DenseDiffusion Analysis and Hypertuning

Athulith Paraselli

# 

# Abstract

Modern text-to-image models struggle with capturing fine details when given dense input prompts. To conduct further research in this issue we analyze the DenseDiffusion training free framework proposed by CMU and NAIVER AI. By generating various images using consistent dense prompts we were able to explore how we can better apply DenseDiffusion. We focused on three key hyperparameters: wc (cross-attention modulation), ws (self-attention modulation), and m (mask-area adaptive adjustment). Using Stable Diffusion v1.5, trained on the LAION dataset, we conducted a series of experiments to evaluate the impact of these hyperparameters on image quality and fidelity. Evaluation metrics included CLIP score, SOA-I score, and human preference rankings. Our findings reveal potential optimal ranges for the hyperparameters and demonstrate that fine-tuning these parameters enhances the model's ability to generate high-quality, text-aligned images. Code and data available at: [INSERT LINK]

# 1. Introduction

Diffusion models [[1](https://arxiv.org/pdf/2006.11239)] have been largely researched in recent years. These models add random noise to existing data and then are trained to remove the noise to return to the original image iteratively. Using this process they can transform a random assortment of noise into new synthetic data. One common use of diffusion models is in text-to-image models [[2](https://arxiv.org/pdf/1511.02793)].

Modern text-to-image models commonly include two key components, the first is a text encoder, which stores and encodes the features of the text. The second component is an image generator which will take in the encoded text and output a synthetic image. The generation step is usually where diffusion models are applied. Diffusion models have advantages over alternatives such as GAN [[3](https://papers.nips.cc/paper_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf),[4](https://arxiv.org/abs/1605.05396)] such as offering fine-grained control over the generating process. DenseDiffusion [[5](https://arxiv.org/pdf/2308.12964)] is a training-free method that can be adapted to diffusion-based text-to-image models to handle cases when inputted prompts are “dense”. A dense prompt includes significant details, such as multiple descriptors of the subject and many specifications. The method highlighted in Naver AI Lab’s paper builds upon Stable Diffusion [[6](https://arxiv.org/abs/2112.10752)] and modulates the model’s attention maps according to both text and layout conditions.

In Stable Diffusion, there are two sets of layers known as the self-attention layers and the cross-attention layers. Generally speaking, self-attention focuses on relationships within the same sequence, in Stable Diffusion, it allows the communication between intermediate and context features to create globally coherent structures. Cross attention is conditional on textual features and largely considers the relationship between the textual features and the to-be-outputted image features (aka the intermediate image features). DenseDiffusion attempts to modulate both of these layers. In the self-attention layers, DenseDiffusion limits the interaction between intermediate features which represent distinct separate objects. Every pair of intermediate features has an attention score, and the attention score of features in the same object is boosted, whereas the attention scores of features that represent separate objects are decreased. In the cross-attention layers, DenseDiffusion congregates certain text features to specific regions based on the corresponding layout condition.

The degree to how much we should modulate the different attention layers is one which we aimed to answer without experiments. By testing and analyzing the results from various modulations we attempted to conclude how to best apply the DenseDiffusion framework to Stable Diffusion.

# 

# 2. Methods

Our goal was to increase the performance of the DenseDiffusion application used in the paper by adjusting the hyperparameters. Through this process, we could analyze trends within the changes of hyperparameters in order to learn more about how DenseDiffusion could be applied to future diffusion models and how strong the layers should be modulated. We tested various hyperparameters and then inputted the results through evaluation metrics to quantify our improvement.

### 2.1 Hyperparameters

DenseDiffusion has three hyperparameters that can be adjusted. The first is labeled as wc, which represents the degree of modulation in the cross-attention layers. As mentioned earlier, the cross-attention layer modulation enforces the positioning of objects based on the layout conditions. The next hyper-parameter is ws, which is the degree of modulation in the self-attention layers, which will increase the faithfulness to the inputted text however it has the cost of potentially decreasing the quality of the photo. Finally, there is m which is the degree of mask-area adaptive adjustments. This parameter changes the size of segments to make them more similar, which in turn helps to preserve the image quality.

### 2.2 Implementation details

The model that was used in our tests was Stable Diffusion v1.5 which was trained on the LAION dataset [[7](https://arxiv.org/pdf/2210.08402)]. Using the framework set up by CMU and NAVER AI, we generated various images while tweaking the hyperparameters. The generation of the photos was completed using a Nvidia RTX 3060M. In our experiment we set a baseline for which future images would be compared, these values were wc = 1, ws = 0.3, m = 1.

### 2.3 Evaluation

Throughout the experiment, the evaluation of images was done using three different metrics. The first evaluation method was the CLIP score [[8](https://arxiv.org/pdf/2104.08718)] which compares the image and text embeddings and checks their cosine similarity. The next metric was the SOA-I score [[9](https://ieeexplore.ieee.org/document/9184960)], which parses for objects within the sentence and then checks for their existence in the reference image. In this sense, SOA-I accomplishes a similar objective as recall. The final metric was human evaluation scores. For this metric, we presented humans with sets of images and asked them to rank them relative to each other to determine which images were preferred by humans.

# 3. Experiment

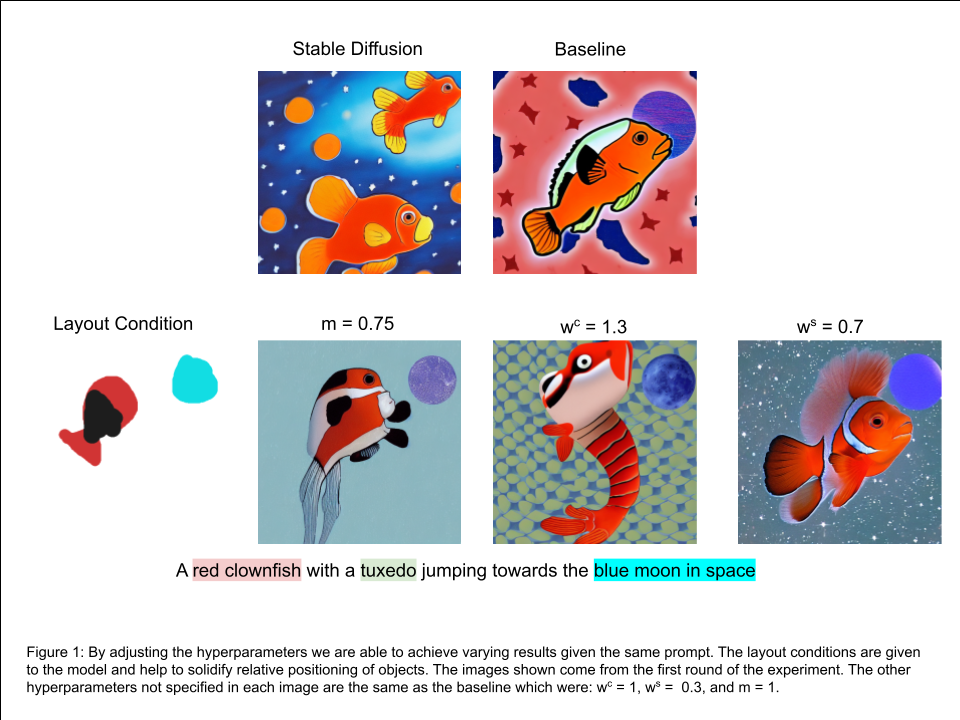
### 3.1 Format

The experiment we conducted aimed to gather data about the performance of prompts relative to different hyperparameters and determine which hyperparameter outputted the best results generally. To achieve this goal we designed an experiment which was split into three rounds. The first round consisted of a large search. For each other parameters, we tested 4 different values which spanned large ranges. Each time we tweaked one hyperparameter we kept the other 2 the same as the baseline mentioned before. For wc we tested the values 0.4, 0.8, 1, and 1.3. For ws we tested the values 0.1, 0.3, 0.5, and 0.7. Finally, for m we tested the values 0.25, 0.50, 0.75, and 1.00. Each prompt that we used for testing followed the following format: “A descriptive subject with an extra description of the subject doing something at a specific location”. We ensured all prompts followed a format similar to this to ensure that we were tuning the model to work well with dense prompts. We also ran base stable diffusion to see if the dense diffusion results were indeed performing better.

In the second round, we attempted to narrow our search for each of the hyperparameters. We once again changed only one hyperparameter at a time and kept the other two the same as the baseline. In addition to changing and doing a more in-depth search of the hyperparameters, we also generated a photo using the best hyperparameters for each prompt in round one to verify that introducing the best hyperparameters together would lead to better results. The hyperparameters that we tested in round two for wc were 0.9 and 1.15. For ws we tested the values 0.4 and 0.6. Finally, for m we tested 0.33 and 0.41. These values were decided using our results from round one. Additionally, for the second round, we introduced another prompt to verify that the results we were looking at generalized well for various prompts.

For the third and final round, we chose two hyperparameter values which we considered to be optimal, and then various permutations of the hyperparameters we tried with each other at random to evaluate which results values returned the best results. In addition for each of the prompts, we collected data about human preferences. We asked subjects the generate images based on their preference. In addition to the image included in the third round, we also included the image created by the baseline parameter and the image generated by base stable diffusion.

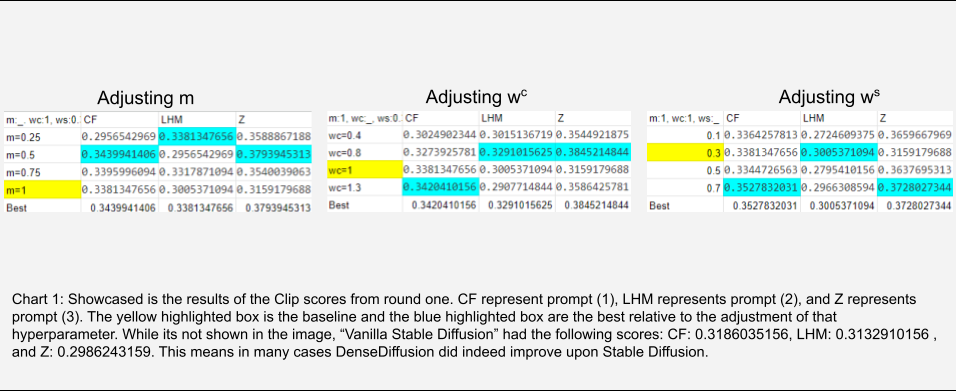
### 3.2 Results



During the first round, we conducted a large search for each of the hyperparameters using the following prompts:

1. A red clownfish with a tuxedo jumping towards the blue moon in space
2. A man with long hair wearing a jacket singing in a high school classroom
3. A zebra with wings on a skateboard skating at a commercial gym

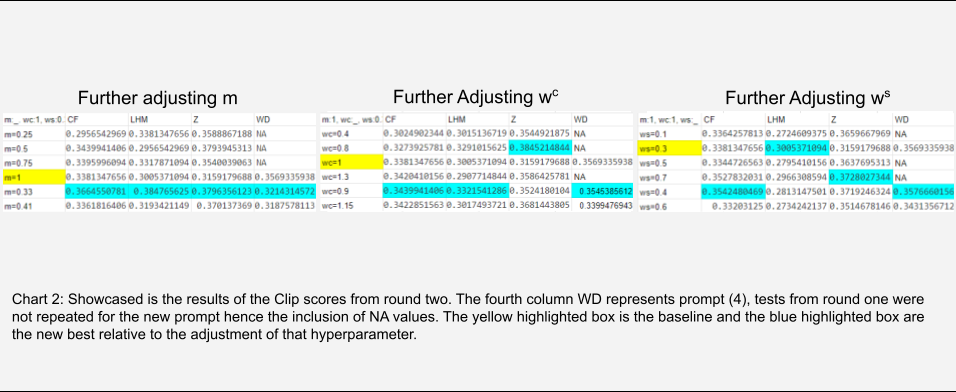
We found that the best values for m lay between 0.25 - 0.5, for wc we observed values between 0.8 - 1.3 and for ws we saw values 0.3 - 0.7. Additionally, the results we observed were consistent with the modulations that occur with DenseDiffusion. As we observed greater values for wc, we saw that the images mapped more closely to the given layout conditions at the cost of looking unrealistic. Additionally, as we increased ws we saw that the images contained the objects listed in the sentence more often. To ensure that the different parameters truly resulted in different results and clip scores we ran an ANOVA across the different parameters tested to see if they came from a different distribution. Our results showed a p-value of 0.0 which implies that changing the parameters did result in significantly different clip scores. It is worth noting though that since we only tested three prompts, our sample size is small leading to an unreliable ANOVA.



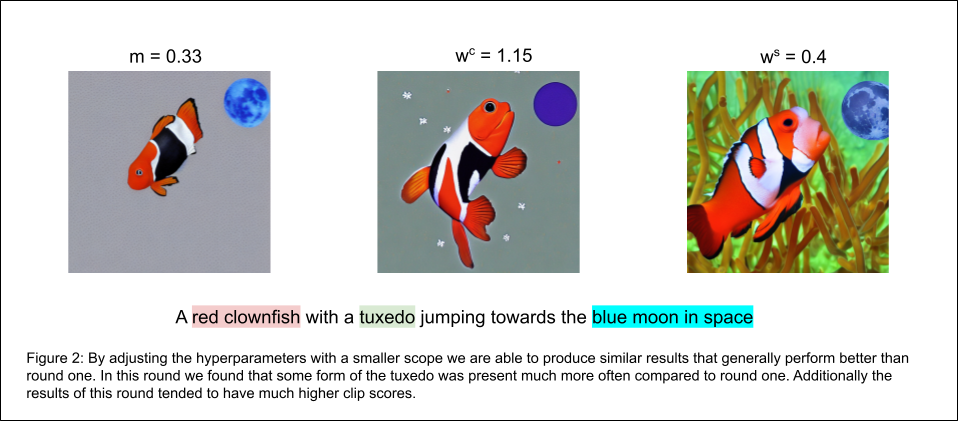
For the second round, we narrowed our search and tested two values for each other hyperparameters. The values were determined by creating a range that was dependent on the best values observed for each of the hyperparameters. For example, when looking at the hyperparameter m, we observed the best results were either 0.25 or 0.5 so our range was between [0.25 - 0.5]. The two values chosen were the two midpoints of the range, so for m it was 0.33 and 0.41. Additionally, we introduced a fourth prompt, which was:

1. A white fluffy dog wearing a pink bowtie and stealing a piece of bread at the park

We introduced this prompt to see if our results generalized well. We found that overall the results in this round were much better with higher clip scores on average. Additionally, for many of the prompts, many of the specified details were present more often.



As Chart 2 shows, in round two we observed relatively consistent results among all the prompts. For m we saw one value perform better than all the others. Additionally, for wc and ws, we saw that the two values which performed the best across the categories for each of the two hyperparameters were adjacent to one another which allowed us to create a new range for round three’s values. Once again we ran an ANOVA and observed that the various hyperparameters did indeed return results that stem from separate distributions, which implied that changing the hyperparameter by the smaller amounts still led to significantly different results.



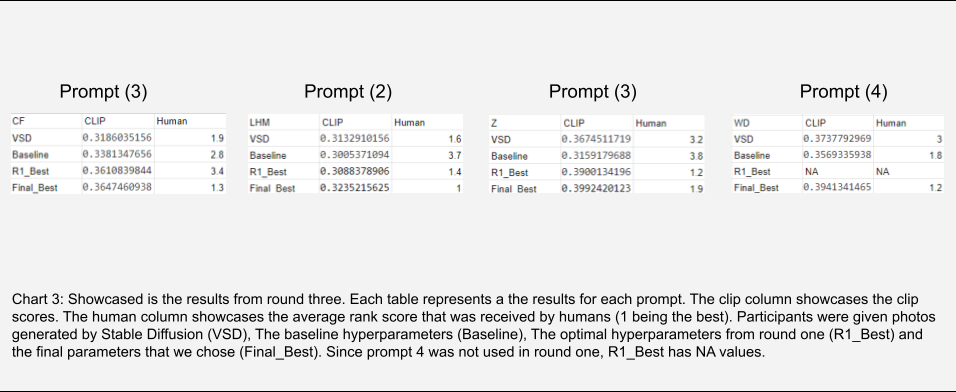
In the third and final round, we used the results from round 2 to form our final images. We chose a value at random for each of the parameters from the ranges that round two implied were optimal. The range that was defined as optimal from round two was the lower value of the best clip scores from one of the four prompts, and the largest value. The ranges were:

ws: 0.3 - 0.4

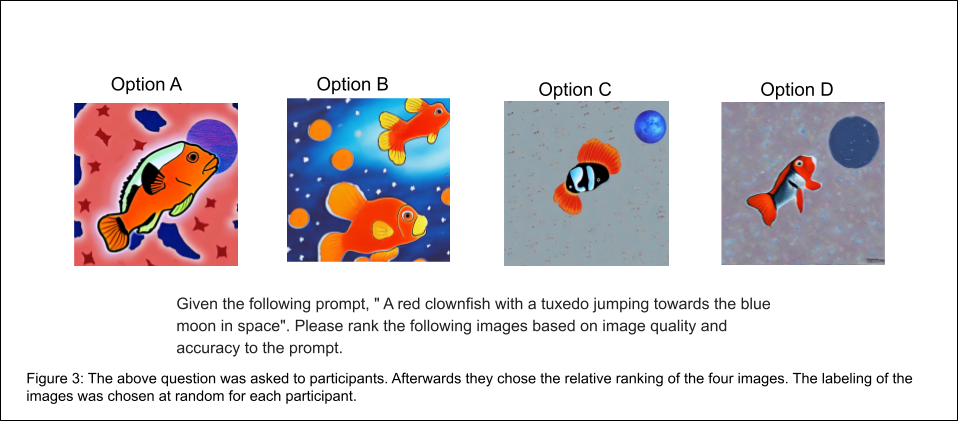
wc: 0.8 - 0.9

m: 0.25 - 0.33

The final images were generated using ws = 0.36, wc = 0.88, and m = 0.31. After generating the images we had 10 participants rank the images in order of preference given the context of the prompt. The images chosen for participants to rank were base stable diffusion, baseline, best parameters from round 1, and the final images generated in round 3. By using these four images we had a wide range of values to be tested and the base stable diffusion acted as our “null” sample.



As chart 3 shows, we observed across all four prompts that humans tended to prefer the final hyper parameters that we choose. Additionally human preference did indeed tend to follow clip scores. However it is also worth noting that for base stable diffusion we noticed that it performed much better in regard to human preference than it did relative to its clip score. One reason that we propose this could have occurred is that DenseDiffusion tends to produce its images to align with the layout condition specified by the user (in other words us). This means that depending on the subjects preferences they might have preferred stable diffusion’s location choices more than the layout conditions that we provided to the model.



### 3.3 Limitations and Further Work

The experiment we conducted was heavily bottlenecked by the equipment we used to generate the images. Due to a singular GPU and low VRAM, we were only able to generate one image at a time and each image took approximately 10-20 minutes to generate. With better equipment, we would have tested more prompts in rounds one and two (approximately 10 in round one and 15 in round two). Additionally, in our experiment, we only generated one image for each of the hyperparameter values tested. This is problematic because image generation using stable diffusion has a significant amount of randomness which must be addressed by looking at variance between samples, which we were unable to do within a reasonable amount of time given the equipment we had access to. We believe that in the future further experimentation could be done to test the generalizability of our results. Additionally, with more generated data, further analysis could be conducted to observe relationships between tweaking the self-attention and cross-attention layers relative to each other.

# 4. Conclusion

In this work, we were able to explore the abilities and limitations of DenseDiffusion when applied to Stable Diffusion. Our findings show that independent of the dense prompt given, certain values for the hyperfeatures tend to show better performance in both clip score and human preference. Notably by adjusting the hyperparameters, we can see a larger amount of the specified objects given in the text prompt being reflected in the generated images. The ranges for the values of the hyperparameters that we have found success with are:

ws: 0.3 - 0.4

Wc: 0.8 - 0.9

m: 0.25 - 0.33

Through our data collection and findings, we observed that the modulation of attention layers leads to significantly different outputs when given the same prompt. Our results suggest that careful tuning of the self-attention and cross-attention modulations, as well as adaptive adjustments, can significantly improve the quality of generated images. This indicates that DenseDiffusion can be a valuable tool in the development of more advanced and nuanced text-to-image generation frameworks.

However, our work had heavy limitations due to computational power. Future research could benefit from more extensive testing with a larger variety of prompts and multiple image generations per setting to account for the inherent randomness in the diffusion process. Additionally, exploring the interactions between different hyperparameters in more depth could provide further insights into DenseDiffusion on future models.

In conclusion, our study confirms that DenseDiffusion, when properly tuned, enhances the performance of Stable Diffusion in generating high-quality, text-aligned images from dense prompts. This opens up new avenues for refining and applying diffusion-based models in various fields requiring precise and detailed image synthesis from textual descriptions.

# References

[1] Ho, Jonathan, et al. Denoising Diffusion Probabilistic Models. 19 June 2020.

[2] Mansimov, Elman, et al. Published as a Conference Paper at ICLR 2016 GENERATING IMAGES from CAPTIONS with ATTENTION. 29 Feb. 2016.

[3] Goodfellow, Ian, et al. Generative Adversarial Nets. 2014.

[4] Reed, Scott, et al. “Generative Adversarial Text to Image Synthesis.” ArXiv.org, 2016, [arxiv.org/abs/1605.05396](http://arxiv.org/abs/1605.05396).

[5] Kim, Yunji, et al. “Dense Text-To-Image Generation with Attention Modulation.” International Conference on Computer Vision, 1 Oct. 2023, <https://doi.org/10.1109/iccv51070.2023.00708>.

[6] Rombach, Robin, et al. “High-Resolution Image Synthesis with Latent Diffusion Models.” ArXiv:2112.10752 [Cs], 13 Apr. 2022, [arxiv.org/abs/2112.10752](http://arxiv.org/abs/2112.10752).

[7] Schuhmann, Christoph, et al. LAION-5B: An Open Large-Scale Dataset for Training next Generation Image-Text Models. 16 Oct. 2022.

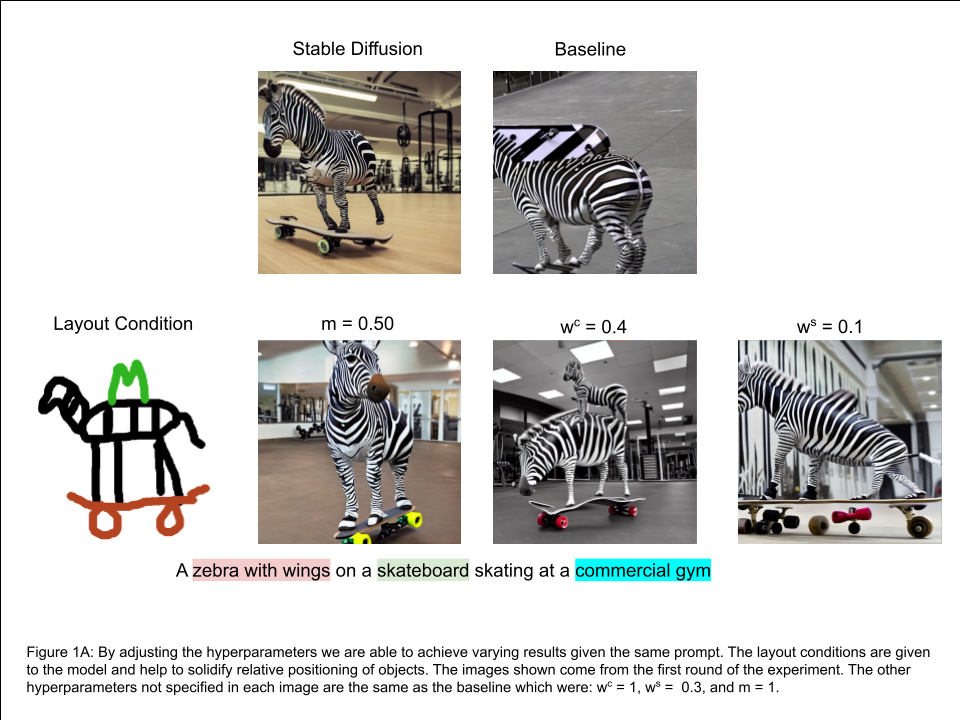
[8] Hessel, Jack, et al. CLIPScore: A Reference-Free Evaluation Metric for Image Captioning. 23 Mar. 2022.

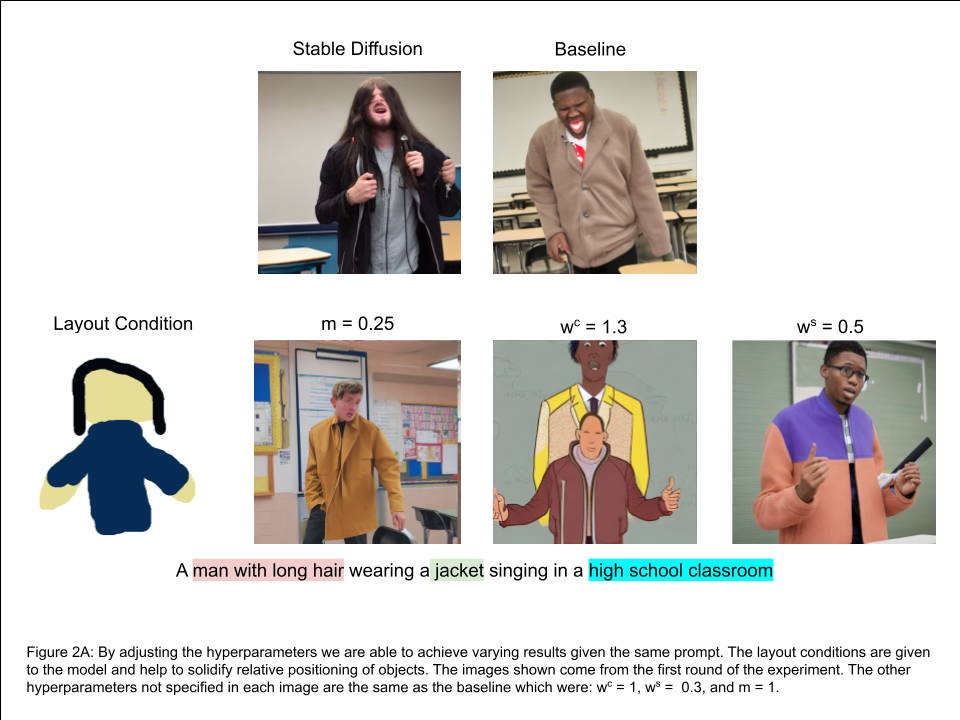
[9] Hinz, Tobias, et al. “Semantic Object Accuracy for Generative Text-To-Image Synthesis.” IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, pp. 1–1, https://doi.org/10.1109/tpami.2020.3021209. Accessed 10 Nov. 2020.

# 

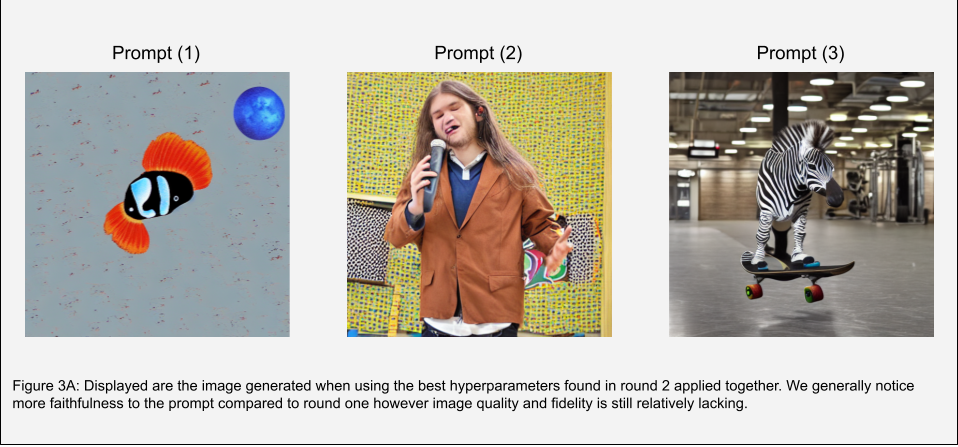
# A. Additional Comparison

### A1. Round one images





### A2. Round two images



### A3. Round three images

