ATTENTION-ERASER: TRAINING LATENTS IN THE DENOISING PROCESS TO ADJUST THE SIZE OF USER-SELECT TOKENS

Kathy Peng
Visual Computing Lab
Brown University
Providence, RI
kathy_peng@brown.edu

Advisors:
James Tompkin
Stephen Bach

ABSTRACT

"A dog on the street" "A small dog on the street" with Attention-Eraser

Latent Diffusion Models (LDM) use cross attention mechanism in the denoising process to guide image generation from user-provided prompts. Intuitively, the attention values should contain semantics information about tokens in the prompt, including the probability of generating the tokens in each of the spatial patches. This paper demonstrates a method of adjusting the size of the user-selected object by updating the latents using loss objectives based on attention values. This would gradually force the latents to change into ones that would minimize the attention values to particular tokens at specific spatial region. To the best of my knowledge, there has not been studies trying to control the size and proportion of objects being generated. Additionally, the paper will discuss the existing methods of achieving control over diffusion models and compare them in terms of cost of computation, types of edits, and level of performance.

Keywords  Latent Diffusion Models · Cross Attention · DDPM · UNet

1 Introduction

Since the introduction of open-sourced large text-to-image generation models [1], many attempts have been made to achieve some level of control over the image generation process and these methods vary in both training costs and effect of edits. The commonly known types of edits include style transfer using prompt as instruction, details editing of a selected object in the image, and swapping objects while keeping the background intact. These edits come at different costs, ranging from fine-tuning the entire diffusion model [2] to simply swapping attention maps without any training [3]. In this paper, I will present a method of controlling the size of the object being generated by shaping the latents in the denoising process based on a loss objective that is a function of the attention values of the user-selected tokens. Particularly, the method proposed in the paper works with the cross attention layers in the denoising UNet architecture, which are key to the control of image generation. In terms of the cost of computation, the method requires less training than finetuning the entire diffusion model but more than simply swapping the attention weights of two images. In terms
of editing effects, since the method does not bind tokens to objects through finetuning, the objects appeared after applying the method might appear to be significantly different from before, representing objects generated from different initial latents or random seeds.

2 Related Work

2.1 Denoising

Diffusion models are text-guided image generation models that originates from the hierarchical VAEs. It is a probabilistic model which aims to learn a variational lower bound of a data distribution \( p(x) \) by learning to reverse a fixed Markov Chain of sequentially adding Gaussian noise, also known as the forward diffusion process. The Latent Diffusion Model, released by Stability AI, differs from previous models by using a UNet for the process and the cross attention layers in the UNet for matching the tokens in the prompt to corresponding locations of the images.

In the denoising process, the Latent Diffusion Model is trained to minimize the reconstruction loss of the noise predicted by the UNet at the current time step \( t \) and the noise from the previous time step \( t-1 \). The term ‘noise’ and ‘latents’ will be used interchangeably in the paper. As shown by the pipeline below, the diffusion process without the conditioning can be seen as a process of gradually adding noise to an input image until it becomes purely Gaussian noise, and then learning a denoising UNet to reverse the process and obtaining the original image. The conditioning network is central to the text-guided image generation process, and in the Latent Diffusion Model, this is implemented as the cross attention mechanism in the UNet.

![Diagram of Latent Diffusion Model](image)

2.2 Cross Attention

In the denoising UNet, there are cross attention layers in both downsampling and upsampling of the UNet architecture. They are of different resolution size which represent the number of spatial patches it could fit, including 32x32, 16x16, 8x8, and 4x4. To compute the attention matrix, the user-supplied text prompt is first passed through a text encoder to convert into text embeddings. The text embeddings are then converted to K matrix and V matrix using learn-able projection matrices. We obtain the Q matrix from the similar projections of the intermediate features of UNet. Finally, the attention matrix is computed by the following formula:

\[
Attention(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d}}) \cdot V
\]
The resulting attention matrix at each individual cross attention layer is four-dimensional: (number of attention heads, dimension size, dimension size, prompt length). Intuitively, the attention matrix values represent the probability that the object corresponding to the token channel would appear in one of the spatial patches. Since the attention values represent probabilities, their range is between 0 and 1 and the values will sum up to 1 across all spatial patches for a particular token channel.

3 Methods

3.1 Loss Objective

The token loss objective is defined by the sum of the top $n$ largest attention values in the spatial patches, where $n$ is a hyperparameter, then summed across all user selected tokens $S$. After computing the token loss, the latents will be updated by the loss such that the new latents could better reflect the objective to minimize the attention values for the user-selected token, thus reducing the size of the object. Making direct changes to the latents of the diffusion model has been proven to generate changes to the images [8] [9] in the past.

$$\text{Loss}_{\text{tokens}} = \sum_{j=0}^{S} \sum_{i=0}^{n} \text{sorted} (\text{FlattenedAttentionValues}),$$

$$\text{Latent}_{t+1} = \text{Latent}_t - \alpha \times \text{Loss}_{\text{tokens}}$$

The hyperparameter $n$ will control the extent of attention paid to the selected token object. With larger values of $n$, the latents will learn to update itself attention to the object in larger spatial patches.

3.2 Training

The training happens in the first 20 timesteps out of the total 50 timesteps. This decision is based on the intuition that the early denoising timesteps are most important in determining the semantics of the images being generated, including of the composition of the images (background and foreground), the locations of the objects, and the structural relationships between them. We could define the Attention-Eraser pipeline to consist of the following steps:

<table>
<thead>
<tr>
<th>step</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>step 1</td>
<td>forward pass of denoising with text conditioning</td>
</tr>
<tr>
<td>step 2</td>
<td>compute token loss given $n$ and given $S$</td>
</tr>
<tr>
<td>step 3</td>
<td>update latents with the computed loss</td>
</tr>
<tr>
<td>step 4</td>
<td>predict the noise using UNet</td>
</tr>
<tr>
<td>step 5</td>
<td>perform classifier-free guidance</td>
</tr>
<tr>
<td>step 6</td>
<td>compute the previous latents</td>
</tr>
</tbody>
</table>

4 Results

The Attention-Eraser method has produced better control in object size than simple prompt engineering. For example, if the user wants to generate a dog on the street with the dog occupying only a small proportion of the image generated, using simple prompt engineering cannot achieve that.

A dog on the street far away  A little dog on the street  A small dog on the street
Using Attention-Eraser, the size of the dog can be controlled using the hyperparameter $n$. When $n$ is sufficiently large, the diffusion model will not generate a dog in the image.

"A dog on the street"  
Attention-Eraser with $n = 1$  
Attention-Eraser with $n = 3$

In the other example, our goal is to generate a man walking down the street with the man occupying a small proportion of the image frame. Similarly, we tried simple prompt engineering and compared results with attention-eraser. As demonstrated by the images below, there is no difference or change between the size of the man.

A man walking down the street  
A man walking down the street far away

When using Attention-Eraser to control the size of the object, we first tried to only modify the man token but found that increasing $n$ does not seem to control the size very well. A possible explanation for this is that in the prompt "A man walking down the street", the token "man" and token "walking" have strong association in constructing the man object. We then tried to modify both tokens and successfully achieved the desired outcomes.

Attention-Erase "man", $n=1$  
Attention-Erase "man", $n=2$  
Attention-Erase "man" and "walking", $n=2
References


