Debiasing Scores and Prompts of 2D Diffusion for Robust Text-to-3D Generation

Susung Hong* Donghoon Ahn* Seungryong Kim

Korea University, Seoul, Korea

Abstract

The view inconsistency problem in score-distilling text-to-3D generation, also known as the Janus problem, arises from the intrinsic bias of 2D diffusion models, which leads to the unrealistic generation of 3D objects. In this work, we explore score-distilling text-to-3D generation and identify the main causes of the Janus problem. Based on these findings, we propose two approaches to debias the score-distillation frameworks for robust text-to-3D generation. Our first approach, called score debiasing, involves gradually increasing the truncation value for the score estimated by 2D diffusion models throughout the optimization process. Our second approach, called prompt debiasing, identifies conflicting words between user prompts and view prompts utilizing a language model and adjusts the discrepancy between view prompts and object-space camera poses. Our experimental results show that our methods improve realism by significantly reducing artifacts and achieve a good trade-off between faithfulness to the 2D diffusion models and 3D consistency with little overhead.

1. Introduction

Recently, significant advancements have been made in the field of zero-shot text-to-3D generation [8], particularly with the integration of score distillation techniques [10, 11, 16, 27] and diffusion models [3, 5, 9, 18, 19, 23–25] to optimize Neural Radiance Fields (NeRF) [13]. These methods provide a solution for generating a wide range of 3D objects from a textual input, without requiring 3D supervision. Despite their considerable promise, these approaches often encounter a view inconsistency problem. One of the most notable problems is the multi-face issue, also referred to as *the Janus problem*, which is illustrated in the "Baseline" of Fig. 1. This problem constrains the applicability of the methods [10, 11, 16, 27], but the Janus problem is rarely formulated or carefully analyzed in previous literature.

To address the view inconsistency problem, we delve into the formulation of score-distilling text-to-3D generation presented in [10, 11, 16, 27]. We generalize and expand the assumptions about the gradients regarding the 3D parameterization in previous works, such as DreamFusion [16] and Score Jacobian Chaining (SJC) [27], and identify the main causes of the problem within the estimated score. The score further derives the unconditional score and pose-prompt gradient, both of which interrupt the estimation of unbiased gradients concerning the 3D volume. Therefore, refining them is necessary for generating more realistic and view-consistent 3D objects.

Building on this concept and drawing inspiration from gradient clipping [12] and dynamic thresholding [19], we propose a score debiasing method that performs dynamic score thresholding. Specifically, this method cuts off the score estimated by 2D diffusion models to mitigate the impact of erroneous bias. With this debiasing approach, we reduce artifacts in generated 3D objects and alleviate the view inconsistency problem by striking a balance between faithfulness to 2D models and 3D consistency. Furthermore, we

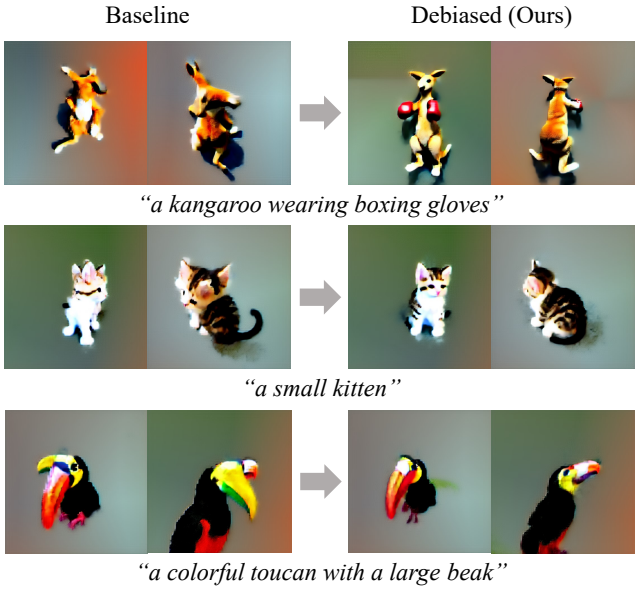

Figure 1. **Comparison between the baseline (SJC [27]) and ours.** Our debiasing methods qualitatively reduce view inconsistencies in zero-shot text-to-3D and the so-called *Janus problem*.

* Equal contribution

find that by gradually increasing the truncation value, which aligns with the coarse-to-fine nature of generating 3D objects [4,13], we achieve a better trade-off for 3D consistency without significantly compromising faithfulness.

On the other hand, as the first attempt to address the bias issue in prompts, we further present a prompt debiasing method. In contrast to prior works [10, 16, 27] that simply concatenate a view prompt and user prompt, our method reduces inherent contradiction between them by leveraging a language model trained with a masked language modeling (MLM) objective [2], computing the pointwise mutual information. Additionally, we decrease the discrepancy between the assignment of the view prompt and object-space pose by adjusting the range of view prompts. These enable text-to-image models [14, 18, 19] to predict accurate 2D scores, resulting in 3D objects that possess more realistic and consistent structures.

2. Score Distillation and the Janus Problem

SJC [27] defines the probability density function of parameters θ of 3D volume (*e.g.*, NeRF [13]) as an expectation of the likelihood of 2D rendered images $\mathbf{z}_\theta$ from uniformly sampled object-space viewpoints. Unlike this definition, our approach defines the density function of the parameters θ as a product of conditional likelihoods given a set of uniformly sampled viewpoints Π and user prompt ω. This can be expressed as:

$$\tilde{p}_{\text{3D}}(\theta) = \prod_{\xi \in \Pi} p_{\text{2D}}(\mathbf{z}_\theta | \xi, \omega), \tag{1}$$

where p_{2D} and $\tilde{p}_{\text{3D}}$ denote the probability density of 2D image distribution and unnormalized density of 3D parametrizations, respectively. By using this formulation, we avoid using Jensen's inequality, in contrast to [27]. Applying the logarithm to each side of the equation yields:

$$\log \tilde{p}_{\text{3D}}(\theta) = \sum_{\xi \in \Pi} \log p_{\text{2D}}(\mathbf{z}_\theta | \xi, \omega). \tag{2}$$

By taking the gradient of $\log \tilde{p}_{\text{3D}}(\theta)$, we can directly obtain $\nabla_\theta \log p_{\text{3D}}(\theta)$, since the normalizing constant is irrelevant to θ. Using the chain rule, we obtain:

$$\begin{aligned} \nabla_\theta \log p_{\text{3D}}(\theta) &= \sum_{\xi \in \Pi} \nabla_\theta \log p_{\text{2D}}(\mathbf{z}_\theta | \xi, \omega), \\ &= Z \cdot \mathbb{E}_{\xi \in \Pi} \left[\nabla_\theta \log p_{\text{2D}}(\mathbf{z}_\theta | \xi, \omega) \right] \\ &= Z \cdot \mathbb{E}_{\xi \in \Pi} \left[\nabla_{\mathbf{z}_\theta} \log p_{\text{2D}}(\mathbf{z}_\theta | \xi, \omega) \frac{\partial \mathbf{z}_\theta}{\partial \theta} \right], \end{aligned} \tag{3}$$

where $Z = |\Pi|$ is a constant, and $\nabla_{\mathbf{z}_\theta} \log p_{\text{2D}}(\mathbf{z}_\theta | \xi, \omega)$ is practically estimated by diffusion models [9]. Note that this definition generalizes SJC [27] and even $\nabla_\theta \mathcal{L}_{\text{SDS}}$ in DreamFusion [16], which can be easily seen as the estimation of

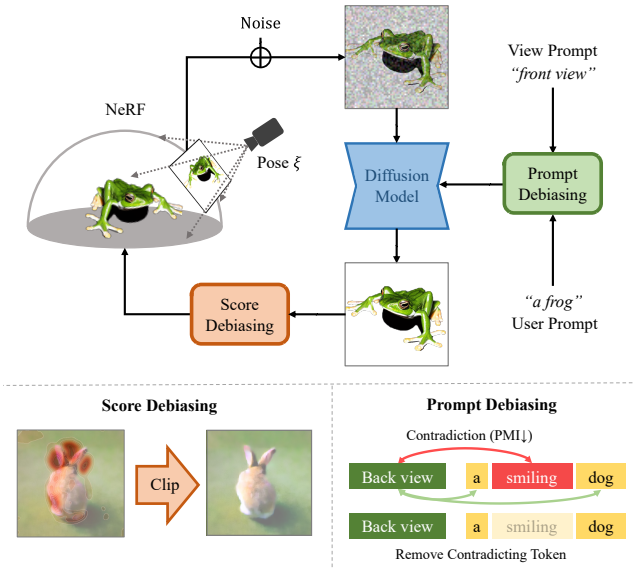

Figure 2. **Illustration of our framework.** We propose prompt and score debiasing techniques to estimate robust and unbiased gradients of the 3D parameters w.r.t. the viewpoints.

Eq. 3 with a different weighting rule and sampler. This is further expanded by applying Bayes' rule as follows:

$$\begin{aligned} &= Z \cdot \mathbb{E}_{\xi \in \Pi} \Big[\big(\underbrace{\nabla_{\mathbf{z}_\theta} \log p_{\text{2D}}(\mathbf{z}_\theta)}_{\text{Unconditional score}} \\ &\qquad + \underbrace{\nabla_{\mathbf{z}_\theta} \log p_{\text{2D}}(\xi, \omega | \mathbf{z}_\theta)}_{\text{Pose-prompt gradient}} \big) \frac{\partial \mathbf{z}_\theta}{\partial \theta} \Big]. \end{aligned} \tag{4}$$

The first gradient term, reflecting the unconditional score modeled by 2D diffusion models [5, 25], contains a bias that affects images viewed closely from specific viewpoints during early 3D optimization when $\mathbf{z}_\theta$ is noisy. This contributes to the Janus problem, as facial views are more prevalent in 2D data distribution for some objects.

On the other hand, the pose-prompt gradient in Eq. 4 is guidance [3,6,7,25] that drives the rendered image to better represent a specific camera pose and user prompt. The term is further expanded:

$$\begin{aligned} \nabla_{\mathbf{z}_\theta} \log p_{\text{2D}}(\xi, \omega | \mathbf{z}_\theta) = &\nabla_{\mathbf{z}_\theta} \log p_{\text{2D}}(\xi | \mathbf{z}_\theta) \\ &+ \nabla_{\mathbf{z}_\theta} \log p_{\text{2D}}(\omega | \mathbf{z}_\theta) \\ &+ \nabla_{\mathbf{z}_\theta} \log C, \end{aligned} \tag{5}$$

where C is defined as $\frac{p_{\text{2D}}(\xi, \omega | \mathbf{z}_\theta)}{p_{\text{2D}}(\xi | \mathbf{z}_\theta) p_{\text{2D}}(\omega | \mathbf{z}_\theta)} = \frac{p_{\text{2D}}(\xi | \omega, \mathbf{z}_\theta)}{p_{\text{2D}}(\xi | \mathbf{z}_\theta)}$, which represents the pointwise conditional mutual information (PCMI). If a viewpoint ξ and a user prompt ω are contradictory, *i.e.*, $p_{\text{2D}}(\xi | \omega, \mathbf{z}_\theta) \ll p_{\text{2D}}(\xi | \mathbf{z}_\theta)$, then C approximates to 0 for every $\mathbf{z}_\theta$. Furthermore, the terms $\nabla_{\mathbf{z}_\theta} \log p_{\text{2D}}(\xi | \mathbf{z}_\theta)$ and $\nabla_{\mathbf{z}_\theta} \log p_{\text{2D}}(\omega | \mathbf{z}_\theta)$ have an adverse effect, making the optimization particularly challenging.

(a) Rendered image (b) 2D score (c) Generated result

Figure 3. **Visualization of the magnitude of the estimated $\nabla_{\mathbf{z}_\theta} \log p_{\mathbf{2D}}(\mathbf{z}_\theta|\xi,\omega)$ during the optimization.** This visualization demonstrates that erroneous 2D scores result in critical artifacts, *e.g.*, additional legs, beaks, and horns in this figure.

3. Score Debiasing

If the unconditional score, $\nabla_{\mathbf{z}_\theta} \log p_{\text{2D}}(\mathbf{z}_\theta)$, is biased towards some viewing directions, which is likely in 2D data as mentioned in Sec. 2, it can negatively affect the 3D consistency and realism of generated objects through the chain rule (Eq. 3). Moreover, large magnitudes in the user prompt gradient, $\nabla_{\mathbf{z}_\theta} \log p_{\text{2D}}(\omega|\mathbf{z}_\theta)$, can also cause issues by introducing text-related artifacts that are not present in the image rendered from a 3D field. Such artifacts include extra faces, beaks, and horns (see Fig. 1 and Fig. 3), which are unrealistic or inconsistent with the 3D object's structure.

High magnitude in those two terms is typically observed when the perturbed-and-denoised image by diffusion models significantly deviates from the rendered image in the corresponding pixels (Fig. 3). Hence, adjusting this gradient is necessary to reduce the artifacts and improve the realism of the generated 3D objects. However, the 2D bias that flows into the 3D field has hardly been formulated or adjusted for better optimization and 3D consistency.

Dynamic thresholding of 2D-to-3D scores. In light of the need to control the flow of 2D scores to 3D volume (Sec. 3) and inspired by the clipping methods [12, 19], we propose an effective method that dynamically truncates the scores in order to mitigate the effects of bias and artifacts in the predicted 2D scores. Specifically, we linearly increase the truncation value throughout the optimization:

$$\begin{aligned} \psi_{\text{dynamic}} &:= (1-\tau)\psi_{\text{start}} + \tau\psi_{\text{end}}, \\ \nabla_{\mathbf{z}_\theta} \log p_{\text{2D}} &= \text{Clip}(\nabla_{\mathbf{z}_\theta} \log p_{\text{2D}}(\mathbf{z}_\theta|\xi,\omega), \psi_{\text{dynamic}}), \end{aligned} \tag{6}$$

"Back view of a dog"

"Back view of a smiling dog"

Figure 4. **Samples from Stable Diffusion [18] given a text prompt with contradiction.** Despite "Back view of" is given in the prompts, the word "smiling" in the prompt makes diffusion models biased towards the front view of an object.

where $\tau = \frac{\text{(step)}}{\text{(max step)}}$ and $\text{Clip}(\mathbf{a}, b) = \max(\min(\mathbf{a}, b), -b)$. This is a coarse-to-fine strategy [4, 13], since in the early stages of optimization, we need to focus on the overall structure and shape, which do not require the large magnitudes of the 2D scores, while in later stages, we focus more on the details that require higher magnitudes. With this strategy, we obtain a better trade-off between 3D consistency and 2D faithfulness than statically clipping the scores.

4. Prompt Debiasing

Most text-to-3D generation methods that distill diffusion models [10, 16, 27] achieve a certain level of view consistency by concatenating view prompts (*e.g.*, "back view of") with user prompts. This simple and effective method utilizes the knowledge of large-scale text-to-image models. However, we argue that the current strategy of creating a view-dependent prompt by simply concatenating a view prompt with a user prompt is intrinsically problematic, as it can result in a contradiction between them. This contradiction is one of the causes that make diffusion models not follow a view prompt. Therefore, in the following subsection, we propose identifying the contradiction between the view prompt and user prompt using off-the-shelf language models trained with masked language modeling (MLM) [2].

Additionally, instead of naively assigning regular regions to view prompt augmentations, in the next subsection, we reduce the discrepancy between the view prompt and object-space pose by adjusting the regions.

Identifying contradiction utilizing language models. The prompt gradient term $\nabla_{\mathbf{z}_\theta} \log p_{\text{2D}}(\omega|\mathbf{z}_\theta)$ may cancel out the pose gradient term $\nabla_{\mathbf{z}_\theta} \log p_{\text{2D}}(\xi|\mathbf{z}_\theta)$ needed for the view consistency of generated 3D objects, as we can derive from Eq. 5. For example, if the view prompt is "back view of" and the user prompt is "a smiling dog", it results in a contradiction since an observer cannot see the dog's smile viewing from the back. This contradiction is one of the causes that makes diffusion models not follow a view prompt, but instead follow a word like "smiling" in a user prompt, as depicted in Fig. 4.

In this regard, we propose a method for identifying contradictions using language models trained with masked language modeling (MLM). Specifically, let V represent a set of possible view prompts, and let U be a set of size 2, which contains the presence and absence of a word in the user prompt for brevity. We then compute the following:

$$\text{PMI}(v, u) = \frac{P(v|u)}{\sum_{u' \in U} P(v|u')P(u')}, \tag{7}$$

where we can model $P(v|u)$ with masked language modeling by alternating the view prompts and normalizing them, and $P(u)$ is a user-defined faithfulness. If $P(u) = 1$, the word will never be removed from the user prompt. We then filter out the word u that has a lower value of $\text{PMI}(v, u)$ than a threshold. Note that Eq. 7 is equal to the pointwise mutual information (PMI) since:

$$\frac{P(v|u)}{\sum_{u' \in U} P(v|u')P(u')} = \frac{P(v|u)}{P(v)} = \frac{P(v, u)}{P(v)P(u)}, \tag{8}$$

and removing a contradiction implies removing the word that has a low PMI with the view prompts.

Reducing discrepancy between view prompts and object-space poses. Existing methods [10, 11, 16, 27] use view prompt augmentations by dividing the camera space into some regular sections (*e.g.*, front, back, side, and overhead in DreamFusion [16]). However, this approach does not match the real distribution of object-space poses in image-text pairs, *e.g.*, front view may cover a narrower region. Therefore, we make practical adjustments to the range of view prompts, such as reducing the azimuth range of the "front view" by half. Furthermore, we search for precise view prompts [16, 27] that give us improved results.

5. Comparison with Baseline

For the experiments, we use the highest-performing public codebase of SJC [27], using the same optimization hyperparameters for both for a fair comparison.

As shown in the qualitative results in Fig. 1, our methods reduce view inconsistencies in the 3D objects and mitigate the so-called Janus problem. This improvement come with little overhead compared to the baseline.

Method	A-LPIPS$_{\text{VGG}}$ ↓	A-LPIPS$_{\text{Alex}}$ ↓
Baseline [27]	0.2054	0.1526
Debiased (Preserved)	<u>0.1963</u>	<u>0.1450</u>
Debiased (Ours)	**0.1940**	**0.1445**

Table 1. **Quantitative evaluation.** The best values are in bold, and the second best are underlined. *Preserved* means user prompts are preserved, *i.e.*, $P(u) = 1$ for all u.

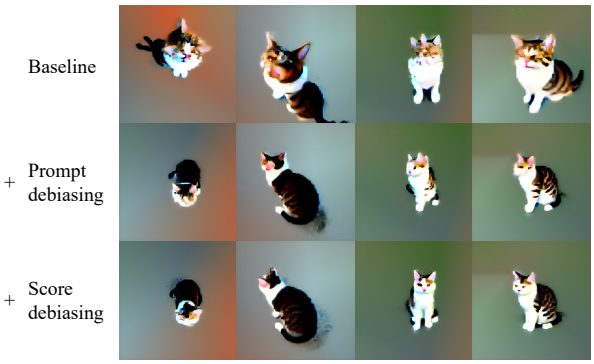

Figure 5. **Improvement of view consistency through prompt and score debiasing.** The baseline is original SJC [27], and *Prompt* and *Score* denote prompt and score debiasing, respectively. The given user prompt is *"a smiling cat,"* and the images are rendered from arbitrary viewpoints.

We propose a new metric (Adjacent LPIPS; A-LPIPS) to quantitatively measure the view consistency of 3D fields. It computes the average LPIPS [28] between adjacent images rendered from evenly spaced azimuths, using two different backbones [1, 22]. The intuition behind this metric is that two images from adjacent viewpoints are perceptually similar if the 3D field is consistent. Our method produces more consistent 3D objects than the baseline, as demonstrated in Table 1 based on 70 prompts. Note that removing contradictions in prompts leads to better results.

We present ablation results in Fig. 8, where we sequentially added prompt debiasing and score debiasing on top of the baseline. This demonstrates that they gradually improve the view consistency and reduce artifacts as intended.

Overall, the experiments corroborate that our debiasing methods improve the realism and alleviate the Janus problem of generated 3D objects, without requiring any 3D guide [21] or introducing significant overhead to the zero-shot text-to-3D setting.

6. Conclusion

In this paper, we formulate and identify the sources of the Janus problem in zero-shot text-to-3D generation. In this light, we argue that debiasing the prompts and raw 2D scores is essential for the realistic generation. Therefore, we propose two methods that increase the quality and are applicable to existing frameworks with little overhead without 3D supervision, showing potential for future research in this promising area.

References

[1] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009. 4

[2] 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. 2, 3, 8

[3] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *NeurIPS*, 34:8780–8794, 2021. 1, 2, 7

[4] Emilien Dupont, Hyunjik Kim, SM Eslami, Danilo Rezende, and Dan Rosenbaum. From data to functa: Your data point is a function and you should treat it like one. *ICML*, 2022. 2, 3

[5] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33:6840–6851, 2020. 1, 2, 7

[6] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 2, 7

[7] Susung Hong, Gyuseong Lee, Wooseok Jang, and Seungryong Kim. Improving sample quality of diffusion models using self-attention guidance. *arXiv preprint arXiv:2210.00939*, 2022. 2, 7

[8] Ajay Jain, Ben Mildenhall, Jonathan T Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. In *CVPR*, pages 867–876, 2022. 1, 8

[9] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *NeurIPS*, 2022. 1, 2, 7

[10] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. *arXiv preprint arXiv:2211.10440*, 2022. 1, 2, 3, 4

[11] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. *arXiv preprint arXiv:2211.07600*, 2022. 1, 4

[12] Tomas Mikolov. *Statistical Language Models based on Neural Networks*. PhD thesis, Brno University of Technology, 2012. 1, 3

[13] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1, 2, 3

[14] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 2, 7

[15] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *ICML*, pages 8162–8171. PMLR, 2021. 7

[16] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 1, 2, 3, 4, 6, 7, 8

[17] 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. 7

[18] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 1, 2, 3, 7, 8

[19] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022. 1, 2, 3, 7

[20] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4104–4113, 2016. 8

[21] Junyoung Seo, Wooseok Jang, Min-Seop Kwak, Jaehoon Ko, Hyeonsu Kim, Junho Kim, Jin-Hwa Kim, Jiyoung Lee, and Seungryong Kim. Let 2d diffusion model know 3d-consistency for robust text-to-3d generation. *arXiv preprint arXiv:2303.07937*, 2023. 4, 8

[22] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 4

[23] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 1, 7

[24] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *NeurIPS*, 32, 2019. 1, 7

[25] 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*, 2020. 1, 2, 7

[26] Jiaxiang Tang. Stable-dreamfusion: Text-to-3d with stable-diffusion, 2022. https://github.com/ashawkey/stable-dreamfusion. 6, 8

[27] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. *arXiv preprint arXiv:2212.00774*, 2022. 1, 2, 3, 4, 6, 7, 8, 10

[28] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 4, 8

Appendix

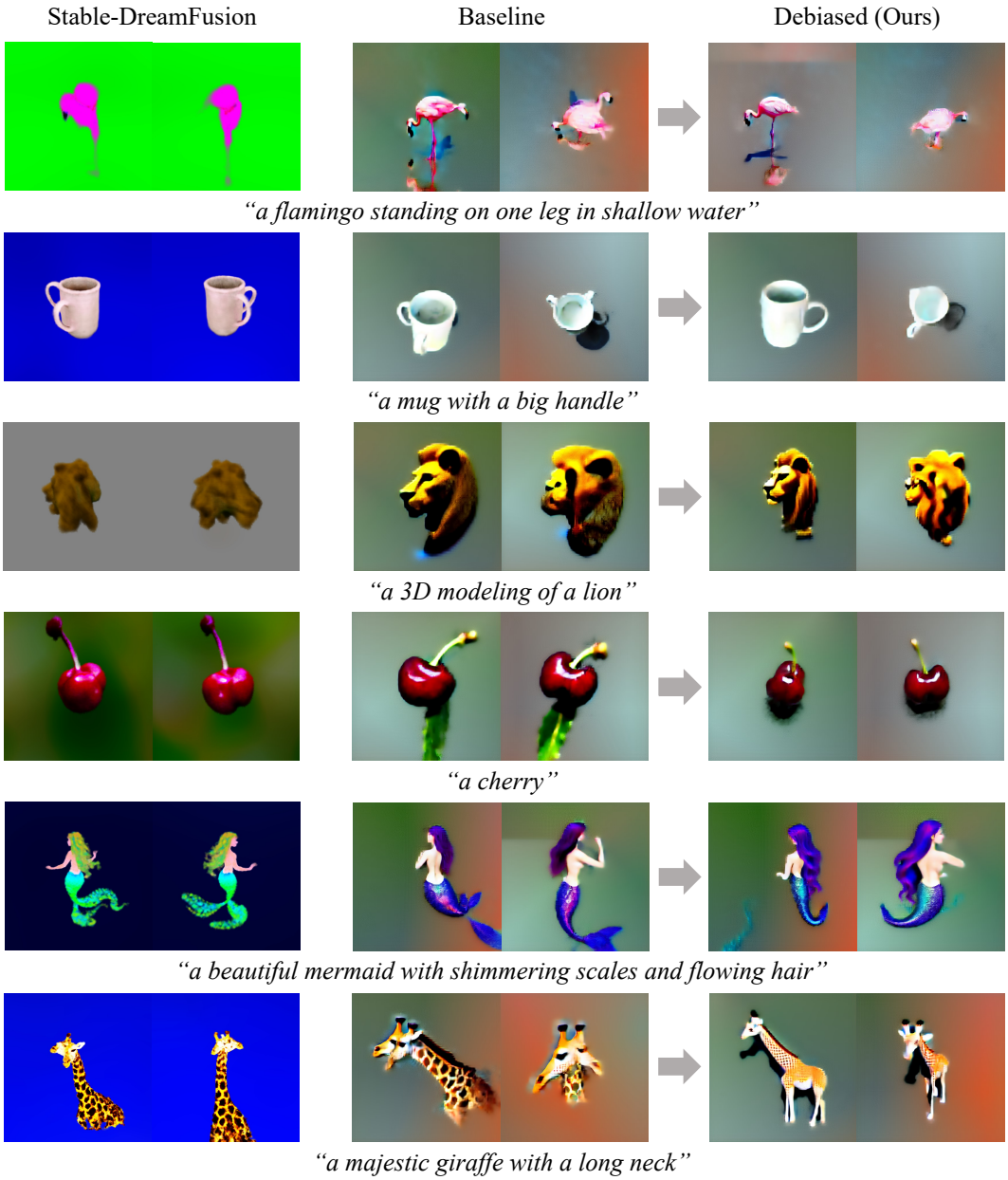

Figure 6. **Comparison between Stable-DreamFusion [16, 26], SJC [27], and ours.** The baseline is original SJC [27]. Our debiasing methods qualitatively reduce view inconsistencies in zero-shot text-to-3D and the so-called *Janus problem*.

A. Background

A.1. Diffusion models

Denoising diffusion models [5, 23] generate images through progressive denoising process. During training, denoising diffusion probabilistic models (DDPM) [5] optimize the following simplified objective:

$$L_{\text{DDPM}} := \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I}),\mathbf{x}_0,t}\left[\left\|\boldsymbol{\epsilon}-\boldsymbol{\epsilon}_\phi(\mathbf{x}_t,t)\right\|^2\right], \tag{9}$$

where $\boldsymbol{\epsilon}_\phi$ is a network of the diffusion model, $t \in \{T, T-1, ..., 1\}$ is a timestep, $\mathbf{x}_0$ is an original image, and $\mathbf{x}_t$ denotes a perturbed image according to the timestep t. During inference, starting from x_T, DDPMs sample a previous sample $\mathbf{x}_{t-1}$ from a normal distribution $p_\phi(\mathbf{x}_{t-1}|\mathbf{x}_t)$ defined by ϕ.

Some works fit DDPM into the generalized frameworks, *e.g.*, non-Markovian [23], score-based [9, 25], *etc*. Notably, denoising diffusion models have a tight relationship with score-based models [9, 25] in the continuous form. Furthermore, the work [9] show that denoising diffusion models can be refactored into the canonical form of denoising score matching using the same network parameterization. This formulation further facilitates the direct computation of 2D scores [9, 24] with the following equation:

$$\nabla_{\mathbf{x}} \log p(\mathbf{x};\sigma) = \frac{D_\phi(\mathbf{x};\sigma)-\mathbf{x}}{\sigma^2}, \tag{10}$$

where D_ϕ is an optimal denoiser network trained for every σ. With some preconditioning, a diffusion model $\boldsymbol{\epsilon}_\phi$ [5, 15, 18, 23] turns into a denoiser D_ϕ. In other words, 2D scores of diffusion models can be calculated this way.

Recent advancements in diffusion models have sparked increased interest in text-to-image generation [14, 17–19]. Diffusion guidance techniques [3, 6, 7, 14] have been developed to enable the control of the generation process based on various conditions such as class labels [3, 6], text captions [14], or internal information [7]. In particular, our work conditions text prompts with classifier-free guidance [6], which is formulated as follows given a conditional diffusion model $\boldsymbol{\epsilon}_\phi(\mathbf{x}_t, t, c)$:

$$\tilde{\boldsymbol{\epsilon}} = \boldsymbol{\epsilon}_\phi(\mathbf{x}_t, t, c) + s \cdot (\boldsymbol{\epsilon}_\phi(\mathbf{x}_t, t, c) - \boldsymbol{\epsilon}_\phi(\mathbf{x}_t, t)), \tag{11}$$

where $\tilde{\boldsymbol{\epsilon}}$ is the guided output, and s is the guidance scale.

A.2. Score distillation

Diffusion models have shown remarkable performance in text-to-image modeling [7, 14, 17–19]. On top of this, DreamFusion [16] proposes the score-distillation sampling (SDS) method that uses text-to-image diffusion models to optimize neural fields, achieving encouraging results. The score-distillation sampling utilizes the gradient computed by the following equation:

$$\nabla_\theta L_{\text{SDS}} \triangleq \mathbb{E}_{\boldsymbol{\epsilon}\sim\mathcal{N}(0,\mathbf{I}),\mathbf{z}_\theta,t}\left[w(t)(\boldsymbol{\epsilon}_\phi(\mathbf{z}_t,t)-\boldsymbol{\epsilon})\frac{\partial \mathbf{z}_\theta}{\partial\theta}\right], \tag{12}$$

where $\mathbf{z}_t$ denotes the t-step noised version of $\mathbf{z}_\theta$ which is a rendered image, and $w(t)$ is a scaling function only dependent on t. This gradient omits the Jacobian of the diffusion backbone, leading to tractable optimization of differential neural parameters [16].

On the other hand, in light of the interpretation of diffusion models as denoisers, SJC [27] presents a new approach directly using the score estimation for which the authors call perturb-and-average scoring (PAAS). The work shows that the U-Net Jacobian emerging in DreamFusion is not even necessary, as well as forming a strong baseline using publicly open Stable Diffusion [18]. The perturb-and-average score approximates to a score with an inflated noise level:

$$\nabla_{\mathbf{z}_\theta} \log p_{\sqrt{2}\sigma}(\mathbf{z}_\theta) \approx \mathbb{E}_{n\sim\mathcal{N}(0,\mathbf{I}),\mathbf{z}_\theta}\left[\frac{D(\mathbf{z}_\theta+\sigma n;\sigma)-\mathbf{z}_\theta}{\sigma^2}\right], \tag{13}$$

where the expectation is practically estimated by Monte Carlo sampling. This score estimate is then directly plugged into the 2D-to-3D chain rule and produces:

$$\nabla_\theta L_{\text{PAAS}} \triangleq \mathbb{E}_{\mathbf{z}_\theta}\left[\nabla_{\mathbf{z}_\theta}\log p_{\sqrt{2}\sigma}(\mathbf{z}_\theta)\frac{\partial \mathbf{z}_\theta}{\partial\theta}\right]. \tag{14}$$

Although the derivation is different from SDS in DreamFusion [16], it is straightforward to show that the estimation $\nabla_\theta L_{\text{PAAS}}$ is same as $\nabla_\theta L_{\text{SDS}}$ with a different weighting rule and sampler [9].

B. Implementation Details

B.1. Common settings

We build our debiasing methods upon the public repository of SJC [27]. For all the results, including SJC and ours, we run 10000 steps to optimize the 3D fields. We set the hyperparameters of SJC to specific constants and do not change them throughout the experiments.

B.2. Prompt debiasing

To compute the pointwise mutual information (PMI), we use the uncased model of BERT [2] to obtain the conditional probability. Additionally, we set $P(u) = 1$ for words that should not be erroneously omitted. Otherwise, we set $P(u) = 1/2$. To use a general language model for the image-related task, we concatenated "This image is depicting a" when evaluating the PMI between the view prompt and user prompt. We first get u, v pairs such that $\frac{P(v,u)}{P(v)P(u)} < 1$. Then, given a view prompt, we remove words whose PMI for that view prompt, normalized across all view prompts, is below 0.95.

For the view prompt augmentation, we typically follow the view prompt assignment rule of DreamFusion [16] and SJC [27]. However, we slightly modify the view prompts and azimuth ranges for each prompt as mentioned in Sec. 4. For example, we assign an azimuth range of $[-22.5°, 22.5°]$ for the "front view." Also, we empirically find that using a view prompt augmentation $v \in \{$"$front\ view$", "$back\ view$", "$side\ view$", "$top\ view$"$\}$ without "of" depending on a viewpoint gives us improved results for Stable Diffusion v1.5 [18].

B.3. Score debiasing

In terms of score debiasing, we gradually increase the truncation threshold from one fourth of the pre-defined threshold to the pre-defined threshold, according to the optimization step. Specifically, we linearly increase the threshold from 2.0 to 8.0 for all experiments that leverage dynamic thresholding of 2D-to-3D scores.

B.4. Evaluation metrics (A-LPIPS$_{\text{VGG, Alex}}$)

Quantitatively evaluating a zero-shot text-to-3D framework is challenging due to the absence of ground truth 3D scenes that correspond to the text prompts. Existing works employ CLIP R-Precision [8,16]. However, it measures retrieval accuracy through projected 2D images and text input, making it unsuitable for quantifying the view consistency of a scene.

To address this issue, a concurrent work [21] proposes a new metric that utilizes COLMAP [20] to measure the consistency of a generated 3D scene. However, we find that this metric largely depends on the accuracy of the sequential reconstruction provided by off-the-shelf COLMAP, which we empirically determine to be inaccurate for rendered images from SJC [27]. For instance, it occasionally omits camera poses for some images if features are not matched, resulting in extremely large variances between distances.

Therefore, to measure the view consistency of generated 3D objects quantitatively, we compute the average LPIPS [28] between adjacent images, which we refer to as A-LPIPS. We sample 100 uniformly spaced camera poses from an upper hemisphere of a fixed radius, all directed towards the sphere's center at an identical elevation, and render 100 images from a 3D scene. Then, we average the LPIPS values evaluated for all adjacent pairs of images in the 3D scene, finally aggregating those averages across the scenes. The intuition behind this is that if there exist artifacts or view inconsistencies in a generated 3D scene, the perceptual loss will be large near those points.

C. More Results

C.1. Dynamic thresholding of 2D-to-3D scores

To show some examples of the effect of dynamic thresholding, we compare results of dynamic thresholding with those of static thresholding and no thresholding in Fig. 7. It demostrates that dynamic clipping reduce artifacts with better realism.

C.2. Qualitative results

We present additional qualitative results in Fig. 6 and ablation results in Fig. 8. In addition to the results of SJC [27], which serves as the baseline for our experiments, we include those of Stable-DreamFusion [26], an unofficial re-implementation of DreamFusion [16] that utilizes Stable Diffusion [18]. The results demonstrate that our methods significantly reduces the Janus or view inconsistency problem.

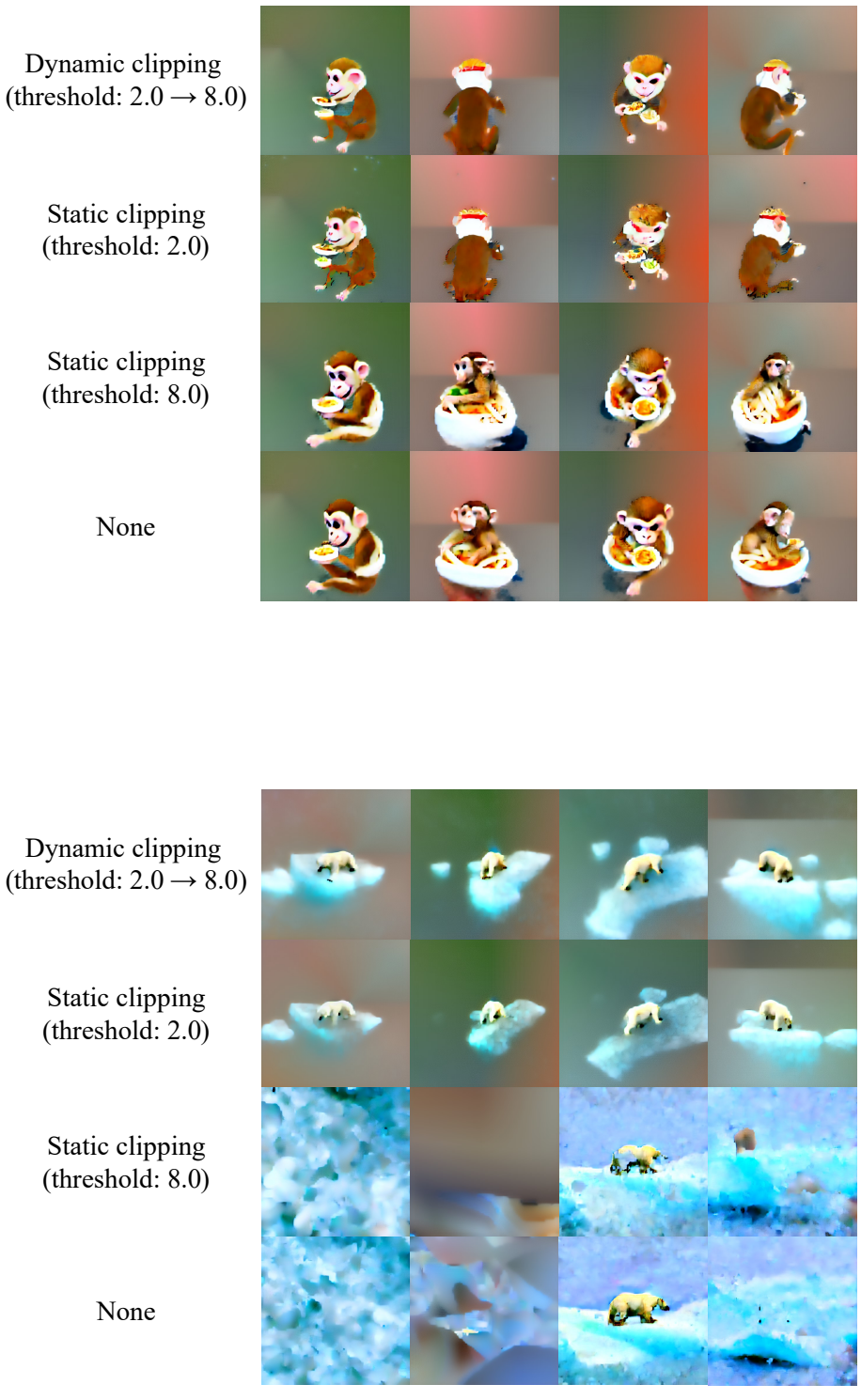

Figure 7. **Dynamic thresholding of 2D-to-3D scores.**

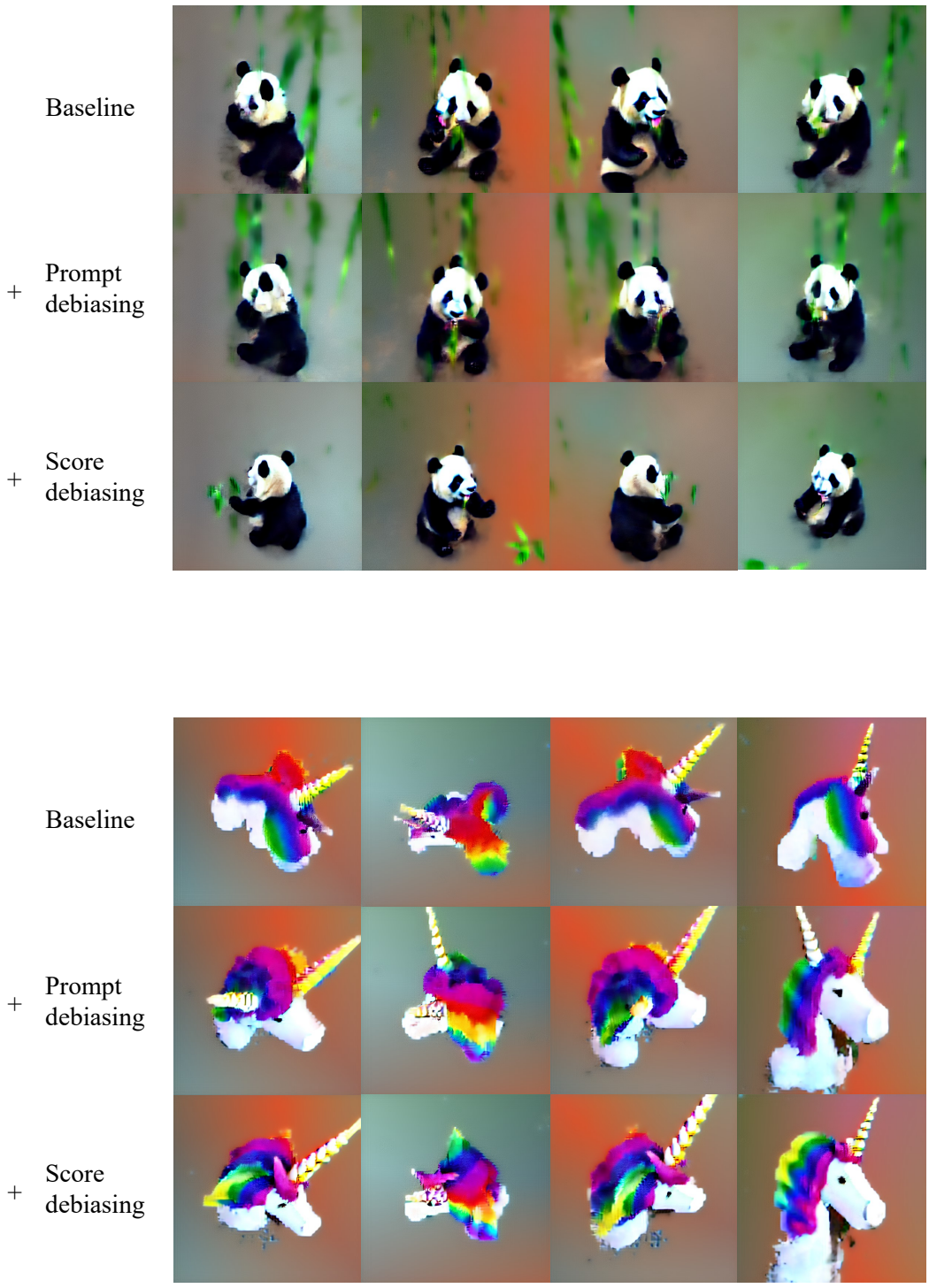

Figure 8. **Improvement of view consistency through prompt and score debiasing.** The baseline is original SJC [27], and *Prompt* and *Score* denote prompt and score debiasing, respectively. The given user prompts are *"a cute and chubby panda munching on bamboo,"* and *"an unicorn with a rainbow horn."*