Color Alignment in Diffusion – Supplementary Material

Ka Chun Shum[1] Binh-Son Hua[2] Duc Thanh Nguyen[3] Sai-Kit Yeung[1]

[1]Hong Kong University of Science and Technology [2]Trinity College Dublin [3]Deakin University

Abstract

In this supplementary material, we present additional comparisons of our method and existing baselines in Sec. 1. We provide ablation studies on auxiliary technical components of our method in Sec. 2. Finally, we describe more details of our core technique, experimented dataset, and implementation settings in Secs. 3 to 5.

1. Additional comparison results

In Sec. 4.4 in the main paper, we present qualitative comparisons of our color-aligned diffusion method and existing baselines, highlighting the effectiveness of our method in color-conditioned image synthesis on in-the-wild color conditions. In this section, we provide additional comparison results on manual drawing conditions in Fig. 1, which also confirms the superior effectiveness of our method over existing baselines.

We also conduct comparisons of our work with recent commercial products (Ideogram-2.0 [10] and Playground-v3 [6]) which offer non-spatial palette conditioning, and a recent VLM GPT-4o [1]. We perform qualitative comparisons due to limited access to these methods. As shown in Fig. 2, these approaches condition the image synthesis roughly on input palettes, leading to less accurate results with unwanted or missing colors.

Moreover, we include 300 extra comparison results in the supplementary files, packaged in the `./Additional_Results` folder. We encourage readers to review them for a more thorough assessment of our color-conditioned image synthesis based on our defined criteria, including the *accuracy*, the *completeness*, and the *disentanglement* of generated colors, where our method significantly outperforms existing baselines. Please note that all of these results are generated using the implementation settings described in Secs. 4.1 to 4.3 in the main paper, and in Secs. 4 and 5 in this supplementary material.

2. Additional ablation studies

We conduct additional ablation studies on auxiliary technical components of our proposed color alignment method.

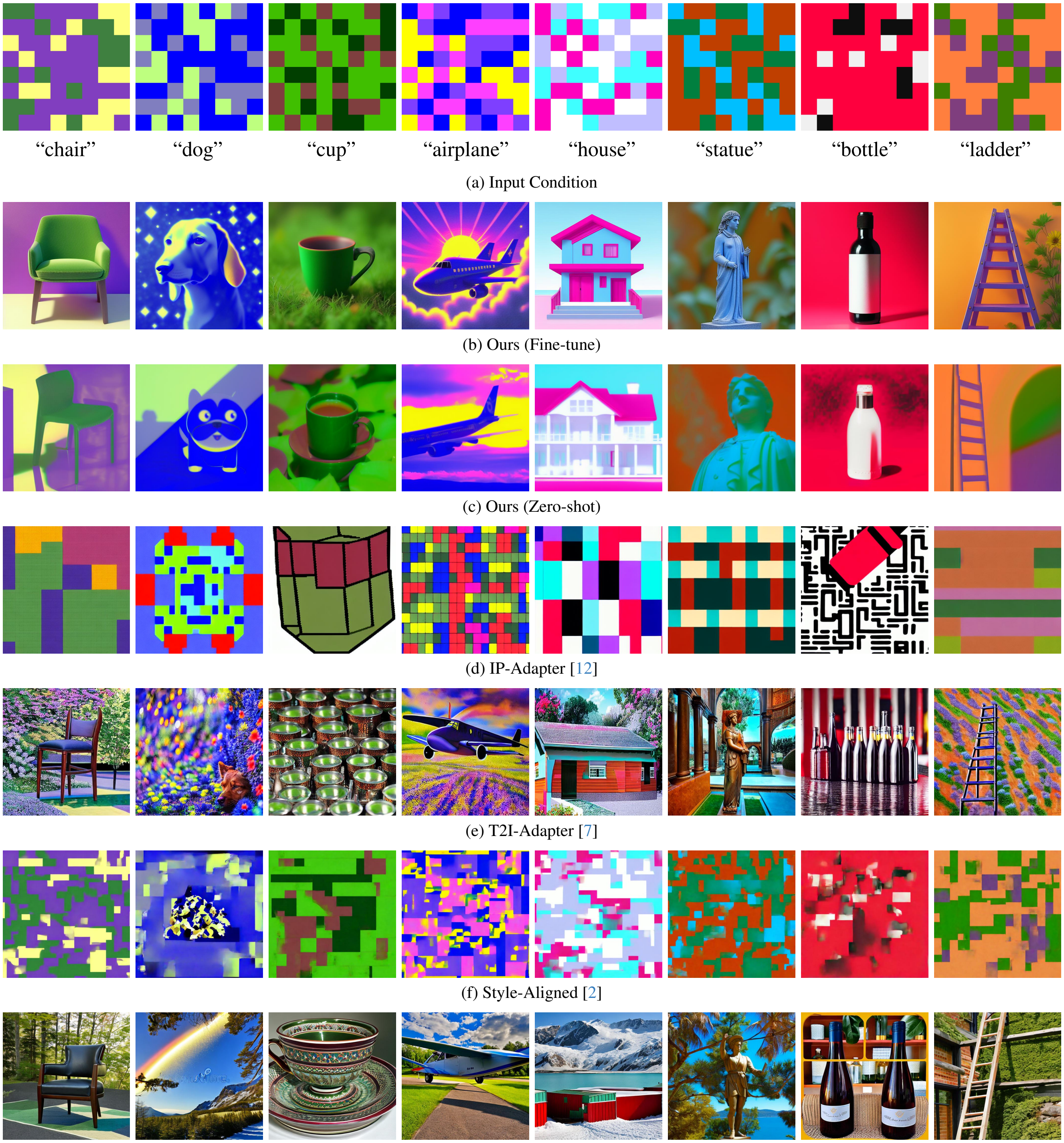
2.1. Blurring before latent encoding

Recall that for color-aligned latent diffusion, we blur the image color condition $\mathbf{x}_0$ before encoding it into the latent representation $\mathbf{z}_0$ for subsequent processes (Secs. 3.3 and 3.4 in the main paper). We found it beneficial for reducing local high-frequency information in $\mathbf{x}_0$, leading to more accurate color encoding in $\mathbf{z}_0$. Particularly, we implement the blurring operation as bilinear down-sampling and then bilinear up-sampling of $\mathbf{x}_0$, with the down- and up-sampling sizes defined as the strength of the blur (e.g., a strength of 3 means down-sampling to one-third of its size and then up-sampling back to its original size).

We demonstrate the effect of the blurring operation in Fig. 3. As shown, without blurring $\mathbf{x}_0$, the results tend to have dotted and fragmented texture (see the second column, better to be zoomed in). We found that a strength of 3 effectively balances texture smoothing and color conditioning, avoiding excessive strength that can make the output overly smooth, blurry, and texture lacking. A quantitative evaluation of the blurring operation is presented in Tab. 1, which clearly shows that our current setting (i.e., strength=3) well balances all the color-conditioned image synthesis criteria evident by respective performance metrics.

2.2. Late-time stopping in latent color alignment

In our latent diffusion process, we propose to suspend the color alignment (Eqs. 5 and 8 in the main paper) at late time steps to allow for refinement of the final generated latent. This aligns with the goal of late time steps in the original diffusion, which focuses on refining local details of the final output. As shown in Fig. 4, without late-time stopping of the alignment, generated colors lack photo-realistic attributes such as natural lighting, vivid shadow, and clear semantics. We found stopping at time steps $t < 200$ enables the generation of these attributes. A too early stopping results in violations of the input color conditions, such as unintended



(a) Input Condition

(b) Ours (Fine-tune)

(c) Ours (Zero-shot)

(d) IP-Adapter [12]

(e) T2I-Adapter [7]

(f) Style-Aligned [2]

(g) ControlNet-Color [13]

Figure 1. **Additional qualitative results of color-aligned latent diffusion on manual drawing conditions**. Each input (first row) includes a manual drawing as color condition. Targeting a text prompt, each column presents results of experimented methods.

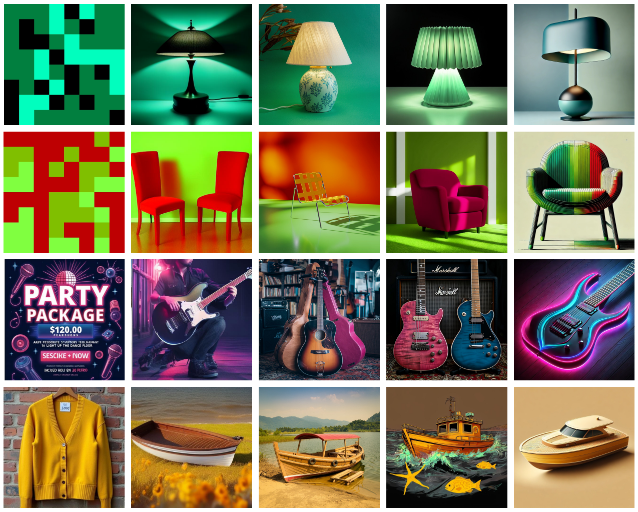
generation of never-seen colors. As further validated quantitatively in Tab. 2, stopping at $t < 200$ (or optionally $t < 400$) balances all the color-conditioned image synthesis criteria as indicated by performance metrics.

	FID ↓	CLIPScore ↑	CD-A ↓	CD-C ↓
No blurring	73.0	29.2	8.67	3.32
Strength = 3 (Our current setting)	70.5	29.4	4.63	5.12
Strength = 6	75.6	29.5	3.60	6.41
Strength = 18	78.9	29.3	3.36	8.67
Strength = 54	82.2	28.9	3.15	14.2
Strength = 128	79.9	29.0	3.88	13.6
Strength = 256	86.5	28.2	4.70	33.9

Table 1. **Quantitative ablation study of blurring color condition before latent encoding**. Note that all runs start with the same random seed for fair comparisons.

	FID ↓	CLIPScore ↑	CD-A ↓	CD-C ↓
No stopping	76.8	29.3	3.32	4.58
Stop at $t < 100$	70.6	29.3	3.77	5.78
Stop at $t < 200$ (Our current setting)	70.5	29.4	4.63	5.12
Stop at $t < 400$	70.2	29.8	6.99	4.46
Stop at $t < 600$	71.4	30.4	10.8	4.07
Stop at $t < 800$	73.7	30.7	17.0	4.02
Stop at $t < 1000$	78.4	30.6	26.8	4.42

Table 2. **Quantitative ablation study of late-time stopping of our latent color alignment**. Note that all runs start with the same random seed (also used in Tab. 1) for fair comparisons.



(a) Input Condition (b) Ours (Fine-tune) (c) Ideogram-2.0 [10] (d) Playground-v3 [6] (e) GPT-4o (VLM) [1]

Figure 2. **Additional qualitative comparisons with non-spatial palette conditioning baselines**. The first column shows the input color condition, and the remaining columns present the results of the experimented methods.

3. More technical visualizations

We provide additional visualizations to illustrate the differences between our color-aligned diffusion method and the regular diffusion method.

Our method only modifies the intermediate pathway for reverse sampling, without affecting the overall objective image distribution across all diffusion steps. Specifically, Eq. 5-7 in the main paper only influence the *model query* (i.e., what the model θ sees and predicts) in the reverse sampling steps. The forward process (denoted by q) with no *model query* involved, including $q(x_t|x_{t-1})$, $q(x_t|x_0)$, $q(x_{t-1}|x_t, x_0)$, and Eq. 4 in the main paper, remain identical to those in regular diffusion. As illustrated in Fig. 5, the original objective of the diffusion process (i.e., learning a mapping from Gaussian to image distribution) is preserved in our setting. Additionally, in such a pathway, our model can adapt to inputs with non-Gaussian noise (distributed as in Eq. 6 in the main paper). We visualize this capability in Fig. 6.

4. Data description details

We provide additional details about the data used in our experiments, supplementing the information in Sec. 4.1 in the main paper.

Recall that to quantitatively assess the color *disentanglement* in generated contents, we randomly generated 50 daily object prompts using ChatGPT [1]. The text prompts are: “chair”, “dog”, “car”, “book”, “table”, “house”, “cat”, “pen”, “shirt”, “bicycle”, “shoe”, “cup”, “bed”, “clock”, “door”, “flower”, “fish”, “camera”, “blanket”, “guitar”, “bag”, “bot-

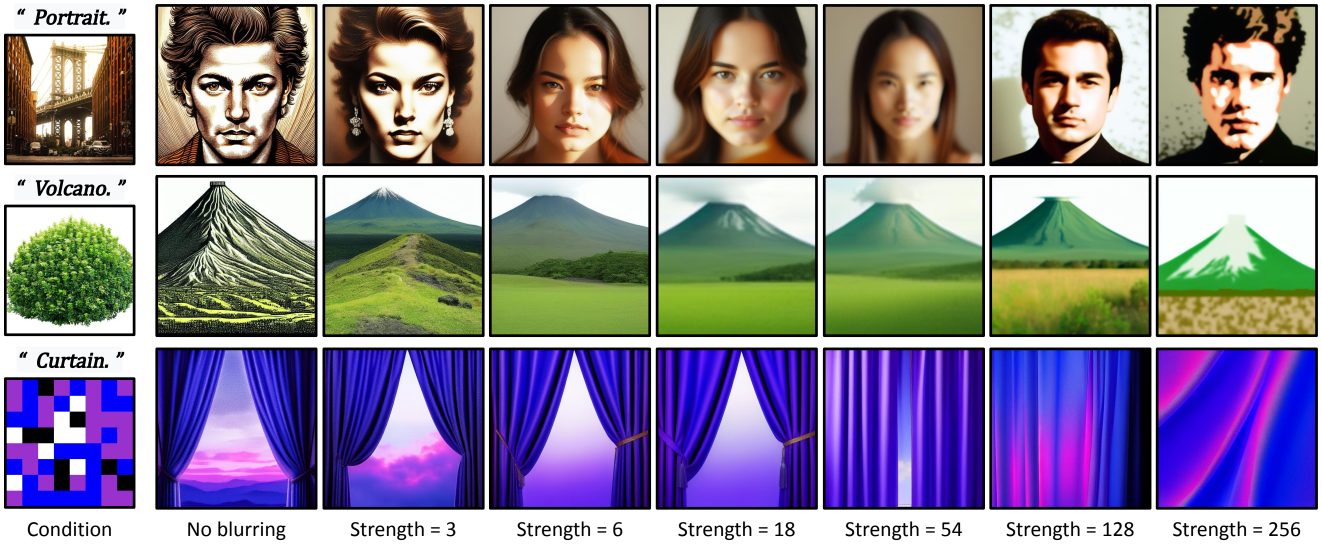


Figure 3. **Qualitative ablation study of blurring color condition before latent encoding**. The first column is the input condition. The rest columns, presenting from left to right, are the results from increasing blurring strength.

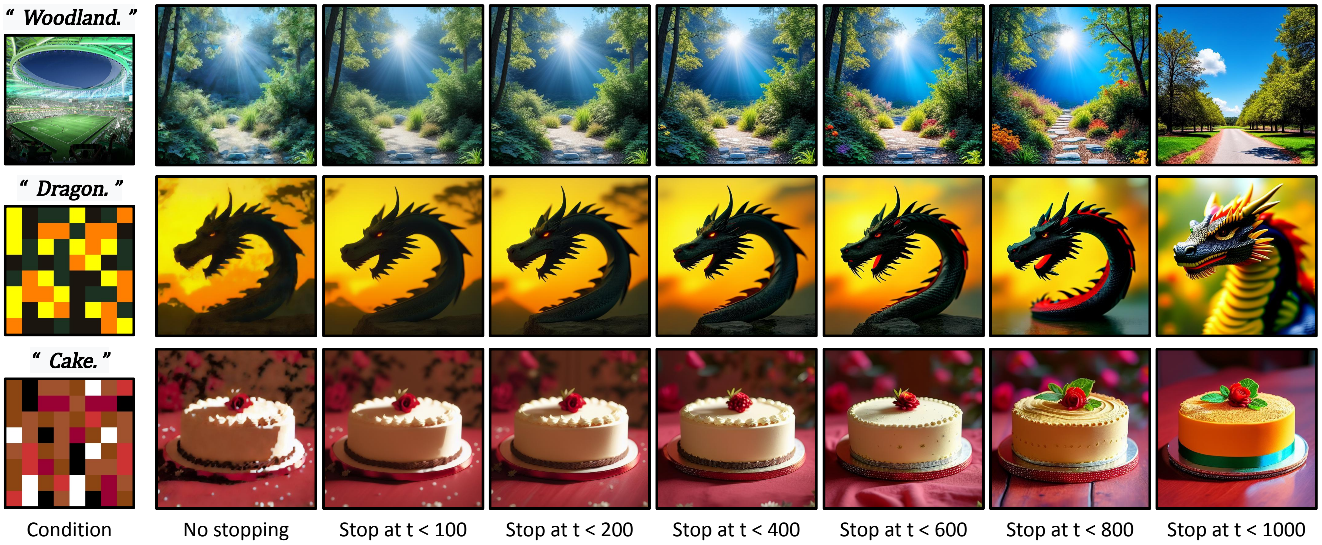


Figure 4. **Qualitative ablation study of late-time stopping of our latent color alignment**. The first column is the input condition. The rest columns, presenting from left to right, are the results from earlier late-time stopping of the alignment.

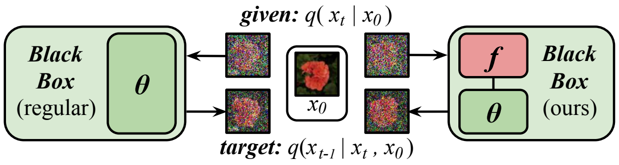


Figure 5. **Pipeline visualization** of our color-aligned diffusion compared to the regular pipeline.

tle", "lamp", "desk", "towel", "suitcase", "basket", "helmet", "skateboard", "umbrella", "soap", "shampoo", "ladder", "painting", "brush", "glove", "hat", "belt", "wallet", "ring", "vase", "statue", "map", "ticket", "kite", "bus", "airplane", "rocket", "boat", and "crystal". During evaluation, we randomly selected the prompts for generation.

To quantitatively evaluate our method under manual color conditions, we simulate user inputs with randomly selected color values and proportions. Each condition image includes 1 to 4 distinct random colors, with random color proportions of 25%, 50%, 75%, or 100%. If only one color is used, we

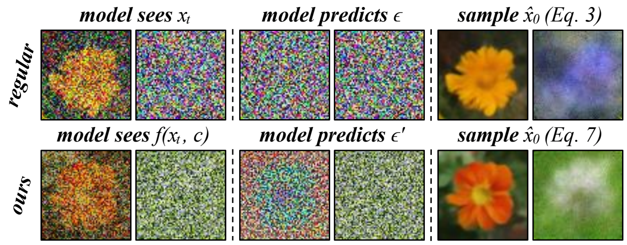

model sees x_t
model predicts ϵ
sample $\hat{x}_0$ (Eq. 3)
regular
model sees $f(x_t, c)$
model predicts ϵ'
sample $\hat{x}_0$ (Eq. 7)
ours

Figure 6. **Input and output visualization** of our color-aligned model compared to the regular model.

replace 25% of pixels of that color with pure black and white to simulate lighting and shadow colors.

5. More implementation details

In this section, we provide additional implementation details of our method and other experimented baselines. This information supplements the descriptions in Secs. 4.2 and 4.3 in the main paper.

We followed the huggingface-diffusers [11] implementation of DDPM [4] and Stable Diffusion [9]. Specifically, we adopted the Stable Diffusion v1.5[1] as the backbone for our image synthesis. This backbone was also used by all the compared baselines. We used 1,000 time steps for training and 50 time steps for inference. We applied classifier-free guidance [3] to all methods with guidance scale 5, using the negative prompt "Low quality, low resolution, blurry, ugly.". We employed Adam Optimizer [5] with learning rate $1e-5$ and betas $(0.95, 0.999)$. For other hyperparameters, we followed the default settings in the Stable Diffusion v1.5.

To implement the color alignment in Eq. 5 in the main paper, we updated $\mathbf{x}_t$ by searching for its most similar colors in $\mathbf{c}$ to achieve $f(\mathbf{x}_t, \mathbf{c})$. Specifically, we applied the GPU-parallelizable PyTorch3D [8] Chamfer-loss[2] on every pixel color $\mathbf{x}_t[p]$ in $\mathbf{x}_t$ to find its most similar color $\mathbf{c}[q]$ in $\mathbf{c}$. This process results in a set of pixel pairs $(\mathbf{x}_t[p], \mathbf{c}[q])$. Then, the color values of $\mathbf{c}[q]$ were assigned to the spatial locations of $\mathbf{x}_t[p]$ to form $f(\mathbf{x}_t, \mathbf{c})$.

We applied the same idea to implement Eq. 8 in the main paper. Specifically, we constructed $g(\hat{\mathbf{x}}_0, \mathbf{c})$ by applying the Chamfer-loss updates multiple times on $\hat{\mathbf{x}}_0$ using $\mathbf{c}$. For each update, the pixels $\hat{\mathbf{x}}_0[p]$ were paired with their most proximate pixels $\mathbf{c}[q]$ (similar to the implementation of f described earlier). However, for all pixel pairs $(\hat{\mathbf{x}}_0[p], \mathbf{c}[q])$, we only selected those satisfied the one-to-one relation, that is, for the pairs where $\mathbf{c}[q]$ was repeatedly used, we only randomly selected one pair to form $g(\hat{\mathbf{x}}_0, \mathbf{c})$. The remaining unused pixels in $\hat{\mathbf{x}}_0$ and $\mathbf{c}$ were deferred to the next round of Chamfer-loss update, until all pixels were eventually paired to form a complete $g(\hat{\mathbf{x}}_0, \mathbf{c})$. This approach approximates the optimal one-to-one mapping at an acceptable cost.

[1] https://huggingface.co/stable-diffusion-v1-5/stable-diffusion-v1-5

[2] https://pytorch3d.readthedocs.io/en/latest/modules/loss.html

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. arXiv preprint arXiv:2303.08774, 2023. 1, 3

[2] Amir Hertz, Andrey Voynov, Shlomi Fruchter, and Daniel Cohen-Or. Style aligned image generation via shared attention. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 4775–4785, 2024. 2

[3] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598, 2022. 5

[4] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. Advances in neural information processing systems, 33:6840–6851, 2020. 5

[5] Diederik P Kingma. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014. 5

[6] Bingchen Liu, Ehsan Akhgari, Alexander Visheratin, Aleks Kamko, Linmiao Xu, Shivam Shrirao, Chase Lambert, Joao Souza, Suhail Doshi, and Daiqing Li. Playground v3: Improving text-to-image alignment with deep-fusion large language models. arXiv preprint arXiv:2409.10695, 2024. 1, 3

[7] Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, and Ying Shan. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In Proceedings of the AAAI Conference on Artificial Intelligence, pages 4296–4304, 2024. 2

[8] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. arXiv:2007.08501, 2020. 5

[9] 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. 5

[10] Ideogram-2.0 Development Team. Ideogram-2.0 technical report. =https://about.ideogram.ai/2.0. 1, 3

[11] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig

Davaadorj, Dhruv Nair, Sayak Paul, Steven Liu, William Berman, Yiyi Xu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. 5

[12] 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

[13] 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. 2