Training CoCo: Continuity and Consistency in Subject-Driven Diffusion Models

Eric Lee
Stanford University
ericlee7@stanford.edu

Abstract

While diffusion models have achieved impressive results in text-to-image generation, maintaining visual consistency of a subject across multiple images remains a significant challenge. This project investigates methods to improve identity preservation in generative pipelines by systematically evaluating combinations of fine-tuning and inference-time techniques. We introduce a pipeline that takes up to five reference images, a name, and a brief description of a subject to produce a sequence of images aligned with input sentences. The best performing pipeline combines a Stable Diffusion baseline model with DreamBooth and style-based LoRA fine-tuning, with token substitution at evaluation time. Human evaluations and quantitative metrics (CLIP similarity and LPIPS distance) show that this composite model generates high quality outputs with strong subject similarity and output consistency. Discussion includes failure cases related to overfitting and visual ambiguity, suggesting directions for future work on disentangling style from identity in generative models.

1. Introduction

Modern generative models have made significant progress in producing highly realistic images from text prompts. However, a persistent challenge remains: generating consistent depictions of the same character across multiple images. Even when given identical or similar prompts, diffusion models often produce visually distinct outputs, making it difficult to preserve a subject's identity across different scenes and contexts.

This inconsistency presents a major limitation for applications that depend on coherent visual narratives, such as illustrated storytelling, personalized gaming avatars, or brand representation. Solving this problem could unlock a range of creative possibilities, from enabling children's authors to illustrate entire stories with a consistent protagonist to allowing gamers to bring personalized avatars to life.

Although many methods have been developed to improve identity preservation in diffusion models, the interactions between these approaches is rarely explored. To address this gap, this project aims to develop a model pipeline that, during training, takes as input up to five reference images of a subject, along with the subject's name and a brief description (fewer than 25 words). At inference time, the pipeline receives a series of sentences that reference the subject by name and generates one 512 $\times$ 512 image per sentence. Each generated image should depict a character that is visually consistent with the reference photos while also aligning semantically with the content of the corresponding input sentence.

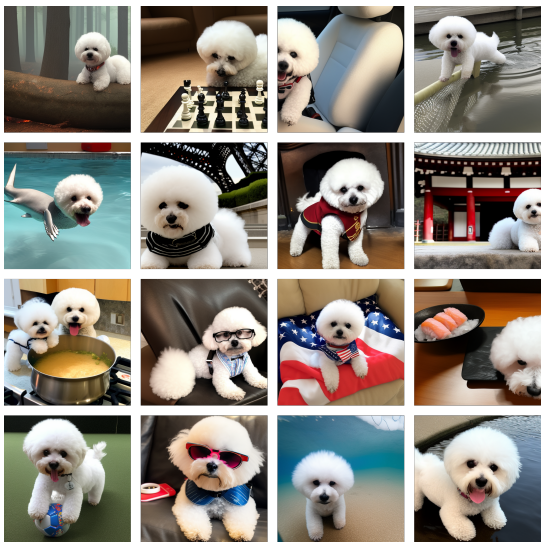

Figure 1. Example images generated by this pipeline, with stable diffusion, DreamBooth, LoRA, and token substitution.

After systematically exploring combinations of techniques, the best pipeline fulfilling this objective is a Stable Diffusion model fine-tuned with DreamBooth and a style-based LoRA, with data augmentation during training and token substitution at evaluation.

2. Related Work

2.1. Diffusion Models

Early generative models for image synthesis, such as Generative Adversarial Networks (GANs) [1], demonstrated impressive image generation capabilities but suffered from notable limitations, including training instability and mode collapse. GANs involve training a generator and a discriminator in a minimax game: the generator tries to produce realistic images to fool the discriminator, while the discriminator learns to distinguish real images from fake ones. While effective in many domains, balancing this adversarial setup proved challenging, motivating the search for more stable generative methods.

Denoising Diffusion Probabilistic Models (DDPMs) [2] offered a breakthrough in this regard by framing image generation as a two-stage process: a forward diffusion process gradually adds Gaussian noise to an image over many steps, and a learned reverse process denoises the noisy input step-by-step to reconstruct the image. At evaluation time, simply using the reverse process yields an effective generative model. Although initially used for unconditional generation, DDPMs provided a flexible and robust foundation that was later extended to conditional settings with text prompts.

Text-to-image generation became feasible with models such as DALL-E [3] and Stable Diffusion [4]. DALL-E employs a discrete VAE for image representation and a transformer-based decoder conditioned on text embeddings. In contrast, Stable Diffusion adopts a latent diffusion approach that operates in a lower-dimensional latent space learned by a variational autoencoder (VAE), enabling efficient computation. Its core architecture uses a U-Net with cross-attention layers conditioned on text embeddings from a CLIP encoder.

Due to the proprietary nature and limited accessibility of DALL-E, we focus on the open-source Stable Diffusion model in this work. Its extensibility and compatibility with fine-tuning techniques such as DreamBooth and LoRA make it a practical foundation for exploring identity consistency in text-to-image generation.

2.2. Fine-Tuning

As diffusion models became increasingly viable for image generation, much work went into fine-tuning foundation models to achieve personalization with minimal computation power. These fine-tuning methods entail additional training on a set of reference examples, with some subset of the parameters being adjusted while the others remain frozen. The fundamental trade-off here is between additional shifting of the weights for more personalization and the computational strain of modifying many parameters.

One method that made strides in fine-tuning with very little compute is Low Rank Adaptation (LoRA) [5], which freezes the model weights and instead inserts learnable low-rank matrices. While LoRA was found to be effective in applications such as style transfer, its minimal changes to the foundation model implied less power in strong personalization to the reference image.

As an alternative to LoRA, DreamBooth [6] performs full fine-tuning of the model weights, using a small set of images tied to a unique text identifier. By associating a rare token with a particular subject (e.g., a person or object), DreamBooth enables the model to internalize specific visual features of that subject and reproduce them in novel contexts. This method enables more faithful identity preservation than LoRA but comes at the cost of greater computational overhead and risk of overfitting.

While DreamBooth and LoRA focus on modifying the model to internalize new identities or styles, there have been additional models like ControlNet [7] which take a different approach. Rather than fine-tuning for personalization, it augments diffusion models with structural guidance (e.g., poses, edge maps, or depth). ControlNet operates by adding a parallel trainable branch to the existing model, enabling spatial or compositional control over the generated output.

2.3. Character Consistency Approaches

As fine-tuning methods have advanced, there has been an increased demand for character consistency across image generations beyond the performance of existing methods. New approaches include StoryMaker [8], a tuning-free pipeline designed to generate coherent characters across multiple images in a narrative setting. This framework focuses on preserving clothing, hairstyle, and body structure across scenes. It accomplishes this by conditioning on facial features and full-body character crops, using a Perceiver Resampler (PPR) to extract character representations. Another modern model is The Chosen One [9], which uses iterative procedures to extract identity features from the reference images.

While these approaches yield successful results, they are more computationally expensive and complex than the optimized fine-tuning methods mentioned above. In this project, we seek to see if we can recreate the same level of character consistency with a simpler pipeline, leveraging the well-established fine-tuning methods that have widespread usage and greater general support.

3. Data

This project uses the Stable Diffusion v2 model [10] from StabilityAI (accessed via Hugging Face) as the baseline. This model is a latent diffusion model composed of a U-Net backbone [11] with cross-attention layers, operating in the latent space of an autoencoder.

As an example to enable comparison between models, the subject used is my dog, CoCo (the namesake of this

project). The following five reference images of CoCo were collected and augmented during training. These modifications included random cropping for diversity, cropping to only the face of the subject, scaling to increase the size of the subject, and adding randomly generated color filters.

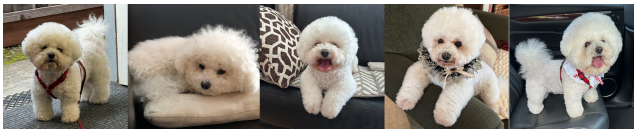

Figure 2. Reference images of example character, CoCo the dog.

The following five sentences were used as test-time prompts for generating a sample story, which are consistent as a reference for all figures below.

(1) *CoCo is sleeping on a big green sofa.*
(2) *CoCo is taking a walk on a grassy field with flowers.*
(3) *CoCo walks along the ocean and trots on the beach.*
(4) *CoCo rests inside of an orange towel.*
(5) *CoCo drinks from a silver bowl full of water.*

4. Methods

4.1. Stable Diffusion

As the baseline model, we use the latent Stable Diffusion v2 model [10] from StabilityAI. This model was trained by learning latent representations $z_0 = \mathcal{E}(x_0)$ when given original images x_0. In training, the forward diffusion process gradually adds noise, producing a noisy latent z_t sampled from the distribution $q(z_t \mid z_0)$. The U-Net model $\epsilon_\theta(z_t, t, \tau)$ predicts the noise component ϵ needed to denoise z_t, conditioned on the timestep t and text embedding τ. The denoising loss $\mathcal{L}$ is then minimized:

$$\mathcal{L} = \mathbb{E}_{z_0,t,\epsilon}\left[\|\epsilon - \epsilon_\theta(z_t, t, \tau)\|_2^2\right]$$

Given the empirical success of stable diffusion models and their ability to generate specified images conditioned on text prompts, this model is well suited for our task. Further, given our limitations on data, training time, and compute, we opt to leverage this foundation model and fine-tune it to personalize it for our task.

4.2. Token Substitution

Fine-tuned language models often rely on learned token associations to generate consistent depictions of subjects. To encourage consistency in the model's outputs at test time [12] and leverage descriptive learned features within the model, we provide additional textual context by augmenting the character identifier with descriptive information. Specifically, we replace occurrences of *Coco* in the prompt with the phrase: *Coco, a white Bichon Frise puppy with dark black eyes and wearing a red collar.*

4.3. DreamBooth and Data Augmentation

Our approach builds upon DreamBooth, a personalized fine-tuning method that enables a pretrained text-to-image diffusion model (such as Stable Diffusion) to generate faithful, novel images of a specific subject from just a few reference images [6]. The core challenge is to encode subject-specific visual features into the model without sacrificing its generalization capabilities or inducing overfitting. We selected DreamBooth because it directly supports this goal through a lightweight yet effective fine-tuning procedure with a loss function that is optimized to reconstruct the subject in the reference images.

To bind a subject to a text prompt, DreamBooth introduces a unique identifier token (e.g., *CoCo*) and fine-tunes the model so that this token becomes associated with the subject's appearance. Given an input set of images $I = i_1, i_2, i_3, i_4, i_5$ representing the subject, we generate augmented prompts such as "a photo of *CoCo* in a [scene]" and train the model to reconstruct the subject within those scene contexts. This encourages the model to learn the subject's representation while maintaining its ability to generalize to unseen prompts.

Formally, the objective combines a reconstruction loss (to align the generated output with the subject) and a prior preservation loss (to prevent overfitting by encouraging outputs similar to class-generic images). The loss function is:

$$\mathcal{L} = \mathcal{L}_{\text{reconstruction}} + \lambda \cdot \mathcal{L}_{\text{prior preservation}}$$

where λ controls the balance between subject fidelity and retention of the model's original distribution.

Given that the number of subject images is small, we experimented with input augmentations to improve generalization and robustness. In particular, we experimented with random crops, facial cropping, and random filtering. These techniques are motivated by image augmentation strategies in contrastive learning and regularization, which help the model focus on distinguishing features necessary for character consistency.

Specifically, we fine-tuned the models using DreamBooth for 750 epochs at learning rate $2.5 \cdot 10^{-6}$ for the main U-net layers and 250 epochs at learning rate $1 \cdot 10^{-6}$ for the text encoder, adapting code from DreamBooth scripts [13].

4.4. Low-Rank Adaptation (LoRA)

To tackle the challenge of adapting large text-to-image models on limited data while maintaining photorealism and stylistic consistency, we apply LoRA (Low-Rank Adaptation) [5], as a parameter-efficient fine-tuning method. Rather than updating all model weights, LoRA introduces learnable low-rank matrices $A \in \mathbb{R}^{d \times r}$ and $B \in \mathbb{R}^{r \times k}$ to approximate the weight update

$$\Delta\Theta = AB$$

with $r \ll \min(d, k)$. These updates are inserted into the cross-attention and feed-forward layers at scalable weight levels, while the pretrained weights remain frozen.

This approach is well-suited for our goal because it allows precise stylistic adaptation with minimal computational overhead, reducing the risk of overfitting commonly seen in full fine-tuning. We chose LoRA over alternatives like full model updates or adapters due to its balance of efficiency and quality, enabling the model to retain generalization while specializing to new subjects. Our experiments compare the output quality and style preservation of LoRA with other methods, confirming its effectiveness in low-data scenarios and assessing the effect of the LoRA weight scaling hyperparameter.

Specifically, we fine-tuned LoRA for 750 epochs at learning rate $1 \cdot 10^{-4}$ using an AdamW optimizer, adapting code from Kohya fine-tuning scripts [14].

4.5. Composite Approaches and Evaluation

We construct a comprehensive set of configurations by taking the Cartesian product over variations across DreamBooth, LoRA, and token substitution, resulting in 8 distinct fine-tuned models built on top of the Stable Diffusion baseline. This composite setup enables systematic ablation to assess the individual and combined impact of each component. We evaluate the models both qualitatively and quantitatively to identify configurations that best preserve subject fidelity, photorealism, and stylistic consistency.

For qualitative analysis, we inspect the model outputs on the given example on subject *CoCo*. Quantitatively, we evaluate using the following 4 metrics.

4.5.1 CLIP Similarity - Reference Similarity

This metric measures how similar the output images are to the reference images of our character, calculated by the average directed Hausdorff distance between the cosine similarities of CLIP embeddings for reference images and output images. Specifically, note that we have 5 input images $I = \{i_1, i_2, i_3, i_4, i_5\}$ and 5 output images $O = \{o_1, o_2, o_3, o_4, o_5\}$. We define the similarity s across all images as the average cosine similarity of CLIP embeddings between the output image and its closest input image. Let c be the cosine similarity function:

$$c(h_x, h_y) = \frac{h_x \cdot h_y}{||h_x||_2 \cdot ||h_y||_2}$$

Then namely, with $h(\cdot)$ as the CLIP embedding function of an image:

$$s = \frac{1}{|O|} \sum_{n=1}^{|O|} \max_{i \in I} c(h(i), h(o_n)).$$

This formulation ensures that each output image is compared to the reference image it most closely resembles in the CLIP embedding space.

4.5.2 CLIP Similarity - Character Consistency

This metric measures how consistent the character is displayed across the output images of each model, measured by the average cosine similarity of the CLIP embeddings for each unique pair of output images from a given model. Consider O_{pairs} as a list of all unique subsets of O with cardinality 2. Let o_{n1} and o_{n2} represent the first output image and second output image respectively in the nth pair of the O_{pairs} list.

With $h(\cdot)$ as the CLIP embedding function of an image, we define the score

$$s = \frac{1}{|O_{\text{pairs}}|} \sum_{n=1}^{|O_{\text{pairs}}|} c(h(o_{n1}), h(o_{n2})).$$

4.5.3 LPIPS Distance - Character Consistency

This metric is the Learned Perceptual Image Patch Similarity (LPIPS) distance [15] between output images per model, which evaluates perceptual similarity between images by comparing deep features extracted from multiple layers of AlexNet [16]. We included this metric because LPIPS is designed to better align with human perception of image differences than traditional pixel-wise metrics. For two images x, y, we use the original LPIPS distance $\ell(x, y)$ where

$$\ell(x, y) = \sum_{l=1}^{L} \frac{1}{H_l W_l} \sum_{(h,w)} ||w_l \odot (f_l^x - f_l^y)||_2^2$$

such that f_l^x, f_l^y refer to feature activations of layer l of AlexNet, w_l is the per-channel weight vector across feature maps, and H_l, W_l are the height and weight of the feature maps at layer l.

Then, to compute the total LPIPS distance across output images for a single model, we again average the distance across all pairs of output images so the final distance is

$$d = \frac{1}{|O_{\text{pairs}}|} \sum_{n=1}^{|O_{\text{pairs}}|} \ell(o_{n1}, o_{n2}).$$

4.5.4 Human Evaluation

Participants (n = 53) were asked to rank each of the 8 models from 1 (best) to 8 (worst) on two metrics: similarity of the output to the reference images (provided to the participants) and the consistency of the character within the model outputs. These ranks were averaged, with lower average ranks being best.

5. Experiments

5.1. Baseline

Without any additional fine-tuning, the baseline generates photorealistic images, but clearly has many flaws. Each image features a different dog and is stylized in a different way, from black and white to colorful. Crucially, the model does not fully understand that *CoCo* is a dog, as seen by the lack of a dog in the fifth image. Given the training of the baseline model, the token *CoCo* was unlikely to have been attributed directly to a white dog like in our reference, and may have been associated with other unrelated images.

Figure 3. Output images for baseline stable diffusion model.

5.2. Token Substitution

As an improvement from the baseline, we see that token substitution at test time offers a simple yet effective way to guide the model toward incorporating key visual elements—such as *Bichon*, *puppy*, and *red collar*—that define the target character. By replacing generic tokens in the prompt with more descriptive or specialized terms, we can steer the generation process to consistently reflect certain features across images.

While this strategy leads to more faithful inclusion of desired attributes, it does not address deeper identity consistency. The resulting images still exhibit notable variation in the dog's appearance, such as changes in face shape, fur texture, and pose. This is expected, as token substitution operates purely at the level of prompting without altering any model parameters. Consequently, the model does not develop an internalized representation of the character, limiting the effectiveness of this method.

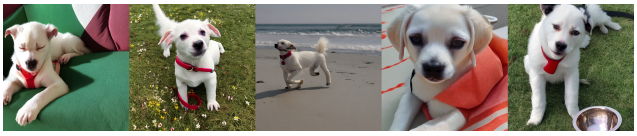

Figure 4. Outputs for stable diffusion with token substitution.

5.3. Character Grounding with DreamBooth

After fine-tuning the baseline model using DreamBooth for 800 epochs, the generated outputs exhibit significantly stronger resemblance to the reference character. The dog is consistently recognizable as a Bichon puppy, with similar facial structure, fur texture, and color across all images. These improvements stem from the fact that the model's weights have been updated to associate these personalized features with the custom token *CoCo*.

While structural consistency has improved notably, especially in terms of facial features and overall body proportions, stylistic consistency remains an issue. The lighting conditions vary across images, with some generations appearing darker and others more brightly lit. Additionally, the artistic rendering differs: the first two images adopt a photorealistic style, whereas the final image appears more like a sketch or digital painting.

These inconsistencies are likely due to the nature of the DreamBooth training data, which consisted of realistic images but lacked explicit stylistic constraints. Without conditioning on style or employing additional control mechanisms, the model defaults to varying interpretations of the prompt during sampling.

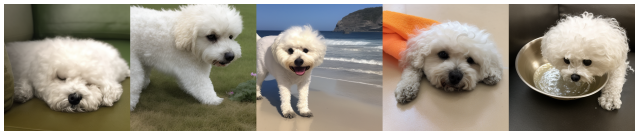

Figure 5. Outputs for DreamBooth fine-tuning.

As before, we see an improvement in consistency with the addition of token substitution to draw attention to certain features, leading to a more uniform style across outputs. However, there is still notable variation in lighting and shadows that distinguishes the images.

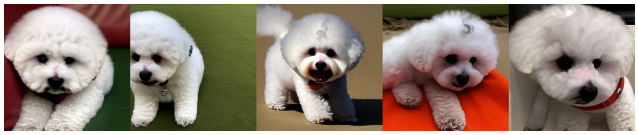

Figure 6. Outputs for Dreambooth with token substitution.

5.4. Investigating Input Perturbations

In fine-tuning with DreamBooth, I experimented with the reference images used in the fine-tuning process. Given that the fine-tuning occurs on very few examples, it becomes clear that small perturbations in the input images translate to large distortions in the output images as a symptom of overfitting.

When the model is fine-tuned on whole-body images of the subject in neutral lighting (Figure 6), the output images are structurally well-formed without notable lighting defects. However, when cropping the input images to only include the subject's head, DreamBooth only shifts the model parameters relative to the head of the subject. As seen in the first and third images below, the output may include only a floating head or the subject's head attached to a mismatched body, as other the other parameters have not been updated.

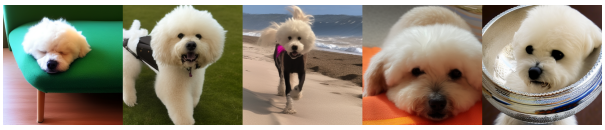

Figure 7. Input photos cropped to the subject's head.

Meanwhile, when cropping the input images randomly such that parts of character are occluded, the parameter updates in DreamBooth are unable to stitch together a full representation of the character. These poorly cropped inputs lead to fully malformed outputs (Figure 8), with oddly shaped bodies, missing facial features, and blurred textures.

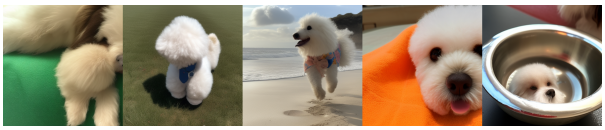

Figure 8. Input photos randomly cropped.

Similarly, input images which are overexposed or have unusual tints yield outputs which have those photo effects magnified. As seen in Figure 9, the generated images from these perturbed inputs feature strange lighting, strong shadows, and odd tints which fully distort the image.

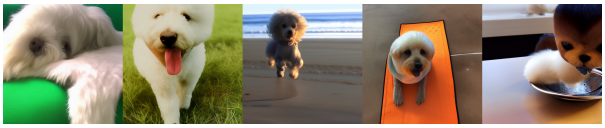

Figure 9. Input photos with random filters.

These experiments highlight the importance of choosing consistent and neutral input images of the character with balanced lighting, limited occlusion, and proper cropping.

5.5. Texture Consistency with LoRA

To explore a computationally lighter approach, we examined the outputs of fine-tuning LoRA on the baseline model. As clearly seen in Figure 10, the characters in the output lack consistency and bear little resemblance to the reference images. While the overall performance of LoRA fails to compare to DreamBooth, this lightweight adjustment impressively maintains consistent lighting effects, photorealistic style, and the correct dog color across all the outputs.

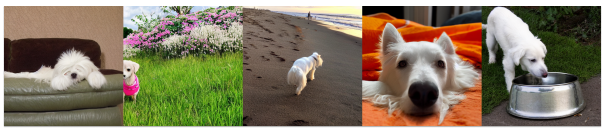

Figure 10. Outputs for LoRA fine-tuning.

Given that LoRA freezes the main U-Net layers of the model, it is expected that its ability to preserve character identity is weaker than DreamBooth's. However, this update successfully captures lower-level features like general lighting and style, which matches expectations given LoRA's use in style transfer. Again, we see improvement in using token substitution for emphasizing identifying features, but with visual differences between outputs given the low-rank update on the reference images.

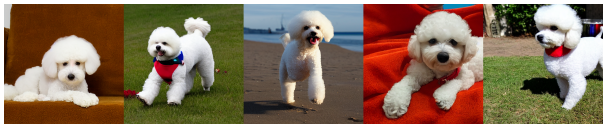

Figure 11. Outputs for LoRA with token substitution.

5.6. Analyzing LoRA Weight Sensitivity

In applying the learned LoRA parameters to the baseline model, I explored the impact of the LoRA weight scaling on the output images.

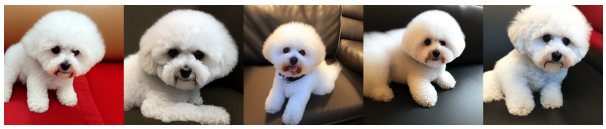

Figure 12. Outputs for DreamBooth with LoRA weights 0, 0.25, 0.50, 0.75, and 1 from left to right.

As seen in Figure 12, lower LoRA strength leads to images with blurrier edges and less strict adherence to style with regard to the reference images. The lighting is darker with heavier shadows and more stylized fur. Meanwhile, higher LoRA strength yields images with textures closer to the reference images, as exemplified in the stylization of the subject's fur.

LoRA Weight	Reference Similarity (CLIP)	Character Consistency (CLIP)	Character Consistency (LPIPS)
0	0.9109	0.9258	0.6071
0.25	0.9074	0.9331	0.6436
0.50	0.9156	0.9247	0.6045
0.75	0.9132	0.9359	0.5935
1	**0.9222**	**0.9365**	**0.5932**

Table 1. Comparing reference similarity and character consistency across LoRA weights.

These qualitative observations are supported by quantitative results in Table 1. These results show that with regard to both CLIP similarity and LPIPS distance, the full LoRA weight 1 yields images with the greatest similarity to other outputs as well as the original reference images. This LoRA weight was used in the following models.

5.7. Composite Models

When combining the two fine-tuning techniques, the outputs are more balanced with regard to style, texture, and lighting, with reasonable resemblance of the character in the outputs. The character's likeness holds across the images, but to a lesser degree than DreamBooth without LoRA. This result highlights the trade-off in style consistency and identity preservation between DreamBooth and LoRA.

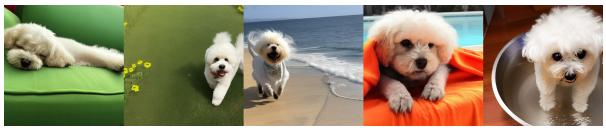

Figure 13. Outputs for composite model.

With the addition of token substitution, the additional prompting of identifiable features seems to offset some of the loss of identity caused by LoRA. In Figure 14, the subject bears clear resemblance to the reference images as well as across output images, in a photorealistic style with clear lighting and exposure.

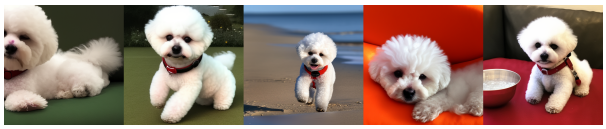

Figure 14. Outputs for composite model with token substitution.

6. Results

6.1. Qualitative Analysis and Human Evaluation

Model	Reference Similarity Avg. Rank	Output Consistency Avg. Rank
SD (Baseline)	7.9423	7.8274
SD + TS	6.4423	5.9615
SD + DB	2.8462	2.9808
SD + DB + TS	2.5769	**1.9038**
SD + LoRA	6.4615	6.7308
SD + LoRA + TS	3.9615	3.8272
SD + DB + LoRA	4.1923	4.1519
SD + DB + LoRA + TS	**1.7115**	2.3076

Table 2. Average ranking by humans measuring similarity between the reference images and output photos (left) and the consistency between output photos from the same model (right). Lower numbers are desireable (best is rank 1, worst is rank 8). SD = Stable Diffusion, TS = Token Substitution, and DB = DreamBooth.

The average rankings across the outputs of the 8 models from n = 53 human participants are displayed in Table 2. The final composite model trained with DreamBooth and LoRA and using token substitution ranked highest for similarity to the reference images and ranked second highest for character consistency across outputs. The model with only DreamBooth and using token substitution also performed very well, being ranked second highest for reference image similarity and ranked highest for output consistency.

Notably, the addition of LoRA improves the similarity between model outputs and the reference images. This distinction is due to the impact of LoRA on style, as the final composite model yielded photorealistic images with neutral lighting reminiscent of the input photos. Meanwhile, the DreamBooth model without LoRA showed greater consistency without needing to balance for style, thus ranking higher across outputs.

6.2. Input-to-Output Reference Similarity

The CLIP similarity across model outputs corroborates the human evaluation, both when the output images are uncropped and cropped to the subject's face. By these semantic metrics, the final composite model with token substitution has the highest similarity between the output images and reference images, with the DreamBooth token substitution model in a close second.

Model	CLIP (Original)	CLIP (Cropped)
SD (Baseline)	0.6996	0.7095
SD + TS	0.7798	0.7973
SD + DB	0.8373	0.8591
SD + DB + TS	0.8722	0.8589
SD + LoRA	0.7542	0.8054
SD + LoRA + TS	0.8599	0.8320
SD + DB + LoRA	0.7933	0.8482
SD + DB + LoRA + TS	**0.8789**	**0.8592**

Table 3. CLIP similarity scores comparing input photos and output photos. SD = Stable Diffusion, TS = Token Substitution, and DB = DreamBooth.

6.3. Within-Output Character Consistency

Similarly, the CLIP similarity and LPIPS distance metrics across only the output images per model suggest that the top models are the composite model and DreamBooth model, both with token substitution at test time. Notably, the composite model is only outranked by the DreamBooth model without LoRA on CLIP similarity when the output images are cropped, likely due to the lesser effect on lighting and shadows where LoRA makes a big difference on backgrounds.

Model	CLIP (Orig.)	CLIP (Crop.)	LPIPS Dist.
SD (Baseline)	0.6602	0.6934	0.6519
SD + TS	0.7629	0.8130	0.6512
SD + DB	0.8167	0.8367	0.5655
SD + DB + TS	0.8774	**0.8907**	0.5547
SD + LoRA	0.7299	0.7922	0.6979
SD + LoRA + TS	0.8682	0.8759	0.5510
SD + DB + LoRA	0.7884	0.8729	0.5306
SD + DB + LoRA + TS	**0.8950**	0.8903	**0.5142**

Table 4. Metrics measuring character consistency between output photos. Higher numbers are desirable for CLIP similarity while lower numbers are desirable for LPIPS distance. SD = Stable Diffusion, TS = Token Substitution, and DB = DreamBooth.

6.4. Failure Modes and Analysis

Even in the best model, there were patterns in failure modes. In particular, the final model often hallucinated extra appendages or failed to distinguish between the fur of the subject and an object with a similar texture (like a rug). Another common failure case was the generation of the collar, which often featured other visually similar objects such as pocketknives, razor, and can openers.

Figure 15. Examples of failures in generating the subject.

These issues can likely be attributed to DreamBooth overfitting to sparse visual cues due to the small data set of reference images. When these input images have correlated background elements or lighting artifacts, the model can magnify these unintended features (as in Section 5.4) and incorrectly associate certain shapes with broader object categories. Especially since Stable Diffusion operates in a dense latent space, the model may struggle to cleanly separate terms like *fur* and *collar*. This effect can also be amplified by the applied LoRA, as the weights in LoRA are designed to amplify certain stylistic features and in these cases accidentally override the realistic style of local objects.

7. Conclusion

This project investigated how to generate consistent, high-quality images of a character across different scenes using text prompts. The key finding was that the most effective pipeline combined a Stable Diffusion model fine-tuned with both DreamBooth and LoRA, along with token substitution at inference time to emphasize key identifying features. DreamBooth was highly effective at encoding subject identity but prone to overfitting, which underscored the importance of carefully selected, well-cropped, and diverse reference images. LoRA introduced valuable stylistic control, though experiments revealed a trade-off between style adherence and identity preservation. Across all models, token substitution consistently improved fidelity to the reference subject, particularly when used alongside both DreamBooth and LoRA to balance the competing demands of realism and stylistic alignment.

Despite these promising results, we recognize the limitations of this pipeline. The model remains sensitive to small background features in the input images, likely due to overfitting in DreamBooth and the amplifying effect of LoRA. Future work could focus on developing automated or guided techniques for selecting optimal reference images to mitigate this issue. Additionally, the project was limited to a single LoRA layer. Further research could explore using multiple LoRA modules to capture distinct elements of style, or integrating alternative style transfer techniques such as neural style transfer for greater flexibility in downstream applications across art styles.

8. Contributions & Acknowledgments

This project was completed independently, without collaborators or shared work across courses. It builds upon publicly available training scripts for fine-tuning with DreamBooth [13] and LoRA [14], which were adapted to suit the specific objectives of this work. My contributions include designing the study, collecting and augmenting data, tuning hyperparameters, fine-tuning models with DreamBooth and LoRA, developing and computing task-specific evaluation metrics, performing ablation studies on the composite models, and conducting error analysis and symptom diagnosis.

References

[1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.

[2] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.

[3] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation, 2021.

[4] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bjorn Ommer. High-resolution image synthesis with latent diffusion models. *CoRR*, abs/2112.10752, 2021.

[5] Edward Hu, Shen Yelong, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. 2021.

[6] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation, 2022.

[7] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023.

[8] Zhengguang Zhou, Jing Li, Huaxia Li, Nemo Chen, and Xu Tang. Storymaker: Towards holistic consistent characters in text-to-image generation, 2024.

[9] Omri Avrahami, Amir Hertz, Yael Vinker, Moab Arar, Shlomi Fruchter, Ohad Fried, Daniel Cohen-Or, and Dani Lischinski. The chosen one: Consistent characters in text-to-image diffusion models, 2023.

[10] StabilityAI. Stable diffusion 2 model: Text-to-image model. `https://huggingface.co/stabilityai/stable-diffusion-2`, 2022.

[11] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

[12] Haoran Feng, Zehuan Huang, Lin Li, Hairong Lv, and Lu Sheng. Personalize anything for free with diffusion transformer, 2025.

[13] TheLastBen. Dreambooth fine-tuning scripts: fast-stable-diffusion. `https://github.com/TheLastBen/fast-stable-diffusion`, 2023.

[14] HollowStrawberry. kohya-colab. `https://github.com/hollowstrawberry/kohya-colab`, 2024.

[15] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

Library	Version
PyTorch	2.1.0
NumPy	1.26.4
Transformers	4.41.1
Diffusers	0.28.0
Huggingface Hub	0.23.0
Torchvision	0.16.0
Pillow (PIL)	10.3.0
LPIPS	0.1.4
SciPy	1.13.1

Table 5. Library dependencies used in this project.