Adapting Interpretability Techniques to Generative Image Diffusion Models

Caleb Gupta, Vani Kanoria

University of Pennsylvania
https://github.com/CalebG1/TrustworthyML

Abstract: Interpretable models play a pivotal role in understanding the inner workings of complex black box models. Among the established techniques, LIME and SHAPley Values have gained prominence for their effectiveness. However, their application to generative diffusion models remains underexplored with regard to relating specific text in the prompt to generated imagery. Further, knowing what specifically in the image is contributing to which tokens would be useful in contexts of identifying misrepresentation of semantic meanings. In this project, we bridge this gap by adapting SHAP and LIME interpretability techniques to generative diffusion models and then working backwards from the generated models to highlight key imagery associated with each term. Specifically, we aim to elucidate the correspondence between words and objects within images generated by diffusion models. To achieve this, we analyze the intermediate steps of the diffusion process, which can be achieved with open source models. Alternatively, we use SOTA image segmenting to understand objects in the scene that could have direct correlations to specific phrases within the input prompt. This enables us to pinpoint the significance of words in relation to specific concepts depicted in the image. Our findings not only highlight the importance of individual words but also provide insights into the underlying concepts represented by these words. By offering a deeper understanding of the importance of tokens in a prompt and the interpretability of shapes within images, our approach contributes to advancing the state-of-the-art in explainable generative models.

1. BACKGROUND

Generative text-to-image models such as Stable Diffusion have become highly popular in recent times due to their ability to generate high quality images from natural language prompts. These models are trained on large natural language and image datasets, and are largely black box in nature which therefore motivates an approach to explainability that is accessible using black-box only techniques. The images produced by the model are a result of the prompt inputted by the user, and how the model perceives the prompt and the composition of images. This paper explores how the diffusion models detects the importance of different parts of a prompt, and also which parts of an image are relevant to each token in the prompt.

The importance of difference parts of a prompt and different parts of the image as it realtes to the prompt can be seen as a feature attribution problem. The feature attribution problem refers to the setup: given a model f and an input x of finding the subset of features of a model that contribute the most to prediction f(x).

We adapt three explainability methods, Shapley values, LIME, and Saliency maps, all of which are popularly applied to classical machine learning models to interpret the outputs of the stable diffusion model, which is a text-to-image generative model. We then use the images that are generated to reanalyze this problem on a per-image-idea basis.

1.1. LIME Algorithm

The LIME algorithm is a black-box way of understanding which part of the input of a model are able to relate to specific aspects of the output. LIME itself stands for Local interpretable model-agnostic explanations. This is a linear approximation that strives to subtly manipulate the input in a computationally acceptable level, so that it is understandable which movements cause which outputs. This is especially important when attempting to understand why it is that a model is acting a specific way and can be helpful in debugging a model or getting better guarantees on thought process when decisions are critical and not merely accuracy-based. This means that LIME does not need to have access to the training data and essentially creates a mini-data-set of its own with slight perturbations and observation of the output set. (Molnar, 2023) In order to use LIME in the first problem of the paper, we are interested in natural language input. We can treat each of the tokens as a perturbable feature that can be used in order to move in a region space around the original natural language input. LIME can also be used for images which is important in understanding which parts of the image are contributing to an understanding of a specific phrase. This would be far too computationally expensive to alter random pixels and therefore 'superpixels' are used to perturb the input. This is essentially just groups of similar pixels within an image that can be turned to a neutral color that would theoretically and ideally not have a new large meaning on the predicted output. Because linear models of random perturbations can result in overfitting that is not replicated in images that are epsilon away, it is important to understand the imperfection of this technique and compare results with those generated by SHAP or Saliency maps.

1.2. SHAP Algorithm

Shapley values are an explanation method adapted from cooperative game theory to provide a solution to the feature attribution problem. The cooperative game (Shapley et al., 1953) consists of:

- a set of players $D = \{1, ...d\}$
- a coalition $S \subseteq D$ that represents a game by specifying the set of players
- a characteristic function v, that represents the payoff of the given set.

The Shapley value is a technique for allocating credit to players in the cooperative game. For G denoting the set of games on d players, the shapley values are each:

$$\phi : G \mapsto R$$

For a game v, Shapley values are $\phi_1(v), ...\phi_d(v)$

The following equation is used to calculate $\phi_i(v)$:

$\phi_i(v) = \sum_{S \subseteq D \setminus i} \frac{|S|!(d-1-|S|)!}{d!} [v(S \cup \{i\}) - v(S)]$

This calculates the average contribution of each player across all player orderings. $[v(S \cup \{i\}) - v(S)]$ is the marginal contribution of a player to subset S, and $\frac{|S|!(d-1-|S|)!}{d!}$ is the probability of that coalition.

The Shapley values are applied to machine learning by considering model features as players and some quantification of model behavior as the payoff e.g. the prediction or the loss (Lundberg & Lee, 2017a). The Shapley values are used to quantify each feature's impact on a given prediction.

The adaptation of Shapley values to generative models such as Stable Diffusion is explained in section 3.1.2.

1.3. Saliency Maps

Saliency Maps are a form of highlighting the relevant parts of an image for analysis so that we don't have to view just the output of the model, but can understand where it looks. It is worth noting that this is also possible using LIME as described above, so Saliency Maps is an additional perspective that can be used in conjunction to verify both methods are able to identify the same areas. An important distinction with the methods of SHAP and LIME from Saliency Maps is whether the explanability measurement is perturbation based as the former or gradient based of the latter. We can combine many elements for saliency maps so that there is not a huge gap in the simple gradient which can end up quite noisy - this includes Gradient, SmoothGrad, Guided Backprop, Integrated Gradients, and more (Alur & Bastani, 2024a). In terms of evaluating the outlines that are given as an output with this technique, there are various ways of explaining what this truly means. This includes additive (how much does adding the features improve similarity), subtractive (how much does removing the features decrease similarity), perturbation, compactness, etc. (Alur & Bastani, 2024b)

1.4. Diffusion Models

Stable diffusion is a computational framework applied across fields like computer vision, natural language processing, and generative modeling. Inspired by statistical physics and stochastic differential equations, it refines initial noise vectors iteratively to produce high-quality samples with desired characteristics, with less large leaps from traditional generative techniques. Through a series of diffusion steps, it progressively transforms input noise distributions, maintaining key features of data distributions while reducing noise, enabling the generation of realistic and diverse samples with precise attribute control (Rombach et al., 2022).

1.5. CLIP

CLIP (Contrastive Language-Image Pretraining) is a deep learning model from OpenAI that closes the gap between images and text using a transformer-based architecture and contrastive learning (Radford et al., 2021). Unlike conventional models, CLIP learns to link images and their corresponding text descriptions in a shared embedding space during pretraining, eliminating the need for labeled data for image classification. Its pretrained representations enable accurate performance across tasks like image classification and natural language understanding without task-specific fine-tuning, thanks to its generalizability. This makes it especially important for our project in that it is able to capture embeddings for any idea that is represented by a stable diffusion prompt.

1.6. Variational Autoencoders (VAE)

Variational Autoencoders (VAEs) are generative models that compress data into a latent space and then reconstruct it. The embeddings after the encoder can be very useful for tasks of image comparison. We chose not to use this due to constraints on compute and datasets, but not that this is a viable option for similarity that we plan to further explore.

2. EVALUATION METHODS

We evaluate the results of the Shapley and LIME values using two methods that we have developed:

2.0.1. Percentage of Ranks Aligned

The dataset contains importance scores associated with each token of a prompt, as ground truth i.e. we rank the tokens in each prompt in order of importance. This evaluation metric is the simple percentage of ranks determined by the Shapley or LIME value aligning with the ground truth ranks.

$$\text{Percent of Ranks Aligned} = \frac{\text{No. of tokens with correctly assigned ranks}}{\text{No. of tokens in prompt}}$$

Example: calculating the Percent of Ranks Aligned for Shapley values of *a cat's house*:

Tokens in *a cat's house*	a	cat's	house
ground truth score	0.05	0.45	0.5
ground truth rank	3	2	1
Shapley score	0.13	0.59	0.28
Shapley rank	3	1	2
Ranks are aligned	T	F	F

Percent of Ranks Aligned $= \frac{1}{3} = 33.33\%$

2.0.2. Root Mean Squared Error

The Root Mean Squared Error (RMSE) calculates the root mean squared difference between ground truth importance scores and importance scores outputted by the explainability method of each token in the prompt

$$\text{Root Mean Squared Error} = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \hat{x_i})^2}{N}}$$

where x_i is the ground truth importance score, $\hat{x_i}$ is the importance score of the explainability method (normalized to add up to one), and N is the number of tokens.

An example calculating the Root Mean Squared Error for Shapley values of *a cat's house* is shown below:

Tokens in *a cat's house*	a	cat's	house
ground truth score	0.05	0.45	0.5
SHAP score	0.13	0.59	0.28
Squared Error	0.0064	0.02	0.05

$$\text{Root Mean Squared Error} = \sqrt{\frac{0.0064+0.02+0.05}{3}} = 0.2728$$

3. METHODS

3.1. Token to Image Interpretability

The first question that the interpretability algorithms are able to answer is whether or not the given token in the input is going to have a drastic change on the image as a whole. This 'change' for images is a difficult thing to measure in a context in which one must compare two elements of an image to determine which plays a larger role in the general meaning of the input. There are several ways in which these results can be measured and each of these comes with contextual upsides and downsides

- **CLIP similarity** is a way to move both of the images into a smaller latent space that can measure similarity by comparing the embeddings directly. It captures semantic similarities between images, even if visually dissimilar, by leveraging its pretrained knowledge. This has the benefit of being able to generally understand what semantic features of images are important, but it is not related to this task specifically. Still, this remains the best metric.

- **VAE similarity** encodes images into a latent space, where similar images have nearby representations. Similarity is determined by measuring distances between these latent vectors similar to in CLIP, but this task is specifically for prompt to image similarities. While there are several datasets of this, we decided cultivating more specific similarities and the compute required to train the VAE would not have had significant enough improvement over CLIP, so we decided not to implement this technique.

- **Latent Diffusion Embedding similarity** similar to the previous two methods, comparing the latent space. Notably this is within the specific model that is being used and requires open sourcing the model so that the intermediate modes can be seen. Further, this has the major disadvantage of not having similar spaces when similar objects are in different parts of the image (it does not extrapolate or embed to a sufficient degree).

3.1.1. LIME Approach

The LIME approach differs from classical methods in that the output of the model is an image rather than a classification with simple similarity. Further, implementing LIME to be able to use generative models required rewriting much of the algorithm to allow for the needed setup that this problem entailed. The figure below represents some of the differently generated models that can exist from the LIME generation. The main image is generated from the prompt ”A Lamp with flowers”, while the three images below are subsets consisisting from (in order from left to right) ”Lamp with flowers”, ”A with flowers”, ”A lamp with”. In practice, all of the subsets are chosen and weighted based off of how many of the other tokens are included in order to measure the difference between the original picture and the generated pictures. Due to compilation issues, this code must be manually completed in notebooks and cannot be run from the terminal the way other LIME libraries are able to function. This is big on the todo list as we continue this project into the summer!

stablediffusionweb.com

3.1.2. Shapley Values Approach

The Shapley Values approach, similar to LIME, is adapted to generative models by adapting the nature of inputs and outputs. While in classical machine learning models, the model is trained on features and outputs a prediction, stable diffusion models take prompts as inputs and produce images as outputs. We adapt Shapley values (Shapley et al., 1953) to image-to-text generative models by assigning the following as inputs and outputs:

- each token in a prompt is a player e.g. when the prompt is *a cat's house*, the players are *a*, *cat's*, and *house*.
- the complete coalition D is the original prompt e.g. a cat's house, and the coalitions or subsets $S \subseteq D$ of the game are all possible combinations of tokens in the same order e.g. a, cat's, house, a cat's, a cat's house, cat's house, a house. There are $2^d - 1$ subsets of a prompt (since we exclude the empty set).
- the image similarity of a subset of tokens S to the original prompt D is its payoff, or value $v(S)$.

While there are methods to estimate Shapley values for games with larger values of d, this paper employs the original formula to calculate Shapley values (Lundberg & Lee, 2017b):

$$\phi_i(v) = \sum_{S \subseteq D \setminus i} \frac{|S|!(d-1-|S|)!}{d!}[v(S \cup \{i\}) - v(S)]$$

The permutation-based algorithm (Lundberg & Lee, 2017b) is applied, that considers all possible orderings of tokens, and average marginal contributions across them for each token. The marginal contributions are normalized by dividing by the total of marginal contributions of each token to a prompt to get the Shapley value.

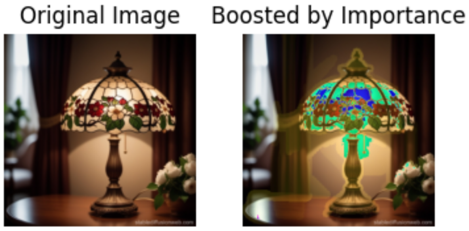
3.2. Image to Generator

Once we have an image generated and a description of the importance of tokens that went into the model as an input, the question still remains of what within the image relates to what tokens within the prompt. A motivating case for this could include wanting to make sure that when the model generates images that generated sections relate to the correct descriptors. In the case of radiology, one would want to know what in the image not only relates to the general prompt, but relates to the specific elements of the prompt. The same would go for many industries where precision about how to model views parts of the image could have large consequences.

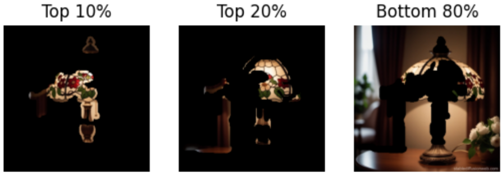
The first step in correlating parts of the image to the desired parts of the prompt is segmenting the input in meaningful ways. We analyze this in two different ways. The first is in terms of what the model is looking at which would make this technique useful in understanding the heirarchy within the models interpretation of the prompt. The second approach is in terms of object segmentation and allows for a more human interpretable approach where the onlooking can quickly recognize what 'things' in the image relate instead of general regions of the image. The two methods are further explained below.

3.2.1. Segmenting using Saliency Maps

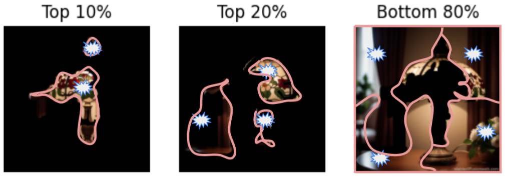
Salency Maps are important in their ability to be able to identify what the object is looking at when it is attempting to compare what is in the image to the prompt. Through doing this at multiple steps in the stable diffusion pipeline there is the ability to see the importance that can progress through how the model is unnoising the image and prioritizing as primary and secondary objects. This is highly representative to interpretability in its own with conjunction on the rest of the project because we are able to see what parts of the image are relating to the specific tokens that are weighted by importance in the earlier section.



Here, we also use this segmentation to decide what the object groups are representing a similar concepts to 'superpixels'. That is, this is a way in which we can treat as a feature in order to decide what has the highest relation to specific tokens.



This segmentation is not enough just as a levels of importance because there are many instances of which multiple objects have similar importance but are completely separate ideas. Therefore, we augment this process with the inclusion of k-means clustering that is able to bring together close elements of the image.



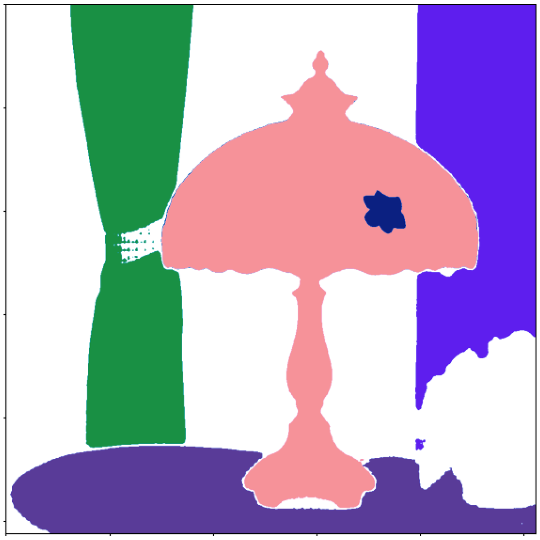
3.2.2. Segmenting using SAM

SAM or the 'Segment Anything Model' is an open-source project that is provided by Meta (**meta˙sam**). The purpose of the project is to provide a solution to the segmentation that is a foundational model opposed to fine-tuned for a specific task. This lends itself to our use as a project that wants to have high success rates for any generated prompt. SAM will take in an image as input and will output pertinent information relating to each of the masks including the mask of pixels that are in the segment, area, bounding box, predicted mask quality, input point used to generate the mask, etc.

The SAM model in our use case is strong in its generality and in its ability to be able to have human interpretation. Consdier the following image that was generated using the prompt "A lamp with flowers"

stablediffusionweb.com

Ranking off of the size and quality of the masks, we can limit the number of objects within the generated maps that we would want to be interpretable by the model. This would be important if we want to be explicit about seeing if the model is misunderstanding the ground as an important factor in a models prediction. Consdier the following mask generated by SAM as follows



After iterating through all possible combinations of the generated subsets from parts of the image, we compare the CLIP embedding of the prompt and words within it relative to the parts of the image. Through doing so we are able to identify what best relates. Note that because we are now using subsets on both the input prompt and on the mask potential options, there is a much greater computational challenge especially given the compute required to generate embeddings. However, because most images can be capped at no more than 10 objects depending on the use case, this can be a feasible option in certain domains.

4. DATASET

We employ a toy dataset consisting of 10 prompts. For the purpose of simplicity, each prompt consists of three or four tokens. We manually assigned importance scores to each token in each prompt as the ground truth. The importance scores of each prompt add up to 1.

prompt	ground truth scores			
Token	1	2	3	4
a flower lamp	0.1	0.4	0.5	0
a lamp with flowers	0.02	0.5	0.08	0.4
a dog wearing hat	0.05	0.45	0.2	0.3
a Spanish flag	0.05	0.4	0.55	0
books and coffee	0.5	0.1	0.4	0
Girls watching movies	0.35	0.25	0.4	0
a cat's house	0.05	0.45	0.5	0
a map without land	0.05	0.4	0.25	0.3
anc. egyptian airpods	0.15	0.45	0.4	0
astronauts at parties	0.55	0.05	0.4	0

5. RESULTS

5.1. Shapley Values

We applied the adapted version of the Shapley Value to the dataset above and evaluated the results using the Percent of Ranks aligned and Root Mean Squared Error metrics. Of the ten prompts, five had perfect alignment of ranks of Shapley values of each token in the prompt with the ground truth ranks, and those five prompts also had low RMSE scores. The remaining five prompts, when examined closely, seem to contain more semantic ambiguity and abstractness in the meaning of the prompt.

5.1.1. Percent of Ranks Aligned

Prompt	Ranks obtained from Shapley values			
	token 1	token 2	token 3	token 4
a flower lamp	3	2	1	-
a lamp with flowers	3	1	2	4
a dog wearing hat	4	1	3	2
a Spanish flag	4	2	3	-
books and coffee	1	3	2	-
Girls watching movies	2	3	1	-
a cat's house	3	1	2	-
a map without land	3	1	2	4
ancient egyptian airpods	1	2	3	-
astronauts at parties	1	2	3	-

On average, 64% of ranks of all tokens according to the SHAP importance scores aligned with the true (ground truth) ranks. For five out of ten prompts, the Shapley values were in perfect order. The remaining tokens can be noted to be either semantically ambiguous or rather abstract in nature: a cat's house, a lamp with flowers, a map without land, ancient Egyptian airpods and astronauts at parties.

5.1.2. Root Mean Squared Error

Prompt	Squared errors of each prompt				RMSE of prompt
	tok 1	tok 2	tok 3	tok 4	
a flower lamp	0.0004	0.0025	0.0009	-	0.06
a lamp with flowers	0.0361	0.01	0.0169	0.0484	0.25
a dog wearing hat	0.0036	0.0025	0.0009	0.0016	0.08
a Spanish flag	0.0441	0.0064	0.0169	-	0.26
books and coffee	0.0009	1E-04	0.0004	-	0.04
Girls watching movies	0.0001	0.0036	0.0025	-	0.08
a cat's house	0.0064	0.0196	0.0484	-	0.27
a map without land	0.0001	0.0784	0.0025	0.0484	0.28
anc. egyptian airpods	0.0961	0.0049	0.0576	-	0.40
astronauts at parties	0.0484	0.0169	0.1225	-	0.43
				Avg RMSE	0.2161

The root mean squared error of the prompts in the given dataset range from 0.04 for *books and coffee* to 0.28 for *a map without land*. The average RMSE is 0.22. This metric penalizes large differences in values between the ground truth and the evaluated values, as seen in the large RMSE for *astronauts at parties* due to the token *parties*.

It must be noted that the prompts that have lower percentages of ranks aligned also have high RMSE: namely, *a cat's house, a lamp with flowers, a map without land, ancient Egyptian airpods* and *astronauts at parties*.

5.1.3. Visual Analysis

In this section, we will investigate one prompt with low RMSE, *books and coffee*, as well as one with a high RMSE, *astronauts at parties.*

- books and coffee

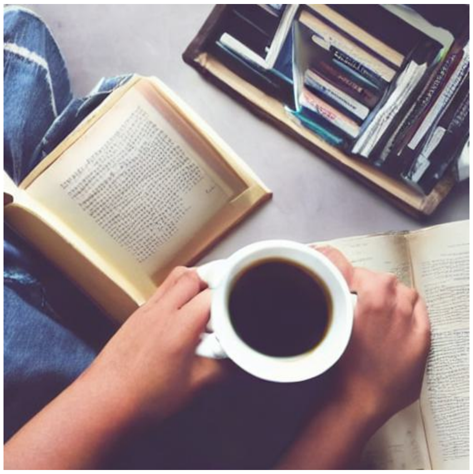


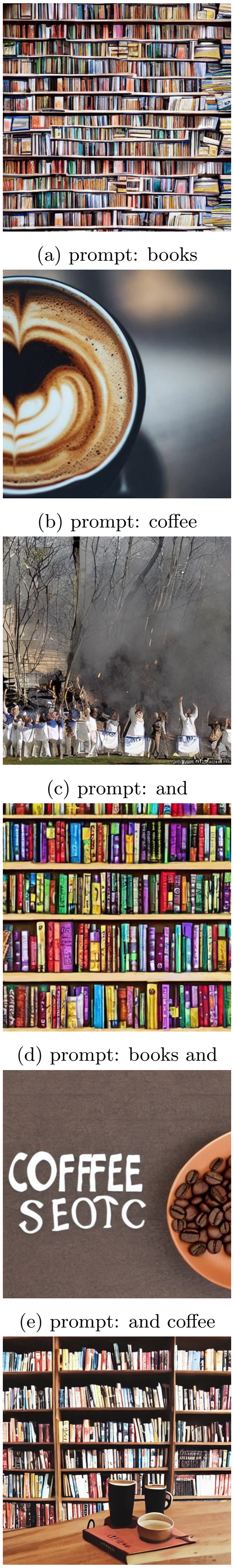
Figure 1: Image generated from prompt *books and coffee*

The image generated had a low Root Mean Squared Error of 0.04 and perfect alignment of ranks with the ground truth:

Token	books	and	coffee
ground truth score	0.50	0.10	0.40
ground truth rank	1	3	2
SHAP score	0.47	0.11	0.42
SHAP rank	1	3	2
Squared Error	0.00	0.00	0.00

Books are the most important 'player' in the image generated originally (Figure 1), as per its Shapley value.

The Shapley values for the tokens in *books, and*, and *images*, were calculated by using the images generated from subsets of the original prompt, as shown below in Figure 2. The clear presence of the coffee cup and a stack of books in both the originally prompted image and the images prompted of the subsets containing *coffee* and *books* respectively may have led to the alignment in importance scores of each tokens *coffee* and *books.* The word *and* is somewhat meaningless on its own, and the subsets containing it are not too affected by its presence either, leading to a low Shapley value. For example, the images prompted from *books* and *books and* are similar to each other, and so are the images prompted from *coffee* and *coffee and.*



(a) prompt: books

(b) prompt: coffee

(c) prompt: and

(d) prompt: books and

(e) prompt: and coffee

(f) prompt: books coffee

- astronauts at parties

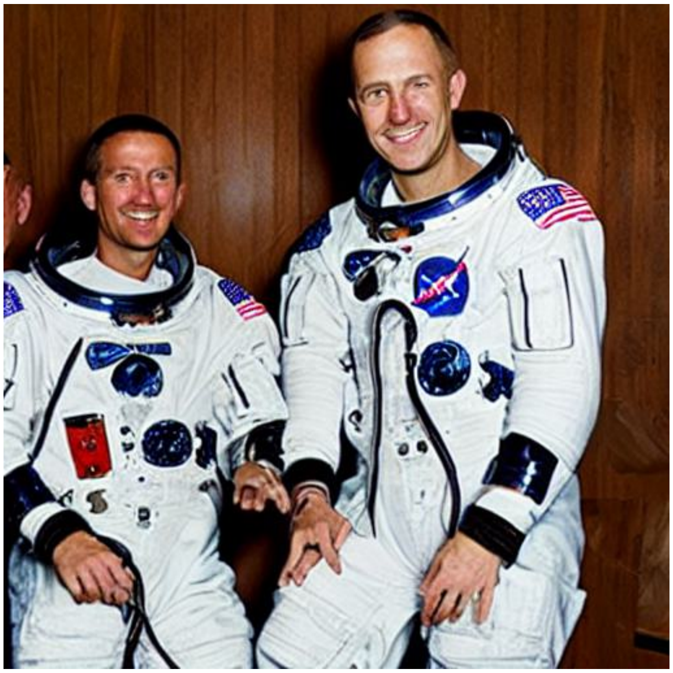


Figure 3: Image generated from prompt *astronauts at parties*

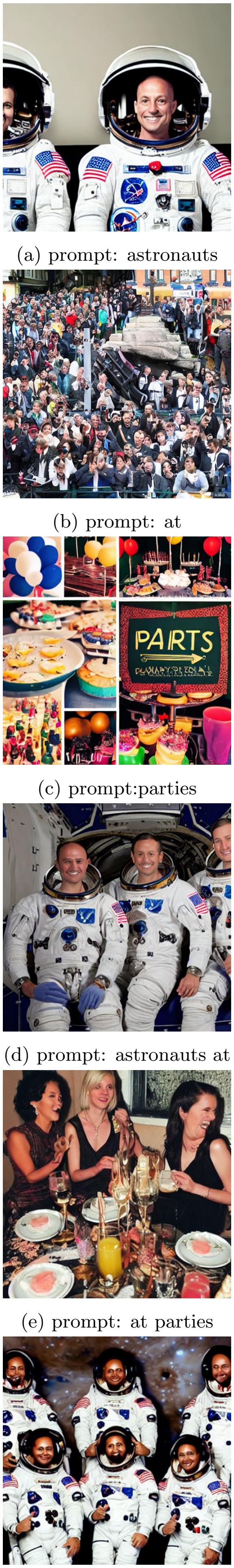
The image generated had a high RMSE score of 0.43 and only 33% alignment of ranks with the ground truth:

Token	astronauts	at	parties
ground truth score	0.55	0.05	0.4
ground truth rank	1	3	2
SHAP score	0.77	0.18	0.05
SHAP rank	1	2	3
Squared Error	2.9584	0	0

Astronauts are the most important 'player' in the image generated originally (Figure 3), as per its Shapley value.

The Shapley values for the tokens in *astronauts, at*, and *parties*, were calculated by using the images generated from subsets of the original prompt, as shown below in Figures 3 and 4. The clear presence of the astronauts in both the originally prompted image and the images prompted of the subsets containing astronauts may have led to the alignment in the token *astronaut*'s importance scores.

The word *parties* has the smallest SHAP score, despite its ground truth ranking score being 0.4, which means that the researcher scoring the importance of the token thought that the token *parties* was 40% importance to the prompt. Figure 3, which depicts the image generated from *astronauts at parties* does not indicate much of a party to the viewer, which may contribute to the Shapley value of the token *parties* being so low. It is interesting that when prompted with only *parties*, the model generates an image of balloons and cake and other items associated with parties, but the image prompted from *astronauts at parties* does not include such items.



(a) prompt: astronauts

(b) prompt: at

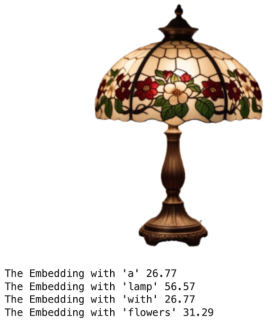
(c) prompt:parties

(d) prompt: astronauts at

(e) prompt: at parties

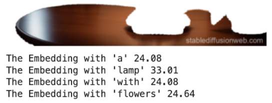
(f) prompt: astronauts parties

Next, consider some of the results that are generated from the input prompt of ”A Lamp with Flowers”. These results are generated using the SAM segmentation approach. As you can see below, we have the correct expected behavior regarding the lamp itself having appropriate higher correlations on the first segmentation, similar to the flowers that are within the design. Appropriately, the other two objects have a neutral association with the prompts due to their lack of importance regarding the prompt.



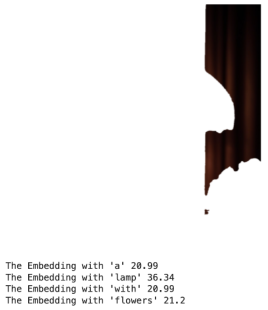
```
The Embedding with 'a' 26.77
The Embedding with 'lamp' 56.57
The Embedding with 'with' 26.77
The Embedding with 'flowers' 31.29
```

(a) Lamp Object



```
The Embedding with 'a' 24.08
The Embedding with 'lamp' 33.01
The Embedding with 'with' 24.08
The Embedding with 'flowers' 24.64
```

(b) Tabletop Object



```
The Embedding with 'a' 20.99
The Embedding with 'lamp' 36.34
The Embedding with 'with' 20.99
The Embedding with 'flowers' 21.2
```

(c) Backrgound Object

6. NEXT STEPS

Our appraoch to explaining stable diffusion text-to-image outputs involves comparing images using CLIP and so far, we have only applied the methods to prompts consisting of less than four tokens. Our next steps involve testing our methods to investigate their robustness to other image similarity methods and larger prompts.

6.0.1. Shapley values

We would like to investigate whether all fairness axioms of Shapley values are met in this adaptation of Shapley values, namely efficiency, symmetry, null player, and linearity.

We would also want to adapt our approach to estimating Shapley values of longer prompts by applying Monte carlo estimation. This is the industry practice for estimating Shapley values of models with a large number of features since the calculation of Shapley values involved summarizing across 2^{d-1} subsets where d is the number of features.

6.0.2. Applying VAEs as an alternative to CLIP

We would also want to explore capturing image similarity by applying variational auto encoders (VAEs) as opposed to CLIP.

]

REFERENCES

Alur & Bastani. (2024a). Trustworthy machine learning course slides 20a. https://www.seas.upenn.edu/~obastani/cis7000/spring2024/schedule.html

Alur & Bastani. (2024b). Trustworthy machine learning course slides 20b. https://www.seas.upenn.edu/~obastani/cis7000/spring2024/schedule.html

Lundberg, S. M., & Lee, S.-I. (2017a). A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems 30* (pp. 4765–4774). Curran Associates, Inc. http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf

Lundberg, S. M., & Lee, S.-I. (2017b). A unified approach to interpreting model predictions. *Advances in neural information processing systems*, *30*.

Molnar, C. (2023). Interpretable machine learning - a guide for making black box models explainable. https://christophm.github.io/interpretable-ml-book/lime.html

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

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10684–10695.

Shapley, L. S., et al. (1953). A value for n-person games.