Supplementary — Enhancing Diffusion Models with Text-Encoder Reinforcement Learning

Chaofeng Chen[1⋆], Annan Wang[1⋆], Haoning Wu[1], Liang Liao[1], Wenxiu Sun[3], Qiong Yan[3], and Weisi Lin[2]

[1] S-Lab, Nanyang Technological University
[2] CCDS, Nanyang Technological University [3] Sensetime Research
chaofeng.chen@ntu.edu.sg c190190@e.ntu.edu.sg wslin@ntu.edu.sg
https://github.com/chaofengc/TexForce

1 Results and Test Codes

For the convenience of the reviewers, we provide the test codes and results of our approach on the ImageReward test dataset in the following files:

```
Supplementary Material
├── results_imagereward
│   ├── results_imagereward_sd14 ... # Results with SDv1.4 backbone
│   ├── results_imagereward_sd15 ... # Results with SDv1.5 backbone
│   ├── results_imagereward_sd21 ... # Results with SDv2.1 backbone
│   └── imagereward_benchmark.json ... # Test prompts from ImageReward
├── gpt4v_evaluation_results ... # GPT4V api response
├── lora_weights     ... # LoRA weights for TexForce and ReFL
├── | gpt4v_compare.py ... # Code for GPT4V test
├── | demo.html         ... # Html demo to show all results, open
it with chrome
├── | test.ipynb       ... # IPython notebook to test our model, including
instructions
└── | supplementary.pdf ... # This file
```

2 Method Details

2.1 Reinforcement Learning Formulations

We first give the formulations of each term for the PPO algorithm in diffusion models. According to the DDPM paper [4], the t step denoising can be formulated as:

$$\mathbf{x}_{t-1} = \mu_{\theta,\phi}(\mathbf{x}_t, t, \tau_\phi(s)) + \sigma_t \epsilon, \tag{1}$$

$$\mu_{\theta,\phi} = \frac{1}{\sqrt{\alpha_t}} \left(\mathbf{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_{\theta,\phi}(\mathbf{x}_t, t, \tau_\phi(s)) \right) \tag{2}$$

⋆ These authors contributed equally to this work.

where $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$, $\sigma_t^2 = \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t$, $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and β_t is a predefined variance schedule. Since the U-Net parameter ϕ is fixed during the training, we omit it in the following formulations. The objective function of the PPO algorithm is:

$$J(\phi) = \mathbb{E}\left[\min(r_t(\phi)A, \text{clip}(r_t(\phi), 1-\lambda, 1+\lambda)A)\right], \tag{3}$$

where

$$r_t(\phi) = \frac{p_\phi(\mathbf{x}_{t-1}|\mathbf{x}_t)}{p_{\phi_{\text{old}}}(\mathbf{x}_{t-1}|\mathbf{x}_t)}. \tag{4}$$

Since p_ϕ is an isotropic Gaussian distribution, we have:

$$\begin{aligned} r_t(\phi) &= \exp(\log p_\phi(\mathbf{x}_{t-1}|\mathbf{x}_t) - \log p_{\phi_{\text{old}}}(\mathbf{x}_{t-1}|\mathbf{x}_t)) \\ &= \exp\left(-\frac{1}{2\sigma_t^2}\|\mathbf{x}_{t-1} - \mu_{\phi,t}\|^2 + \frac{1}{2\sigma_t^2}\|\mathbf{x}_{t-1} - \mu_{\phi_{\text{old}},t}\|^2\right). \end{aligned} \tag{5}$$

The reward value A is obtained with reward function $A = R(\mathbf{x}_0, s)$. With equations above, we can get $J(\phi)$.

2.2 Face and Hand Rewards

Figure 1 shows the pipeline of face reward and hand reward. For the face quality reward, we finetuned the TOPIQ model[3] [1] with the GFIQA-20k dataset [6] which is specifically designed for face quality. Since the faces of GFIQA are all aligned, we also need to align the generated face before calculating the reward scores. For hand reward function, there is no existing quality model for hands. We found that simple hand detection confidence score can already give reliable reward for the generation quality of hands. Therefore, we directly use the hand detection confidence as rewards. Thanks to the flexibility of RL, the reward function is not required to be differentiable. We use the pretrained YOLOv5[4] for hand detection.

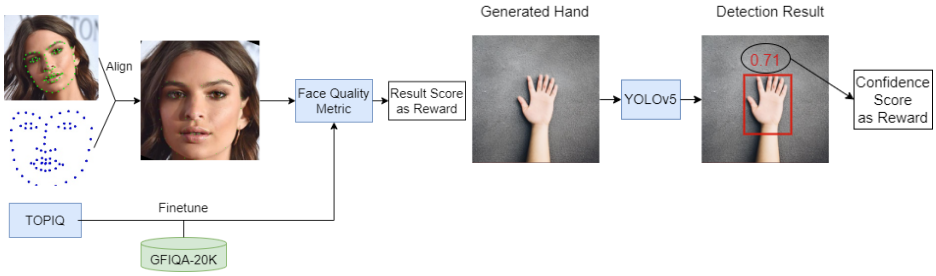
3 Experiment Settings

3.1 Training Details

As shown in Tab. 1, we use the same hyper-parameters for all the experiments. The only difference is the batch size and the number of samples per epoch. For the single prompt dataset, we use a batch size of 8 and 256 samples per epoch. For the multi-prompts dataset, we use a batch size of 64 and 2048 samples per epoch.

[3] https://github.com/chaofengc/IQA-PyTorch

[4] https://github.com/MahmudulAlam/Unified-Gesture-and-Fingertip-Detection



(a) Face quality reward. **(b)** Hand detection confidence reward.

Fig. 1: Calculation of face quality and hand detection reward.

Table 1: Hyper-parameters and training settings for single prompt and multi-prompts datasets.

	Hyper-parameters	Single prompt	Multiple prompts
Diffusion	Sampler	DDIM [5]	
	Guidance Scale	7.5	
	Sampling Steps	50	
Optimizer	Type	AdamW	
	Learning rate	3e-4	
	Weight decay	1e-4	
	(β_1, β_2)	(0.9, 0.999)	
	Gradient clip	1.0	
RL Config	Ratio clip (γ)	1e-4	
	Advantage clip	10	
LoRA	Rank	16	
	Alpha (α)	1	
	Module	q, k, v, out	
Training	Trainable #Params.	1.18M	
	Numerical precision	torch.float16	
	Batch size	8	64
	Samples per epoch	256	2048
	Total epochs	100	100
	GPUs	1 V100	4 A100
	Time	∼2 days	∼3 days

3.2 Evaluation Details

We use the `seed_everything()` function provided by PyTorch Lightning[5] to set the random seed for all the experiments. **We set the start random seed to 234 for ALL prompts.**

[5] https://lightning.ai/

For the single prompt dataset, we generate 50 examples for each prompt. For the multi-prompts dataset, ImageReward and HPSv2, we generate only one example for each prompt to save time.

4 GPT4V Evaluation

We use the `gpt-4-1106-vision-preview` API[6] provided by OpenAI to evaluate the quality of the generated images. The API takes a text prompt and a list of images as input, and returns a rank of the image names based on their aesthetic quality and their coherence with the prompt. Below is an example we used for the evaluation:

GPT4V Evaluation Example (gpt-4-1106-vision-preview)

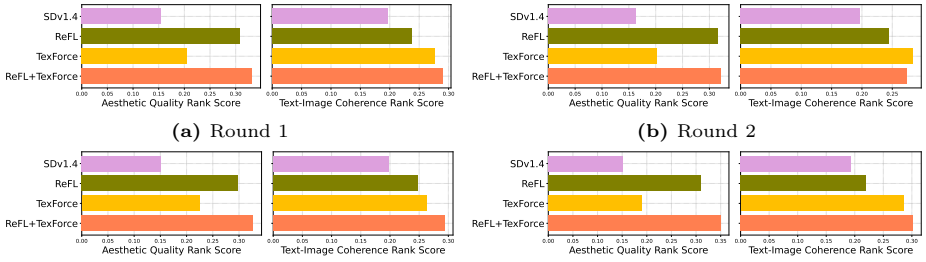
Prompt (#User):
The following four images, Image 1, Image 2, Image 3 and Image 4, are generated with the prompt [*alien landscape with futuristic portal to another alien planet, astronaut stepping through the portal*], please rank them based on their aesthetic quality and their coherence with the given prompt. Answer with only two lists with the image file names, from good to bad. Please only give two lists.

Image 1(SDv1.4) Image 2(ReFL) Image 3(TexForce) Image 4(ReFL+TexForce)

GPT4V Output:
Aesthetic quality, from good to bad:
- Image 2 # ReFL
- Image 4 # ReFL+TexForce
- Image 3 # TexForce
- Image 1 # SDv1.4

Coherence with prompt, from good to bad:
- Image 4 # ReFL+TexForce
- Image 3 # TexForce
- Image 2 # ReFL
- Image 1 # SDv1.4

[6] https://platform.openai.com/docs/guides/vision



(a) Round 1 **(b)** Round 2

(c) Round 3 **(d)** Average

Fig. 2: Complete results of GPT4V evaluation with SDv1.4 backbone on the ImageReward dataset.

As shown in the example above, GPT4V returns two lists of the image names, ranking from good to bad. We assign score 3 for the best image, 0 for the worst image and the final score is normalized to $[0, 1]$. For reliability, we run the evaluation for 3 times and report the average score. The results for each round and the final test score is reported in Fig. 2. Please refer to the demo.html file for the detailed results of each individual test image.

5 Additional Experiments

5.1 More Results of Incompression Task

In the main paper, we briefly discussed different behaviors of finetuing U-Net and text encoder in the incompression task. Here, we provide more quantitative and qualitative results about the comparison between finetuning U-Net and text encoder, with the unseen animal prompts below:

Animal prompts. Following previous works, we use the 45 simple animals for training. The prompt is defined as *A photo of a <animal>*. We use the following animals for testing: *cheetach, elephant, girraffe, hippo, jellyfish, panda, penguin, swan.*

We generate 10 samples for each prompt and report their incompression scores in Tab. 2. We can have the following observations:

- Although the training rewards of text encoder and U-Net are similar, the text encoder is more robust in unseen animals than U-Net, and obtained higher incompression scores.
- The quantitative results also confirm that simply combining U-Net and text encoder is quite effective in improving the reward scores.

Figure 3 shows visual examples of combining U-Net and text encoder at different checkpoints. We can observe that the text encoder can introduce extra reasonable visual concepts to the image to increase complexity, however, U-Net mainly changes the appearance and is easy to disrupt the original structure such as the hippo head.

Table 2: Quantitative results of combining U-Net and text encoder in the incompression task. The comparison checkpoints are the epochs for U-Net and text encoder to achieve the same evaluation reward.

Comparison checkpoints	0	10, 70	20, 100	30, 120
Finetune U-Net	84.01	107.16	143.02	174.15
Finetune text encoder		109.48	156.10	202.79
Fusion		126.59	207.80	250.18

Fig. 3: Visual examples of combining U-Net and text encoder at different checkpoints in the incompression task.

5.2 Results of Aesthetic Reward with Animal Prompts

Aesthetic rewards. Same as previous works, we also conduct experiments using the LAION Aesthetics Predictor [2] as the aesthetic reward function. It should be noted that the aesthetic reward does not consider the coherence with the prompt, and it is easy for the model to hack the reward as shown in [3].

We compare results with DDPO and AlignProb in Fig. 4. Both DDPO and our approach are trained with 10K samples, while AlignProb used early stop to avoid model collapse. Figure 4 presents results for three distinct animals: *jellyfish*, *penguin*, and *swan*. We can observe that both DDPO and AlignProb are over-optimized to the rewards, resulting in over stylized images to maximize

SDv1.4 DDPO AlignProb **TexForce**

Fig. 4: Comparison with others on unseen animal prompts: *jellyfish*, *penguin* and *swan*.

Table 3: Quantitative results of ReFL-LoRA on the ImageReward dataset.

Backbone	Original	ReFL	ReFL-LoRA	TexForce	ReFL+TexForce	ReFL-LoRA+TexForce
SDv1.4	0.2154	0.4485	0.4425	0.4556	**0.6553**	**0.7093**
SDv1.5	0.2140	0.5484	0.5558	0.4086	**0.6703**	**0.7438**
SDv2.1	0.3891	0.5223	0.5181	0.5084	**0.6158**	**0.6263**

aesthetic scores. For instance, DDPO tends to produce images characterized by a blurred yellow background, and AlignProb tends to generate images with oversaturated colors and unrealistic textures. In contrast, our method can generate more lifelike images that closely align with the provided prompts. For example, the composition of the jellyfish image is aesthetically pleasing, and the unrealistic features of the penguin and swan images have been rectified.

5.3 Training ReFL with LoRA

To make our model easier to use, we modified the original ReFL to train LoRA weights instead of the entire U-Net, and the results are shown in Tab. 3. We can notice that the performance of ReFL-LoRA is similar to ReFL but much better when combining with TexForce.

6 More Qualitative Results

In this section, we select more examples on different backbones in Figs. 5 to 10. Please refer to the demo.html for all results.

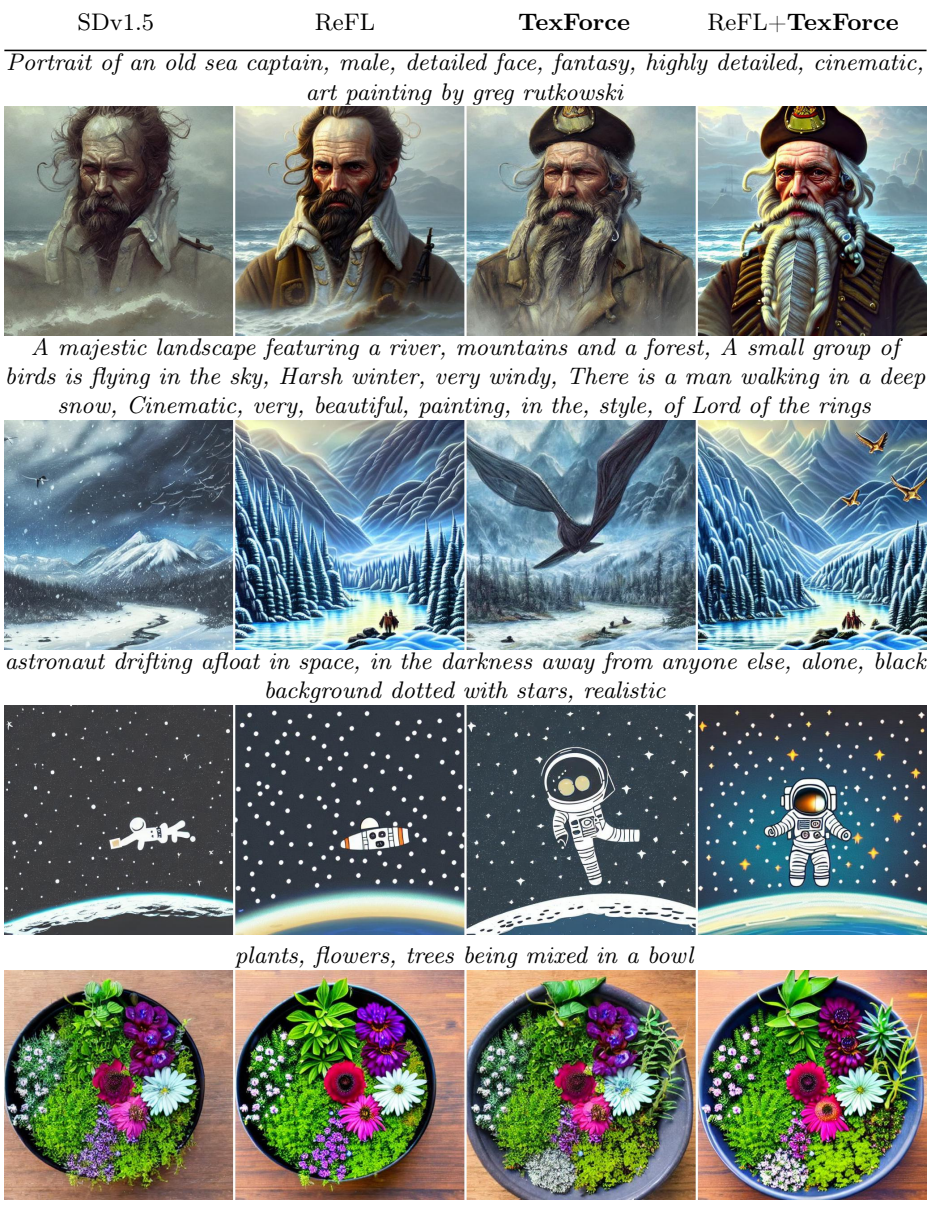


Fig. 5: Results of different methods with Stable Diffusion V1.4 as backbone on ImageReward test dataset.

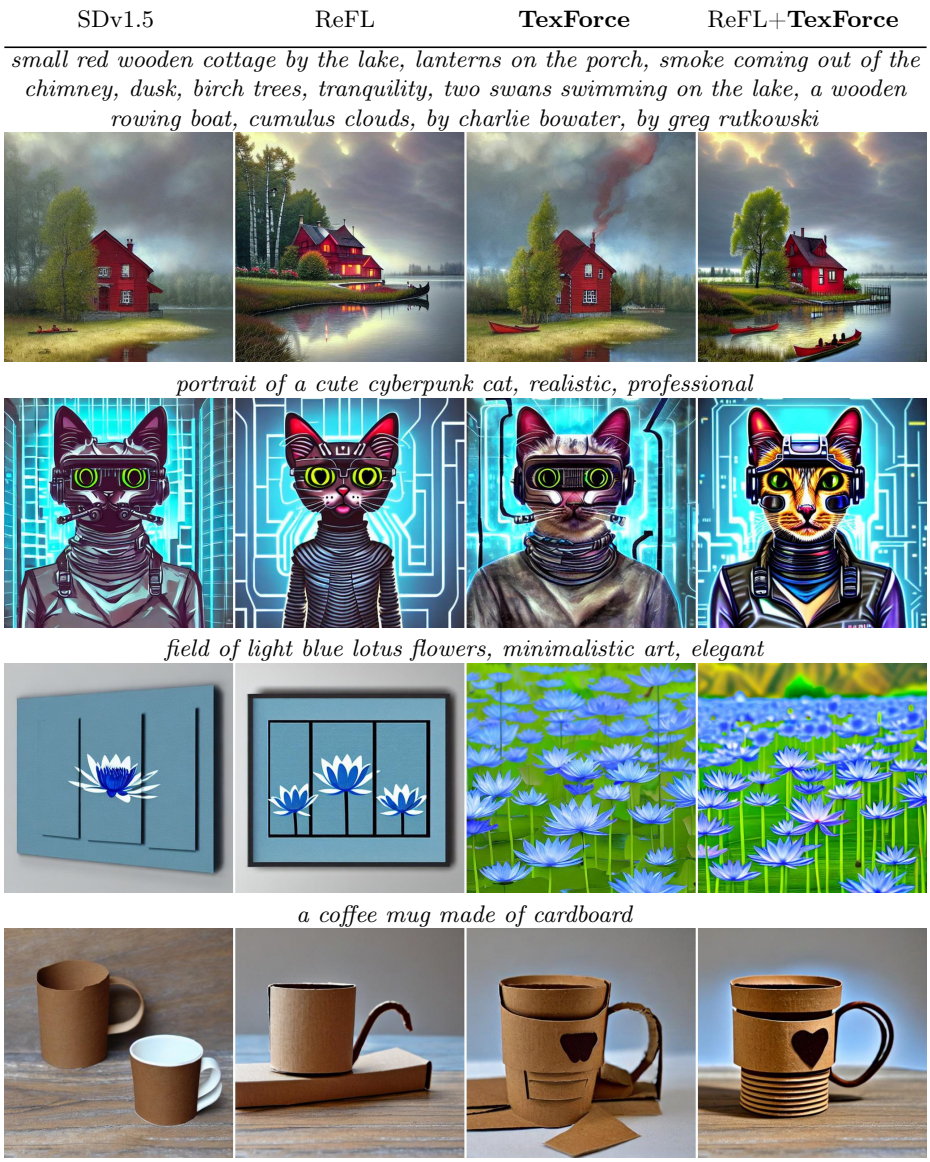


Fig. 6: Results of different methods with Stable Diffusion V1.5 as backbone on ImageReward test dataset.

7 Limitations

Similar to other RL-based methods, our method also faces the challenges of sample efficiency and the complexity of reward function engineering.

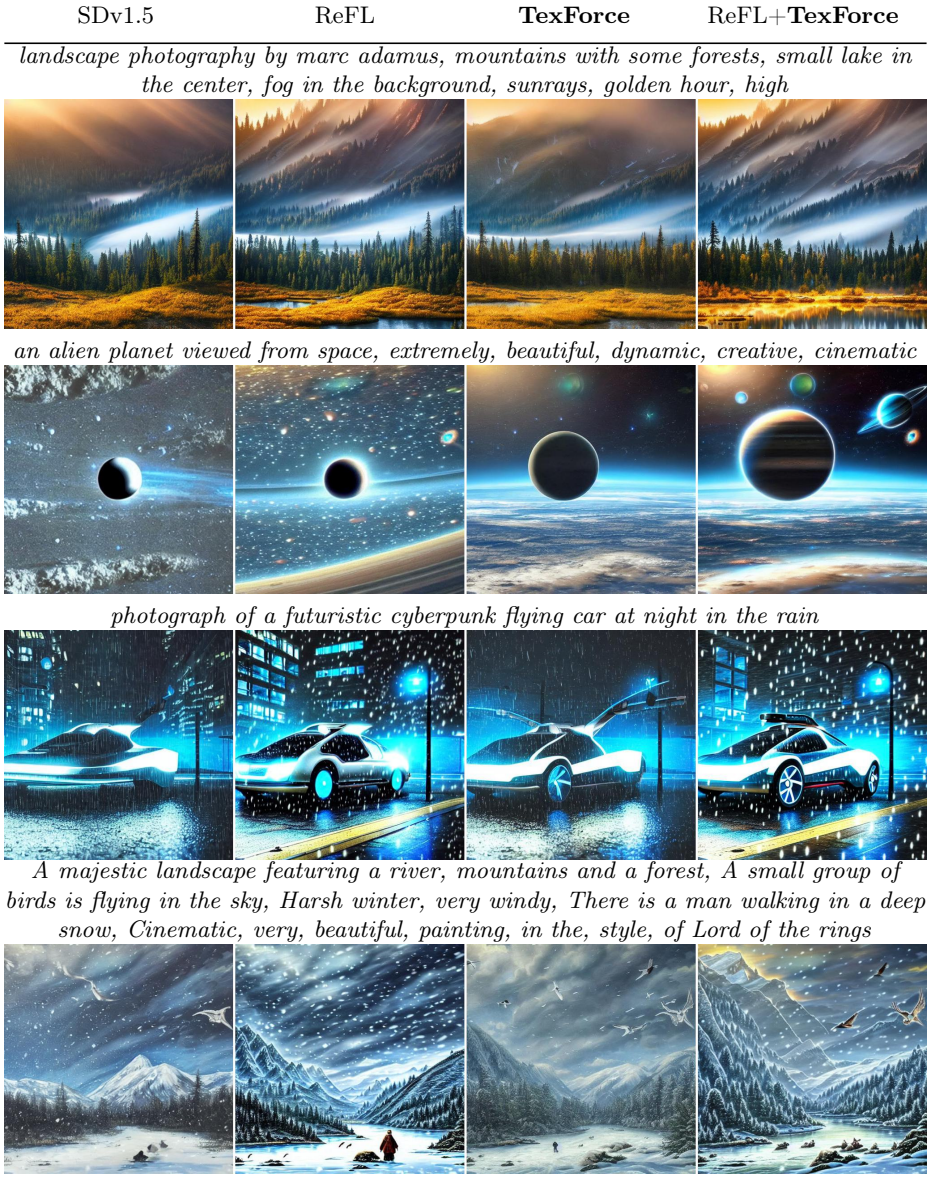


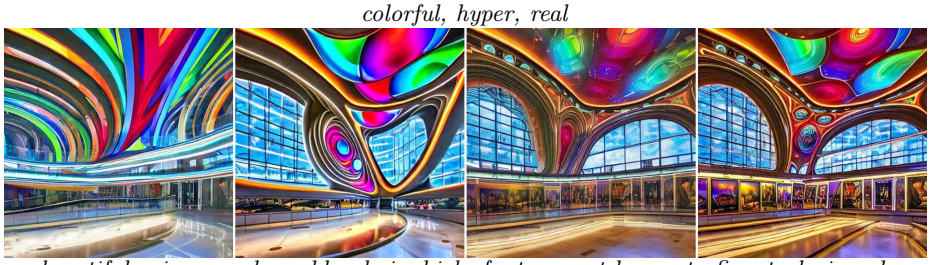
Fig. 7: Results of different methods with Stable Diffusion V1.5 as backbone on ImageReward test dataset.

8 Broader Impacts

Since TexForce can finetune text-to-image models to satisfy specific rewards, it presents potential societal concerns regarding misinformation, intellectual property rights, and illegal usage of the model.

SDv1.5 ReFL **TexForce** ReFL+**TexForce**

extremely detailed stunning beautiful futuristic smooth curvilinear museum interior, colorful, hyper, real



beautiful prince, male, golden hair, high, fantasy, art by anato finnstark, joseph leyendecker, peter mohrbacher, ruan jia, marc simonetti, ayami kojima, cedric peyravernay, finnian macmanus, alphonse mucha, victo ngai

small red wooden cottage by the lake, lanterns on the porch, smoke coming out of the chimney, dusk, birch trees, tranquility, two swans swimming on the lake, a wooden rowing boat, cumulus clouds, by charlie bowater, by greg rutkowski

natural suburb with multiple low rise apartment buildings in a park like setting with green hilly lawns and lush trees, pine wooden walls, rustic, large glass windows, cobblestone, grass, white, natural pathways, natural materials, minimalist, swedish design, bright, feng shui, modern, technology, frank lloyd wright

Fig. 8: Results of different methods with Stable Diffusion V1.5 as backbone on ImageReward test dataset.

References

1. Chen, C., Mo, J., Hou, J., Wu, H., Liao, L., Sun, W., Yan, Q., Lin, W.: Topiq: A top-down approach from semantics to distortions for image quality assessment (2023) 2

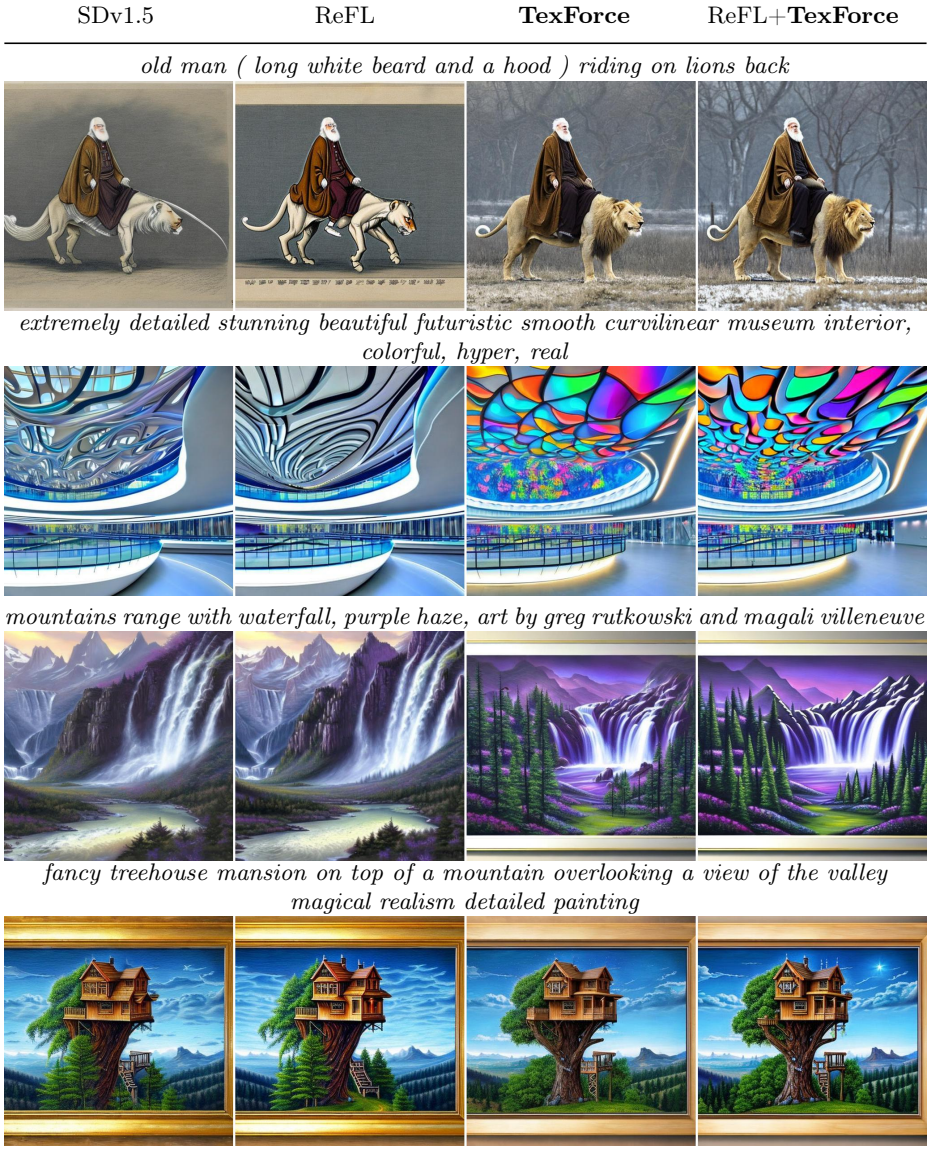


Fig. 9: Results of different methods with Stable Diffusion V2.1 as backbone on ImageReward test dataset.

2. Christoph, S., Romain, B.: Laion-aesthetics. (2022) 6
3. Dong, H., Xiong, W., Goyal, D., Pan, R., Diao, S., Zhang, J., Shum, K., Zhang, T.: Raft: Reward ranked finetuning for generative foundation model alignment. arXiv preprint arXiv:2304.06767 (2023) 6
4. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: Advances in Neural Information Processing Systems (2020) 1
5. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020) 3

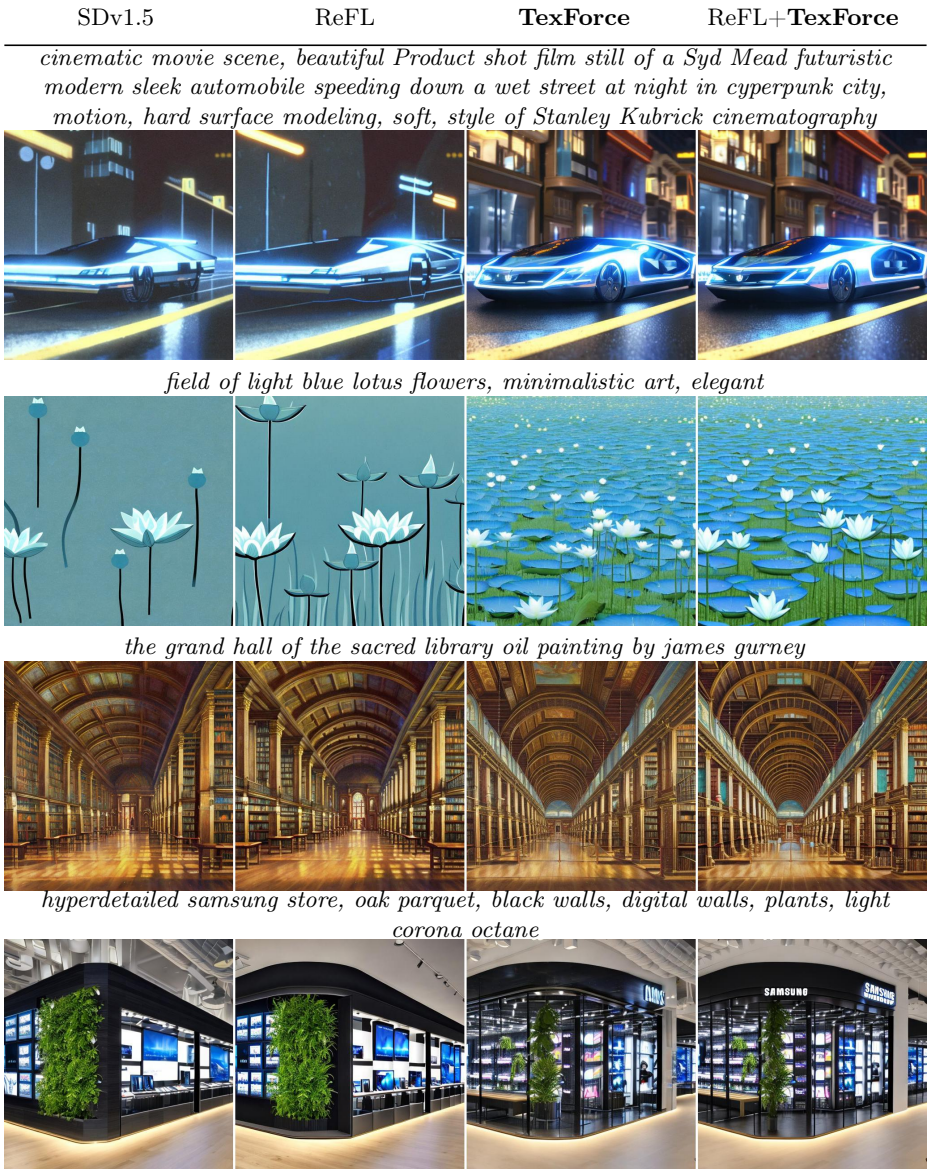


Fig. 10: Results of different methods with Stable Diffusion V2.1 as backbone on ImageReward test dataset.

6. Su, S., Lin, H., Hosu, V., Wiedemann, O., Sun, J., Zhu, Y., Liu, H., Zhang, Y., Saupe, D.: Going the extra mile in face image quality assessment: A novel database and model. IEEE Transactions on Multimedia (2023) 2