

Dream-in-Style: Text-to-3D Generation using Stylized Score Distillation

Supplementary Material

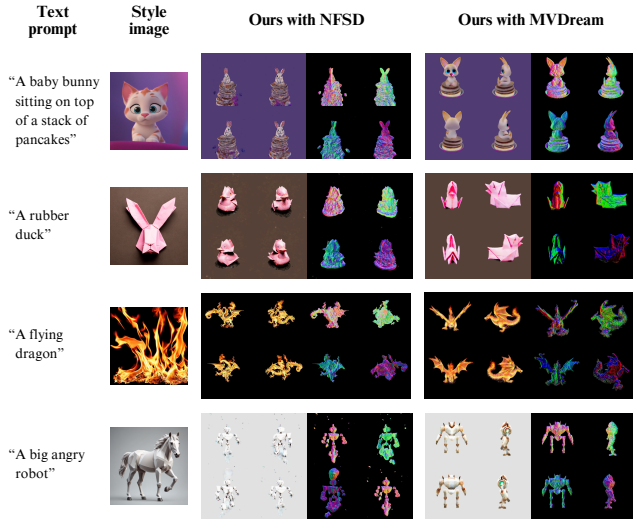


Figure 1. Example cases where multi-face Janus problem occurred and was mitigated by using MVDream.

Abstract

In this supplementary document, we (1) further discuss our limitations and potential mitigations of the Janus problem using MVDream. We then (2) provide more evaluations of our method and compare it with image-to-3D methods as well as a second style transfer method. We also (3) provide a detailed derivation of our stylized score distillation, and (4) more details of our GPT-based user study. For more qualitative results, please see our supplementary video.

1. Further discussion on limitations

1.1. Multi-face Janus problem

Our method inherits the Janus problem [3] known to score distillation sampling because we use Stable Diffusion, a single-view image diffusion model as the 3D generation prior. To mitigate this problem, one can perform score distillation with multi-view diffusion models. Here, we demonstrate that our stylized score distillation (SSD) extends naturally to MVDream, a popular multi-view diffusion model [5].

In Figure 1 we present cases where multi-face Janus problem is mitigated when our method is combined with MVDream. Multi-view videos of these examples can be found in our supplementary video.

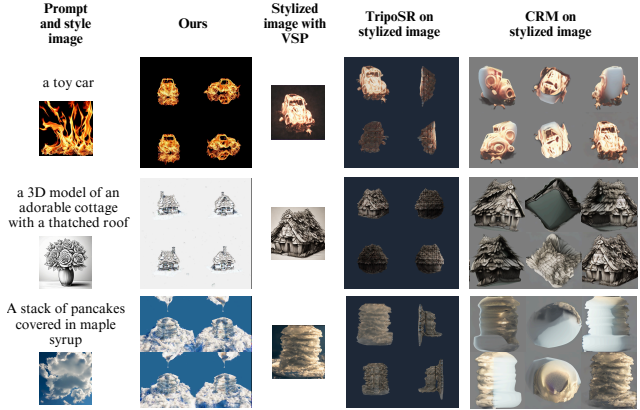


Figure 2. Comparison of our method with results from image-to-3D on stylized images.

1.2. Discussion on hard cases

We found style images with ambiguous content or complex backgrounds particularly challenging, e.g., fire style from Figure 1 or cloud style from Figure 2. We found that our style ratio scheduling is particularly important to address these cases during optimization. Without proper scheduling, these cases produced poor results with little or no information about the object in the final results.

2. More evaluations

2.1. Image-to-3D on stylized images

We experimented with lifting stylized images to 3D using CRM and TripoSR, two popular large reconstruction models for image-to-3D reconstruction. CRM also uses ImageDream [6], an image-conditioned multi-view diffusion model as an initialization to train their reconstruction model. We found that there are two limitations to this approach. First, when the stylized images have ambiguous 2D geometry (fire around a car), the 3D results by CRM and TripoSR are worse than ours. Second, CRM and TripoSR are trained with clean object rendering, which does not generalize well to stylized images with complex visual effects. Comparison of these cases can be found in Figure 2.

2.2. Visual Style Prompting vs. StyleAligned

We additionally tested our score distillation on a different training-free style transfer method using diffusion model. Particularly, we adopt StyleAligned [1] instead of visual style prompting [2]. We found that our stylized score distillation works well with StyleAligned, and we did not find

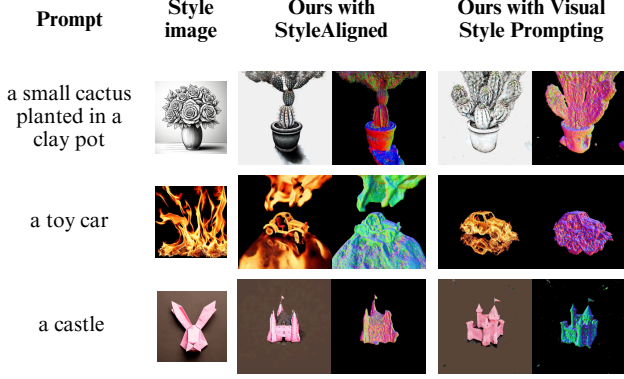


Figure 3. Visual style prompting vs. StyleAligned.

significant differences in the final results. The biggest difference occurs for harder cases mentioned in Section 1.2 where StyleAligned produced results with more focus on surroundings of generated object. The comparison can be found in Figure 3.

3. Derivation

Here we provide a detailed derivation of our stylized score distillation. Recall that our stylized score distillation aims to minimize $KL(q(\mathbf{z}_t | \mathbf{x} = g(\theta)) \parallel p_\phi(\mathbf{z}_t | y, \mathbf{s}))$, which is equivalent to

$$\min_{\theta} \mathbb{E}_{\varepsilon} [\log(q(\mathbf{z}_t | \mathbf{x} = g(\theta))) - (1 - \lambda) \log(p_\phi(\mathbf{z}_t | y)) - \lambda \log(\hat{p}_\phi(\mathbf{z}_t | y, \mathbf{s}))]. \quad (1)$$

Taking the derivative w.r.t. θ yields a gradient with three terms:

$$\begin{aligned} \nabla_{\theta} KL(q(\mathbf{z}_t | \mathbf{x} = g(\theta)) \parallel p_\phi(\mathbf{z}_t | y, \mathbf{s})) \\ = \mathbb{E}_{\varepsilon} \left[\underbrace{\nabla_{\theta} \log(q(\mathbf{z}_t | \mathbf{x} = g(\theta)))}_{(A)} - (1 - \lambda) \underbrace{\nabla_{\theta} \log(p_\phi(\mathbf{z}_t | y))}_{(B)} - \lambda \underbrace{\nabla_{\theta} \log(\hat{p}_\phi(\mathbf{z}_t | y, \mathbf{s}))}_{(C)} \right]. \end{aligned} \quad (2)$$

Among these, term (B) and term (C) can be expanded by

$$\nabla_{\theta} \log(p_\phi(\mathbf{z}_t | y)) = -\frac{\alpha_t}{\sigma_t} \varepsilon_\phi(\mathbf{z}_t | y) \frac{\partial \mathbf{x}}{\partial \theta}, \quad (3)$$

and

$$\nabla_{\theta} \log(\hat{p}_\phi(\mathbf{z}_t | y, \mathbf{s})) = -\frac{\alpha_t}{\sigma_t} \hat{\varepsilon}_\phi(\mathbf{z}_t | y, \mathbf{s}) \frac{\partial \mathbf{x}}{\partial \theta}. \quad (4)$$

Term (A) can be expanded to

$$\begin{aligned} \nabla_{\theta} \log q(\mathbf{z}_t | \mathbf{x}) &= \left(\underbrace{\frac{\partial \log q(\mathbf{z}_t | \mathbf{x})}{\partial x}}_{\text{parameter score}} + \underbrace{\frac{\partial \log q(\mathbf{z}_t | \mathbf{x})}{\partial \mathbf{z}_t} \frac{\partial \mathbf{z}_t}{\partial \mathbf{x}}}_{\text{path derivative}} \right) \alpha_t \frac{\partial \mathbf{x}}{\partial \theta} \\ &= \left(\frac{\alpha_t}{\sigma_t} \epsilon - \frac{\alpha_t}{\sigma_t} \epsilon \right) \alpha_t \frac{\partial \mathbf{x}}{\partial \theta} = 0. \end{aligned} \quad (5)$$

Similarly to DreamFusion [3], based on Sticking-the-Landing [4], we can discard the parameter score term and keep only the path derivative term.

The final gradient becomes

$$\begin{aligned} \nabla_{\theta} \mathcal{L}_{SSD} \\ = \mathbb{E}_{t, \mathbf{z}_t | \mathbf{x}} \left[\omega(t) \frac{\sigma_t}{\alpha_t} \nabla_{\theta} KL(q(\mathbf{z}_t | \mathbf{x} = g(\theta)) \parallel p_\phi(\mathbf{z}_t | y, \mathbf{s})) \right] \\ = \mathbb{E}_{t, \varepsilon} \left[\omega(t) ((1 - \lambda) \varepsilon_\phi(\mathbf{z}_t | y) + \lambda \hat{\varepsilon}_\phi(\mathbf{z}_t | y, \mathbf{s}) - \varepsilon) \frac{\partial \mathbf{x}}{\partial \theta} \right]. \end{aligned} \quad (6)$$

4. GPT-4 Evaluation Template

We extend the GPTEval3D toolbox [7] to evaluate the style alignment between the generated 3D objects and the style reference image. The toolbox prompts the GPT-4 language model from OpenAI to perform the evaluation. The entire text prompt to GPT-4 is listed below. Instruction #6 is for style alignment evaluation. Figure 4 shows an example of the image grid sent to GPT-4 for evaluation. In Section 4, we present the detailed Elo scores obtained from this evaluation.

Our task here is the compare two 3D objects , both generated from the same text description .
We want to decide which one is better according to the provided criteria .

I will provide you with a specific multi-view images of two 3D objects , where the left part of it are image renderings and normal renderings of 3D object 1, and the right part denotes those of 3D object 2.

At the bottom of the image , last row , you can see the style image duplicated four times . This image is the reference image for the style of the 3D object .

Instruction

1. Text prompt and Asset Alignment . Focus on how well they correspond to the given text description . An ideal model should accurately reflect all objects and surroundings mentioned in the text prompt , capturing the corresponding attributes as described . Please first describe each of the two models , and then evaluate how well it covers all the attributes in the original text prompt .
2. 3D Plausibility . Look at both the RGB and normal images and imagine a 3D model from the multi-view images . Determine which model appears more natural , solid , and plausible . Pay attention to any irregularities , such as abnormal body proportions , duplicated parts , or the presence of noisy or meaningless 3D structures . An ideal model should possess accurate proportions , shapes , and structures that closely resemble the real-world object or scene .
3. Geometry-Texture Alignment . This examines how well the texture adheres to the geometry . The texture and shape should align with each other locally . For instance , a flower should resemble a flower in both the RGB and normal map , rather than solely in the RGB . The RGB image and its corresponding normal image should exhibit matching structures .
4. Low-Level Texture Details . Focus on local parts of the RGB images . Assess which model effectively captures fine details without appearing blurry and which one aligns with the desired aesthetic of the 3D model . Note that overly abstract and stylized textures are not desired unless specifically mentioned in the text prompt .
5. Low-Level Geometry Details . Focus on the local parts of the normal maps . The geometry should accurately represent the intended shape . Note that meaningless noise is not considered as high-frequency details . Determine which one has a more well-organized and efficient structure , which one exhibits intricate details , and which one is more visually pleasing and smooth .
6. Style Image Alignment . Look at the style image at the bottom and determine which model better aligns with the desired style . Do you see any patterns from style image that are present in any of 3D objects ? 3D object should ideally represent the provided prompt , but in the style from the style image . It should be a good combination of the prompt and reference style .
7. Considering all the degrees above , which one is better overall ?

Take a really close look at each of the multi-view images for these two 3D objects before providing your answer .

When evaluating these aspects , focus on one of them at a time .

Try to make independent decisions between these criteria .

Output format

To provide an answer , please provide a short analysis for each of the abovementioned evaluation criteria .

The analysis should be very concise and accurate .

For each of the criteria , you need to make a decision using these three options :

1. Left (object 1) is better;
2. Right (object 2) is better;
3. Cannot decide.

IMPORTANT: PLEASE USE THE THIRD OPTION SPARSELY.

Then , in the last row , summarize your final decision by "<option for criterion 1> <option for criterion 2> <option for criterion 3> <option for criterion 4> <option for criterion 5> <option for criterion 6> <option for criterion 7>".

An example output looks like follows :

Analysis :

1. Text prompt & Asset Alignment: The left one xxxx; The right one xxxx; The left/right one is better or cannot decide .

2. 3D Plausibility . The left one xxxx; The right one xxxx; The left/right one is better or cannot decide .

3. Geometry-Texture Alignment . The left one xxxx; The right one xxxx; The left/right one is better or cannot decide .

4. Low-Level Texture Details . The left one xxxx; The right one xxxx; The left/right one is better or cannot decide .

5. Low-Level Geometry Details . The left one xxxx; The right one xxxx; The left/right one is better or cannot decide .

6. Style Image Alignment . The left one xxxx; The right one xxxx; The left/right one is better or cannot decide .

7. Overall , xxxxxx
The left/right one is better or cannot decide .

Final answer :

x x x x x x x (e.g., 1 2 2 3 2 1 1/ 3 3
3 2 1 3 3 / 3 2 2 1 1 1 1)
"

Following is the text prompt from which these two 3D objects are generated :
"<PROMPT>"

Please compare these two 3D objects as instructed .

Methods	Text-Asset Alignment	3D Plausibility	Text-Geometry Alignment	Texture Details	Geometry Details	Style Alignment	Overall
Style-in-prompt	1000.000	1000.000	1000.000	1000.000	1000.000	1000.000	1000.000
Neural style loss	1022.913	1045.297	1063.513	1039.004	1058.329	960.737	1038.849
Textual inversion	1035.892	1037.247	1045.499	1028.978	1039.247	961.984	1025.917
Ours	1118.967	1161.566	1158.723	1150.614	1162.029	1046.029	1140.604

Table 1. Detailed results from GPTEval3D evaluation.

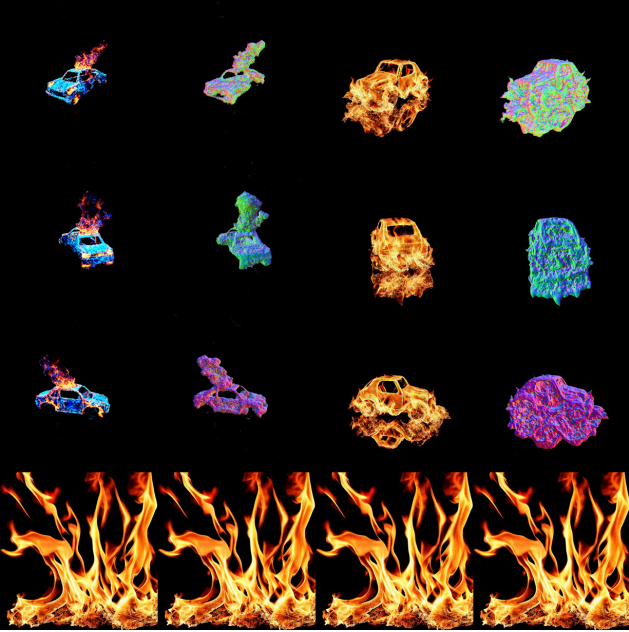


Figure 4. An image grid sent to GPT for evaluation. The first and second column show the color rendering and normal map of the first method, and the third and fourth column are for the second method. Each row shows a camera view of the objects, and the last row shows the style reference. The prompt is "a toy car" and the style image represents fire on a black background. GPT is asked to pick the better result out of the two presented methods.

References

- [1] Amir Hertz, Andrey Voynov, Shlomi Fruchter, and Daniel Cohen-Or. Style aligned image generation via shared attention. In *CVPR*, 2024. [1](#)
- [2] Jaeseok Jeong, Junho Kim, Yunjey Choi, Gayoung Lee, and Youngjung Uh. Visual style prompting with swapping self-attention. *arXiv preprint arXiv:2402.12974*, 2024. [1](#)
- [3] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *The Eleventh International Conference on Learning Representations*, 2023. [1](#), [2](#)
- [4] Geoffrey Roeder, Yuhuai Wu, and David Duvenaud. Sticking the landing: Simple, lower-variance gradient estimators for variational inference, 2017. [2](#)
- [5] Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. In *ICLR*, 2024. [1](#)
- [6] Peng Wang and Yichun Shi. Imagedream: Image-prompt multi-view diffusion for 3d generation. *arXiv preprint arXiv:2312.02201*, 2023. [1](#)
- [7] Tong Wu, Guandao Yang, Zhibing Li, Kai Zhang, Ziwei Liu, Leonidas Guibas, Dahua Lin, and Gordon Wetzstein. Gpt-4v(ision) is a human-aligned evaluator for text-to-3d generation. In *CVPR*, 2024. [3](#)