Target Attribute Diffusion Models

William Loh
University of Waterloo
Vector Institute
wmloh@uwaterloo.ca

Yanting Miao
University of Waterloo
Vector Institute
y43miao@uwaterloo.ca

Pascal Poupart
University of Waterloo
Vector Institute
ppoupart@uwaterloo.ca

Suraj Kothawade
Google
skothawade@google.com

Abstract

Diffusion models have shown notable success in generating images conditioned on textual prompts, enabling users to edit images at a coarse scale with well-aligned text-to-image models. ControlNet [31] enhances these capabilities by allowing diffusion models to edit aspects such as pose, position, and edges according to reference visual motion information in a qualitative manner. However, diffusion models still face challenges in measurable and quantitative applications, such as applying sharpening or color enhancement effects. We call quantities such as brightness and saturation, ***attributes****. In this work, we introduce* ***Target Attribute Diffusion Models*** *(TADM), which enable diffusion models to incorporate additional conditioning on continuous random variables. Unlike classifier-guidance methods, which require training an explicit classifier [30], TADM supports real-valued conditional variables. We also propose a new architecture called* ***attribute carrier*** *between the text embeddings and the new conditioning variable. Experiments were conducted on three attributes: color saturation, sharpness and human preference. TADM outperformed the baseline algorithm on a single prompt, single attribute experiment. In addition, TADM demonstrates improvement in the multiple prompt experiments with respect to two of the three attributes.*

1. Introduction

With the advancement of text-to-image diffusion models [19–21], users can create high-quality images simply by typing text prompts. However, precisely controlling the properties of these models to achieve desired results remains a significant challenge. For instance, specifying details like color saturation, sharpness, and brightness can be difficult to convey through text alone. Generating images that meet users' exact requirements often requires numerous cycles of trial and error with prompt adjustments, leading to substantial time and resource consumption.

In this work, we use the term *attribute* to refer to real-valued properties, including color saturation, sharpness, and more. A natural question arises: how can we enable fine-grained attribute control in pre-trained text-to-image diffusion models by allowing users to provide a target attribute that directly specifies their desired properties? The attribute can represent various control goals, abstracted as a scalar reward measuring the quality of generated images. The machine learning community has already taken steps to leverage reinforcement learning for enhancing text-to-image attribute alignment [1, 6, 17], aligning with human/AI preference attributes [27], and subject-driven attributes [14]. There is also a training-free method for controlling motion attributes [7]; however, it is only suitable for specific tasks and requires careful design of the guidance function. In practice, most attribute-based tasks still require end-to-end fine-tuning.

Learning additional conditional controls for text-to-image diffusion models has been explored. ControlNet [31] introduces a trainable copy for each block in diffusion models to address the overfitting problem during fine-tuning, although this approach is memory-inefficient for attribute tasks. [30] proposes a reward classifier to guide generation; however, this method makes a strong assumption about the continuous reward random variable, specifically assuming a Gaussian distribution for the reward. Therefore, designing an efficient and expressive method is essential for handling the attribute problem.

This paper introduces Target Attribute Diffusion Models (TADM), a new adapter for text-to-image diffusion models. TADM comprises two key components: the attribute carrier and decoupled attribute cross-attention, which inject attribute information into the pre-trained diffusion models

while preserving the quality of the generated images. This new adapter freezes the parameters of the original model and trains only the parameters of the new modules. To avoid introducing disruptive noise during the initial training phase, the output of the decoupled attribute cross-attention is connected to a close-to-zero initialized output layer, with the weights progressively growing during training to maintain the high quality of the generated images.

There are two tiers on experiments. Experiment 1 consists of a single prompt experiment on the color saturation attribute. Experiment 2 extends by evaluating on multiple prompts, and was conducted with three attributes: color saturation, sharpness and human preference. In the color saturation and sharpness experiments, TADM shows better alignment compared to RCGDM [30], a baseline algorithm. For human preference, it shows the limits of both algorithms, where they cannot handle the complexity of human preference.

In summary, (1) we propose TADM, a new adapter for diffusion models that incorporates real-valued conditions; (2) we describe the attribute carrier, a new neural network architecture that combines information from different subspaces; and (3) we offer insights on the growth of norms in adapter modules, which we solve by introducing norm clipping.

2. Related Work

2.1. Diffusion Models

Text-to-Image Diffusion Models. Image diffusion models were introduced by [10, 22, 23]. The original diffusion models perform denoising steps in the image space. To address the computational cost, Rombach et al. [20] proposed Latent Diffusion Models (LDM), which transfer the denoising process into a latent space, significantly reducing computational requirements. SDXL [16] implements a large-scale version of LDM. Instead of diffusing in the latent space, Imagen [21] introduces a novel pyramid structure to directly perform denoising steps on pixels.

Controllable Diffusion Models. Dhariwal and Nichol [4] introduce an additional classifier to guide the diffusion steps, a method that outperforms BigGAN [3]. Ho and Salimans [9] propose classifier-free diffusion guidance, which eliminates the need for a separate classifier and simplifies the training process for diffusion models. In practice, classifier-guidance is preferred for tasks requiring additional conditioning. The works on reward-conditioned generation via diffusion model (RCGDM) [30] introduce a reward classifier to model the reward distribution. Motion-guidance [7] proposes a training-free method that designs a task-specific loss function and uses its gradient to guide generation. The classifier-free method can also be applied to additional conditional information. ControlNet [31] is designed to capture spatial information, including edges, segmentation, and human poses, from reference images. This method requires copying trainable modules for each block of the Stable Diffusion Models [20]. Ip-Adapter [29] and SSR-Encoder [32] leverage the pre-trained CLIP image encoder [18] to extract features from reference images and use decoupled cross-attention layers to guide generation based on these unique features. Instead of introducing new parameters, RPO [14] provides controllable generation for subject-driven tasks through preference-based reinforcement learning.

2.2. Fine-Tuning Foundation Models

Low-Rank Adaptation (LoRA). Hu et al. [11] propose the LoRA technique for fine-tuning large language models to mitigate catastrophic forgetting. Based on the observation that language models operate within a low intrinsic-dimensional subspace, LoRA alleviates overfitting by introducing parameter offsets represented by low-rank matrices. Additionally, LoRA addresses disruptive noise issues by initializing parameters using a Gaussian distribution with zero mean.

3. Background

Diffusion Models. Diffusion models [10, 22, 24] are a family of probabilistic models of the form $p_{\boldsymbol{\phi}}(\mathbf{x}_0) = \int p_{\boldsymbol{\phi}}(\mathbf{x}_{0:T}) d\mathbf{x}_{1:T}$, where $\mathbf{x}_{1:T}$ are noised latent variables of the same dimensionality as $\mathbf{x}_0 \sim p_{\text{data}}(\mathbf{x}_0)$. The diffusion process is a Markov chain that gradually adds Gaussian noise to the input data $\mathbf{x}_0$ according to a variance schedule β_t such that $q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t \mathbf{I})$, and thus, $q(\mathbf{x}_t \mid \mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\alpha_t}\mathbf{x}_0, (1-\alpha_t)\mathbf{I})$, where $\alpha_t = \prod_{i=1}^{t}(1-\beta_t)$. Therefore, $\mathbf{x}_t$ can be rewritten as $\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x}_0 + \sqrt{1-\alpha_t}\boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. A variational Markov chain in the reverse direction is parameterized with $p_{\boldsymbol{\phi}}(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}\left(\mathbf{x}_{t-1}; \frac{1}{\sqrt{1-\beta_t}}\left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\alpha_t}}\boldsymbol{\epsilon}_{\boldsymbol{\phi}}(\mathbf{x}_t)\right), \beta_t \mathbf{I}\right)$ and $\boldsymbol{\epsilon}_{\boldsymbol{\phi}}(\mathbf{x}_t)$ is trained by a re-weighted evidence lower bound (ELBO):

$$\min_{\boldsymbol{\phi}} \mathbb{E}_{\mathbf{x}_0, t, \boldsymbol{\epsilon}} \left[w(t) \|\boldsymbol{\epsilon}_{\boldsymbol{\phi}}(\mathbf{x}_t) - \boldsymbol{\epsilon}\|_2^2 \right], \tag{1}$$

where $t \sim \mathcal{U}\{1, \ldots, T\}$. In practice, $w(t)$ can be simplified as 1 according to [10, 23].

Classifier-free Guidance. The classifier free guidance [9, 15] approximates samples from the distribution

$$\tilde{q}(\mathbf{x}_t \mid \mathbf{c}) \propto q(\mathbf{x}_t) q^s(\mathbf{c} \mid \mathbf{x}_t),$$

where $s \geq 1$ is the guidance scale. The score function of the implicit classifier [4], $q(\mathbf{c} \mid \mathbf{x}_t)$, can be represented as

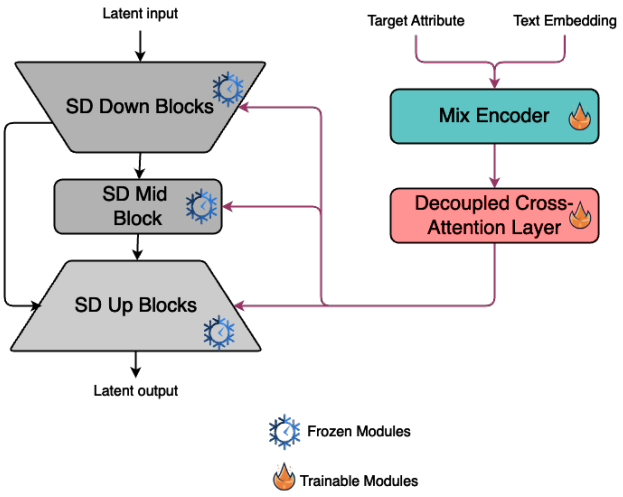

Figure 1. The model architecture of TADM. Only the new modules (indicated by the flame icon) are trained while the pretrained UNet modules are frozen.

$$\nabla_{\mathbf{x}_t} \log q(\mathbf{c} \mid \mathbf{x}_t) = \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t \mid \mathbf{c}) - \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t) \propto \boldsymbol{\epsilon}_{\phi}(\mathbf{x}_t, \mathbf{c}) - \boldsymbol{\epsilon}_{\phi}(\mathbf{x}_t, \emptyset),$$

where $\boldsymbol{\epsilon}_{\phi}$ is the learned diffusion models and $\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, \emptyset) = \boldsymbol{\epsilon}_{\phi}(\mathbf{x}_t)$. During sampling, the denoised step is extrapolated in the direction of $\boldsymbol{\epsilon}_{\phi}(\mathbf{x}_t, \mathbf{c})$ and away from $\boldsymbol{\epsilon}_{\phi}(\mathbf{x}_t)$ as follows:

$$\tilde{\boldsymbol{\epsilon}}_{\phi}(\mathbf{x}_t, \mathbf{c}) = \boldsymbol{\epsilon}_{\phi}(\mathbf{x}_t) + s \cdot (\boldsymbol{\epsilon}_{\phi}(\mathbf{x}_t, \mathbf{c}) - \boldsymbol{\epsilon}_{\phi}(\mathbf{x}_t)) \quad (2)$$

4. Method

A Target Attribute Diffusion Model (TADM) is an architecture that enables large pretrained text-to-image diffusion models to incorporate scalar conditions and Figure 1 shows the overall architecture of TADM. In Section 4.1, we first introduce the new architecture's core component, the *attribute carrier*. Next, we describe how to apply this structure using attribute-decoupled cross-attention within pretrained diffusion models, such as Stable Diffusion [20], in Section 4.2. Finally, in Section 4.3, we provide details on the training process for TADM.

4.1. Attribute Carrier

An attribute carrier can be described as a mapping from a query and an attribute to a latent output, where both the query and attribute are tensors (Figure 2). In practice, the input is represented as a query $\mathbf{Q} \in \mathbb{R}^{L \times d}$ and an attribute $\mathbf{A} \in \mathbb{R}^{d}$. Similar to [26], we compute the scaled dot-product of the query with the attribute and apply the sigmoid function to obtain the *update gate*:

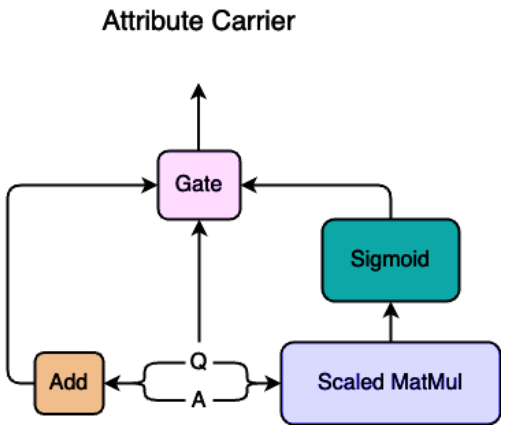

Figure 2. Overview of attribute carrier

$$\mathbf{G} = \text{Sigmoid}\left(\frac{\mathbf{Q}\mathbf{A}^{\top}}{\sqrt{d}}\right). \quad (3)$$

The new query, which incorporates the attribute information, is denoted as Q' and is defined through broadcast element-wise addition:

$$\mathbf{Q}' = \mathbf{Q} + \mathbf{A}. \quad (4)$$

The gate function is defined as a convex combination of the new query and the original query using the update gate:

$$\text{Gate}(\mathbf{Q}, \mathbf{Q}', \mathbf{G}) := \mathbf{G} \odot \mathbf{Q}' + (1 - \mathbf{G}) \odot \mathbf{Q}, \quad (5)$$

where $\odot$ denotes element-wise multiplication.

Intuitively, the elements in the update gate $\mathbf{G}$ are independent Bernoulli-distributed random variables. The update gate only updates to the new query if the original query is highly correlated with the attribute tensor. Formally, the attribute carrier can be written as

$$\text{AttributeCarrier}(\mathbf{Q}, \mathbf{A}) := \text{Sigmoid}\left(\frac{\mathbf{Q}\mathbf{A}^{\top}}{\sqrt{d}}\right) \odot (\mathbf{Q} + \mathbf{A}) + \left(1 - \text{Sigmoid}\left(\frac{\mathbf{Q}\mathbf{A}^{\top}}{\sqrt{d}}\right)\right) \odot \mathbf{Q} \quad (6)$$

4.2. Attribute Carrier for Diffusion Models

To incorporate the attribute carrier into diffusion models, we design a simple architecture called the *mix encoder* (Figure 3). The feed forward network maps the target attribute from $\mathbb{R}$ to a high-dimensional space $\mathbb{R}^{D}$, and we denote the attribute embedding as $\mathbf{A} \in \mathbb{R}^{D}$. The text embedding, $\mathbf{c} \in \mathbb{R}^{L \times d_{\mathbf{c}}}$, serves as the query feature. The mix-up embedding is then represented as:

$$\mathbf{c}' = \text{AttributeCarrier}(\mathbf{c}\mathbf{W}_{\mathbf{c}}, \mathbf{A}\mathbf{W}_{\mathbf{A}}), \quad (7)$$

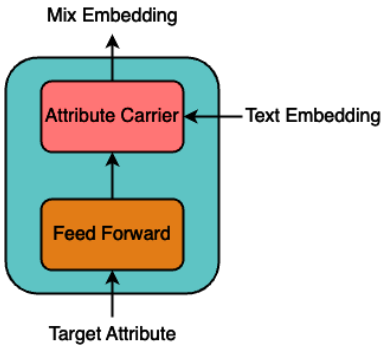

Figure 3. Model architecture for the mix encoder.

where $\mathbf{W_c} \in \mathbb{R}^{d_\mathbf{c} \times d}$ and $\mathbf{W_A} \in \mathbb{R}^{d_\mathbf{A} \times d}$. These matrices project tensors from two different spaces into the same subspace. Consequently, the output $\mathbf{c}'$ is a text embedding that incorporates the target attribute information, and this mix-up embedding is fed into the models. The decoupled attribute cross-attention layers process this new embedding tensor. Figure 4 illustrates the mechanism of the decoupled attribute cross-attention layer. Mathematically, given a query $\mathbf{z}$, a text embedding $\mathbf{c}$, and a mix-up embedding $\mathbf{c}'$, the output of the decoupled attribute cross-attention layer is defined as:

$$\begin{aligned} \mathbf{z}' := \mathbf{W}_o \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V} \\ +\lambda \cdot \mathbf{W}_{o'} \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}'^\top}{\sqrt{d_k}}\right)\mathbf{V}', \\ \text{where } \mathbf{Q} = \mathbf{z}\mathbf{W}_q, \mathbf{K} = \mathbf{c}\mathbf{W}_k, \mathbf{V} = \mathbf{c}\mathbf{W} \\ \mathbf{K}' = \mathbf{c}'\mathbf{W}'_k, \mathbf{V}' = \mathbf{c}'\mathbf{W}'_v, \end{aligned} \tag{8}$$

and d_k is the inner dimension of the cross-attention layers. For simplicity, we share the same inner dimension with the pretrained cross-attention layers. To ensure the model retains its ability to generate text-to-image-aligned images, we freeze the parameters of the pretrained cross-attention layers, making only the matrices $\mathbf{W}'_k$, $\mathbf{W}'_v$, and $\mathbf{W}'_o$ trainable.

While c' serves to incorporate attribute information towards the image generation process, it should not cause catastrophic forgetting. Therefore, the outputs from the attribute cross-attention layers are clipped such that their norms cannot exceed a predefined value, which is set as a hyperparameter.

4.3. Training

To address the potential influence of disruptive noise on the hidden states of the cross-attention layers, we initialize $\mathbf{W}_o$ with values close to zero to ensure that minimal noise is added to the features at the start of fine-tuning. During training, we use a dataset comprising image-text-attribute pairs, i.e., $\mathcal{D} = (\mathbf{x}_0^{(i)}, \mathbf{c}^{(i)}, a^{(i)})_{i=1}^N$, and employ the same loss function as standard diffusion models:

$$\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_{\mathbf{x}_0,\mathbf{c},a,\boldsymbol{\epsilon},t}\left[\|\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, \mathbf{c}, a) - \boldsymbol{\epsilon}\|^2\right]. \tag{9}$$

Additionally, we randomly drop attributes and prompts during fine-tuning to enable the learned model to perform classifier-free guidance:

$$\tilde{\boldsymbol{\epsilon}}_{\boldsymbol{\theta}}(\mathbf{x}_t, \mathbf{c}, a) = \boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t) + s \cdot (\boldsymbol{\epsilon}_{\boldsymbol{\theta}}(\mathbf{x}_t, \mathbf{c}, a) - \epsilon_{\boldsymbol{\theta}}(\mathbf{x}_t)). \tag{10}$$

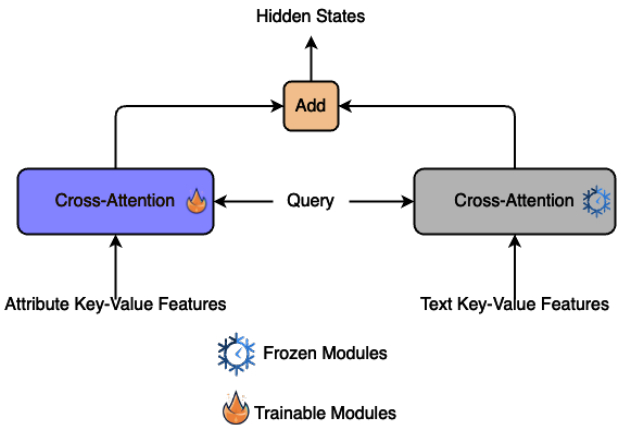

Figure 4. Overview of decoupled attribute cross attention layer. Only the module with the flame icon will be trained.

5. Experiments

This section will demonstrate the relative performance between our method (TADM) and reward-conditioned generation via diffusion model (RCGDM) by Yuan et al. [30]. There will be four sets of experiments: Experiment 1 evaluates color saturation on a single prompt. Experiment 2 evaluates with multiple prompts on three attributes: (a) color saturation, (b) sharpness, and (c) human preference.

Experiment 1 is meant to demonstrate the capabilities of TADM on a simpler task where it is finetuned and tested on a single prompt and attribute. Experiment 2 is intended to show generalization toward multiple prompts and multiple attributes, which is a significantly more difficult task.

5.1. Attribute Measures

The experiments use three attributes: color saturation, sharpness and human preference. Color saturation and sharpness attribute measures can be mathematically defined. Let x be an input image, and can be decomposed into its corresponding RGB channels x_R, x_G, x_B. Color saturation, denoted by f_c, is

$$f_c(x) = 1 - \min(\rho(x_R, x_G), \rho(x_G, x_B), \rho(x_R, x_B)) \tag{11}$$

Figure 5. Example images with different color saturations, where the f_c of the left image is 0.21 while the f_c of the right image is 0.68 [28].

where ρ is the Pearson correlation coefficient between the two channels over the pixels. Intuitively, if the RGB channels are perfectly positively correlated, i.e. $\rho(a, b) = 1$ for all pairs of channels a, b, this implies that all channels are of the same pixel values hence it must be grayscale. Therefore, this leads to $f_c = 0$. On the other hand, if there is a pair that is not correlated or negatively correlated, this leads to a small ρ which means f_c will be large. Note that a high f_c does not necessarily mean that it is colorful, but could be monochromatic with one particular intense shade of color. Two example images are shown in Figure 5.

Sharpness f_s is defined to be the average central difference of all pixels of an image converted to grayscale. This is implemented using NumPy's `gradient` function [25]. Intuitively, this measures the average rate of change in all local, spatial regions of an image, which can determine how sharp or blurry the image is.

The measure of human preference f_p comes from a pretrained ImageReward model by Xu et al. [28]. It was trained with a preference learning objective, and when given an image, it can produce a numeric score.

5.2. Experiment Configurations

Both methods use Stable Diffusion 2.1 [20] as the base model. TADM requires finetuning of the new modules, while RCGDM needs to train a classifier to predict an attribute of an input image. To achieve this, we use the Cats Vs. Dogs dataset [5] for Experiment 1, and the ImageReward dataset called ImageRewardDB 8K by [28] for Experiment 2. The dataset is augmented such that it becomes a set of tuples containing the image, prompt, and the corresponding attribute value.

The datasets include some minor editing processes to improve the diversity of attributes. To improve the diversity of f_c, we randomly enhance the color saturation of the sampled images. Specifically, we use `ImageEnhance.Color` from Python Imaging Library (PIL) [2] with the enhancement factor selected uniformly between 1 and 2. To improve the diversity of f_s, we use

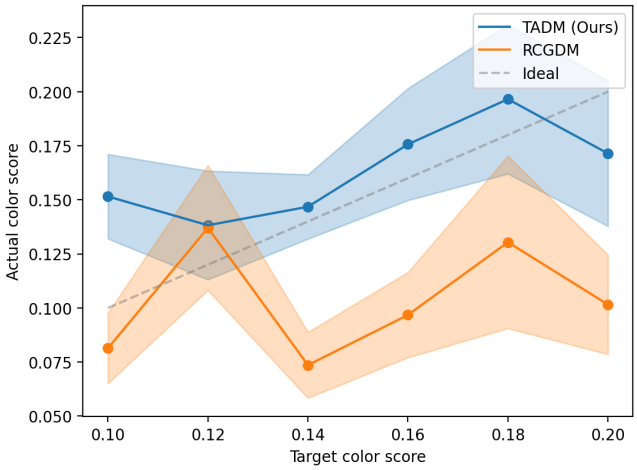

Figure 6. Experiment 1: Alignment of color saturation attribute for TADM and RCGDM.

`ImageFilter.GaussianBlur` from PIL with the radius of blur selected from an Exponential distribution with a scale of 0.6. We do not edit images for the human preference attribute.

The ImageReward dataset consists of 8,000 prompts [28] that will be used for training. The prompts used for validation and testing are manually constructed and listed in Table 1. At each validation and testing step, four images are generated for each target attribute value. Experiment 1 does not have validation since there is a single prompt experiment.

In Experiment 1, the target color saturation attribute values are 0.1, 0.12, 0.14, 0.16, 0.18, and 0.2 which are chosen based on the distribution of attribute values for cats in the dataset. The target values are chosen separately for each attribute in Experiment 2. For the color saturation attribute, the targets are 0.3, 0.4, 0.5, and 0.6. For the sharpness attribute, the targets are 0.8, 1.0, 1.2, and 1.4. For the human preference attribute, the targets are 0.8, 0.9, 1.0, and 1.1.

Recall that in Section 4.2, the attribute cross-attention outputs are clipped to a predefined value. The maximum norm is set to 0.1 for Experiment 1, and 0.2 for all models in Experiment 2. For RCGDM, we used the default classifier architecture to learn the mapping from image to attribute values as provided in their GitHub page [30]. The guidance strength (for the gradient flows) is set to 200.

5.3. Experimental Results

Experimental data contains the degree of alignment between the given target attribute value and the actual attribute value obtained from the resultant generated image. In general, the ideal trend is where the actual attribute value is equal to the target attribute value, forming a perfect diagonal $y = x$ graph.

However, it is important to note that it is a very difficult

Table 1. Validation and testing prompts for all attributes.

Validation prompts	Testing prompts
An astronaut with a galaxy background	A scuba diver in the ocean
A retro vintage bar	A cozy living room of wooden cabin
Roman soldiers at the Siege of Carthage	World war 2 soldiers in the trenches
A white unicorn in a fantasy meadow	A pegasus flying over grassland
A realistic painting of a Japanese village	A hyperrealistic painting of a medieval city

Figure 7. Sample cat images from TADM (left) and RCGDM (right).

task for any generative model to perfectly align to the diagonal line. This is essentially an interpolation task where the output space contains images. A more extensive related discussion can be found in Section 6.1. As such, one should primarily treat the diagonal line as a guide for the ratio of the y-axis scale to the x-axis scale, and not exclusively gauge the performance relative to that diagonal line but also to another baseline algorithm.

Experiment 1 The prompt is fixed to be "A photo of a cat". To effectively illustrate the relative improvement of TADM over RCGDM, we summarize them by showing the mean color saturation score (over 20 images) and its corresponding standard error in Figure 6. We can see that TADM aligns significantly better and is more monotonically increasing than RCGDM.

Sample generated cat images can be viewed in Figure 7. One image is sampled per target value for each method. In general, TADM has slightly more distortion in its generated images, resulting in a minor grainy texture. However, this might be a natural artefact of guiding the denoising process since TADM has a greater color diversity than RCGDM.

Experiment 2 Figure 8 shows the mean and standard error of all 20 images (5 prompts, 4 images per prompt) for each target attribute value.

In a multi-prompt experiment setting, the performance of both methods is not as good as in Experiment 1. Nevertheless, in relative terms, TADM shows better alignment than RCGDM in the color saturation experiments despite having a slight bias towards higher color saturation than intended. In the sharpness experiment, while TADM lies closer to the ideal trend than RCGDM, both do not demonstrate much variance across the target sharpness. As for the human preference attribute, TADM and RCGDM are generally moving in the right direction, although there is a slight bias towards lower scores. Also, there is a greater uncertainty here compared to the other two attributes.

To concretely assess the generated images, Figures 9 and 10 show the images from TADM and RCGDM that best align with the target color saturation values for a given prompt. RCGDM has better color alignment than our method, but notice that RCGDM varies the color saturation by adding unnatural artefacts to boost color saturation. This is most noticeable in the rightmost image in Figure 10. TADM, instead, boosts the colors mainly in the background to vary the color saturation values, and in the rightmost image of Figure 9, plausibly augments the colors of the flippers and hands. In general, TADM preserves image quality better while aligning the attributes.

Since the other two attributes have low variance with respect to the target values, we instead compare the highest attained values between TADM and RCGDM. Figure 11 shows the illustration for the sharpness attribute f_s, and that prompt was chosen because it has a noticeable difference between TADM and RCGDM. The image from RCGDM is arguably more aesthetic-looking but fails to produce a sharper image. TADM was able to produce a sharper image by adding rougher textures onto the dirt. As for human preference, there is not much that is visually discernible in terms of actual attribute values.

6. Discussions

This section highlights insights and analyses that extend beyond the tasks investigated in the experiments.

6.1. Relationship between Attribute and Prompt

Attributes and prompts provide the diffusion model with additional context, reducing the search space. Textual prompts

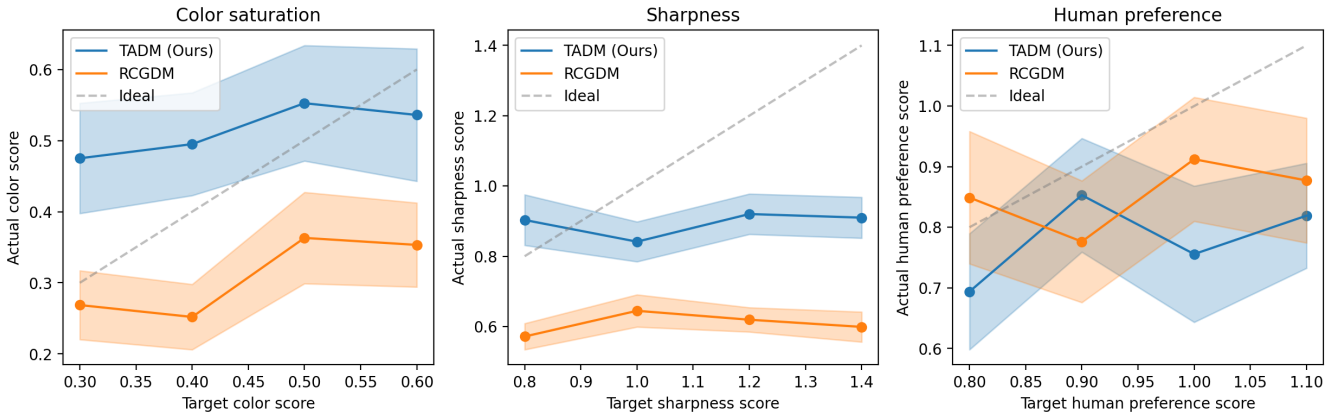

Figure 8. Experiment 2: Color saturation, sharpness, and human preference alignment results for TADM and RCGDM.

Figure 9. Sample TADM generated images for "A scuba diver in the ocean" (color saturation attribute). From left to right, the targets are 0.3, 0.4, 0.5, and 0.6 while the attained attribute values are 0.195, 0.352, 0.419, and 0.621.

Figure 10. Sample RCGDM generated images for "A scuba diver in the ocean" (color saturation attribute). From left to right, the targets are 0.3, 0.4, 0.5, and 0.6 while the attained attribute values are 0.288, 0.324, 0.488, and 0.612.

provide a holistic and subjective desiderata of the generated image while attribute values provide precise and quantifiable specifications of the generated image. Both inputs provide different types of control over the diffusion model, but there could be unintended contradictory signals.

For example, consider when the prompt is "A colorful rainbow" but the color saturation attribute is set to 0. This sends a contradictory message to the diffusion model and the output would highly depend on the distribution of images that it was trained on.

While this is an extreme example, this effect can be prevalent in less well-defined attributes such as human preference. People may inherently dislike a particular subject but it is highly associated with a subject found in the input prompt. As a result, it is highly non-trivial to increase human preference scores by removing subjects not mentioned in the prompts.

Figure 11. Sample TADM (left) and RCGDM (right) images for "World war 2 soldiers in the trenches". Both images are selected such that they have the highest f_s across all images for that prompt (independent on the targets). The TADM image has f_s of 1.558. The RCGDM image has f_s of 1.025.

In TADM, we offer a lightweight solution for adding a numeric conditioning input by attaching a new module to a pretrained Stable Diffusion model. As a result, the model is likely to favour the prompt in the event of a contradiction since the norm of the attribute cross-attention output is clipped but most parameters in the diffusion model have been optimized towards text-image alignment tasks. It might require a complete retraining of all parameters towards this objective, but that is beyond the scope of our paper.

6.2. Norm Clipping

As discussed in Section 4.2, the resultant attribute output should have its norm clipping prior to the addition. Here, we can verify the hypothesis that this clipping is in fact necessary. Figure 12 shows the growth of attribute output norm (orange) over the gradient steps. With norm clipping, the maximum norm of the output is 0.4.

In terms of the effect of image generation, experiments

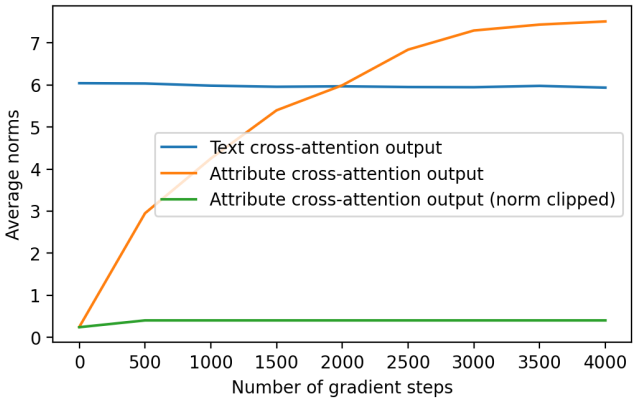

Figure 12. Growth of attribute cross-attention norm relative to the text cross-attention norm.

without a bounded norm always produce images that are incomprehensible within a few hundred gradient steps. This is likely the result of the attribute output greatly interfering with the original text output.

An interesting perspective on norm clipping comes from the lens of adversarial machine learning. Adversarial attacks are typically limited by adversarial budgets [13], and these budgets are typically small. Take the fast gradient sign method by Goodfellow et al. [8] for example. The attack simply involves an addition between an input and a specially crafted vector with a small norm.

In this paper, we leverage a similar principle to induce an additional conditioning variable for an image generation task. The attribute cross-attention output, essentially bounded by an ε-ball, aims to perturb the text cross-attention output such that it sufficiently influences the outcome of the generated image without causing too much internal covariate shift [12].

6.3. Limitation and Future Works

The experiments were conducted on the ImageReward dataset [28] which contains image-prompt-human rating triplets. While this dataset was originally selected for having human rating targets, it contains synthetic images generated by Stable Diffusion [28] with fairly unnatural prompts and varying image quality. Future works should utilize a more varied non-synthetic dataset for the purpose of aligning attributes with images and texts. In addition, due to our limited resources, we are unable to perform a complete fine-tuning of all parameters in the pretrained Stable Diffusion model but that would be future work.

7. Conclusion

Diffusion models have been lacking fine-grained control on attributes that can be numerically measured, and in ways that textual prompts cannot precisely guide the image generation process. Our work aims to rectify that shortcoming and to generalize that control to any numeric attribute and any textual prompt. By introspecting the current literature on adapter modules, we identified gaps in different areas, from the constraints on conditioning variables to the growth of norms in the cross-attention outputs of adaptors modules. TADM has shown success in a single prompt, single attribute experiment, and shown promise in the more generalized case with multiple prompts on different attributes.

8. Acknowledgements

We thank Eugene Ie and Hongliang Fei for providing constructive feedback. This work was supported by a Google grant with Cloud TPUs from Google's TPU Research Cloud (TRC). In addition, resources used in preparing this research were provided, in part, by the Province of Ontario, the Government of Canada through CIFAR, and companies sponsoring the Vector Institute `www.vectorinstitute.ai/#partners`.

References

[1] Kevin Black, Michael Janner, Yilun Du, Ilya Kostrikov, and Sergey Levine. Training diffusion models with reinforcement learning. In *The Twelfth International Conference on Learning Representations*, 2024. 1

[2] Alex Clark et al. Pillow (pil fork) documentation. *readthedocs*, 2015. 5

[3] Google Deepmind. Find pre-trained models. `https://tfhub.dev/deepmind/`. Accessed: 2024-11-13. 2

[4] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 2

[5] Jeremy Elson, John (JD) Douceur, Jon Howell, and Jared Saul. Asirra: A captcha that exploits interest-aligned manual image categorization. In *Proceedings of 14th ACM Conference on Computer and Communications Security (CCS)*. Association for Computing Machinery, Inc., 2007. 5

[6] Ying Fan, Olivia Watkins, Yuqing Du, Hao Liu, Moonkyung Ryu, Craig Boutilier, Pieter Abbeel, Mohammad Ghavamzadeh, Kangwook Lee, and Kimin Lee. Reinforcement learning for fine-tuning text-to-image diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024. 1

[7] Daniel Geng and Andrew Owens. Motion guidance: Diffusion-based image editing with differentiable motion estimators. *arXiv preprint arXiv:2401.18085*, 2024. 1, 2

[8] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 8

[9] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 2

[10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2

[11] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 2

[12] Sergey Ioffe. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 8

[13] Chen Liu, Mathieu Salzmann, Tao Lin, Ryota Tomioka, and Sabine Süsstrunk. On the loss landscape of adversarial training: Identifying challenges and how to overcome them. *Advances in Neural Information Processing Systems*, 33:21476–21487, 2020. 8

[14] Yanting Miao, William Loh, Suraj Kothawade, Pascal Poupart, Abdullah Rashwan, and Yeqing Li. Subject-driven text-to-image generation via preference-based reinforcement learning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 1, 2

[15] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 2

[16] 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. *arXiv preprint arXiv:2307.01952*, 2023. 2

[17] Mihir Prabhudesai, Anirudh Goyal, Deepak Pathak, and Katerina Fragkiadaki. Aligning text-to-image diffusion models with reward backpropagation. *arXiv preprint arXiv:2310.03739*, 2023. 1

[18] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 2

[19] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1 (2):3, 2022. 1

[20] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2, 3, 5

[21] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S. Sara Mahdavi, Rapha Gontijo Lopes, Tim Salimans, Jonathan Ho, David J Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding, 2022. 1, 2

[22] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015. 2

[23] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2

[24] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 2

[25] Stefan Van Der Walt, S Chris Colbert, and Gael Varoquaux. The numpy array: a structure for efficient numerical computation. *Computing in science & engineering*, 13(2):22–30, 2011. 5

[26] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017. 3

[27] Bram Wallace, Meihua Dang, Rafael Rafailov, Linqi Zhou, Aaron Lou, Senthil Purushwalkam, Stefano Ermon, Caiming Xiong, Shafiq Joty, and Nikhil Naik. Diffusion model alignment using direct preference optimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8228–8238, 2024. 1

[28] 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. *Advances in Neural Information Processing Systems*, 36, 2024. 5, 8

[29] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023. 2

[30] Hui Yuan, Kaixuan Huang, Chengzhuo Ni, Minshuo Chen, and Mengdi Wang. Reward-directed conditional diffusion: Provable distribution estimation and reward improvement. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 1, 2, 4, 5

[31] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 1, 2

[32] Yuxuan Zhang, Yiren Song, Jiaming Liu, Rui Wang, Jinpeng Yu, Hao Tang, Huaxia Li, Xu Tang, Yao Hu, Han Pan, et al. Ssr-encoder: Encoding selective subject representation for subject-driven generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8069–8078, 2024. 2