JointDreamer: Ensuring Geometry Consistency and Text Congruence in Text-to-3D Generation via Joint Score Distillation —Supplementary Material—

Chenhan Jiang[1*†], Yihan Zeng[2*], Tianyang Hu[2], Songcun Xu[2], Wei Zhang[2], Hang Xu[2], and Dit-Yan Yeung[1]

[1] The Hong Kong University of Science and Technology
[2] Huawei Noah's Ark Lab
[*]Equal contribution. [†]Corresponding Author: jchcyan@gmail.com
https://jointdreamer.github.io

This supplementary material consists of five parts, including technical details of the experimental setup (Sec. A), the derivation of Joint Score Distillation (JSD) (Sec. B), additional ablation analysis (Sec. C), additional experimental results (Sec. D) and the Janus prompt list (Sec. E).

A Experimental Setup

A.1 Details of JointDreamer Pipeline.

In our main text, we adopt MVDream $\mathcal{M}_{\text{MVS}}$ as the energy function for the overall JointDreamer pipeline. Since MVDream fine-tunes on SD-V2.1, we retain SD-V2.1 as a diffusion model. The whole training procedure includes 6k iterations, taking around 1.5 h with batch size 4 on 1 Nvidia Tesla A800 GPU. Specifically, we warm up NeRF for the initial 600 training iterations with SDS and adopt JSD for the remaining iterations. We adopt the common time-annealing and resolution-increasing tricks from the open-source implementation, together with the two proposed mechanisms including the Geometry Fading scheme and Classifier-Free Guidance (CFG) Scale switching strategy. We set $t = 0.98$ with resolution 64 for the first 3k iterations and then anneal into $t \sim U(0.02, 0.50)$ with resolution 256 for the extra 2k iterations. Starting from iteration 5k, we scale up the resolution to 512 and conduct the two proposed mechanisms, where the learning rate of the density network is reduced from $1e-2$ to $1e-6$ and the CFG scale is switched from 30 to 50. The Geometry Fading scheme and Classifier-Free Guidance (CFG) Scale switching strategy allow greater influence from coherence guidance in JSD on geometry optimization in the early training stages and enhance the fidelity of textures in later stages.

A.2 Details of Binary Classification Model.

In this part, we will elaborate on the model architecture and training procedure of the binary classification model that is discussed in Sec.4.2 in the main paper.

Model Architecture. We build the model based on the DINO framework. Specifically, we employ ViT-s16 as the backbone for extracting image features. The backbone is initially pre-trained following the DINO method, and during training, the first 9 blocks of the backbone are frozen. Besides, we use a 4-layer MLP with 256 hidden layer channels to extract the relative camera embedding of the transformation matrix between input images, which captures the camera-specific information. Next, we calculate the cross-attention between camera embedding and the concatenated image features of input image pairs. This cross-attention mechanism generates a residual feature input, combined with the concatenated image features as the final feature. Finally, the combined features are fed into the classification head consisting of a 3-layer MLP, which produces the classification logit prediction for input image pairs.

Training Procedure. For training data, we use rendered images from Objaverse [1] following Zero-1-to-3 [3]. For the binary classification training objective, we adopt the pairs of images from the same object equipped with the correct camera pose as the positive samples and assign the image pairs from different objects or incorrect relative camera poses as negative samples. Before training, we prepare the index list of positive and negative pairs for efficient training. During training, we randomly sample 1 million positive pairs and 1 million negative pairs from the index list as training sets. The design of the training set ensures that the classification model can identify the 3D consistency between rendered images conditioned on relative camera pose. We adopt adamW optimizer with $5e-4$ learning rate and 0.04 weight decay. We also adopt random color jitter, gaussian blur, and polarization following DINO as data augmentation. We use an image size of 224×224 and a total batch size of 640 and train the model for 10 epochs. The training takes about 1 day on 2 Nvidia Tesla A800 GPUs. To validate the classification accuracy, We random sample 5000 pairs as the validation set. The training loss and validation accuracy curve can be found in Fig. A1.

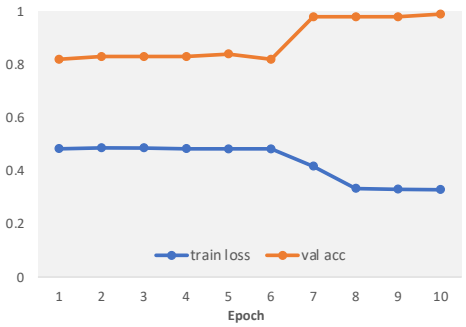

Fig. A1: Training loss and validation accuracy curves of the proposed Binary Classification Model.

A.3 Details of Text-to-3D Generation Comparison

Baseline Setup. We implement the experiments in an open-source three-studio project and reproduce DreamFuion-IF, Magic3D-IF-SD, and ProlificDreamer as baselines following the comparisons in the main paper of MVDream. Our MVDream baseline is reproduced by its officially released code. We adopt DeepFloyd-IF [7] as the 2D diffusion model for baseline DreamFuion-IF and

Fig. A2: More quality results of JSD with Classification Model.

the first stage of Magic3D-IF-SD following MVDream. To make a fair comparison with our JointDreamer, we equip the same batch size, resolution, and time annealing strategy with JointDreamer for DreamFuion-IF.

Evaluation Details. We conducted a user study from 100 users on the 153 generated models from the object-centric MS-COCO subset. Each user is given 4 rendered videos with their corresponding text input from generations of different methods. We ask the users to select a preferred 3D model from four options, and then calculate the mean proportion of each method selected over all 153 prompts as the score. The higher score indicates the greater user preference. For the Clip Score and Clip R-Precision, we adopt the CLIP ViT-B/32 as the feature extractor.

A.4 Details of Computational Resource Comparison

We analyze the geometry consistency and computation efficiency of various view-aware models in main paper Table **??**, using 16 complex multi-Janus prompts in Sec. E from the DreamFusion [5] library. We maintain consistent experimental parameters, including a batch size of 4, training 5k iterations and a resolution of 64, as well as the same optimizer and time annealing hyperparameters. The only variation is in the camera parameters, which align with each view-aware model's settings. For the baseline SDS model, we adopt the DreamFusion camera parameters. We present some examples showcasing these results incorporating $\mathcal{C}_{\mathbf{CLS}}$ in Figure A2. And the results incorporating $\mathcal{C}_{\mathbf{MVS}}$ can be found in Section D.

B Theory of Joint Score Distillation

Given a well-trained text-to-image diffusion model, like Stable Diffusion, the objective is to distill its knowledge into a 3D representation network parameterized by θ, such as NeRF and ensures coherent 3D generations. To achieve this, we aim to model the joint rendering distribution across multiple views of θ.

For ease of notation, we define $\tilde{\boldsymbol{x}}$ as the joint random variable comprising $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^V$, which are rendered images sampled from the 3D representation θ. It is important to note that these views are not independent. In a 3D model, the views are inherently connected as they originate from the same underlying 3D object. This means that the rendered images, $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^V$, exhibit dependencies and correlation.

Denote the joint rendering distribution of $\tilde{\boldsymbol{x}}$ as $\tilde{q}^\theta$. We can still define the marginal distributions as

$$q^\theta(\boldsymbol{x}^i) = \int \tilde{q}^\theta(\tilde{\boldsymbol{x}}) d\tilde{\boldsymbol{x}}^{-i},$$

where $\tilde{\boldsymbol{x}}^{-i} = \boldsymbol{x}^1, \ldots, \boldsymbol{x}^{i-1}, \boldsymbol{x}^{i+1}, \ldots, \boldsymbol{x}^V$. This marginal distribution is the same as if only a single view is considered, i.e., $V = 1$.

We can further define the log density ratio as

$$R(\tilde{\boldsymbol{x}}) = \log \frac{\tilde{q}^\theta(\tilde{\boldsymbol{x}})}{\prod_{i=1}^V q^\theta(\boldsymbol{x}^i)}$$

to capture the inter-relationship among different views. Equivalently, we can write

$$\tilde{q}^\theta(\tilde{\boldsymbol{x}}) = \exp(R(\tilde{\boldsymbol{x}})) \prod_{i=1}^V q^\theta(\boldsymbol{x}^i).$$

To get the evaluations of $\tilde{\boldsymbol{x}}$ from the 2D diffusion model, we have

$$\tilde{p}(\tilde{\boldsymbol{x}}) \propto \exp(\mathcal{C}(\tilde{\boldsymbol{x}})) \prod_{i=1}^V p(\boldsymbol{x}^i)$$

since the diffusion model only takes a single image as input and different views are weighted by the introduced joint energy function $\mathcal{C}$.

Now we consider learning $\tilde{q}^\theta(\tilde{\boldsymbol{x}})$ such that the following Integral Kullback–Leibler (IKL) divergence is minimized along the forward diffusion process $\boldsymbol{x}_t = \alpha_t \boldsymbol{x}_0 + \sigma_t \epsilon$ where ϵ follows standard Gaussian distribution.

$$\begin{aligned}\min_\theta D_{\mathrm{IKL}}(\tilde{q}^\theta(\tilde{\boldsymbol{x}})||\tilde{p}(\tilde{\boldsymbol{x}})) &= \min_\theta \int_0^T w(t)\frac{\sigma_t}{\alpha_t} D_{\mathrm{KL}}(\tilde{q}_t^\theta(\tilde{\boldsymbol{x}})||\tilde{p}_t(\tilde{\boldsymbol{x}}))dt \\ &= \min_\theta \int_0^T w(t)\frac{\sigma_t}{\alpha_t} \mathbb{E}_{\tilde{\boldsymbol{x}}_t\sim\tilde{q}_t^\theta}\left(\log\frac{\tilde{q}_t^\theta(\tilde{\boldsymbol{x}}_t)}{\tilde{p}_t(\tilde{\boldsymbol{x}}_t)}\right)dt.\end{aligned}$$

Taking gradient with respect to θ gives

$$\begin{aligned}&\frac{\partial}{\partial\theta} D_{\mathrm{IKL}}(\tilde{q}^\theta(\boldsymbol{x})||\tilde{p}(\boldsymbol{x})) \\ &= \int_0^T w(t)\frac{\sigma_t}{\alpha_t}\frac{\partial}{\partial\theta}\mathbb{E}_{\tilde{\boldsymbol{x}}_t\sim\tilde{q}_t^\theta}\left(\log\frac{\tilde{q}_t^\theta(\tilde{\boldsymbol{x}}_t)}{\tilde{p}_t(\tilde{\boldsymbol{x}}_t)}\right)dt \\ &= \int_0^T w(t)\frac{\sigma_t}{\alpha_t}\mathbb{E}_{\tilde{\boldsymbol{x}}_t\sim\tilde{q}_t^\theta}\left[\frac{\partial}{\partial\tilde{\boldsymbol{x}}_t}\left(\log\frac{\tilde{q}_t^\theta(\tilde{\boldsymbol{x}}_t)}{\tilde{p}_t(\tilde{\boldsymbol{x}}_t)}\right)\frac{\partial\tilde{\boldsymbol{x}}_t}{\partial\theta} + \frac{\partial}{\partial\theta}\log\tilde{q}_t^\theta(\boldsymbol{x})|_{\boldsymbol{x}=\tilde{\boldsymbol{x}}_t}\right]dt \\ &:= A + B.\end{aligned}$$

The term B vanishes since

$$\begin{aligned}B &= \int_0^T w(t)\frac{\sigma_t}{\alpha_t}\mathbb{E}_{\tilde{\boldsymbol{x}}_t\sim\tilde{q}_t^\theta}\frac{\partial}{\partial\theta}\log\tilde{q}_t^\theta(\boldsymbol{x})|_{\boldsymbol{x}=\tilde{\boldsymbol{x}}_t}dt \\ &= \int_0^T w(t)\frac{\sigma_t}{\alpha_t}\mathbb{E}_{\tilde{\boldsymbol{x}}_t\sim\tilde{q}_t^\theta}\frac{\frac{\partial}{\partial\theta}\tilde{q}_t^\theta(\boldsymbol{x})|_{\boldsymbol{x}=\tilde{\boldsymbol{x}}_t}}{\tilde{q}_t^\theta(\tilde{\boldsymbol{x}}_t)}dt \\ &= \int_0^T w(t)\frac{\sigma_t}{\alpha_t}\int\frac{\partial}{\partial\theta}\tilde{q}_t^\theta(\boldsymbol{x})|_{\boldsymbol{x}=\tilde{\boldsymbol{x}}_t}dt \\ &= \int_0^T w(t)\frac{\sigma_t}{\alpha_t}\frac{\partial}{\partial\theta}\int\tilde{q}_t^\theta(\boldsymbol{x})dt \\ &= 0\end{aligned}$$

The term A is the score distillation loss

$$A = \int_0^T w(t)\frac{\sigma_t}{\alpha_t}\mathbb{E}_{\tilde{\boldsymbol{x}}_0\sim\tilde{q}_0^\theta,\tilde{\epsilon}}\left(\nabla\log\tilde{q}_t^\theta(\tilde{\boldsymbol{x}}_t) - \nabla\log\tilde{p}_t(\tilde{\boldsymbol{x}}_t)\right)\frac{\partial\tilde{\boldsymbol{x}}_t}{\partial\theta}dt,$$

where $\tilde{\epsilon} = (\epsilon^1, \ldots, \epsilon^V)$ are the noises along the forward diffusion process. Putting things together we have

$$\frac{\partial}{\partial\theta} D_{\mathrm{IKL}}(\tilde{q}^\theta(\boldsymbol{x})||\tilde{p}(\boldsymbol{x})) = \mathbb{E}_{\tilde{\boldsymbol{x}}_0\sim\tilde{q}_0^\theta,\tilde{\epsilon},t}\left[w(t)\frac{\sigma_t}{\alpha_t}\left(\nabla\log\tilde{q}_t^\theta(\tilde{\boldsymbol{x}}_t) - \nabla\log\tilde{p}_t(\tilde{\boldsymbol{x}}_t)\right)\frac{\partial\tilde{\boldsymbol{x}}_t}{\partial\theta}\right]$$

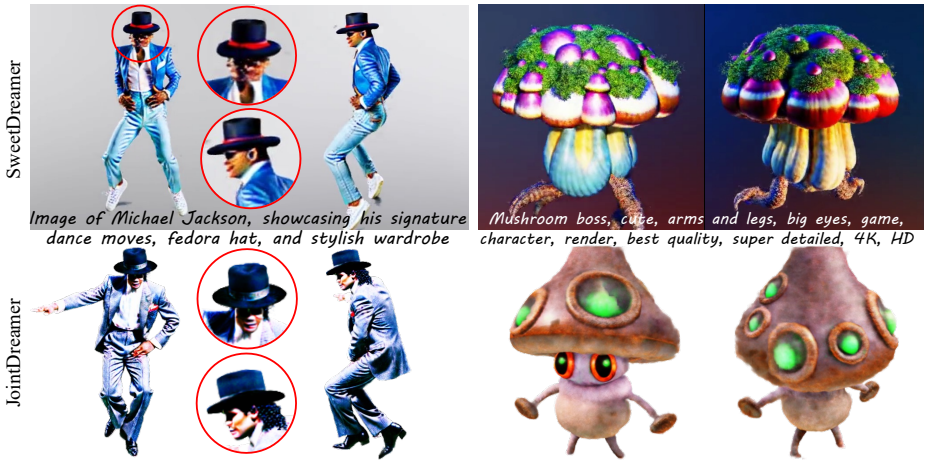

Fig. A3: Comparison with SweetDreamer. SweetDreamer suffers from multi-faces (left) and missing components such as "legs" and "eyes" (right).

Notice that the NeRF rendering is a deterministic process given the view information. Therefore, the conditional distribution and marginal distribution coincide, i.e.,

$$\tilde{q}_t^\theta(\tilde{\boldsymbol{x}}_t) \sim N(\alpha_t \tilde{\boldsymbol{x}}_0, \sigma_t^2), \quad \nabla \log \tilde{q}_t^\theta(\tilde{\boldsymbol{x}}_t) = -\tilde{\epsilon}/\sigma_t.$$

On the other hand, direct score matching tells us that

$$\nabla \log p_t(\boldsymbol{x}_t^i) = \frac{\partial \mathcal{C}(\tilde{\boldsymbol{x}})}{\partial \boldsymbol{x}_t^i} - \hat{\epsilon}_\Phi(\boldsymbol{x}_t^i, t)/\sigma_t.$$

Finally, combining $\frac{\partial \boldsymbol{x}_t^i}{\partial \theta} = \alpha_t \frac{\partial \boldsymbol{x}_0^i}{\partial \theta}$, we have

$$\frac{\partial}{\partial \theta} D_{\mathrm{IKL}}(\tilde{q}^\theta(\boldsymbol{x}) || \tilde{p}(\boldsymbol{x})) = \mathbb{E}_{\tilde{\boldsymbol{x}}_0 \sim \tilde{q}_0^\theta, \tilde{\epsilon}, t} \left[w(t) \sum_{i=1}^{V} \left(\hat{\epsilon}_\Phi(\boldsymbol{x}_t^i, t) - \frac{\partial \mathcal{C}(\tilde{\boldsymbol{x}})}{\partial \boldsymbol{x}_t^i} - \epsilon^i \right) \frac{\partial \boldsymbol{x}_0^i}{\partial \theta} \right]. \tag{1}$$

Now we have finished extending SDS to multiple views. As it turns out, the joint energy term $R(\tilde{\boldsymbol{x}})$ does not show up in the gradient formula.

C Additional Ablation Study

C.1 Comparison with SweetDreamer

We also conduct a comparison with SweetDreamer [2]. SweetDreamer aligns geometric priors (AGP) in a finetuned diffusion model and combines AGP with SDS to address the Janus issue. In contrast, JSD improves the optimization objective of SDS with various energy functions, and AGP can be one of them. For 3D generation, Fig. **??** shows that a simple combination, like SweetDreamer's, uses more

memory and complicates balancing components. Compared to SweetDreamer's demos from its website, our JointDreamer achieves better shape and text congruence without multi-faces and missing components ("arms", "big eyes") in Fig. A5.

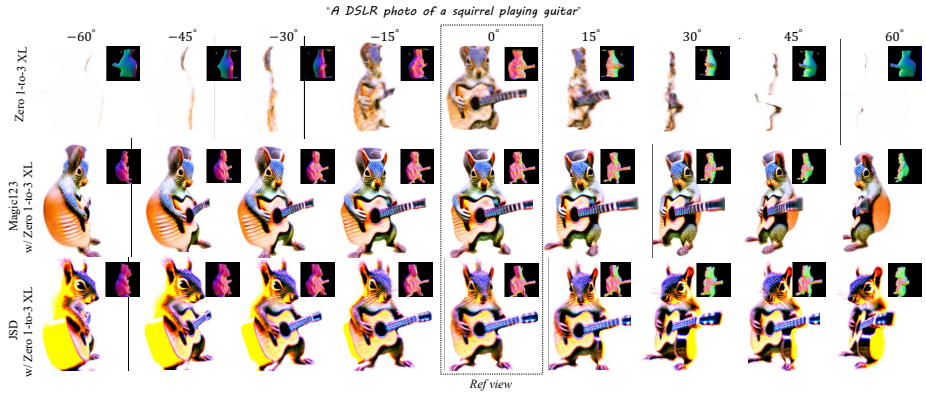

Fig. A4: Comparison with Image-to-3D methods. Compared with two alternative methods, all employing the Zero-1-to-3 XL model, our proposed JSD exhibits superior generative quality in novel view synthesis as evidenced by its geometric consistency.

C.2 Discussions on Image-to-3D Methods

Since the view-aware models can engage in 3D generation through SDS besides JSD, we make comparisons to showcase the superiority of JSD. Section 5.2 details the comparative use of MVDream, and herein, we extend this comparison to different applications of the image-to-image translation model, Zero-1-to-3 XL, which excels in image-to-3D tasks. Unlike text-to-3D approaches that generate 3D models from textual descriptions, the image-to-3D method uses a reference image to fix the reference view and generate the remaining views. As shown in Fig. A4, we input a reference image, exemplified by the front-view rendered image of the case of "A DSLR photo of a squirrel playing guitar" in Fig. A6 and compare with two alternative utilizations of Zero-1-to-3 XL. *(i)Zero-1-to-3 XL [3]*, which directly utilizes Zero-1-to-3 XL to calculate SDS loss for novel rendered views according to reference view. The overfitting generalizability of Zero-1-to-3 XL reduces the generative quality, especially for the views distant from the reference view. *(ii)Magic123 [6]*, which merges the SDS loss of SD-V2.1 and Zero-1-to-3 XL as objective function. By combining the generalizability from the original diffusion model, it can eliminate the distortion in novel views, but the effect is not satisfactory. By contrast, our JSD achieves better generation quality in novel views, where the overall geometric structure is more reasonable. Notably, when

applying JSD in image-to-3D generation, we calculate the inter-view coherence between the reference view and random novel views to fix the reference view, differing from the two random novel views used in text-to-3D generation. The comparisons further illustrate that JSD provides the optimal solution to combine generalizability from 2D models and geometric understanding from 3D-aware models.

C.3 Discussion on Failure Cases

Despite JointDreamer's impressive performance in handling detailed descriptions and multi-object combinations in long texts (as depicted in Fig. 1 of the main paper), it faces difficulties in comprehending complex relationships among objects. Specifically, it struggles to grasp relative spatial arrangements and hierarchical dependencies, as evidenced in Fig. A5. Exploring the use of larger diffusion models, such as SDXL [4], may offer a potential solution to overcome these limitations.

Fig. A5: Failure Cases on MS-COCO Subset.

D Additional Results of JointDreamer

We present more comparisons of text-to-3D generation as shown in Fig. A6, A7 and A8. The results indicate that JointDreamer outperforms current text-to-3D generation methods regarding generation fidelity, geometric consistency, and text congruence. This further validates the effectiveness and generalization of the proposed JSD. We also provide more images and normal maps from additional generated results in Fig. A9, demonstrating the generalizability of JointDreamer with arbitrary textual descriptions.

E Janus Prompts.

Our list of 16 Janus prompts is shown below:

"a blue jay standing on a large basket of rainbow macarons",
"a confused beagle sitting at a desk working on homework",
"Albert Einstein with grey suit is riding a moto",
"a panda rowing a boat in a pond",

"a wide angle zoomed out DSLR photo of a skiing penguin wearing a puffy jacket",
"a zoomed out DSLR photo of a baby monkey riding on a pig",
"a zoomed out DSLR photo of a fox working on a jigsaw puzzle",
"a DSLR photo of a pigeon reading a book",
"a DSLR photo of a cat lying on its side
batting at a ball of yarn"
"A crocodile playing a drum set"
"a rabbit cutting grass with a lawnmower",
"A red dragon dressed in a tuxedo and playing chess",
"a zoomed out DSLR photo of a bear playing electric bass",
"A bald eagle carved out of wood, more detail",
"A pig wearing back pack".
"a lemur drinking boba".

References

1. Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., VanderBilt, E., Schmidt, L., Ehsani, K., Kembhavi, A., Farhadi, A.: Objaverse: A universe of annotated 3d objects. In: CVPR. pp. 13142–13153 (2023)
2. Li, W., Chen, R., Chen, X., Tan, P.: Sweetdreamer: Aligning geometric priors in 2d diffusion for consistent text-to-3d. In: ICLR (2024)
3. Liu, R., Wu, R., Van Hoorick, B., Tokmakov, P., Zakharov, S., Vondrick, C.: Zero-1-to-3: Zero-shot one image to 3d object. In: ICCV. pp. 9298–9309 (2023)
4. Podell, D., English, Z., Lacey, K., Blattmann, A., Dockhorn, T., Müller, J., Penna, J., Rombach, R.: Sdxl: Improving latent diffusion models for high-resolution image synthesis. arXiv preprint arXiv:2307.01952 (2023)
5. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion. In: ICLR (2023)
6. Qian, G., Mai, J., Hamdi, A., Ren, J., Siarohin, A., Li, B., Lee, H.Y., Skorokhodov, I., Wonka, P., Tulyakov, S., et al.: Magic123: One image to high-quality 3d object generation using both 2d and 3d diffusion priors. arXiv preprint arXiv:2306.17843 (2023)
7. Shonenkov, A., Konstantinov, M., Bakshandaeva, D., Schuhmann, C., Ivanova, K., Klokova, N.: Deepfloyd. `https://huggingface.co/DeepFloyd` (2023)

Fig. A6: More comparison of text-to-3D generation.

Fig. A7: More comparison of text-to-3D generation.

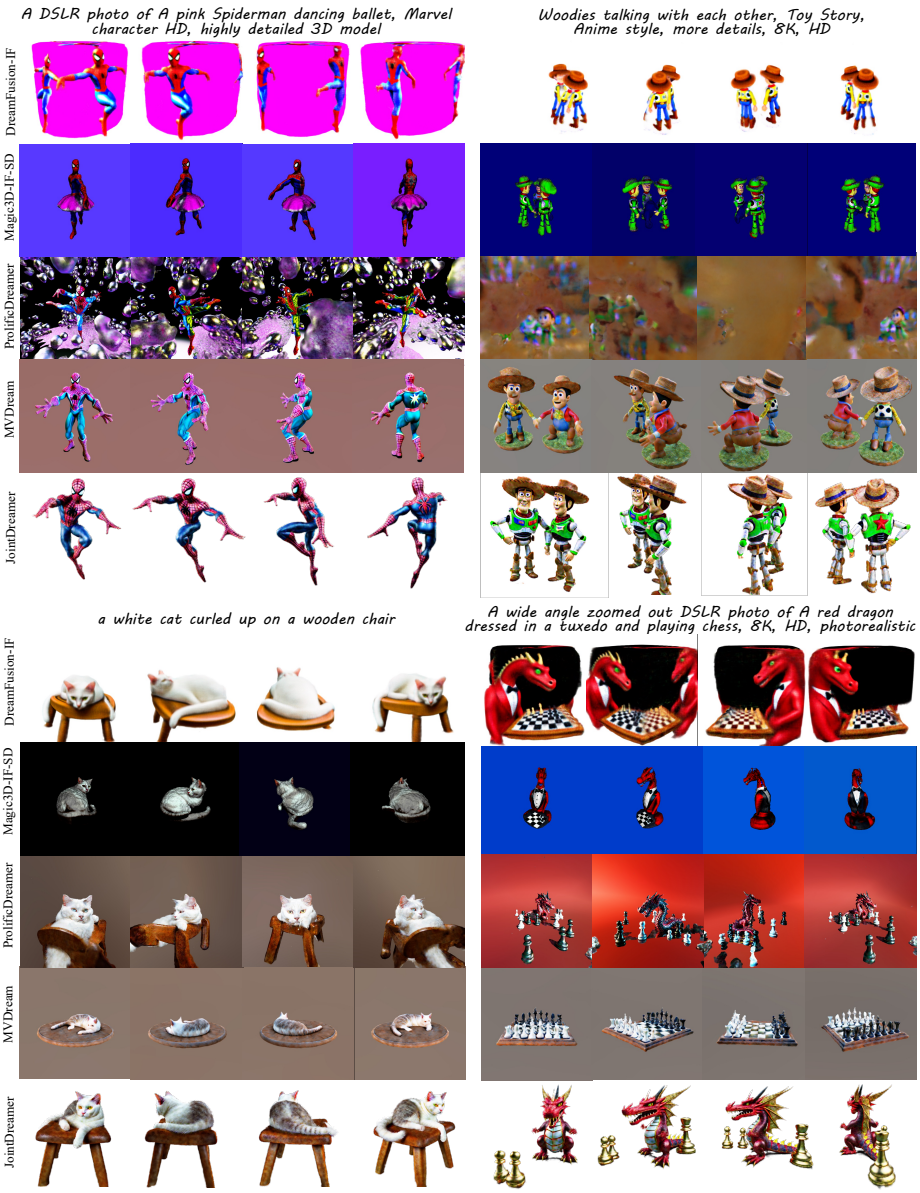

Fig. A8: More comparison of text-to-3D generation.

Fig. A9: More results of JointDreamer.