Controlling Text-to-Image Diffusion by Orthogonal Finetuning

Zeju Qiu[1,*] **Weiyang Liu**[1,2,*,†] **Haiwen Feng**[1] **Yuxuan Xue**[3] **Yao Feng**[1] **Zhen Liu**[1,4]
Dan Zhang[3,5] **Adrian Weller**[2,6] **Bernhard Schölkopf**[1]
[1]MPI for Intelligent Systems - Tübingen [2]University of Cambridge [3]University of Tübingen
[4]Mila, Université de Montréal [5]Bosch Center for Artificial Intelligence [6]The Alan Turing Institute
[*]Equal contribution [†]Project lead oft.wyliu.com

Abstract

Large text-to-image diffusion models have impressive capabilities in generating photorealistic images from text prompts. How to effectively guide or control these powerful models to perform different downstream tasks becomes an important open problem. To tackle this challenge, we introduce a principled finetuning method – Orthogonal Finetuning (OFT), for adapting text-to-image diffusion models to downstream tasks. Unlike existing methods, OFT can provably preserve hyperspherical energy which characterizes the pairwise neuron relationship on the unit hypersphere. We find that this property is crucial for preserving the semantic generation ability of text-to-image diffusion models. To improve finetuning stability, we further propose Constrained Orthogonal Finetuning (COFT) which imposes an additional radius constraint to the hypersphere. Specifically, we consider two important finetuning text-to-image tasks: subject-driven generation where the goal is to generate subject-specific images given a few images of a subject and a text prompt, and controllable generation where the goal is to enable the model to take in additional control signals. We empirically show that our OFT framework outperforms existing methods in generation quality and convergence speed.

1 Introduction

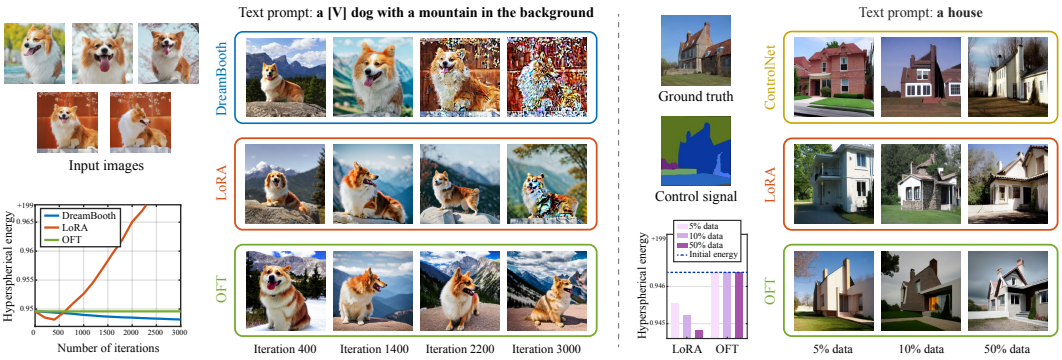
Recent text-to-image diffusion models [45, 50, 53] achieve impressive performance in text-guided control for high-fidelity image generation. Despite strong results, text guidance can still be ambiguous and insufficient to provide fine-grained and accurate control to the generated images. To address this shortcoming, we target two types of text-to-image generation tasks in this paper:

- **Subject-driven generation** [51]: Given just a few images of a subject, the task is to generate images of the same subject in a different context using the guidance of a text prompt.
- **Controllable generation** [38, 68]: Given an additional control signal (*e.g.*, canny edges, segmentation maps), the task is to generate images following such a control signal and a text prompt.

Both tasks essentially boil down to how to effectively finetune text-to-image diffusion models without losing the pretraining generative performance. We summarize the desiderata for an effective finetuning method as: (1) *training efficiency*: having fewer trainable parameters and number of training epochs, and (2) *generalizability preservation*: preserving the high-fidelity and diverse generative performance. To this end, finetuning is typically done either by updating the neuron weights by a small learning rate (*e.g.*, [51]) or by adding a small component with re-parameterized neuron weights (*e.g.*, [22, 68]). Despite simplicity, neither finetuning strategy is able to guarantee the preservation of pretraining generative performance. There is also a lack of principled understanding towards designing a good finetuning strategy and finding suitable hyperparameters such as the number of training epochs. A key difficulty is the lack of a measure for quantifying the preservation of pretrained generative ability. Existing finetuning methods implicitly assume that a smaller Euclidean distance between the

This work was finished when ZQ was a research intern hosted by WL at MPI for Intelligent Systems.

37th Conference on Neural Information Processing Systems (NeurIPS 2023).



(a) Subject-driven generation (b) Controllable generation

Figure 1: (a) Subject-driven generation: OFT preserves the hyperspherical energy and yields more stable finetuning performance across different number of iterations, while both DreamBooth [51] and LoRA [22] do not. OFT can preserve hyperspherical energy and perform stable finetuning, while both LoRA and DreamBooth are unable. (b) Controllable generation: OFT is more sample-efficient in training and converges well with only 5% of the original dataset, while both ControlNet [68] and LoRA [22] cannot converge until 50% of the data is present. The hyperspherical energy comparison between LoRA and OFT is fair, since they finetune the same layers. ControlNet uses a different layer finetuning strategy, so its hyperspherical energy is not comparable. The detailed settings are given in the experiment section and Appendix A.

finetuned model and the pretrained model indicates better preservation of the pretrained ability. Due to the same reason, finetuning methods typically work with a very small learning rate. While this assumption may occasionally hold, we argue that the Euclidean difference to the pretrained model alone is unable to fully capture the degree of semantic preservation, and therefore a more structural measure to characterize the difference between the finetuned model and the pretrained model can greatly benefit the preservation of pretraining performance as well as finetuning stability.

Inspired by the empirical observation that hyperspherical similarity encodes semantic information well [7, 35, 36], we use hyperspherical energy [32] to characterize the pairwise relational structure among neurons. Hyperspherical energy is defined as the sum of hyperspherical similarity (*e.g.*, cosine similarity) between all pairwise neurons in the same layer, capturing the level of neuron uniformity on the unit hypersphere [34]. We hypothesize that a good finetuned model should have a minimal difference in hyperspherical energy compared to the pretrained model. A naive way is to add a regularizer such that the hyperspherical energy remains the same during the finetuning stage, but there is no guarantee that the hyperspherical energy difference can be well minimized. Therefore, we take advantage of an invariance property of hyperspherical energy – the pairwise hyperspherical similarity is provably preserved if we apply the same orthogonal transformation for all neurons. Motivated by such an invariance, we propose Orthogonal Finetuning (OFT) which adapts large text-to-image diffusion models to a downstream task without changing its hyperspherical energy. The central idea is to learn a layer-shared orthogonal transformation for neurons such that their pairwise angles are preserved. OFT can also be viewed as adjusting the canonical coordinate system for the neurons in the same layer. By jointly taking into consideration that smaller Euclidean distance between the finetuned model and the pretrained model implies better preservation of pretraining performance, we further propose an OFT variant – Constrained Orthogonal Finetuning (COFT) which constrains the finetuned model within the hypersphere of a fixed radius centered on the pretrained neurons.

The intuition for why orthogonal transformation works for finetuning neurons partially comes from 2D Fourier transform, with which an image can be decomposed as magnitude and phase spectrum. The phase spectrum, which is angular information between input and basis, preserves the major part of semantics. For example, the phase spectrum of an image, along with a random magnitude spectrum, can still reconstruct the original image without losing its semantics. This phenomenon suggests that changing the neuron directions is the key to semantically modifying the generated image (which is the goal of both subject-driven and controllable generation). However, changing neuron directions with a large degree of freedom will inevitably destroy the pretraining generative performance. To constrain the degree of freedom, we propose to preserve the angle between any pair of neurons, largely based on the hypothesis that the angles between neurons are crucial for representing the knowledge of neural networks. With this intuition, it is natural to learn layer-shared orthogonal transformation for neurons in each layer such that the hyperspherical energy stays unchanged.

We also draw inspiration from orthogonal over-parameterized training [33] which trains classification neural networks from scratch by orthogonally transforming a randomly initialized neural network. This is because a randomly initialized neural network yields a provably small hyperspherical energy in

expectation and the goal of [33] is to keep hyperspherical energy small during training (small energy leads to better generalization in classification [30, 32]). [33] shows that orthogonal transformation is sufficiently flexible to train generalizable neural networks for classification problems. In contrast, we focus on finetuning text-to-image diffusion models for better controllability and stronger downstream generative performance. We emphasize the difference between OFT and [33] in two aspects. First, while [33] is designed to minimize the hyperspherical energy, OFT aims to preserve the same hyperspherical energy as the pretrained model so that the intrinsic pretrained structure will not be destroyed by finetuning. In the case of finetuning diffusion models, minimizing hyperspherical energy could destroy the original semantic structures. Second, OFT seeks to minimize the deviation from the pretrained model, which leads to the constrained variant. In contrast, [33] imposes no such constraints. The key to finetuning is to find a good trade-off between flexibility and stability, and we argue that our OFT framework effectively achieves this goal. Our contributions are listed below:

- We propose a novel finetuning method – Orthogonal Finetuning for guiding text-to-image diffusion models towards better controllability. To further improve stability, we propose a constrained variant which limits the angular deviation from the pretrained model.
- Compared to existing finetuning methods, OFT performs model finetuning while provably preserving the hyperspherical energy, which we empirically find to be an important measure of the generative semantic preservation of the pretrained model.
- We apply OFT to two tasks: subject-driven generation and controllable generation. We conduct a comprehensive empirical study and demonstrate significant improvement over prior work in terms of generation quality, convergence speed and finetuning stability. Moreover, OFT achieves better sample efficiency, as it converges well with a much smaller number of training images and epochs.
- For controllable generation, we introduce a new control consistency metric to evaluate the controllability. This core idea is to estimate the control signal from the generated image and then compare it with the origin control signal. The metric further validates the strong controllability of OFT.

2 Related Work

Text-to-image diffusion models. Tremendous progress [16, 39, 45, 50, 53] has been made in text-to-image generation, largely thanks to the rapid development in diffusion-based generative models [12, 20, 55, 56] and vision-language representation learning [1, 28, 29, 37, 44, 54, 57, 61]. GLIDE [39] and Imagen [53] train diffusion models in the pixel space. GLIDE trains the text encoder jointly with a diffusion prior using paired text-image data, while Imagen uses a frozen pretrained text encoder. Stable Diffusion [50] and DALL-E2 [45] train diffusion models in the latent space. Stable Diffusion uses VQ-GAN [14] to learn a visual codebook as its latent space, while DALL-E2 adopts CLIP [44] to construct a joint latent embedding space for representing images and text. Other than diffusion models, generative adversarial networks [27, 48, 65, 67] and autoregressive models [13, 46, 62, 66] have also been studied in text-to-image generation. OFT is inherently a model-agnostic finetuning approach and can be applied to any text-to-image diffusion model.

Subject-driven generation. To prevent subject modification, [2, 39] consider a given mask from users as an additional condition. Inversion methods [8, 12, 15, 45] can be applied to modify the context without changing the subject. [18] can perform local and global editing without input masks. The methods above are unable to well preserve identity-related details of the subject. In Pivotal Tuning [49], a generator is finetuned around an initial inverted latent code with an additional regularization to preserve the identity. Similarly, [41] learns a personalized generative face prior from a collection of a person's face images. [6] can generate difference variations of an instance, but it may lose the instance-specific details. With a customized token and a few subject images, DreamBooth [51] finetunes the text-to-image diffusion model using a reconstruction loss and a class-specific prior preservation loss. OFT adopts the DreamBooth framework, but instead of performing naive finetuning with a small learning rate, OFT finetunes the model with orthogonal transformations.

Controllable generation. The task of image-to-image translation can be viewed as a form of controllable generation, and previous methods mostly adopt conditional generative adversarial networks [9, 23, 42, 60, 71]. Diffusion models are also used for image-to-image translation [52, 58, 59]. More recently, ControlNet [68] proposes to control a pretrained diffusion model by finetuning and adapting it to additional control signals and achieves impressive controllable generation performance. Another concurrent and similar work, T2I-Adapter [38], also finetunes a pretrained diffusion model in order to gain stronger controllability for the generated images. Following the same task setting

in [38, 68], we apply OFT to finetune pretrained diffusion models, yielding consistently better controllability with fewer training data and less finetuning parameters. More significantly, OFT does not introduce any additional computational overhead during test-time inference.

Model finetuning. Finetuning large pretrained models on downstream tasks has been increasingly popular nowadays [3, 11, 17]. As a form of finetuning, adaptation methods (*e.g.*, [21, 22, 43]) are heavily studied in natural language processing. LoRA [22] is the most relevant work to OFT, and it assumes a low-rank structure for the additive weight update during finetuning. In contrast, OFT uses layer-shared orthogonal transformation to update neuron weights in a multiplicative manner, and it provably preserves the pair-wise angles among neurons in the same layer, yielding better stability.

3 Orthogonal Finetuning

3.1 Why Does Orthogonal Transformation Make Sense?

We start by discussing why orthogonal transformation is desirable in finetuning text-to-image diffusion models. We decompose this question into two smaller ones: (1) why we want to finetune the angle of neurons (*i.e.*, direction), and (2) why we adopt orthogonal transformation to finetune angles.

For the first question, we draw inspiration from the empirical observation in [7, 35] that angular feature difference well characterizes the semantic gap. SphereNet [36] shows that training a neural network with all neurons normalized onto a unit hypersphere yields comparable capacity and even better generalizability, implying that the direction of neurons can fully capture the most important information from data. To better demonstrate the importance of neuron angles, we conduct a toy experiment in Figure 2 where we train a standard convolutional autoencoder on some flower images. In the training stage, we use the standard inner product to produce the feature map (z denotes the element output of the convolution kernel $\boldsymbol{w}$ and $\boldsymbol{x}$ is the input in the sliding window). In the testing stage, we compare three ways to generate the feature map: (a) the inner product used in training, (b) the magnitude information, and (c) the angular information. The results in Figure 2 show that the angular information of neurons can almost perfectly recover the input images, while the magnitude of neurons contains no useful information. We emphasize that we do not apply the cosine activation (c) during training, and the training is done only based on inner product. The results imply that the angles (directions) of neurons play the major role in storing the semantic information of the input images. In order to modify the semantic information of images, finetuning the neuron directions will likely be more effective.

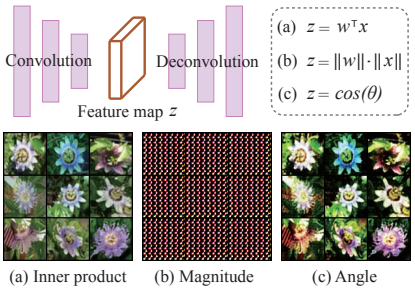


Figure 2: A toy experiment to demonstrate the importance of angular information. The autoencoder is trained in a standard way using inner product activation, and (a) shows the standard reconstruction. In testing, the angular information of neurons alone can well recover the input image, even if the autoencoder is not trained with angles.

For the second question, the simplest way to finetune direction of neurons is to simultaneously rotate / reflect all the neurons (in the same layer), which naturally brings in orthogonal transformation. It may be more flexible to use some other angular transformation that rotates different neurons with different angles, but we find that orthogonal transformation is a sweet spot between flexibility and regularity. Moreover, [33] shows that orthogonal transformation is sufficiently powerful for learning neural networks. To support our argument, we perform an experiment to demonstrate the effective regularization induced by the orthogonality constraint. We perform the controllable generation experiment using the setting of ControlNet [68], and the results are given in Figure 3. We can observe that our standard OFT performs quite stably and achieves accurate control after the training is finished (epoch 20). In comparison, OFT without the orthogonality constraint fails to generate any realistic image and achieve no control effect. The experiment validates the importance of the orthogonality constraint in OFT.

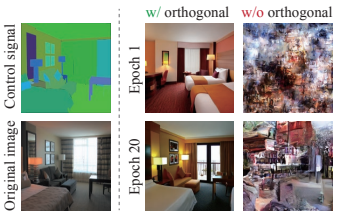


Figure 3: Controllable generation with or without orthogonality. Middle column is from the original OFT, and the right column is given by OFT without the orthogonality constraint.

3.2 General Framework

The conventional finetuning strategy typically uses gradient descent with a small learning rate to update a model (or certain layers of a model). The small learning rate implicitly encourages a small

deviation from the pretrained model, and the standard finetuning essentially aims to train the model while implicitly minimizing $\|\boldsymbol{M}-\boldsymbol{M}^0\|$ where $\boldsymbol{M}$ is the finetuned model weights and $\boldsymbol{M}^0$ is the pretrained model weights. This implicit constraint makes intuitive sense, but it can still be too flexible for finetuning a large model. To address this, LoRA introduces an additional low-rank constraint for the weight update, *i.e.*, $\text{rank}(\boldsymbol{M}-\boldsymbol{M}^0)=r'$ where r' is set to be some small number. Different from LoRA, OFT introduces a constraint for the pair-wise neuron similarity: $\|\text{HE}(\boldsymbol{M})-\text{HE}(\boldsymbol{M}^0)\|=0$ where $\text{HE}(\cdot)$ denotes hyperspherical energy of a weight matrix. As an illustrative example, we consider a fully connected layer $\boldsymbol{W}=\{\boldsymbol{w}_1,\cdots,\boldsymbol{w}_n\}\in\mathbb{R}^{d\times n}$ where $\boldsymbol{w}_i\in\mathbb{R}^d$ is the i-th neuron ($\boldsymbol{W}^0$ is the pretrained weights). The output vector $\boldsymbol{z}\in\mathbb{R}^n$ of this fully connected layer is computed by $\boldsymbol{z}=\boldsymbol{W}^\top\boldsymbol{x}$ where $\boldsymbol{x}\in\mathbb{R}^d$ is the input vector. OFT can be interpreted as minimizing the hyperspherical energy difference between the finetuned model and the pretrained model:

$$\min_{\boldsymbol{W}}\left\|\text{HE}(\boldsymbol{W})-\text{HE}(\boldsymbol{W}^0)\right\| \quad\Leftrightarrow\quad \min_{\boldsymbol{W}}\left\|\sum_{i\neq j}\|\hat{\boldsymbol{w}}_i-\hat{\boldsymbol{w}}_j\|^{-1}-\sum_{i\neq j}\|\hat{\boldsymbol{w}}_i^0-\hat{\boldsymbol{w}}_j^0\|^{-1}\right\| \tag{1}$$

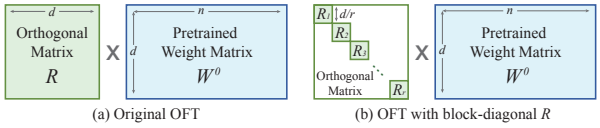
where $\hat{\boldsymbol{w}}_i:=\boldsymbol{w}_i/\|\boldsymbol{w}_i\|$ denotes the i-th normalized neuron, and the hyperspherical energy of a fully connected layer $\boldsymbol{W}$ is defined as $\text{HE}(\boldsymbol{W}):=\sum_{i\neq j}\|\hat{\boldsymbol{w}}_i-\hat{\boldsymbol{w}}_j\|^{-1}$. One can easily observe that the attainable minimum is zero for Eq. (1). The minimum can be achieved as long as $\boldsymbol{W}$ and $\boldsymbol{W}^0$ differ only up to a rotation or reflection, *i.e.*, $\boldsymbol{W}=\boldsymbol{R}\boldsymbol{W}^0$ in which $\boldsymbol{R}\in\mathbb{R}^{d\times d}$ is an orthogonal matrix (The determinant 1 or -1 means rotation or reflection, respectively). This is exactly the idea of OFT, that we only need to finetune the neural network by learning layer-shared orthogonal matrices to transform neurons in each layer. Formally, OFT seeks to optimize the orthogonal matrix $\boldsymbol{R}\in\mathbb{R}^{d\times d}$ for a pretrained fully connected layer $\boldsymbol{W}^0\in\mathbb{R}^{d\times n}$, changing the forward pass from $\boldsymbol{z}=(\boldsymbol{W}^0)^\top\boldsymbol{x}$ to

$$\boldsymbol{z}=\boldsymbol{W}^\top\boldsymbol{x}=(\boldsymbol{R}\cdot\boldsymbol{W}^0)^\top\boldsymbol{x},\quad \text{s.t. } \boldsymbol{R}^\top\boldsymbol{R}=\boldsymbol{R}\boldsymbol{R}^\top=\boldsymbol{I} \tag{2}$$

where $\boldsymbol{W}$ denotes the OFT-finetuned weight matrix and $\boldsymbol{I}$ is an identity matrix. OFT is illustrated in Figure 4. Similar to the zero initialization in LoRA, we need to ensure OFT to finetune the pretrained model exactly from $\boldsymbol{W}^0$. To achieve this, we initialize the orthogonal matrix $\boldsymbol{R}$ to be an identity matrix so that the finetuned model starts with the pretrained weights. To guarantee the orthogonality of the matrix $\boldsymbol{R}$, we can use differential orthogonalization strategies discussed in [26, 33]. We will discuss how to guarantee the orthogonality in a computationally efficient way.

3.3 Efficient Orthogonal Parameterization

Standard orthogonalization such as Gram-Schmidt method, despite differentiable, is often too expensive to compute in practice [33]. For better efficiency, we adopt Cayley parameterization to generate the orthogonal matrix. Specifically, we construct the orthogonal matrix with $\boldsymbol{R}=(\boldsymbol{I}+\boldsymbol{Q})(I-\boldsymbol{Q})^{-1}$ where $\boldsymbol{Q}$ is a skew-symmetric matrix satisfying $\boldsymbol{Q}=-\boldsymbol{Q}^\top$. Such an efficiency comes at a small price – the Cayley parameterization can only produce orthogonal matrices with determinant 1 which belongs to the special orthogonal group. Fortunately, we find that such a limitation does not affect the performance in practice. Even if we use Cayley transform to parameterize the orthogonal matrix, $\boldsymbol{R}$ can still be very parameter-inefficient with a large d. To address this, we propose to represent $\boldsymbol{R}$ with a block-diagonal matrix with r blocks, leading to the following form:



(a) Original OFT (b) OFT with block-diagonal R

Figure 4: (a) Original OFT without a diagonal structure. (b) OFT with r diagonal blocks of the same size. When $r=1$, the case of (b) recovers the case of (a).

$$\boldsymbol{R}=\text{diag}(\boldsymbol{R}_1,\boldsymbol{R}_2,\cdots,\boldsymbol{R}_r)=\begin{bmatrix}\boldsymbol{R}_1\in\text{O}(\frac{d}{r}) & & \\ & \ddots & \\ & & \boldsymbol{R}_r\in\text{O}(\frac{d}{r})\end{bmatrix}\in\text{O}(d) \tag{3}$$

where $\text{O}(d)$ denotes the orthogonal group in dimension d, and $\boldsymbol{R}\in\mathbb{R}^{d\times d}$ and $\boldsymbol{R}_i\in\mathbb{R}^{d/r\times d/r},\forall i$ are orthogonal matrices. When $r=1$, then the block-diagonal orthogonal matrix becomes a standard unconstrained one. For an orthogonal matrix with size $d\times d$, the number of parameters is $d(d-1)/2$, resulting in a complexity of $\mathcal{O}(d^2)$. For an r-block diagonal orthogonal matrix, the number of parameter is $d(d/r-1)/2$, resulting in a complexity of $\mathcal{O}(d^2/r)$. We can optionally share the block matrix to further reduce the number of parameters, *i.e.*, $\boldsymbol{R}_i=\boldsymbol{R}_j,\forall i\neq j$. This reduces the parameter complexity to $\mathcal{O}(d^2/r^2)$. Despite all these strategies to improve parameter efficiency, we note that the resulting matrix $\boldsymbol{R}$ remains orthogonal, so there is no sacrifice in preserving hyperspherical energy.

We discuss how OFT compares to LoRA in terms of parameter efficiency. For LoRA with a low-rank parameter r', we have its number of trainable parameters as $r'(d+n)$. If we consider both r and r' to be dependent on the neuron dimension d (*e.g.*, $r=r'=\alpha d$ where $0<\alpha\leq 1$ is some constant), then the parameter complexity of LoRA becomes $\mathcal{O}(d^2+dn)$ and the parameter complexity of OFT becomes $\mathcal{O}(d)$. We illustrate the difference in complexity between OFT and LoRA with a concrete example. Suppose we have a weight matrix with size 128×128, LoRA has $2,048$ trainable parameters with $r'=8$, while OFT has 960 trainable parameters with $r=8$ (no block sharing is applied).

3.4 Constrained Orthogonal Finetuning

We can further limit the flexibility of original OFT by constraining the finetuned model to be within a small neighborhood of the pretrained model. Specifically, COFT uses the following forward pass:

$$\boldsymbol{z}=\boldsymbol{W}^\top\boldsymbol{x}=(\boldsymbol{R}\cdot\boldsymbol{W}^0)^\top\boldsymbol{x},\quad \text{s.t. } \boldsymbol{R}^\top\boldsymbol{R}=\boldsymbol{R}\boldsymbol{R}^\top=\boldsymbol{I},\ \|\boldsymbol{R}-\boldsymbol{I}\|\leq\epsilon \tag{4}$$

which has an orthogonality constraint and an ϵ-deviation constraint to an identity matrix. The orthogonality constraint can be achieved with the Cayley parameterization introduced in Section 3.3. However, it is nontrivial to incorporate the ϵ-deviation constraint to the Cayley-parameterized orthogonal matrix. To gain more insights on the Cayley transform, we apply the Neumann series to approximate $\boldsymbol{R}=(\boldsymbol{I}+\boldsymbol{Q})(I-\boldsymbol{Q})^{-1}$ as $\boldsymbol{R}\approx\boldsymbol{I}+2\boldsymbol{Q}+\mathcal{O}(\boldsymbol{Q}^2)$ (under the assumption that the Neumann series converges in the operator norm). Therefore, we can move the constraint $\|\boldsymbol{R}-\boldsymbol{I}\|\leq\epsilon$ inside the Cayley transform, and the equivalent constraint is $\|\boldsymbol{Q}-\boldsymbol{0}\|\leq\epsilon'$ where $\boldsymbol{0}$ denotes an all-zero matrix and ϵ' is another error hyperparameter (different than ϵ). The new constraint on the matrix $\boldsymbol{Q}$ can be easily enforced by projected gradient descent. To achieve identity initialization for the orthogonal matrix $\boldsymbol{R}$, we initialize $\boldsymbol{Q}$ as an all-zero matrix. COFT can be viewed as a combination of two explicit constraints: minimal hyperspherical energy difference and constrained deviation from the pretrained model. The second constraint is usually implicitly used by existing finetuning methods, but COFT makes it an explicit one. Despite the excellent performance of OFT, we observe that COFT yields even better finetuning stability than OFT due to this explicit deviation constraint. Figure 5 provides an example on how ϵ affects the performance of COFT. We can observe that ϵ controls the flexibility of finetuning. With larger ϵ, the COFT-finetuned model resembles the OFT-finetuned model. With smaller ϵ, the COFT-finetuned model behaves increasingly similar to the pretrained text-to-image diffusion model.

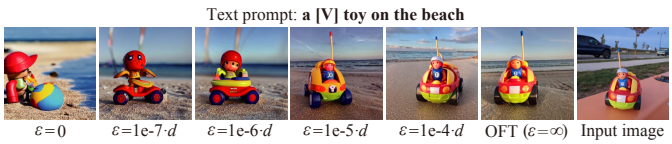


Figure 5: How ϵ affects the flexibility of COFT in subject-driven generation.

3.5 Re-scaled Orthogonal Finetuning

We propose a simple extension to the original OFT by additionally learning a magnitude scaling coefficient for each neuron. This is motivated by the fact that re-scaling neurons does not change the hyperspherical energy (the magnitude will be normalized out). Specifically, we use the forward pass: $\boldsymbol{z}=(\boldsymbol{R}\boldsymbol{W}^0\boldsymbol{D})^\top\boldsymbol{x}$[1] where $\boldsymbol{D}=\text{diag}(s_1,\cdots,s_n)\in\mathbb{R}^{n\times n}$ is a learnable diagonal matrix with all the diagonal element $s_1,\cdots,s_n$ larger than zero. In contrast to OFT's original forward pass in Eq. (2) where only $\boldsymbol{R}$ is learnable, we have both the diagonal matrix $\boldsymbol{D}$ and the orthogonal matrix $\boldsymbol{R}$ learnable. The re-scaled OFT further improves the flexibility of OFT with a neglectable number of additional parameters. We stick to the original OFT in the experiment to show the effectiveness of orthogonal transformation alone, but we find that the re-scaled OFT is generally better (see Appendix C).

4 Intriguing Insights and Discussions

OFT is agnostic to different architectures. We can apply OFT to any type of neural network in principle. For Transformers, LoRA is typically applied to the attention weights [22]. To compare fairly to LoRA, we only apply OFT to finetune the attention weights in our experiments. Besides fully connected layers, OFT is also well suited for finetuning convolution layers, because the block-diagonal structure of $\boldsymbol{R}$ has interesting interpretations in convolution layers (unlike LoRA). When we use the same number of blocks as the number of input channels, each block only transforms a unique neuron channel, similar to learning depth-wise convolution kernels [10]. When all the blocks

[1] **Errata**: In the NeurIPS camera ready version, the forward pass of re-scaled OFT is mistakenly written as $\boldsymbol{z}=(\boldsymbol{D}\boldsymbol{R}\boldsymbol{W}^0)^\top\boldsymbol{x}$. The original implementation is correct, so the results in Appendix C are unaffected.

in $\boldsymbol{R}$ are shared, OFT transforms the neurons with an orthogonal matrix shared across channels. We conduct a preliminary study on finetuning convolution layers with OFT in Appendix D

Connection to LoRA. By adding a low-rank matrix, LoRA prevents the information in the pretrained weight matrix from shifting dramatically. In contrast, OFT controls the transform that applies to the pretrained weight matrix to be orthogonal (full-rank), which prevents the transform to destroy the pretraining information. We can rewrite OFT's forward pass as $\boldsymbol{z}=(\boldsymbol{R}\boldsymbol{W}^0)^\top\boldsymbol{x}=(\boldsymbol{W}^0+(\boldsymbol{R}-\boldsymbol{I})\boldsymbol{W}^0)^\top\boldsymbol{x}$ where $(\boldsymbol{R}-\boldsymbol{I})\boldsymbol{W}^0$ is analogous to LoRA's low-rank weight update. Since $\boldsymbol{W}^0$ is typically full-rank, OFT also performs low-rank weight update when $\boldsymbol{R}-\boldsymbol{I}$ is low-rank. Similar to LoRA that has a rank parameter r', OFT has a diagonal block parameter r to reduce the number of trainable parameters. More interestingly, LoRA and OFT represent two distinct ways to be parameter-efficient. LoRA exploits the low-rank structure to reduce the number of trainable parameters, while OFT takes a different route by exploiting the sparsity structure (*i.e.*, block-diagonal orthogonality).

Why OFT converges faster. On one hand, we can see from Figure 2 that the most effective update to modify the semantics is to change neuron directions, which is exactly what OFT is designed for. On the other hand, OFT can be viewed as finetuning neurons on a smooth hypersphere manifold, which yields better optimization landscape. This is also empirically verified in [33].

Why not minimize hyperspherical energy. A key difference to [33] is that we do not aim to minimize hyperspherical energy. In classification problems, neurons without redundancy are desired. The minimum hyperspherical energy means all neurons are uniformly spaced around the hypersphere. This is not a meaningful objective for finetuning, as it may destroy the pretraining information.

Trade-off between flexibility and regularity in finetuning. We discover an underlying trade-off between flexibility and regularity. Standard finetuning is the most flexible method, but it yields poor stability and easily causes model collapse. Being surprisingly simple, OFT finds a good balance between flexibility and regularity by preserving the pairwise neuron angles. The block-diagonal parameterization can also be viewed as a stronger regularization of the orthogonal matrix.

No additional inference overhead. Unlike ControlNet, our OFT framework introduces no additional inference overhead to the finetuned model. In the inference stage, we can simply multiply the learned orthogonal matrix $\boldsymbol{R}$ into the pretrained weight matrix $\boldsymbol{W}^0$ and obtain an equivalent weight matrix $\boldsymbol{W}=\boldsymbol{R}\boldsymbol{W}^0$. Thus the inference speed is the same as the pretrained model.

5 Experiments and Results

General settings. In the experiment, we use Stable Diffusion v1.5 [50] as the pretrained text-to-image model. For fairness, we randomly pick generated images from each method. For subject-driven generation, we generally follow DreamBooth [51]. For controllable generation, we generally follow ControlNet [68] and T2I-Adapter [38]. To ensure a fair comparison to LoRA, we only apply OFT or COFT to the same layer where LoRA is used. More results and details are given in Appendix A.

5.1 Subject-driven Generation

Settings. We use DreamBooth [51] and LoRA [22] as the baselines. All the methods adopt the same loss function as in DreamBooth. For DreamBooth and LoRA, we generally follow the original paper and use the best hyperparameter setup. More results are provided in Appendix A,E,F,J.

Figure 6: Generated images across different iterations.

Finetuning stability and convergence. We first evaluate the finetuning stability and the convergence speed for DreamBooth, LoRA, OFT and COFT. Results are given in Figure 1 and Figure 6. We can observe that both COFT and OFT are able to finetune the diffusion model quite stably. After 400 iterations, both DreamBooth and OFT variants achieve good control, while LoRA fails to preserve the subject identity. After 2000 iterations, DreamBooth starts to generate collapsed images, and LoRA fails to generate yellow shirt (and instead generates yellow fur). In contrast, both OFT and COFT are still able to achieve stable and consistent control over the generated image. These results validate the fast yet stable convergence of our OFT framework in subject-driven genera-

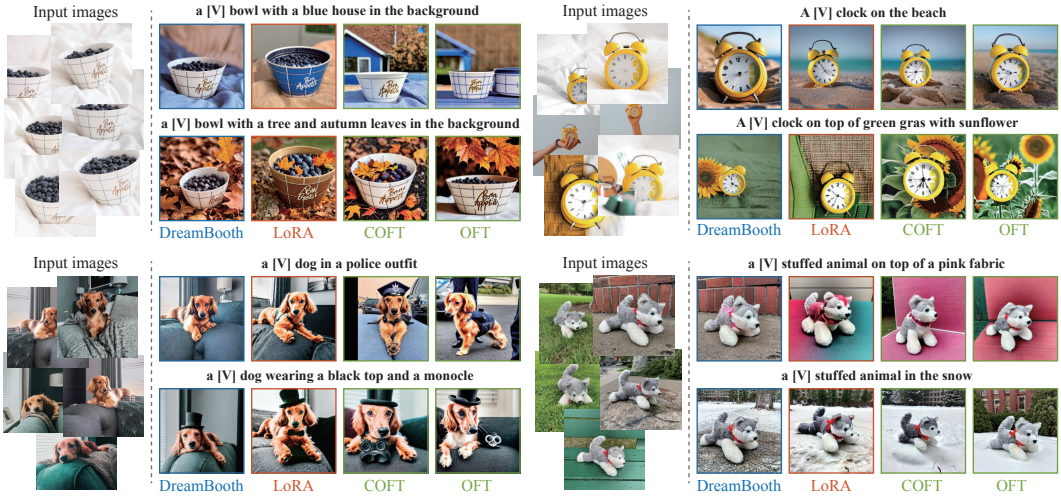


Figure 7: Qualitative comparison of subject-driven generation among DreamBooth, LoRA, COFT and OFT. Results are generated with the same finetuned model from each method. All examples are randomly picked. The figure is best viewed digitally, in color and significantly zoomed in.

tion. We note that the insensitivity to the number of finetuning iteration is quite important, since it can effectively alleviate the trouble of tuning the iteration number for different subjects. For both OFT and COFT, we can directly set a relatively large iteration number without carefully tuning it. For COFT with a proper ϵ, both the learning rate and the iteration number become effortless to set.

Quantitative comparison. Following [51], we conduct a quantitative comparison to evaluate subject fidelity (DINO [5], CLIP-I [44]), text prompt fidelity (CLIP-T [44]) and sample diversity (LPIPS [69]). CLIP-I computes the average pairwise cosine similarity of CLIP embeddings between generated and real images. DINO is similar to CLIP-I, except that we use ViT S/16 DINO embeddings. CLIP-T is the average cosine similarity of CLIP embeddings between text prompt and generated images. We also evaluate average LPIPS cosine similarity between generated images of the same subject with the same text prompt. Table 1 show that both COFT and OFT outperforms DreamBooth and LoRA in the DINO and CLIP-I metrics by a considerable margin, while achieving slightly better or comparable performance in prompt fidelity and diversity metric. For each method, we repeatedly finetune the same pretrained model with 30 different random seeds to minimize randomness. The results show that our OFT framework not only achieves better convergence and stability, but also yields consistently better final performance.

Method	DINO ↑	CLIP-I ↑	CLIP-T ↑	LPIPS ↑
Real Images	0.703	0.864	-	0.695
DreamBooth	0.614	0.778	**0.239**	0.737
LoRA	0.613	0.765	0.237	0.744
COFT	0.630	0.783	0.235	0.744
OFT	**0.632**	**0.785**	0.237	**0.746**

Table 1: Quantitative comparison of subject fidelity (DINO, CLIP-I), prompt fidelity (CLIP-T) and diversity metric (LPIPS). The evaluation images and prompts are the same as [51] (25 subjects with 30 text prompts each subject).

Qualitative comparison. To have a more intuitive understanding of OFT's benefits, we show some randomly picked examples for subject-driven generation in Figure 7. For a fair comparison, all the examples are generated from the same finetuned model using each method, so no text prompt will be separately optimized for its final results. For each method, we select the model that achieves the best validation CLIP metrics. From the results in Figure 7, we can observe that both OFT and COFT deliver excellent semantic subject preservation, while LoRA often fails to preserve the subject identity (*e.g.*, LoRA completely loses the subject identity in the bowl example). In the meantime, both OFT and COFT have much more accurate control using text prompts, while DreamBooth, despite its preservation of subject identity, often fails to generate the image following the text prompt (*e.g.*, the first row of the bowl example). The qualitative comparison demonstrates that our OFT framework achieves better controllability and subject preservation at the same time. Moreover, the number of iterations is not sensitive in OFT, so OFT performs well even with a large number of iterations, while neither DreamBooth nor LoRA can. More qualitative examples are given in Appendix F. Moreover, we conduct a human evaluation in Appendix H which further validates the superiority of OFT.

5.2 Controllable Generation

Settings. We use ControlNet [68], T2I-Adapter [38] and LoRA [22] as the baselines. We consider three challenging controllable generation tasks in the main paper: Canny edge to image (C2I) on the

COCO dataset [31], segmentation map to image (S2I) on the ADE20K dataset [70] and landmark to face (L2F) on the CelebA-HQ dataset [25, 63]. All the methods are used to finetune Stable Diffusion (SD) v1.5 on these three datasets for 20 epochs. More results are given in Appendix F,G,J.

Convergence. We evaluate the convergence speed of ControlNet, T2I-Adapter, LoRA and COFT on the L2F task. We provide both quantitative and qualitative evaluation. Specifically for the evaluation metric, we compute the mean ℓ_2 distance between control face landmarks and predicted face landmarks. In Figure 8, we plot the face landmark error obtained by the model finetuned with different number of epochs. We can observe that both COFT and OFT achieve significantly faster convergence. It takes 20 epochs for LoRA to converge to the performance of our OFT framework at the 8-th epoch. We note that OFT and COFT use a similar number of trainable parameters to LoRA (much fewer than ControlNet), while being much more efficient to converge than existing methods. On the other hand, the fast convergence of OFT is also validated by the results in Figure 1. The right example in Figure 1 shows that OFT is much more data-efficient than ControlNet and LoRA, since OFT can converge well with only 5% of the ADE20K dataset. For qualitative results, we focus on comparing OFT, COFT and ControlNet, because ControlNet achieves the closest landmark error to ours. Results in Figure 9 show that both OFT and COFT converge stably and the generated face pose is gradually aligned with the control landmarks. In contrast to our stable and smooth convergence, the controllability in ControlNet suddenly emerges after the 8-th epoch, which perfectly matches the sudden convergence phenomenon observed in [68]. Such a convergence stability makes our OFT framework much easier to use in practice, since the training dynamics of OFT is far more smooth and predictable. Thus it will be easier to find good OFT's hyperparameters.

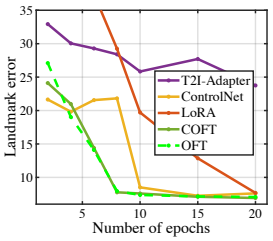


Figure 8: Face landmark error.

Quantitative comparison. We introduce a control consistency metric to evaluate the performance of controllable generation. The basic idea is to compute the control signal from the generated image and then compare it with the original input control signal. For the C2I task, we compute IoU and F1 score. For the S2I task, we compute mean IoU, mean and overall accuracy. For the L2F task, we compute the mean ℓ_2 distance between control landmarks and predicted landmarks. More details regarding the consistency metrics are given in Appendix A. For all the compared method, we use the best possible hyperparameter settings. Results in Table 2 show that both OFT and COFT yield much stronger and accurate control than the other methods. We observe that the adapter-based approaches (*e.g.*, T2I-Adapter and ControlNet) converge slowly and also yield worse final results. Compared to ControlNet, LoRA performs better in the S2I task and worse in the C2I and L2F tasks. In general, we find that the performance ceiling of LoRA is relatively low, even if we have carefully tuned its hyperparameters. As a comparison, the performance of our OFT framework has not yet saturated, since we empirically find that it still gets better as the number of trainable parameters gets large. We emphasize that our quantitative evaluation in controllable generation is one of our novel contributions, since it can accurately evaluate the control performance of the finetuned models (up to the accuracy of the off-the-shelf segmentation/detection model).

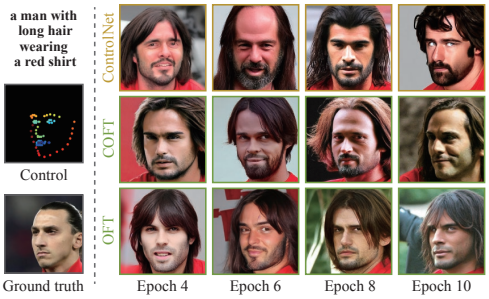


Figure 9: Qualitative examples with different number of epochs.

Qualitative comparison. We also qualitatively compare OFT and COFT to current state-of-the-art methods, including ControlNet, T2I-Adapter and LoRA. Randomly generated images in Figure 10 show that OFT and COFT not only yield high-fidelity and realistic image quality, but also achieve accurate control. In the S2I task, we can see that LoRA completely fails to generate images following the input segmentation map, while ControlNet, OFT and COFT can well control the generated images. In contrast to ControlNet, both OFT and COFT are able to generate high-fidelity images with more vivid details and more reasonable

Task	Metric	SD	ControlNet	T2I-Adapter	LoRA	COFT	OFT
C2I	IoU ↑	0.049	0.189	0.078	0.168	**0.195**	0.193
	F1 ↑	0.093	0.317	0.143	0.286	**0.325**	0.323
S2I	mIoU ↑	7.72	20.88	16.38	22.98	26.92	**27.06**
	mAcc ↑	14.40	30.91	26.31	35.52	40.08	**40.09**
	aAcc ↑	33.61	61.42	51.63	58.03	**62.96**	62.42
L2F	Error ↓	146.19	7.61	23.75	7.68	**6.92**	7.07

Table 2: Quantitative comparison of control signal consistency for three control tasks (Canny edge to image, segmentation to image and landmark to face).

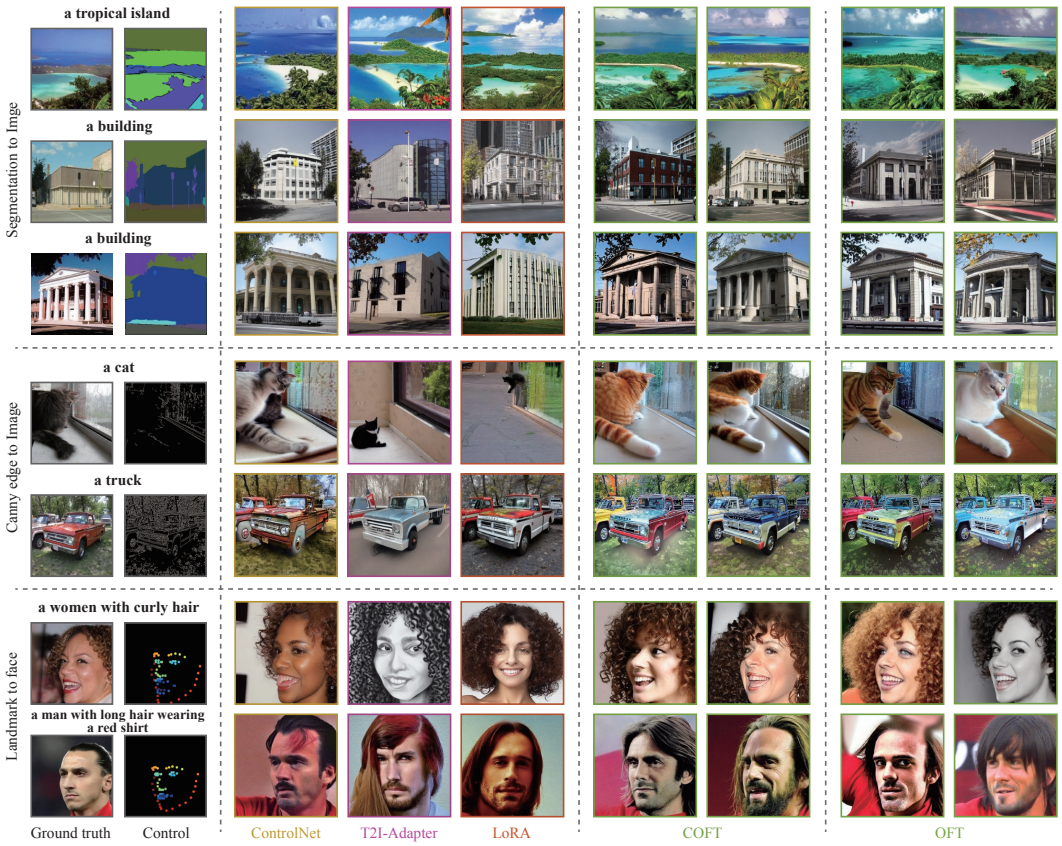


Figure 10: Qualitative comparison of controllable generation. The figure is best viewed digitally, in color and significantly zoomed in.

geometric structures with far less model parameters. In the C2I task, both OFT and COFT are able to hallucinate semantically similar images based on a rough Canny edges, while T2I-Adapter and LoRA perform much worse. In the L2F task, our method produces the most accurate pose control for the generated faces even under challenging face poses. In all three control tasks, we show that both OFT and COFT produce qualitatively better images than the state-of-the-art baselines, demonstrating the effectiveness of our OFT framework in controllable generation. To give a more comprehensive qualitative comparison, we provide more qualitative examples for all the three control tasks in Appendix F.2, and moreover, we demonstrate OFT can perform well on more control tasks (including dense pose to human body, sketch to image and depth to image) in Appendix G.

6 Concluding Remarks and Open Problems

Motivated by the observation that angular information among neurons crucially determines visual semantics, we propose a simple yet effective finetuning method – orthogonal finetuning for controlling text-to-image diffusion models. Specifically, we target two text-to-image applications: subject-driven generation and controllable generation. Compared to existing methods, OFT demonstrates stronger controllability and finetuning stability with fewer number of finetuning parameters. More importantly, OFT does not introduce additional inference overhead, leading to an efficient deployable model.

OFT also introduces a few interesting open problems. First, OFT guarantees the orthogonality via Cayley parametrization which involves a matrix inverse. It slightly limits the scalability of OFT. Although we address this limitation using block diagonal parametrization, how to speed up this matrix inverse in a differentiable way remains a challenge. Second, OFT has unique potential in compositionality, in the sense that the orthogonal matrices produced by multiple OFT finetuning tasks can be multiplied together and remains an orthogonal matrix. Whether this set of orthogonal matrices preserve the knowledge of all the downstream tasks remains an interesting direction to study. Finally, the parameter efficiency of OFT is largely dependent on the block diagonal structure which inevitably introduces additional biases and limits the flexibility. How to improve the parameter efficiency in a more effective and less biased way remains an important open problem.

Acknowledgement

The authors would like to sincerely thank Luigi Gresele, Yandong Wen, Yuliang Xiu and many other colleagues at Max Planck Institute for Intelligent Systems for many helpful suggestions.

This work was supported by the German Federal Ministry of Education and Research (BMBF): Tubingen AI Center, FKZ: 01IS18039B, and by the Machine Learning Cluster of Excellence, EXC number 2064/1 – Project number 390727645. WL was supported by the German Research Foundation (DFG): SFB 1233, Robust Vision: Inference Principles and Neural Mechanisms, TP XX, project number: 276693517. AW acknowledges support from a Turing AI Fellowship under grant EP/V025279/1, The Alan Turing Institute, and the Leverhulme Trust via CFI.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, 2022. 3

[2] Omri Avrahami, Ohad Fried, and Dani Lischinski. Blended latent diffusion. *arXiv preprint arXiv:2206.02779*, 2022. 3

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *NeurIPS*, 2020. 4

[4] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). In *ICCV*, 2017. 17

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 8

[6] Arantxa Casanova, Marlene Careil, Jakob Verbeek, Michal Drozdzal, and Adriana Romero Soriano. Instance-conditioned gan. In *NeurIPS*, 2021. 3

[7] Beidi Chen, Weiyang Liu, Zhiding Yu, Jan Kautz, Anshumali Shrivastava, Animesh Garg, and Animashree Anandkumar. Angular visual hardness. In *ICML*, 2020. 2, 4

[8] Jooyoung Choi, Sungwon Kim, Yonghyun Jeong, Youngjune Gwon, and Sungroh Yoon. Ilvr: Conditioning method for denoising diffusion probabilistic models. In *ICCV*, 2021. 3

[9] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *CVPR*, 2018. 3

[10] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, 2017. 6

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 4

[12] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *NeurIPS*, 2021. 3

[13] Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. Cogview: Mastering text-to-image generation via transformers. In *NeurIPS*, 2021. 3

[14] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, 2021. 3

[15] 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. *arXiv preprint arXiv:2208.01618*, 2022. 3

[16] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *CVPR*, 2022. 3

[17] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 4

[18] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. *arXiv preprint arXiv:2208.01626*, 2022. 3

[19] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NeurIPS*, 2017. 17

[20] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *NeurIPS*, 2020. 3

[21] Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for nlp. In *ICML*, 2019. 4

[22] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *ICLR*, 2022. 1, 2, 4, 6, 7, 8, 20

[23] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 3

[24] Yuming Jiang, Shuai Yang, Haonan Qiu, Wayne Wu, Chen Change Loy, and Ziwei Liu. Text2human: Text-driven controllable human image generation. *ACM Transactions on Graphics (TOG)*, 41(4):1–11, 2022. 16

[25] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018. 9, 16

[26] Mario Lezcano-Casado and David Martınez-Rubio. Cheap orthogonal constraints in neural networks: A simple parametrization of the orthogonal and unitary group. In *ICML*, 2019. 5

[27] Bowen Li, Xiaojuan Qi, Thomas Lukasiewicz, and Philip Torr. Controllable text-to-image generation. In *NeurIPS*, 2019. 3

[28] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 3

[29] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022. 3, 16

[30] Rongmei Lin, Weiyang Liu, Zhen Liu, Chen Feng, Zhiding Yu, James M Rehg, Li Xiong, and Le Song. Regularizing neural networks via minimizing hyperspherical energy. In *CVPR*, 2020. 3

[31] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 9, 16

[32] Weiyang Liu, Rongmei Lin, Zhen Liu, Lixin Liu, Zhiding Yu, Bo Dai, and Le Song. Learning towards minimum hyperspherical energy. In *NeurIPS*, 2018. 2, 3

[33] Weiyang Liu, Rongmei Lin, Zhen Liu, James M Rehg, Liam Paull, Li Xiong, Le Song, and Adrian Weller. Orthogonal over-parameterized training. In *CVPR*, 2021. 2, 3, 4, 5, 7

[34] Weiyang Liu, Rongmei Lin, Zhen Liu, Li Xiong, Bernhard Schölkopf, and Adrian Weller. Learning with hyperspherical uniformity. In *AISTATS*, 2021. 2

[35] Weiyang Liu, Zhen Liu, Zhiding Yu, Bo Dai, Rongmei Lin, Yisen Wang, James M Rehg, and Le Song. Decoupled networks. In *CVPR*, 2018. 2, 4

[36] Weiyang Liu, Yan-Ming Zhang, Xingguo Li, Zhiding Yu, Bo Dai, Tuo Zhao, and Le Song. Deep hyperspherical learning. In *NIPS*, 2017. 2, 4

[37] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, 2019. 3

[38] Chong Mou, Xintao Wang, Liangbin Xie, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. *arXiv preprint arXiv:2302.08453*, 2023. 1, 3, 4, 7, 8

[39] Alexander Quinn 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. In *ICML*, 2022. 3

[40] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics & Image Processing*, 2008. 16

[41] Yotam Nitzan, Kfir Aberman, Qiurui He, Orly Liba, Michal Yarom, Yossi Gandelsman, Inbar Mosseri, Yael Pritch, and Daniel Cohen-Or. Mystyle: A personalized generative prior. *ACM Transactions on Graphics (TOG)*, 41(6):1–10, 2022. 3

[42] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *CVPR*, 2019. 3

[43] Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. Adapterfusion: Non-destructive task composition for transfer learning. *arXiv preprint arXiv:2005.00247*, 2020. 4

[44] 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 *ICML*, 2021. 3, 8

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

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

[47] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *IEEE transactions on pattern analysis and machine intelligence*, 44(3):1623–1637, 2020. 16

[48] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *ICML*, 2016. 3

[49] Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. Pivotal tuning for latent-based editing of real images. *ACM Transactions on Graphics*, 42(1):1–13, 2022. 3

[50] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 1, 3, 7

[51] 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. In *CVPR*, 2023. 1, 2, 3, 7, 8, 16

[52] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *SIGGRAPH 2022*, pages 1–10, 2022. 3

[53] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. In *NeurIPS*, 2022. 1, 3

[54] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *arXiv preprint arXiv:2210.08402*, 2022. 3

[55] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 3

[56] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021. 3

[57] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. Vl-bert: Pre-training of generic visual-linguistic representations. In *ICLR*, 2020. 3

[58] Andrey Voynov, Kfir Aberman, and Daniel Cohen-Or. Sketch-guided text-to-image diffusion models. *arXiv preprint arXiv:2211.13752*, 2022. 3

[59] Tengfei Wang, Ting Zhang, Bo Zhang, Hao Ouyang, Dong Chen, Qifeng Chen, and Fang Wen. Pretraining is all you need for image-to-image translation. *arXiv preprint arXiv:2205.12952*, 2022. 3

[60] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *CVPR*, 2018. 3

[61] Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. Simvlm: Simple visual language model pretraining with weak supervision. *arXiv preprint arXiv:2108.10904*, 2021. 3

[62] Chenfei Wu, Jian Liang, Lei Ji, Fan Yang, Yuejian Fang, Daxin Jiang, and Nan Duan. Nüwa: Visual synthesis pre-training for neural visual world creation. In *ECCV*, 2022. 3

[63] Weihao Xia, Yujiu Yang, Jing-Hao Xue, and Baoyuan Wu. Tedigan: Text-guided diverse face image generation and manipulation. In *CVPR*, 2021. 9

[64] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *NeurIPS*, 2021. 17

[65] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *CVPR*, 2018. 3

[66] Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, et al. Scaling autoregressive models for content-rich text-to-image generation. *Transactions on Machine Learning Research*, 2022. 3

[67] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *ICCV*, 2017. 3

[68] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023. 1, 2, 3, 4, 7, 8, 9, 16

[69] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 8

[70] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017. 9, 16

[71] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A Efros, Oliver Wang, and Eli Shechtman. Toward multimodal image-to-image translation. In *NeurIPS*, 2017. 3

Appendix

Table of Contents

A Experimental Details **16**

B Effect of Different Number of Diagonal Blocks **18**

C Experiments on Re-scaled OFT **19**

D Applying OFT to Convolution Layers **20**

E Comparison between COFT and OFT **21**

F More Qualitative Results **23**
F.1 Subject-driven Generation 23
F.2 Controllable Generation 25

G More Controllable Generation Tasks **34**
G.1 Dense Pose to Human Body 34
G.2 Sketch to Image 36
G.3 Depth to Image 38

H Human Evaluation **40**

I Style Transfer by Adapting Stable Diffusion with Orthogonal Finetuning **41**

J Failure Cases **42**
J.1 Failure Cases in Subject-driven Generation 42
J.2 Failure Cases in Controllable Generation 43

A Experimental Details

To verify the effectiveness of our Orthogonal Fine-tuning (OFT) method, we extensively evaluate the performance of our method in two common text-to-image generation tasks: subject-driven generation and controllable generation. More specifically, we use the exact same task setting as ControlNet [68] and Dreambooth [51] and the baseline implementations were sourced from the GitHub repository Diffusers[2] and ControlNet[3].

Data and Model. For training the convolutional autoencoder from Figure 2, we use 1000 random images from the Oxford 102 Flower dataset [40]. For the task of subject-driven generation, we use the official DreamBooth dataset, which consists of 30 subjects from 15 different classes. For each subject, there are several images and 25 different text prompts. For generating the image-control-caption combinations, we use BLIP [29] to automatically caption the images (pre-trained model weight and code for captioning based on the GitHub repository BLIP[4]). Note, although COCO provides captions for the training and validation split, to be consistent with other image-control-caption combinations, we instead use the BLIP-generated captions as text prompts. For the C2I task, we use the whole COCO 2017 dataset [31] with in total of 180K images; we generate canny edge images as the control signal using the same canny edge detector as ControlNet. For the S2I task, we use the semantic segmentation dataset ADE20K [70] with in total of 24K image-segmentation mask pairs. For the L2F dataset, we use the CelebA-HQ dataset [25], which contains 30K images. Additionally, we demonstrate that OFT also works well in other controllable generation tasks, including depth-to-image (D2I), densepose-to-image (P2I), and sketch-to-image (Sk2I). For the D2I task, we also use the COCO dataset and employ MiDaS [47] to generate depth maps; the pre-trained weights are obtained from the GitHub repository MiDaS[5]. For the P2I task, we use the DeepFashion-MultiModal dataset [24] with in total of 44K clothed human images with the corresponding densepose. For the Sk2I task, we use a subset of the LAION-Aesthetics dataset with approximately 350K images to learn sketch-guided image generation. We use the Stable Diffusion v1.5[6] as the base model.

Subject-driven generation. For training our subject-driven generation diffusion model, we follow the training objective of Dreambooth. More specifically, we use the class-specific prior preservation loss to fine-tune our orthogonal matrices:

$$\mathbb{E}_{\boldsymbol{x},\boldsymbol{c},\boldsymbol{\epsilon},\boldsymbol{\epsilon}',t}[w_t\|\hat{\boldsymbol{x}}_\theta(\alpha_t\boldsymbol{x}+\sigma_t\boldsymbol{\epsilon},\boldsymbol{c})-\boldsymbol{x}\|_2^2+\lambda w_{t'}\|\hat{\boldsymbol{x}}_\theta(\alpha_{t'}\boldsymbol{x}_{\text{pr}}+\sigma_{t'}\boldsymbol{\epsilon}',\boldsymbol{c}_{\text{pr}})-\boldsymbol{x}_{\text{pr}}\|_2^2], \tag{5}$$

with $\boldsymbol{c}_{\text{pr}}$ being the class conditioning vector. For calculating the prior-preservation loss, we additionally need to generate 200 images using the subject's class prompt. Similar to LoRA, we inject our trainable orthogonal matrices into the attention modules of the stable diffusion model. To be comparable with LoRA, we choose the exact same linear layers as LoRA to affect upon: the linear layers $\boldsymbol{W}_q$, $\boldsymbol{W}_k$, $\boldsymbol{W}_v$ and $\boldsymbol{W}_o$. We perform training on 1 Tesla V100-SXM2-32GB GPU using a learning rate of 6×10^{-5}, batch size of 1, and train for approximately 1000 iterations. In the case of COFT, we use $\epsilon=6\times10^{-5}$ to constrain the orthogonal matrices.

Controllable generation. Apart from injecting the trainable OFT weights into the stable diffusion model, we need to add a small encoding model to stable diffusion to encode the control signal. To be comparable with ControlNet [68], we use the same encoding module, which is a shallow 8-layer convolutional network with Scaled Exponential Linear Unit (SELU) activation functions. We also the same training objective as ControlNet. The control signal is encoded and concatenated once with the input to the stable diffusion U-Net. For the LoRA baseline, we use the same encoding module to encode the control signal. For S2I, L2I and P2I, we fine-tune the model for 20 epochs; for C2I and D2I we fine-tune the model for 10 epochs; for Sk2I we fine-tune the model for 8 epochs. We perform training on 4 NVIDIA A100-SXM4-80GB GPUs using a learning rate of 1×10^{-5}, batch size of 4 for L2I and batch size of 16 for the rest of tasks. For fine-tuning with COFT, we use $\epsilon=1\times10^{-3}$.

[2] https://github.com/huggingface/diffusers
[3] https://github.com/lllyasviel/ControlNet
[4] https://github.com/salesforce/BLIP
[5] https://github.com/isl-org/MiDaS
[6] https://huggingface.co/runwayml/stable-diffusion-v1-5/blob/main/v1-5-pruned.ckpt

Evaluation. When evaluating the effectiveness of controllable generation, we primarily focus on evaluating the controllability. Using the consistency metrics introduced in the main paper, we can effectively compute the difference between the control signal and the generated image. For the C2I task, we apply the identical canny filter on the generated image to determine a canny image of the predicted image. Both the control signal canny image and the canny image obtained from the generated images are black-and-white images, with pixel values being either 0 or 1. We evaluate the pixel-wise Intersection over Union (IoU) and F1 score between these two canny predictions. For the S2I task, we compute mean IoU, mean and overall accuracy by deploying a pre-trained semantic segmentation model. More specifically, we use the Segformer[7] [64] model, which is trained on ADE20K (Segformer-B4), to perform semantic segmentation on our generated images. We use the segmentation accuracy as an indication for the overall semantically and structural resemblance of the generated images to the ground truth image. For the L2F task, we compute the mean ℓ_2 distance between the input control landmarks and the landmarks estimated from generated images using facial landmark detector [4].

We also evaluate the generation performance by calculating Fréchet Inception Distance (FID) [19], we use the default setting of the GitHub repository pytorch-fid[8]. The FID is a metric quantifying the similarity between two image dataset. It utilizes 2048-dimensional features, which are derived from the final average pooling layer of a pretrained InceptionV3 network trained on ImageNet dataset. A lower FID score indicates a higher similarity between the datasets.

[7] https://github.com/NVlabs/SegFormer
[8] https://github.com/mseitzer/pytorch-fid

B Effect of Different Number of Diagonal Blocks

We note that the number of diagonal blocks r is an important hyperparameter that effectively controls the number of trainable parameters. It is necessary to perform a sensitivity study on this hyperparameter. Following the same settings as the main paper, we evaluate how r affects OFT in the S2I task. Results in Table 3 show that smaller r (closer to recovering the standard orthogonal matrix) generally works better than larger r. However, we find that a good trade-off between flexibility and parameter-efficiency indeed exists. Empirically, we find that we can use a much bigger r if the dataset is simple, leading to better parameter-efficiency and faster convergence. In the main paper, we always use $r = 4$ because we find that $r = 4$ works well across datasets and tasks. Note that, in terms of the number of inference parameters, both LoRA and OFT have the exact same number of parameters, which is equal to the number of parameters of the underlying stable diffusion model, while ControlNet has an additional control model with 361M parameters.

	ControlNet	$r = 2$	$r = 4$	$r = 8$	$r = 16$
Trainable Parameters	361.3 M	29.5 M	16.3 M	9.7 M	6.4 M
Inference Parameters	1.42 B	1.06 B	1.06 B	1.06 B	1.06 B
mIoU ↑	20.88	**27.18**	27.06	24.09	21.0
mAcc ↑	30.91	39.39	**40.09**	36.95	32.55
aAcc ↑	61.42	**65.24**	62.96	60.25	55.5

Table 3: How the number of diagonal blocks affects the control capability of OFT.

C Experiments on Re-scaled OFT

Since both OFT and COFT transform neurons with orthogonal matrices and do not affect the magnitude of neurons, their magnitude may be sub-optimal with their updated orientations. To address this issue, we propose a re-scaled OFT where the neuron magnitude is refined using the same set of data in the downstream task. Specifically, re-scaled OFT further finetunes the magnitude of neurons without changing their directions. re-scaled OFT can be performed in two manners: (1) *joint fitting*: magnitude fitting can be used simultaneously with OFT or COFT, and (2) *Post-stage fitting*: magnitude fitting can be used after OFT or COFT is finished. An important motivation for re-scaled OFT comes from Figure 2, where we observe that constructing images only with angular information perfectly preserves visual structures, but it also results in a certain degree of color distortion. We hypothesize that this minor color distortion is caused by magnitude loss and fixing this issue can potentially improve the visual quality of generated images.

Notably, re-scaled OFT does not change the hyperspherical energy since it does not change the direction of neurons - all the nice properties of OFT and COFT on hyperspherical energy are still perfectly preserved. Therefore, the advantage of structural preservation is also inherited.

To simplify the experiments and validate the effectiveness of re-scaled OFT, we perform post-stage magnitude fitting on the COFT model and compare the FID between the original validation images and the generated images (using the control signals extracted from validation images). The reason we use FID here is that FID is more sensitive to color distortion, while the consistency metric only measures the structural preservation. Table 4 shows that magnitude fitting can indeed improve the FID of COFT and is beneficial to COFT.

Magnitude fitting is lightweight and can be implemented by simply adding one trainable parameter for each layer we modify; the parameter has the shape of $(N \times 1)$, with N corresponds to the number of neurons in that specific layer. The performance gain shown in Table 4 is achieved by performing *Post-stage fitting* on a COFT-finetuned model for only one additional epoch. Moreover, we expect that the joint fitting re-scaled OFT can lead to better performance.

	SD	ControlNet	T2I	LoRA	COFT	Re-scaled COFT
FID ↓	41.2	30.9	33.1	30.9	30.8	**30.2**

Table 4: FID on the segmentation to image task (ADE20K). $r = 4$ is used here.

D Applying OFT to Convolution Layers

In the original setting [22], LoRA is only applied to the linear layers of the attention modules. To be a fair comparison, we also apply OFT to these weights. However, OFT is not limited to linear layers but can easily be adapted to convolution layers by transforming the convolutional neurons. We highlight the compatibility of OFT and COFT for finetuning convolution layers. More interestingly, sharing the parameters of diagonal blocks in $\boldsymbol{R}$ becomes interpretable in convolution layers. With a suitable setup, orthogonal matrices with sharing diagonal blocks can transform the convolution kernel in a channel-sharing manner (or in a spatial manner), implying that the same orthogonal transformation is applied to all channels. This shares similar intuition with depth-wise convolution.

For this ablation experiment, we study the performance of applying OFT to the convolution layers in the ResNet blocks of the stable diffusion model. In this experiment, we use COFT as the baseline method and consider the controllable generation (segmentation to image) as an example. We have both quantitative (Table 5) and qualitative results (Figure 11). We can empirically observe that by only fine-tuning the convolutional layers, we can also achieve some degree of control. By simultaneously fine-tuning both linear and convolutional layers, we achieve a slightly better FID score. Note, for fine-tuning convolutional layers, we let r be equal to the number of channels of convolutional neurons in that layer.

	COFT (attention)	COFT (conv)	COFT (extended)
FID ↓	30.8	39.8	**30.4**

Table 5: FID results of applying COFT to different types of layers. (with $r = 4$)

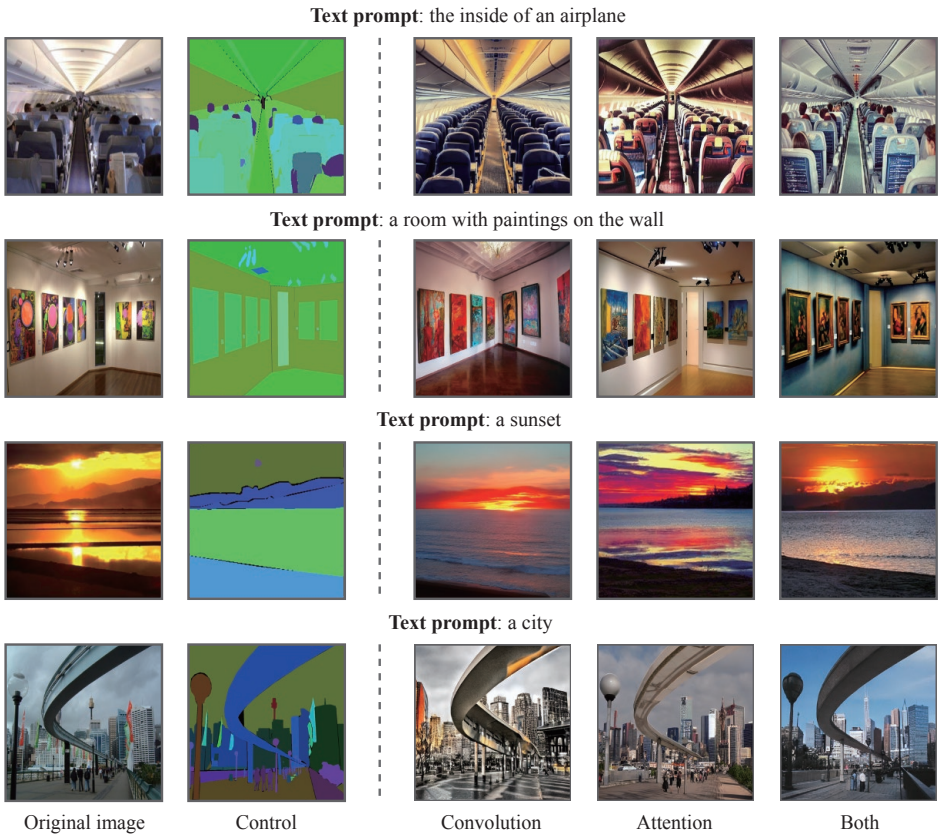


Figure 11: Controllable generation results of applying COFT to different types of layers.

E Comparison between COFT and OFT

We have already provided many qualitative examples for COFT and OFT in the main paper. One may question the fundamental difference between OFT and COFT. Based on the intuition behind COFT, the deviation constraint is introduced to improve the training stability. We demonstrate the training stability of COFT with a qualitative example in subject-driven generation. Results in Figure 12 and Figure 13 show that, despite being much more stable than existing methods, OFT will eventually generate collapsed images at the 9000-th iteration. In contrast, COFT still produces visually appealing images. We train both OFT and COFT with a learning rate of 1×10^{-5} and constrain COFT with $\epsilon = 1 \times 10^{-5}$.

Figure 12: Qualitative comparison between COFT and OFT on subject-driven generation.

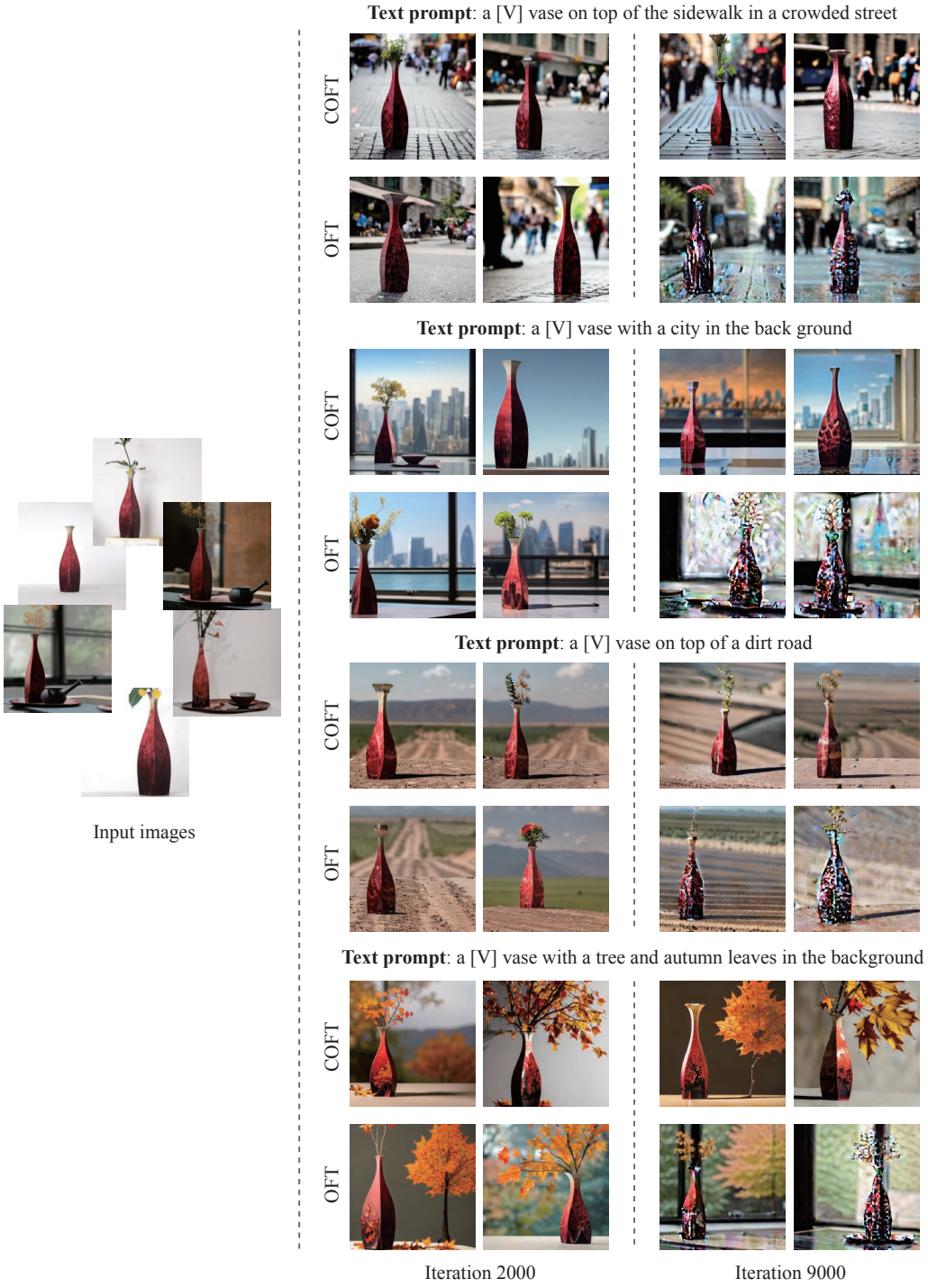


Figure 13: Qualitative comparison between COFT and OFT on subject-driven generation.

F More Qualitative Results

F.1 Subject-driven Generation

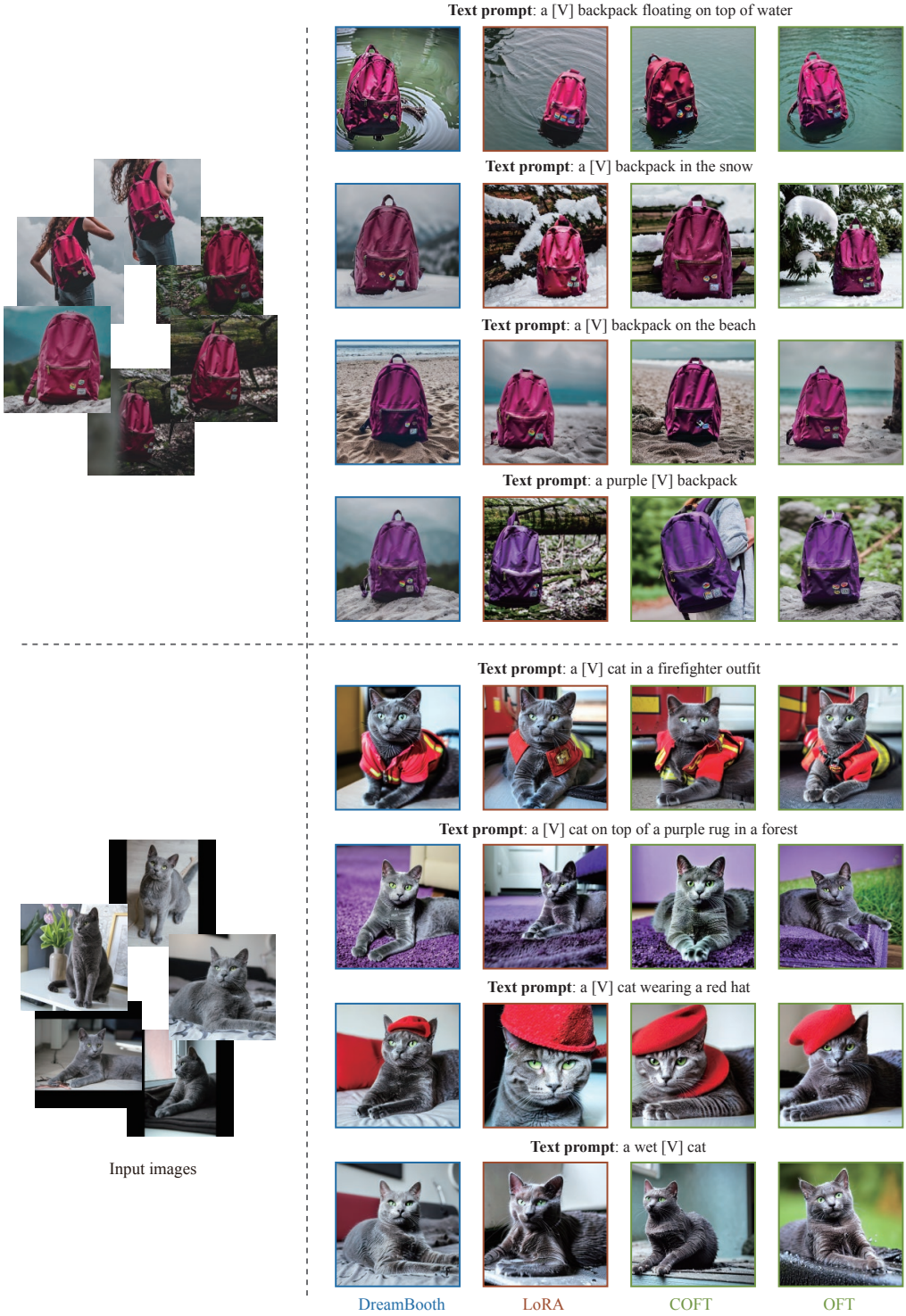


Figure 14: More qualitative results on subject-driven generation.

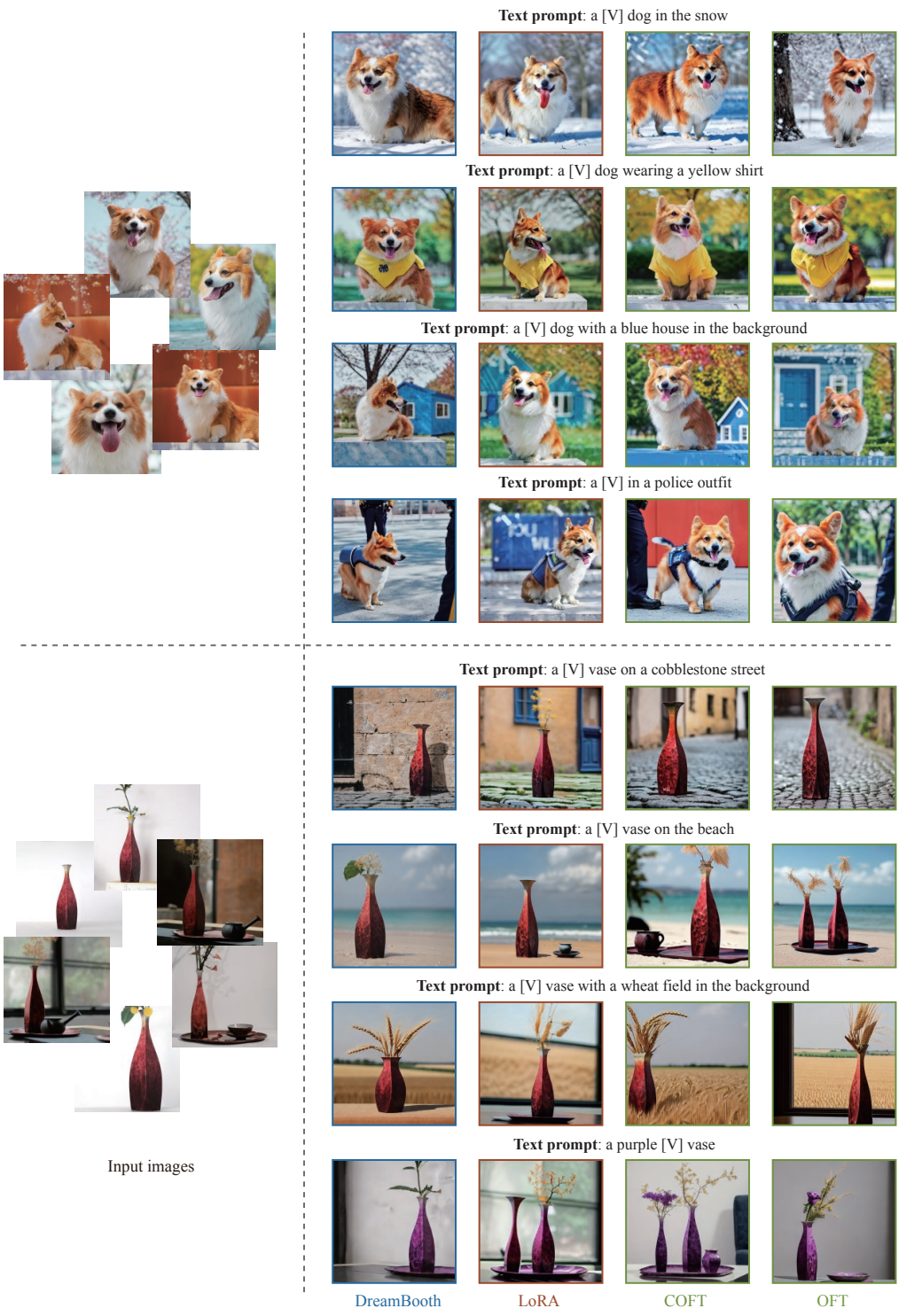


Figure 15: More qualitative results on subject-driven generation.

F.2 Controllable Generation

F.2.1 Segmentation to Image

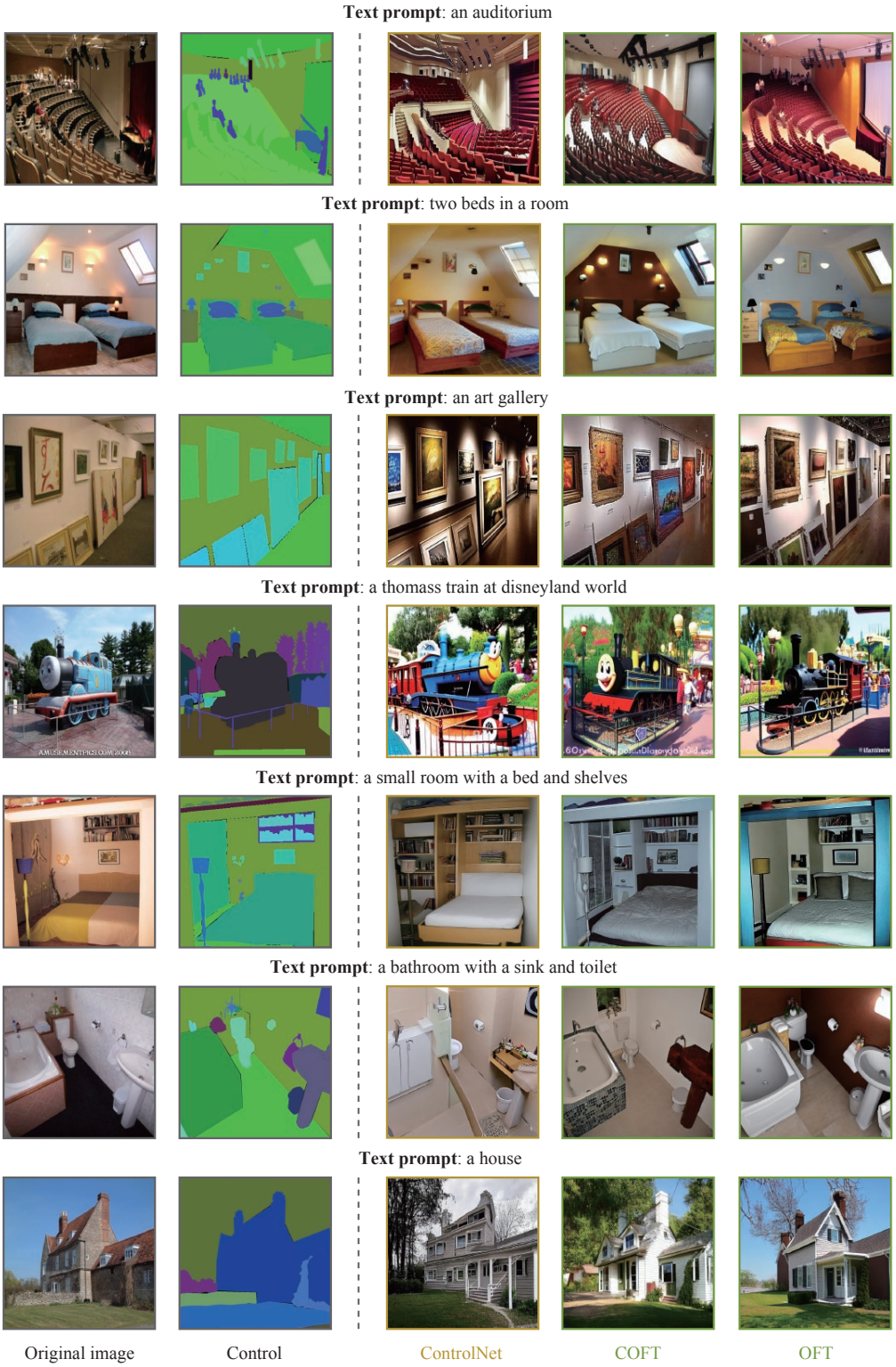


Figure 16: More qualitative results of OFT and COFT on the segmentation to image generation task.

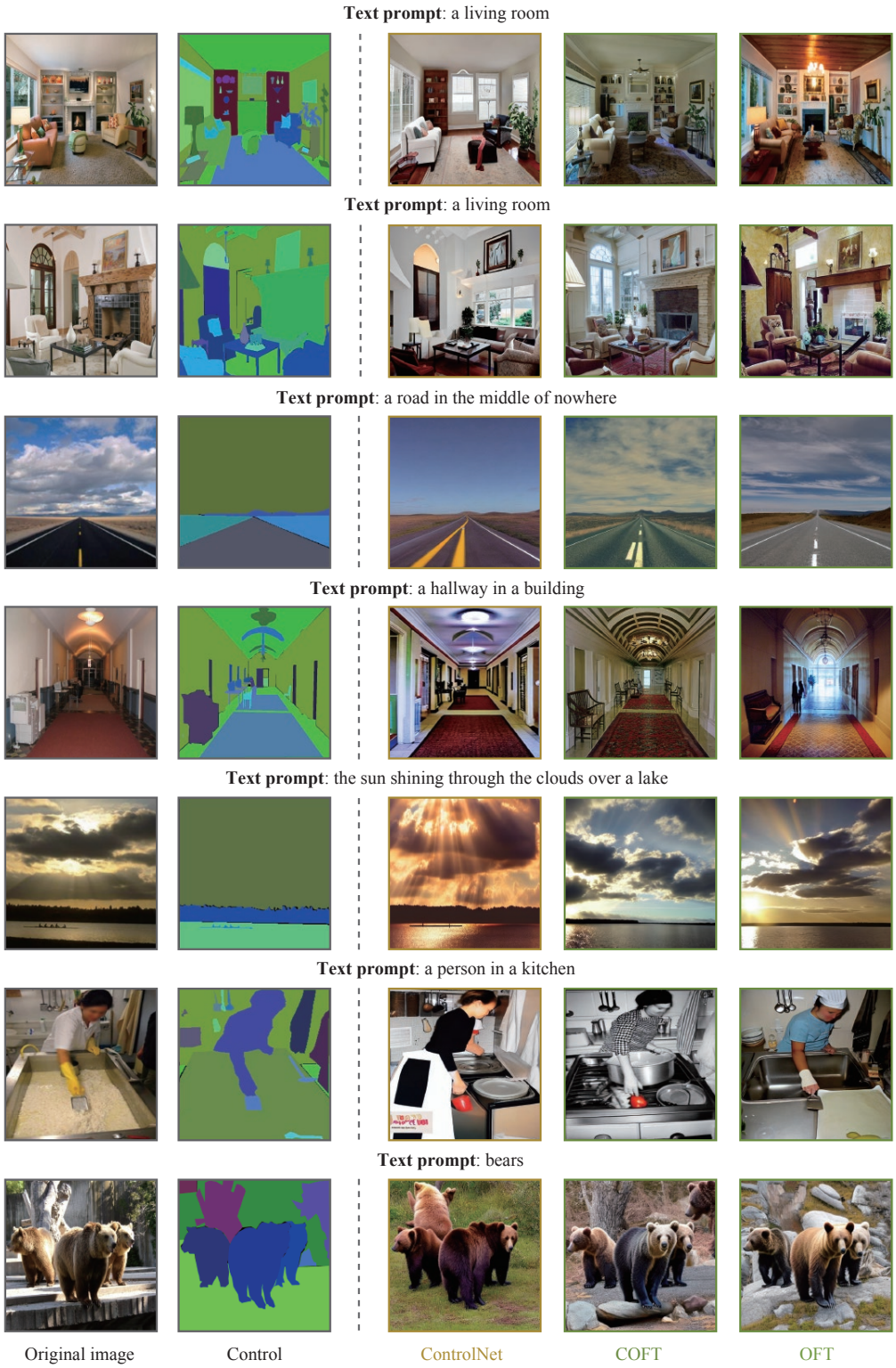


Figure 17: More qualitative results of OFT and COFT on the segmentation to image generation task.

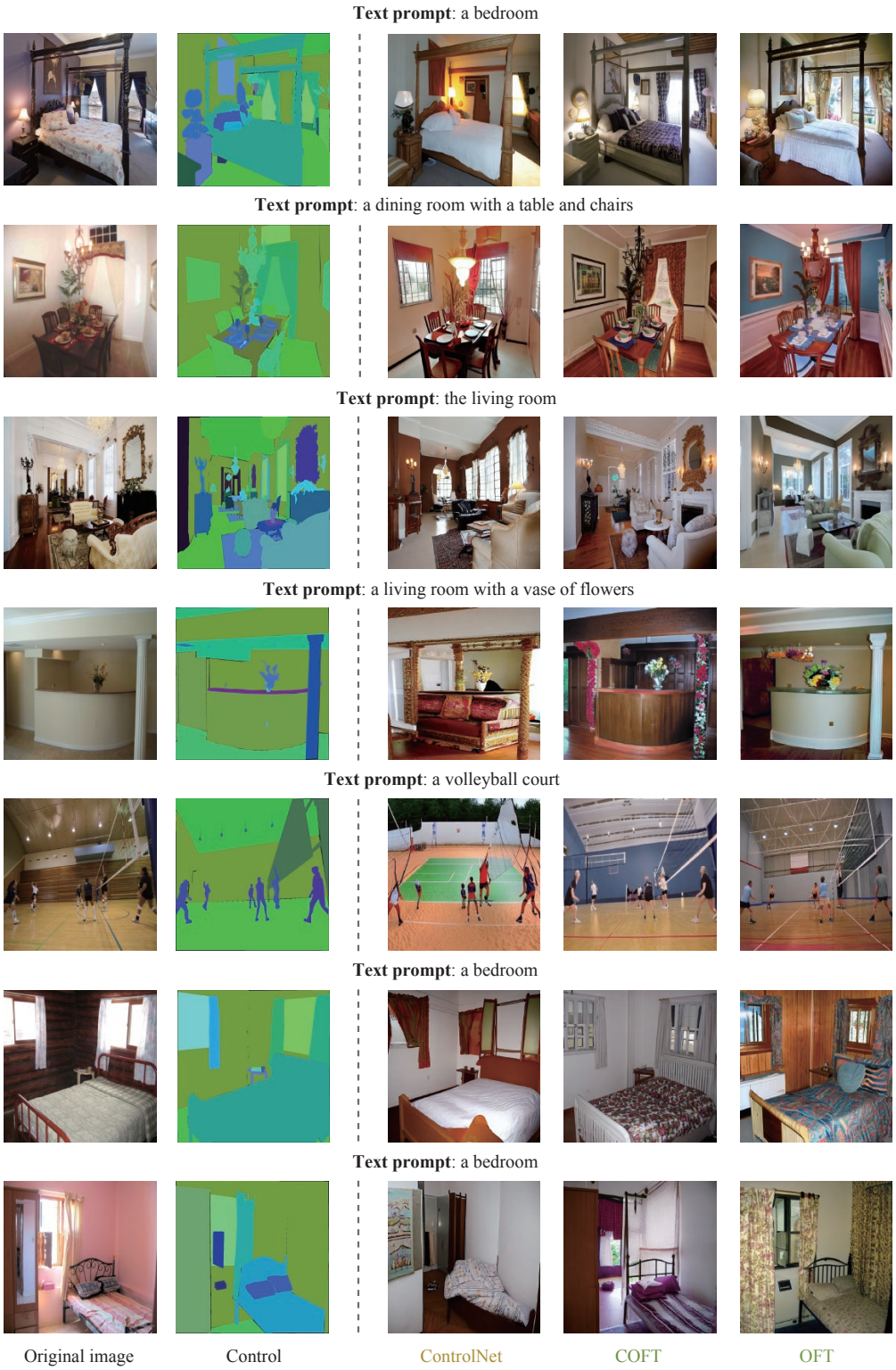


Figure 18: More qualitative results of OFT and COFT on the segmentation to image generation task.

F.2.2 Canny Edge to Image

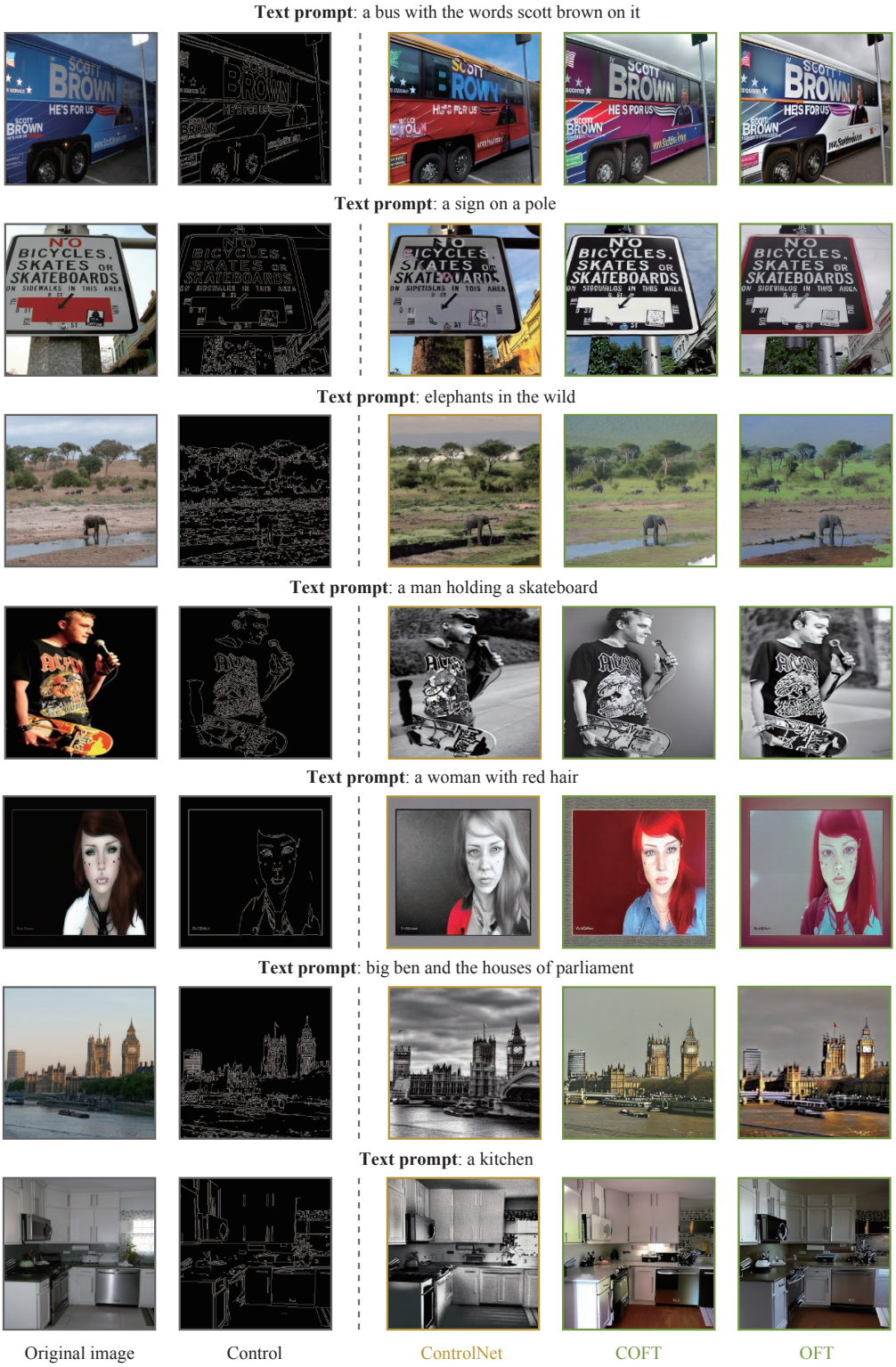


Figure 19: More qualitative results of OFT and COFT on the Canny edge to image generation task.

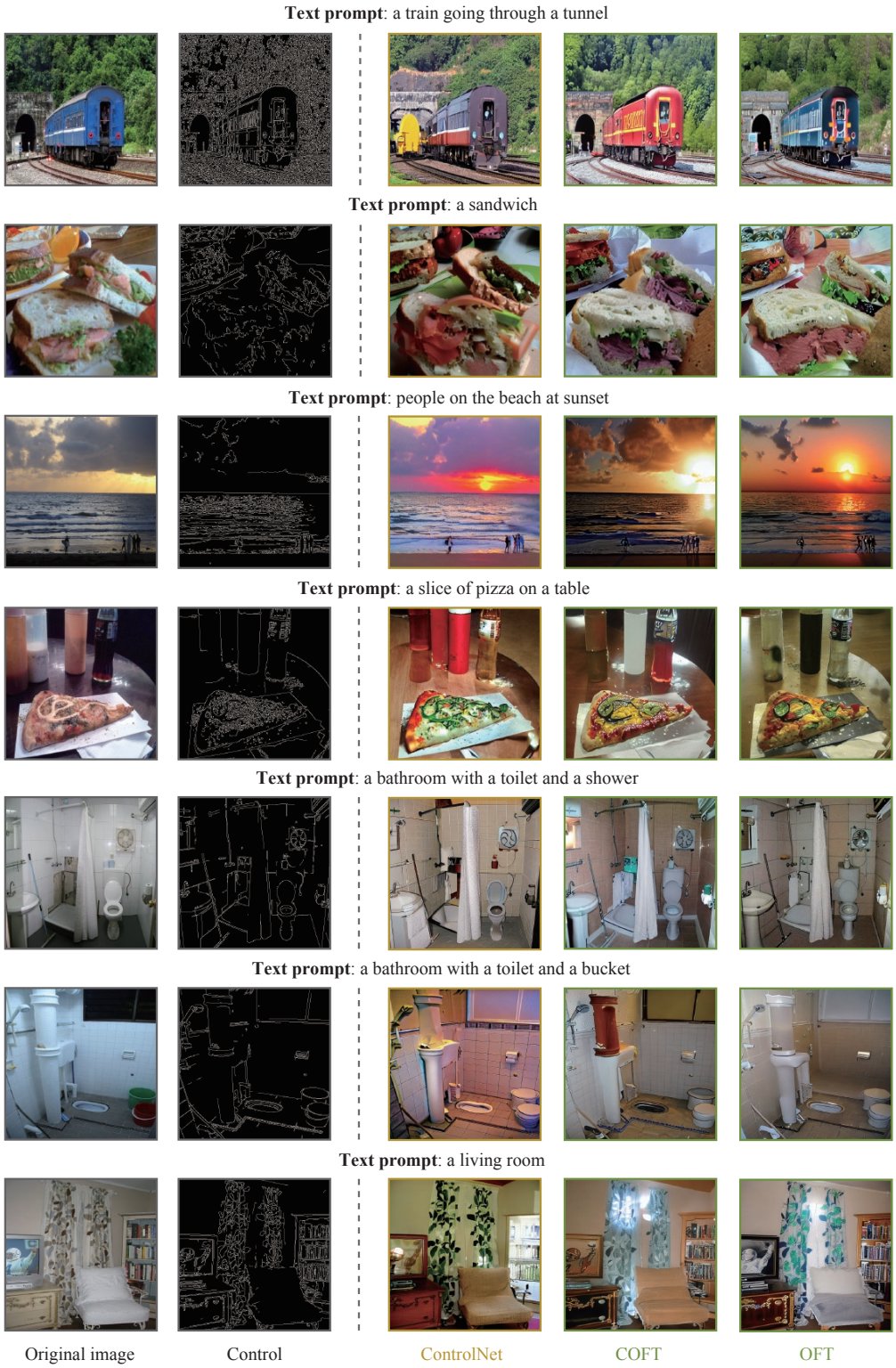


Figure 20: More qualitative results of OFT and COFT on the Canny edge to image generation task.

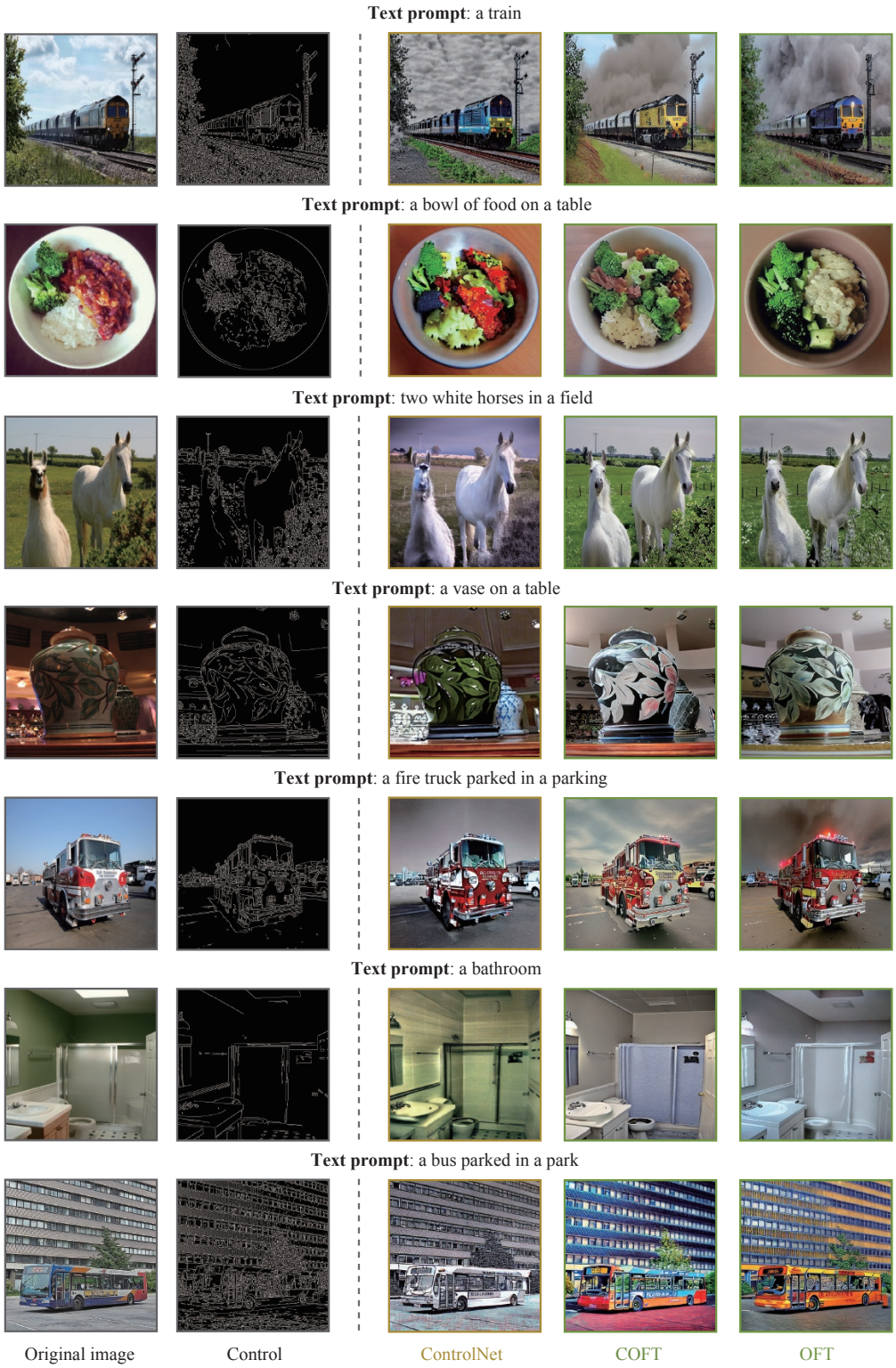


Figure 21: More qualitative results of OFT and COFT on the Canny edge to image generation task.

F.2.3 Landmark to Face

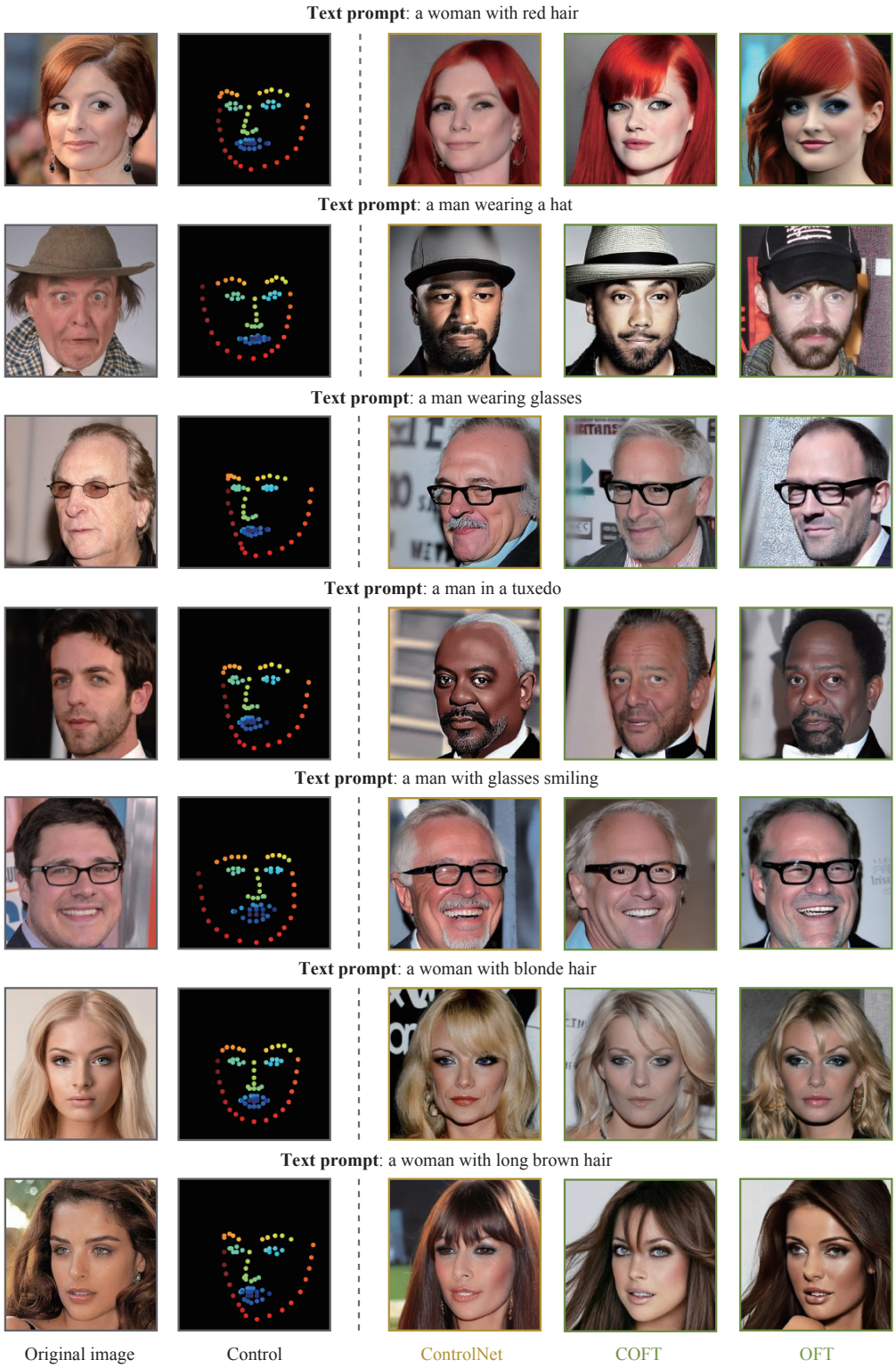


Figure 22: More qualitative results of OFT and COFT on the landmark to face generation task.

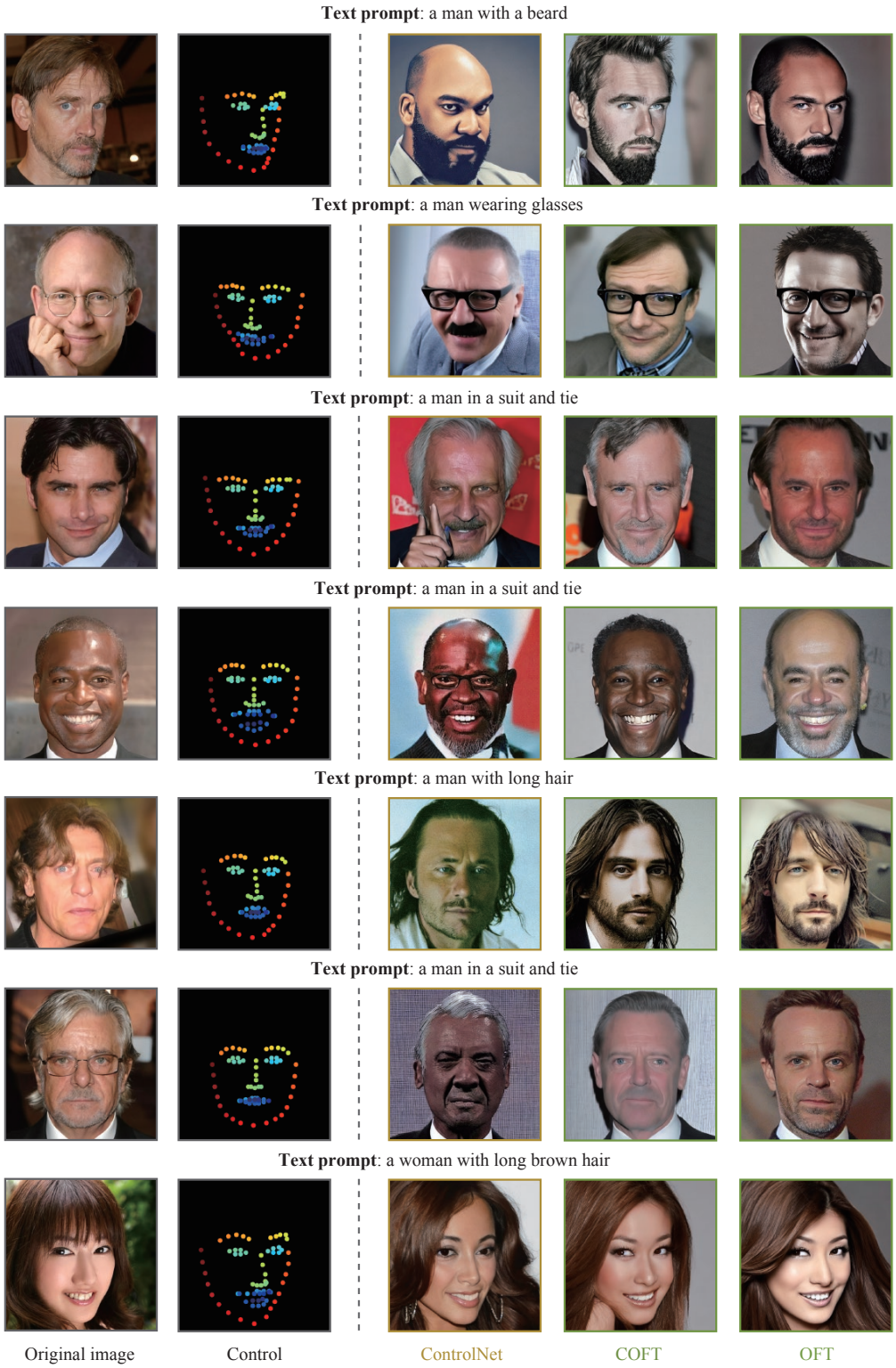


Figure 23: More qualitative results of OFT and COFT on the landmark to face generation task.

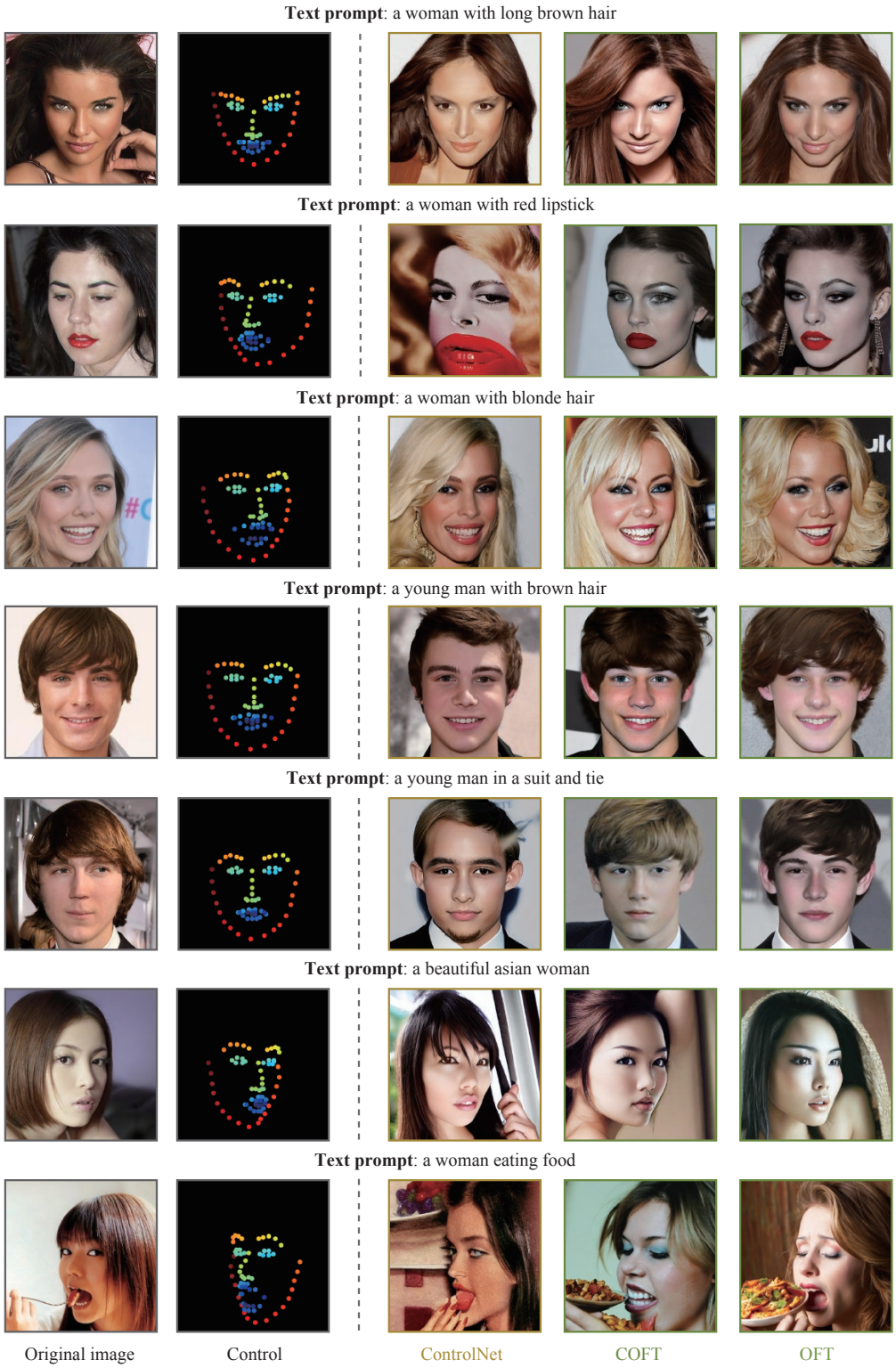


Figure 24: More qualitative results of OFT and COFT on the landmark to face generation task.

G More Controllable Generation Tasks

G.1 Dense Pose to Human Body

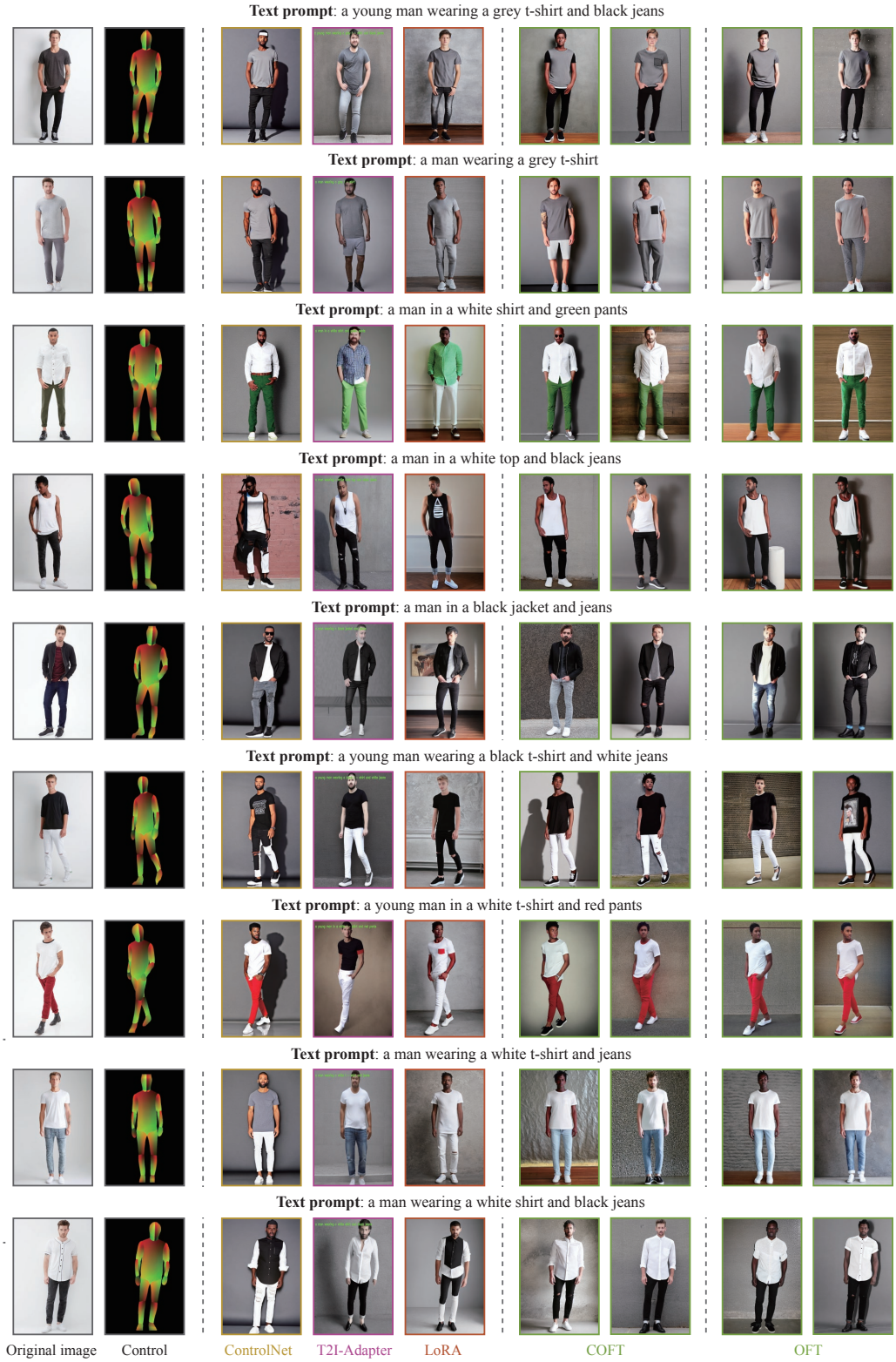


Figure 25: Qualitative comparison among different methods on the dense pose to human body generation task.

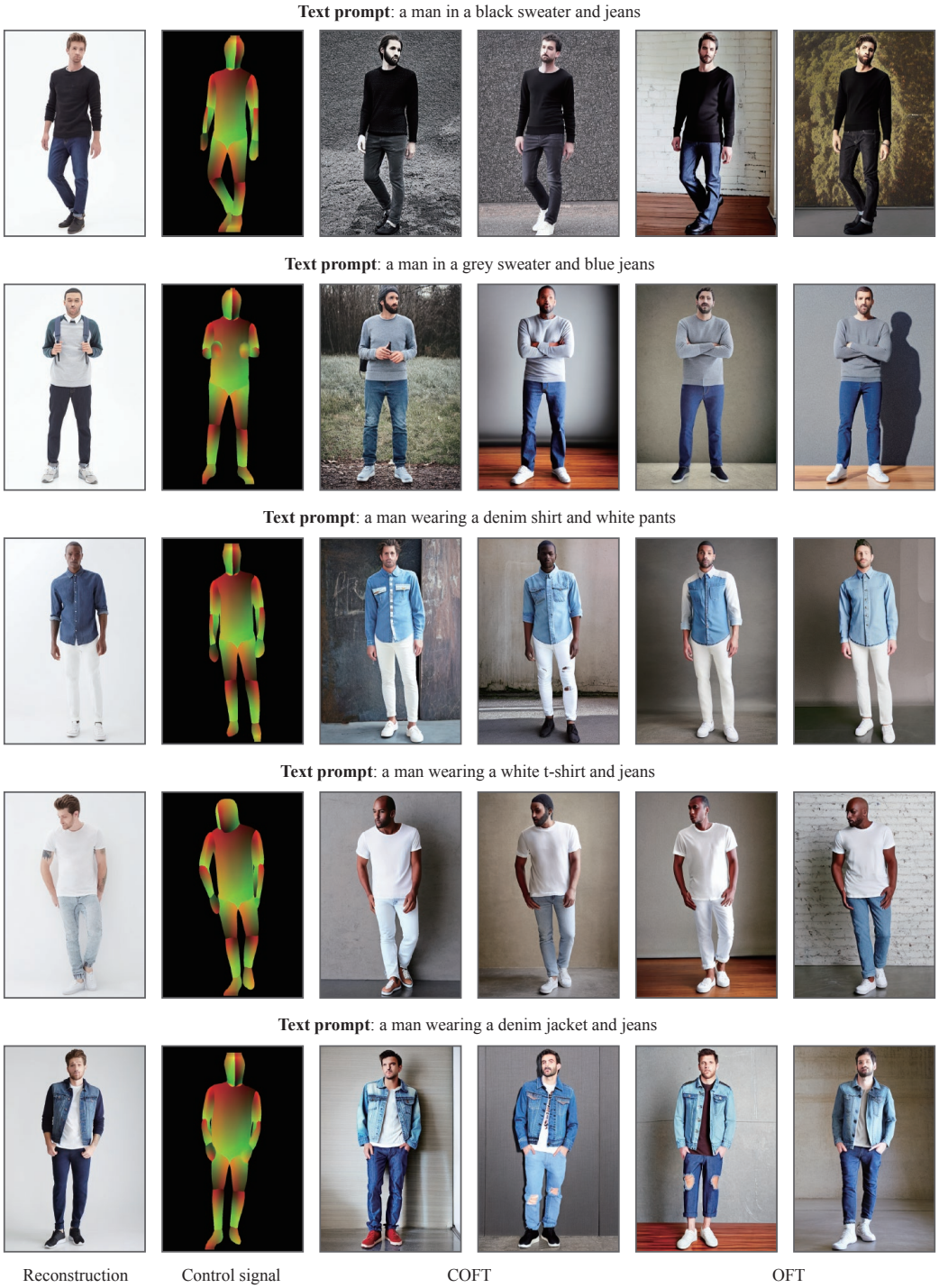


Figure 26: More qualitative results of COFT and OFT on the dense pose to human body task.

G.2 Sketch to Image

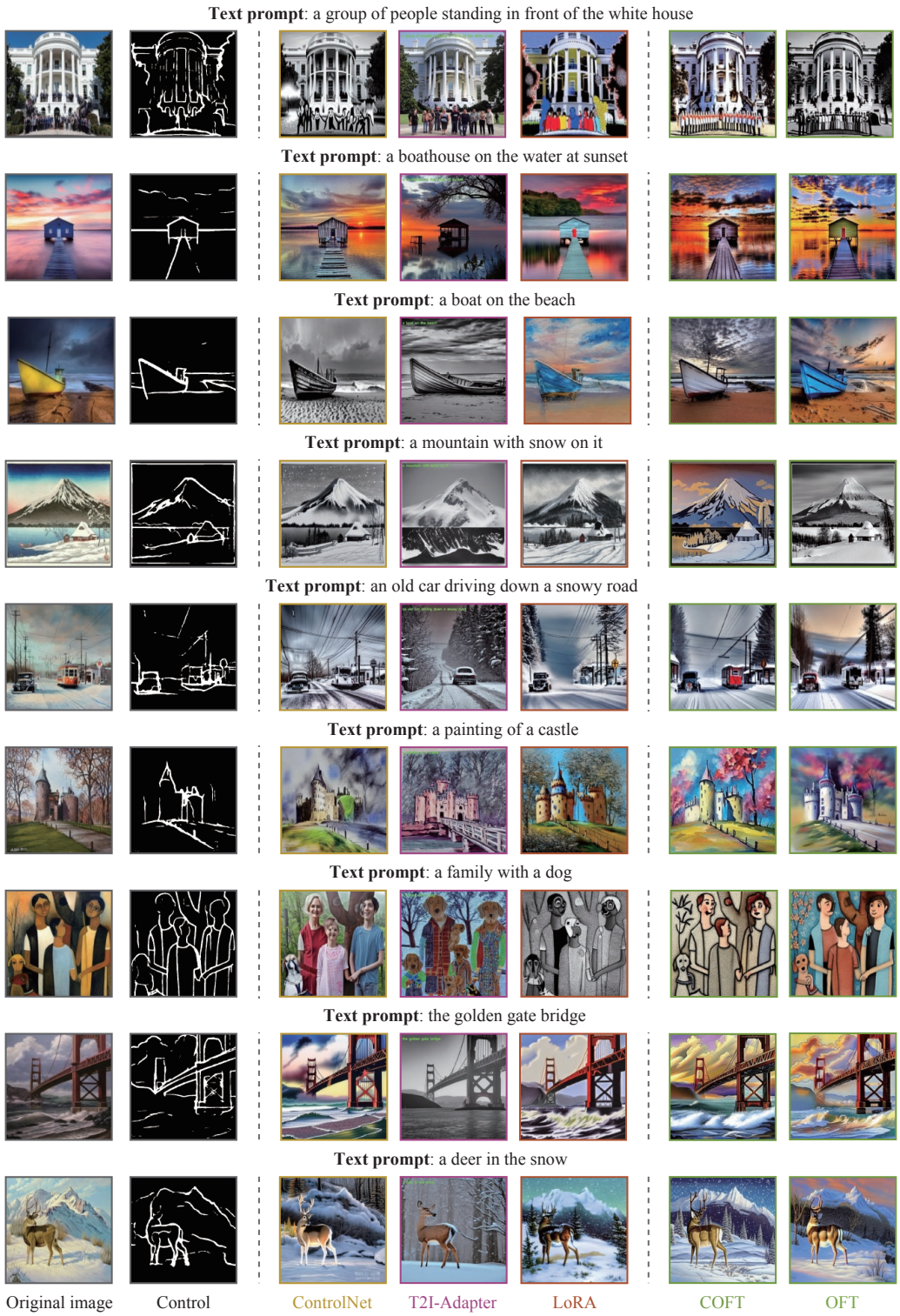


Figure 27: Qualitative comparison among different methods on the sketch to image generation task.

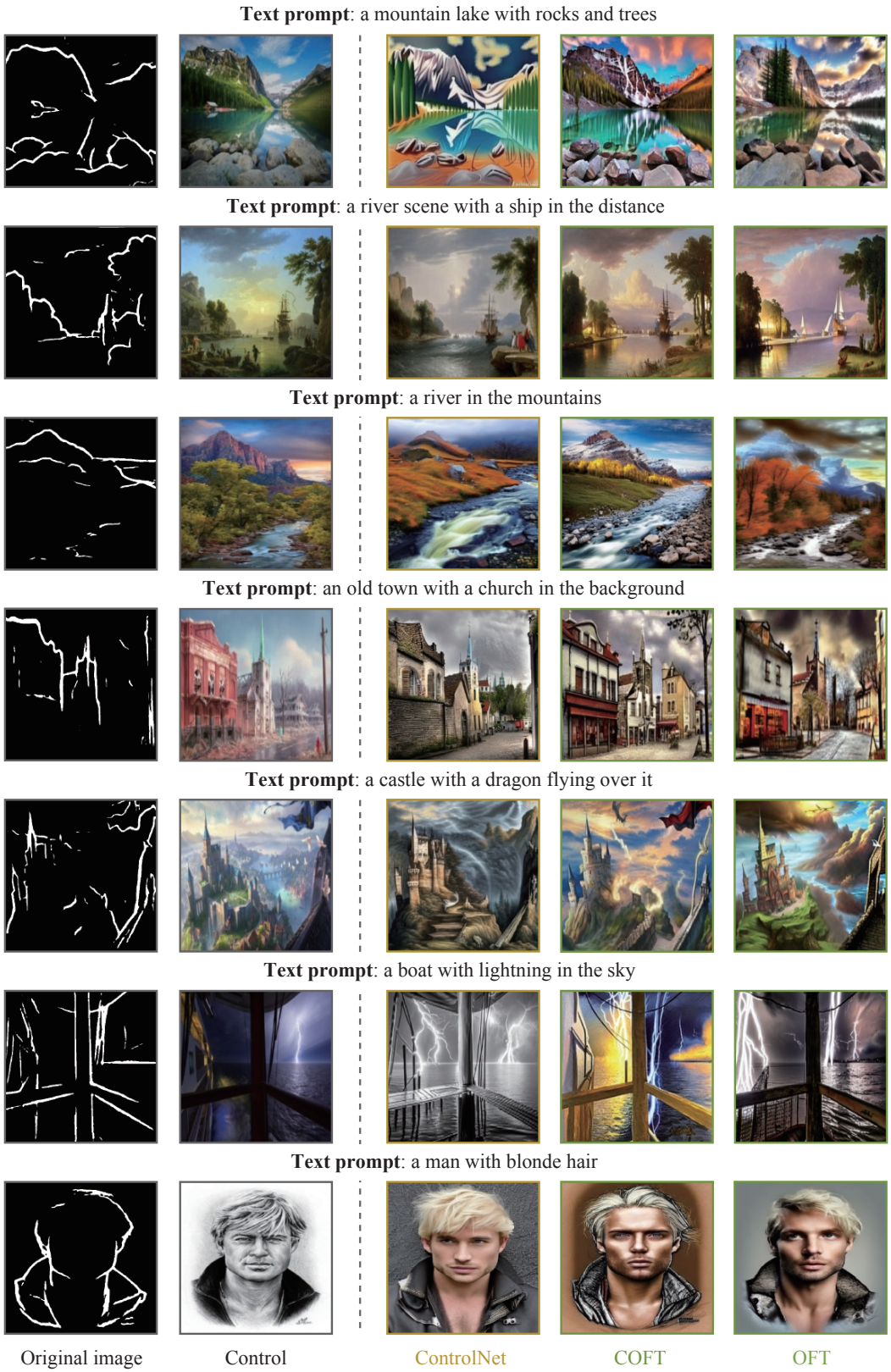


Figure 28: More qualitative comparison on the sketch to image generation task.

G.3 Depth to Image

Figure 29: Qualitative comparison among different methods on the depth to image generation task.

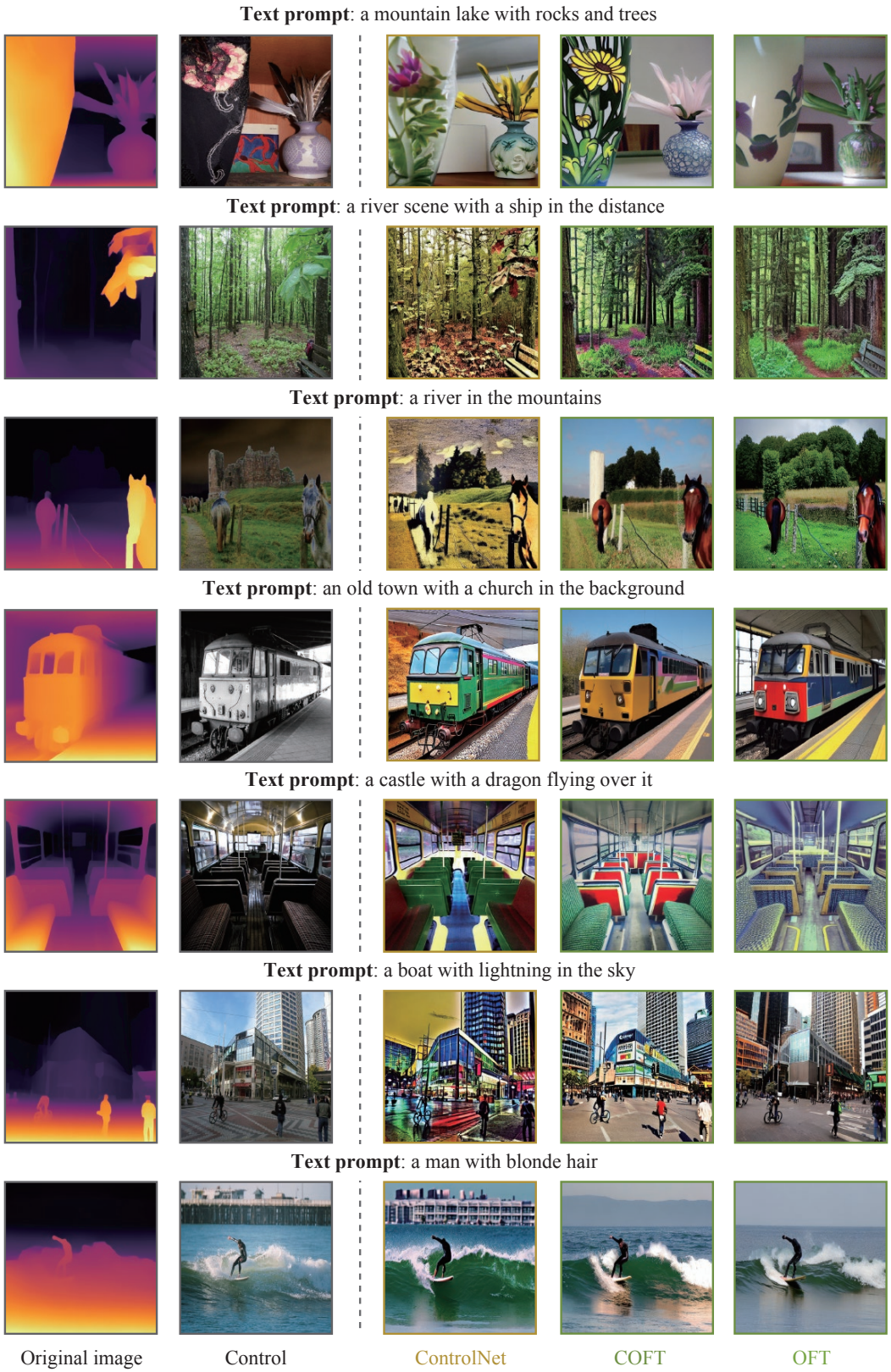


Figure 30: More qualitative comparison on the depth to image generation task.

H Human Evaluation

Human evaluation settings. We also carried out a structured human evaluation for the **subject-driven generation** task, involving 50 participants. Here's a breakdown of our evaluation process:

- **Selection of subjects**: we picked 7 subjects from the DreamBooth dataset[9] at random.
- **Image and prompt**: for every subject, 4 unique text prompts were chosen at random. This resulted in a total of 28 distinct subject-prompt combinations. For every single one of the 28 tasks, we randomly sampled an image generated by each of the three methods - DreamBooth, LoRA, and OFT.

Every participant was asked to answer three single-selection questions for each task:

- **Subject fidelity**: which image best preserves the identity of the subject? In other words, which generated image resembles the most the original subject?
- **Text alignment**: which image matches the given text description the best?
- **Overall image quality**: out of the options, which image has the best overall quality?

The methods were assessed at two specific points during their fine-tuning phase: at the **1000th** iteration, a checkpoint where these methods typically exhibit best performance, and at the **10,000th** iteration, a checkpoint used to measure the stability of the finetuning process over an extended period.

Results. The results are given in Table 6, indicating the proportion of participants who chose a particular method based on the above criteria. We can see that OFT is more favored after finetuning Stable Diffusion with 1000 iterations and after 10000 iterations. We note that OFT delivers significantly better image quality and text-following ability than both DreamBooth and LoRA after a relatively large number of finetuning iterations.

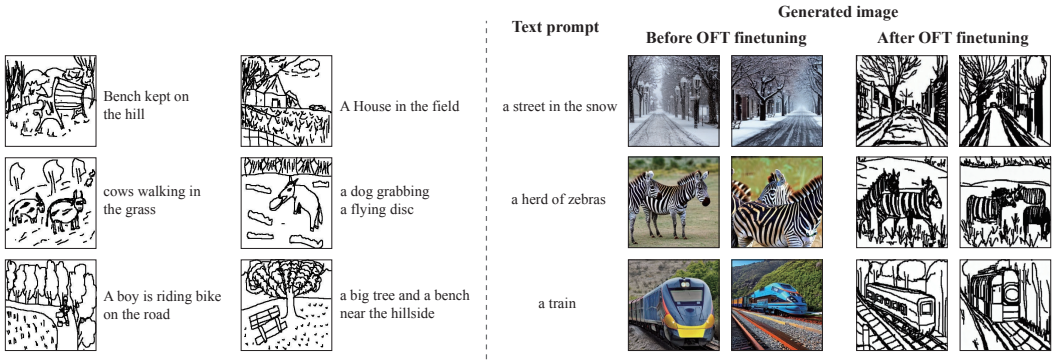
	Iteration 1000			Iteration 10000		
Metric	DreamBooth	LoRA	OFT	DreamBooth	LoRA	OFT
Subject Fidelity	42.0%	15.4%	**42.6%**	22.4%	1.4%	**76.2%**
Text Alignment	18.6%	24.7%	**56.7%**	2.6%	1.4%	**96.0%**
Overall Image Quality	35.7%	19.2%	**45.1%**	11.6%	0.8%	**87.6%**

Table 6: Participant voting percentages for subject fidelity, text alignment and overall image quality.

[9] https://github.com/google/dreambooth

I Style Transfer by Adapting Stable Diffusion with Orthogonal Finetuning

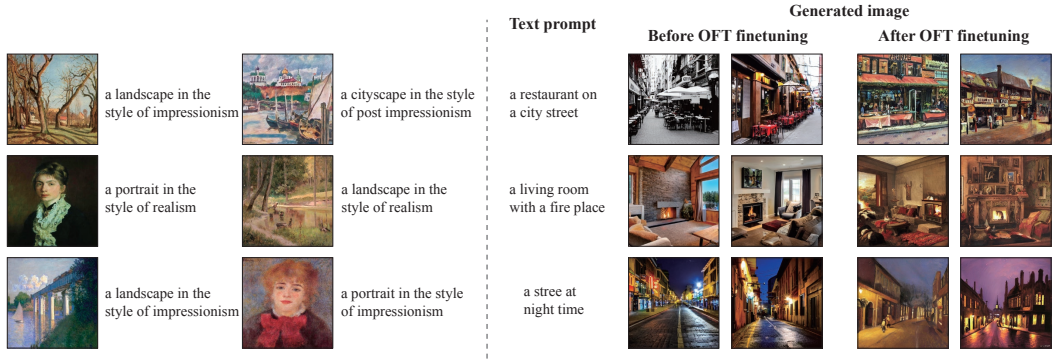
Stable Diffusion can generate images based on the input text prompts. Without any adaptation, inputting text prompts to a pre-trained Stable Diffusion model will result in images that resemble natural images. We can finetune the pre-trained Stable Diffusion model on a custom dataset, to adapt the style of the generated images to the custom dataset. To demonstrate the effectiveness of orthogonal finetuning, we show qualitative results of adapting Stable Diffusion on the Sketch Scene dataset[10] after finetuning for 20000 iterations in Figure 31 and on the Wikiart dataset[11] after finetuning for 30000 iterations in Figure 32. We train on a single NVIDIA A100-SXM4-80GB GPU using a learning rate of 1×10^{-4}, batch size of 1, and 4 as the number of gradient accumulation steps.



(a) Example image-text pairs in the sketch-scene dataset

(b) Text-to-image generation after finetuning on the sketch-scene dataset

Figure 31: Direct OFT Finetuning of Stable Diffusion on the sketch-scene dataset.



(a) Example image-text pairs in the wikiart-caption dataset

(b) Text-to-image generation after finetuning on the wikiart-caption dataset

Figure 32: Direct OFT Finetuning of Stable Diffusion on the wikiart-caption dataset.

[10] https://huggingface.co/datasets/zoheb/sketch-scene

[11] https://huggingface.co/datasets/fusing/wikiart_captions

J Failure Cases

We also show a few failure cases of OFT and COFT. Figure 33 gives three failure cases in subject-driven generation. Figure 34 gives three failure cases in controllable generation.

J.1 Failure Cases in Subject-driven Generation

In subject-driven generation, OFT and COFT will sometimes fail to ground the text attribute to the intended object. In the cat example, both OFT and COFT will sometimes generate other red objects, instead of generating a red cat.

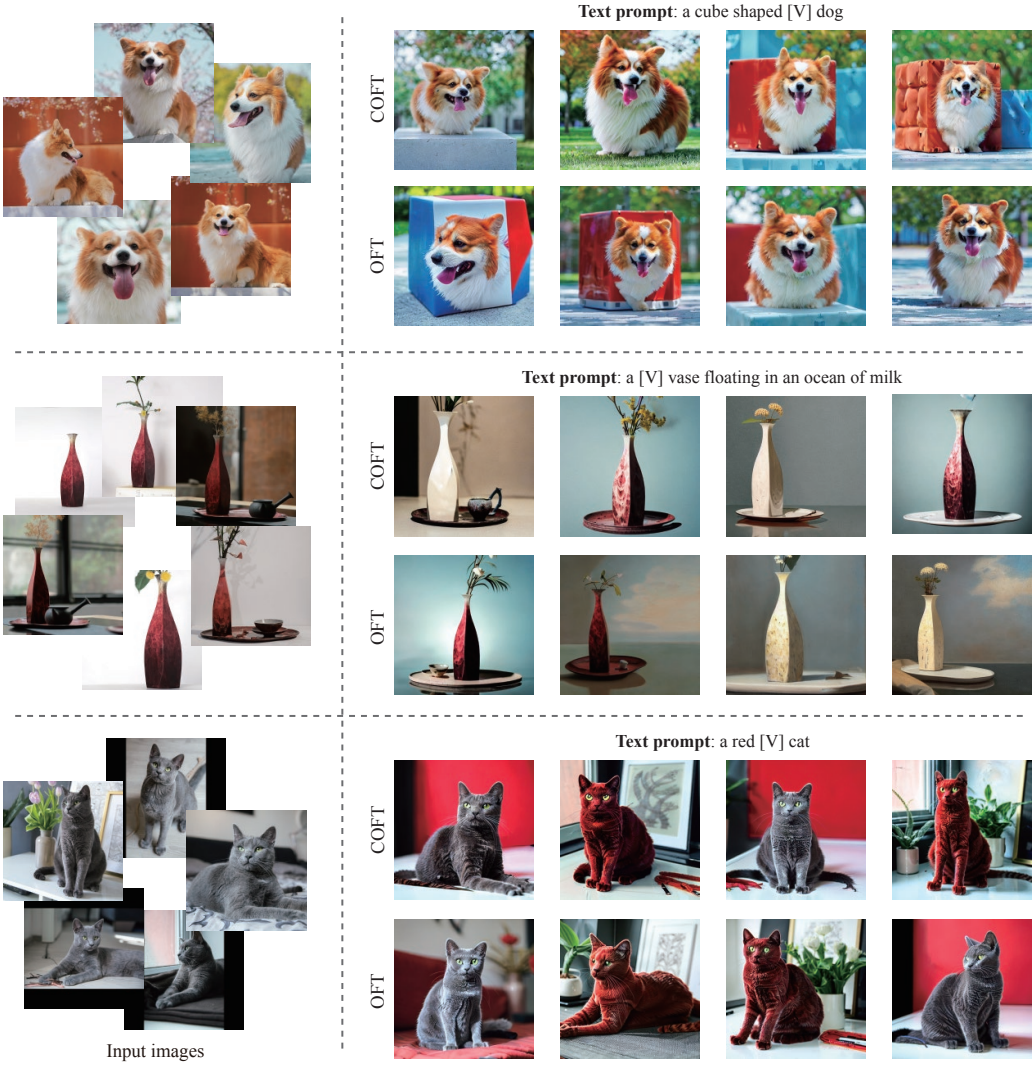


Figure 33: Some failure cases in subject-driven generation.

J.2 Failure Cases in Controllable Generation

Both OFT and COFT will sometimes hallucinate complicated structural details in a large region with the same semantics. Despite still being visually plausible, these generated images cannot match the original segmentation maps.

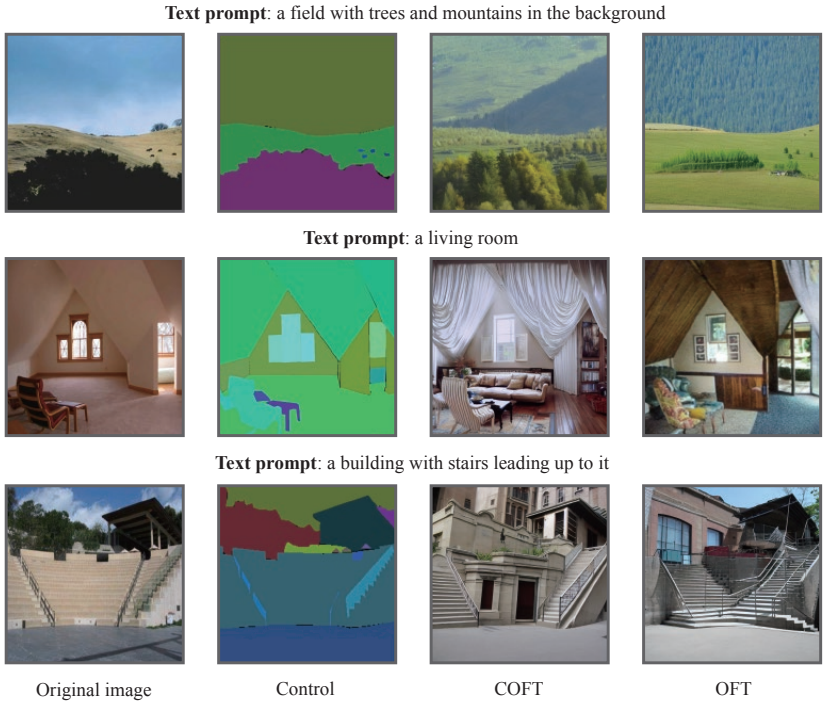


Figure 34: Some failure cases in controllable generation.