal tasks.

Attention mechanisms form the backbone of diffusion models by allowing information to flow between different spatial regions based on content relevance. When mathematically described, attention is defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V, \tag{1}$$

where Q, K, and V denote the query, key, and value matrices, respectively, and d is the dimensionality of the key vectors. When a model's output is conditioned on an external signal, such as a text prompt, a variant known as cross-attention is employed, in which the key and value matrices are derived from text embeddings while the query matrix originates from image features. This formulation aligns linguistic information with visual representations, enabling text-conditioned image generation.

Earlier latent diffusion models, e.g., Stable Diffusion v1.5, for image generation utilized a UNet model for noise prediction at each step in the denoising process. A UNet model is composed of gradually downsampled input blocks that feed to middle blocks, which then expand again to recreate the input images through output blocks. UNet models also feature skip connections that relay information from input blocks to output blocks to preserve spatial details lost during downsampling and overall for better model training. More recent models (Stable Diffusion v3 onwards) have shifted to using a Diffusion Transformer (DiTs) architecture for noise prediction.

2.2 Network Bending

Network bending refers to the deliberate manipulation of an AI model's internal activations or computational pathways to intervene in its generative process, typically for expressive or exploratory purposes [10]. This is achieved by injecting a *bending operator* into a model to transform its intermediate outputs during inference, without the need for additional training or data. Bending operators can perform operations such as adding noise, multiplying/adding scalar values, rotation, and scaling; morphological operations like erosion and dilation, or other custom operations. Network bending is inspired by circuit bending, a practice originating in experimental electronic music, in which artists intentionally short-circuit or rewire electronic devices (often children's toys or synthesizers) to produce unpredictable or expressive behaviours [12].

Previous work on network bending has applied interventions at various layers of the synthesizer network in StyleGAN models [10, 13, 14]. More recently, Pavlov's exploration focuses on exchanging the parameter-free activation functions of StyleGAN2 and BigGAN models [15]. In the context of diffusion models, bending operations have been implemented at user-specified timesteps within the denoising process [16] (as inferred from source code), and at select layers of the noise prediction model (UNet) [17]. Though not labelled as model bending, the work by Grabe et al. on Hidden Layer Interaction similarly explores the manipulation of attention layers in a Stable Diffusion 1.4 model [18]. In this work, we introduce a set of tools for model bending within ComfyUI that enable artists to bend various components of the latent diffusion pipeline, including and extending beyond the components covered by previous work [10, 16, 17]. Our approach does not utilize clustering of features to capture semantic directions in the model, instead we find that large-scale text-to-image diffusion models appear to exhibit visually rich latent spaces that are sufficient for productive bending [1].

In the music domain, Collins recently explored the impact of bending on the musical features of music generated by the Stable Audio 1.0 model [20]. Earlier work by Yee-King and McCallum explored network bending for the Differentiable Digital Signal Processing (DDSP) model [21] for sound synthesis. Aside from exploring bending within creative domains, multiple generic frameworks exist for tracing and then intervening on the internal states of PyTorch models, including NNSight [22], Pyvene [23] and baukit [24]. Model bending can be easily implemented using any of these frameworks.

Shared among the above approaches is the manipulation of model weights without requiring additional training, e.g., in contrast to the work by Aldegheri et al. [25] insert a small trainable module to push the model towards creative outputs. However, we focus on training-free approaches as we find them more accessible to non-technical artists [2].

---

[1] Though certain parts (*h-space*) seem to be more potent for semantic manipulation[19]

[2] The Wekinator for predictive models and Low Rank Adaptation (LoRAs) for generative models present successful examples of accessible and effective training-based approaches

2.3 Bending Interfaces

Model bending exposes all of the model's parameters, which can range from millions (in StyleGAN) to billions (in Stable Diffusion 1.4), that are distributed across numerous parts of the model, such as fully connected, convolutional, normalization or attention layers, to name a few. Furthermore, multiple types of operations can be applied to the bent parts. This presents a challenge in effectively applying model bending in practice. As a response, we find multiple systems for model bending (or related) that introduce interfaces to assist in specifying the parts of the model to bend. These systems include Autolume [26], which allows users to visualize and transform intermediate layers in StyleGAN2 models, Pavlov's interface [15], where users pick neural layers and specify how they wish to change their activation functions, and Grabe's interface for hidden layer interaction [27], which enables users to target specific ranges within a convolutional channel of a Deep Convolutional GAN model. Lastly, the PatchExplorer interface [18] allows users to visualize and manipulate attention heatmaps in diffusion models. All of these interfaces present the final model inference outputs [27], but some also visualize the model layers' structure [15], while others visualize select intermediate feature maps [26], or all the attention heatmaps for a given prompt [18] to facilitate semantic and targeted edits. Our interface focuses on presenting the structure and bending outputs of the model instead of its intermediate representations. By being implemented in ComfyUI, our system can be extended to cover a wide range of generative models or custom bending operators.

3 Diffusion Bending: System and Interface

To explore the potential of model bending with diffusion models, we implement custom nodes that enable intervening on diffusion model parts in ComfyUI, and build an interactive interface that enables model bending within the context of ComfyUI. In what follows, we briefly describe the key elements of our approach.

3.1 Stable Diffusion Models

In this work, we explore bending the Stable Diffusion 1.5 (SD1.5) model [28], an early large-scale text-to-image system. While newer and more powerful models such as Flux or Z-Image have since been developed, the overall components of these systems remain largely the same, including text conditioning, encoding and decoding in a compressed latent space, multi-step denoising, and the use of a neural network—whether a UNet or a Diffusion Transformer—for noise prediction.

Stable Diffusion 1.5 employs a UNet architecture for denoising, composed of multiple input blocks, a middle block, and multiple output blocks. These blocks

contain residual blocks, which consist of convolutional layers with skip connections that pass feature maps from the encoder to the decoder at corresponding resolutions. These skip connections help preserve spatial detail and stabilize training by enabling the direct reuse of low-level features during reconstruction. The Spatial Transformers incorporate both self-attention and cross-attention layers, allowing image features to interact with each other and with text embeddings used for conditioning. In addition, the UNet includes downsampling and upsampling operations that progressively reduce and then restore spatial resolution, enabling the model to capture both fine-grained details and high-level semantic structure during the denoising process.

Finally, we note that components beyond the UNet in the Stable Diffusion pipeline can also be manipulated. These include the Variational Autoencoder (VAE) used for image encoding and decoding, the text embeddings produced by the Contrastive Language–Image Pre-training (CLIP) model, intermediate latent representations, LoRA matrices appended to the UNet, and even activations at selected timesteps during sampling. Bending operations can be applied at these points to introduce controlled variations while keeping other generation parameters fixed. Although we provide custom nodes to support these manipulations, they are beyond the scope of this paper and are therefore not discussed further.

3.2 ComfyUI

ComfyUI is an open-source, node-based interface for designing and executing advanced diffusion pipelines [11]. Unlike other interfaces such as AUTOMATIC1111 [29], which prioritizes simplicity and abstracts away the model's internal workings, ComfyUI adopts a low-code, modular approach that decomposes the latent diffusion process into discrete nodes. These nodes represent stages such as loading base models and applying LoRA adaptations, encoding images into latent representations, generating textual embeddings (e.g., via a CLIP model), and configuring the denoising process in diffusion models. ComfyUI's backend also manages model loading in a way that helps circumvent GPU memory constraints.

We chose ComfyUI because it exposes the inner components of the diffusion pipeline, encouraging users to explore, customize, and develop an understanding of each component. Additionally, ComfyUI often supports the most up-to-date models for image and video generation, making it a reasonable choice for research-driven or artistic experimentation. A basic workflow in ComfyUI is illustrated in Figure 1, and includes steps such as model loading, prompt encoding with CLIP, sampling with user-defined parameters (e.g., number of steps, CFG scale), and decoding the resulting latent image.

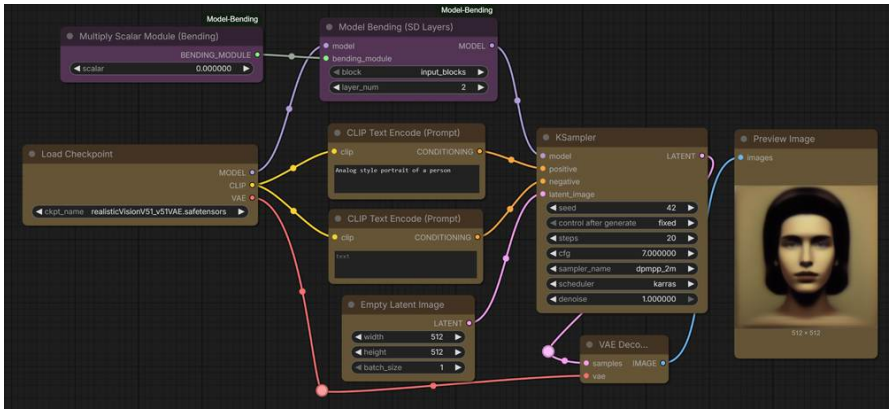


Fig. 1: Basic workflow in ComfyUI (gold). The UNet bending nodes are shown in purple.

3.3 Diffusion Bending Custom Nodes

In this work, we implement model bending through custom nodes in ComfyUI. The overarching goals of this work are: (1) to provide tools for introducing visual variations and diversity into the text-to-image diffusion generative process, (2) facilitate a better understanding of the generative process and its parts, (3) lower the barrier of entry to model manipulations, and (4) integrate into an existing ecosystem of tools and access the latest generative models. We address goals by developing a diverse set of custom nodes that enable interventions at different stages of the latent diffusion process, alongside an interactive interface for visualizing and manipulating model components. By integrating the system directly into ComfyUI, the tool benefits from ComfyUI's growing community, support for the most recent models, and access to a wide range of existing plugins. The tool can be downloaded manually through a git repository [3] or by installing it directly through ComfyUI's Nodes Manager.

In general, bending works by manipulating the outputs of parts of the generative process before passing them downstream. Users specify the model to be bent, the parts of the model they wish to manipulate through paths that describe the hierarchy leading to these parts (e.g. *diffusion_model.middle_block.0.in_layers*), as well as choosing the bending operation to be applied to the tensor output at that part. The bending operations are PyTorch modules, and can be chosen out of a predefined collection (e.g. multiplication, rotation, or noise addition, among others) or by creating custom operations as new PyTorch modules. Lastly, users may optionally provide the range of denoising steps within the diffusion process at which to bend; by default, bending is applied to all steps.

With these inputs in place, the bending node copies the model definition (so bending changes only impact downstream parts of the workflow) and assigns hooks (event

[3] `https://github.com/abuzreq/ComfyUI-Model-Bending`

listeners) with a function that calls the bending module when any of the bending paths are reached during inference. Alternatively, but functionally equivalent, users can use NNSight as a backend for implementing their specified model interventions (i.e. bending operations).

We provide multiple ways for specifying the bending paths and operations. The simplest *Model Bending (SD Layers)* node as in Figure 1) allows users to choose from a pre-computed list of convolutional layers in the UNet. Alternatively, users can use the *Model Bending* or *Model Bending (NNSight)*, which enable user to input a list of comma-separated paths at which to bend, and a bending operation to apply. Lastly, users can specify the bending parameters in a JSON format, including multiple operations at multiple paths, and the range of timesteps to bend at (Figure 2).

```
{
  "bends": [
    {
      "path": "input_blocks.5.0.in_layers.0",
      "module_type": "rotate",
      "module_args": {
        "angle_degrees": 90
      }
    }
  ],
  "steps_min": null,
  "steps_max": null,
  "max_denoising_steps": 200
}
```

Fig. 2: JSON description of bending operations.

3.4 Interactive Bending

Considering the complexity and scale of modern diffusion models, as well as the wide range of bending parameters, we found it important to support users in interactively identifying the parts they wish bend. We describe these supports next.

3.4.1 Model Inspection

For more granular control, including the ability to select specific blocks or layers, we provide the Model Inspector node (Figure 3). This tool displays the model's architecture as a nested, expandable tree, allowing users to visually navigate and select any bending path to pass to the model bending node. It currently supports both UNet and VAE models.

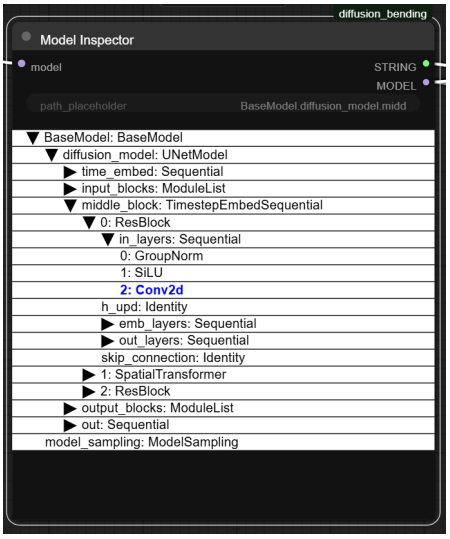


Fig. 3: With the model inspector node, artists can interactively pick layers from the model hierarchy to manipulate. Picked layers are highlighted.

3.4.2 Interactive Bending

The Interactive Bending Web Interface is a web-based tool that allows users to explore and apply interventions (bending operations) to the UNet of a latent diffusion model. The interface presents a model structure diagram as an SVG visualization of the diffusion pipeline, including components such as the UNet, VAE, CLIP, and scheduler, with a brief explanation revealed when each part is clicked (Figure 4 - A). A dedicated UNet layer explorer displays the network in an interactive U-shaped layout, where major sections like input blocks, the middle block, and output blocks can be expanded (Figure 4 - B). Within each block, layers are organized by module type -such as ResBlock, SpatialTransformer, Upsample, and Downsample- using collapsible panels for easier navigation. An early version of the interface, with pre-computed results, can be found at `https://diffusion-bending-demo.netlify.app`. To access and run the interactive interface as well as the scripts used to generate the results in the in Section 4, please download the project from `https://drive.google.com/file/d/1DJx3WVfdQLsd73VEoe-T95OAzbMpw6OD/view?usp=sharing`.[4]

For each bendable layer, the interface provides clear selection and bending controls that display the layer name alongside a slider (Figure 5). The slider allows users to choose the type and amount of intervention, including rotation-based bending (0°, 90°, 180°, or 270°), additive Gaussian noise (with standard deviations of 0, 1, or 2), and multiplicative scaling (0 or ablation, 0.5, 1, 1.5, or 2). Layers are colour-coded by type according to a legend, making it easy to distinguish different kinds of mod-

[4] This is prepared for the review process and will be made public later.

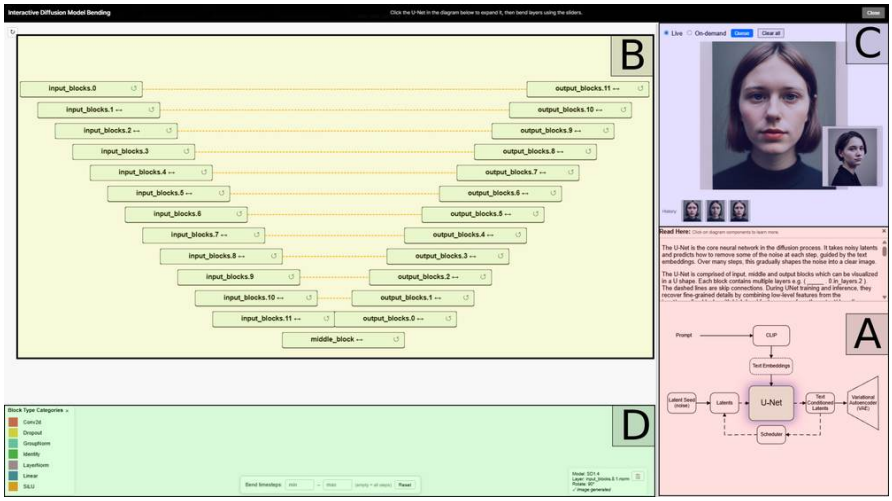


Fig. 4: Interactive Interface for Bending Diffusion Models. The interface shows the structure of the UNet (B), presents the current and past bending outputs (C), and shows the high-level structure of the entire latent diffusion process (A).

ules at a glance. At the bottom of the UNet diagram (Figure 4 - D) are the legend mapping colours to module types, controls for the range of denoising steps to impact, and a brief summary of currently applied operations with the option to copy these operations as a JSON string (c.f., Figure 2)

A preview panel displays both the unbent (default) output and the bent output based on the current configuration, while an output history strip shows thumbnails of past generations; selecting a thumbnail restores the bending settings that produced that result. Queue controls allow users to choose between a live mode, where prompts are automatically queued when sliders change, and an on-demand mode, where generation is triggered manually with a button (Figure 4 - C).

Users can open the interface directly from a ComfyUI node (Interactive Bending WebUI) through a button or context menu. The node then reads the model structure, locates the UNet and visualizes it, and then applies the user-chosen bending parameters before passing the modified model downstream. Bending parameters are transmitted from the Web UI to the backend over HTTP, and the extension exposes REST endpoints for retrieving model structure, managing bending selections, clearing settings, and accessing image history.

4 Preliminary Study

We present a systematic study of the outputs of bending different parts of a latent diffusion model and analyze it qualitatively and quantitatively.

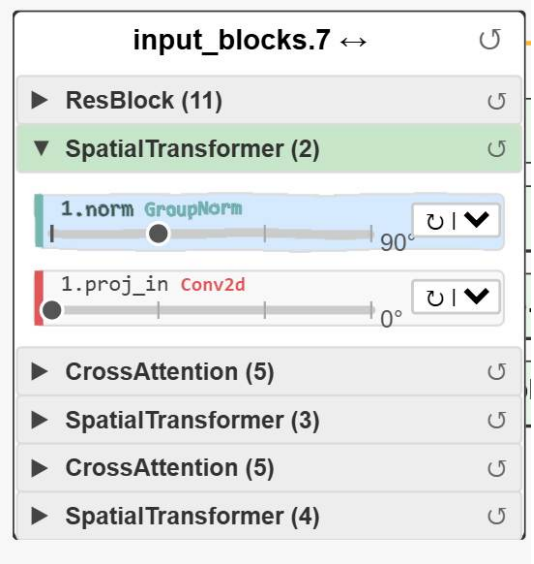


Fig. 5: In each UNet block, users can drill down into one of the containers (higher-level components), pick a low-level layer (e.g. Normalization layer), specify a bending operation from a pulldown, and drag the slider to apply it. Multiple layers can be bent at once, and current bending operation can be reset on a block or container level.

4.1 Study Setup

We conducted a systematic study of bending interventions by enumerating combinations of bend location within the UNet, seeds, prompts, and denoising step ranges. As a bending operator, we use multiplication by a scalar, with values of 0.0 (i.e. ablation), 0.5, 1.5, and 2.0.

For each configuration, one aspect was fixed while varying the others resulting in: (1) Same Prompt/Seed, Different Layers (Section 4.2.1), (2) Same Prompt, Different Seeds (Section 4.2.2), (3) Same Seed, Different Prompts (Section 4.2.3), (4) Same Seed/Prompt/Layer, Different Timesteps (Section 4.2.4). In all, the unbent (default) run executed first to serve as a baseline. All bent outputs were recorded along with their associated parameters, generated image, and their final latent representation (i.e., ones passed to the decoder). This setup enables the interactive WebUI bending interface to be driven from cached experimental results for quick feedback and supports subsequent quantitative analysis of the outcomes.

To quantify the effects of bending, we measured deviation from the unbent baseline using the cosine distance between the latents of the results (i.e. processed latents produced through the denoising process). Metrics were computed post hoc for all experimental conditions and associated with each generated sample.

Results were analyzed by grouping bent outputs according to structural and procedural factors such as UNet region, bend type, and denoising step range, and by summarizing metric distributions within each group using standard descriptive statistics. This framework was designed to support large-scale, automated sweeps of bending interventions, enable quantitative comparison across diverse manipulation regimes, and facilitate reproducible analysis of how localized interventions propagate through the diffusion process.

4.2 Qualitative Results

In the following sections, we present results from bending experiments across multiple configurations. All experiments use a variant of Stable Diffusion v1.5 - specifically, RealisticVisionV51_v51VAE—which has been fine-tuned for realistic image generation, with a particular emphasis on faces [5]. We deliberately chose a face-capable model because humans are especially sensitive to subtle variations in facial features. This makes faces a useful domain for illustrating how bending can introduce visual differences and experimental aesthetics that may be difficult to achieve through prompting alone. Furthermore, all denoising experiments were conducted using seed 42 (unless otherwise specified), with 20 denoising steps, a Classifier-Free Guidance (CFG) scale of 7, the DPMPP_2m sampler, and the Karras scheduler.

The prompt ”Analog style portrait of a person” was deliberately constructed to separate style, composition, and subject. The phrase ”analog style” specifies a broad stylistic regime, ”portrait” constrains the image format and composition, and ”of a person” defines the semantic subject. This decomposition allows us to examine how bending affects each facet (or does not). An analog style was selected specifically for its aesthetic flexibility and tolerance for variation.

4.2.1 Same Prompt/Seed, Different Layers

Using the same prompt and seed, we bend different layers in the model using multiplication values of 0 (ablation), 0.5, 1.5 and 2.0. Figures 6 and 7 follow.

[5] https://huggingface.co/lllyasviel/fav_models/blob/main/fav/realisticVisionV51_v51VAE.safetensors

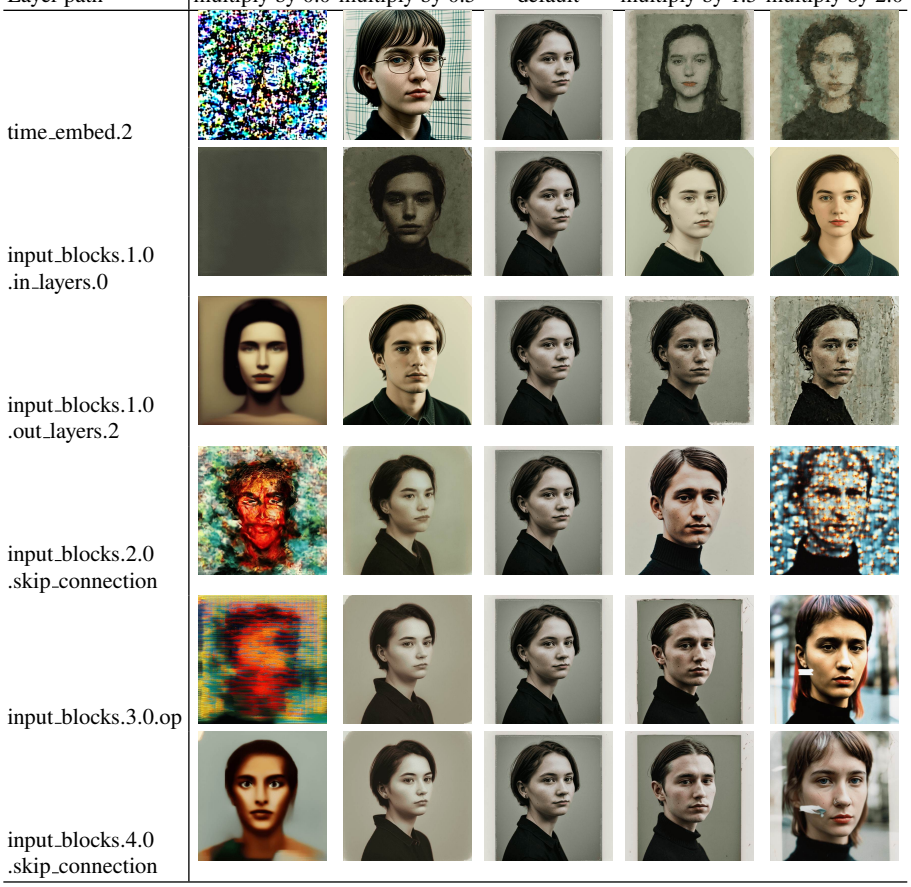


Fig. 6: One prompt; rows: bending layer path, columns: module_args. Prompt: Analog style portrait of a person

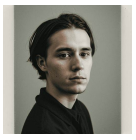
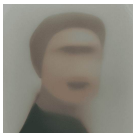
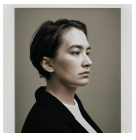
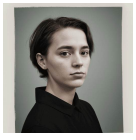
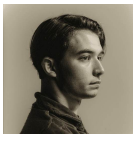
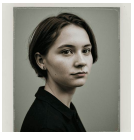
Layer path	multiply by 0.0	multiply by 0.5	default	multiply by 1.5	multiply by 2.0
input_blocks.5.1 .proj_in					
input_blocks.7.1 .proj_in					
input_blocks.8.1 .transformer _blocks.0.norm1					
middle_block.1.transformer _blocks.0.attn2.to_out.0					
output_blocks.3.1.norm					
output_blocks.4.1 .transformer _blocks.0.norm1					
output_blocks.5.1 .transformer _blocks.0.norm1					
output_blocks.7.1 .transformer _blocks.0.norm1					

Fig. 7: One prompt; rows: bending layer path, columns: module_args. Prompt: ”Analog style portrait of a person”

4.2.2 Same Prompt, Different Seeds

Using the same prompt, ”Analog style portrait of a person”, we examine whether the same bending effects are consistent across different seeds. We repeat the experiment for select layers, hand-picked to span input, middle and output blocks in the UNet. The first row (seed 42) is the one used across the rest of the paper. Figures 8, 9, 10, 11, 12 follow.

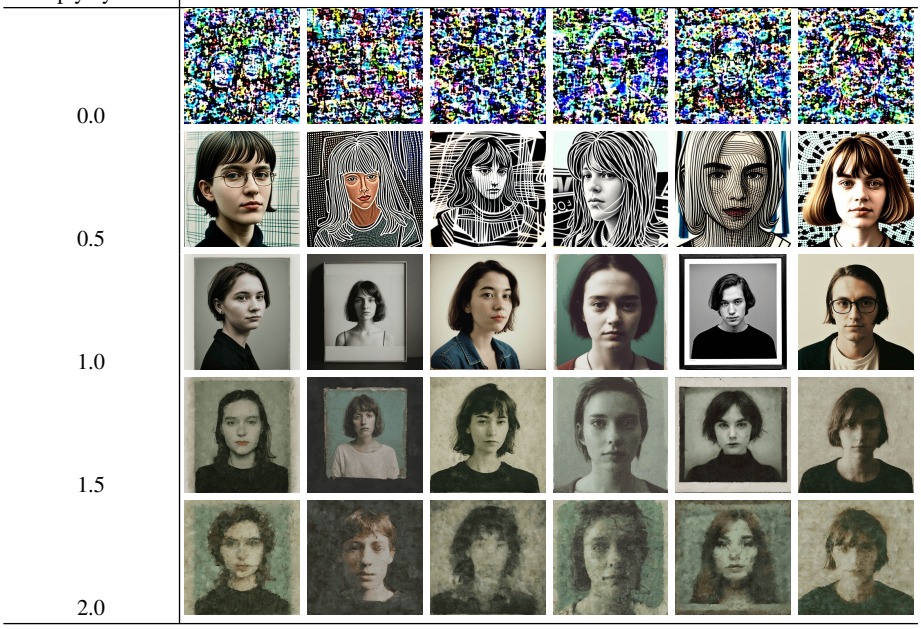


Fig. 8: Bending *time_embed.2*; Same prompt, different seeds.

Fig. 9: Bending *input_blocks.1.0.out_layers.2*; Same prompt, different seeds.

Fig. 10: Bending *input_blocks.4.0.skip_connection*; Same prompt, different seeds.

Fig. 11: Bending *middle_block.1.transformer_blocks.0.attn2.to_out*; Different seeds.

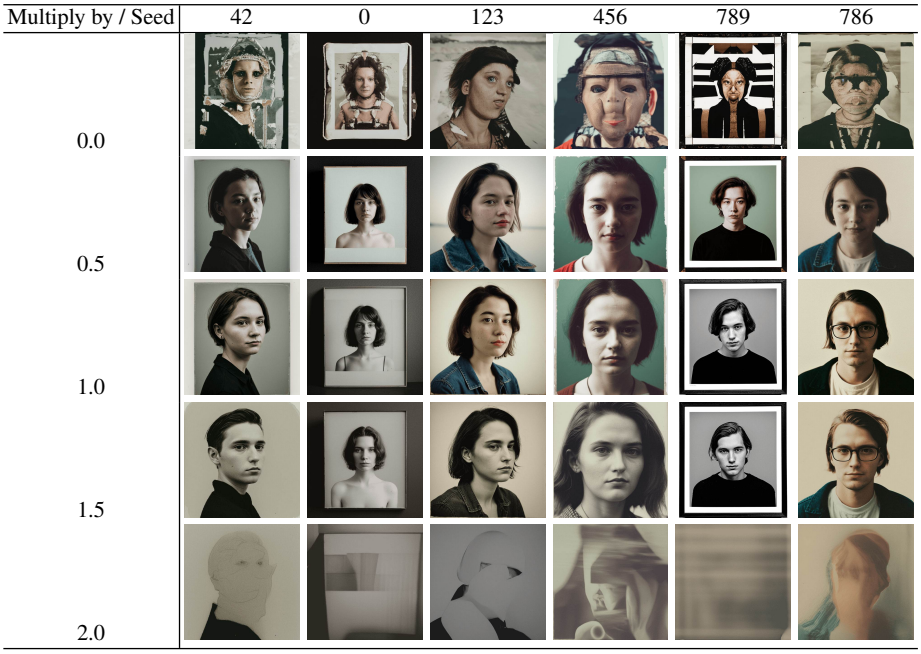


Fig. 12: Bending *output_blocks.3.1.norm*; Same prompt, different seeds.

4.2.3 Same Seed, Different Prompts

Using the same prompt, we examine whether the same bending effects are consistent across different prompts where we change the style, composition and subject. In other words, we examine whether the bending effects relate to what a specific layer learns and its functional role during inference, or due to interactions with the prompt guidance, or both. The prompts are: (1) ”Analog style portrait of a person”; (2) ”A red apple on a white background”; (3) ”Mountain landscape at sunset”; (4) ”Abstract geometric shapes, vibrant colors”; (5) ”unicorn”; (6) ”a floating orb”. Figures 13, 14, 15, 16, 17 follow.

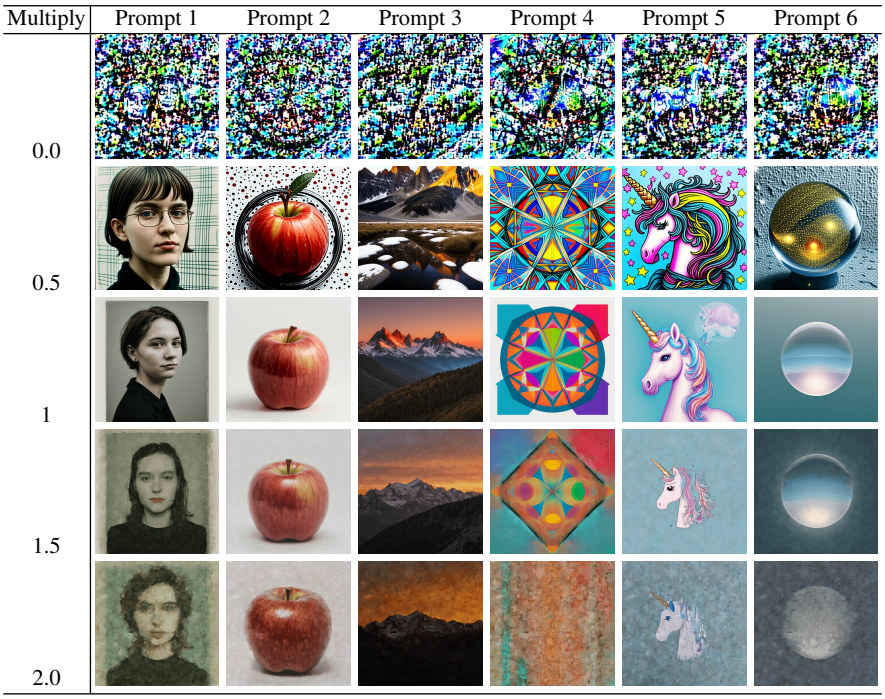


Fig. 13: Results of bending the layer *time_embed.2*; Different prompts, same seed.

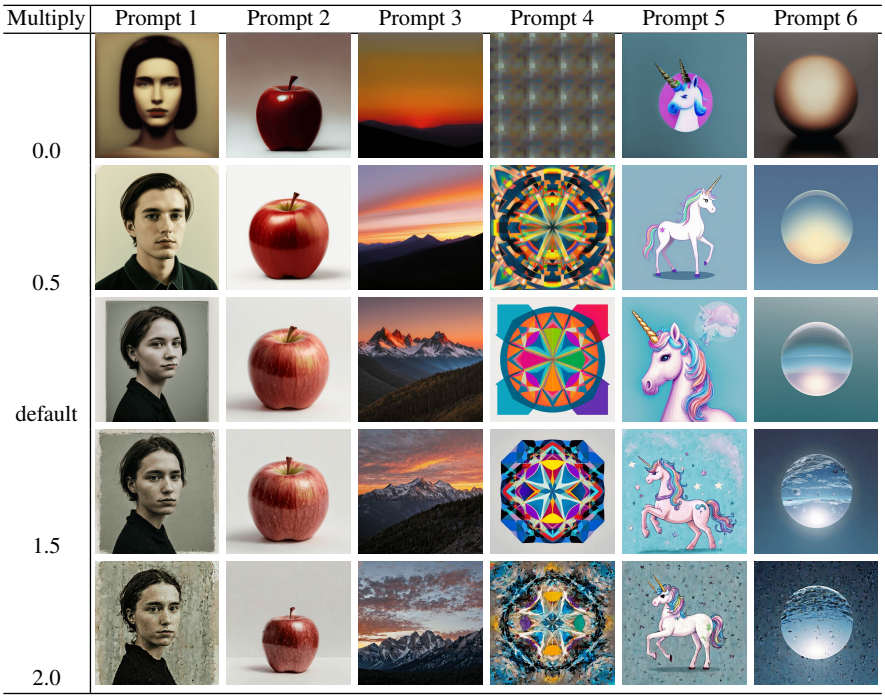


Fig. 14: Results of bending the layer *input_blocks.1.0.out_layers.2*; Different prompts, same seed.

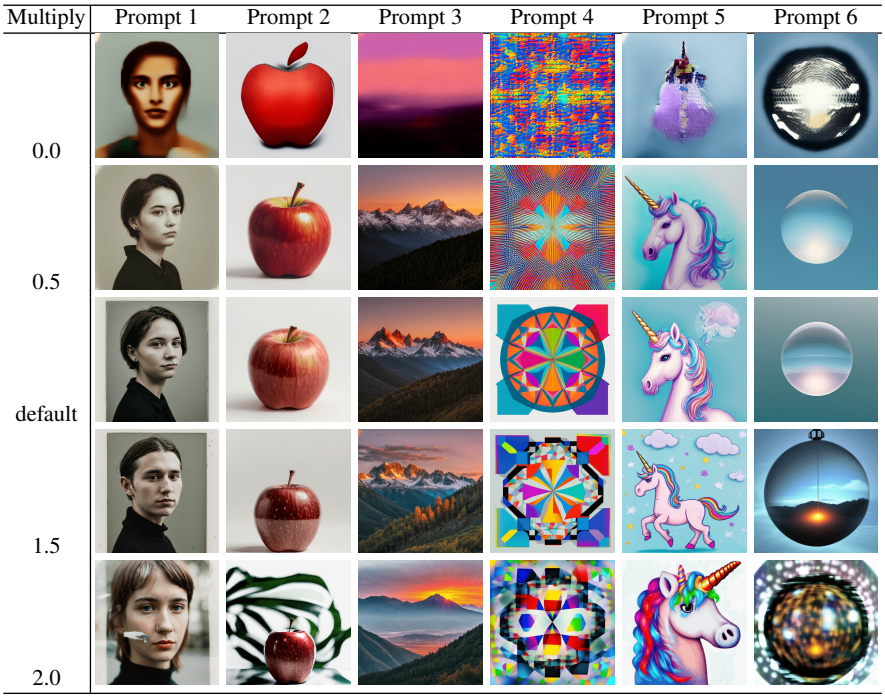


Fig. 15: Results of bending the layer *input_blocks.4.0.skip_connection*; Different prompts, same seed.

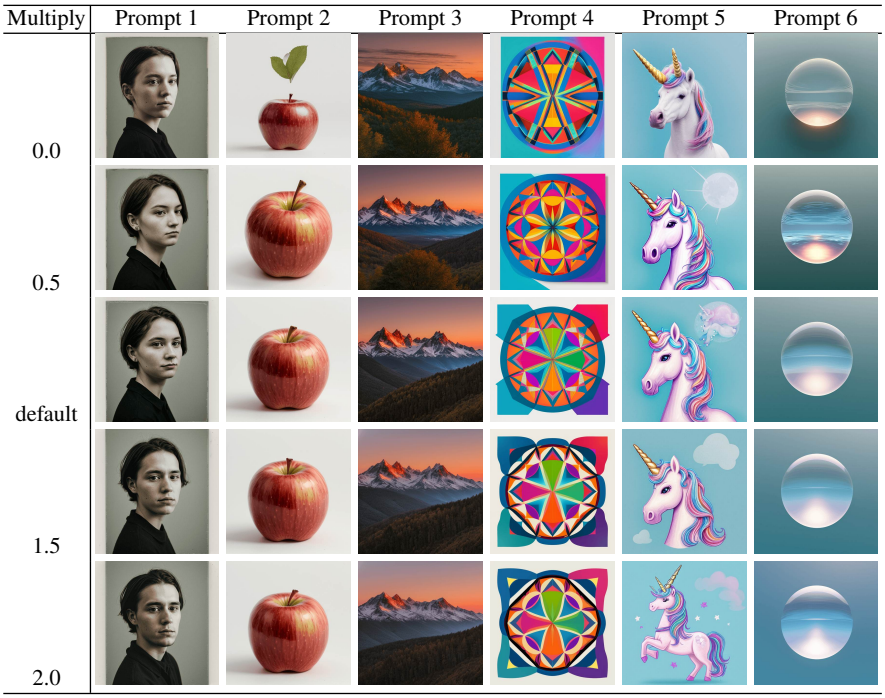


Fig. 16: Bending *middle_block.1.transformer_blocks.0.attn2.to_out.0*; Different prompts, same seed.

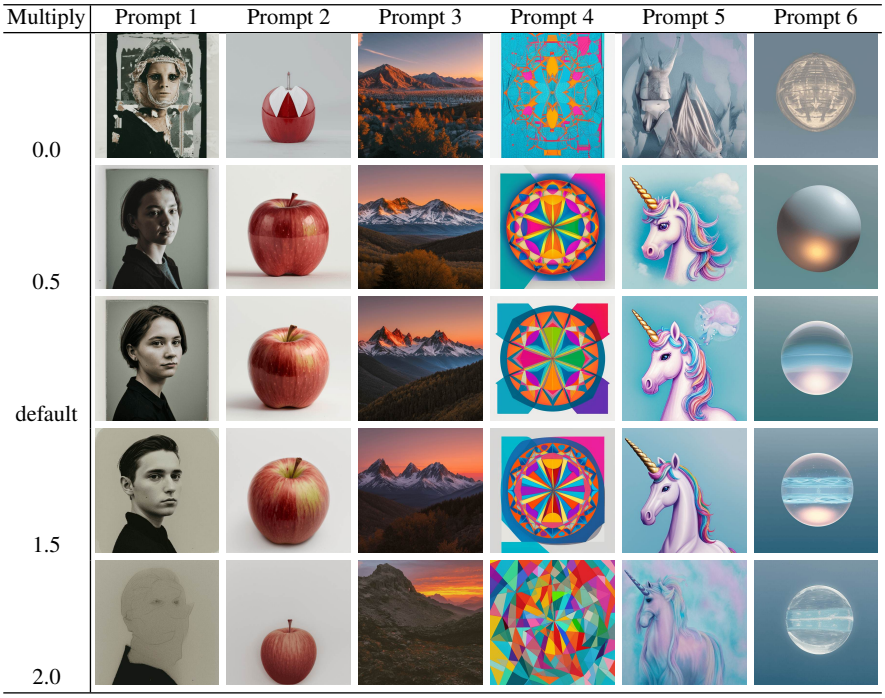


Fig. 17: Bending the layer *output_blocks.3.1.norm*; Different prompts, same seed.

4.2.4 Same Seed/Prompt/Layer, Different Timesteps

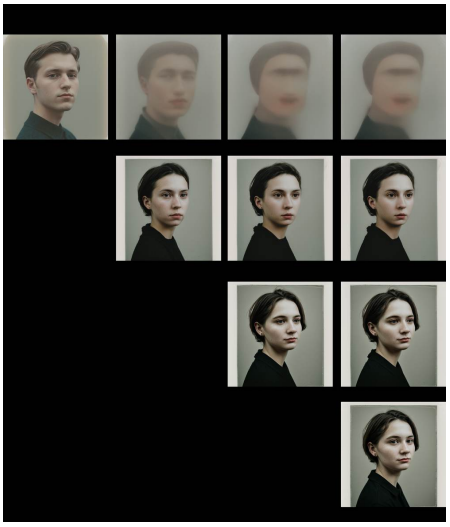


Fig. 18: Varying the denoising time steps (out of 20) during which bending is applied.

4.3 Quantitative Results

By analyzing the generated images and their associated latents, we can identify which parts of the model produce the most impact when bent. In particular, we can compute the cosine distance between the latents of the bent results and the latent of the default unbent result. Next, we can place bent results into groups and compare which of these groups, on average, produces a higher mean distance. The grouping can be based on whether bending was part of the input or output block, a Residual Block, a Spatial Transformer, a Convolutional layer or a Normalization layer, etc.

Our empirical analysis reveals significant disparities in impact across different architectural components. Our findings indicate that the input region of the UNet architecture exerts a stronger influence on feature divergence than the output region, with a mean cosine distance of 0.1776 ± 0.1674 versus 0.1088 ± 0.1234. When comparing structural components, Residual Blocks (ResBlocks) demonstrated a greater impact (0.1333 ± 0.1541) compared to Cross-Attention mechanisms (0.1070 ± 0.1110), suggesting that the convolutional backbone is more sensitive to bending than the attention-based pathways.

At the layer level, convolutional layers (Conv2d) yielded a higher mean impact (0.1685 ± 0.1658) than linear layers (0.1185 ± 0.1268). Notably, normalization layers were among the most impactful atomic components, with LayerNorm (0.1688 ± 0.1562) and GroupNorm (0.1642 ± 0.1627) showing results comparable

to convolutional layers. The most significant deviations were observed in high-level structural containers; specifically, the TimestepEmbedSequential (0.4518 ± 0.2453) and Sequential (0.4117 ± 0.2454) containers produced the highest mean distances. These areas, which include critical entry (e.g. timestep embedding such as *time_embed.2*) and exit points like *input_blocks.0* and *out.0*, represent the most sensitive points in the model. Overall, these results suggest a clear hierarchy where Input > Output and ResBlocks > Cross-Attention in terms of their contribution to the total cosine distance.

4.4 Findings

Our qualitative results demonstrate relatively consistent visual impacts across different seeds and prompts for the same layer. In other words, the layer type/location/container and the bending magnitude ($0 < -- > 2$) were more influential on the results, whereas images generated with the same bending magnitude and layer—but different seeds or prompts—tended to share greater resemblance than images where the magnitude or layer was changed. Resemblance within the same seed or prompt, keeping everything else fixed, was higher than when the seed or prompt was changed, which is expected given the role of text conditioning in guiding generation.

The type of layer appears to have a stronger impact than its location. This contrasts with StyleGAN models, where earlier layers consistently produced more drastic changes, while later layers primarily affected fine-grained features [10]. This difference is likely due to two factors. First, the UNet used in our study is composed of heterogeneous layers that serve different functions during inference. Second, diffusion models operate through multiple denoising iterations, where the input is gradually refined at each step.

In most of our study (Sections 4.2.1, 4.2.2, and 4.2.3), we applied bending across all denoising steps, which ensured consistent results and, in effect, reinforced the bending impacts. In Section 4.2.4, we observe that covering the initial steps (i.e., from the 0th timestep onward) produces most of the blurring visual effects seen when ablating *output_blocks.4.1.transformer_blocks.0.norm1*. Skipping the first 5 or 10 timesteps, by contrast, primarily leads to fine-grained or decorative changes—similar to the role played by later layers in StyleGAN models.

The quantitative analysis confirms that the layer type has the strongest impact measured by the latent distance. It also confirms a big difference between input and output blocks, even if our qualitative assessment was not conclusive in that regard.

5 Discussion

5.1 Model Bending Cheat Sheets

Through conducting both qualitative and quantitative analyses, our intent is not to establish findings that generalize across different UNet architectures. Rather, we present this work as a demonstration of the kind of systematic experimentation that may support artistic practice. The observations derived from the specific UNet of SD1.5 can be translated into a model-specific "cheat sheet" that other artists using the same model may consult when directing their bending efforts, depending on the type and magnitude of effect they wish to produce.

Analyses of this kind can be carried out either through interactive interfaces (subsection 3.4) or through script-driven systematic sampling (section 4). Interactive approaches enable direct and exploratory manipulation, but diffusion models generally incur slower inference due to iterative denoising, despite emerging alternatives for near real-time generation [30]. Systematic exploration allows more exhaustive and controlled variation, yet requires computational resources and technical expertise to implement, visualize, and interpret results.

A balance between these modes may be achieved in at least two ways. First, analyses can be community-driven: if artists systematically document and share results for a given model, others can reuse and build upon this knowledge. Our preliminary findings suggest that bending effects exhibit a degree of consistency within the same model, supporting the feasibility of shared reference mappings. Second, systematic scripts can be prepared in advance and their results integrated into interactive interfaces. For example, visual indicators, such as colour-coded badges placed beside the layers/containers in Figure 5, could denote the relative impact of specific containers or layers (e.g., low, medium, or high).

Impact may be quantified through multiple complementary metrics, including perceptual difference (LPIPS) [31], visual feature distance (e.g., DINOv2 embeddings) [32], or semantic distance (e.g., CLIP image embeddings) [33]. Presenting such analyses within the interface—whether generated locally or contributed by others—would allow artists to situate their experimentation within a broader, collectively developed understanding of the model's behaviour.

5.2 Direction Manipulation vs. Systematic Exploration

Unlike direct manipulation, systematic exploration is relatively novel to the arts [34] compared to other design and engineering disciplines [35]. For example, the study by Davis et al. [36] recognizes the utility of a *design space exploration* framework when working with generative AI for ideation, and first-person accounts of AI-based video generation in ComfyUI [37] paint a picture of current generative AI workflows as

characterized by deliberation and systematic exploration rather than solely intuitive iteration.

5.3 From Models as Commodity to Models as Material

Model repositories like CivitAI [38] and TensorArt [39] provide users with thousands of pre-trained and fine-tuned models, generated images with associated prompts and parameters, and reproducible workflows for a wide variety of use-cases and styles. The quantity and diversity of models and workflows shared online are simply unprecedented within the context of generative art, and cloud computing services are making them more accessible. A large portion of these models are adaptations and personalizations (e.g. Low-Rank Adaptations – LoRAs [40]) of a smaller set of base models (e.g. Stable Diffusion [41] and Flux variants). Following Abonamah et al. [42], we argue that the proliferation of generative AI models is likely to lead to their commodification. Such a development may further entrench existing biases [43], as machine learning models are not neutral technologies but systems shaped by the data on which they are trained. When treated as interchangeable commodities or opaque services, these embedded biases risk being obscured rather than critically examined.

When technology becomes commodified, its design priorities shift—from serving as a raw material for creation (as with early computer terminals) to emphasizing accessibility and replaceability. This shift can bring innovation benefits, as seen with personal computers. However, in the case of generative AI, commodification is occurring at an unprecedented pace. The challenge, however, is that this rapid commodification may discourage sustained engagement with individual models. Instead of fostering deep familiarity with their inner workings, affordances, and limitations, users are incentivized to sample from an ever-expanding catalogue without developing the tacit knowledge necessary for critical or innovative use. This dynamic may also contribute to *AI fatigue*—a feeling of exhaustion or overwhelm, potentially driven by the rapid pace of updates to AI models and tools, as well as the opaqueness and complexity of AI systems [6]. Historically, artists have cultivated a deep, material understanding of their tools—whether brushes, cameras, or software—as a prerequisite for meaningful creative expression. Generative models require the same kind of sustained engagement to be used critically and responsibly. At the same time, many artists—who are uniquely positioned to critique and subvert these systems [4, 7]—are choosing not to engage with them on principle, citing concerns that these models are trained on the work of unattributed artists [44]. Therefore, we argue that it's imperative to encourage and offer tools for artists to engage with AI in material [27], sustained [3] and critical [7] ways, and the model understanding that is accrued through model bending could facilitate these forms of engagement.

[6] https://newsletter.victordibia.com/p/you-have-ai-fatigue-thats-why-you

5.4 Manipulation for Behavioural and Theoretical Understanding

The analyses presented here primarily contribute to what may be described as a *behavioural* understanding of diffusion models: a working mental model of how specific interventions—at particular layers, timesteps, or magnitudes—affect outputs. Rather than offering a comprehensive mechanistic account of internal representations, our approach demonstrates how systematic experimentation can generate actionable knowledge about input–output relationships. In this sense, explainability becomes valuable insofar as it enables practitioners to develop reliable heuristics for achieving desired effects [45]. The relative consistency observed across seeds and prompts within the same bending configuration further suggests that such heuristics may remain stable within a given model.

At the same time, our findings point to the importance of a degree of *theoretical* understanding. Theoretical knowledge can guide intervention—for example, recognizing that text conditioning is mediated through cross-attention layers may motivate their manipulation when semantic modification is desired, such as replacing attention weights mid-inference [18]. Conversely, direct engagement with and manipulation of the model can reinforce *theoretical* insight. Observations such as the stronger influence of layer type over spatial depth, or the disproportionate impact of early denoising steps on global structure, reflect structural properties of diffusion architectures. Attending to these characteristics moves beyond surface-level parameter exploration toward a more principled mode of manipulation—one that may enable outputs extending beyond the model's original design constraints [46].

Taken together, these considerations suggest that art creation tools exposing generative model structure and enabling direct intervention may support the development of both *behavioural* and *theoretical* understanding. In addition, lightweight in-context explanations—such as tooltips describing the functional role of specific layers or modules—could further support *theoretical* understanding by situating experimentation within an architectural framework. Such design elements may help users connect observed visual effects to underlying model components without requiring extensive prior technical knowledge. However, these remain design suggestions rather than empirically validated outcomes. Future work involving artist studies or longitudinal deployments would be required to assess how such interfaces shape creative workflows in practice.

5.5 Model Bending and AI Literacy

Creative projects have often been used to help the public engage with and understand technology—for example, game development has been widely adopted as a way of teaching programming concepts. In a similar way, artistic experimentation that involves manipulating large-scale generative models may offer an accessible path toward understanding how these systems work. By exposing and intervening in architectural components, artists and audiences can move beyond viewing generative AI

as a "black box" and instead engage with it as material, similar in spirit to early circuit bending or hacking. Future work could explore installations, workshops, or participatory art experiences that contribute to AI literacy through model bending, in line with previous explorations [47]. Towards that end, we created a public website with precomputed bending results: `https://diffusion-bending-demo.netlify.app`

6 Limitations

This work focuses on a single model architecture (SD1.5) and a specific UNet configuration. As such, the observations reported here should not be assumed to generalize to other diffusion models, model versions, or architectures. Differences in training data, schedulers, or architectural design may lead to different bending behaviours. Our evaluation relies primarily on qualitative inspection supported by quantitative distance metrics. While measures such as latent distance provide a useful proxy for impact, they do not fully capture perceptual, aesthetic, or semantic aspects of the generated outputs. Complementary metrics could offer a more comprehensive assessment. Ultimately, human studies would be required to evaluate how bending effects are perceived and interpreted in practice.

7 Ethics Statement

In this work, we explore the implications of model bending for creative exploration and AI explainability. Given the wide range of effects that bending can produce, it is possible that such interventions could bypass or disrupt guardrails placed on publicly deployed models, e.g., through Reinforcement Learning [48]. At the same time, this capacity highlights bending's potential as a tool for probing the limits, robustness, and reliability of generative systems.

8 Future Work

Future work could extend the interface to support additional diffusion models and architectural variants beyond SD1.5, enabling comparisons across systems. Optional visualizations of intermediate representations—such as layer activations or attention maps—could be added for users who want a closer look at how the model operates during inference. These features would remain optional, allowing users to continue experimenting based solely on outputs without requiring a technical understanding of model function.

From a research perspective, collaborations with artists and longer-term deployments would help assess how model bending fits into creative workflows. Crowd-

sourced studies could also explore shared preferences regarding which model components produce compelling transformations, helping refine interface design and further develop the systematic mapping approach introduced in this work.

9 Conclusion

This paper reframes explainability for generative art as a form of craft knowledge built through manipulation rather than post-hoc interpretation. We argued that large diffusion models need not remain opaque commodities: when their structure can be inspected and intervened on, they can be treated as creative materials. To support this stance, we presented a ComfyUI-integrated model-bending toolkit that lets artists select internal components of a Stable Diffusion–style pipeline and apply training-free interventions, alongside an interactive interface for navigating model structure and iterating on bends. We also reported a preliminary, systematic exploration of bending in SD1.5 that combines qualitative examples with latent-space distance measures, highlighting that impact varies substantially by component type, region, and denoising-step range, and that many effects remain relatively consistent across prompts and seeds within a given configuration.

Together, these contributions suggest a practical path toward ”doing-based” XAI in the arts: artists can develop actionable heuristics—what to bend, when to bend, and by how much—while also gaining architectural understanding through repeated engagement. Future work should evaluate these tools in longitudinal artist studies, expand coverage to newer diffusion architectures and additional intervention types, and support community sharing of model-specific ”cheat sheets” and workflows. Ultimately, we argue that model-crafting tools like the one presented here—tools that encourage artists to treat large-scale generative models as creative materials—may contribute to ongoing conversations around authorship and agency in GenAI-assisted arts.

References

[1] David Gunning et al. “XAI—Explainable artificial intelligence”. In: *Science robotics* 4.37 (2019).

[2] Nick Bryan-Kinns et al. “Exploring XAI for the Arts: Explaining Latent Space in Generative Music”. In: *1st Workshop on eXplainable AI approaches for debugging and diagnosis*. 2023.

[3] Austin Tecks, Thomas Peschlow, and Gabriel Vigliensoni. “Explainability Paths for Sustained Artistic Practice with AI”. In: *In Proceedings of Explainable AI for the Arts Workshop 2024 (XAIxArts 2024)*. arXiv, July 21, 2024. DOI: 10.48550/arXiv.2407.15216. arXiv: 2407.15216 [cs]. URL: http://arxiv.org/abs/2407.15216 (visited on 04/08/2025).

[4] Nick Bryan-Kinns et al. “XAIxArts Manifesto: Explainable AI for the Arts”. In: *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. CHI. Yokohama, Japan, 2025. DOI: 10.1145/3706599.3716227. arXiv: 2502.21220 [cs]. URL: http://arxiv.org/abs/2502.21220 (visited on 04/13/2025).

[5] Gabriel Vigliensoni, Phoenix Perry, and Rebecca Fiebrink. “A Small-Data Mindset for Generative AI Creative Work”. In: *In Generative AI in HCI Workshop, CHI ’22*. New York, NY, USA, 2022, p. 5.

[6] Baptiste Caramiaux and Sarah Fdili Alaoui. “”Explorers of Unknown Planets” Practices and Politics of Artificial Intelligence in Visual Arts”. In: *Proceedings of the ACM on Human-Computer Interaction* 6.CSCW2 (2022), pp. 1–24.

[7] Terence Broad. “Using Generative AI as an Artistic Material: A Hacker’s Guide”. In: *In Proceedings of Explainable AI for the Arts Workshop 2024*. 2024.

[8] Sonja Rozental, Michel Van Dartel, and Alwin De Rooij. “How Artists Use AI as a Responsive Material for Art Creation”. In: ISEA 2025: 30th International Symposium on Electronic/Emerging Art. Korea, 2025. DOI: 10.31234/osf.io/gjdnw. URL: https://osf.io/gjdnw (visited on 06/10/2025).

[9] Ahmed M. Abuzuraiq and Philippe Pasquier. “Seizing the Means of Production: Exploring the Landscape of Crafting, Adapting and Navigating Generative AI Models”. In: *the 3rd Generative AI and HCI Workshop*. CHI ’24. Hawaii, USA, 2024.

[10] Terence Broad, Frederic Fol Leymarie, and Mick Grierson. “Network Bending: Expressive Manipulation of Deep Generative Models”. In: *Artificial Intelligence in Music, Sound, Art and Design: 10th International Conference, EvoMUSART 2021, Held as Part of EvoStar 2021, Virtual Event, April 7–9, 2021, Proceedings 10*. Springer, 2021.

[11] comfyanonymous. https://github.com/comfyanonymous/ComfyUI. 2024. (Visited on 02/13/2024).

[12] Reed Ghazala. *Circuit-Bending: Build Your Own Alien Instruments*. John Wiley & Sons, 2005.

[13] Jonas Kraasch and Philippe Pasquier. “Autolume-Live: Turning GANs into a Live VJing Tool”. In: *Proceedings of the 10th Conference on Computation, Communication, Aesthetics & X*. Coimbra, Portugal, 2022, pp. 152–169.

[14] Metacreation Lab. *Autolume: A Neural-network Based Visual Synthesizer*. https://www.metacreation.net/autolume. 2024. (Visited on 02/13/2024).

[15] Ilia Pavlov. “Controlling the Image Generation Process with Parametric Activation Functions”. In: *International Conference on Computational Creativity ICCC25*. Brazil, 2025.

[16] Luke Dzwonczyk, Carmine Emanuele Cella, and David Ban. “Network Bending of Diffusion Models for Audio-Visual Generation”. In: 27th International Conference on Digital Audio Effects (DAFx24). arXiv, 2024-06-28. DOI: 10.48550/arXiv.2406.19589. arXiv: 2406.19589 [cs]. URL: http://arxiv.org/abs/2406.19589 (visited on 02/20/2025).

[17] Garin Curtis. *Beyond Prompts: Developing An Expressive Tool for Real-Time Manipulation of Diffusion Models through Active Divergence*. 2025. URL: https://garincurtis.com/projects/beyond-prompts.

[18] Imke Grabe et al. "Patch Explorer: Interpreting Diffusion Models through Interaction". In: *Mechanistic Interpretability for Vision at CVPR 2025 (Non-proceedings Track)*. Mar. 31, 2025. URL: https://openreview.net/forum?id=0n9wqVyHas (visited on 07/02/2025).

[19] Mingi Kwon, Jaeseok Jeong, and Youngjung Uh. *Diffusion Models Already Have a Semantic Latent Space*. Mar. 29, 2023. DOI: 10.48550/arXiv.2210.10960. arXiv: 2210.10960 [cs]. URL: http://arxiv.org/abs/2210.10960 (visited on 05/07/2025). Pre-published.

[20] Nick Collins. "Unstable Audio: Code Bending Text-to-Music Generation". In: *Proceedings of the AES International Conference on Machine Learning and Artificial Intelligence for Audio*. 2025. URL: https://durham-repository.worktribe.com/output/4280622 (visited on 11/20/2025).

[21] Matthew Yee-King and Louis McCallum. "Studio Report: Sound Synthesis with DDSP and Network Bending Techniques". In: *Proceedings of 2nd Conference on AI Music Creativity. AIMC*. 2021.

[22] Jaden Fiotto-Kaufman et al. *NNsight and NDIF: Democratizing Access to Open-Weight Foundation Model Internals*. Apr. 1, 2025. DOI: 10.48550/arXiv.2407.14561. arXiv: 2407.14561 [cs]. URL: http://arxiv.org/abs/2407.14561 (visited on 02/12/2026). Pre-published.

[23] Zhengxuan Wu et al. "Pyvene: A Library for Understanding and Improving PyTorch Models via Interventions". In: *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: System Demonstrations)*. Ed. by Kai-Wei Chang, Annie Lee, and Nazneen Rajani. Mexico City, Mexico: Association for Computational Linguistics, June 2024, pp. 158–165. DOI: 10.18653/v1/2024.naacl-demo.16. URL: https://aclanthology.org/2024.naacl-demo.16/.

[24] David Bau. *Baukit*. 2022. URL: https://github.com/davidbau/baukit..

[25] Giacomo Aldegheri et al. "Hacking Generative Models with Differentiable Network Bending". 2023. arXiv: 2310.04816. URL: https://github.com/GAldegheri/gan-bending.

[26] Arshia Sobhan et al. "Autolume: A GAN-based No-Coding Small Data and Model Crafting Visual Synthesizer for Artistic Creation". In: *Proceedings of the Conference on Animation and Interactive Art*. Expanded '25. New York, NY, USA: Association for Computing Machinery, Nov. 5, 2025, pp. 165–172. ISBN: 979-8-4007-1532-7. DOI: 10.1145/3749893.3749968. URL: https://dl.acm.org/doi/10.1145/3749893.3749968 (visited on 02/11/2026).

[27] Imke Grabe and Tom Jenkins. "Hidden Layer Interaction: A Technique to Explore the Material of Generative AI". In: *Proceedings of the 2025 ACM Designing Interactive Systems Conference*. DIS '25. New York, NY, USA: Association for Computing Machinery, July 4, 2025, pp. 1913–1927. ISBN:

979-8-4007-1485-6. DOI: 10.1145/3715336.3735437. URL: https://dl.acm.org/doi/10.1145/3715336.3735437 (visited on 07/14/2025).

[28] Robin Rombach et al. "High-Resolution Image Synthesis With Latent Diffusion Models". In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022, pp. 10684–10695. URL: https://openaccess.thecvf.com/content/CVPR2022/html/Rombach_High-Resolution_Image_Synthesis_With_Latent_Diffusion_Models_CVPR_2022_paper (visited on 03/08/2025).

[29] Automatic1111 WebUI. https://github.com/AUTOMATIC1111/stable-diffusion-webui. 2024. (Visited on 02/13/2024).

[30] Akio Kodaira et al. "Streamdiffusion: A pipeline-level solution for real-time interactive generation". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2025, pp. 12371–12380.

[31] Richard Zhang et al. "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric". In: *CVPR*. 2018.

[32] Maxime Oquab et al. "Dinov2: Learning robust visual features without supervision". In: *arXiv preprint arXiv:2304.07193* (2023).

[33] Alec Radford et al. "Learning transferable visual models from natural language supervision". In: *International conference on machine learning*. PmLR. 2021, pp. 8748–8763.

[34] Julian Hummel. "Visual Parameter Space Exploration for AI Art Design". B.S. thesis. 2023.

[35] Michael Sedlmair et al. "Visual Parameter Space Analysis: A Conceptual Framework". In: *IEEE Transactions on Visualization and Computer Graphics* 20.12 (Dec. 2014), pp. 2161–2170. ISSN: 1077-2626. DOI: 10.1109/TVCG.2014.2346321.

[36] Richard Lee Davis et al. "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*. Chi '24. New York, NY, USA: Association for Computing Machinery, 2024. ISBN: 979-8-4007-0330-0. DOI: 10.1145/3613904.3642908. URL: https://doi.org/10.1145/3613904.3642908.

[37] David Ledo. "Generative Rotoscoping: A First-Person Autobiographical Exploration on Generative Video-to-Video Practices". In: *Proceedings of the 2025 Conference on Creativity and Cognition*. C&C '25. New York, NY, USA: Association for Computing Machinery, June 22, 2025, pp. 931–948. ISBN: 979-8-4007-1289-0. DOI: 10.1145/3698061.3726926. URL: https://dl.acm.org/doi/10.1145/3698061.3726926 (visited on 06/25/2025).

[38] *Civitai: The Home of Open-Source Generative AI*. https://civitai.green/. 2025. (Visited on 04/29/2025).

[39] *Tensor.art: AI model sharing platform*. https://tensor.art/. 2025. (Visited on 04/29/2025).

[40] Edward J Hu et al. “Lora: Low-rank Adaptation of Large Language Models”. 2021. arXiv: 2106.09685.

[41] StabilityAI. *Stable Diffusion*. https://stability.ai/stable-image. 2024. (Visited on 08/30/2024).

[42] Abdullah A Abonamah, Muhammad Usman Tariq, and Samar Shilbayeh. “On the commoditization of artificial intelligence”. In: *Frontiers in psychology* 12 (2021).

[43] Adriana Fernández de Caleya Vázquez and Eduardo C Garrido-Merchán. “A Taxonomy of the Biases of the Images created by Generative Artificial Intelligence”. In: *Current Trends in Business Management 3* (2024).

[44] Reishiro Kawakami and Sukrit Venkatagiri. “The Impact of Generative AI on Artists”. In: *Proceedings of the 16th Conference on Creativity & Cognition*. 2024, pp. 79–82.

[45] Aditya Bhattacharya. “Towards Directive Explanations: Crafting Explainable AI Systems for Actionable Human-AI Interactions”. In: *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*. 2024, pp. 1–6.

[46] Terence Broad et al. “Active Divergence with Generative Deep Learning–A Survey and Taxonomy”. In: *Proceedings of the 12th International Conference on Computational Creativity (ICCC ’21)*. 2021. ISBN: 978-989-54-1603-5. DOI: arXivpreprintarXiv:2107.05599.

[47] Drew Hemment et al. “AI in the public eye: Investigating public AI literacy through AI art”. In: *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*. 2023, pp. 931–942.

[48] Long Ouyang et al. *Training Language Models to Follow Instructions with Human Feedback*. Mar. 4, 2022. DOI: 10.48550/arXiv.2203.02155. arXiv: 2203.02155 [cs]. URL: http://arxiv.org/abs/2203.02155 (visited on 11/28/2025). Pre-published.