Patch Explorer: Interpreting Diffusion Models through Interaction

Imke Grabe*
IT University of Copenhagen
Copenhagen, Denmark
imgr@itu.dk

Jaden Fiotto-Kaufman*
Northeastern University
Boston, Massachusetts
j.fiotto-kaufman@northeastern.edu

Rohit Gandikota
Northeastern University
Boston, Massachusetts
gandikota.ro@northeastern.edu

David Bau
Northeastern University
Boston, Massachusetts
d.bau@northeastern.edu

Figure 1. **Interface components of Patch Explorer.** After (1) providing a text prompt and a seed to generate (2) an image, users can inspect the (3) contribution of the individual cross-attention heads visualized as patch grids along three axes. In the current 2D-view, the third axis is collapsed, so that users see all timesteps overlayed for a selected (4) timestep range. Users can choose to (5) apply an intervention to (6) selected patch grids, as we will demonstrate in more detail in Fig. 4.

Abstract

We introduce Patch Explorer, an interactive interface for visualizing and manipulating the patches as they are processed by cross-attention heads. Built on interventions via NNsight, our interface lets users inspect and manipulate individual attention heads over layers and timesteps. Interaction via the interface reveals that attention heads independently capture semantics, like a unicorn's horn, in diffusion models. Next to offering a way to analyze its behavior, users can also intervene with Patch Explorer to edit semantic associations within diffusion models, like adding a unicorn horn to a horse. Our interface also helps understand the role of a diffusion timestep through precise interventions. By providing a visualization tool with interactivity based on attention heads, we aim to shed light on their role in generative processes.

*Equal contribution.

1. Introduction

Through careful visualization and model interventions, can we reveal the hidden knowledge structure of a diffusion model? Understanding the mechanisms of text-to-image diffusion models has been mainly oriented towards visualizing cross-attention heatmaps as they link the encoded text prompt provided by a user to the image patches being generated [4, 7, 9, 17, 17]. However, visualizing the attention heatmap for a token in the prompt restricts us from revealing the hidden semantic structure of the model. For example, when prompted for "unicorn," can we visualize if the model thinks of "horn" without the user explicitly prompting or probing for it?

Latent diffusion takes place in a latent space, where images are processed in patches. We propose a powerful tool for visualizing and intervening the cross-attention patches of diffusion model for interpreting how a diffusion model processes information. By visualizing how patches are altered by individual attention heads and intervening on them, we show that we can localize the effects of cross-attention to patches. For example, we find that given the prompt "unicorn," certain heads of the model are responsible for generating a horn without explicitly mentioning it in the text prompt.

We propose (1) a method to intervene on individual attention heads transform patches according to text in the diffusion models denoising process and show that approaching attention heads offers opportunities for both interpreting semantics and controlling them during interaction. To do so, we present (2) Patch Explorer[1], a user interface designed to visualize and manipulate individual attention heads via patches. Through interaction with our application, we find (3) that individual attention heads might be responsible for specific semantic or structural attributes in generated images and advocate for focusing on attention heads as a means of interpreting and interacting with diffusion models.

2. Related Work

Interpretability of Diffusion Models: Prior work has investigated various components of diffusion models to enhance their interpretability [10]. At the architectural level, Liu et al. [11] analyze self- and cross-attention blocks, while Kim et al. [8] reveal semantic hierarchies in up-blocks, demonstrating how different model components capture information at varying levels of granularity. The layer-wise investigation by Agarwal et al. [1] examines the relationship between layers and timesteps, revealing how representations evolve during the denoising process. At a finer scale, Liu et al. [12, 13] identify concept neurons through their CONES framework, enabling targeted manipulation of specific semantic elements. Temporal analysis by Zhang et al. [20] shows that earlier timesteps govern semantic content while later steps refine visual details, providing insights into the staged nature of diffusion generation. These works inform our Patch Explorer approach, which focuses specifically on cross-attention heads as modulable units that directly link text embeddings to image patches across both layers and timesteps.

Visualization and Interactive Approaches: The visualization of diffusion models has centered largely on cross-attention maps as they connect text prompts to image generation [4–6, 9, 14, 17]. Hertz et al. [7] introduce Prompt-to-Prompt for cross-attention control at the prompt level, while Fiotto-Kaufman et al. [2] analyze how blocks contribute to concept formation. Park et al. [15] provide tools for visualizing denoising levels to enhance interpretability for non-experts. These approaches typically reveal only the relationship between explicit tokens and images, missing hidden semantic structures. Other visualization tools for Vision Transformers like AttentionViz [19] use dimensionality reduction to reveal how attention heads capture visual properties, while Darkspark[2] visualizes broader semantic concepts. Our Patch Explorer differs by providing an interactive interface that both visualizes and enables intervention on specific cross-attention heads, revealing semantic associations not explicitly specified in prompts (such as "horn" from "unicorn") and allowing precise manipulation of patch hidden states during the diffusion process.

3. Background

In this section, we provide an overview of the key components and mechanisms that form the foundation of modern latent diffusion models.

3.1. Latent Representations and Patch Embeddings

Latent diffusion models [16] operate in a compressed latent space rather than directly in pixel space. This design choice significantly reduces computational complexity while preserving generative capabilities. The latent space is organized into **patches**, which are spatial units that correspond to regions in the output image. Unlike the sequential tokens in language transformers, patches in vision transformers capture a 2D grid-like spatial structure [19]. Each patch is represented by an embedding vector that encodes the visual information of that region in a high-dimensional feature space.

[1] The interface is available at https://patch.baulab.info/.

[2] https://darkspark.dev/

3.2. Attention Mechanisms

At the core of diffusion models is the **attention** mechanism, which enables content-based interactions between different spatial locations. The self-attention operation is formally defined as:

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

where Q, K, and V represent query, key, and value matrices, respectively, and d is the dimension of the key vectors. The scaling factor $\sqrt{d}$ prevents the softmax function from entering regions with extremely small gradients.

In the cross-attention layers of diffusion models, the K and V matrices are derived from text encodings, while Q comes from the image representation. This mechanism creates an alignment between textual descriptions and visual features, enabling text-conditioned image generation.

3.3. Multi-Head Attention

To capture diverse relationships simultaneously, latent diffusion models employ **multi-head attention** [18]. This approach splits the attention mechanism into multiple parallel heads, each focusing on different aspects of the input:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O \quad (2)$$

where each head computes:

$$\text{head}_\text{i} = \text{Attention}(hW_i^Q, hW_i^K, hW_i^V) \quad (3)$$

The weight matrices W_i^Q, W_i^K, and W_i^V project the input h into different subspaces, allowing each head to focus on different interaction patterns. The outputs of all heads are concatenated and linearly transformed by W^O before being added to the model's residual stream.

3.4. U-Net Architecture

Latent diffusion models typically adopt a U-Net architecture with varying spatial resolutions across different blocks. As shown in Table 1, the model progressively downsamples the spatial resolution in the encoder path and then upsamples it in the decoder path. This multi-scale approach allows the model to capture both fine details and global context.

4. Patch Explorer: Interface Design

Since multi-head attention blocks perform several attention functions in parallel, we propose an interface to analyze patches by considering each cross-attention head individually.

Blocks	Layers	Patches
Down-1	1, 2	64×64
Down-2	3, 4	32×32
Down-3	5, 6	16×16
Mid	7	8×8
Up-1	8, 9, 10	16×16
Up-2	11, 12, 13	32×32
Up-3	14, 15, 16	64×64

Table 1. The U-Net architecture of Stable Diffusion 1.4, showing layers and their corresponding spatial resolutions. All layers consist of 8 cross-attention heads.

4.1. Visualizing Attention Heads as Patch Grids

To investigate the contribution of an attention head to the residual, we consider its output before it is added to the residual stream and after the linear projection following the multiplication with V. The output consists of multi-dimensional latents for each patch. By summing these activations for each patch and arranging them spatially, we create a *Patch Grid*, which indicates how much an attention head contributes to a patch region (see Fig. 2). In the visualization, patches of positive activation are represented as magenta, patches of negative activation as cyan squares. We visualize the activation for each patch by altering its transparency: Full opacity represents maximum activation ranging to full transparency for no activation.

The patch grids for the heads ($H1 - H8$) are arranged vertically for each of the layers ($L1 - L16$), which are placed in ascending order from left to right, revealing the shape of the U-Net in Fig. 1. Users can choose between 2D and 3D view: The 2D view lets users traverse through the layers without perspective but with the help of a slider determining the range of visible timesteps overlayed. In the 3D view, users can move along the three axes from a perspective projection using panning.

4.2. Interventions via Patch Grids

Besides visualization, the patch grids function as a tool to apply interventions to patches of a specific attention head. To select a patch grid, the user clicks on it in the visualization. They can then choose specific patches by drawing in the grid, as described in Fig. 3. Interventions will be applied to all marked patches. In the visualization, we use the color green to indicate the selected patches. The same slider that is used for visualization, is used to indicate the timestep range to which the intervention is applied.

Users can choose between two different types of interventions to influence the attention head's output, as depicted in Fig. 2. **Scaling** applies a scalar to the selected patches, which can be used to ablate, increase or decrease the output of an attention head, localized to latent patches. **En-**

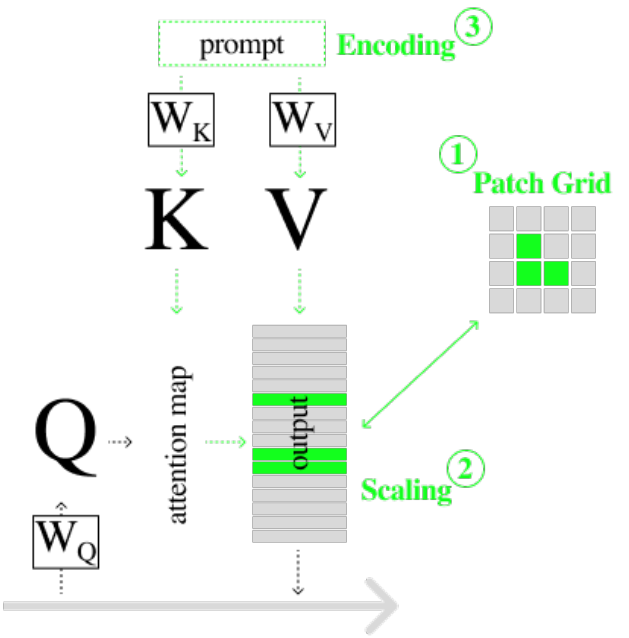


Figure 2. A (1) **Patch Grid** offers a representation to spatially target the outputs of attention heads with interventions. (2) **Scaling** multiplies the output of the attention head by a given *factor*. (3) **Encoding** replaces the output for targeted patches with the output for an alternative *text encoding* provided by the user.

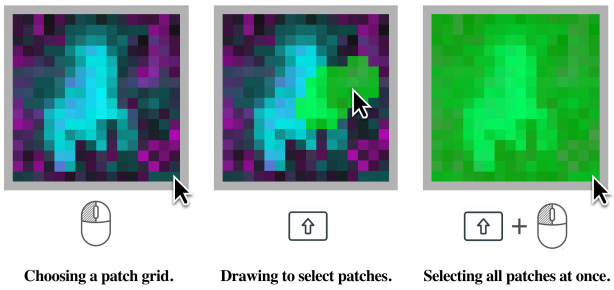


Figure 3. **Applying interventions.** The user can target an intervention to a range of timesteps via the slider, and to attention heads by interacting with the patch grid. By clicking on a patch grid in the interface, the user can mark selected patches by holding the Shift key and "drawing" on patches, which changes their color to green. Double-clicking marks all patches in a grid at once.

coding interventions take a secondary prompt as input, and recalculate a selected attention head's output given the new prompt. The recalculated output then overwrite the original run's output for the whole head or only at selected patch locations.

4.3. Implementation Details

The Patch Explorer application runs on an HTTP server back-end (FastAPI) hosting an instance of StableDiffusion

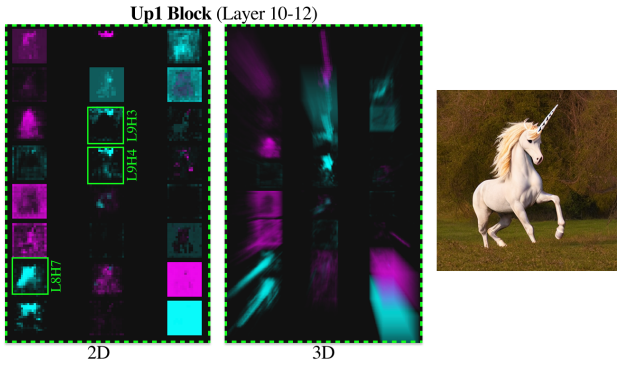


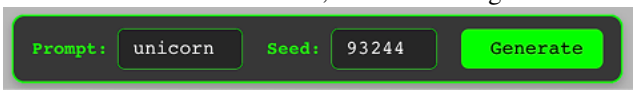
Figure 4. **Zoom in on Up1 Block**. While the 2D overlays all timesteps for a selected range, the 3D view shows all 50 timesteps stacked along the third dimension.

1.4 in 16bit precision. A request endpoint ingests the prompt, intervention type, parameters, and intervention locations as (x,y) coordinates of a selected layer and head. The *NNsight* [3] library is used to apply interventions during the diffusion process, intervening by editing the computation graph of the diffusion model and enabling the injection of patch-based intervention logic at selected attention heads and timesteps, while also caching the attention head's output. The front-end uses Vue and ThreeJS for the visualization. Currently, only one intervention is supported at the same time.

5. Interpretability through Interaction

In this section, we demonstrate how the interactive capabilities of Patch Explorer can enhance our understanding of diffusion models through a step-by-step walkthrough.

Consider a user seeking to identify components that encode semantic knowledge with the goal of understanding diffusion model mechanisms, with goal to understand: ***How does the model create specific visual concepts, like the horn of a unicorn?*** The exploration begins by generating a "unicorn" with a random seed, as shown in Fig. 1.



The interface then displays the generated image and the visualization of the cross-attention heads' output. The user can inspect the attention heads in 2D or 3D mode, as depicted in Fig. 4, where we zoom into the model's first up-block for the following analysis.

5.1. Finding Semantic Features in Attention Heads

Through visual inspection of the patch grids, the user identifies several attention heads that activate in specific regions corresponding to semantic features. For unicorns, some attention heads, like L9H3 (Head 3 of Layer 9) and L9H4,

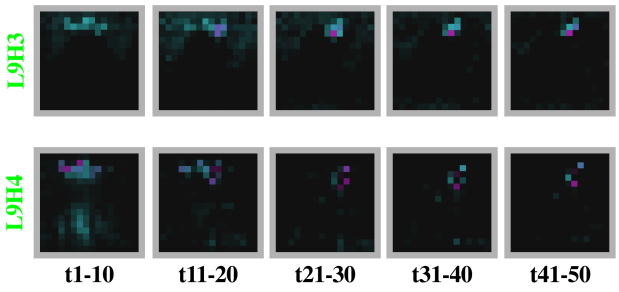


Figure 5. **Inspecting L9H3 and L9H4 over timesteps.** By adjusting the timestep slider, the user can "move" along the z-axis to have the 2D view show only the activation over selected timestep ranges, revealing how the horn feature evolves over time.

activate especially around the horn.

To better understand how the head's contribution evolves over timesteps, the user adjusts the timestep slider, as shown in Fig. 5. They observe that while attention is spread widely over the patches for early ranges t1-10 and t11-20, the attention heads' focus becomes more localized around the horn towards later timesteps. The evolution over timesteps becomes even more apparent when switching to 3D-view (Fig. 4), where the patches surrounding the unicorn's horn resemble a spike advancing toward the viewer as timesteps increase.

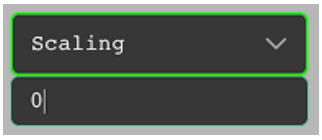
This initial observation leads the user to hypothesize that these attention heads are connected to the generation of the unicorn's horn. To investigate further, they apply interventions to interact with the attention heads' behavior, as depicted in Fig. 6.

Similarly, while exploring different concepts, the user notices that for horses and pegasi that appear very similar with the same seed (931911), head L8H7 activates distinctly in regions corresponding to the wings in pegasi. This suggests that L8H7 might be responsible for generating the wings feature.

5.2. Validating Feature Attribution with Scaling

To confirm if the identified attention heads are indeed related to specific semantic features, the user applies Scaling interventions.

For the unicorn's horn, the user selects the Scaling intervention and types "0" as the factor to apply.



This will ablate a heads' contribution to the residual. To apply this intervention to the heads, the user clicks on the two patch grids (L9H3 and L9H4) and selects all patches, either by double-clicking each grid or by holding Shift

Figure 6. **Applying interventions**. To apply an intervention, like Encoding, the user (1) types in the prompt, (2) selects the timestep range, and (3) marks the patches to apply it to.

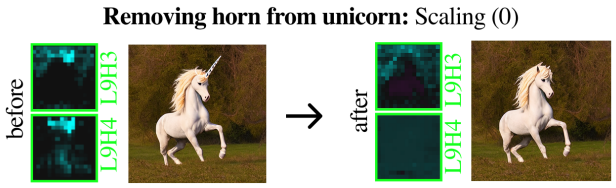


Figure 7. **Applying Scaling at L9H3 and L9H4.** The user types in a factor and selects the targeted attention heads. The intervention alters the unicorn to lose its horn.

while "drawing" on the patches to mark them. After selecting the patches, they initiate a new generation, keeping the same prompt and seed but with the intervention applied.

Without adjusting the timestep slider, the intervention is applied to all timesteps by default. The result shows a major difference: while the attention heads activated strongly around the horn in the first run, the Scaling intervention ablated their effect. Now, the attention heads' activation appears weak with no particular spatial focus as shown in Fig. 7.

To verify this effect across different examples, the user repeats the process with other seeds. As shown in Fig. 8, ablating these heads consistently removes the horn from all unicorns, confirming that these attention heads are responsible for the horn feature. Additionally, increasing the scaling factor to 2 amplifies the horn feature, making it larger or creating multiple horns.

Curious about the generalizability of these findings, the user wonders whether the same attention heads are re-

Figure 8. **Scaling L9H3 and L9H4.** The intervention amplifies or removes the horn from unicorns, confirming these heads' role in horn generation.

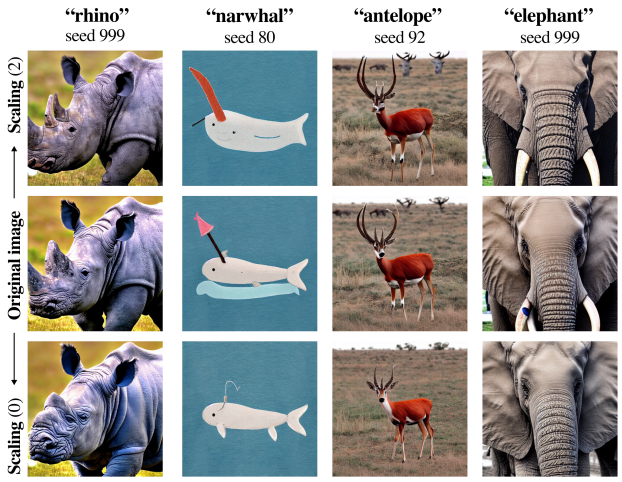


Figure 9. **Testing Generalisability of L9H3 and L9H4.** The intervention amplifies or removes the horn from other horned animals, confirming these heads' general role in horn generation.

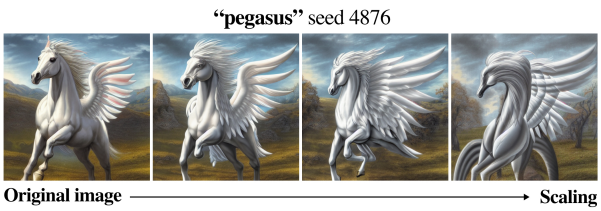


Figure 10. **Scaling L8H7.** Increasing the scaling factor amplifies the pegasus' wings, confirming this head's association with wing generation.

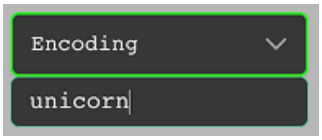
sponsible for generating horns in other animals, not just unicorns. To investigate this, they generate images with prompts for various horned animals like "rhinoceros," "elephant," "antelope," and "narwhal" (often called the "unicorn of the sea"). Using the visualization, they observe that the same attention heads (L9H3 and L9H4) consistently show strong activation in the horn regions across these different species. As shown in Figure 9, when applying the Scaling intervention with factor 0 to these heads, the horns are significantly reduced or removed from all these animals, while amplifying with factor 2 results in more prominent or multiple horns. This confirms that the model uses the same attention mechanism to generate horn-like features across diverse animal concepts, suggesting that these heads encode a generalized representation of "horn-ness" rather than being specific to unicorns alone. In the Appendix, we show that other heads in the same layer correspond more to "antlers" (another form of horn).

Similarly, for the pegasus' wings, the user applies a Scaling intervention to L8H7. Increasing the scaling factor notably amplifies the wing features (Fig. 10), confirming that this head is strongly associated with wing generation.

5.3. Transferring Features Across Concepts

Having discovered attention heads responsible for specific features, the user explores whether these components can transfer semantic features between related concepts.

To test this, the user changes the original prompt to generate a "horse" and applies the Encoding intervention with the text "unicorn" to heads L9H3 and L9H4.



After typing the desired prompt and selecting the targeted attention heads, they initiate a new generation.

As shown in Fig. 11, the result successfully adds horns to the horse. To see if this works for other horse-like concepts, the user then changes the original prompt from "horse" to "pegasus" (a winged horse in Greek mythology) and repeats the intervention. As shown in Fig. 12, the intervention consistently adds horns to different horse-like animals while preserving their overall appearance.

Interestingly, the transferred horns adapt to match each animal's appearance in size, texture, and color. Some animals receive multiple horns, with their exact appearance varying based on the animal's existing features. For cases where multiple unwanted horns appear, the user can apply more precise control through patch-based interventions.

To do this, the user selects specific patches by clicking on the patch grid and holding Shift while "drawing" on in-

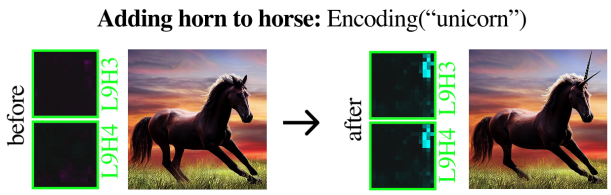


Figure 11. **Applying Encoding ("unicorn") at L9H3 and L9H4.** The attention head focus around the forehead as the intervention adds horn to a horse.

Figure 12. Encoding the prompt "unicorn" at L9H3 and L9H4 adds horns to horses and pegasi.

Figure 13. **Evolution of horn over timesteps.** By Encoding "unicorn" at L9H3 and L9H4 until a certain timestep t, we can observe how a horn grows on a horse's forehead over time.

Figure 14. **Evolution of wings over timesteps.** By ablating the contribution of L8H7 for an increasing number of timesteps (read from right to left), we can inspect how this pegasus' wings were formed over timesteps.

dividual patches, as shown in Fig. 3. This enables targeting specific spatial areas rather than the entire attention head, allowing more granular manipulation of feature transfers.

5.4. Tracing Features Develop Across Timesteps

To understand precisely how features develop during the generation process, the user applies interventions to specific timestep ranges.

Using the timestep slider, the user can restrict interventions to particular stages of the generation process. To understand how a unicorn horn is formed on a horse's head, they apply the Encoding intervention with "unicorn" until a specific timestep for L9H3 and L9H4 while generating a horse, as shown in Fig. 13. This allows them to observe the horn's formation process: its mane changes shape at t12, taking the shape of a horn at t19, and is transformed into a horn at t25, after which details are refined and the swirl pattern emerges.

Similarly, to trace the development of the pegasus' wing, the user applies an Encoding intervention with an empty prompt to L8H7 for all timesteps after various cutoff points, as seen in Figure 14. This reveals the wing formation sequence: at t10, the horse's hair begins changing into a wing-like shape; by t12, the wing structure forms; and the following timestep adds feather details and coloring. By t13, the wings appear nearly complete, with minimal contributions from later timesteps.

These temporal interventions not only confirm which attention heads generate specific features but also reveal the precise developmental sequence of these features during the generation process.

6. Discussion and Conclusion

With Patch Explorer, we suggest to approach cross-attention heads for investigating and interacting with diffusion models. The interface allows users to visualize and manipulate individual cross-attention heads to understand their role in the generation of certain semantic features. Patch Explorer offers a target point for interpreting models through visualization and interventions to edit images. Through a use case scenario, we demonstrated how Patch Explorer benefits users to easily explore the complex diffusion models through simple interventions and localize semantic features in the model. Our work opens avenues to how humans can form an intuitive understanding of diffusion models' inner workings by inspecting and interacting with their internal components.

References

[1] Aishwarya Agarwal, Srikrishna Karanam, Tripti Shukla, and Balaji Vasan Srinivasan. An Image is Worth Multiple Words: Multi-attribute Inversion for Constrained Text-to-Image Synthesis, 2023. arXiv:2311.11919 [cs]. 2

[2] Jaden Fiotto-Kaufman. SEMANTIC REPRESENTATION IN STABLE DIFFUSION. https://github.com/JadenFiotto-Kaufman/thesis/blob/master/JadenFiottoKaufman_Thesis-Revised.pdf, 2023. 2

[3] Jaden Fiotto-Kaufman, Alexander R. Loftus, Eric Todd, Jannik Brinkmann, Koyena Pal, Dmitrii Troitskii, Michael Ripa,

Adam Belfki, Can Rager, Caden Juang, Aaron Mueller, Samuel Marks, Arnab Sen Sharma, Francesca Lucchetti, Nikhil Prakash, Carla Brodley, Arjun Guha, Jonathan Bell, Byron C. Wallace, and David Bau. NNsight and NDIF: Democratizing Access to Open-Weight Foundation Model Internals. In *ICLR 2025*. arXiv, 2025. arXiv:2407.14561 [cs]. 4

[4] Rohit Gandikota, Joanna Materzynska, Jaden Fiotto-Kaufman, and David Bau. Erasing Concepts from Diffusion Models, 2023. arXiv:2303.07345 [cs]. 2

[5] Rohit Gandikota, Joanna Materzynska, Tingrui Zhou, Antonio Torralba, and David Bau. Concept Sliders: LoRA Adaptors for Precise Control in Diffusion Models, 2023. arXiv:2311.12092 [cs].

[6] Rohit Gandikota, Hadas Orgad, Yonatan Belinkov, Joanna Materzyńska, and David Bau. Unified concept editing in diffusion models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 5111–5120, 2024. 2

[7] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-Prompt Image Editing with Cross Attention Control, 2022. arXiv:2208.01626 [cs]. 2

[8] Dahye Kim, Xavier Thomas, and Deepti Ghadiyaram. Revelio: Interpreting and leveraging semantic information in diffusion models, 2024. arXiv:2411.16725 [cs]. 2

[9] Nupur Kumari, Bingliang Zhang, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. Multi-Concept Customization of Text-to-Image Diffusion, 2023. arXiv:2212.04488 [cs]. 2

[10] Mingi Kwon, Jaeseok Jeong, and Youngjung Uh. Diffusion Models already have a Semantic Latent Space, 2023. arXiv:2210.10960 [cs]. 2

[11] Bingyan Liu, Chengyu Wang, Tingfeng Cao, Kui Jia, and Jun Huang. Towards Understanding Cross and Self-Attention in Stable Diffusion for Text-Guided Image Editing, 2024. arXiv:2403.03431 [cs]. 2

[12] Zhiheng Liu, Ruili Feng, Kai Zhu, Yifei Zhang, Kecheng Zheng, Yu Liu, Deli Zhao, Jingren Zhou, and Yang Cao. Cones: Concept Neurons in Diffusion Models for Customized Generation, 2023. arXiv:2303.05125 [cs]. 2

[13] Zhiheng Liu, Yifei Zhang, Yujun Shen, Kecheng Zheng, Kai Zhu, Ruili Feng, Yu Liu, Deli Zhao, Jingren Zhou, and Yang Cao. Cones 2: Customizable Image Synthesis with Multiple Subjects, 2023. arXiv:2305.19327 [cs]. 2

[14] Hadas Orgad, Bahjat Kawar, and Yonatan Belinkov. Editing implicit assumptions in text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7053–7061, 2023. 2

[15] Ji-Hoon Park, Yeong-Joon Ju, and Seong-Whan Lee. Explaining generative diffusion models via visual analysis for interpretable decision-making process. *Expert Systems with Applications*, 248:123231, 2024. 2

[16] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-Resolution Image Synthesis with Latent Diffusion Models, 2022. arXiv:2112.10752 [cs]. 2

[17] Raphael Tang, Linqing Liu, Akshat Pandey, Zhiying Jiang, Gefei Yang, Karun Kumar, Pontus Stenetorp, Jimmy Lin, and Ferhan Ture. What the daam: Interpreting stable diffusion using cross attention, 2022. 2

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need, 2017. arXiv:1706.03762 [cs]. 3

[19] Catherine Yeh, Yida Chen, Aoyu Wu, Cynthia Chen, Fernanda Viégas, and Martin Wattenberg. AttentionViz: A Global View of Transformer Attention, 2023. arXiv:2305.03210 [cs]. 2

[20] Wentian Zhang, Haozhe Liu, Jinheng Xie, Francesco Faccio, Mike Zheng Shou, and Jürgen Schmidhuber. Cross-Attention Makes Inference Cumbersome in Text-to-Image Diffusion Models, 2024. arXiv:2404.02747 [cs]. 2

Patch Explorer: Interpreting Diffusion Models through Interaction

Supplementary Material

Figure .15. Examples of adding horns to horses.

Figure .16. Examples of adding and removing unicorns' horns.

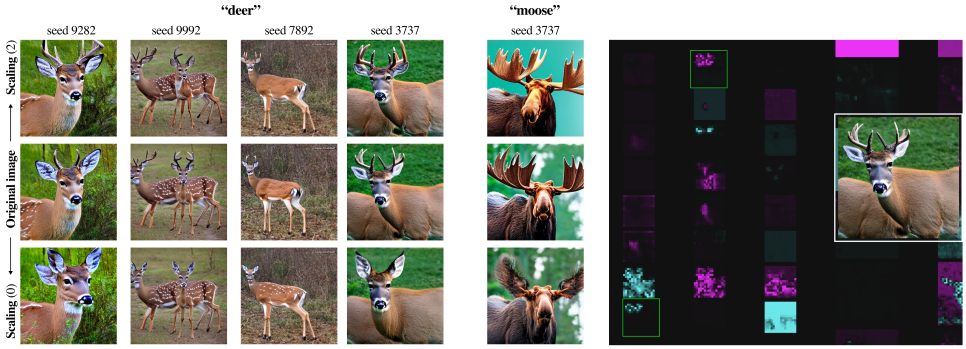


Figure .17. For another concept, antlers, we find that two other relevant attention heads to generate the feature for different animals.

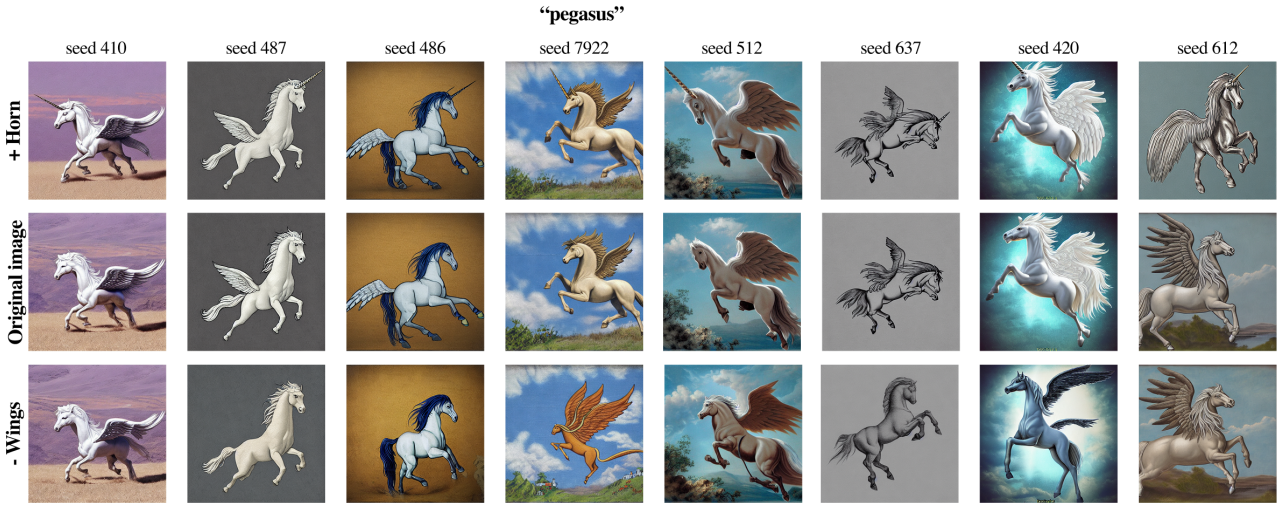


Figure .18. Examples adding horns to pegasi or removing their wings.