DreamView: Injecting View-specific Text Guidance into Text-to-3D Generation

Junkai Yan[1,2,3,†], Yipeng Gao[1,2,3,†], Qize Yang[3], Xihan Wei[3], Xuansong Xie[3], Ancong Wu[1,2,⋆], and Wei-Shi Zheng[1,2,⋆]

[1] School of Computer Science and Engineering, Sun Yat-sen University, China; [2] Key Laboratory of Machine Intelligence and Advanced Computing, Ministry of Education, China; [3]Institute for Intelligent Computing, Alibaba Group

{yanjk3,gaoyp23}@mail2.sysu.edu.cn,
{qize.yqz,xihan.wxh,xingtong.xxs}@alibaba-inc.com,
wuanc@mail.sysu.edu.cn, wszheng@ieee.org

Fig. 1: Text-to-3D generation (RGB images and normal maps) of our **DreamView**, where users can control what and where to generate via providing an overall text description (surrounded by a black box on the figure) and view-specific texts, thus achieving customizable 3D generation. The subject, object, and the position of the object in the overall text are marked in red, blue, and green, respectively. For the two generated results on the right, we only show the overall text.

Abstract. Text-to-3D generation, which synthesizes 3D assets according to an overall text description, has significantly progressed. However, a challenge arises when the specific appearances need customizing at designated viewpoints but referring solely to the overall description for generating 3D objects. For instance, ambiguity easily occurs when producing a T-shirt with distinct patterns on its front and back using a single overall text guidance. In this work, we propose **DreamView**, a text-to-image approach enabling multi-view customization while maintaining

⋆ : Corresponding authors are A.Wu and WS.Zheng. † : Equal contribution. This work was done when J. Yan and Y. Gao were interns at Alibaba.

overall consistency by adaptively injecting the view-specific and overall text guidance through a collaborative text guidance injection module, which can also be lifted to 3D generation via score distillation sampling. DreamView is trained with large-scale rendered multi-view images and their corresponding view-specific texts to learn to balance the separate content manipulation in each view and the global consistency of the overall object, resulting in a dual achievement of customization and consistency. Consequently, DreamView empowers artists to design 3D objects creatively, fostering the creation of more innovative and diverse 3D assets. Code and model will be released at here.

Keywords: Generative model · Text-to-image generation · Text-to-3D generation

1 Introduction

The surge in demand for diverse 3D asset creation spans many domains, including robotics simulation [7, 20], vison recognition with synthesis [10–12, 24, 41, 46, 49, 60], and architecture design [5, 17, 31], especially with the advancement of virtual and augmented reality technologies [19, 45, 58]. Despite the broadening scope of application, massively producing professional-grade 3D content necessitates artistic sensibility coupled with specialized skills in 3D modeling. Recent 3D synthesis works [6, 17, 26, 31, 33, 34, 43, 51, 52] attempt to produce high-quality 3D assets without much labor effort. Among them, text-to-3D generation [17,22,31,33,44,53] has garnered considerable attention for its ability to create 3D assets from text prompts, utilizing text-2D prior for text-3D representation learning [9,25,56,57,59,61].

Existing text-to-3D works can be divided into two streams: one conducts a direct generation [17, 31], and another adopts 2D pre-trained text-to-image models to optimize differentiable 3D representations [6,33,51], known as the 2D-lifting method. The latter has shown a promising ability to produce high-fidelity 3D assets. However, in the evolving landscape of text-to-3D generation, a notable limitation of existing methods is their reliance on a shared text description among all views of a generated 3D object. Therefore, the inherent diversity in views that a single 3D object can present is generally overlooked, such as varying patterns on different views of a T-shirt, as shown in Figure 1. Consequently, their approaches are inadequate for customizing viewpoints in 3D instances, which restricts their usability to meet tailored or complicated requirements, as shown in Figure 2.

In this work, we propose **DreamView** for customizable text-to-3D generation. We achieve this by constructing a highly customizable and consistent text-to-image model that can synthesize specific image views of an object according to the provided overall and view-specific texts, where the overall text describes the object from the global level and the view-specific text only contains contents appearing in a specific viewpoint, as shown by the samples in Figure 1. In DreamView, the overall text is shared among views to promote **consistent** image generation across viewpoints. Moreover, the view-specific text serves its

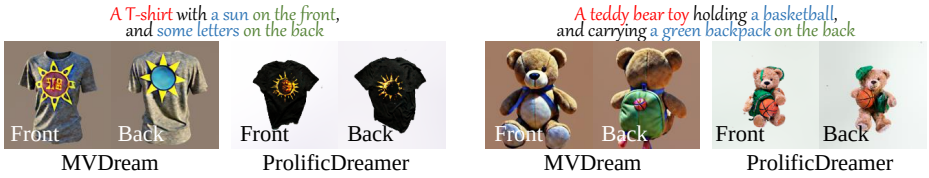

Fig. 2: Text-to-3D generation (front and back views) of recent works, which cannot generate content strictly aligned with texts while may suffer from inconsistent problems.

corresponding view for **customizable** image view generation. To balance these two kinds of text guidance rather than letting one of them dominate, we propose an adaptive text guidance injection module, which acts on each block of the diffusion model and dynamically selects which text should be used in the current block, achieving a dynamic balance between consistency and customization.

DreamView is trained on a large-scale rendered dataset containing multi-view images with paired texts to seek the ability to generate customizable 2D multi-view images from varying viewpoints and maintain instance-level consistency across views. More importantly, this ability can easily transfer to 3D generation via 2D-lifting methods [33,51], facilitating a consistent and customizable text-to-3D synthesis. The dual achievement of viewpoint customization and 3D instance-level consistency marks a significant advancement in the realm of text-to-3D generation, making a modest step in this field.

We present several impressive results in Figure 1, where our DreamView generates high-quality 3D assets that faithfully adhere to the text prompts, showcasing unique appearances defined by each view-specific text. Additionally, we conduct qualitative and quantitative comparisons with other methods regarding text-to-image generation and text-to-3D instance generation. Moreover, we conduct a user study to analyze user preferences. The results collectively highlight the abilities of DreamView to produce customizable and consistent 3D objects.

2 Related Works

Text-to-image generation. With the proposal of several large-scale text-image datasets [38, 39, 42, 48], diffusion models [14, 30, 36, 47] have become increasingly popular in 2D image generation. Diffusion models consist of a forward process that gradually adds noise to data and a reverse (sampling) process that denoises pure noise. By leveraging various conditions such as text and mask, diffusion models can generate high fidelity and diverse content faithful to the user-provided prompts in image editing [4, 18], inpainting [29, 55], *etc*. Recently, diffusion models have shown their potential capabilities in 3D generation [33, 51] due to the strong 2D generation performance. Compared to previous models, ours absorbs the strong viewpoint consistency and customization from a 3D rendered dataset via an adaptive text guidance injection module, making it more suitable for 3D synthesis.

2D-lifting text-to-3D generation. Recently, how to exploit powerful text-to-image generative models [14, 30, 36, 47] to perform text-to-3D synthesis has received considerable attention [23,33,44,51,53]. A notable contribution in this field is the SDS (score distillation sampling) proposed in DreamFusion [33], utilizing diffusion priors as score functions to supervise the optimization of 3D representation. Additionally, Wang *et al*. [51] apply the chain rule on the learned gradients of a diffusion model and back-propagate the score of it through the Jacobian of a differentiable render. Subsequently, a series of efforts were made to improve generation quality [6, 16, 53] and ensure 3D consistency [1, 15, 22, 40, 44, 50]. Despite these methods achieving impressive 3D generation, they still struggle to follow the provided texts strictly, failing to customize the appearance of 3D objects while may also encounter inconsistent problems, as shown in Figure 2. To overcome these challenges, we propose to utilize view-specific text guidance via an adaptive text guidance injection module to introduce customization ability while adaptively maintaining the instance-level consistency, thus empowering a more creative text-to-3D generation.

3 DreamView

Current 2D-lifting text-to-3D generation models predominantly consider a view-shared text prompt, termed 'overall text', to guide generation. However, this simplified pipeline has a significant limitation: encapsulating all appearance attributes of an object within a single text prompt complicates the customization of specific viewpoints. These models facilitate only object-level guidance over the object's generation, lacking the capability to delineate the object's appearance from various viewpoints. For example, as shown in Figure 2, although the text prompts have provided explicit requirements on two views, the current 3D generative models can neither generate letters on the T-shirt nor place the basketball and backpack in the correct position.

In this work, we explore customizing the appearance of the generated 3D object from different views, thereby generating more imaginative 3D assets. We achieve this by making use of the view-specific text, which describes the appearance of an object from a specific viewpoint. Because the view-specific text is not shared among views, viewpoint inconsistency may occur sometimes (see Section 4.2). To overcome this challenge, we propose an adaptive guidance injection module to adaptively collaborate the overall and the view-specific text to achieve a dynamic balance between consistent and customizable generation. The proposed injection module is plugged into a text-to-image model and can be subsequently lifted into 3D generation via score distillation sampling [33]. To distinguish the text-to-image and text-to-3D variants of DreamView, we denote them DreamView-2D (Figure 3) and DreamView-3D (Figure 4), respectively.

3.1 DreamView-2D for Text-to-image Generation

Data preparation. To facilitate view-specific customization, we construct a 3D dataset for training, including paired multi-view images with view-specific

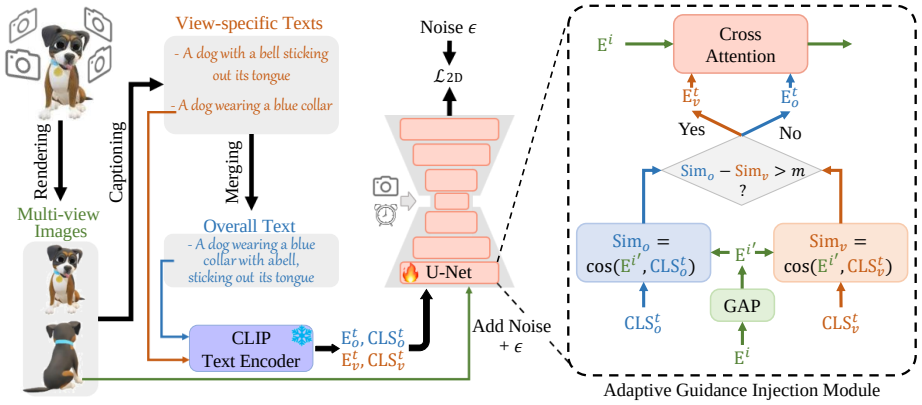

Fig. 3: The overall framework of DreamView-2D. **Left:** the data preparation pipeline and the data flow of DreamView-2D. The raw 3D objects from the Objaverse dataset [8] are first rendered to multi-view images and captioned by BLIP-2 [21]. Finally, the view-specific texts are merged by GPT-4 [32] to form the overall text. With this image-text paired data, DreamView-2D, augmented by an adaptive guidance inject module, is trained to learn a trade-off between 3D consistency and customization. **Right:** the detail of the adaptive guidance injection module working in each U-Net block of the model, which measures the similarity between the image embedding and the two types of text embedding to determine which text guidance should be used in the current U-Net block, thus achieving an adaptive balance between the consistency and customization. A margin hyper-parameter is used in the model to control the balance.

texts and the overall text, as shown in the left side of Figure 3. Our dataset is sourced from the Objaverse dataset [8], which only contains a raw text description for each object and lacks view-specific texts. Thus, it is inadequate to achieve our intention of injecting view-specific texts. To complement the required view-specific text, we utilize advanced large multi-modal models to generate high-quality view-specific and overall text descriptions for each 3D asset in the Objaverse dataset.

Specifically, the construction process is divided into three steps: rendering, captioning, and merging. (1) In the rendering step, each 3D asset is first densely rendered as several multi-view images with 512×512 resolution, and the corresponding camera poses are saved. (2) The captioning stage is carried out via the BLIP-2 [21] model, which is applied to generate a caption for each rendered image, forming the 'view-specific text' for training our DreamView-2D. (3) In the final step, merging, all of the view-specific texts from different views of a 3D object are consolidated by GPT-4 [32], forming the 'overall text description' of this 3D asset. More dataset details are in the *supplement material*.

Adaptive guidance injection module. As mentioned before, we propose an adaptive guidance injection module to balance the guidance from the overall and view-specific texts, whose core idea is determining which guidance dominates the

current diffusion U-Net block [36, 37] and injecting the other guidance into the cross-attention layer as the condition to achieve a dynamic balance, as shown in the right side of Figure 3.

Specifically, we denote the text embeddings of the overall text prompt and view-specific text prompts given by the CLIP text encoder [35] as E_o^t and E_v^t, respectively, where the superscript ^{t} denotes 'text embedding'. Similarly, their global representations, *i.e.*, class tokens are also output by the text encoder of CLIP, denoted as CLS_o^t and CLS_v^t. The image embedding is denoted as E^i. In each U-Net block, we measure the similarity between the image and text embeddings via

$$\mathrm{Sim} = \cos(\mathrm{GAP}(\mathrm{E}^i), \mathrm{CLS}^t), \tag{1}$$

where the 'GAP' denotes the global average pooling operation converting the image feature map to a representation vector, and 'cos' indicates the cosine similarity operator. Applying this equation to CLS_o^t and CLS_v^t, we obtain the image-text similarity of the overall text (Sim_o) and the view-specific text (Sim_v). The magnitude between these two similarity values determines which guidance will be injected into the current block. For example, a larger Sim_o suggests that the current E^i absorbs more overall guidance than view-specific guidance, and thus we inject the view-specific one. This process can be formulated as:

$$\mathrm{Guidance} = \begin{cases} \mathrm{E}_v^t, & \text{if } \mathrm{Sim}_o - \mathrm{Sim}_v > m \\ \mathrm{E}_o^t, & \text{else} \end{cases}, \tag{2}$$

where m is a margin hyper-parameter to control the propensity to consistency or customization. Intuitively, a large margin means more overall guidance will be used, resulting in stronger 3D consistency. Conversely, customization will be dominant with a small margin.

Through the above modeling, we convert the trade-off problem between consistency and customization to adjust the margin, thus achieving an adaptive balance between these two properties in a text-to-image model and subsequently benefiting the text-to-3D generation.

Optimization. Given dataset $\mathcal{D} = \{(x_r, T_r^O)\}_1^N$, where the N is the number of rendered 3D assets in the dataset, the T^O is the overall text description of the 3D asset x. Besides, each 3D asset x is formulated as $\{I_s, T_s, c_s\}_1^M$, representing the rendered image, its text caption given by BLIP-2, and the corresponding rendered camera pose, respectively, and the M denotes the number of rendered views. With these image-text-camera pairs, we can exploit overall and view-specific texts to jointly guide the generative model via the proposed adaptive guidance injection module for seeking consistency and customization capabilities.

In the training phase, the input of the diffusion U-Net can be denoted as $(\mathbf{x}; y, c, t)$, where $\mathbf{x}$, c and t are the image, the camera position, and the sampled time step, respectively. Besides, $y = \{\mathrm{E}_o^t, \mathrm{E}_v^t, \mathrm{CLS}_o^t, \mathrm{CLS}_v^t\}$, and it is prepared for the injection module mentioned in the previous paragraph to provide the condition for each U-Net block according to Equations (1) and (2). Overall, the

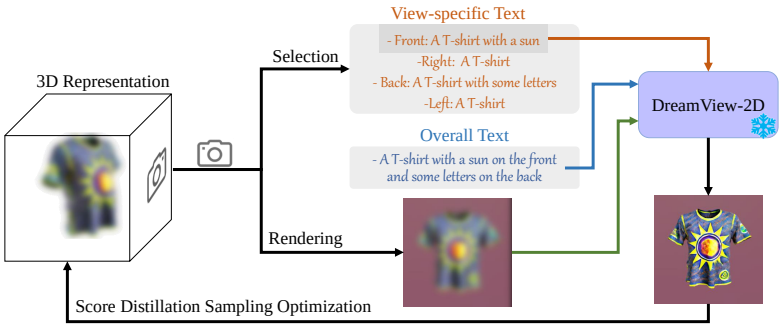

Fig. 4: The overall framework of DreamView-3D, which optimizes a 3D representation via score distillation sampling [33] supervised by DreamView-2D, thus inheriting the consistent and customizable priors.

U-Net parameterized by θ is trained by optimizing the diffusion loss:

$$\mathcal{L}_{2D}(\theta, \mathcal{D}) = \mathbb{E}_{\mathbf{x},y,c,t,\epsilon}[\|\epsilon - \epsilon_\theta(\mathbf{x}_t; y, c, t)\|_2^2], \tag{3}$$

where the ϵ and ϵ_θ are the ground truth and predicted random noise, and $\mathbf{x}_t$ represents the noisy image. In practice, DreamView-2D is trained to generate single or multiple image views for 3D objects with the above loss function. To preserve the identity of the 3D object from multiple viewpoints, an expanded attention mechanism that models the relation among views [3, 27, 44, 50, 54] will be applied when generating multiple views for a 3D object. Through the above optimization, DreamView-2D is expected to learn a trade-off between consistency and customization with the help of the adaptive guidance injection module.

Inference. Unlike the training phase, where each image is paired with a view-specific text and an overall text, users must provide both texts during inference, resulting in a heavy burden of writing text prompts, especially with the increase of generated views. For example, if users hope to generate n image views of a 3D object, they must provide n view-specific texts and one overall text, which is unacceptable. Fortunately, considering the spatial continuity of 3D objects, whose appearance usually does not change intensively within a continuous range of viewpoints, we can roughly divide the object into four views: front, back, left, and right. In this case, only five texts are needed, *i.e.*, one overall text and four view-specific texts, reducing the burdens of users. In addition, one can draw support from large language models to write view-specific texts, and thus, only an overall text is required, which will be discussed in the *supplementary materials*. Furthermore, if users do not require customization, our model can also adopt an overall text to conduct generation.

3.2 DreamView-3D for Text-to-3D Generation

Following typical 2D-lifting text-to-3D generation methods [6, 33, 51, 53], we adopt our DreamView-2D model as a teacher model and distill the priors to supervise 3D generation. We build our DreamView-3D based on the score distillation sampling (SDS) technique proposed by DreamFusion [33], which is supposed to introduce the trade-off ability between consistency and customization in DreamView-2D into 3D generation. To this end, we replace the used text-to-image generation model in DreamFusion with our DreamView-2D. In addition, we also need to introduce view-specific texts together.

Applying view-specific text. Similar to the inference stage discussed in Section 3.1, we divide the azimuth angle from 0-360 degrees into four parts, corresponding to the front, right, left, and back, respectively. Four view-specific texts are associated with four non-intersecting intervals spanning the azimuth angle from 0-360 degrees, respectively. Then, once the camera position for rendering the 3D representation is given, we can determine which view-specific text to use in the current viewpoint. Thus, our DreamView-2D with the adaptive guidance injection module can be easily plugged into 2D-lifting 3D generation techniques.

Overall pipeline. Our DreamView-3D is illustrated in Figure 4. Firstly, a camera position c is sampled, and a 3D representation ϕ is projected to a 2D image $\mathbf{x}$ via differentiable rendering g with the camera position. The camera azimuth is used to select view-specific text embedding, which is then fed into the injection module together with the overall text embedding and their class tokens. The diffusion model θ accepts the rendered image with noise $\mathbf{x}_t$, the text condition $y = \{\mathrm{E}_o^t, \mathrm{E}_v^t, \mathrm{CLS}_o^t, \mathrm{CLS}_v^t\}$, the camera position c, and the sampled time t as inputs, outputting an estimated $\hat{\mathbf{x}}_0$, which is formulated as $\hat{\mathbf{x}}_0 = \epsilon_\theta(\mathbf{x}_t; y, c, t)$. The 3D representation is then optimized by an $\mathbf{x}_0$-reconstruction loss [44]:

$$\mathcal{L}_{3D}(\phi, \mathbf{x} = g(\phi)) = \mathbb{E}_{c,t,\epsilon}[\|\mathbf{x} - \hat{\mathbf{x}}_0\|_2^2], \tag{4}$$

where the gradient of the diffusion model θ is detached to distill its priors into the differentiable 3D representation [33].

Through Equation 4, the 3D representation can inherit the powerful view content customization and instance-level consistency capabilities in our DreamView-2D, thus achieving customizable and consistent text-to-3D generation.

4 Experiments

4.1 Implementation Details

DreamView-2D. We train the DreamView-2D by combining our 3D rendered dataset and the 2D LAION dataset [38] to ensure 3D consistency, customization, and visual quality. The model is initialized by the SD-v2.1 model [36] and is trained on 16 V100 GPUs with a total batch size of 2,048. The learning rate

Table 1: Comparisons on image synthesis quality. The metrics are evaluated on the validation set of our rendered dataset. The numbers separated by the '/' in the line of SD-v2.1 and MVDream denote using overall or view text for generation, respectively. DreamView adaptively selects which text should be used.

Method	CLIP Score (↑)			Inception Score (↑)
	Overall Text	View Text	GT Image	
Ground Truth	34.5	34.8	1.00	10.3
SD-v2.1 [36]	29.2/28.3	26.8/29.4	0.48/0.53	15.3/15.6
MVDream [44]	31.3/29.9	28.6/30.1	0.65/0.67	13.2/13.1
DreamView-2D	31.1	32.1	0.73	14.5

is set to $1e^{-4}$. For each 3D object, we randomly select four orthogonal image views and resize them to 256×256 to train the model, and the corresponding camera positions are normalized into a sphere. Regarding the inference, we adopt the DDIM [47] sampler with 50 sampling steps and a classifier-free guidance (CFG) scale of 7.5 for generating four image views simultaneously. The margin is randomly sampled from -0.1 to 0.1 during training and fixed at -0.025 during inference. Moreover, we also evaluate the model with our validation set, where we measure the CLIP-text score and CLIP-image score between the generated images and their corresponding texts and ground truth images, as well as the inception score for quality judgment.

DreamView-3D. We implement DreamView-3D by building upon threestudio [13] and substituting Stable Diffusion [36] in DreamFusion [33] with our DreamView-2D for text-to-3D generation. To represent the 3D content, we employ the implicit-volume approach [2] and optimize it for 10,000 steps using the AdamW optimizer [28] with a learning rate of 0.01. During optimization, SDS's maximum and minimum time steps are linearly annealed. Initially, the rendering resolution is set to 64×64 for the first 5,000 steps and is then increased to 256×256. After 5,000 steps, we enable soft shading [23]. In most cases, we divide the azimuth angle of the camera position into four intervals: $[10, 170]$ is the front side, $(170, 190)$ is the right side, $[190, 350]$ is the back side, and the remaining part is the left side. The margin is set as -0.025. For further details, including the view-specific prompts, please refer to the *supplement material*.

4.2 Text-to-Image Generation

In this section, we evaluate the image generation capabilities of DreamView-2D and conduct ablation studies on the hyper-parameter balancing the customization and consistency ability, *i.e.*, the margin.

Quantitative comparison with other methods. In Section 4.2, we compare the image synthesis quality of DreamView-2D with other text-to-image generative models using a validation set of 1,000 objects from our dataset. We generate

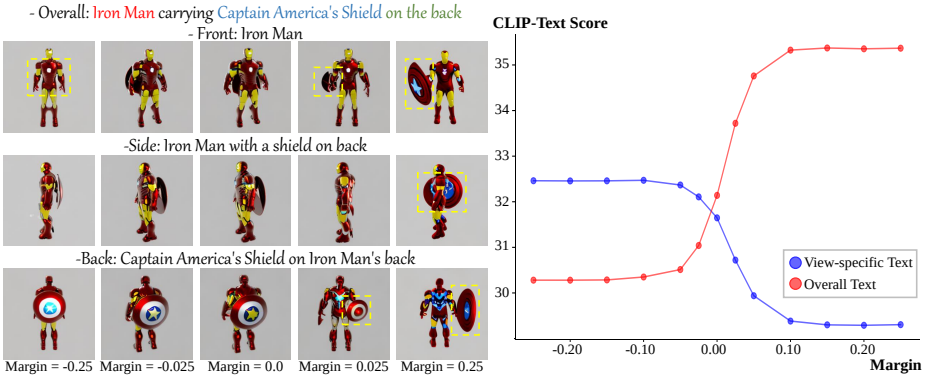

Fig. 5: Left: qualitative text-to-image generation results of DreamView-2D with different margins. **Right:** quantitative evaluation results (CLIP image-text score) on the validation set with the margins change. As the margin gradually increases, customization will weaken while consistency will increase.

four image views for each object and show the average result of all generated images. We mainly focus on four metrics: the CLIP image-text score of the overall and view-specific text, the CLIP image-image score between the generated and the ground truth images, and the inception score. Intuitively, the CLIP score with overall text can roughly represent consistency as it is shared among views, and the CLIP score with view-specific text reflects the customization capability since it is specified in a specific view. The CLIP image-image score comprehensively reflects consistency and customization of the generated image, as the ground truth image is rendered from the 3D object. Besides, the inception score focuses on the visual quality of the image.

The results in Section 4.2 show that SD-v2.1 [36] achieves the best inception score and the worst CLIP score. It is only trained with 2D data and thus cannot generalize to 3D multi-view generation. Moreover, MVDream slightly outperforms ours in the overall text (+0.2) when it generates images according to the overall text. In contrast, its CLIP score with view text is significantly inferior to ours (-3.5), indicating a lack of customization capability. When the view text is applied to guide the image generation, the CLIP score with the view text of MVDream improved (+1.5) with certain consistency losses (-1.4) but still lower than ours. Regarding the CLIP image-image score, DreamView outperforms others clearly, indicating its good balance in consistency and customization. Lastly, the inception score of DreamView is slightly worse than SD-v2.1, but considering its consistency and customization, these losses are acceptable.

Studies on the margin. The margin used in DreamView is designed to balance the guidance from the overall and view-specific text. We conduct qualitative and quantitative ablation studies in Figure 5 to validate this property.

From the left figure, with the margin increases, one can observe a trend of being more consistent. Specifically, as shown by yellow boxes, the view-specific

text for the front view does not specify 'shield', resulting in the loss of the shield when the margin is small, which reflects stronger customization while weaker consistency. When the margin is large, *e.g.*, 0.025 and 0.25, the shield no longer appears on Iron Man's back but on his arm, especially in the front view, demonstrating a better consistency but lacking customization. Besides, setting the margin to -0.025 and 0.0 shows a balanced trade-off between the two properties. By default, we set the margin to be -0.025 during inference.

Furthermore, the quantitative results on the right side of Figure 5 also suggest that the consistency improves and the customization declines with the margin increase, reflected by the CLIP score with the overall text gradually rising while the CLIP score with the view text decreasing.

4.3 Text-to-3D Generation

In this section, we present text-to-3D generation results with DreamView-3D and compare them with other methods, where all methods are implemented based on the open-source threestudio library [13].

Qualitative results. In Figure 6, we show several qualitative results of our proposed DreamView-3D, where all contents specified in the text are accurately presented at their expected location, *e.g.*, the rocket on the bulldog's back, the fire on the Pikachu's tail. Except for customizing from the front and back views, DreamView can also manipulate content from the left and right views, *i.e.*, the hammer and shield on Captain's hand. Moreover, DreamView-3D can not only customize character-like objects but also works with scene objects, as shown in the last sample of Figure 6, where a castle with trees and a red car is presented. In addition, our method not only enjoys an impressive customizable property but also maintains instance-level consistency, where none of the results show the multi-face and multi-foot problems. Overall, our DreamView-3D enables view customization while ensuring 3D consistency, thereby paving the way for generating more imaginative 3D assets.

Furthermore, we compare DreamView-3D with other methods on text-to-3D synthesis requiring customization in Figure 7 (top-left), where we show the generated objects' front, back, and side views. According to the results, all the compared methods successfully generate the main concepts, such as the penguin, the MAC book, and the astronaut. However, in the case of generating the penguin, except for our DreamView-3D, other methods either ignore the objects (the crossbody bag and the scarf) or fail to place the bag in the expected location. Besides, other methods suffer from 3D inconsistency to some extent. For example, they always tend to generate the canonical view of the object, *i.e.*, the face of the penguin and the screen of the MAC book. Despite MVDream being designed for 3D consistency, we still find that in the second example, the back of the MAC book shows the Superman logo instead of the Apple one. Moreover, in the last case, DreamFusion and MVDream fail to generate the horse, and ProlificDreamer only shows the horse in the side view. In comparison, ours can correctly generate all described contents and place each in the expected position.

Fig. 6: Text-to-3D generation of DreamView-3D. The first three columns present the rendered normal maps, and the rest are RGB images. We highlight the subject, object, and the position of the object in red, blue, and green. Only the overall texts are shown in the figure, and detailed view-specific texts are in *supplement material*.

On the other hand, in the bottom side of Figure 7, we compare DreamView-3D with other methods on general prompts, *i.e.*, no customization is required. The results demonstrate that our model can also work with general prompts and generate highly consistent 3D objects with high fidelity. Note that although ProlificDreamer [53] generates more complex and rich details, it severely suffers from the inconsistency problem, *e.g.*, generating multiple faces for the chimpanzee and multiple arms for the bumblebee. On the last row, despite the specification of 'full-body', MVDream [44] still generates the car form of the bumblebee, which is not as user's expected. The above results show that our model can still work correctly even if only the overall text is provided.

User study. We conducted a user study to collect user preferences and to provide an intuitive and comprehensive evaluation. We collected 30 text prompts

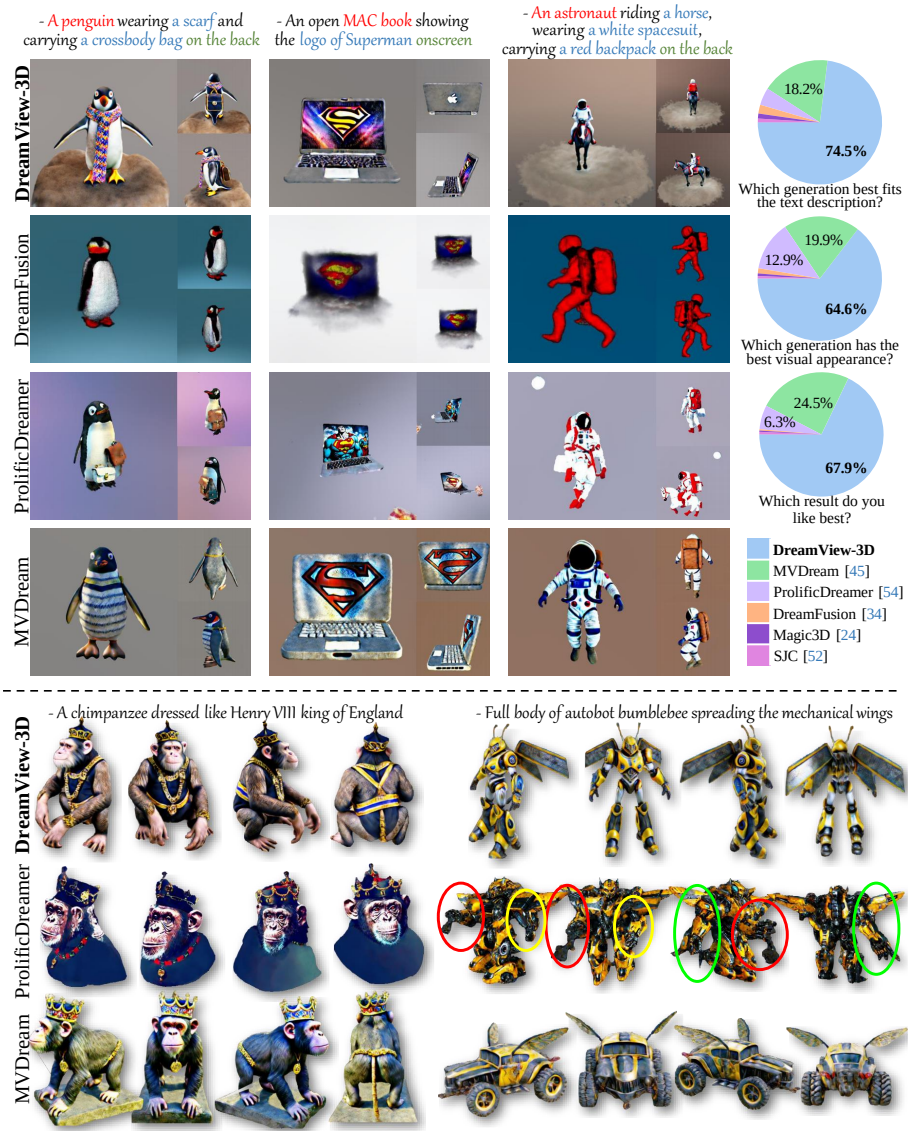

Fig. 7: Top-left: Comparisons with other methods on text-to-3D generation with customization requirements. Three views are shown, including the object's front, back, and side views. Other methods either overlook certain contents in the text prompts or fail to generate specific content in the expected position. Besides, some methods suffer from the 3D inconsistency problem. More comparisons are in the *supplement material*. **Top-right**: The results of the user study. **Bottom:** Comparisons with other methods on text-to-3D generation with general text prompts (without viewpoint customization). The circles with three different colors show that the bumblebee has three arms.

and applied 6 different methods to generate 180 3D objects in total. Among these 30 text prompts, 15 of them have customization requirements, thus including both the overall text and the view-specific text, and the other 15 are general text prompts. The compared methods include DreamFusion [33], SJC [51], Magic3D [23], ProlificDreamer [53], MVDream [44], and ours. Each group of generation results is presented in the form of {overall text, view-specific text (if used), results of 6 methods in random order}. Each user is asked to answer three choice questions for each group of generations: (1) Which result best fits the text description? (2) Which result has the best visual appearance? (3) Which result do you like best? The evaluation criteria for these three questions are: (1) Are all the concepts in the text description presented correctly in the results? (2) Is the geometry complete or includes multi-face, multi-foot, or noise? Moreover, question (3) is subjective.

In our user study, 35 participants with varying expertise and aesthetic views are involved, 25 are online volunteers, and the remaining 10 are researchers of related fields, *e.g.*, 3D modeling, computer graphics, *etc*. We received 1,050 feedbacks in total, and the results are shown in Figure 7 (top-right). According to the results, 74.5% of users choose our DreamView-3D to be more capable of generating 3D assets consistent with text descriptions, which is much higher than other methods. Regarding the quality of visual appearance, users choosing MVDream and ProlificDreamer increased, but it is still less than choosing ours. For the last question, 67.9% users prefer ours over other methods, demonstrating the superior quality of our DreamView-3D.

Generation speed. DreamView-3D takes roughly 55 minutes to generate a 3D asset on a single A100 GPU. Under the same experimental environment, DreamFusion [33], Magic3D [23], and SJC [51] take around 30 minutes, MVDream [44] takes about 50 minutes, and ProlificDreamer [53] takes $\sim$180 minutes.

5 Limitations and Conclusions

Limitations. The generated full-body characters' faces may be blurry and lose details, as shown by the third and fourth rows in Figure 6, probably caused by using low-resolution training images. Training a higher-resolution model may address this problem but requires more training resources and time. (2) Besides, despite supposing customization, DreamView requires texts from different viewpoints to describe the same instance. Otherwise, the generation will fail, *e.g.*, generating a dog from the front while a monkey from the back.

Conclusions. In this work, we introduce DreamView, a text-to-image model that can be lifted to 3D object generation and enables viewpoint customization while maintaining instance-level consistency by collaborating view-specific text and overall text via an adaptive guidance injection module. Extensive quantitative and qualitative results demonstrate the advancement of our method in text-to-3D generation, where our DreamView provides a highly versatile and personalized avenue for producing consistent and customizable 3D assets.

Acknowledgement. This work was partially supported by the Guangdong NSF Project (No. 2023B1515040025, 2020B1515120085) and NSFC (U21A20471, U1911401), and the Guangdong Basic and Applied Basic Research Foundation (2023A1515012974). This work was also supported by Alibaba Group through the Alibaba Innovative Research Program. We also thank Kun-Yu Lin, Yi-Xing Peng, and Yu-Ming Tang for their helpful discussions.

References

1. Armandpour, M., Zheng, H., Sadeghian, A., Sadeghian, A., Zhou, M.: Re-imagine the negative prompt algorithm: Transform 2d diffusion into 3d, alleviate janus problem and beyond. arXiv (2023)
2. Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In: ICCV (2021)
3. Blattmann, A., Rombach, R., Ling, H., Dockhorn, T., Kim, S.W., Fidler, S., Kreis, K.: Align your latents: High-resolution video synthesis with latent diffusion models. In: CVPR (2023)
4. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. In: CVPR (2023)
5. Chang, K.H., Cheng, C.Y., Luo, J., Murata, S., Nourbakhsh, M., Tsuji, Y.: Building-gan: Graph-conditioned architectural volumetric design generation. In: ICCV (2021)
6. Chen, R., Chen, Y., Jiao, N., Jia, K.: Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In: ICCV (2023)
7. Choi, H., Crump, C., Duriez, C., Elmquist, A., Hager, G., Han, D., Hearl, F., Hodgins, J., Jain, A., Leve, F., et al.: On the use of simulation in robotics: Opportunities, challenges, and suggestions for moving forward. PNAS (2021)
8. 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 (2023)
9. Gao, Y., Wang, Z., Zheng, W.S., Xie, C., Zhou, Y.: Sculpting holistic 3d representation in contrastive language-image-3d pre-training. In: CVPR (2024)
10. Gao, Y., Yang, L., Huang, Y., Xie, S., Li, S., Zheng, W.S.: Acrofod: An adaptive method for cross-domain few-shot object detection. In: ECCV (2022)
11. Ge, Y., Xu, J., Zhao, B.N., Joshi, N., Itti, L., Vineet, V.: Beyond generation: Harnessing text to image models for object detection and segmentation. arXiv (2023)
12. Ge, Y., Yu, H.X., Zhao, C., Guo, Y., Huang, X., Ren, L., Itti, L., Wu, J.: 3d copy-paste: Physically plausible object insertion for monocular 3d detection. In: NeurIPS (2024)
13. Guo, Y.C., Liu, Y.T., Shao, R., Laforte, C., Voleti, V., Luo, G., Chen, C.H., Zou, Z.X., Wang, C., Cao, Y.P., Zhang, S.H.: threestudio: A unified framework for 3d content generation. `https://github.com/threestudio-project/threestudio` (2023)
14. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: NeurIPS (2020)
15. Hong, S., Ahn, D., Kim, S.: Debiasing scores and prompts of 2d diffusion for robust text-to-3d generation. In: CVPR-W (2023)

16. Huang, Y., Wang, J., Shi, Y., Qi, X., Zha, Z.J., Zhang, L.: Dreamtime: An improved optimization strategy for text-to-3d content creation. In: ICLR (2024)
17. Jun, H., Nichol, A.: Shap-e: Generating conditional 3d implicit functions. arXiv (2023)
18. Kawar, B., Zada, S., Lang, O., Tov, O., Chang, H., Dekel, T., Mosseri, I., Irani, M.: Imagic: Text-based real image editing with diffusion models. In: CVPR (2023)
19. Li, C., Zhang, C., Waghwase, A., Lee, L.H., Rameau, F., Yang, Y., Bae, S.H., Hong, C.S.: Generative ai meets 3d: A survey on text-to-3d in aigc era. arXiv (2023)
20. Li, C., Zhang, R., Wong, J., Gokmen, C., Srivastava, S., Martín-Martín, R., Wang, C., Levine, G., Lingelbach, M., Sun, J., et al.: Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In: CoRL (2023)
21. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In: ICML (2023)
22. Li, W., Chen, R., Chen, X., Tan, P.: Sweetdreamer: Aligning geometric priors in 2d diffusion for consistent text-to-3d. In: ICLR (2024)
23. Lin, C.H., Gao, J., Tang, L., Takikawa, T., Zeng, X., Huang, X., Kreis, K., Fidler, S., Liu, M.Y., Lin, T.Y.: Magic3d: High-resolution text-to-3d content creation. In: CVPR (2023)
24. Lin, K.Y., Du, J.R., Gao, Y., Zhou, J., Zheng, W.S.: Diversifying spatial-temporal perception for video domain generalization. In: NeurIPS (2024)
25. Liu, M., Shi, R., Kuang, K., Zhu, Y., Li, X., Han, S., Cai, H., Porikli, F., Su, H.: Openshape: Scaling up 3d shape representation towards open-world understanding. In: NeurIPS (2024)
26. 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 (2023)
27. Liu, Y., Lin, C., Zeng, Z., Long, X., Liu, L., Komura, T., Wang, W.: Syncdreamer: Generating multiview-consistent images from a single-view image. In: ICLR (2024)
28. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2018)
29. Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., Van Gool, L.: Repaint: Inpainting using denoising diffusion probabilistic models. In: CVPR (2022)
30. Meng, C., Rombach, R., Gao, R., Kingma, D., Ermon, S., Ho, J., Salimans, T.: On distillation of guided diffusion models. In: CVPR (2023)
31. Nichol, A., Jun, H., Dhariwal, P., Mishkin, P., Chen, M.: Point-e: A system for generating 3d point clouds from complex prompts. arXiv (2022)
32. OpenAI: Gpt-4 technical report. arXiv (2023)
33. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion. In: ICLR (2023)
34. 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. In: ICLR (2024)
35. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML (2021)
36. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR (2022)
37. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: MICCAI (2015)
38. Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Wightman, R., Cherti, M., Coombes, T., Katta, A., Mullis, C., Wortsman, M., et al.: Laion-5b: An open large-scale dataset for training next generation image-text models. In: NeurIPS (2022)

39. Schuhmann, C., Vencu, R., Beaumont, R., Kaczmarczyk, R., Mullis, C., Katta, A., Coombes, T., Jitsev, J., Komatsuzaki, A.: Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. arXiv (2021)
40. Seo, J., Jang, W., Kwak, M.S., Ko, J., Kim, H., Kim, J., Kim, J.H., Lee, J., Kim, S.: Let 2d diffusion model know 3d-consistency for robust text-to-3d generation. In: ICLR (2024)
41. Shao, R., Sun, J., Peng, C., Zheng, Z., Zhou, B., Zhang, H., Liu, Y.: Control4d: Dynamic portrait editing by learning 4d gan from 2d diffusion-based editor. arXiv (2023)
42. Sharma, P., Ding, N., Goodman, S., Soricut, R.: Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In: ACL (2018)
43. Shen, T., Gao, J., Yin, K., Liu, M.Y., Fidler, S.: Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. In: NeurIPS (2021)
44. Shi, Y., Wang, P., Ye, J., Mai, L., Li, K., Yang, X.: Mvdream: Multi-view diffusion for 3d generation. In: ICLR (2024)
45. Shi, Z., Peng, S., Xu, Y., Geiger, A., Liao, Y., Shen, Y.: Deep generative models on 3d representations: A survey. arXiv (2022)
46. Singer, U., Sheynin, S., Polyak, A., Ashual, O., Makarov, I., Kokkinos, F., Goyal, N., Vedaldi, A., Parikh, D., Johnson, J., et al.: Text-to-4d dynamic scene generation. arXiv (2023)
47. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: ICML (2015)
48. Srinivasan, K., Raman, K., Chen, J., Bendersky, M., Najork, M.: Wit: Wikipedia-based image text dataset for multimodal multilingual machine learning. In: ACM SIGIR (2021)
49. Taheri, O., Choutas, V., Black, M.J., Tzionas, D.: Goal: Generating 4d whole-body motion for hand-object grasping. In: CVPR (2022)
50. Tang, S., Zhang, F., Chen, J., Wang, P., Yasutaka, F.: Mvdiffusion: Enabling holistic multi-view image generation with correspondence-aware diffusion. In: NeurIPS (2023)
51. Wang, H., Du, X., Li, J., Yeh, R.A., Shakhnarovich, G.: Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In: CVPR (2023)
52. Wang, P., Liu, L., Liu, Y., Theobalt, C., Komura, T., Wang, W.: Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In: NeurIPS (2021)
53. Wang, Z., Lu, C., Wang, Y., Bao, F., Li, C., Su, H., Zhu, J.: Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. In: NeurIPS (2023)
54. Wu, J.Z., Ge, Y., Wang, X., Lei, S.W., Gu, Y., Shi, Y., Hsu, W., Shan, Y., Qie, X., Shou, M.Z.: Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. In: ICCV (2023)
55. Xie, S., Zhang, Z., Lin, Z., Hinz, T., Zhang, K.: Smartbrush: Text and shape guided object inpainting with diffusion model. In: CVPR (2023)
56. Xue, L., Gao, M., Xing, C., Martín-Martín, R., Wu, J., Xiong, C., Xu, R., Niebles, J.C., Savarese, S.: Ulip: Learning a unified representation of language, images, and point clouds for 3d understanding. In: CVPR (2023)
57. Xue, L., Yu, N., Zhang, S., Li, J., Martín-Martín, R., Wu, J., Xiong, C., Xu, R., Niebles, J.C., Savarese, S.: Ulip-2: Towards scalable multimodal pre-training for 3d understanding. In: CVPR (2023)

58. Yin, X., Wonka, P., Razdan, A.: Generating 3d building models from architectural drawings: A survey. IEEE COMPUT GRAPH (2008)
59. Zhang, Z., Cao, S., Wang, Y.X.: Tamm: Triadapter multi-modal learning for 3d shape understanding. In: CVPR (2024)
60. Zhou, J., Liang, J., Lin, K.Y., Yang, J., Zheng, W.S.: Actionhub: A large-scale action video description dataset for zero-shot action recognition. arXiv (2024)
61. Zhou, J., Wang, J., Ma, B., Liu, Y.S., Huang, T., Wang, X.: Uni3d: Exploring unified 3d representation at scale. In: ICLR (2024)