BRAT: Bonus oRthogonAl Token for Architecture Agnostic Textual Inversion

Anonymous authors
Paper under double-blind review

Abstract

Textual Inversion remains a popular method for personalizing diffusion models, in order to teach models new subjects and styles. We note that textual inversion has been under-explored using alternatives to the UNet, and experiment with textual inversion with a vision transformer. We also seek to optimize textual inversion using a strategy that does not require explicit use of the UNet and its idiosyncratic layers, so we add bonus tokens and enforce orthogonality. We find the use of the bonus token improves adherence to the source images and the use of the vision transformer improves adherence to the prompt. Code is available at REDACTED WHILE UNDER REVIEW.

1 Introduction

When the British pop singer Charli XCX sings *"When you're in the mirror, do you like what you see? When you're in the mirror, you're just looking at me"* in her song 360 (XCX, 2024), she is implicitly acknowledging that mirrors show an instance of a subject. However, due to her eminence as an icon, the specific instance is replaced with her. Charli XCX's lyrics imply that it is very significant and meaningful *which* person appears in the mirror when the audience looks in the mirror. Depicting a generic picture of a cat is less useful than depicting a *specific* cat. The pop singer herself has admitted that she doesn't just wear any pair of boots but strongly prefers a very specific model of black Prada boots (GQ, 2024). This affinity towards specific instances of subjects motivates the personalization of text-to-image models.

While most of the standard pre-trained diffusion models know a few specific instances of specific characters or objects (for example, most off-the-shelf diffusion models can generate images of the Statue of Liberty, not just a generic statue when prompted to do so), there has been no shortage of clever methods to expand their output space to a new subject or style (Ruiz et al., 2022; Li et al., 2023; Ma et al., 2023; Purushwalkam et al., 2024; Wang et al., 2024; Ye et al., 2023; Chen et al., 2024; Zhang et al., 2023a) We revisit one of the earliest methods of personalizing diffusion models, known as Textual Inversion (Gal et al., 2022). Textual inversion is most often used to teach diffusion models new subjects. However, it can also be used effectively to teach new styles, similar to style transfer, but without a content image. We also note that most, if not all, textual inversion literature is wedded to the use of the UNet, and many of the optimizations are UNet-specific. While not all diffusion models use UNet the massive size of the text encoders used with other models such as Vision Transformers (Dosovitskiy et al., 2021) can make textual inversion on them difficult. However, this problem can be circumvented with the use of adapters (Zhao et al., 2024) that map the embeddings of one text encoder into the space of another, opening up the possibility of textual inversion on Vision Transformers. At the same time, we see that the fixation on UNet architecture has meant a dearth of model-agnostic improvements to textual inversion. Given the state of the field, our contributions are as follows:

- We apply textual inversion to a non-UNet architecture
- We use a new token method, which we call BRAT, that is agnostic to choice of denoising model
- We demonstrate that BRAT improves adherence to the source image, and the non-UNet architecture improves human preference rating and prompt adherence

2 Related Work

2.1 Text to Image

The Generative Adversarial Network (Goodfellow et al., 2014) was a milestone in using machine learning to create new visual artifacts. Later works like Deepdream (Mordvintsev et al., 2015) and style transfer using Gram matrices (Gatys et al., 2015) or CycleGAN (Zhu et al., 2020) could generate images conditioned on source images. GANs were also implemented conditioned on text (Mirza & Osindero, 2014; Reed et al., 2016), allowing users even more control of what these models could create. Recently, autoregressive methods (Ramesh et al., 2021; Ding et al., 2021; Yu et al., 2022) and diffusion (Sohl-Dickstein et al., 2015; Ho et al., 2020a; Podell et al., 2023) have emerged and produce even higher quality images than prior works. The latter method usually consists of a denoising model, most commonly the Unet (Ronneberger et al., 2015), conditioned on text embeddings from a pretrained text encoder, progressively tries to guess the amount of random noise in a corrupted image or latent embedding of one, however the "denoising" task can predict other forms of image corruption as well (Bansal et al., 2022; Liu et al., 2024).

2.2 Textual Inversion

Foundation models are trained on massive datasets. For example, stable diffusion (Rombach et al., 2022) was trained on a few billion images (Schuhmann et al., 2022); DALLE-3 (Ramesh et al., 2023) was supposedly trained on the data used for CLIP (Ramesh et al., 2021) and DALL-E (Radford et al., 2021), each containing hundreds of millions of images. Subsequently, these models can accurately portray an extremely wide range of concepts. However, they do not have the ability to consistently generate instances of a specific subject that may be similar to things but not necessarily found in the training data. Emerging almost simultaneously, Dreambooth (Ruiz et al., 2022) tuned the Unet, while Textual Inversion (Gal et al., 2022) tuned a new token in the text encoder vocabulary to teach these models unique new concepts and optimized the embedding to match a few example images. There were many modifications and expansions to Textual Inversion. For example, DreamArtist (Dong et al., 2023) included a "negative" token, ProSpect (Zhang et al., 2023b) used different tokens for different timesteps of the diffusion process, P+ (Voynov et al., 2023) and MATTE (Agarwal et al., 2023) used different tokens for different layers of the UNet, and NeuralSpace (Alaluf et al., 2023) trained an auxiliary network to modify the token, conditioned on the UNet layer and timestep. Similar to textual inversion, some works focus on optimizing a prompt description of an image without necessarily introducing a new token into the vocabulary (Wen et al., 2023; Yu et al., 2024).

3 Method

3.1 Baseline

We describe standard textual inversion. Given:

- Image dimensions Height $H \in \mathbb{N}$, Width $W \in \mathbb{N}$ and Channels $C \in \mathbb{N}$
- Image $x \in \mathbb{R}^{H \times W \times C}$ drawn from real dataset $\mathcal{D}$
- Latent Height $L_H <<< H \in \mathbb{N}$, Latent Width $L_W <<< W \in \mathbb{N}$ and Latent Channels $L_C \in \mathbb{N}$
- Variational Autoencoder (Kingma & Welling, 2014) $\mathcal{E} : \mathbb{R}^{H \times W \times C} \to \mathbb{R}^{L_H \times L_W \times L_C}$
- Timestep $t \in [0, T]$, where T is the maximum number of timesteps
- Time-conditioned noise $\epsilon \sim \mathcal{N}(0, 1|t)$
- Noised Latent Embedding z_t, where $z_t = \mathcal{E}(x) + \epsilon$
- Text token embedding dimension $M \in \mathbb{N}$, Text vocab size $V \in \mathbb{N}$, $M <<< V$, prompt length $N \in \mathbb{N}$
- Prompt $p \in \mathbf{text}$

- Text Encoder $c_\phi : \textbf{text} \rightarrow \mathbb{R}^{N \times M}$, parameterized by ϕ
- Denoising model, ϵ_θ parameterized by θ

The traditional Latent Diffusion Model denoising training objective is

$$\mathcal{L}_{LDM} = \mathbb{E}_{z \sim \mathcal{E}(x), x \sim \mathcal{D}, t, p} ||\epsilon - \epsilon_\theta(z_t, t, c_\phi(p))||_2^2$$

Usually the θ parameters are optimized. For textual inversion, θ and ϕ are frozen, and the only trainable parameter is the embedding of the new token. The new token is usually a pseudoword added to the vocabulary such as **"<sks>"**, and the embedding is represented as v^*. The dataset is much smaller than the original dataset used to train the diffusion model, and is instead 3-5 pictures of the subject. The prompts are usually very basic, along the lines of **"a picture of <sks>"** or **"a photo of <sks>"**.

3.2 Adapting Text Encoders to Different Denoisers

Traditionally, textual inversion finetunes off of a transformer and a UNet. Many of the optimizations have been based on leveraging unique features of the UNet, such as how different layers correspond to different attributes of an image (Voynov et al., 2023; Agarwal et al., 2023; Alaluf et al., 2023). A challenge to using textual inversion with different combinations of text encoder and denoiser is larger text encoders can be harder to train; refer to appendix A.5 for some examples. Work by Zhao et al. (2024) allows us to use a pretrained adapter module that maps the embeddings of one text encoder into the space of another. For example, the PixArt-α vision transformer (Chen et al., 2023) architecture has impressive visual results, but uses a text encoder of a few billion parameters. Training and optimizing this text encoder with textual inversion requires more memory, and thus more expensive hardware. However, using an adapter, we can perform textual inversion on a smaller model of only a few million parameters.

3.3 BRAT: Bonus Orthogonal Token

Given that many of the textual inversion improvements are UNet-specific, we would like to try a token strategy that is agnostic to our choice of denoiser. We also investigate if one token may be insufficient to capture all the relevant information of one concept. Many concepts are better described with multiple words than just one; "orange cat" is more descriptive than "cat". We add an auxiliary pseudoword **"<fkf>"**, which we refer to as the "bonus" token, and corresponding embedding w^*. However, we don't want $w^* = v^*$ or $w^* = -v^*$, as that essentially means w^* and v^* embed the exact same information. We want to encourage the two embeddings to be orthogonal, so we introduce a new regularization term:

$$\mathcal{L}_{Spare} = \lambda[\textbf{cos}(w^*, v^*)^2]$$

Where **cos** = cosine similarity and λ is a scalar weight coefficient, which we choose to be 0.01. This loss penalizes cosine similarity of 1 and -1 the same, thus discouraging $w^* = v^*$ and $w^* = -v^*$ and penalizes a cosine similarity of 0, encouraging w^* and v^*, to be orthogonal and capture different aspects of the subject. The prompts follow the format of baseline textual inversion, such as **"a picture of <sks> <fkf> "** or **"a photo of <sks> <fkf>"**. The objective becomes

$$\mathcal{L} = \mathcal{L}_{Spare} + \mathcal{L}_{LDM}$$

Where the only tunable parameters are the embeddings w^*, v^*. We can then expand this to any number of bonus tokens, all of which are trained to be orthogonal to each other and the initial pseudoword. For our experiments we try with one and three bonus tokens. We call our method **BRAT**, for **B**onus o**R**thogon**A**l **T**oken.

4 Experiments

4.1 Datasets

Following Gal et al. (2022), we used textual inversion to teach new subjects and styles. All images were converted to RGB, padded to be square, resized to $512 \times 512 \times 3$ using bilinear interpolation and normalized to be between $[-1, 1]$.

4.1.1 Subject Data

For the subject data, D_{sub}, we used the 30 non-human subjects (cans, dogs, toys) from the original Dreambooth paper (Ruiz et al., 2022), retrieved from `https://github.com/google/dreambooth`. We only used three images for each subject, even though many of the images in the source repository had more than three images. Each subject was already labeled. We used the embedding of the subject label to initialize the custom placeholder token and spare token embeddings. Sometimes, this subject label had multiple words or extraneous numbers (for example, colorful_sneaker or cat2). In that case, we removed any numbers and/or only used the embedding of the second word (fancy_boot became boot, dog3 became dog, etc.).

4.1.2 Style Data

For the style data, D_{sty}, we selected sixteen artists off of `deviantart.com`, and chose an image from each one that we felt reflected their unique artistic style. Refer to appendix A.1 to see them. New token embeddings were simply initialized with the embedding of the word "art". We should caveat that the task for the style dataset is *not* style transfer; there is no content image that we are attempting to imbue a style unto; we want to generate a *new* image with the stylistic features of the source image.

4.2 Metrics

We used six different metrics for evaluation:

- **CLIP Similarity (CLIP Sim):** CLIP (Radford et al., 2021) Image encodings do not disentangle content and style but are the standard way of condensing images into a vector space. For both datasets, for each source image, we found the average cosine distance between the *CLIP* embedding of each source image and each validation image and reported the average distance across subjects.
- **CLIP Consistency (CLIP Cons):** For both datasets, given the CLIP embeddings of each validation image, we calculated the distance between each pair and averaged them to get the consistency score
- **Style/Content Similarity (Style/Cont Sim):** Previous works (Tumanyan et al., 2022; Kwon & Ye, 2023) have found that using the activations of the intermediate layers of the vision transformer loaded from the **dino-vits16** checkpoint (Caron et al., 2021) are good embeddings of the content of an image. For each subject, we found the average cosine distance between the *content* embedding of each source image and each validation image and reported the average distance across subjects. For each style, we found the average cosine distance between the *style* embedding of the single source image and each validation image and reported the average distance across subjects.
- **Style/Content Consistency (Style/Cont Cons):** For each subject, given the content embeddings of each validation image, we calculated the distance between each pair and averaged them to get the consistency score. For each style, given the style embeddings of each validation image, we calculated the distance between each pair and averaged them to get the consistency score. This metric measured how consistent the representation of the subject was across different prompts.
- **Image Reward (Img Rew):** In order to approximate subjective human preferences on whether an image is "good" or not, we used the pretrained Image Reward model (Xu et al., 2023) downloaded from the python package **image-reward** and used the **ImageReward-v1.0** checkpoint to score each validation image, and averaged them for this metric.

- **Prompt Similarity (Pro Sim):** The CLIP model is multimodal and can embed texts and images into the same space. So, for each validation prompt and subsequent generated image, we can extract embeddings of both prompt and image and find the cosine similarity to compute how close to the prompt the image is. We found the cosine similarity between each validation image and its source prompt and averaged them for this metric.

4.3 Models

We used 2 types of denoising model: a UNet, specifically the **stable-diffusion-v1-4** checkpoint from `https://huggingface.co/CompVis/stable-diffusion-v1-4`, and PixArt-α vision transformer (Chen et al., 2023), based off of Peebles & Xie (2023), specifically the **PixArt-XL-2-512x512** checkpoint from `https://huggingface.co/PixArt-alpha/PixArt-XL-2-512x512`. By default, the UNet uses a CLIP encoder, specifically the transformer-based text encoder used in **clip-vit-large-patch14**, downloaded from `https://huggingface.co/CompVis/stable-diffusion-v1-4` (but identical to the checkpoint from `https://huggingface.co/openai/clip-vit-large-patch14`). The vision transformer uses an extremely large T5 encoder, specifically the **4.3B Flan-T5-XXL** checkpoint downloaded from `https://huggingface.co/PixArt-alpha/PixArt-XL-2-512x512`, respectively. We found that training the vision transformer with the **4.3B Flan-T5-XXL** encoder was extremely slow to converge even with a higher learning rate and more training epochs, so we did not explore this. Refer to appendix section A.5 for some visual examples.

However, we also leverage the pretrained adapters (Zhao et al., 2024) from `https://huggingface.co/shihaozhao/LaVi-Bridge` so that we can use alternative text encoders. This repository contained only three adapters, adapting a **t5-large** encoder to a unet, adapting a **t5-large** encoder to a vision transformer and adapting a llama encoder to a unet. We used the **t5-large** checkpoint from `https://huggingface.co/google-t5/t5-large` (which is a few million parameters, unlike the other T5 encoder) for both the UNet and vision transformer. At a high level, a UNet differs from a vision transformer in that it uses skip connections, and a vision transformer breaks images into patches before applying attention to them. It is best to refer to Ronneberger et al. (2015) or Peebles & Xie (2023) for deeper discussions of the UNet and Vision Transformer architectures, respectively. We experimented with the **Llama-2-7b-hf** checkpoint (Touvron et al., 2023), but we found this often failed to learn the target concept, just like the larger transformer encoder. Refer to appendix section A.5 for some visual examples. This gives us three different combinations of noise predictor and text encoder, as detailed in table 1.

Name	Text Encoder	Denoiser	Uses Adapter?
T5 Trans	t5-large	PixArt-XL-2-512x512	✓
T5 UNet	t5-large	stable-diffusion-v1-4	✓
CLIP UNet	clip-vit-large-patch14	stable-diffusion-v1-4	×

Table 1: Caption

We tested a few token strategies for each model:

1. **Default:** identical to Gal et al. (2022).
2. **Multi 10:** based off of ProSpect (Zhang et al., 2023b), we have ten tokens, each corresponding to five inference steps
3. **Multi 50:** based off of ProSpect (Zhang et al., 2023b), we have a separate token for each inference step, for a total of 50 tokens.
4. **Negative:** based off DreamArtist (Dong et al., 2023), we use a negative token p_-, and the loss is $||\epsilon - f(\epsilon_\theta(z_t, t, c_\phi(p)), \epsilon_\theta(z_t, t, c_\phi(p_-)))$, where $f(a, b) = b + \gamma(a - b)$.
5. **Bonus:** using a bonus token in addition to the placeholder token, with the orthogonal loss between the original placeholder token and the bonus

6. **Triple Spare:** using three bonus tokens in addition to the placeholder token, with the orthogonal loss between all combinations of the placeholder and bonus tokens (i.e. for one placeholder token and three spares, we would have 12 different orthogonal loss terms)

4.4 Prompts

We used a set of prompts for training and a set of prompts for testing. For subjects, the training prompts were similar to the training prompts used in past works, such as *"a photo of a nice* {}*"* The test set, **short prompts**, was similar to the prompts used in past works (Gal et al., 2022), such as *"a picture of* {} *as a policeman"*. For style prompts, we based our training prompts off of the style prompts used in the repo for Dong et al. (2023), such as *"a cropped painting, art by* {}*"*, and based our test prompts off of those used for subjects, such as *"a police officer, art by* {}*"* Refer to Appendix A.2 for a list of all prompts.

4.5 Quantitative Results

Tables 2 and 3 show the quantitative results. For each metric, for each model, we bolded the highest score. All relevant training hyperparameters are listed in Appendix A.4.1. Each method is named for the encoder-denoiser combination used and the token strategy used.

Method	CLIP Sim	CLIP Cons	Cont Sim	Cont Cons	Img Rew	Pro Sim
T5 Trans Default	0.6048	0.5968	0.2952	0.3085	0.2234	0.2785
T5 Trans Bonus	**0.6393**	0.6427	**0.3054**	**0.3235**	0.1217	0.276
T5 Trans Triple Bonus	0.6287	**0.644**	0.2998	0.3197	-0.0625	0.2668
T5 Trans Multi 10	0.5718	0.5581	0.2895	0.3043	0.2604	**0.2802**
T5 Trans Multi	0.5161	0.5293	0.2804	0.3078	**0.3046**	0.2772
t5 Trans Negative	0.4884	0.527	0.2745	0.3111	0.2758	0.2741
T5 UNet Default	0.6044	0.5859	0.2853	0.2711	-0.0436	0.2738
T5 UNet Bonus	0.622	0.6201	0.2885	0.2734	-0.0686	0.269
T5 UNet Triple Bonus	**0.6575**	**0.6493**	**0.2991**	**0.2904**	-0.2908	0.2635
T5 UNet Multi 10	0.4747	0.5157	0.2551	0.2482	0.2796	0.2728
T5 UNet Multi	0.4679	0.5187	0.249	0.2414	**0.2554**	0.2707
T5 UNet Negative	0.5081	0.5435	0.2621	0.2586	0.1888	**0.2789**
CLIP UNet Default	0.7801	0.7999	0.3149	0.2963	-1.9338	0.2032
CLIP UNet Bonus	0.7696	0.7819	0.3201	0.3015	-1.624	0.2169
CLIP UNet Triple Bonus	0.8045	0.8224	0.3471	0.335	-1.8263	0.2101
CLIP UNet Multi 10	**0.8307**	**0.8605**	**0.4019**	**0.4228**	-1.9432	0.2057
CLIP Unet Multi	0.7959	0.7956	0.3672	0.3542	-1.2943	0.2301
CLIP Unet Negative	0.5382	0.5805	0.2589	0.2461	**0.1165**	**0.2756**

Table 2: Subject Scores

Across all models, the use of the bonus token(s) improves content/style similarity and consistency scores compared to the default, at the expense of lower prompt similarity. This reflects the comparison of methods as shown in (Avrahami et al., 2024), where different subject-to-image methods often exist on a Pareto frontier where higher prompt similarity comes at the expense of lower consistency and vice versa. In general, prompt similarity varied little across models and methods, while Image Reward varied a lot, but for both the CLIP UNet architecture had a tendency to fare poorly compared to the T5 UNet and T5 Trans; given that Image Reward is conditioned *on* the text prompt and implicitly measures alignment as well as human "appreciation". We theorize that this is the result of using a frozen adapter between the text and encoder and the denoiser. The adapter maps tokens from the embedding space of the text encoder to the embedding space of the denoiser. Given that the new tokens are created in the embedding space of the text encoder, the denoiser never sees the new tokens and is thus prevented from overfitting to a particular token.

Method	CLIP Sim	CLIP Cons	Sty Sim	Sty Cons	Img Rew	Pro Sim
T5 Trans Default	0.5837	0.6053	0.2338	0.2536	0.493	0.2867
T5 Trans Bonus	**0.6325**	**0.6431**	**0.3152**	**0.3431**	0.3458	0.2802
T5 Trans Triple Bonus	0.6167	0.6202	0.264	0.2957	0.6289	**0.2955**
T5 Trans Multi 10	0.5821	0.6226	0.2295	0.2475	0.6131	0.2882
T5 Trans Multi 50	0.5571	0.6068	0.19	0.2148	**0.7548**	0.293
T5 Trans Negative	0.5411	0.599	0.166	0.1973	0.6017	0.2912
T5 UNet Default	0.5945	0.6228	0.2825	0.3815	0.8184	0.2795
T5 UNet Bonus	**0.6154**	**0.6332**	**0.2988**	**0.4135**	0.4593	0.275
T5 UNet Triple Bonus	0.61	0.6326	0.2889	0.4007	0.7614	0.2837
T5 UNet Multi 10	0.5523	0.6113	0.2108	0.2936	**1.1392**	0.292
T5 UNet Multi 50	0.5546	0.616	0.2017	0.2865	1.0833	**0.2915**
T5 UNet Negative	0.5507	0.6049	0.2066	0.2731	1.0446	0.2903
CLIP UNet Default	0.8003	0.8302	0.5693	0.7011	-1.4752	0.2151
CLIP UNet Bonus	0.8284	0.8496	0.6262	0.732	-1.4739	0.2204
CLIP UNet Triple Bonus	0.8361	0.865	0.6707	0.7651	-1.5033	0.2168
CLIP UNet Multi 10	**0.8835**	**0.9001**	**0.7632**	**0.825**	-1.7364	0.2137
CLIP UNet Multi 50	0.8776	0.8821	0.7333	0.7627	-1.6204	0.2186
CLIP UNet Negative	0.6177	0.5998	0.292	0.2952	**0.6837**	**0.2986**

Table 3: Style Scores

4.6 Visual Results

We display some visual results. Each image was generated with the same prompt. Subjectively, we find that the CLIP UNet tends to "overfit" and sometimes ignore the prompt completely, which is consistent with quantitative results where using the T5 text encoder improved Image Reward and Prompt Similarity.

Figure 1: Subject Images, generated with caption "a photo of {} wearing sunglasses"

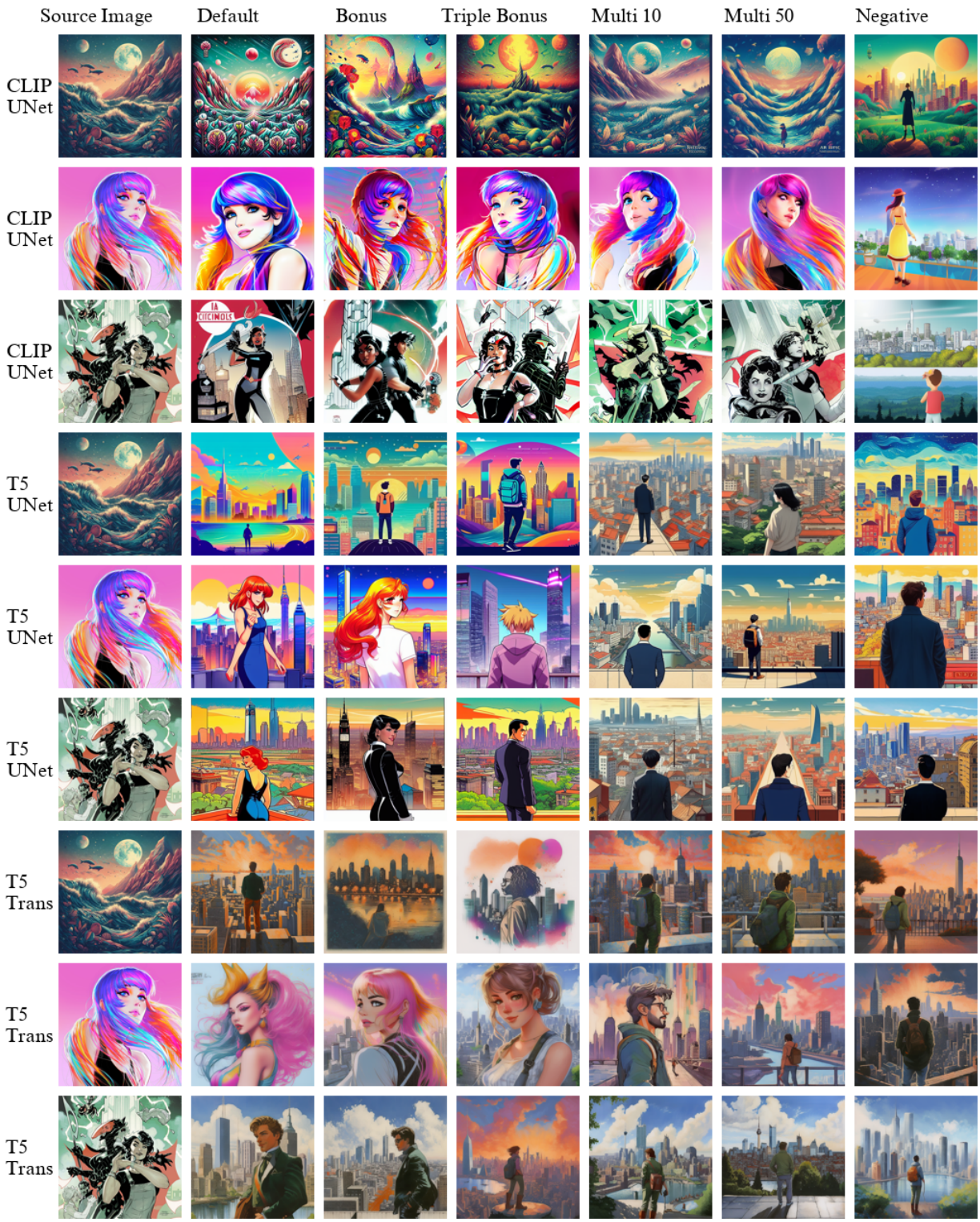

Figure 2: Style Images, generated with caption "a person with a city in the background, art by {}"

4.7 Discussion

5 Conclusion

To summarize, we observed that textual inversion for personalizing diffusion models was wedded to the UNet architecture, so we experimented with textual inversion that relied on the vision transformer instead, and used BRAT to optimize the tokens for such. We saw each of our contributions reflected a movement along the prompt similarity-content adherence pareto frontier. Further work could entail expanding this approach to more alternatives to the UNet, or longer training times on larger text encoders that may capture richer, more meaningful information in their embedding space.

Broader Impact Statement

Many people are worried about the effects of generative AI. By creating art, this technology encroaches on an area once solely occupied by humans. Companies have faced criticism for potentially using AI, and many creatives, such as screenwriters and actors, have expressed concerns about the security of their jobs. However, AI can assist humans by enhancing efficiency, providing inspiration, and generating new ideas. The future of copyright protection for AI-generated art remains uncertain, as current laws are based on the principle that creative works originate from human authors. To mitigate harm and maximize benefits for everyone, clear and consistent policies from governments, industries, and academic groups will be necessary.

6 Assistance

Author Contributions

Redacted while under review

Acknowledgements

Redacted while under review

References

Aishwarya Agarwal, Srikrishna Karanam, Tripti Shukla, and Balaji Vasan Srinivasan. An image is worth multiple words: Multi-attribute inversion for constrained text-to-image synthesis, 2023.

Yuval Alaluf, Elad Richardson, Gal Metzer, and Daniel Cohen-Or. A neural space-time representation for text-to-image personalization, 2023.

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, 2024. URL `https://arxiv.org/abs/2311.10093`.

Arpit Bansal, Eitan Borgnia, Hong-Min Chu, Jie S. Li, Hamid Kazemi, Furong Huang, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Cold diffusion: Inverting arbitrary image transforms without noise, 2022. URL `https://arxiv.org/abs/2208.09392`.

Lukas Biewald. Experiment tracking with weights and biases, 2020. URL `https://www.wandb.com/`. Software available from wandb.com.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021.

Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, and Zhenguo Li. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis, 2023.

Weifeng Chen, Jiacheng Zhang, Jie Wu, Hefeng Wu, Xuefeng Xiao, and Liang Lin. Id-aligner: Enhancing identity-preserving text-to-image generation with reward feedback learning, 2024. URL https://arxiv.org/abs/2404.15449.

Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, and Jie Tang. Cogview: Mastering text-to-image generation via transformers, 2021.

Ziyi Dong, Pengxu Wei, and Liang Lin. Dreamartist: Towards controllable one-shot text-to-image generation via positive-negative prompt-tuning, 2023.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H. Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion, 2022.

Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style, 2015.

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

British GQ. 10 things charli xcx can't live without | 10 essentials. YouTube, 2024. URL https://www.youtube.com/watch?v=oxZn2XpWOXg&ab_channel=BritishGQ.

Sylvain Gugger, Lysandre Debut, Thomas Wolf, Philipp Schmid, Zachary Mueller, Sourab Mangrulkar, Marc Sun, and Benjamin Bossan. Accelerate: Training and inference at scale made simple, efficient and adaptable. https://github.com/huggingface/accelerate, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020a.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020b. URL https://arxiv.org/abs/2006.11239.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

Gihyun Kwon and Jong Chul Ye. Diffusion-based image translation using disentangled style and content representation, 2023.

Dongxu Li, Junnan Li, and Steven C. H. Hoi. Blip-diffusion: Pre-trained subject representation for controllable text-to-image generation and editing, 2023.

Chenxi Liu, Gan Sun, Wenqi Liang, Jiahua Dong, Can Qin, and Yang Cong. Museummaker: Continual style customization without catastrophic forgetting, 2024. URL https://arxiv.org/abs/2404.16612.

Jian Ma, Junhao Liang, Chen Chen, and Haonan Lu. Subject-diffusion:open domain personalized text-to-image generation without test-time fine-tuning, 2023.

Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets, 2014.

Alexander Mordvintsev, Christopher Olah, and Mike Tyka. Inceptionism: Going deeper into neural networks, 2015. URL https://research.google/blog/inceptionism-going-deeper-into-neural-networks/. Google Research Blog, accessed: 2024-06-20.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019. URL https://arxiv.org/abs/1912.01703.

William Peebles and Saining Xie. Scalable diffusion models with transformers, 2023.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis, 2023.

Senthil Purushwalkam, Akash Gokul, Shafiq Joty, and Nikhil Naik. Bootpig: Bootstrapping zero-shot personalized image generation capabilities in pretrained diffusion models, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

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

Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Mark Chu, Xueyan Chen, Pieter Abbeel, and Ilya Sutskever. Dall-e 3: Language models are few-shot image generators, 2023. URL `https://cdn.openai.com/papers/dall-e-3.pdf`.

Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis, 2016.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2022.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015.

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.

Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models, 2022.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, 2015.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models, 2022. URL `https://arxiv.org/abs/2010.02502`.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

Narek Tumanyan, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Splicing vit features for semantic appearance transfer, 2022.

Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, and Shengyi Huang. Trl: Transformer reinforcement learning. https://github.com/huggingface/trl, 2020.

Andrey Voynov, Qinghao Chu, Daniel Cohen-Or, and Kfir Aberman. P+: Extended textual conditioning in text-to-image generation, 2023.

Qixun Wang, Xu Bai, Haofan Wang, Zekui Qin, Anthony Chen, Huaxia Li, Xu Tang, and Yao Hu. Instantid: Zero-shot identity-preserving generation in seconds, 2024.

Yuxin Wen, Neel Jain, John Kirchenbauer, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery, 2023. URL https://arxiv.org/abs/2302.03668.

Charli XCX. 360. brat, 2024.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. Imagereward: Learning and evaluating human preferences for text-to-image generation, 2023.

Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models, 2023.

Hu Yu, Hao Luo, Fan Wang, and Feng Zhao. Uncovering the text embedding in text-to-image diffusion models, 2024.

Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, Ben Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. Scaling autoregressive models for content-rich text-to-image generation, 2022.

Yuechen Zhang, Jinbo Xing, Eric Lo, and Jiaya Jia. Real-world image variation by aligning diffusion inversion chain, 2023a.

Yuxin Zhang, Weiming Dong, Fan Tang, Nisha Huang, Haibin Huang, Chongyang Ma, Tong-Yee Lee, Oliver Deussen, and Changsheng Xu. Prospect: Prompt spectrum for attribute-aware personalization of diffusion models, 2023b.

Shihao Zhao, Shaozhe Hao, Bojia Zi, Huaizhe Xu, and Kwan-Yee K. Wong. Bridging different language models and generative vision models for text-to-image generation, 2024.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks, 2020.

A Appendix

A.1 Style Dataset

The style data can be found at https://huggingface.co/datasets/jlbaker361/stylization. A list of the artists, and links to their profiles across whatever platforms could be found, is as follows:

1. **Lois van Baarle** Deviantart: https://www.deviantart.com/loish; Instagram: https://www.instagram.com/loisvb; Personal Website: https://loish.net/

2. **Kerem Beyit** Deviantart: https://www.deviantart.com/kerembeyit; Instagram: https://www.instagram.com/kerembeyit
3. **sandara** Deviantart: https://www.deviantart.com/sandara
4. **yuumei** Deviantart: https://www.deviantart.com/yuumei; Personal Website: https://www.yuumeiart.com/
5. **Gabriel Picolo** Deviantart: https://www.deviantart.com/picolo-kun; Instagram: https://www.instagram.com/_picolo/
6. **Ilya Kuvshinov** Deviantart: https://www.deviantart.com/kuvshinov-ilya; Instagram: https://www.instagram.com/kuvshinov_ilya/
7. **Cryptid Creations** Deviantart: https://www.deviantart.com/cryptid-creations
8. **alicexz** Deviantart: https://www.deviantart.com/alicexz
9. **Atey Ghailan** Deviantart: https://www.deviantart.com/snatti89; Tumblr: https://snatti.tumblr.com/; Instagram: https://www.instagram.com/snatti89/
10. **cat-meff** Deviantart: https://www.deviantart.com/cat-meff
11. **Gonzalo Ordonez Arias** Deviantart: https://www.deviantart.com/genzoman; Instagram: https://www.instagram.com/mrgenzoman/; Tumblr: https://www.tumblr.com/genzoman
12. **Geoffroy Thoorens** Deviantart: https://www.deviantart.com/djahal; Instagram: https://www.instagram.com/djahal/?hl=en; Personal Website: https://djahalland.com/
13. **Shingo Matsunuma** Deviantart: https://www.deviantart.com/shichigoro756; Personal Website: https://shichigoro.com/en/home/
14. **Stjepan Sejic** Deviantart: https://www.deviantart.com/nebezial;
15. **Cyril Rolando** Deviantart: https://www.deviantart.com/aquasixio; Instagram: https://www.instagram.com/aquasixio/?hl=en; Tumblr: https://cyrilrolando.tumblr.com/
16. **Sophia von Yhlen** Deviantart: https://www.deviantart.com/fealasy; Instagram: https://www.instagram.com/fealasy/

These images are shown in figure 3

A.2 Prompts

Tables 4 and 5 list the training prompts used for the subjects and styles, respectively. Tables 6 and 7 list the evaluation prompts for subjects and styles, respectively.

a photo of a {}	a rendering of a {}	a cropped photo of the {}
the photo of a {}	a photo of a clean {}	a photo of a dirty {}
a dark photo of the {}	a photo of my {}	a photo of the cool {}
a close-up photo of a {}	a bright photo of the {}	a cropped photo of a {}
a photo of the {}	a good photo of the {}	a photo of one {}
a close-up photo of the {}	a rendition of the {}	a photo of the clean {}
a rendition of a {}	a photo of a nice {}	a good photo of a {}
a photo of the nice {}	a photo of the small {}	a photo of the weird {}
a photo of the large {}	a photo of a cool {}	a photo of a small {}

Table 4: Subject Prompts

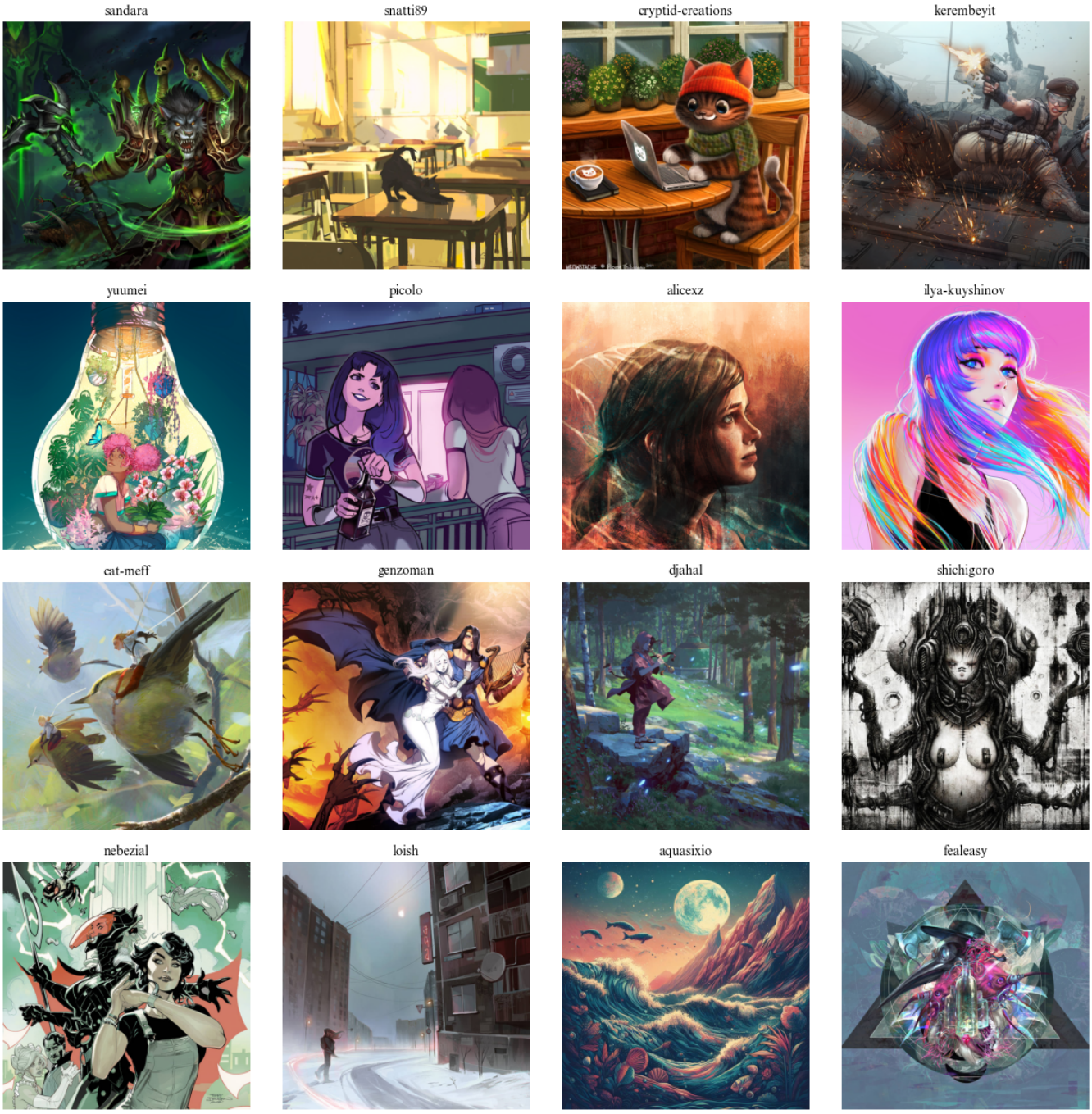

Figure 3: Source Images, labeled by their deviantart usernames

a painting, art by {}	a rendering, art by {}	a cropped painting, art by {}
the painting, art by {}	a clean image, art by {}	a dirty image, art by {}
a dark image, art by {}	an image, art by {}	a cool picture, art by {}
a close-up picture, art by {}	a bright picture, art by {}	a cropped picture, art by {}
a good painting, art by {}	a close-up painting, art by {}	a rendition, art by {}
a nice painting, in the style of {}	a small painting, in the style of {}	a weird painting, in the style of {}
a large painting, in the style of {}		

Table 5: Style Prompts

a photo of {} at the beach	a photo of {} in the jungle
a photo of {} in the snow	a photo of {} in the street
a photo of {} with a city in the background	a photo of {} with a mountain in the background
a photo of {} with the Eiffel Tower in the background	a photo of {} near the Statue of Liberty
a photo of {} near the Sydney Opera House	a photo of {} floating on top of water
a photo of {} eating a burger	a photo of {} drinking a beer
a photo of {} wearing a blue hat	a photo of {} wearing sunglasses
a photo of {} playing with a ball	a photo of {} as a police officer

Table 6: Subject Evaluation Prompt

the beach, art by {}	the jungle, art by {}
the snow, art by {}	the street, art by {}
a person with a city in the background, art by {}	a person with a mountain in the background, art by {}
the Eiffel Tower, art by {}	the Statue of Liberty, art by {}
the Sydney Opera House, art by {}	person floating on top of water, art by {}
eating a burger, art by {}	drinking a beer, art by {}
wearing a blue hat, art by {}	wearing sunglasses, art by {}
playing with a ball, art by {}	a police officer, art by {}

Table 7: Style Evaluation Prompts

A.3 Additional Images

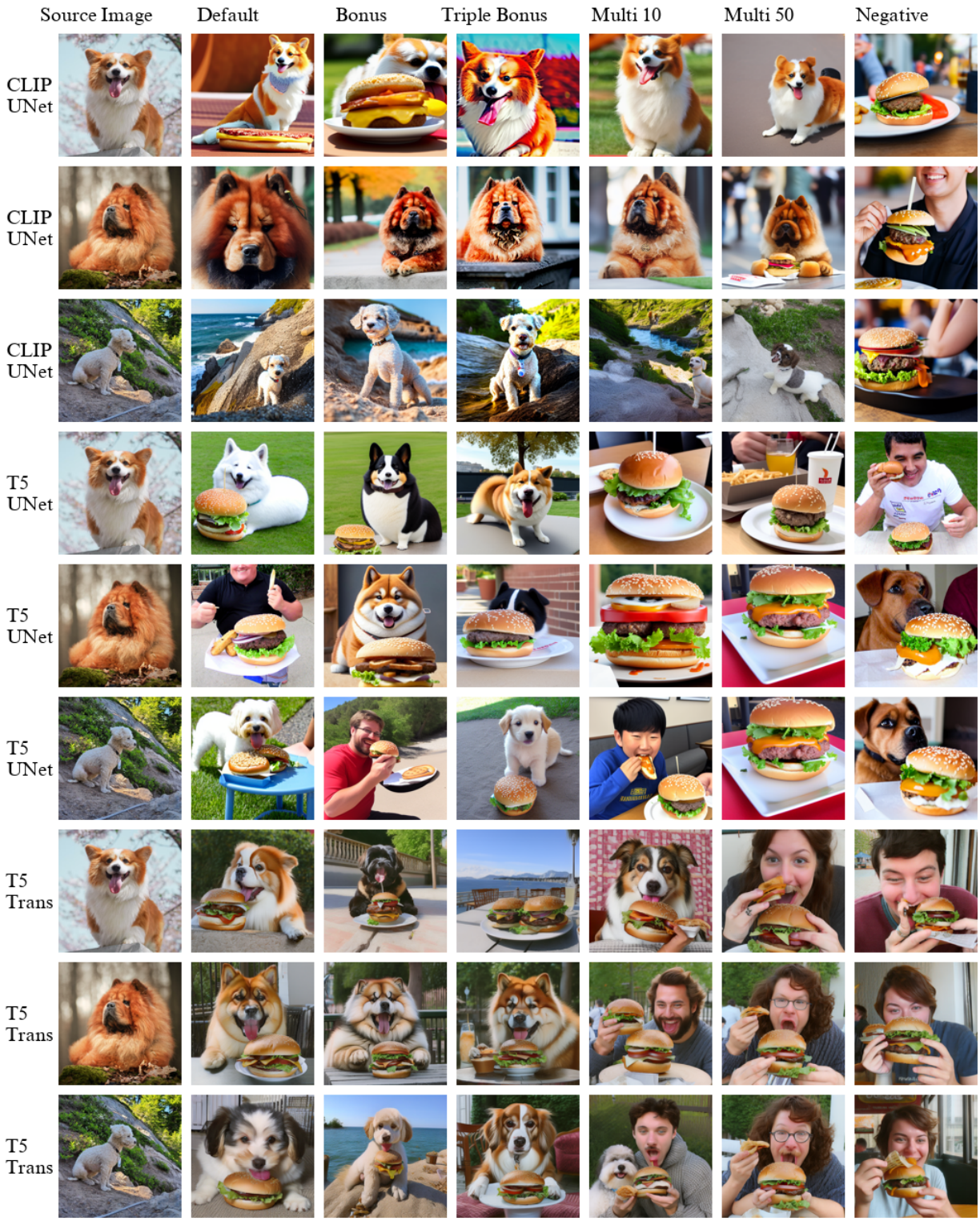

Figure 4: Images generated with the prompt "a photo of {} eating a burger"

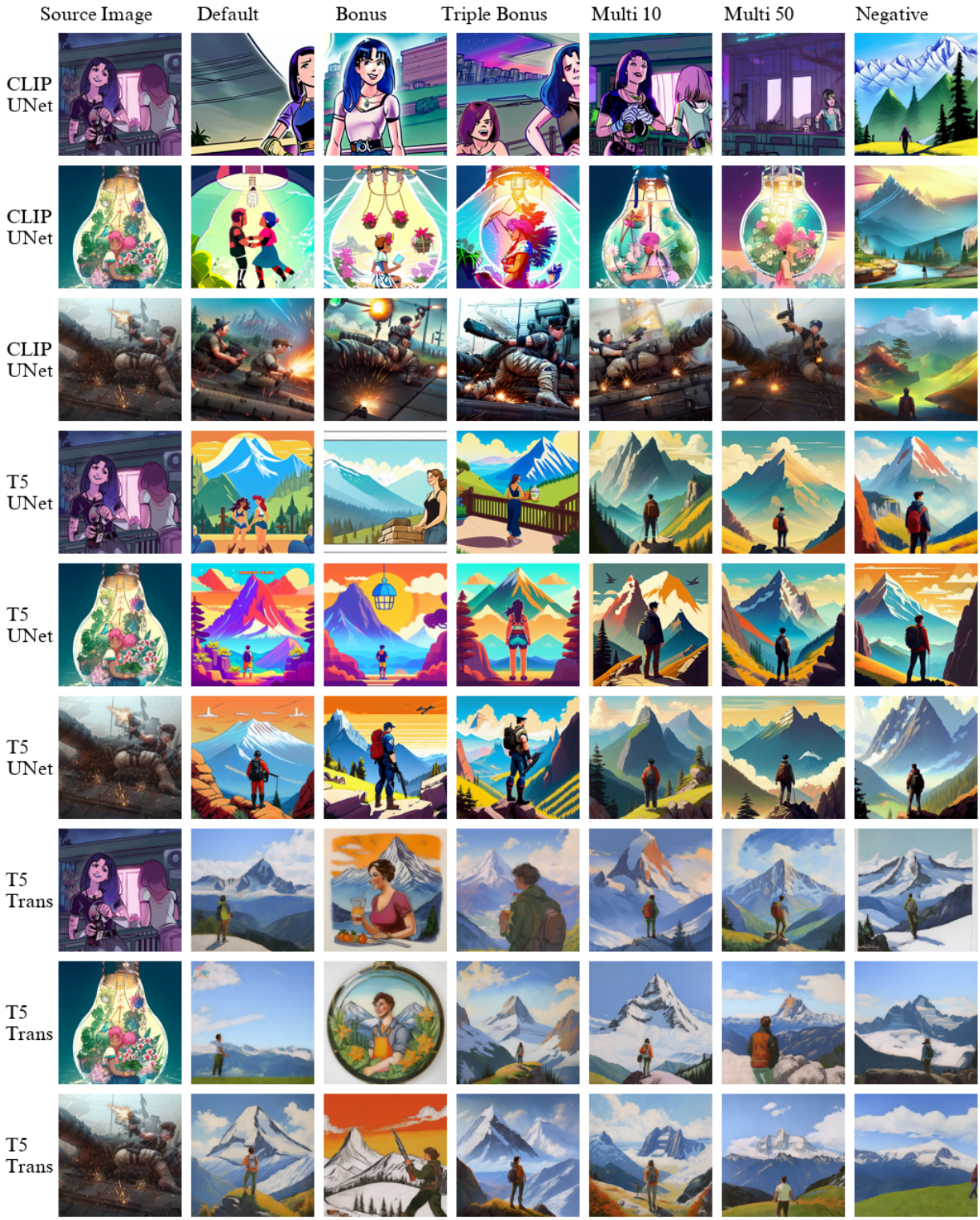

Figure 5: Images generated with prompt "a person with a mountain in the background, art by {}"

A.4 Training

All experiments were done using an A100 GPU with 40GB RAM. All code was written in Python 3.11, leveraging libraries such as PyTorch (Paszke et al., 2019), Diffusers (von Platen et al., 2022), TRL (von Werra et al., 2020), Accelerate (Gugger et al., 2022) and Wandb (Biewald, 2020).

A.4.1 Hyperparameters

Training Hyperparameters are listed in table 8. Training for the style and subject datasets used all of the same hyperparameters except for the number of epochs.

Parameter	Value
Epochs (Subjects)	250
Epochs (Styles)	500
Learning Rate	0.08
Gradient Accumulation Steps	8
Batch Size	1
Spare λ	0.01
Noise Scheduler	DDPM
Max Gradient Norm	10.0

Table 8: Hyperparameters

A.5 Failed Methods

We briefly experimented with large text encoders with more than a billion parameters. The **PixArt-XL-2-512x512** text encoder had roughly 4.7 billion parameters, and the **Llama-2-7b-hf** had roughly 6.7 billion parameters. We found these very difficult to train, failing to capture the subjects at all. We attempted training with both traditional textual inversion and using the spare token. We used 750 epochs, instead of 250. Nonetheless, we found unsatisfying results. Figures 6, 7, 8 and 9 show some examples. The left-most column of each figure shows the source images from the personalization dataset, and each row shows the image generated from the same prompt, where each column denotes a different combination of learning rate for training (0.08 or 0.4) and scheduler (DDIM (Song et al., 2022) or DDPM (Ho et al., 2020b)).

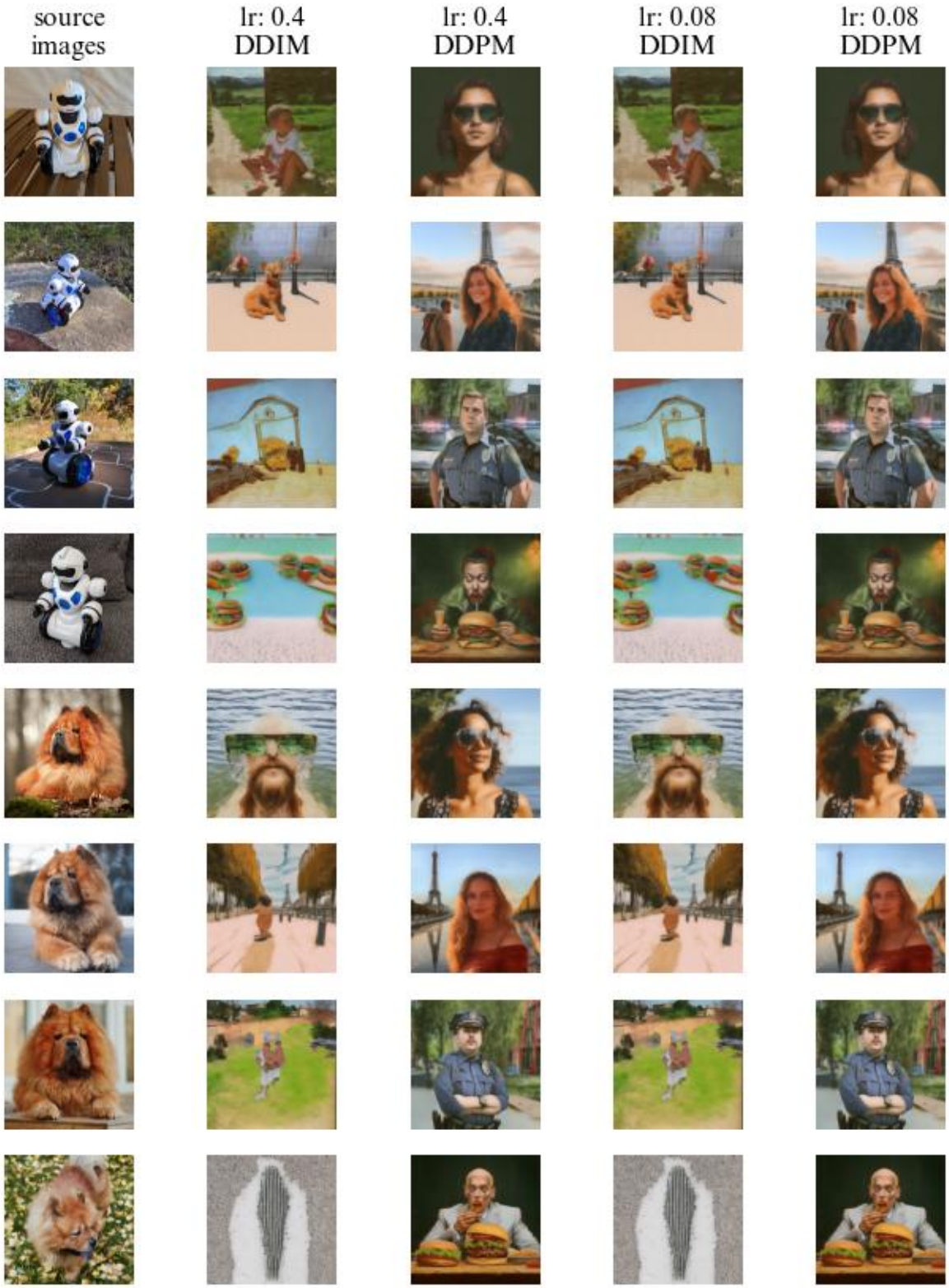

Figure 6: PixArt

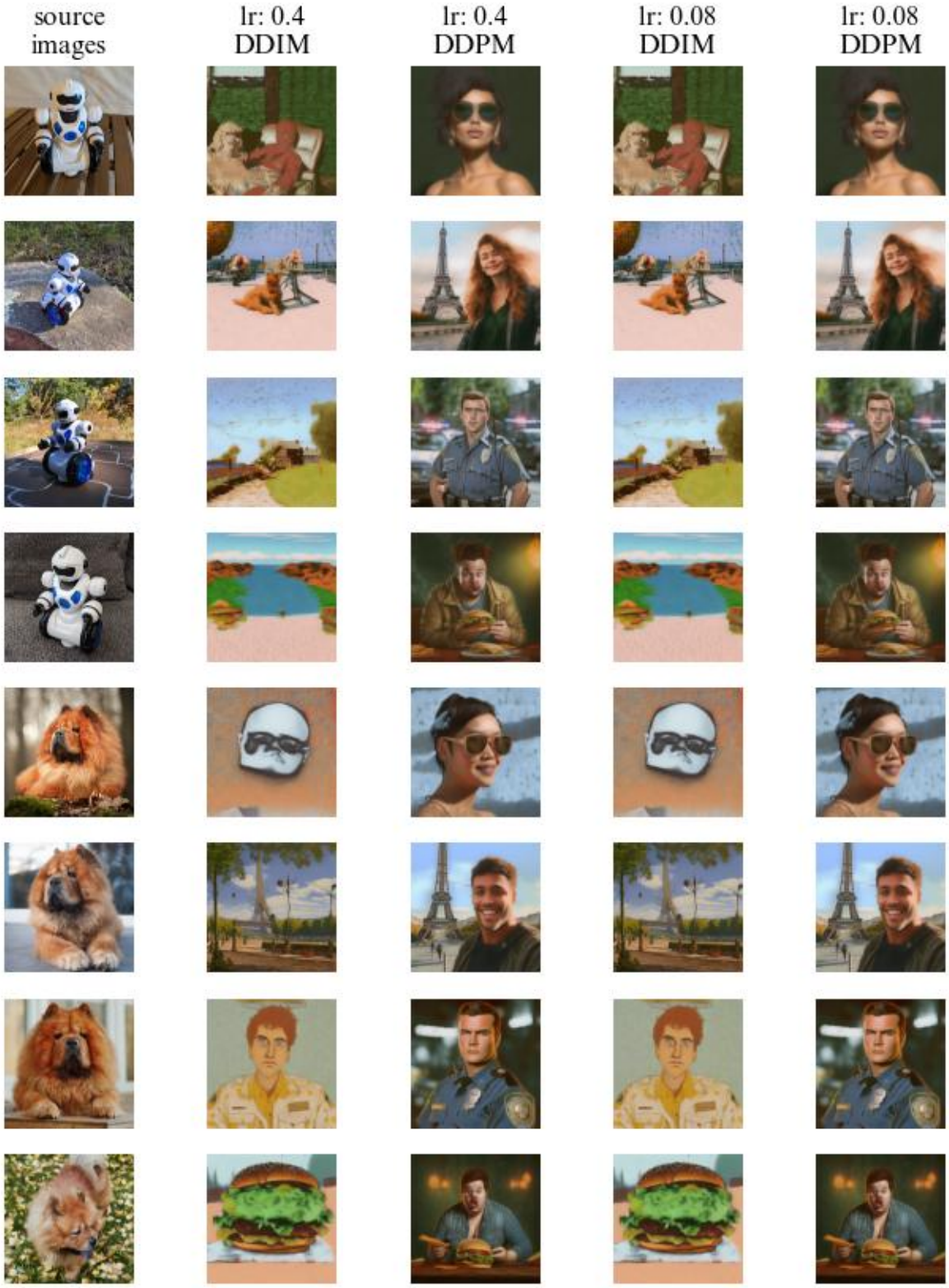

Figure 7: PixArt (With Bonus Token)

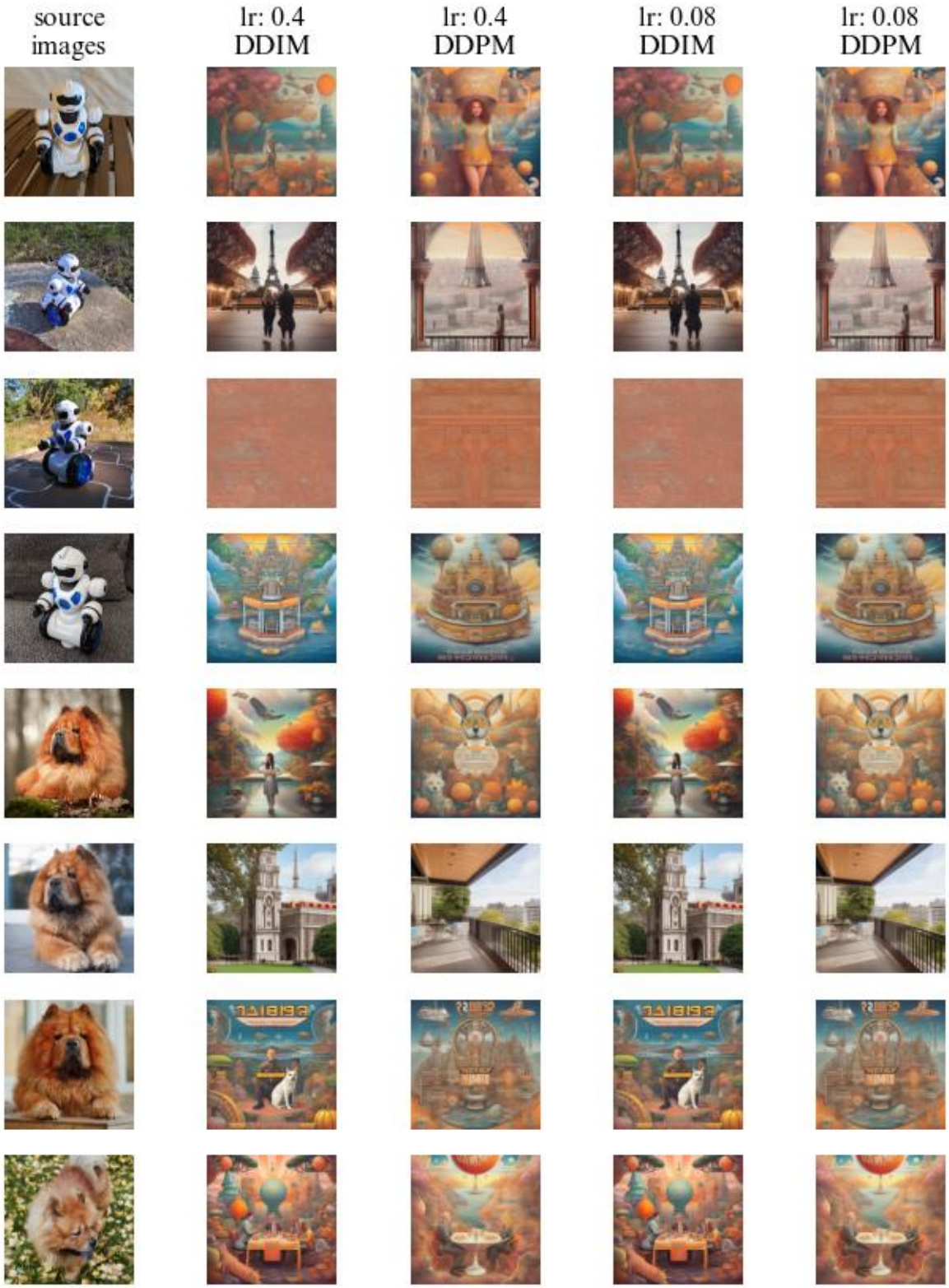

Figure 8: Llama

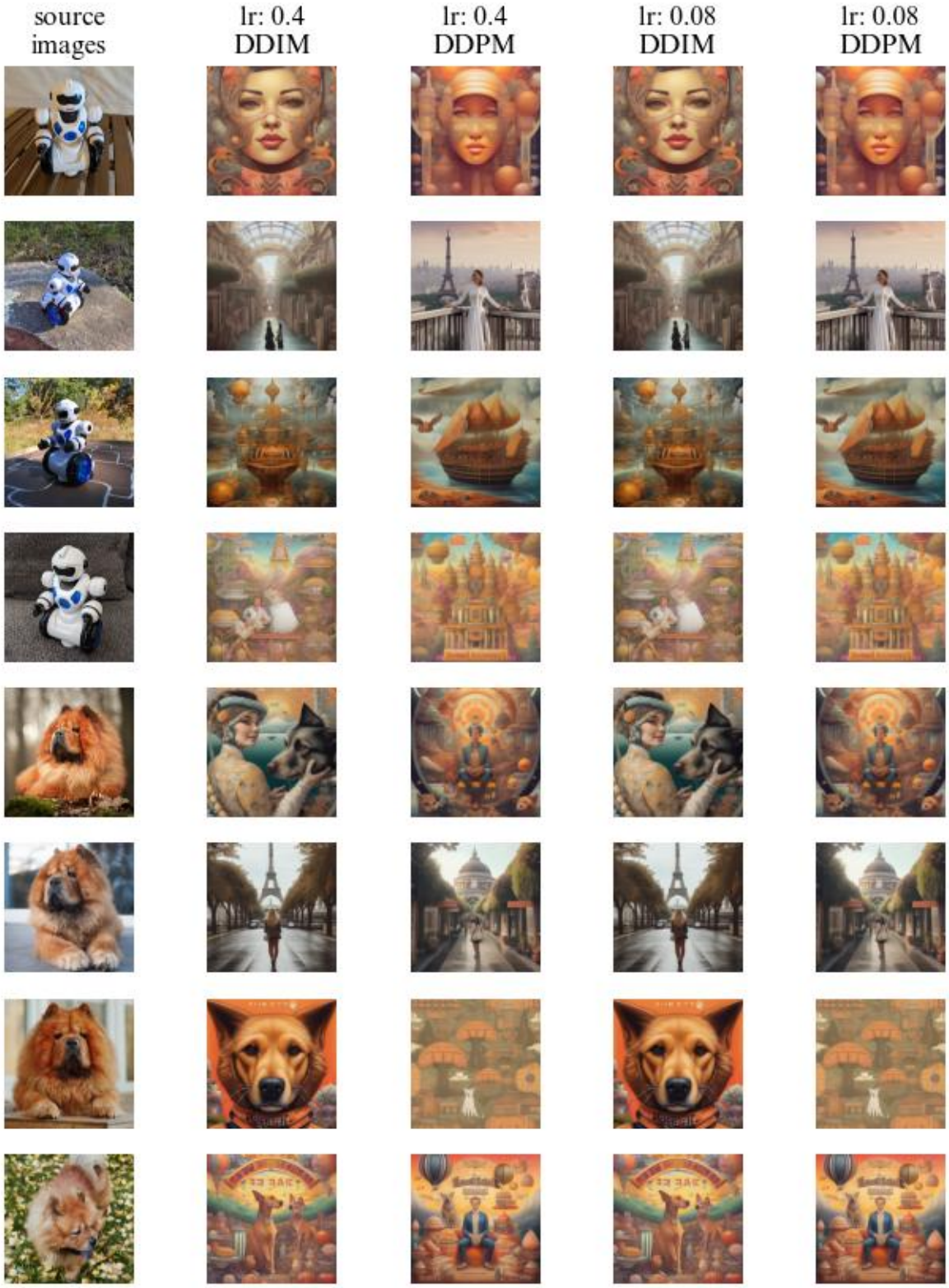

Figure 9: Llama (With Bonus Token)