On Mechanistic Knowledge Localization in Text-to-Image Generative Models

Samyadeep Basu [*1] **Keivan Rezaei** [*1] **Priyatham Kattakinda** [1] **Vlad I Morariu** [2] **Nanxuan Zhao** [2]
Ryan A Rossi [2] **Varun Manjunatha** [2] **Soheil Feizi** [1]

Abstract

Identifying layers within text-to-image models which control visual attributes can facilitate efficient model editing through closed-form updates. Recent work, leveraging causal tracing show that early Stable-Diffusion variants confine knowledge primarily to the first layer of the CLIP text-encoder, while it diffuses throughout the UNet. Extending this framework, we observe that for recent models (e.g., SD-XL, DeepFloyd), causal tracing fails in pinpointing localized knowledge, highlighting challenges in model editing. To address this issue, we introduce the concept of *mechanistic localization* in text-to-image models, where knowledge about various visual attributes (e.g., "style", "objects", "facts") can be *mechanistically* localized to a small fraction of layers in the UNet, thus facilitating efficient model editing. We localize knowledge using our method LOCOGEN which measures the direct effect of intermediate layers to output generation by performing interventions in the cross-attention layers of the UNet. We then employ LOCOEDIT, a fast closed-form editing method across popular open-source text-to-image models (including the latest SD-XL) and explore the possibilities of neuron-level model editing. Using *mechanistic localization*, our work offers a better view of successes and failures in localization-based text-to-image model editing. Code will be available at https://github.com/samyadeepbasu/LocoGen.

1. Introduction

In recent years, substantial strides in conditional image generation have been made through diffusion-based text-to-image generative models, including notable examples like Stable-Diffusion (Rombach et al., 2021), Imagen (Saharia et al., 2022), and DALLE (Ramesh et al., 2021). These models have captured widespread attention owing to their impressive image generation and editing capabilities, as evidenced by leading FID scores on prominent benchmarks such as MS-COCO (Lin et al., 2014). Typically trained on extensive billion-scale image-text pairs like LAION-5B (Schuhmann et al., 2022), these models encapsulate a diverse array of visual concepts, encompassing color, artistic styles, objects, and renowned personalities.

A recent work (Basu et al., 2023) designs an interpretability framework using causal tracing (Pearl, 2001) to trace the location of knowledge about various styles, objects or facts in text-to-image generative models. Essentially, causal tracing finds the indirect effects of intermediate layers (Pearl, 2001), by finding layers which can restore a model with corrupted inputs to its original state. Using this framework, the authors find that knowledge about various visual attributes is distributed in the UNet, whereas, there exists a unique causal state in the CLIP text-encoder where knowledge is localized. This unique causal state in the text-encoder can be leveraged to edit text-to-image models in order to remove style, objects or update facts effectively. However, we note that their framework is restricted to early Stable-Diffusion variants such as Stable-Diffusion-v1-5.

In our paper, we first revisit knowledge localization for text-to-image generative models, specifically examining the effectiveness of causal tracing beyond Stable-Diffusion-v1-5. While causal tracing successfully identifies unique localized states in the text-encoder for Stable-Diffusion variants, including v1-5 and v2-1, it fails to do so for recent models like SD-XL (Podell et al., 2023) and DeepFloyd[1] across different visual attributes. In the UNet, causal states are distributed across a majority of open-source text-to-image models (excluding DeepFloyd), aligning with findings in Basu et al. (2023). Notably, for DeepFloyd, we observe a lack of strong causal states corresponding to visual attributes in the UNet.

To address the *universal* knowledge localization framework absence across different text-to-image models, we introduce the concept of *mechanistic localization* that aims to identify a small number of layers which control the generation of distinct visual attributes, across a spectrum of text-to-image

[*]Equal contribution [1]University of Maryland [2]Adobe Research. Correspondence to: Samyadeep Basu <sbasu12@umd.edu>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

[1]https://github.com/deep-floyd/IF

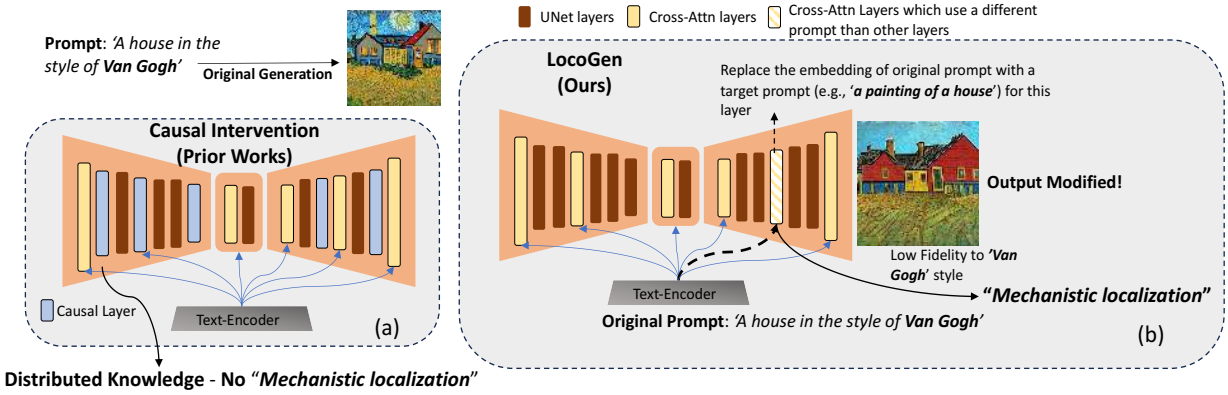


Figure 1. **LocoGen: Identifying UNet layers that, when given different input, can alter visual attributes (e.g., style, objects, facts).** (a) Earlier works (Basu et al., 2023) which show distributed knowledge using causal interventions. (b) LocoGen where a few cross-attention layers receive a different prompt-embedding than the original, leading to generation of images without the particular style.

models. To achieve this, we propose LocoGen, a method that finds a subset of cross-attention layers in the UNet such that when the input to their key and value matrices is changed, output generation for a given visual attribute (e.g., "style") is modified (see Figure 1). This intervention in the intermediate layers has a direct effect on the output – therefore LocoGen measures the direct effect of intermediate layers, as opposed to indirect effects in causal tracing.

Leveraging LocoGen, we probe knowledge locations for different visual attributes across popular open-source text-to-image models such as Stable-Diffusion-v1, Stable-Diffusion-v2, OpenJourney[2], SD-XL (Podell et al., 2023) and DeepFloyd. For all models, we find that unique locations can be identified for visual attributes (e.g., "style", "objects", "facts"). Using these locations, we then perform weight-space model editing to remove artistic "styles", modify trademarked "objects" and update outdated "facts" in text-to-image models. This weight-space editing is performed using LocoEdit which updates the key and value matrices using a closed-form update in the locations identified by LocoGen. Moreover, for certain attributes such as "style", we show that knowledge can be traced and edited to a subset of neurons, therefore highlighting the possibilities of neuron-level model editing.

Contributions. In summary, our contributions include:

- We highlight the drawbacks of existing interpretability methods such as causal tracing for localizing knowledge in latest text-to-image models.
- We introduce LocoGen which can universally identify layers that control for visual attributes across a large spectrum of open-source text-to-image models.
- By examining edited models using LocoEdit along with LocoGen, we observe that this efficient approach is successful across a majority of text-to-image models.

[2]https://huggingface.co/prompthero/openjourney

2. Related Works

Intepretability of Text-to-Image Models. To our understanding, there's limited exploration into the inner workings of text-to-image models, such as Stable-Diffusion. DAAM (Tang et al., 2023; Hertz et al., 2022) scrutinizes diffusion models through the analysis of cross-attention maps between text tokens and images, highlighting their semantic precision. (Chefer et al., 2023) understand the decomposition of concepts in diffusion models. (Basu et al., 2023) leverage causal tracing to understand how knowledge is stored in text-to-image models such as Stable-Diffusion-v1.

Editing Text-to-Image Models. The capacity to modify a diffusion model's behavior without starting from scratch was initially investigated in Concept-Ablation (Kumari et al., 2023) and Concept-Erasure (Gandikota et al., 2023). Another method, TIME (Orgad et al., 2023), alters all the cross-attention layers' key and value matrices to translate between concepts, though lacks interpretability and applications on a real-use case of model editing. (Basu et al., 2023) edits text-to-image models in the text-encoder space by leveraging a singular causal state. However, existing works overlook newer text-to-image models (e.g., SD-XL and DeepFloyd), which we delve into in detail.

3. Preliminaries

Diffusion models start with an initial random real image $\mathbf{x}_0$, the noisy image at time step t is expressed as $\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_0 + \sqrt{(1-\alpha_t)}\epsilon$. Here, α_t determines the strength of the random Gaussian noise, gradually diminishing as the time step increases, ensuring that $\mathbf{x}_T \sim \mathcal{N}(0, I)$. The denoising network $\epsilon_\theta(\mathbf{x}_t, \mathbf{c}, t)$, is pre-trained to denoise the noisy image $\mathbf{x}_t$ and produce $\mathbf{x}_{t-1}$. Typically, the con-

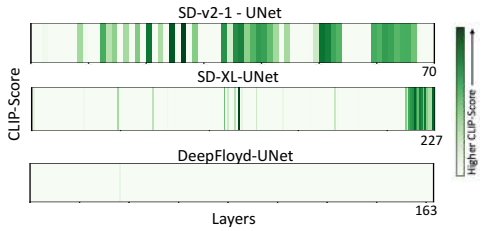


Figure 2. **Causal tracing for UNet.** Similar to (Basu et al., 2023), we find that knowledge is causally distributed across the UNet for text-to-image models such as SD-v2-1 and SD-XL. For DeepFloyd we do not observe any significant causal state in the UNet.

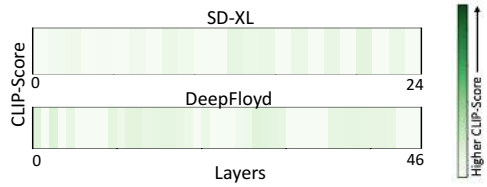


Figure 3. **Causal tracing for text-encoder.** Unlike SD-v1-5 and SD-v2-1, we find that a singular causal states does not exist in the text-encoder for SD-XL and DeepFloyd.

ditional input $\mathbf{c}$ for the denoising network $\epsilon_\theta(.)$ is a text-embedding derived from a caption c through a text-encoder, denoted as $\mathbf{c} = v_\gamma(c)$. The noising as well as the denoising operation can also occur in a latent space defined by $\mathbf{z} = \mathcal{E}(\mathbf{x})$ (Rombach et al., 2021) for better efficiency. The pre-training objective learns to denoise in the latent space as denoted by:

$$\mathcal{L}(\mathbf{z}, \mathbf{c}) = \mathbb{E}_{\epsilon,t}||\epsilon - \epsilon_\theta(\mathbf{z}_t, \mathbf{c}, t)||_2^2,$$

where $\mathbf{z}_t = \mathcal{E}(\mathbf{x}_t)$ and $\mathcal{E}$ is an encoder such as VQ-VAE (van den Oord et al., 2017).

4. On the Effectiveness of Causal Tracing for Text-to-Image Models

In this section, we empirically observe the effectiveness of causal tracing to models beyond Stable-Diffusion-v1-5. In particular, we find the ability of causal tracing to identify localized control points in Stable-Diffusion-v2-1, OpenJourney, SD-XL and DeepFloyd.

Causal Tracing in UNet. In Figure 2, we find that knowledge across different visual attributes is distributed in the UNet for all the text-to-image models (except for DeepFloyd), similar to Stable-Diffusion-v1-5. However, the degree of distribution varies between different text-to-image models. While knowledge about various visual attributes is densely distributed in Stable-Diffusion variants, for SD-XL we find that the distribution is extremely sparse (e.g., only 5% of the total layers are causal). For DeepFloyd, we observe that there are no strong causal states in the UNet. We provide more qualitative visualizations on causal tracing across the these text-to-image models in Appendix A. Overall, these results reinforce the difficulty of editing knowledge in the UNet directly due to (i) distribution of causal states or (ii) absence of any.

Causal Tracing in Text-Encoder. Basu et al. (2023) show that there exists a unique causal state in the text-encoder for Stable-Diffusion-v1-5 and Stable-Diffusion-v2-1 which can be used to perform fast model editing. In Figure 3, we find that such an unique causal state is absent in the text-encoder for DeepFloyd and SD-XL. We note that DeepFloyd uses a T5-text encoder, whereas SD-XL uses a a combination of CLIP-ViT-L and OpenCLIP-ViT-G (Radford et al., 2021). Our empirical results indicate that an unique causal state arises only when a CLIP text-encoder is used by itself in a text-to-image model.

5. LocoGen: Towards Mechanistic Knowledge Localization

Given the lack of generalizability of knowledge localization using causal tracing as shown in Section 4, we introduce LocoGen, which can identify localized control regions for visual attributes across *all* text-to-image models.

5.1. Knowledge Control in Cross-Attention Layers

During the inference process, the regulation of image generation involves the utilization of classifier-free guidance, as outlined in Ho & Salimans (2021) which incorporates scores from both the conditional and unconditional diffusion models at each time-step. Specifically, the classifier-free guidance is applied at each time-step to combine the conditional ($\epsilon_\theta(\mathbf{z}_t, \mathbf{c}, t)$) and unconditional score estimates ($\epsilon_\theta(\mathbf{z}_t, t)$). The result is a combined score denoted as $\hat{\epsilon}(\mathbf{z}_t, \mathbf{c}, t)$.

$$\hat{\epsilon}(\mathbf{z}_t, \mathbf{c}, t) = \epsilon_\theta(\mathbf{z_t}, \mathbf{c}, t) + \alpha\left(\epsilon_\theta\left(\mathbf{z_t}, \mathbf{c}, t\right) - \epsilon_\theta(\mathbf{z_t}, t)\right), \quad \forall t \in [T, 1]\,. \tag{1}$$

This combined score is used to update the latent $\mathbf{z}_t$ using DDIM sampling (Song et al., 2020) at each time-step to obtain the final latent code $\mathbf{z}_0$. We term the model $\epsilon_\theta(\mathbf{z}_t, \mathbf{c}, t)$ as the `Clean Model` and the final image generated as I_{clean}. We note that text is incorporated in the process of generation using cross-attention layers denoted by $\{C_l\}_{l=1}^M$ within $\epsilon_\theta(\mathbf{z}_t, \mathbf{c}, t)\ \forall t \in [T, 1]$. These layers include key and value matrices – $\{W_l^K, W_l^V\}_{l=1}^M$ that take text-embedding $\mathbf{c}$ of the input prompt and guide the generation toward the text prompt. Generally, the text-embedding $\mathbf{c}$ is same across all these layers. However, in order to localize and find control points for different visual attributes, we replace the original text-embedding $\mathbf{c}$ with a target prompt embedding $\mathbf{c}'$ across a small subset of the cross-attention layers and measure its direct effect on the generated image.

5.1.1. Altered Inputs

We say that a model receives *altered input* when a subset of cross-attention layers $C' \subset \{C_l\}_{l=1}^M$ receive a different text-embedding $\mathbf{c}'$ than the other cross-attention layers that

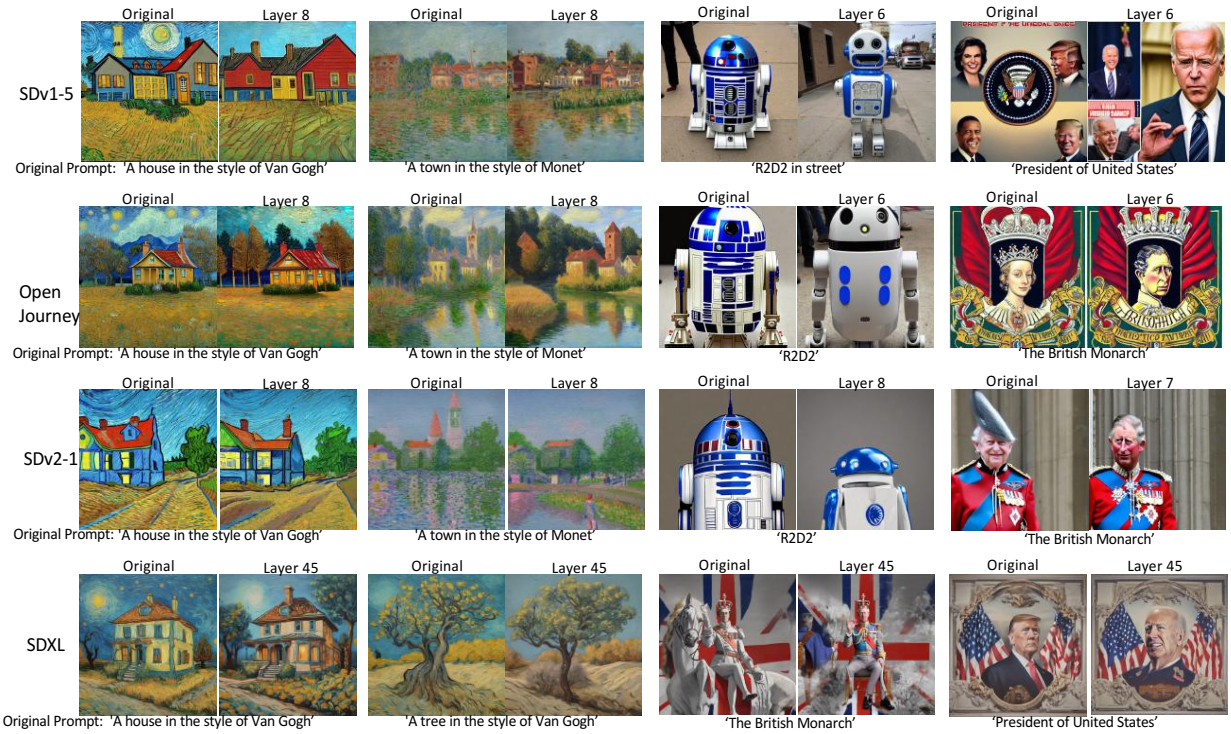


Figure 4. **Interpretability Results: Images generated by intervening on the layers identified by LOCOGEN across various open-source text-to-image models.** We compare the original generation vs. generation by intervening on the layers identified with LOCOGEN along with a target prompt. We find that across various text-to-image models, visual attributes such as *style, objects, facts* can be manipulated by intervening only on a very small fraction of cross-attention layers.

take $\mathbf{c}$ as input. We name these layers as *controlling layers*. We denote by I_{altered} the image generated using this model and Equation (1) with altered inputs when $\mathbf{z}_T$ is given as the initial noise. We denote the model $\epsilon_\theta(\mathbf{z}_t, \mathbf{c}, \mathbf{c}', t)$ with the altered inputs as the `Altered Model` with the following inference procedure:

$$\hat{\epsilon}(\mathbf{z}_t, \mathbf{c}, \mathbf{c}', t) = \epsilon_\theta(\mathbf{z_t}, \mathbf{c}, \mathbf{c}', t) + \alpha(\epsilon_\theta(\mathbf{z_t}, \mathbf{c}, \mathbf{c}', t) - \epsilon_\theta(\mathbf{z_t}, t)) \; .$$

As an example, to find the layers where *style* knowledge corresponding to a particular artist is stored, $\{C_l\}_{l=1}^{M} - C'$ receive text-embeddings corresponding to the prompt *'An <object> in the style of <artist>'*, whereas the layers in C' receive text-embeddings corresponding to the prompt *'An <object> in the style of painting'*. If the generated image with these inputs do not have that particular style, we realize that controlling layers C' are responsible for incorporating that specified style in the output (see Figure 1). In fact, this replacement operation enables finding locations across different cross-attention layers where various visual attribute knowledge is localized.

5.1.2. LOCOGEN ALGORITHM

Our goal is to find controlling layers C' for different visual attributes. We note that the cardinality of the set $|C'| = m$ is a hyper-parameter and the search space for C' is exponential. Given $|C'| = m$, there are $\binom{M}{m}$ possibilities for C', thus, we restrict our search space to only adjacent cross-attention layers. In fact, we consider all C' such that $C' = \{C_l\}_{l=j}^{j+m-1}$ for $j \in [1, M - m + 1]$.

Selecting the hyper-parameter m. To select the cardinality of the set C', we run an iterative hyper-parameter search with $m \in [1, M]$, where M is selected based on the maximum number of cross-attention layers in a given text-to-image generative model. At each iteration of the hyper-parameter search, we investigate whether there exists a set of m adjacent cross-attention layers that are responsible for the generation of the specific visual attribute. We find minimum m that such controlling layers for the particular attribute exists.

To apply LOCOGEN for a particular attribute, we obtain a set of input prompts $T = \{T_i\}_{i=1}^{N}$ that include the particular attribute and corresponding set of prompts $T' = \{T_i'\}_{i=1}^{N}$ where T_i' is analogous to T_i except that the particular attribute is removed/updated. These prompts serve to create altered images and assess the presence of the specified attribute within them. Let $\mathbf{c}_i$ be the text-embedding of T_i and $\mathbf{c}_i'$ be that of T_i'.

Given m, we examine all $M - m + 1$ possible candidates for controlling layers. For each of them, we generate N

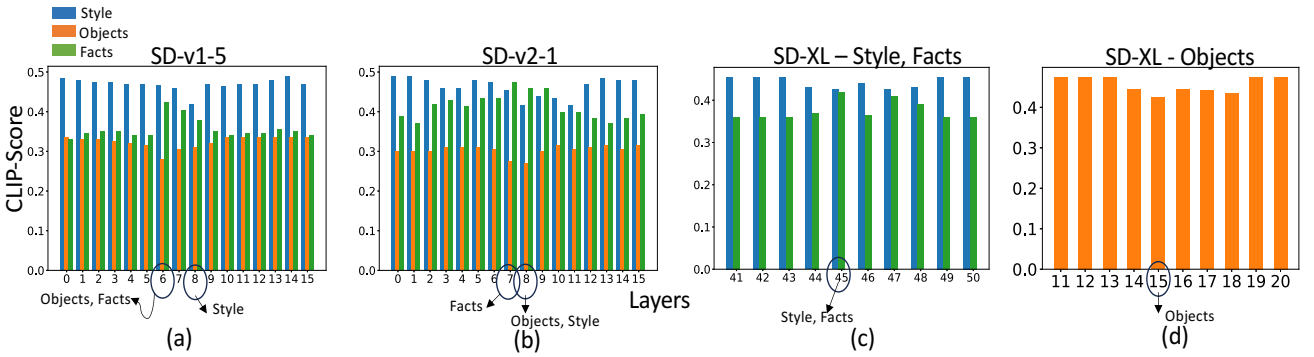


Figure 5. **CLIP-Score of the generated images with original prompt for** *style, objects* **and target prompt for** *facts* **after intervening on layers through LOCOGEN.** Lower CLIP-Score for *objects, style* indicate correct localization, whereas a higher CLIP-Score indicates such for *facts*. (a) For SD-v1-5 (m=2), *objects, facts* can be controlled from Layer 6, whereas *style* can be controlled from Layer 8. (b) For SD-v2-1(m=3), *facts* are controlled from Layer 7, *style* and *objects* from Layer 8. (c,d): For SD-XL, *style (m=3), facts(m=5)* are controlled from Layer 45, whereas *objects* are controlled from Layer 15.

altered images where i-th image is generated by giving $\mathbf{c}'_i$ as the input embedding to selected m layers and $\mathbf{c}_i$ to other ones. Then we measure the CLIP-Score (Hessel et al., 2021) of original text prompt T_i to the generated image for *style, objects* and target text prompt T'_i to the generated image for *facts*. For *style* and *objects*, drop in CLIP-Score shows the removal of the attribute while for *facts* increase in score shows similarity to the updated fact. We take the average of the mentioned score across all $1 \leq i \leq N$. By doing that for all candidates, we report the one with minimum average CLIP-Score for *style, objects* and maximum average CLIP-Score for *facts*. These layers could be candidate layers controlling the generation of the specific attribute. Algorithm 1 provides the pseudocode to find the best candidate. Figure 5 shows CLIP-Score across different candidates.

Algorithm 1 LOCOGEN

Input: $m, \{T_i\}_{i=1}^N, \{T'_i\}_{i=1}^N, \{\mathbf{c}_i\}_{i=1}^N, \{\mathbf{c}'_i\}_{i=1}^N$
Output: Candidate controlling set
  for $j \leftarrow 1, \ldots, M-m$ **do**
    $C' \leftarrow \{C_l\}_{l=j}^{j+m-1}$
    for $i \leftarrow 1, \ldots, N$ **do**
      $s_i \leftarrow \text{CLIP-SCORE}\,(T_i, I_{\text{altered}})$
      $s'_i \leftarrow \text{CLIP-SCORE}\,(T'_i, I_{\text{altered}})$
    $a_j \leftarrow \text{AVERAGE}\left(\{s_i\}_{i=1}^N\right)$ ▷ for objects, style
    $a_j \leftarrow \text{AVERAGE}\left(\{s'_i\}_{i=1}^N\right)$ ▷ for facts
  $j^* \leftarrow \arg\min_j a_j$ ▷ for objects, style
  $j^* \leftarrow \arg\max_j a_j$ ▷ for facts
  return $a_{j^*}, \{C_l\}_{l=j^*}^{j^*+m-1}$

We set a threshold for average CLIP-Score and find the minimum m such that there exists m adjacent cross-attention layers whose corresponding CLIP-Score meets the requirement. We point the reader to Appendix G for the values of m selected for different models and thresholds.

Dataset for Prompts. We use the prompts used in (Basu et al., 2023; Kumari et al., 2023) to extract locations in the UNet which control for various visual attributes such as *objects, style* and *facts*. More details in Appendix C.

5.2. Empirical Results

In this section, we provide empirical results highlighting the localized layers across various open-source text-to-image generative models:

Stable-Diffusion Variants. Across both models, as depicted qualitatively in Figure 4 and quantitatively in Figure 5-(a), we observe the presence of a distinctive subset of layers that govern specific visual attributes. In the case of both SD-v1-5 and SD-v2-1, the control for "style" is centralized at $l = 8$ with $m = 2$. In SD-v1-5, the control for "objects" and "facts" emanates from the same locations: $l = 6$ and $m = 2$. However, in SD-v2-1, "objects" are controlled from $l = 8$, while "facts" are influenced by $l = 7$. Despite sharing a similar UNet architecture and undergoing training with comparable scales of pre-training data, these models diverge in the text-encoder utilized. This discrepancy in text-encoder choice may contribute to the variation in how they store knowledge concerning different attributes.

Open-Journey. We note that Open-Journey exhibits control locations similar to SD-v1-5 for various visual attributes. As illustrated in Figure 4 and Figure 5-(a), "objects" and "facts" are governed from $l = 6$, while "style" is controlled from $l = 8$. Despite the architectural resemblance between Open-Journey and SD-v1-5, it's important to highlight that Open-Journey undergoes fine-tuning on a subset of images generated from Mid-Journey. This suggests that the control locations for visual attributes are more closely tied to the underlying model architecture than to the specifics of the training or fine-tuning data.

SD-XL. Within SD-XL, our investigation reveals that both

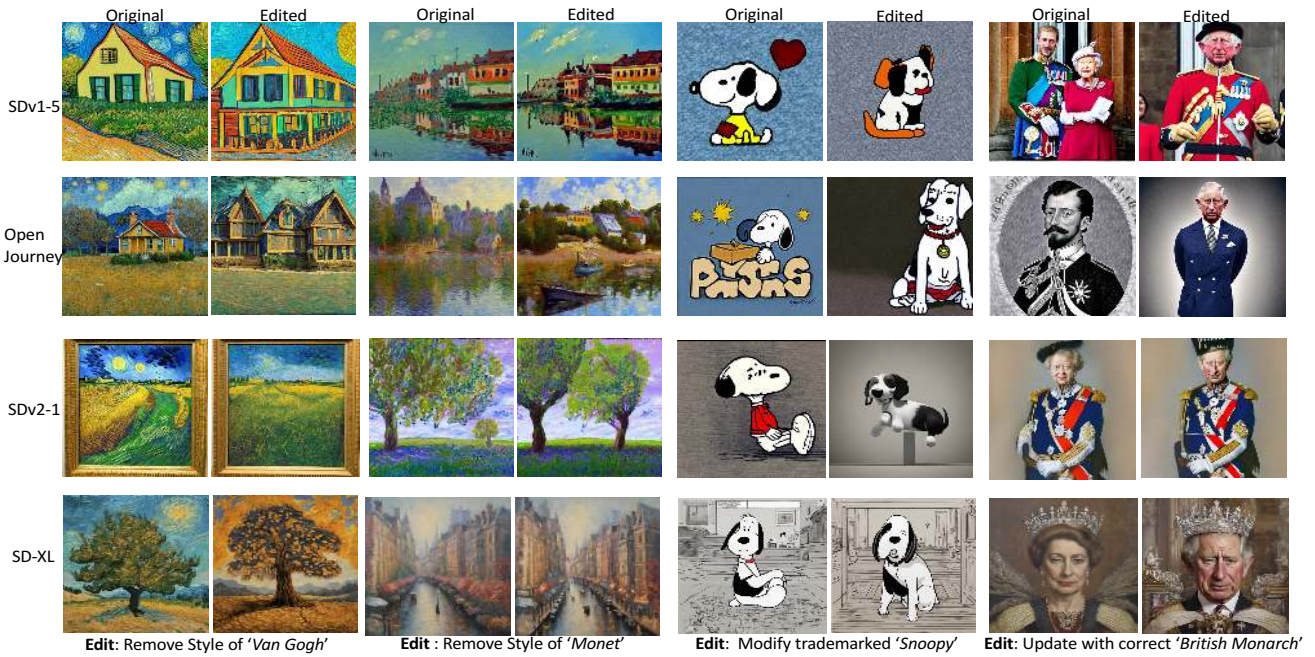


Figure 6. **LOCOEDIT (Model editing) results at locations identified by LOCOGEN across various open-source text-to-image models.** We observe that locations identified by our interpretability framework can be edited effectively to remove *styles*, *objects* and update *facts* in text-to-image models. We provide more visualizations in Appendix B.

Figure 7. **Interpretability Results for DeepFloyd.** We find the control points for visual attributes to be dependent on the underlying prompts, rather than the visual attribute.

"style" and "facts" can be effectively controlled from $l = 45$, with $m = 3$ as evidenced in Figure 4 and Figure 5-(c). For the attribute "objects," control is situated at $l = 15$, albeit with a slightly larger value of $m = 5$. In summary, SD-XL, consisting of a total of 70 cross-attention layers, underscores a significant finding: **various attributes** in image generation can be governed by only **a small subset of layers**.

DeepFloyd. Across SD-v1-5, SD-v2-1, Open-Journey, and SD-XL, our findings indicate that visual attributes like "style", "objects" and "facts," irrespective of the specific prompt used, can be traced back to control points situated within a limited number of layers. However, in the case of DeepFloyd, our observations differ. We find instead, that all attributes display localization dependent on the specific prompt employed. To illustrate, factual knowledge related to "The British Monarch" is governed from $l = 6$ with $m = 3$, whereas factual knowledge tied to "The President of the United States" is controlled from $l = 12$ (see Figure 7). This divergence in localization patterns highlights the nuanced behavior of DeepFloyd in comparison to the other models examined. More results can be referred in Appendix B.5.

Human-Study Results. We run a human-study to verify that LOCOGEN can effectively identify controlling layers for different visual attributes. In our setup, evaluators assess 132 image pairs, each comprising an image generated by `Clean Model` and an image generated by `Altered Model` whose identified cross-attention layers takes different inputs. Evaluators determine whether the visual attribute is changed in the image generated by `Altered Model`(for instance, the artistic Van Gogh style is removed from the original image or not). Covering 33 image pairs, generated with different prompts per model, with five participating evaluators, our experiments reveal a 92.58% verification rate for the impact of LOCOGEN-identified layers on visual attributes. See more details in Appendix J.

6. LOCOEDIT: Editing to Ablate Concepts

In this section, we analyse the effectiveness of edits in the layers identified by LOCOGEN across text-to-image models.

6.1. Method

Algorithm 1 extracts the exact set of cross-attention layers from which the knowledge about a particular visual attribute

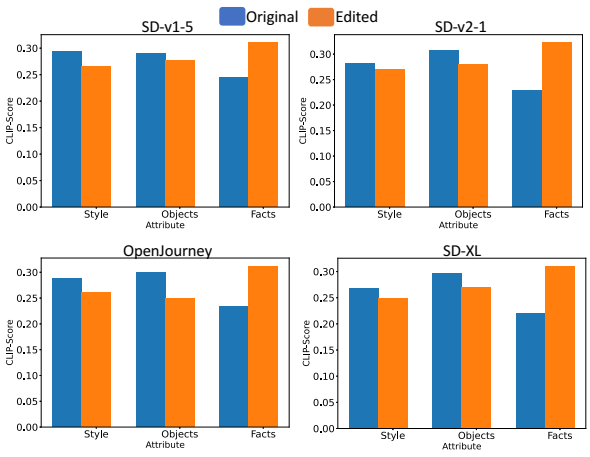


Figure 8. **Quantitative Model Editing Results for Text-to-Image Models.** We observe a drop in CLIP-Score for "style" and "objects", while an increase in CLIP-Score for "facts" therefore highlighting correct edits.

(e.g., *style*) is controlled. We denote this set as C_{loc}, where $C_{\text{loc}} \subset C$ and $|C_{\text{loc}}| = m$. This set of extracted cross-attention layers C_{loc}, each containing value and key matrices is denoted as $C_{\text{loc}} = \{\hat{W}_l^K, \hat{W}_l^V\}_{l=1}^m$. The objective is to modify these weight matrices $\{\hat{W}_l^K, \hat{W}_l^V\}_{l=1}^m$ such that they transform the original prompt (e.g., '*A house in the style of Van Gogh*') to a target prompt (e.g., '*A house in the style of a painting*') in a way that the visual attribute in the generation is modified. Similar to Section 5.1.2, we use a set of input prompts $T_{\text{orig}} = \{T_i^o\}_{i=1}^N$ consisting of prompts featuring the particular visual attribute. Simultaneously, we create a counterpart set $T_{\text{target}} = \{T_i^t\}_{i=1}^N$ where each T_i^t is identical to T_i^o but lacks the particular attribute in focus. Let $\mathbf{c}_i^o \in \mathbb{R}^d$ be the text-embedding of the last subject token in T_i^o and $\mathbf{c}_i^t \in \mathbb{R}^d$ be that of T_i^t. We obtain matrix $\mathbf{X}_{\text{orig}} \in \mathbb{R}^{N \times d}$ by stacking vectors $\mathbf{c}_1^o, \mathbf{c}_2^o, \ldots, \mathbf{c}_N^o$ and matrix $\mathbf{X}_{\text{target}} \in \mathbb{R}^{N \times d}$ by stacking $\mathbf{c}_1^t, \mathbf{c}_2^t, \ldots, \mathbf{c}_N^t$. To learn a mapping between the key and the value embeddings, we solve the following optimization for each layer $l \in [1, m]$ corresponding to the key matrices as:

$$\min_{W_l^K} \|\mathbf{X}_{\text{orig}} W_l^K - \mathbf{X}_{\text{target}} \hat{W}_l^K\|_2^2 + \lambda_K \|W_l^K - \hat{W}_l^K\|_2^2$$

where λ_K is the regularizer. Letting $\mathbf{Y}_{\text{orig}} = \mathbf{X}_{\text{orig}} W_l^K$ the optimal closed form solution for the key matrix is:

$$W_l^K = (\mathbf{X}_{\text{orig}}^T \mathbf{X}_{\text{orig}} + \lambda_1 I)^{-1} (\mathbf{X}_{\text{orig}}^T \mathbf{Y}_{\text{target}} + \lambda_K \hat{W}_l^K)$$

Same is applied to get optimal matrix for value embeddings.

6.2. Model Editing Results

Stable-Diffusion Variants, Open-Journey and SD-XL. In Figure 6 and Figure 8, it becomes apparent that LOCOEDIT effectively integrates accurate edits into the locations identified by LOCOGEN. Qualitatively examining the visual edits in Figure 6, our method demonstrates the capability to remove artistic "styles", modify trademarked "objects," and update outdated "facts" within a text-to-image model with accurate information. This visual assessment is complemented by the quantitative analysis in Figure 8, where we observe that the CLIP-Score of images generated by the edited model, given prompts containing specific visual attributes, consistently registers lower than that of the clean model for "objects" and "style." For "facts," we gauge the CLIP-Score of images from the model with the correct facts, wherein a higher CLIP-Score indicates a correct edit, as illustrated in Figure 8. Combining both qualitative and quantitative findings, these results collectively underscore the effectiveness of LOCOEDIT across SD-v1-5, SD-v2-1, Open-Journey, and SD-XL. However, it's noteworthy that the efficacy of closed-form edits varies among different text-to-image models. Specifically, in the case of "style," we observe the most substantial drop in CLIP-Score between the edited and unedited models for SD-v1-5 and Open-Journey, while the drop is comparatively less for SD-v2-1 and SD-XL. Conversely, for "facts," we find that all models perform similarly in updating with new information.

Limitations with DeepFloyd Closed-Form Edits. DeepFloyd, despite revealing distinct locations through LOCOGEN (albeit depending on the underlying prompt), exhibits challenges in effective closed-form edits at these locations. Appendix M provides qualitative visualizations illustrating this limitation. The model employs a T5-encoder with bi-directional attention, diverging from other text-to-image models using CLIP-variants with causal attention. Closed-form edits, relying on mapping the last-subject token embedding to a target embedding, are typically effective in text-embeddings generated with causal attention, where the last-subject token holds crucial information. However, the T5-encoder presents a hurdle as tokens beyond the last subject token contribute essential information about the target attribute. Consequently, restricting the mapping to the last-subject token alone proves ineffective for a T5-encoder.

While LOCOGEN along with LOCOEDIT makes model editing more interpretable – we also find that localized-model editing is better than updating all layers in the UNet as shown in Appendix F. We also compare our method with existing editing methods (Basu et al., 2023; Kumari et al., 2023; Gandikota et al., 2023) in Appendix I. We find that our editing method is at par with existing baselines, with the added advantage of generalizability to models beyond Stable-Diffusion-v1-5. In Appendix L, we also show the robustness of our method to generic prompts.

7. On Neuron-Level Model Editing

In this section, we explore the feasibility of effecting neuron-level modifications to eliminate stylistic attributes from the

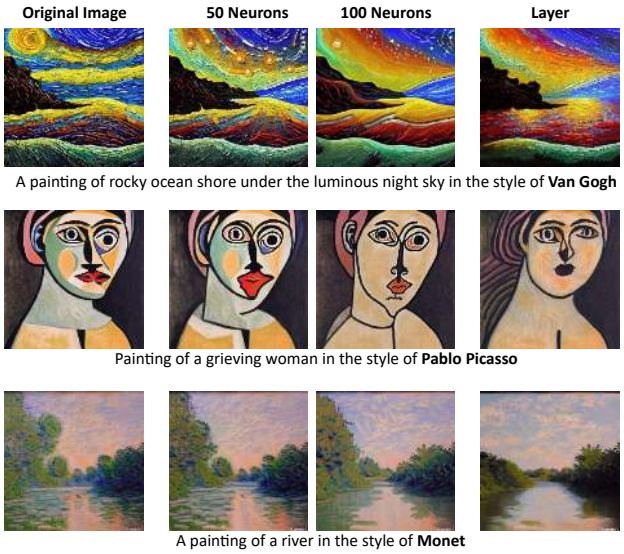


Figure 9. **Neuron-Level Model Editing - Qualitative**. Results when applying **neuron-level dropout** on identified neurons in layers specified with LOCOGEN on Stable Diffusion v1.5. The second and third columns display images with 50 and 100 modified neurons out of 1280 in controlling layers. The last column shows images with a different embedding in controlling layers.

output of text-to-image models. According to layers identified with LOCOGEN, our objective is to ascertain whether the selective dropout of neurons at the activation layers within the specified cross-attention layers (key and value embeddings) can successfully eliminate stylistic elements.

To accomplish this objective, we first need to identify which neurons are responsible for the generation of particular artistic styles, e.g., *Van Gogh*. We examine the activations of neurons in the embedding space of key and value matrices in identified cross-attention layers. More specifically, we pinpoint neurons that exhibit significant variations when comparing input prompts that include a particular style with the case that input prompts do not involve the specified style.

To execute this process, we collect a set of N_1 prompts that feature the specific style, e.g. Van Gogh. We gather text-embeddings of the last subject token of these prompts denoted by $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_{N_1}$, where $\mathbf{c}_i \in \mathbb{R}^d$. We also obtain a set of N_2 prompts without any particular style and analogously obtain $\{\mathbf{c}'_1, \mathbf{c}'_2, \ldots, \mathbf{c}'_{N_2}\}$, where $\mathbf{c}'_i \in \mathbb{R}^d$. Next, for the key or value matrix $W \in \mathbb{R}^{d \times d'}$, we consider key or value embedding of these input prompts, i.e., $\{z_i\}_{i=1}^{N_1} \cup \{z'_i\}_{i=1}^{N_2}$ where $z_i = \mathbf{c}_i W$ and $z'_i = \mathbf{c}'_i W$. We note that $z_i, z'_i \in \mathbb{R}^{d'}$.

Subsequently, for each of these d' neurons, we assess the statistical difference in their activations between input prompts that include a particular style and those without it. Specifically, we compute the z-score for each neuron within two groups of activations: $z_1, z_2, \ldots, z_{N_1}$ and $z'_1, z'_2, \ldots, z'_{N_2}$. The neurons are then ranked based on the absolute value of their z-score, with the top neurons representing those that exhibit significant differences in activations depending on the presence or absence of a particular concept in the input prompt. During generation, we drop-out these neurons and see if particular style is removed or not. As seen in Figure 9, neuron-level modification at inference time is effective at removing styles. This shows that knowledge about a particular style can be even more localized to a few neurons. It is noteworthy that the extent of style removal increases with the modification of more neurons, albeit with a trade-off in the quality of generated images. This arises because modified neurons may encapsulate information related to other visual attributes. To quantify the effectiveness of this approach, we measure the drop in CLIP-Score for modified images across various styles. Figure 10 presents a bar-plot illustrating these similarity scores. Notably, drop in CLIP-Score demonstrates that neuron-level model editing effectively removes the styles associated with different artists in the generated images. We refer to Appendix K.1 for more details on neuron-level model editing experiments.

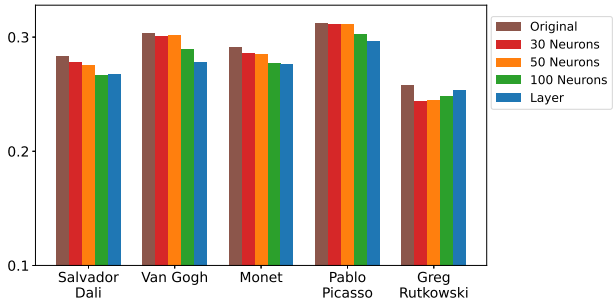


Figure 10. **Neuron-Level Model Editing - Quantitative**. Average CLIP-Score of generated images to text prompt *'style of <artist>'*. Brown bars show similarity to original generated image; red, orange, and green bars show similarity to generated image when 30, 50, and 100 neurons are modified, respectively; and blue bars refer to images when controlling layers receive other prompt.

8. Conclusion

In our paper, we comprehensively examine knowledge localization across various open-source text-to-image models. We initially observe that while causal tracing proves effective for early Stable-Diffusion variants, its generalizability diminishes when applied to newer text-to-image models like DeepFloyd and SD-XL for localizing control points associated with visual attributes. To address this limitation, we introduce LOCOGEN, capable of effectively identifying locations within the UNet across diverse text-to-image models. Harnessing these identified locations within the UNet, we evaluate the efficacy of closed-form model editing across a range of text-to-image models leveraging LOCOEDIT, uncovering intriguing properties. Notably, for specific visual attributes such as "style", we discover that knowledge can even be traced to a small subset of neurons and subsequently edited by applying a simple dropout layer, thereby underscoring the possibilities of neuron-level model editing.

Impact Statement

This paper presents work to advance the understanding of the inner workings of open-source text-to-image generative models. Our interpretability method can advance the understanding of how knowledge is represented in generative models and does not have any potential negative implications on the society. Our editing method can address societal concerns (e.g., an artist asking the model owner to delete their style) in an effective way and to the best of our knowledge does not have any negative societal consequences.

Acknowledgements

This project was supported in part by a grant from an NSF CAREER AWARD 1942230, ONR YIP award N00014-22-1-2271, ARO's Early Career Program Award 310902-00001, HR00112090132 (DARPA/ RED), HR001119S0026 (DARPA/ GARD), Army Grant No. W911NF2120076, the NSF award CCF2212458, NSF Award No. 2229885 (NSF Institute for Trustworthy AI in Law and Society, TRAILS), an Amazon Research Award and an award from Capital One.

References

Basu, S., Zhao, N., Morariu, V., Feizi, S., and Manjunatha, V. Localizing and editing knowledge in text-to-image generative models, 2023.

Chefer, H., Lang, O., Geva, M., Polosukhin, V., Shocher, A., Irani, M., Mosseri, I., and Wolf, L. The hidden language of diffusion models. *arXiv preprint arXiv:2306.00966*, 2023.

Gandikota, R., Materzyńska, J., Fiotto-Kaufman, J., and Bau, D. Erasing concepts from diffusion models. In *Proceedings of the 2023 IEEE International Conference on Computer Vision*, 2023.

Hertz, A., Mokady, R., Tenenbaum, J., Aberman, K., Pritch, Y., and Cohen-Or, D. Prompt-to-prompt image editing with cross attention control, 2022.

Hessel, J., Holtzman, A., Forbes, M., Bras, R. L., and Choi, Y. Clipscore: A reference-free evaluation metric for image captioning. *arXiv preprint arXiv:2104.08718*, 2021.

Ho, J. and Salimans, T. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. URL https://openreview.net/forum?id=qw8AKxfYbI.

Kumari, N., Zhang, B., Wang, S.-Y., Shechtman, E., Zhang, R., and Zhu, J.-Y. Ablating concepts in text-to-image diffusion models. In *'International Conference on Computer Vision (ICCV)'*, 2023.

Lin, T., Maire, M., Belongie, S. J., Bourdev, L. D., Girshick, R. B., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. URL http://arxiv.org/abs/1405.0312.

Orgad, H., Kawar, B., and Belinkov, Y. Editing implicit assumptions in text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

Pearl, J. Direct and indirect effects. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, UAI'01, pp. 411–420, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1558608001.

Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., and Rombach, R. Sdxl: Improving latent diffusion models for high-resolution image synthesis, 2023.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. Zero-shot text-to-image generation. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8821–8831. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/ramesh21a.html.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. *CoRR*, abs/2112.10752, 2021. URL https://arxiv.org/abs/2112.10752.

Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E. L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., Ho, J., Fleet, D. J., and Norouzi, M. Photorealistic text-to-image diffusion models with deep language understanding. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 36479–36494. Curran Associates, Inc., 2022.

Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C. W., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., Schramowski, P., Kundurthy, S. R., Crowson, K., Schmidt, L., Kaczmarczyk, R., and Jitsev, J. LAION-5b: An open large-scale dataset for training next generation image-text models. In *Thirty-sixth*

Conference on Neural Information Processing Systems Datasets and Benchmarks Track, 2022. URL https://openreview.net/forum?id=M3Y74vmsMcY.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. *CoRR*, abs/2010.02502, 2020. URL https://arxiv.org/abs/2010.02502.

Tang, R., Liu, L., Pandey, A., Jiang, Z., Yang, G., Kumar, K., Stenetorp, P., Lin, J., and Ture, F. What the DAAM: Interpreting stable diffusion using cross attention. In Rogers, A., Boyd-Graber, J., and Okazaki, N. (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5644–5659, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.310. URL https://aclanthology.org/2023.acl-long.310.

van den Oord, A., Vinyals, O., and kavukcuoglu, k. Neural discrete representation learning. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/7a98af17e63a0ac09ce2e96d03992fbc-Paper.pdf.

A. Visualizations with Causal Tracing

A.1. SD-v2-1

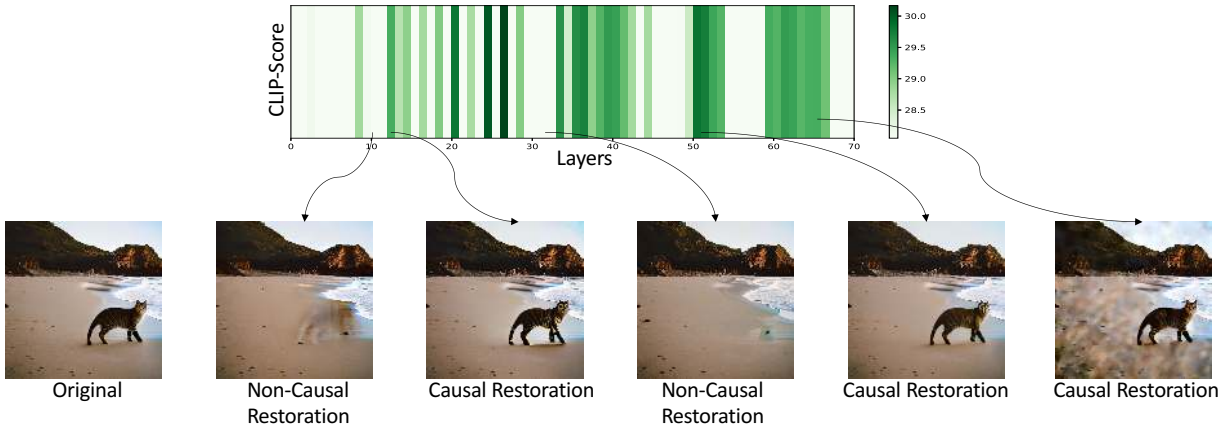


Figure 11. **Causal Tracing in the UNet for SD-v2-1** Similar to earlier works (Basu et al., 2023), we find that knowledge about the attribute "objects" is distributed in the UNet amongst various layers. Given that these layers can independently restore a corrupted model to a clean model, model editing requires editing all these causal layers. Moreover these layers cannot be updated in closed-form updates due to the presence of non-linearities in the layer components.

A.2. SD-XL

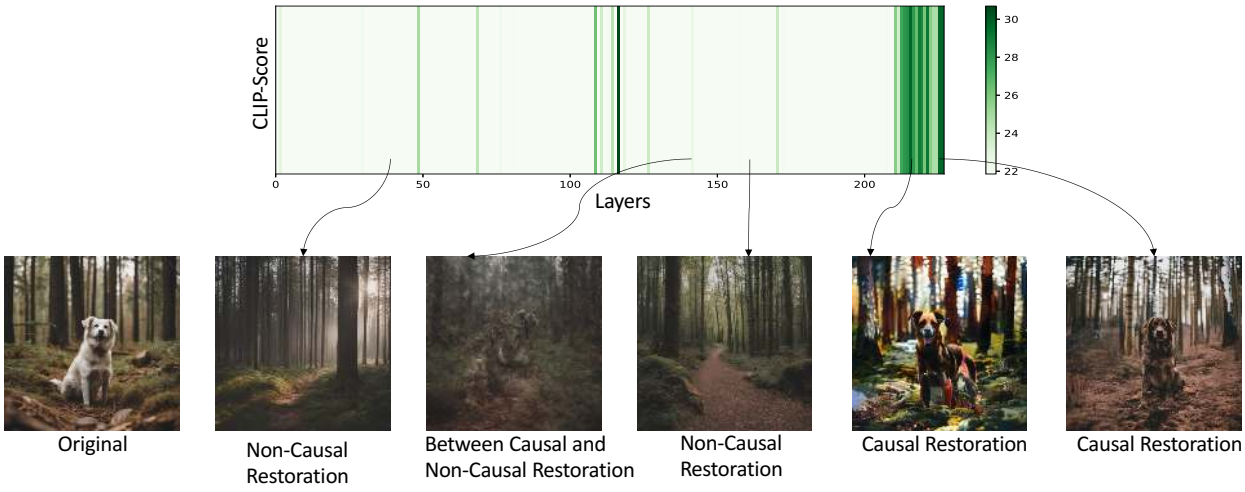


Figure 12. **Causal Tracing in the UNet for SD-XL**. We find that knowledge about the attribute "objects" is distributed amongst the various layers in the UNet. However, compared to other models such as SD-v2-1, the distribution is slightly sparse.

A.3. DeepFloyd

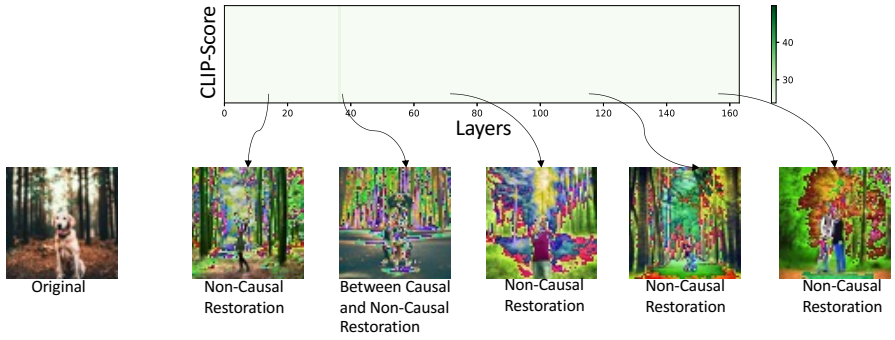


Figure 13. **DeepFloyd**. In the case of the DeepFloyd model, we find that there is no presence of strong causal state which can restore a corrupted model to its clean state. This absence of causal states rules out the possibilities of model editing in the UNet.

B. Visualizations with CrossPrompt

B.1. SD-v1-5

Figure 14. **Layer 8 can control "style" in SD-v1-5**. We perform an intervention in the different cross-attention layers of Stable-Diffusion-v2-1 by using a target prompt - *'a painting'* in those layers while the original prompt is used for other layers. We find that there exists an unique layer which can control output style.

Figure 15. **Layer 8 can control "style" in SD-v1-5**. We perform an intervention in the different cross-attention layers of Stable-Diffusion-v2-1 by using a target prompt - *'a painting'* in those layers while the original prompt is used for other layers. We find that there exists an unique layer which can control output style.

Figure 16. **Layer 8 can control "style" in SD-v1-5**. We perform an intervention in the different cross-attention layers of Stable-Diffusion-v2-1 by using a target prompt - *'a painting'* in those layers while the original prompt is used for other layers. We find that there exists an unique layer which can control output style.

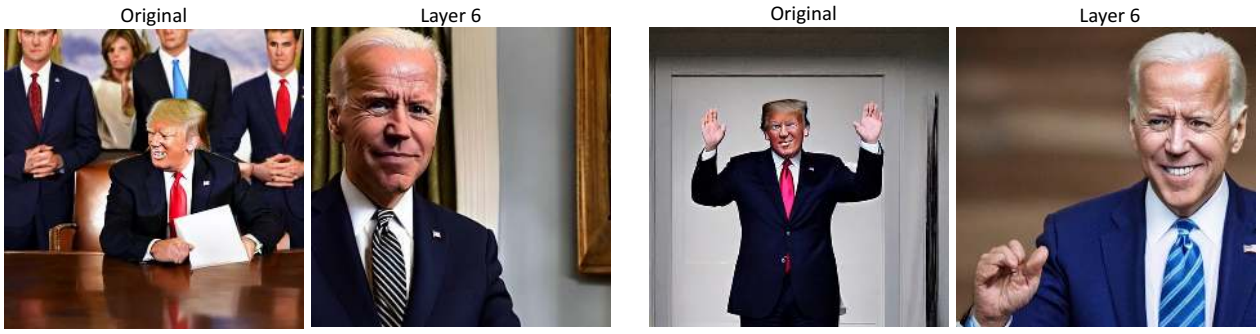


Figure 17. **Layer 6 can control "factual knowledge" in SD-v1-5**. We perform an intervention in the different cross-attention layers of Stable-Diffusion-v2-1 by using a target prompt - *'Joe Biden'* in those layers while the original prompt is used for other layers. We find that there exists an unique layer which can control output generations of "factual knowledge".

Figure 18. **Layer 6 can control "factual knowledge" in SD-v1-5**. We perform an intervention in the different cross-attention layers of Stable-Diffusion-v2-1 by using a target prompt - *'Prince Charles'* in those layers while the original prompt is used for other layers. We find that there exists an unique layer which can control output generations of "factual knowledge".

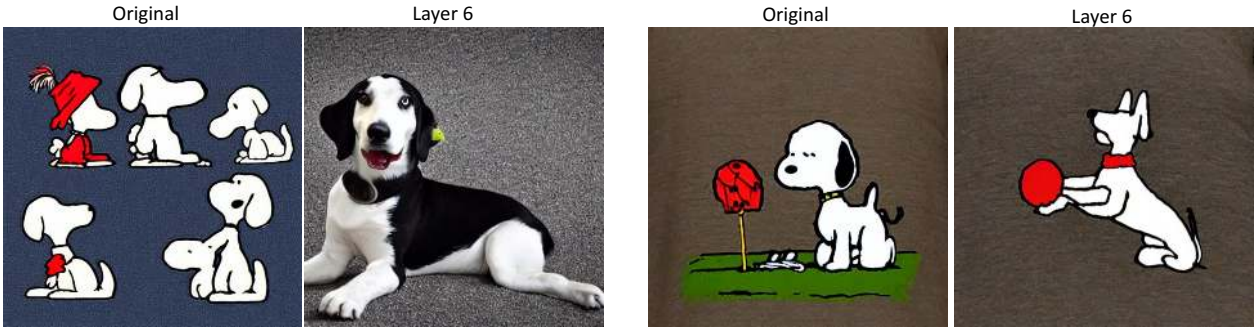


Figure 19. **Layer 6 can control "object knowledge" in SD-v1-5**. We perform an intervention in the different cross-attention layers of Stable-Diffusion-v2-1 by using a target prompt - *'a dog'* in those layers while the original prompt is used for other layers. We find that there exists an unique layer which can control output generations of "object knowledge".

B.2. SD-v2-1

Figure 20. **Layer 8 can control "style" in SD-v2-1**. We perform an intervention in the different cross-attention layers of Stable-Diffusion-v2-1 by using a target prompt - *'a painting'* in those layers while the original prompt is used for other layers. We find that there exists an unique layer which can control output style.

Figure 21. **Layer 8 can control "style" in SD-v2-1**. We perform an intervention in the different cross-attention layers of Stable-Diffusion-v2-1 by using a target prompt - *'a painting'* in those layers while the original prompt is used for other layers. We find that there exists an unique layer which can control output style.

Figure 22. **Layer 8 can control "style" in SD-v2-1**. We perform an intervention in the different cross-attention layers of Stable-Diffusion-v2-1 by using a target prompt - *'a painting'* in those layers while the original prompt is used for other layers. We find that there exists an unique layer which can control output style.

Figure 23. **Layer 7 can control "factual knowledge" in SD-v2-1**. We perform an intervention in the different cross-attention layers of Stable-Diffusion-v2-1 by using a target prompt - *'Joe Biden'* in those layers while the original prompt is used for other layers. We find that there exists an unique layer which can control output generations of "factual knowledge".

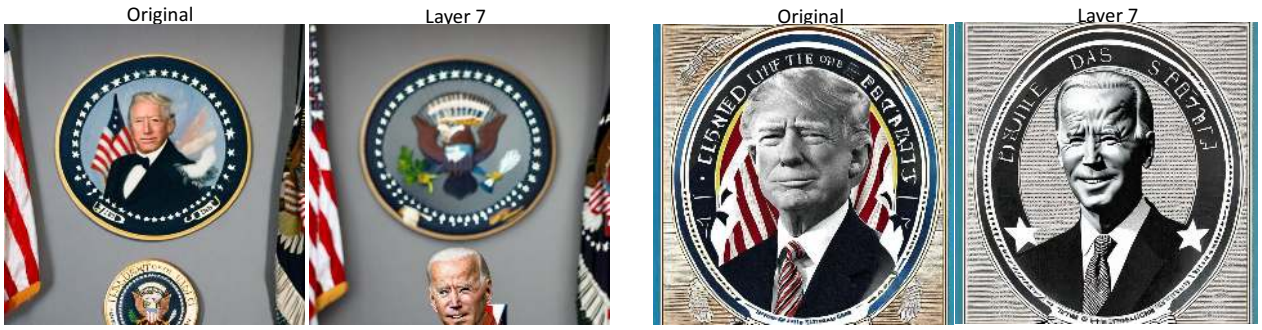


Figure 24. **Layer 7 can control "factual knowledge" in SD-v2-1**. We perform an intervention in the different cross-attention layers of Stable-Diffusion-v2-1 by using a target prompt - *'Prince Charles'* in those layers while the original prompt is used for other layers. We find that there exists an unique layer which can control output generations of "factual knowledge".

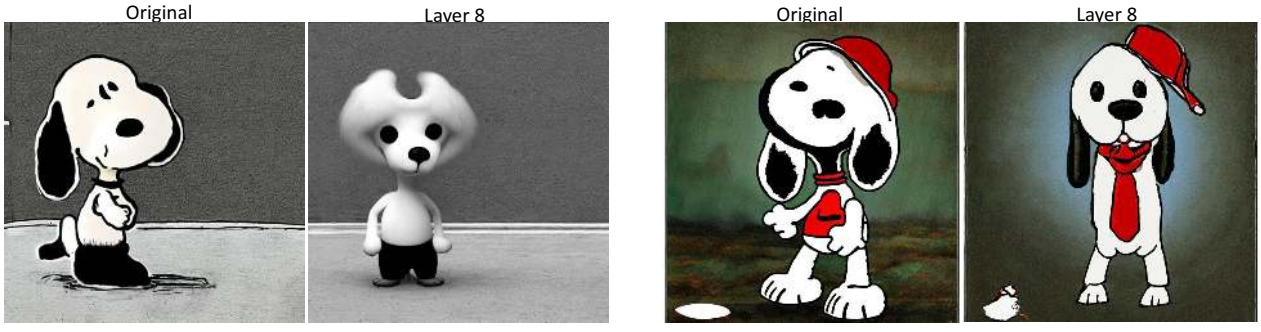


Figure 25. **Layer 8 can control "object knowledge" in SD-v2-1**. We perform an intervention in the different cross-attention layers of Stable-Diffusion-v2-1 by using a target prompt - *'a dog'* in those layers while the original prompt is used for other layers. We find that there exists an unique layer which can control output generations of "object knowledge".

B.3. OpenJourney

Figure 26. **Layer 8 can control "style knowledge" in Open-Journey**. We perform an intervention in the different cross-attention layers of Open-Journey by using a target prompt - *'a painting'* in those layers while the original prompt is used for other layers. We find that there exists an unique layer which can control output generations of "style knowledge".

Figure 27. **Layer 8 can control "style knowledge" in Open-Journey**. We perform an intervention in the different cross-attention layers of Open-Journey by using a target prompt - *'a painting'* in those layers while the original prompt is used for other layers. We find that there exists an unique layer which can control output generations of "style knowledge".

Figure 28. **Layer 8 can control "style knowledge" in Open-Journey**. We perform an intervention in the different cross-attention layers of Open-Journey by using a target prompt - *'a painting'* in those layers while the original prompt is used for other layers. We find that there exists an unique layer which can control output generations of "style knowledge".

Figure 29. **Layer 6 can control "factual knowledge" in Open-Journey**. We perform an intervention in the different cross-attention layers of Open-Journey by using a target prompt - *'Joe Biden'* in those layers while the original prompt is used for other layers. We find that there exists an unique layer which can control output generations of "factual knowledge".

Figure 30. **Layer 6 can control "factual knowledge" in Open-Journey**. We perform an intervention in the different cross-attention layers of Open-Journey by using a target prompt - *'Prince Charles'* in those layers while the original prompt is used for other layers. We find that there exists an unique layer which can control output generations of "factual knowledge".

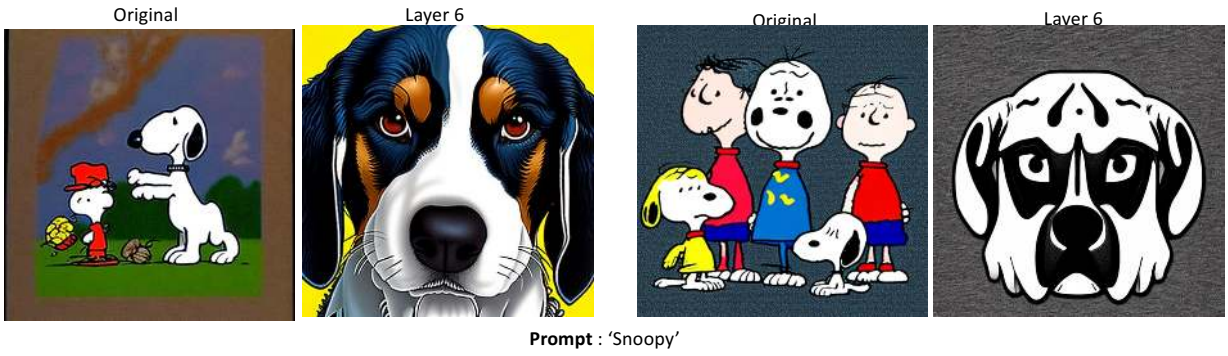


Figure 31. **Layer 6 can control "object knowledge" in Open-Journey**. We perform an intervention in the different cross-attention layers of Open-Journey by using a target prompt - *'a dog'* in those layers while the original prompt is used for other layers. We find that there exists an unique layer which can control output generations of "object knowledge".

B.4. SD-XL

Figure 32. **Layer 45 can control "style knowledge" in Open-Journey**. We perform an intervention in the different cross-attention layers of Open-Journey by using a target prompt - *'a painting'* in those layers while the original prompt is used for other layers. We find that there exists an unique layer which can control output generations of "style knowledge".

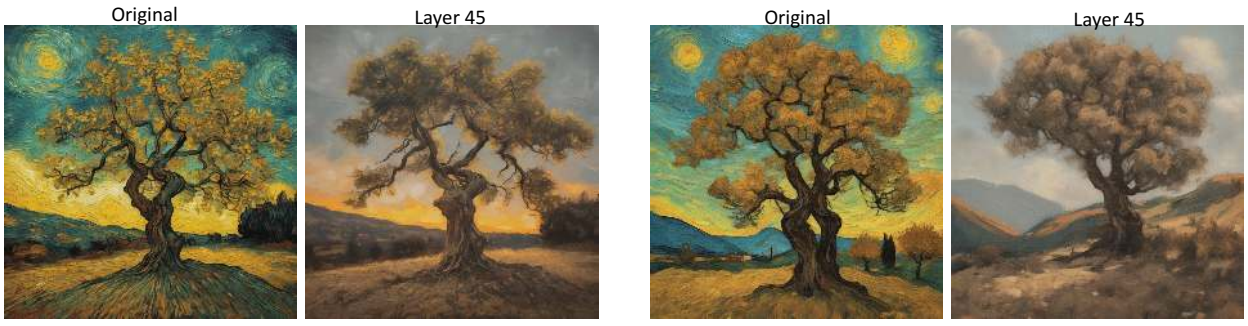


Figure 33. **Layer 45 can control "style knowledge" in Open-Journey**. We perform an intervention in the different cross-attention layers of Open-Journey by using a target prompt - *'a painting'* in those layers while the original prompt is used for other layers. We find that there exists an unique layer which can control output generations of "style knowledge".

Prompt : 'Women working in a field in the style of *Van Gogh*'

Figure 34. **Layer 45 can control "style knowledge" in Open-Journey**. We perform an intervention in the different cross-attention layers of Open-Journey by using a target prompt - *'a painting'* in those layers while the original prompt is used for other layers. We find that there exists an unique layer which can control output generations of "style knowledge".

Prompt : 'Rocks in the ocean in the style of Monet'

Figure 35. **Layer 45 can control "style knowledge" in Open-Journey**. We perform an intervention in the different cross-attention layers of Open-Journey by using a target prompt - *'a painting'* in those layers while the original prompt is used for other layers. We find that there exists an unique layer which can control output generations of "style knowledge".

Figure 36. **Layer 45 can control "style knowledge" in Open-Journey**. We perform an intervention in the different cross-attention layers of Open-Journey by using a target prompt - *'a painting'* in those layers while the original prompt is used for other layers. We find that there exists an unique layer which can control output generations of "style knowledge".

Figure 37. **Layer 45 can control "factual knowledge" in Open-Journey**. We perform an intervention in the different cross-attention layers of Open-Journey by using a target prompt - *'Joe Biden'* in those layers while the original prompt is used for other layers. We find that there exists an unique layer which can control output generations of "factual knowledge".

Figure 38. **Layer 45 can control "factual knowledge" in Open-Journey**. We perform an intervention in the different cross-attention layers of Open-Journey by using a target prompt - *'Prince Charles'* in those layers while the original prompt is used for other layers. We find that there exists an unique layer which can control output generations of "factual knowledge".

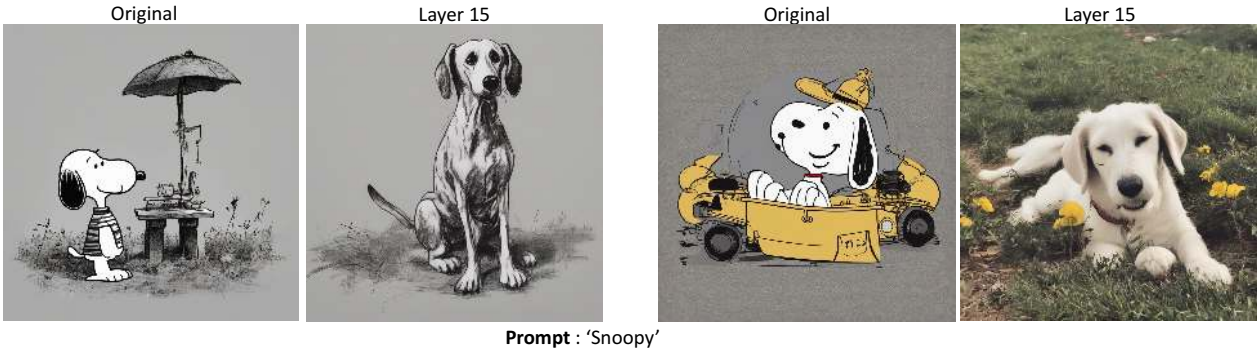


Figure 39. **Layer 15 can control "object knowledge" in Open-Journey**. We perform an intervention in the different cross-attention layers of Open-Journey by using a target prompt - *'a dog'* in those layers while the original prompt is used for other layers. We find that there exists an unique layer which can control output generations of "object knowledge".

B.5. DeepFloyd

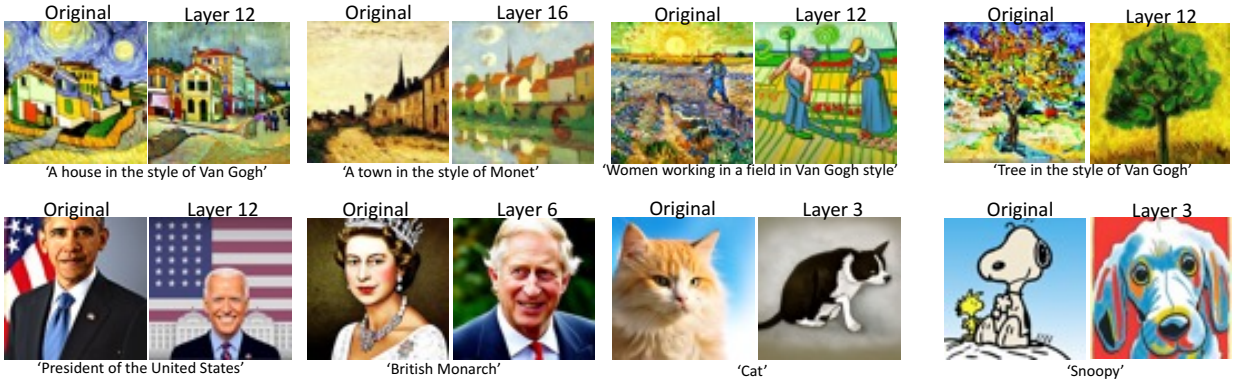


Figure 40. **Knowledge Tracing in Deepfloyd**. We find that control regions for a visual attribute can be different, depending on the prompt. For e.g., prompts involving *Van Gogh* style can be modified from Layer 12, whereas for *Monet* style – modifications are needed from Layer 16. For "facts", prompts involving *The President of the United States* can be controlled from Layer 12, whereas prompts involving *British Monarch* can be controlled from Layer 6.

C. Prompt Dataset

For Interpretability and Model Editing. We use the benchmark dataset from (Basu et al., 2023) and (Kumari et al., 2023) for obtaining prompts for "objects", "style" and "facts". For LOCOGEN, we select the target prompt based on the attribute of interest. For e.g., in the case of "style", we select the target prompt as *'A painting'*. For "facts", we use the correct answer to a given fact, as the target prompt. For e.g., the target prompt for *'The President of the United States'* is *'Joe Biden'* and the target prompt for *'The British Monarch'* is *'Prince Charles'*. For "objects", the target prompt for *'r2d2'* is *'robot'*, for *'snoopy'* is *'dog'*, for *'nemo'* is *'fish'* and for *'cat'* is *'dog'*. These sets of trademarked "objects" and "facts" are chosen from (Basu et al., 2023), whereas the "style" prompts are chosen from (Kumari et al., 2023).

D. Layer Information Across Different Text-to-Image Models

SD-XL

```
• [(0, 'down-blocks.1.attentions.0.transformer-blocks.0.attn2'), (1, 'down-blocks.1.attentions.0.transformer-
  blocks.1.attn2'), (2, 'down-blocks.1.attentions.1.transformer-blocks.0.attn2'), (3, 'down-
  blocks.1.attentions.1.transformer-blocks.1.attn2'), (4, 'down-blocks.2.attentions.0.transformer-blocks.0.attn2'), (5,
  'down-blocks.2.attentions.0.transformer-blocks.1.attn2'), (6, 'down-blocks.2.attentions.0.transformer-blocks.2.attn2'), (7,
  'down-blocks.2.attentions.0.transformer-blocks.3.attn2'), (8, 'down-blocks.2.attentions.0.transformer-blocks.4.attn2'), (9,
  'down-blocks.2.attentions.0.transformer-blocks.5.attn2'), (10, 'down-blocks.2.attentions.0.transformer-blocks.6.attn2'),
  (11, 'down-blocks.2.attentions.0.transformer-blocks.7.attn2'), (12, 'down-blocks.2.attentions.0.transformer-
  blocks.8.attn2'), (13, 'down-blocks.2.attentions.0.transformer-blocks.9.attn2'), (14, 'down-
  blocks.2.attentions.1.transformer-blocks.0.attn2'), (15, 'down-blocks.2.attentions.1.transformer-blocks.1.attn2'), (16,
  'down-blocks.2.attentions.1.transformer-blocks.2.attn2'), (17, 'down-blocks.2.attentions.1.transformer-blocks.3.attn2'),
  (18, 'down-blocks.2.attentions.1.transformer-blocks.4.attn2'), (19, 'down-blocks.2.attentions.1.transformer-
  blocks.5.attn2'), (20, 'down-blocks.2.attentions.1.transformer-blocks.6.attn2'), (21, 'down-
  blocks.2.attentions.1.transformer-blocks.7.attn2'), (22, 'down-blocks.2.attentions.1.transformer-blocks.8.attn2'), (23,
  'down-blocks.2.attentions.1.transformer-blocks.9.attn2'), (24, 'mid-block.attentions.0.transformer-blocks.0.attn2'), (25,
  'mid-block.attentions.0.transformer-blocks.1.attn2'), (26, 'mid-block.attentions.0.transformer-blocks.2.attn2'), (27, 'mid-
  block.attentions.0.transformer-blocks.3.attn2'), (28, 'mid-block.attentions.0.transformer-blocks.4.attn2'), (29, 'mid-
  block.attentions.0.transformer-blocks.5.attn2'), (30, 'mid-block.attentions.0.transformer-blocks.6.attn2'), (31, 'mid-
  block.attentions.0.transformer-blocks.7.attn2'), (32, 'mid-block.attentions.0.transformer-blocks.8.attn2'), (33, 'mid-
  block.attentions.0.transformer-blocks.9.attn2'), (34, 'up-blocks.0.attentions.0.transformer-blocks.0.attn2'), (35, 'up-
  blocks.0.attentions.0.transformer-blocks.1.attn2'), (36, 'up-blocks.0.attentions.0.transformer-blocks.2.attn2'), (37, 'up-
  blocks.0.attentions.0.transformer-blocks.3.attn2'), (38, 'up-blocks.0.attentions.0.transformer-blocks.4.attn2'), (39, 'up-
  blocks.0.attentions.0.transformer-blocks.5.attn2'), (40, 'up-blocks.0.attentions.0.transformer-blocks.6.attn2'), (41, 'up-
  blocks.0.attentions.0.transformer-blocks.7.attn2'), (42, 'up-blocks.0.attentions.0.transformer-blocks.8.attn2'), (43, 'up-
  blocks.0.attentions.0.transformer-blocks.9.attn2'), (44, 'up-blocks.0.attentions.1.transformer-blocks.0.attn2'), (45, 'up-
  blocks.0.attentions.1.transformer-blocks.1.attn2'), (46, 'up-blocks.0.attentions.1.transformer-blocks.2.attn2'), (47, 'up-
  blocks.0.attentions.1.transformer-blocks.3.attn2'), (48, 'up-blocks.0.attentions.1.transformer-blocks.4.attn2'), (49, 'up-
  blocks.0.attentions.1.transformer-blocks.5.attn2'), (50, 'up-blocks.0.attentions.1.transformer-blocks.6.attn2'), (51, 'up-
  blocks.0.attentions.1.transformer-blocks.7.attn2'), (52, 'up-blocks.0.attentions.1.transformer-blocks.8.attn2'), (53, 'up-
  blocks.0.attentions.1.transformer-blocks.9.attn2'), (54, 'up-blocks.0.attentions.2.transformer-blocks.0.attn2'), (55, 'up-
  blocks.0.attentions.2.transformer-blocks.1.attn2'), (56, 'up-blocks.0.attentions.2.transformer-blocks.2.attn2'), (57, 'up-
  blocks.0.attentions.2.transformer-blocks.3.attn2'), (58, 'up-blocks.0.attentions.2.transformer-blocks.4.attn2'), (59, 'up-
  blocks.0.attentions.2.transformer-blocks.5.attn2'), (60, 'up-blocks.0.attentions.2.transformer-blocks.6.attn2'), (61, 'up-
  blocks.0.attentions.2.transformer-blocks.7.attn2'), (62, 'up-blocks.0.attentions.2.transformer-blocks.8.attn2'), (63, 'up-
  blocks.0.attentions.2.transformer-blocks.9.attn2'), (64, 'up-blocks.1.attentions.0.transformer-blocks.0.attn2'), (65, 'up-
  blocks.1.attentions.0.transformer-blocks.1.attn2'), (66, 'up-blocks.1.attentions.1.transformer-blocks.0.attn2'), (67, 'up-
  blocks.1.attentions.1.transformer-blocks.1.attn2'), (68, 'up-blocks.1.attentions.2.transformer-blocks.0.attn2'), (69, 'up-
  blocks.1.attentions.2.transformer-blocks.1.attn2')]
```

Figure 41. **Layers to Probe for SD-XL**. Indexing of cross-attention layers in the UNet.

SD-v1-5

```
[(0, 'down_blocks.0.attentions.0.transformer_blocks.0.attn2.to_k'), (1, 'down_blocks.0.attentions.0.transformer_blocks.0.attn2.to_v'), (2,
'down_blocks.0.attentions.1.transformer_blocks.0.attn2.to_k'), (3, 'down_blocks.0.attentions.1.transformer_blocks.0.attn2.to_v'), (4,
'down_blocks.1.attentions.0.transformer_blocks.0.attn2.to_k'), (5, 'down_blocks.1.attentions.0.transformer_blocks.0.attn2.to_v'), (6,
'down_blocks.1.attentions.1.transformer_blocks.0.attn2.to_k'), (7, 'down_blocks.1.attentions.1.transformer_blocks.0.attn2.to_v'), (8,
'down_blocks.2.attentions.0.transformer_blocks.0.attn2.to_k'), (9, 'down_blocks.2.attentions.0.transformer_blocks.0.attn2.to_v'), (10,
'down_blocks.2.attentions.1.transformer_blocks.0.attn2.to_k'), (11, 'down_blocks.2.attentions.1.transformer_blocks.0.attn2.to_v'), (12,
'mid_block.attentions.0.transformer_blocks.0.attn2.to_k'), (13, 'mid_block.attentions.0.transformer_blocks.0.attn2.to_v'), (14,
'up_blocks.1.attentions.0.transformer_blocks.0.attn2.to_k'), (15, 'up_blocks.1.attentions.0.transformer_blocks.0.attn2.to_v'), (16,
'up_blocks.1.attentions.1.transformer_blocks.0.attn2.to_k'), (17, 'up_blocks.1.attentions.1.transformer_blocks.0.attn2.to_v'), (18,
'up_blocks.1.attentions.2.transformer_blocks.0.attn2.to_k'), (19, 'up_blocks.1.attentions.2.transformer_blocks.0.attn2.to_v'), (20,
'up_blocks.2.attentions.0.transformer_blocks.0.attn2.to_k'), (21, 'up_blocks.2.attentions.0.transformer_blocks.0.attn2.to_v'), (22,
'up_blocks.2.attentions.1.transformer_blocks.0.attn2.to_k'), (23, 'up_blocks.2.attentions.1.transformer_blocks.0.attn2.to_v'), (24,
'up_blocks.2.attentions.2.transformer_blocks.0.attn2.to_k'), (25, 'up_blocks.2.attentions.2.transformer_blocks.0.attn2.to_v'), (26,
'up_blocks.3.attentions.0.transformer_blocks.0.attn2.to_k'), (27, 'up_blocks.3.attentions.0.transformer_blocks.0.attn2.to_v'), (28,
'up_blocks.3.attentions.1.transformer_blocks.0.attn2.to_k'), (29, 'up_blocks.3.attentions.1.transformer_blocks.0.attn2.to_v'), (30,
'up_blocks.3.attentions.2.transformer_blocks.0.attn2.to_k'), (31, 'up_blocks.3.attentions.2.transformer_blocks.0.attn2.to_v')]
```

Figure 42. **Layers to Probe for SD-v1-5**. Indexing of cross-attention layers in the UNet.

SD-v2-1

```
[(0, 'down_blocks.0.attentions.0.transformer_blocks.0.attn2.to_k'), (1, 'down_blocks.0.attentions.0.transformer_blocks.0.attn2.to_v'), (2,
'down_blocks.0.attentions.1.transformer_blocks.0.attn2.to_k'), (3, 'down_blocks.0.attentions.1.transformer_blocks.0.attn2.to_v'), (4,
'down_blocks.1.attentions.0.transformer_blocks.0.attn2.to_k'), (5, 'down_blocks.1.attentions.0.transformer_blocks.0.attn2.to_v'), (6,
'down_blocks.1.attentions.1.transformer_blocks.0.attn2.to_k'), (7, 'down_blocks.1.attentions.1.transformer_blocks.0.attn2.to_v'), (8,
'down_blocks.2.attentions.0.transformer_blocks.0.attn2.to_k'), (9, 'down_blocks.2.attentions.0.transformer_blocks.0.attn2.to_v'), (10,
'down_blocks.2.attentions.1.transformer_blocks.0.attn2.to_k'), (11, 'down_blocks.2.attentions.1.transformer_blocks.0.attn2.to_v'), (12,
'mid_block.attentions.0.transformer_blocks.0.attn2.to_k'), (13, 'mid_block.attentions.0.transformer_blocks.0.attn2.to_v'), (14,
'up_blocks.1.attentions.0.transformer_blocks.0.attn2.to_k'), (15, 'up_blocks.1.attentions.0.transformer_blocks.0.attn2.to_v'), (16,
'up_blocks.1.attentions.1.transformer_blocks.0.attn2.to_k'), (17, 'up_blocks.1.attentions.1.transformer_blocks.0.attn2.to_v'), (18,
'up_blocks.1.attentions.2.transformer_blocks.0.attn2.to_k'), (19, 'up_blocks.1.attentions.2.transformer_blocks.0.attn2.to_v'), (20,
'up_blocks.2.attentions.0.transformer_blocks.0.attn2.to_k'), (21, 'up_blocks.2.attentions.0.transformer_blocks.0.attn2.to_v'), (22,
'up_blocks.2.attentions.1.transformer_blocks.0.attn2.to_k'), (23, 'up_blocks.2.attentions.1.transformer_blocks.0.attn2.to_v'), (24,
'up_blocks.2.attentions.2.transformer_blocks.0.attn2.to_k'), (25, 'up_blocks.2.attentions.2.transformer_blocks.0.attn2.to_v'), (26,
'up_blocks.3.attentions.0.transformer_blocks.0.attn2.to_k'), (27, 'up_blocks.3.attentions.0.transformer_blocks.0.attn2.to_v'), (28,
'up_blocks.3.attentions.1.transformer_blocks.0.attn2.to_k'), (29, 'up_blocks.3.attentions.1.transformer_blocks.0.attn2.to_v'), (30,
'up_blocks.3.attentions.2.transformer_blocks.0.attn2.to_k'), (31, 'up_blocks.3.attentions.2.transformer_blocks.0.attn2.to_v')]
```

Figure 43. **Layers to Probe for SD-v2-1**. Indexing of cross-attention layers in the UNet.

OpenJourney

```
[(0, 'down_blocks.0.attentions.0.transformer_blocks.0.attn2.to_k'), (1,
'down_blocks.0.attentions.0.transformer_blocks.0.attn2.to_v'), (2, 'down_blocks.0.attentions.1.transformer_blocks.0.attn2.to_k'),
(3, 'down_blocks.0.attentions.1.transformer_blocks.0.attn2.to_v'), (4,
'down_blocks.1.attentions.0.transformer_blocks.0.attn2.to_k'), (5, 'down_blocks.1.attentions.0.transformer_blocks.0.attn2.to_v'),
(6, 'down_blocks.1.attentions.1.transformer_blocks.0.attn2.to_k'), (7,
'down_blocks.1.attentions.1.transformer_blocks.0.attn2.to_v'), (8, 'down_blocks.2.attentions.0.transformer_blocks.0.attn2.to_k'),
(9, 'down_blocks.2.attentions.0.transformer_blocks.0.attn2.to_v'), (10,
'down_blocks.2.attentions.1.transformer_blocks.0.attn2.to_k'), (11, 'down_blocks.2.attentions.1.transformer_blocks.0.attn2.to_v'),
(12, 'mid_block.attentions.0.transformer_blocks.0.attn2.to_k'), (13, 'mid_block.attentions.0.transformer_blocks.0.attn2.to_v'),
(14, 'up_blocks.1.attentions.0.transformer_blocks.0.attn2.to_k'), (15, 'up_blocks.1.attentions.0.transformer_blocks.0.attn2.to_v'),
(16, 'up_blocks.1.attentions.1.transformer_blocks.0.attn2.to_k'), (17, 'up_blocks.1.attentions.1.transformer_blocks.0.attn2.to_v'),
(18, 'up_blocks.1.attentions.2.transformer_blocks.0.attn2.to_k'), (19, 'up_blocks.1.attentions.2.transformer_blocks.0.attn2.to_v'),
(20, 'up_blocks.2.attentions.0.transformer_blocks.0.attn2.to_k'), (21, 'up_blocks.2.attentions.0.transformer_blocks.0.attn2.to_v'),
(22, 'up_blocks.2.attentions.1.transformer_blocks.0.attn2.to_k'), (23, 'up_blocks.2.attentions.1.transformer_blocks.0.attn2.to_v'),
(24, 'up_blocks.2.attentions.2.transformer_blocks.0.attn2.to_k'), (25, 'up_blocks.2.attentions.2.transformer_blocks.0.attn2.to_v'),
(26, 'up_blocks.3.attentions.0.transformer_blocks.0.attn2.to_k'), (27, 'up_blocks.3.attentions.0.transformer_blocks.0.attn2.to_v'),
(28, 'up_blocks.3.attentions.1.transformer_blocks.0.attn2.to_k'), (29, 'up_blocks.3.attentions.1.transformer_blocks.0.attn2.to_v'),
(30, 'up_blocks.3.attentions.2.transformer_blocks.0.attn2.to_k'), (31, 'up_blocks.3.attentions.2.transformer_blocks.0.attn2.to_v')]
```

Figure 44. **Layers to Probe for OpenJourney**. Indexing of cross-attention layers in the UNet.

DeepFloyd

```
[(0, 'down_blocks.1.attentions.0.add_k_proj'), (1, 'down_blocks.1.attentions.0.add_v_proj'), (2,
'down_blocks.1.attentions.1.add_k_proj'), (3, 'down_blocks.1.attentions.1.add_v_proj'), (4,
'down_blocks.1.attentions.2.add_k_proj'), (5, 'down_blocks.1.attentions.2.add_v_proj'), (6,
'down_blocks.2.attentions.0.add_k_proj'), (7, 'down_blocks.2.attentions.0.add_v_proj'), (8,
'down_blocks.2.attentions.1.add_k_proj'), (9, 'down_blocks.2.attentions.1.add_v_proj'), (10,
'down_blocks.2.attentions.2.add_k_proj'), (11, 'down_blocks.2.attentions.2.add_v_proj'), (12,
'down_blocks.3.attentions.0.add_k_proj'), (13, 'down_blocks.3.attentions.0.add_v_proj'), (14,
'down_blocks.3.attentions.1.add_k_proj'), (15, 'down_blocks.3.attentions.1.add_v_proj'), (16,
'down_blocks.3.attentions.2.add_k_proj'), (17, 'down_blocks.3.attentions.2.add_v_proj'), (18,
'mid_block.attentions.0.add_k_proj'), (19, 'mid_block.attentions.0.add_v_proj'), (20, 'up_blocks.0.attentions.0.add_k_proj'),
(21, 'up_blocks.0.attentions.0.add_v_proj'), (22, 'up_blocks.0.attentions.1.add_k_proj'), (23,
'up_blocks.0.attentions.1.add_v_proj'), (24, 'up_blocks.0.attentions.2.add_k_proj'), (25, 'up_blocks.0.attentions.2.add_v_proj'),
(26, 'up_blocks.0.attentions.3.add_k_proj'), (27, 'up_blocks.0.attentions.3.add_v_proj'), (28,
'up_blocks.1.attentions.0.add_k_proj'), (29, 'up_blocks.1.attentions.0.add_v_proj'), (30, 'up_blocks.1.attentions.1.add_k_proj'),
(31, 'up_blocks.1.attentions.1.add_v_proj'), (32, 'up_blocks.1.attentions.2.add_k_proj'), (33,
'up_blocks.1.attentions.2.add_v_proj'), (34, 'up_blocks.1.attentions.3.add_k_proj'), (35, 'up_blocks.1.attentions.3.add_v_proj'),
(36, 'up_blocks.2.attentions.0.add_k_proj'), (37, 'up_blocks.2.attentions.0.add_v_proj'), (38,
'up_blocks.2.attentions.1.add_k_proj'), (39, 'up_blocks.2.attentions.1.add_v_proj'), (40, 'up_blocks.2.attentions.2.add_k_proj'),
(41, 'up_blocks.2.attentions.2.add_v_proj'), (42, 'up_blocks.2.attentions.3.add_k_proj'), (43,
'up_blocks.2.attentions.3.add_v_proj')]
```

Figure 45. **Layers to Probe for DeepFloyd**. Indexing of cross-attention layers in the UNet.

E. More Visualizations for Model Editing

Figure 46. **SDv1-5 Edits for "style"**. We show successful model editing on the layers identified by LOCOGEN. In case of the images generated by the edited model, we can observe that the trademarked brushstrokes of the artist *Van Gogh* are missing.

Figure 47. **Open-Journey Edits for "style"**. We show successful model editing on the layers identified by LOCOGEN. In case of the images generated by the edited model, we can observe that the trademarked brushstrokes of the artist *Van Gogh* are missing. For some of the images from the edited model, we even find that the patterns in the sky which is another trademark *Van Gogh* signature have them deleted.

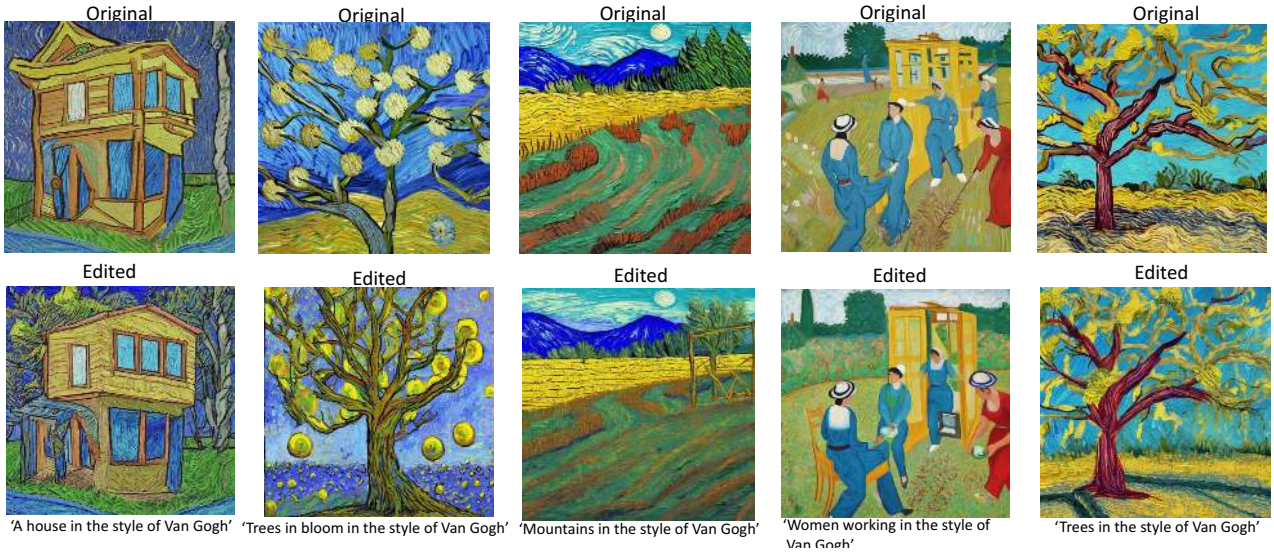


Figure 48. **SD-v2-1 Edits for "style"**. We show successful model editing on the layers identified by LOCOGEN. In case of the images generated by the edited model, we can observe that the trademarked brushstrokes of the artist *Van Gogh* are missing. For some of the images from the edited model, we even find that the patterns in the sky which is another trademark *Van Gogh* signature have them deleted.

Figure 49. **SDXL Edits for "style"**. We show successful model editing on the layers identified by LOCOGEN. In case of the images generated by the edited model, we can observe that the trademarked brushstrokes of the artist *Van Gogh* are missing. For some of the images from the edited model (e.g., Trees in the style of Van Gogh), we even find that the patterns in the sky which is another trademark *Van Gogh* signature have them deleted.

Figure 50. **Fact Editing Across Different Text-to-Image Models.**. We show that LOCOEDIT can successfully update out-dated facts in text-to-image models with the correct facts.

Figure 51. **Fact Editing Across Different Text-to-Image Models.**. We show that LOCOEDIT can successfully update out-dated facts in text-to-image models with the correct facts.

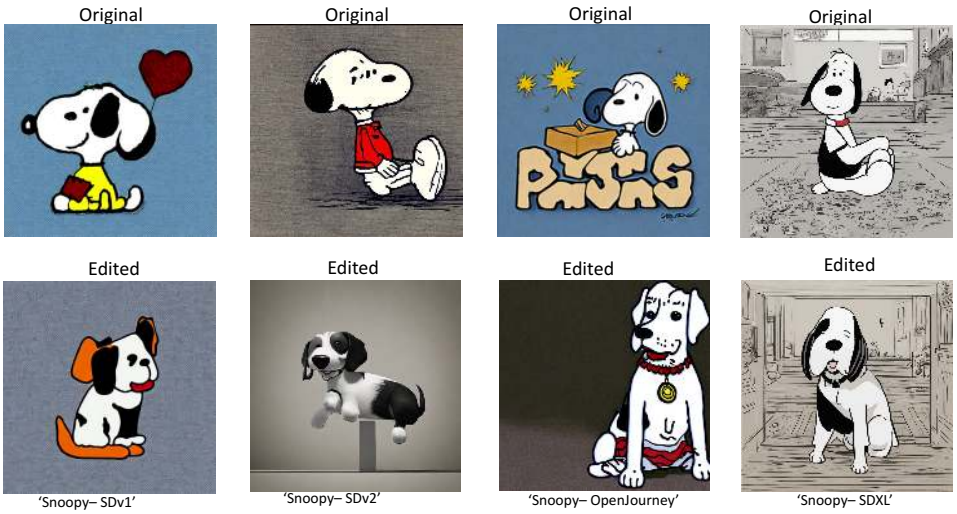


Figure 52. **Trademarked Object Editing Across Different Text-to-Image Models.**. We show that LOCOEDIT can successfully modify trademarked objects by performing weight-space editing in the locations identified by LOCOGEN.

Figure 53. **Trademarked Object Editing Across Different Text-to-Image Models.**. We show that LOCOEDIT can successfully modify trademarked objects by performing weight-space editing in the locations identified by LOCOGEN.

F. Updating All Layers vs. Localized Layers

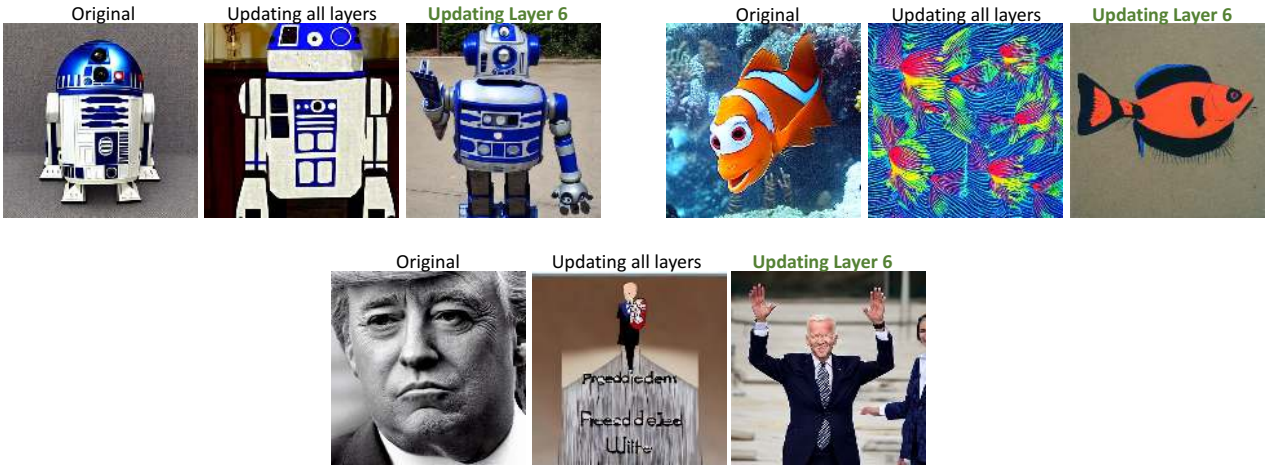


Figure 54. **Localized Editing vs. Non-Localized Editing**. For "objects" and "facts", we find that updating the layers identified by LOCOGEN is better than editing all the layers. For e.g., in this qualitative study, we find that updating all the layers does not lead to correct outputs for certain cases involving prompts corresponding to "objects" and "facts".

G. Hyper-parameter Search

In this section, we enumerate the hyper-parameter m for each text-to-image model. In particular, we use Stable-Diffusion-v1-5 as a base to first obtain the optimal value of m. For Stable-Diffusion-v1-5, we find this value to be $m = 2$. Based on the percentage of the total layers, in the model it encompasses, we perform a local search around that hyper-parameter. In this way, we find the optimal value of $m = 3$ for Stable-Diffusion-v2-1, $m = 2$ for OpenJourney, $m = 5$ for SD-XL and $m = 3$ for DeepFloyd.

Figure 55. **Selection of m for SD-v1-5.** At Layer 8, which encodes "style", we vary the value of m from 1 to the maximum value of m for the given model (to $m = 9$ for a total of 16 layers, starting from Layer 8). We find that at $m = 2$, the style of *Van Gogh* for the prompt *A 'house in the style of Van Gogh'* is significantly removed. We therefore choose $m = 2$ as the layers to edit.

H. Model Editing Hyper-parameters

We set the following hyper-parameters for λ_K and λ_V in LOCOEDIT as 0.01 for all the text-to-image models, as it led to the best editing results.

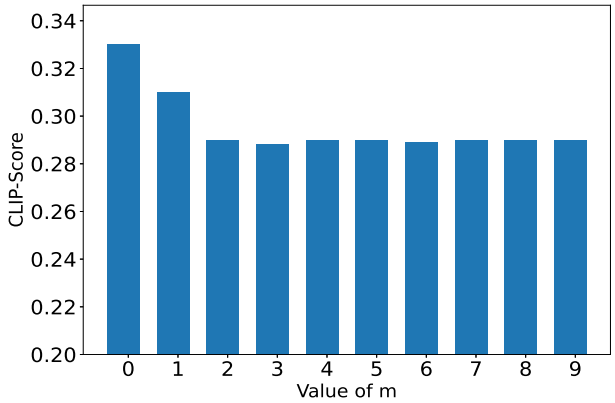


Figure 56. **Selection of m for SD-v1-5 - CLIP-Score.** m=0 corresponds to the original generation. We find that the CLIP-Score saturates after $m = 2$, therefore we choose $m = 2$ in our experiments. This figure is for the "style" attribute; We choose the value of m in a similar way for "objects" and "facts", where the optimal m comes out to be 2.

I. Comparison with Other Model Editing Baselines

Given that other model editing baselines (Kumari et al., 2023; Gandikota et al., 2023; Basu et al., 2023) primarily operate on SD-v1 versions, we compare our method with these baselines on SD-v1-5. In Fig 57 – we provide a comprehensive comparison and analysis of how LOCOEDIT compares to other methods.

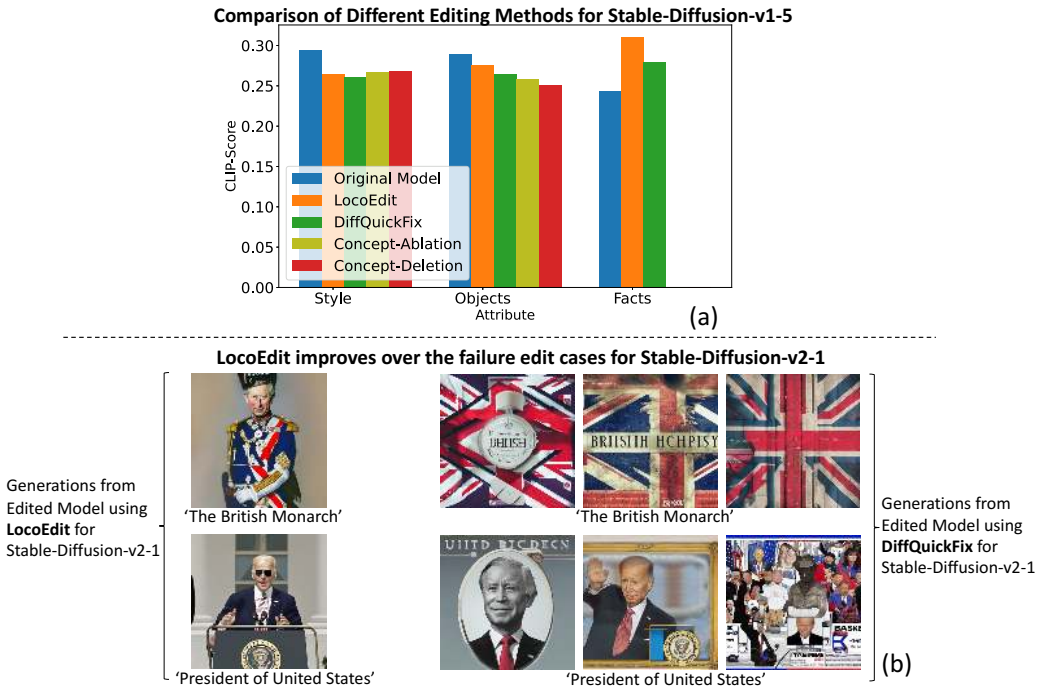


Figure 57. **Comparison of our editing method when compared to other baselines.** (a) We compare our method with existing model editing methods – DiffQuickFix (Basu et al., 2023), Concept-Ablation (Kumari et al., 2023) and Concept-Deletion (Gandikota et al., 2023). Amongst them only DiffQuickFix is a closed-form update method, whereas the other methods are fine-tuning based approaches. We find that LOCOEDIT has comparable performance to existing methods for style, better performance for facts and for objects slightly worse performance. The drop in CLIP-Score is lesser in our method as model edits via LOCOEDIT modifies certain characteristics of the object, rather than the whole object which is enough to remove the trademarked characteristics of the object. For facts, we find that our method has a better increase in the CLIP-Score than (Basu et al., 2023) therefore highlighting effective factual edits. We note that both Concept-Ablation and Concept-Deletion do not perform factual editing. (b) Although DiffQuickFix from (Basu et al., 2023) was primarily built on Stable-Diffusion-v1 versions, we extend their framework to Stable-Diffusion-v2-1. In particular, we identify certain failure cases primarily on factual edits, where our method LOCOEDIT performs better.

J. Human-Study

In this section, we report additional details about human experiments we did to verify LOCOGEN. Figure 58 shows visualization of some pairs that we used in our experiments. We used 11 different prompts listed below (visual attribute is written in paranthesis) and 3 generations per each prompt for each of the models.

- "a house in the style of van gogh" **(style)**
- "a town in the style of monet" **(style)**
- "a town in the style of monet" **(style)**
- "british monarch" **(fact)**
- "cat" **(object)**
- "elephant painting the style of salvador dali" **(style)**
- "nemo" **(object)**
- "president of the united states"
- "rocks in the ocean in the style of monet" **(style)**
- "snoopy" **(object)**
- "women working in a garden in the style of van gogh" **(style)**

For each image pair based on the visual attribute of the corresponding prompt, we ask one of the following questions from evaluators and they have give a rating from 1 to 5.

- **style**: Has the artistic style (‘<artist name>’) been decreased from original image (Left) compared to modified image (Right)?
  5 – maximally removed; 1 – not removed (modified image still has the artistic style).
- **object**: Has the main object (‘<object name>’) been modified from original image (Left) compared to modified (Right)?
  5 – maximally modified (other object is there); 1 – not modified at all (object is still there).
- **fact**: Has the fact (‘<fact name>’) been updated from original image (Left) compared to modified (Right)?
  5 – maximally updated (other person is there); 1 – not updated at all (previous person is still there).

in 92.58% of pairs, evaluators give a rating **above** 1 which shows that the edit is effective. Now, we report more detailed scores for each of different attributes:

- style
  - in 90.28% of cases, evaluators give a rating of at least 2.
  - in 63.61% of cases, evaluators give a rating of at least 3.
  - in 37.50% of cases, evaluators give a rating of at least 4.
  - in 13.89% of cases, evaluators give the rating 5.
- fact
  - in 100.00% of cases, evaluators give a rating of at least 2.
  - in 95.83% of cases, evaluators give a rating of at least 3.
  - in 86.67% of cases, evaluators give a rating of at least 4.
  - in 59.17% of cases, evaluators give the rating 5.
- object
  - in 92.22% of cases, evaluators give a rating of at least 2.
  - in 81.11% of cases, evaluators give a rating of at least 3.
  - in 62.22% of cases, evaluators give a rating of at least 4.
  - in 39.44% of cases, evaluators give the rating 5.

For different models, ratings are as follows:

- SD-v1
  - evaluators give the rating of at least 2 in 95.76% of cases.
  - evaluators give the rating of at least 3 in 82.42% of cases.
  - evaluators give the rating of at least 4 in 57.58% of cases.
  - evaluators give the rating of at least 5 in 32.73% of cases.
- SD-v2
  - evaluators give the rating of at least 2 in 87.27% of cases.
  - evaluators give the rating of at least 3 in 61.82% of cases.
  - evaluators give the rating of at least 4 in 43.03% of cases.
  - evaluators give the rating of at least 5 in 13.33% of cases.
- SD-XL
  - evaluators give the rating of at least 2 in 90.91% of cases.
  - evaluators give the rating of at least 3 in 69.70% of cases.
  - evaluators give the rating of at least 4 in 49.70% of cases.

  - evaluators give the rating of at least 5 in 31.52% of cases.

- OpenJourney
  - evaluators give the rating of at least 2 in 96.36% of cases.
  - evaluators give the rating of at least 3 in 83.03% of cases.
  - evaluators give the rating of at least 4 in 62.42% of cases.
  - evaluators give the rating of at least 5 in 38.79% of cases.

Above results shows that LOCOGEN is effective and works better for facts and for models OpenJourney and SD-v1.

K. Neuron-Level Model Editing

In this section we discuss more details about experiments for neuron-level model editing.

K.1. Prompts

In this section, we provide more details about the set of prompts used to report CLIP-Score in Figure 10.

- **Van Gogh**
  - "a painting of rocky ocean shore under the luminous night sky in the style of Van Gogh"
  - "A painting of lone figure contemplates on a cliff, surrounded by swirling skies in the style of Van Gogh"
  - "Majestic mountains take on an ethereal quality in sky, painted by Van Gogh"
  - "Two trees in a sunlit field, painted by Van Gogh"
  - "A flower-filled field in the style of Van Gogh"
  - "painting of a river on a warm sunset in the style of Van Gogh"
  - "painting of olive trees in the style of Van Gogh"
  - "painting of a field with mountains in the background in the style of Van Gogh"
- **Monet**
  - "painting of women working in the garden in the style of Monet",
  - "rocks in the ocean, in the style of Monet",
  - "a painting of a city in the style of Monet",
  - "a painting of a river in the style of Monet",
  - "Monet style painting of a person on a cliff",
  - "a painting of a town, in the style of Monet",
  - "a painting of a sunset, in the style of Monet",
  - "a painting of mountains, in the style of Monet",
  - "Monet style painting of flowers in a field",
  - "a painting of a landscape in the style of Monet",
  - "two trees in a field, painting in the style of Monet"
- **Salvador Dali**
  - "the persistence of memory painting in the style of Salvador Dali",
  - "the elephant painting in the style of Salvador Dali",
  - "soft construction with boiled beans painting in the style of Salvador Dali",
  - "galatea of the spheres painting in the style of Salvador Dali",
  - "the temptation of st. anthony painting in the style of Salvador Dali",
  - "swans reflecting elephants painting in the style of Salvador Dali",
  - "enigma of desire painting in the style of Salvador Dali",

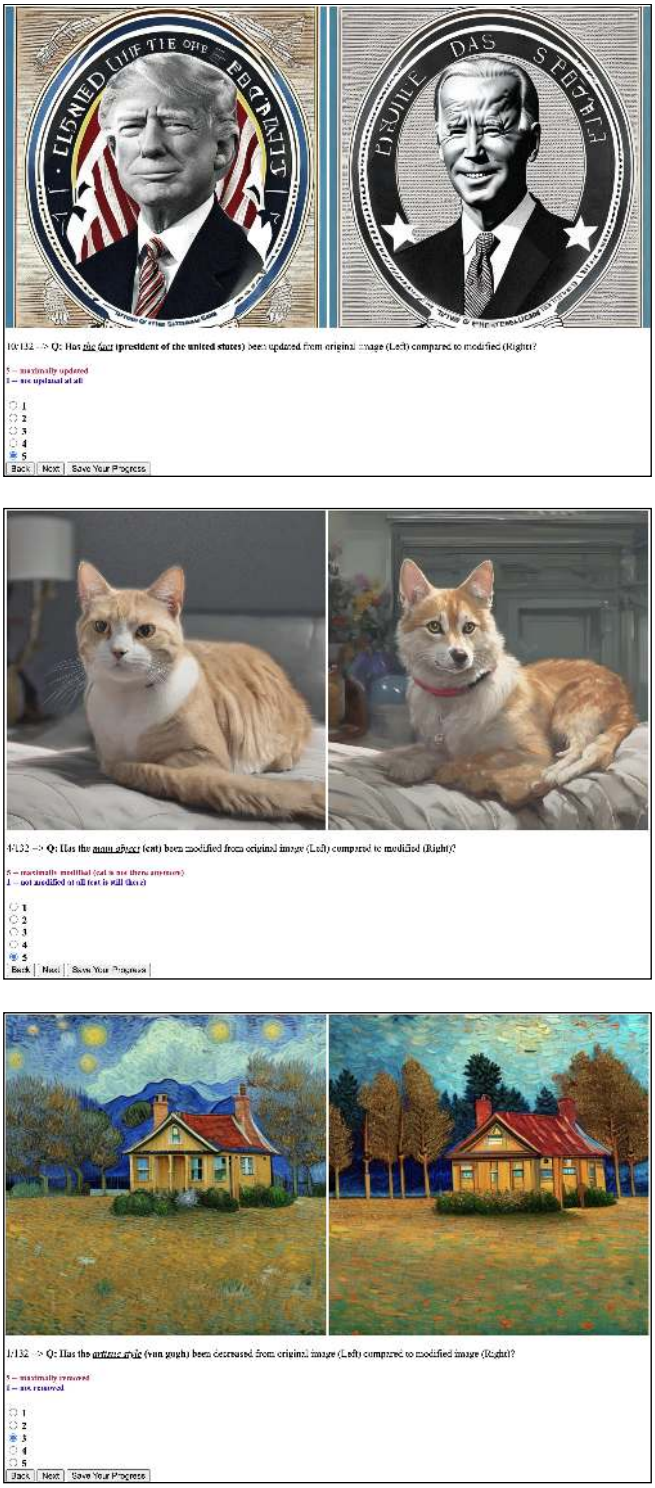


Figure 58. Examples of some tests that evaluators see in human-study.

  - "slave market with the disappearing bust of voltaire painting of Salvador Dali",
  - "the meditative rose painting in the style of Salvador Dali",
  - "melting watch painting in the style of Salvador Dali",

- **Jeremy Mann**
  - "In the style of Jeremy Mann, a view of a city skyline at sunset, with a warm glow spreading across the sky and the buildings below",
  - "In the style of Jeremy Mann, an urban scene of a group of people gathered on a street corner, captured in a moment of quiet reflection",
  - "In the style of Jeremy Mann, a surreal composition of floating objects, with a dreamlike quality to the light and color",
  - "In the style of Jeremy Mann, a view of a city street at night, with the glow of streetlights and neon signs casting colorful reflections on the wet pavement",
  - "In the style of Jeremy Mann, a moody, atmospheric scene of a dark alleyway, with a hint of warm light glowing in the distance",
  - "In the style of Jeremy Mann, an urban scene of a group of people walking through a park, captured in a moment of movement and energy",
  - "In the style of Jeremy Mann, a landscape of a forest, with dappled sunlight filtering through the leaves and a sense of stillness and peace",
  - "In the style of Jeremy Mann, a surreal composition of architectural details and organic forms, with a sense of tension and unease in the composition",
  - "In the style of Jeremy Mann, an abstract composition of geometric shapes and intricate patterns, with a vibrant use of color and light",
  - "In the style of Jeremy Mann, a painting of a bustling city at night, captured in the rain-soaked streets and neon lights",

- **Greg Rutkowski**
  - "a man riding a horse, dragon breathing fire, painted by Greg Rutkowski",
  - "a dragon attacking a knight in the style of Greg Rutkowski",
  - "a demonic creature in the wood, painting by Greg Rutkowski",
  - "a man in a forbidden city, in the style of Greg Rutkowski",
  - "painting of a group of people on a dock by Greg Rutkowski",
  - "a king standing, with people around in a hall, painted by Greg Rutkowski",
  - "two magical characters in space, painting by Greg Rutkowski",
  - "a man with a fire in his hands in the style of Greg Rutkowski",
  - "painting of a woman sitting on a couch by Greg Rutkowski",
  - "a man with a sword standing on top of a pile of skulls, in the style of Greg Rutkowski",

- **Pablo Picasso**
  - "Painting of nude figures by Pablo Picasso",
  - "painting of a grieving woman in the style of Pablo Picasso",
  - "painting of three dancers by Pablo Picasso",
  - "portrait of a girl in front of mirror, painted by Pablo Picasso",
  - "painting of a bird, in the style of Pablo Picasso",
  - "painting of a blind musician, by Pablo Picasso",
  - "painting of a room in the style of Pablo Picasso",
  - "painting of an acrobat in performance, by Pablo Picasso",

Prompts that we used to compute z-score are same as above except that for each artist, we put the artist name in all of the above prompts. To generate prompts withput any artistic style, we remove artist name from the prompts listed above.

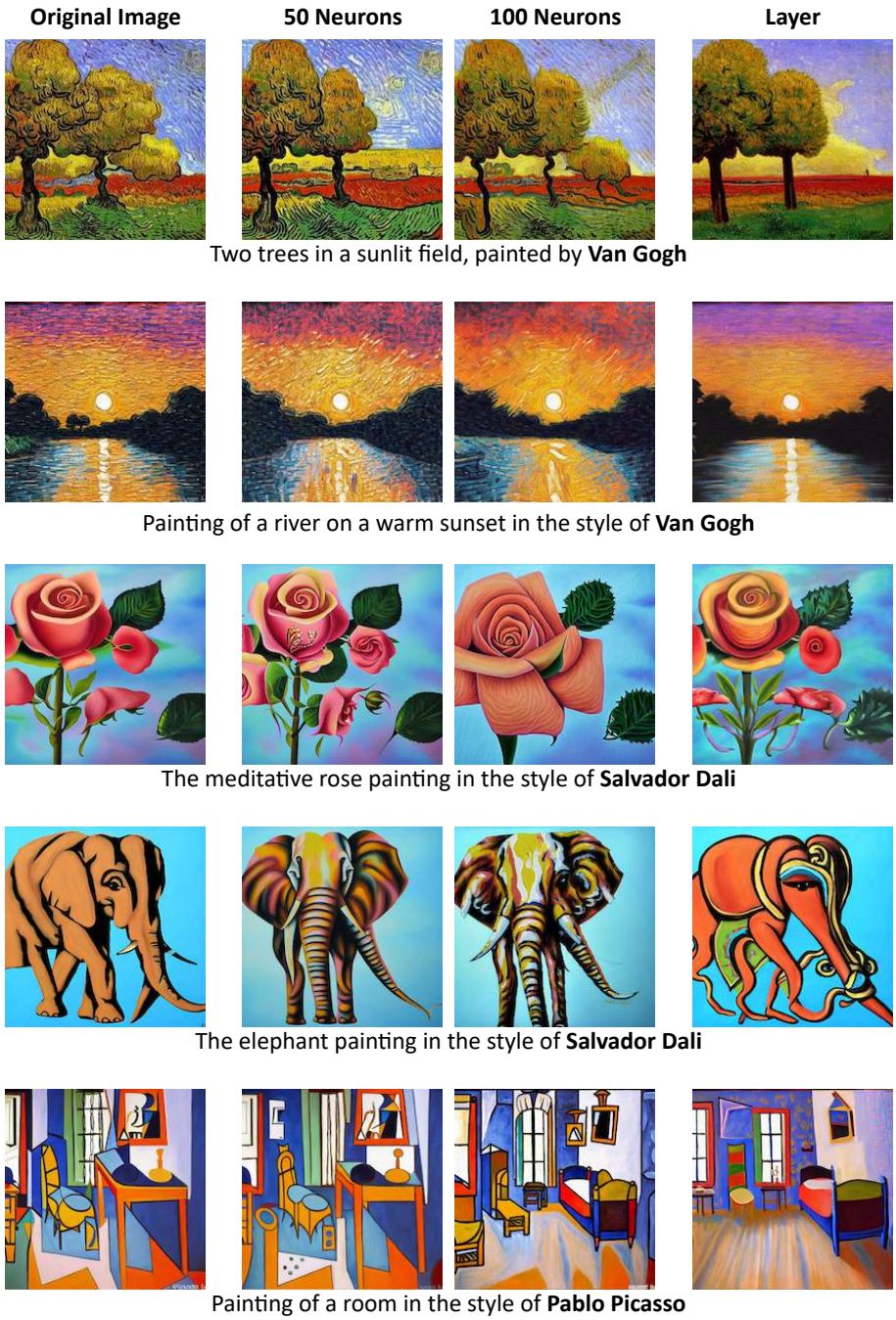


Figure 59. **Neuron Level Model Editing - Qualitative**. Results when applying **neuron-level dropout** on identified neurons in layers specified with LOCOGENon Stable Diffusion v1.5. Each row corresponds to an input text prompt featuring a particular artistic style. First column shows the image generated without any intervention while second and third column visualize images when 50 and 100 neurons out of 1280 neurons in controlling layers are modified, respectively. Last column shows images when a different embedding is given to controlling layers.

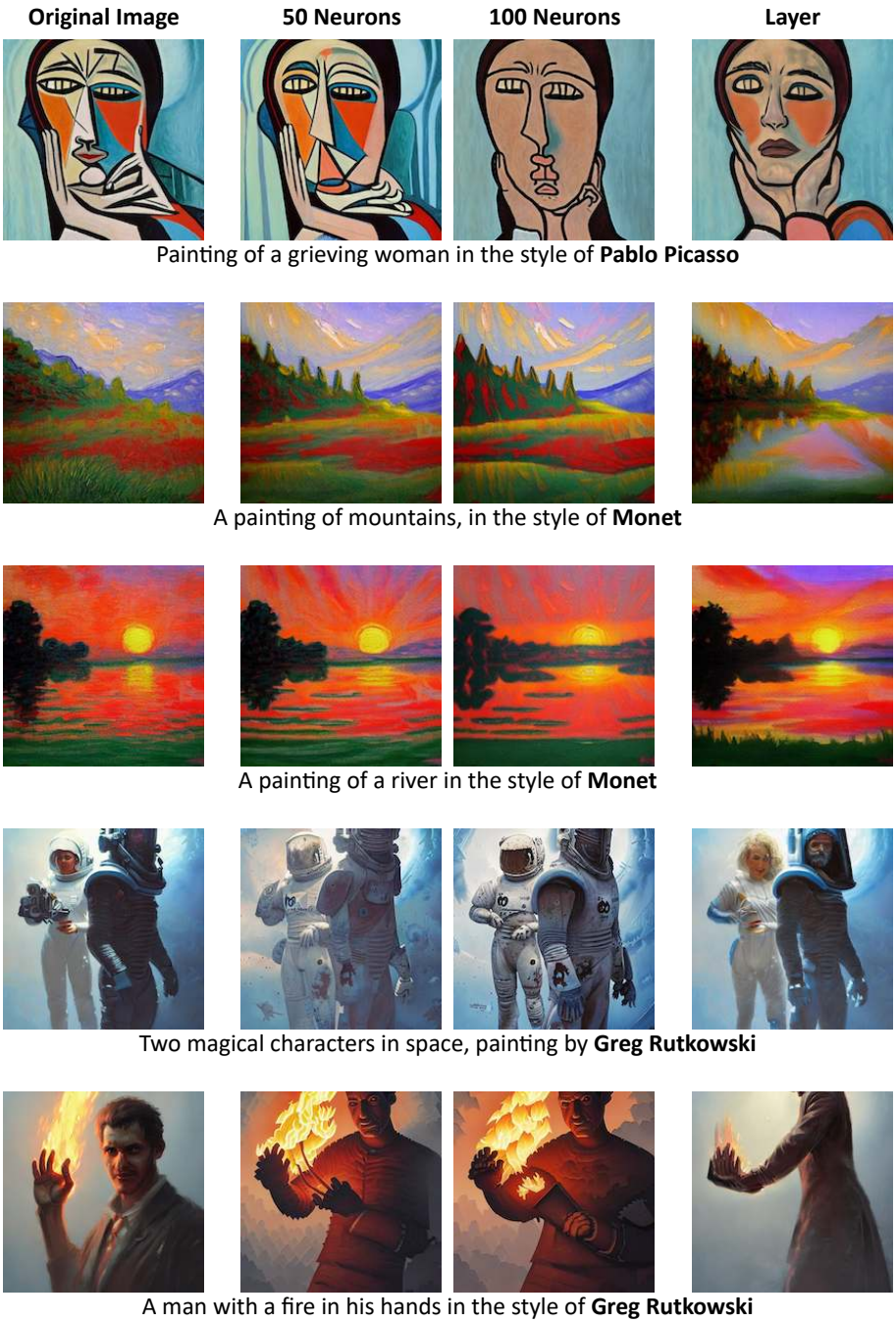


Figure 60. **Neuron Level Model Editing - Qualitative**. Results when applying **neuron-level dropout** on identified neurons in layers specified with LOCOGENon Stable Diffusion v1.5. Each row corresponds to an input text prompt featuring a particular artistic style. First column shows the image generated without any intervention while second and third column visualize images when 50 and 100 neurons out of 1280 neurons in controlling layers are modified, respectively. Last column shows images when a different embedding is given to controlling layers.

Editing Method	Style Edited Model	Object Edited Model
No Editing	30.04	30.04
DiffQuickFix	29.61	29.32
LocoEdit(Ours)	29.99	29.40

Table 1. **CLIP-Score of the generated images with the original prompt.** We find that while both DiffQuickFix (Basu et al., 2023) and our method leads to a slight drop in the CLIP-Score for the edited model, the decrease in the CLIP-Score from our method is slightly lesser – thereby highlighting that our localized editing method does not affect generic prompts significantly.

K.2. Visualization

L. Robustness of the Edited Model

In this section, we discuss the robustness of the edited model to generic prompts. In particular, we curate a set of 320 prompts from MS-COCO with 80 objects and 4 locations ("beach", "forest", "city", "house") for each. We then compare the average CLIP-Score of the generations from original model vs. edited model across "style" and "object" attributes. For edited models, we compare our method with DiffQuickFix, as it also edits models using a closed-form update. Overall from Table 1 – we find that LOCOEDIT does not harm generic prompts, highlighting the robustness of our method.

M. DeepFloyd Edit Limitations

While LOCOGEN is adept at localizing knowledge in DeepFloyd, we find that the closed-form edits are not suitable for the DeepFloyd model. In particular, we find that performing edits using LOCOEDIT at the locations identified by LOCOGEN does not lead to semantically correct edits. We hypothesize that this is due to the difference in the text-encoder between [SD-v1, SD-v2, OpenJourney, SDXL] and DeepFloyd. Text-to-image models (except DeepFloyd) consist of text-encoders (e.g., CLIP) which utilizes a causal attention mechanism. Given that closed-form edits require a mapping between the last-subject token embedding to a target embedding, there is less information leakage compared to a T5 text-encoder which implements a bi-directional self-attentio (see Figure. 62 for a visualization of this non-causal attention). In the case of T5, mapping *only* the last-subject token embedding to a target embedding might be insufficient and mapping all the tokens can lead to huge informational loss from the model. Below we provide some examples from editing the DeepFloyd model – we primarily observe non-semantic generations from the edited model. Designing fast editing methods for DeepFloyd like models which utilizes bi-directional attention is an important future course of study.

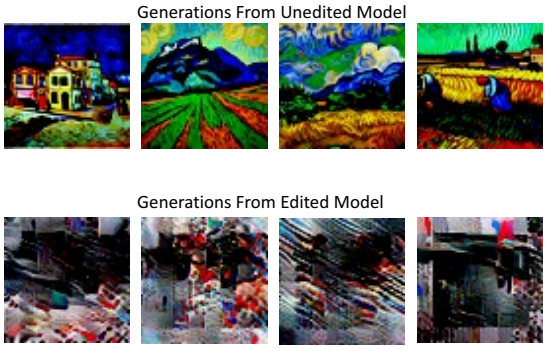
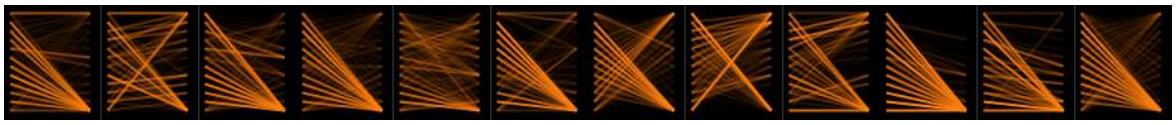
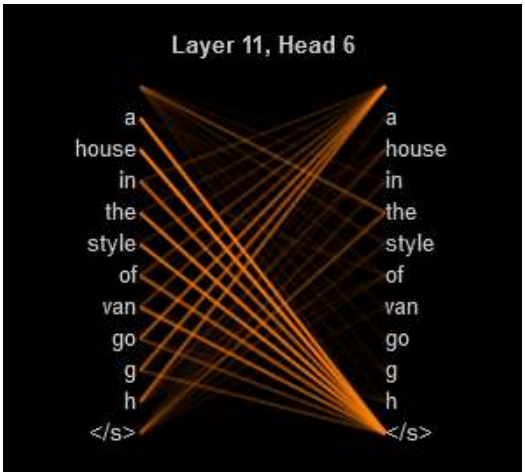


Figure 61. **Model Editing for DeepFloyd.** We find that the edited DeepFloyd model leads to generations of non-semantic outputs. In this figure, we show results for editing "style". However, we observe similar results for other attributes such as "objects" and "facts". Prompts used are corresponding to *Van Gogh*.



(a)
t



(b)
t

Figure 62. Attention map from the last layer of T5 encoder for the prompt “a house in the style of van gogh”. Figure (a) shows the attention map between the tokens in the prompt for each of the 12 heads in the last layer of T5 encoder. The non causal nature of the mask is evident from the significant attention weights from later tokens to previous tokens in the prompt. This is more easily observed in the Figure (b) which highlights the same attention map from head 6.