https://doi.org/10.1007/s11704-025-40774-x

RESEARCH ARTICLE

HexaDream: hexaview prior and constraint for text to 3D creation

Zhi-Chao ZHANG[1*], Hui CHEN[1*], Jin-Sheng DENG[2,1]✉, Ming XU[1], Zheng-Bin PANG[1]

1. College of Computer Science and Technology, National University of Defense Technology, Changsha 410000, China
2. Military Intelligent Research Institute, Academy of Military Sciences, Beijing 100091, China

Received July 28, 2024; accepted February 17, 2025

E-mail: djs20201030@163.com. * These authors contributed equally to this work.

Abstract

The burgeoning field of text-to-3D synthesis offers transformative potential in diverse domains such as computer-aided design, gaming, virtual reality, and artistic creation. However, the generation struggles with issues of inconsistency and low resolution, primarily due to the lack of critical visual clues like views and attributes. Furthermore, random constraint in rendering may impair model inference, leading to the Janus problem. In response to these challenges, we introduce HexaDream to produce high-quality 3D content. Hexaview Generation Diffusion Model is designed to merge object types, attributes, and view-specific text into unified latent space. Besides, the feature aggregation attention significantly enhances the detail and consistency of the generated output by mapping point features from orthogonal view into the 3D domain. Another innovation is the Dynamic-weighted HexaConstraint. This module employs a projection matrix to generate projected views and calculates the differential loss between these projections and the hexaviews, ensuring high fidelity. Our comparative experiments show that HexaDream achieves improvements of 8% in CLIP-R, 12% in Keypart Fidelity, and especially 20.6% in Multihead Alleviation compared with existing methods respectively.

Keywords

text to 3D; janus problem; HexaConstraint

1 Introduction

In the dynamic sphere of AI-Generated Creative Content (AIGC), the creation of three-dimensional objects from textual descriptions is gaining significant attention. Text-to-3D is notably impacting fields such as computer-aided design (CAD), gaming, virtual reality, and artistic creation. Despite the huge potential of 3D synthesis task, the process of converting text to 3D content is fraught with challenges, including inconsistencies and low resolution. The reason is due to the absence of critical visual details like perspectives and attributes. Prevailing methodologies, exemplified by DreamFusion [1], leverage frozen text-to-image models for generating images that align with the textual input. However, these methods fall short in capturing a comprehensive array of viewpoints, leading to issues such as distorted geometrical representations. To address these shortcomings, innovative solutions have been developed, such as RealFusion [2] and HoloDiffusion [3], in conjunction with other notable strategies [4,5], which aim to enrich the visual data inputs [6,7]. Nevertheless, these approaches encounter difficulties in accurately rendering complex 3D structures due to missing viewpoints. Facing the above challenges, in our paper, a novel viewpoint-attribute-sensitive object generation diffusion model are trained with a comprehensive image-text dataset derived from large-scale 3D collections. This model is adept at producing six orthogonal views from a single text input, thereby surpassing the limitations inherent in previous methodologies.

Recent advancements have also seen the integration of 3D attention mechanisms to ensure depth information accuracy. Models such as MVDream, MVDiffusion, and FastMETRO [4,8,9] are pivotal in augmenting geometric details, thus optimizing the generation of 3D objects from text through a systematic two-stage process. Similarly, the 3DFuse model [10] utilizes a consistency injection module, incorporating sparse depth injectors and semantic codes to synthesize detailed depth maps, thereby integrating 3D perceptions from external priors into the synthesis process. Central to our innovation is the Hexaview Generation Diffusion Model, adept at creating multidimensional perspectives, which are subsequently refined by the Feature Aggregation Module. This module skillfully merges 2D features into a cohesive 3D entity, enhancing the overall representation of 3D objects generated from textual descriptions.

In addressing high-quality text-to-3D content creation, recent innovations, including Magic123 [11] and ProlificDreamer [12], have introduced groundbreaking approaches. Magic123 employs a novel 2D and 3D joint-weighted loss supervision technique to generate detailed 3D meshes from non-specific image sources.

ProlificDreamer, on the other hand, introduces the Variational Score Distillation (VSD) algorithm, applying Bayesian modeling and variational inference to reconceptualize the text-to-3D transformation process. Despite these advances, challenges remain particularly in the domain of supervised learning and the accurate calculation of loss functions between rendering projections and generating orthogonal views. These challenges manifest as geometric distortions as Fig. 1 shows, inconsistencies in texture and color, and detail inaccuracies, ultimately impacting the uniformity across different views of the generated 3D objects. To mitigate these issues, our study proposes the Dynamic-weighted HexaConstraint Module, employing a sophisticated projection matrix to calculate differential loss between various projected views and hexaviews.

Above all, our paper presents HexaDream, an innovative framework composed of three essential modules. Distinct from DreamFusion [1], HexaDream introduces several key advancements: the Hexaview Generation Diffusion Model, the Feature Aggregation Attention Mechanism, and the Dynamic-weighted HexaConstraint Module. These modules collectively enable the generation of semantically and visually coherent views from a unique triple-text format, enhance spatial correlations between images from multiple perspectives, and address data imbalance in 3D object generation. Our extensive experimental analysis has yielded significant improvements in key metrics such as multihead alleviation, CLIP-R-Precision, and keypart fidelity, demonstrating the effectiveness of HexaDream in resolving common challenges like the multi-head problem and enhancing the consistency and fidelity of 3D objects generated from text.

- The hexaview generation diffusion model, specifically tailored for creating coherent views, effectively reduces multiple-head occurrences in the images.
- The feature aggregation attention mechanism strengthens the spatial interplay between multi-view images, ensuring a unified representation.
- The dynamic-weighted HexaConstraint module enhances the overall fidelity of the models by addressing data imbalances in the generation process.
- Our comparative experiments show that HexaDream achieves improvements of 8% in CLIP-R, 12% in Keypart Fidelity, and especially 20.6% in Multihead Alleviation compared to existing methods respectively.

Fig. 1 Multihead and multiface problem

2 Relevant studies

2.1 Text-to-image generation

Recent advancements in text-to-image generation have been driven by pretrained models like Imagen [13], CLIP [14], GLIDE [15], DALL-E2 [16], Parti [17], and CogView2 [18]. These models, particularly Stable Diffusion Models [19], focus on high-resolution single-view generation. DreamBooth [20] further extends this by creating personalized text-to-image diffusion models. The challenge of generating multiple perspectives of a single object while maintaining its structure and appearance has led to innovations like One-2-3-45++ [21], which generates 360-degree panoramic views, and models like Zero123 [22] and Zero123++ [23], which utilize geometric priors for multi-view image generation. Our paper introduces a viewpoint-attribute-sensitive object generation diffusion model trained on a large-scale 3D dataset, capable of generating six orthogonal images of an object from various perspectives.

2.2 3D reconstruction with neural fields

Significant advances in 3D reconstruction have been made using implicit models [24,25], voxel grids [26,27], and point clouds [28–30]. NeRF [31] is a key development in implicit models, employing differentiable rendering and neural networks to reconstruct 3D scenes from images. To address the challenges of NeRF's requirement for numerous viewpoints and parameters, research has focused on reducing input data, such as images without specified poses [32,33] or sparse views [34,35]. However, limited data points can complicate the optimization process.

Another research direction in NeRF involves novel view prediction, as seen in Niemeyer's approach [36], which pretrains scenes to predict new views from limited images. However, this method is constrained by the finite set of scene categories learned from training data. NeRDi [37] introduces a NeRF synthesis framework using a single image without 3D supervision, employing diffusion models for 2D prior knowledge and ensuring semantic and visual consistency in new views. Our paper leverages images from NeRF renderings, supervised against multi-view images generated at orthogonal angles by a frozen model. This method takes inspiration from various loss functions from the cited works to enhance object generation's efficiency and quality.

2.3 Text-to-3D generation

Progress in text-to-3D generation has been propelled by the integration of diffusion models and neural radiance fields. Initial efforts like CLIPMesh [38], Text2Mesh [39], and Dreamfield [40] utilize CLIP models for supervision but face limitations in shape and texture quality. DreamFusion [1] introduces a method using score distillation sampling and a customized NeRF variant [41] for diverse 3D generation from text prompts. Magic3D [42] further develops this into a two-stage coarse-to-fine process, enhancing text-to-3D synthesis resolution.

The quality of 3D content generated through diffusion models remains unstable, often leading to issues like the Janus problem and inconsistent object parts. Approaches like word vector constraint [43,44] and multi-perspective fusion during diffusion [2–4,23] have

been explored to address these challenges, but issues with the training data and constraint persist. While progress has been made in generating simple geometric structures, complexities in creating realistic, high-fidelity textures remain.

Recent research focuses on high-fidelity, stylized text-to-3D content. Magic123 [11] adopts a two-stage approach, combining 2D and 3D priors, while Edit-DiffNeRF [45] integrates NeRF [31] with diffusion models for better cross-view consistency. OmniObject3D [46] introduces a comprehensive 3D dataset for real-world object generation. Our model integrates multi-view training, an attention mechanism with weight adjustment, and orthogonal projection to optimize perspective richness, focus on details, and enhance generation consistency in text-to-3D tasks.

3 Implementation

In this section, our proposed HexaDream framework are depicted in Fig. 2, including Hexaview generation diffusion model in Subsection 3.1, feature aggregation module in Subsection 3.2, and HexaConstraint module in Subsection 3.3. Figure 2(a) shows the training details of the orthogonal hexaview diffusion model while Fig. 2(b) shows the whole framework of HexaDream.

3.1 Hexaview generation diffusion model

The denoising diffusion probabilistic model is served as a generative model, capable of learning the noise distribution in training samples. In recent years, its application for text to image synthesis has yielded substantial enhancements in terms of quality and efficiency. Thus, we build our method upon the recent Latent Diffusion Model (LDM) [19] for hexaview generation.

Regarding the reasons for choosing the LDM, we choose it primarily for its efficient computation and storage in latent space, which significantly reduces computational complexity and storage requirements, making high-resolution image generation more efficient. Additionally, operations in latent space help capture higher-level semantic information, aiding in the generation of more detailed and higher-quality multi-view images. LDM's compatibility with pre-trained models like Stable Diffusion also facilitates fine-tuning on large-scale datasets.

From the given text, it seems easy to extract semantic information of the rendered image, such as object types, colors, and other attributes. However, it is challenging to cover all details. For more prior knowledge, we modify RealFusion [2] by adding view prompts as guidance additionally.

Hexaview Generation Diffusion Model is primarily trained into three steps. 1) Decompose word vectors (like entity and attribute) and matching images from LAION datasets with the features; 2) Project the 3D dataset into six orthogonal views and annotate the view text; 3) Encode the above images 1) and 2) into latent space and train a hexaview generation diffusion model.

Firstly, the input text is embedded by a unified text-to-text transformer [47]. Then the CNNs extracts the features of entity and attribute with text vectors. Assisted by LAION dataset, it could obtain the set of images I_0 and attribute images I_* corresponding to the text features. Furthermore, to obtain view clues, we preprocess

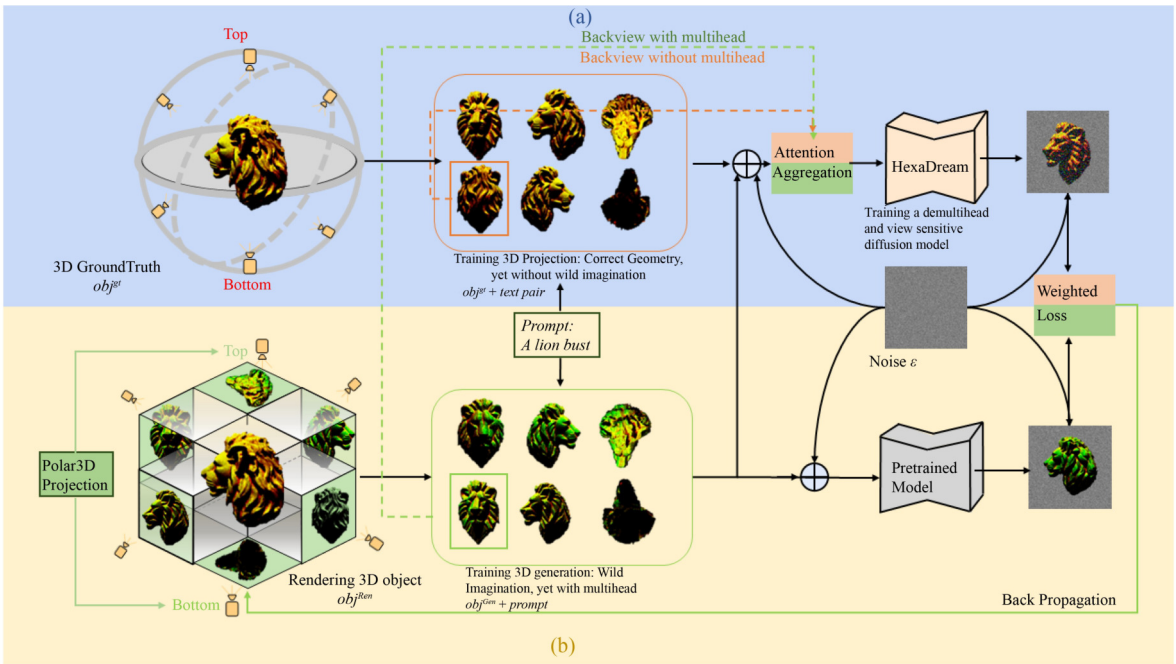

Fig. 2 The HexaDream Framework. Our approach commences with the training of an orthogonal-hexaview generation diffusion model, utilizing a tripartite semantic feature as the conditional input. This model is adept at translating textual prompts into hexaviews. Subsequently, feature vectors, synthesized through an attention-based aggregation mechanism, are sent to the Neural Radiance Field (NeRF) model to construct a 3D representation. The final step involves the refinement of the 3D scene representation. This is achieved by calculating the HexaConstraint loss, which assesses the congruency between the 3D object projections and the orthogonal-hexaviews generated by our pre-trained model, thereby optimizing the overall scene depiction. (a) Demulti-Head Training; (b) HexaProjection and Constraints

the LAION dataset with six orthogonal projections. This stage is beneficial for learning the whole representation of 3D object structure. We embed the above images (including entity, attribute, and view images) into the latent space p using a pre-trained image encoder [19] $\mathcal{E}(\mathbf{x}) = \mathbf{z}$. Subsequently, the image is recovered through decoder $\mathcal{D}(\mathcal{E}(\mathbf{x})) = \mathbf{x}$. By minimizing Eq. (1) in latent space, we train the diffusion model for generating multi-view images:

$$(p_0, p_*, p_v) = \arg\min \mathbb{E}_{\mathbf{z}\sim\mathcal{E}(\mathbf{x}_0),\mathbf{p},\epsilon\sim\mathcal{N}(0,1),t} \left[\|\epsilon - \epsilon_\theta(\mathbf{z}_t, t, c_\theta(\mathbf{p}))\|_2^2\right], \quad (1)$$

where p_0 represents the vector encoding of object type images (e.g., cat, dog), p_* represents object attribute encoding (e.g., color, attributes), and p_v represents object view encoding (e.g., front view, left view). By minimizing the function in Eq. (2) iteratively, we train new view encodings in latent space.

$$\mathbb{E}_{\mathbf{z}\sim\mathcal{E}(\mathbf{x}),p,\epsilon\sim\mathcal{N}(0,1),t} \left[\|\epsilon - \epsilon_\theta(\mathbf{z}_t, t, c_\theta(\mathbf{p}))\|_2^2\right], \quad (2)$$

where t is the temporal parameter in latent space, $\epsilon \sim \mathcal{N}(0,1)$ is a random noise sample, z_t is the latent space encoding z with noise for time t, ϵ_θ is the denoising network, and $c_\theta(p)$ is the new view encoding, aiming to information embedding such as object type, attributes, and views from images into the latent space.

3.2 Feature aggregation attention mechanism

We draw inspiration from the principles of view matching in multi-view geometry to establish pixel-level or feature-level correspondences between different views. This facilitates the matching and integration of three-dimensional spatial features, aiding in identifying the projections of the same object in different views and associating them with a common three-dimensional entity. We map the 2D features extracted by the feature aggregation attention mechanism into three-dimensional point space, utilizing an attention mechanism to aggregate these feature points, effectively preserving spatial geometric details in the generation of three-dimensional objects.

Feature aggregation attention mechanism is mainly divided into three steps.

1) Pixel-level feature extractor for multi-viewimages.

As each pixel in a 2D image describes a specific three-dimensional point in space, this module aims to extract general features for each pixel, enabling the learning of regional descriptions and geometric features for each ray. Due to the sensitivity of RGB images to lighting conditions and environmental noise, we do not directly extract noise from RGB images. Instead, we combine RGB images with corresponding viewpoint information as input to the feature extraction network. This ensures that the learned pixel features explicitly understand their relative positions in three-dimensional space. Taking the right view as an example, the pixel extraction process can be represented by Eq. (3):

$$F_{\mathrm{p}}^{\mathrm{right}} = \mathrm{CNNs}(\mathrm{x}_{\mathrm{right}} \oplus \mathrm{V}_{\mathrm{right}}). \quad (3)$$

2) Mapping 2D Point Features to 3D Space.

For each three-dimensional point p in space, we can retrieve a feature vector from each input image. The feature set for point p is denoted as $\hat{F}_p = \{F_p^{\mathrm{right}}, F_p^{\mathrm{left}}, ..., F_p^{\mathrm{bottom}}\}$ representing the six orthogonal views (front, back, left, right, top, bottom). For each retrieved feature vector F_p of point p, we first use a shared MLP network to integrate the feature and position information of query point p, generating a new feature vector that understands its relative distance from point p. This is represented by Eq. (4):

$$\hat{\mathbf{F}}_p^m = MLPs(\mathbf{F}_p^m \oplus [x_p, y_p, z_p]), (\oplus\text{is concatenation}). \quad (4)$$

3) Applying an attention mechanism to aggregate featurevectors.

After obtaining the new position-aware feature set, we use an attention mechanism to compute a unique feature vector for the three-dimensional point p. The attention mechanism, as specified in Eq. (5), ensures coverage invariance and can handle any number of elements in the input feature vector set.

$$\mathtt{attn}_{i,j} := \frac{e^{M_{i,j}}}{\sum_l e^{M_{i,l}}}, \quad \text{where } M := \frac{1}{\sqrt{D}} k(\mathtt{inputs}) \cdot q(\mathtt{slots})^{\mathrm{T}} \in \mathbb{R}^{N\times K}. \quad (5)$$

3.3 HexaConstraint module

NeRF. The NeRF model has been pivotal in 3D rendering by introducing a novel rendering approach using a neural radiance field for rendering and reconstructing 3D scenes. This technology eliminates the need for expensive scanning equipment or extensive manually labeled data, enabling the reconstruction of high-quality 3D scenes from relatively few image data. In this section, we leverage a variant of NeRF [31] to generate 3D scenes from 2D views, building upon the new perspective views reconstructed using the diffusion model in Subsection 3.1.

Given an input image x_0, we aim to learn the NeRF representation $F_\Theta : (x, y, z, \gamma, \phi) \longrightarrow (c, \sigma)$ for its 3D reconstruction, where (x, y, z) represents the camera position, and (γ, ϕ) represents the camera orientation. The core idea of NeRF lies in sampling camera rays $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ for any camera view V and rendering the image under that view using Eq. (6):

$$\hat{\mathbf{C}}(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(t)\mathbf{c}(t)\mathrm{d}t, \quad (6)$$

where $T(t) = \exp\left(-\int_{t_n}^{t} \sigma(s)\mathrm{d}s\right)$. Specific details of NeRF are omitted for brevity. For simplicity, we denote the entire rendering equation as $x = f(P, \Theta)$ using Eq. (7), indicating the rendering of image x under the camera view V using NeRF, with training parameters Θ.

$$\forall \mathbf{V},\ f(\mathbf{V}, \Theta) \sim \mathbb{P} \mid f(\mathbf{V}_0, \Theta) = \mathbf{x}_0(p). \quad (7)$$

Combining the newly generated perspective views from Subsection 3.1, we use these views as prior information. The overall objective is to maximize the conditional probability as expressed in Eq. (8):

$$\max_{\Theta} \mathbb{E}_{\mathbf{V}}\, \mathbb{P}(f(\mathbf{V}, \Theta) \mid f(\mathbf{V}_0, \Theta) = \mathbf{x}_0(p_0)), \quad (8)$$

where $x_0(p_0)$ represents the image x_0 generated based on the text

p_0. It is used to further constrain the prior image distribution, aiming to limit the features of the generated 3D object to be consistent with the input text.

HexaConstraint. The purpose of the HexaConstraint module is to correct the detailed attributes of objects through mutual supervision between real and generated images. Specifically, it involves differential supervision by comparing the six orthogonal views generated by the Hexaview Generation Diffusion Model with the six projected views rendered by the HexaConstraint module. Through training, the predicted images iteratively approach the global convergence of the loss function, gradually aligning with real images to generate more consistently synthesized views of 3D objects. We introduce the view projection matrix M_p, which projects the 3D object rendered by NeRF onto six orthogonal planes. The projection formula is given by Eq. (9):

$$X_p(V) = x(V) * M_p. \tag{9}$$

We apply differential supervision with the new perspective views to achieve fast convergence of the model. The specific implementation is shown in Eq. (10):

$$\mathrm{L_p} = \|\mathrm{X_p(V)} - \mathrm{X(V)}\|_2^2. \tag{10}$$

During training, we need to calculate the HexaConstraint loss for each of the six faces. Due to data imbalance, the convergence speed of the six views may be inconsistent. Uncommon views, such as top and bottom views, may converge more slowly, while other views may converge faster. Therefore, we do not directly calculate the overall loss for the six views. Instead, we use a Pareto optimization approach to find a set of compromise solutions among the six view loss functions to satisfy a predefined threshold.

Regarding the reason for adopting the Pareto optimization method, we adopt it to address the trade-off among the losses of six views. This method allows for multi-objective optimization, finding a compromise solution when different view losses conflict. It also handles data imbalance among views by dynamically adjusting the weights of each view, improving training results. We define this Pareto optimization problem as follows and minimize the sum of the HexaConstraint loss for the six views, as expressed in Eq. (11):

$$\min L_p = (L_{p_1}(x), \ldots, L_{p_6}(x)). \tag{11}$$

For the weight of each view, we define $\lambda_1, \ldots, \lambda_6$, where $\lambda_1 + \cdots + \lambda_6 = 1$. The goal is to minimize the sum of the loss for the six faces, as in Eq. (12):

$$F(x) = \lambda_1 * L_{p_1}(x) + \cdots + \lambda_6 * L_{p_6}(x). \tag{12}$$

When update the λ parameter, they must satisfy the following constraint, as given in Eq. (13):

$$\min_{\lambda_i} \| \sum_{i=1}^{m} \lambda_i \nabla_x f_i(x) \|_2$$
$$\text{subject to } \sum_{i=1}^{m} \lambda_i = 1 \text{ and } \lambda_i \geqslant 0 \text{ for all } i \in [m]. \tag{13}$$

4 Experimental evaluation

In this section, we assess the performance of the model in the reconstruction of 3D objects when giving textual descriptions. We will present details of the experimental setup, including dataset, parameter configurations, and evaluation metrics in Subsection 4.1. Additionally, we will conduct comparative experiments in Subsection 4.2, comparing our approach with state-of-the-art models through qualitative and quantitative analyses. Subsection 4.3 involves ablation experiments to individually validate the impact of each module proposed in this paper on the quality of generated 3D objects.

4.1 Implementation details

4.1.1 Data preparation

During the training process, our dataset primarily consists of two parts: Objaverse (60%) and LAION-Aesthetics v2 [48] (40%). Objaverse is a large-scale open dataset containing over 800,000 3D models. Objaverse includes multiple of high-quality 3D models with rich geometric shapes, fine details, and material properties. For each object in the dataset, we extract six orthographic camera extrinsic matrices, each pointing towards the center of the object, and render six views using a ray tracing engine.

LAION-Aesthetics v2 is derived from a subset of the publicly available LAION-5B dataset, containing a subset of 120 million images suitable for large-scale pretraining, text-image matching, and image generation tasks. During training, we use the LAION dataset to train a view-sensitive diffusion model.

4.1.2 Training parameters

Regarding hyperparameters, we use a basic set of hyperparameters across all experiments without specific optimization for each scenario. In the training phase, we utilize Adam optimization with a learning rate of 0.001 and no weight decay over 10,000 iterations. To maintain consistency across multiple views, we reduce the image resolution to 256×256 instead of 512×512, and the total batch size for training is set to 128 (equivalent to 768 objects). The training takes 5.5 days on eight A100 GPUs. For camera sampling, lighting, and shadow aspects, we retain nearly all parameters from DreamFusion. In the optimization process, we randomly use diffuse reflection and textureless shading after the initial warm-up optimization. The weights are initialized as 0.167 for each view. The evaluation encompasses 500 3D models generated based on specific text prompts as benchmark data.

4.1.3 Evaluation metrics

Two metrics are considered in the statistical evaluation: CLIP-R-Presion [40] and FréchetInception Distance (FID) [49]. CLIP-R-Presion evaluates the generated 3D objects by measuring the consistency between the generated 3D object with the generated scene and the given text. FID is an indicator for assessing the quality of images generated by the model, with lower scores indicating better model performance. Additionally, three qualitative metrics: 1) multi-view coherence, 2) richness of details, and 3) fidelity of major parts, are designed through 200 questionnaires.

4.2 Comparative experiments

In this section, we compare our proposed method for generating high-quality textual-driven 3D objects, HexaConstraint, with state-of-the-art models such as DreamFusion [1], PerpNeg [43], RealFusion [2], and Magic123 [11].

4.2.1 Qualitative evaluation

Firstly, we provide an intuitive depiction of the quality of 3D objects generated for text-to-3D generation, as illustrated in Fig. 3. For the same text description, "a Barbie doll," different methods may randomly generate Barbie dolls of various colors and shapes due to the stochastic nature of image generation in the T2I frozen model. As observed in Fig. 3, all methods achieve excellent rendering results in terms of color and realism. However, upon closer inspection of the magnified details, it is evident that the RealFusion and PerpNeg, although superior to our approach in texture details and colors, exhibit distortions in the facial features of the Barbie doll, resulting in a multi-faceted effect. There are even perplexing occurrences such as the appearance of a third arm with an ambiguous block in the lower right corner of the image. Similarly, Magic123 also exhibits multi-faceted phenomena. Besides, the combination of red and green brings out an inconsistency considering the color matching of attire. In contrast, our method generates a Barbie doll that aligns well with common sense, demonstrating better color coordination and realism in detail. Moreover, the coupling between views is enhanced by introducing multi-view constraint, effectively alleviating the occurrence of multi-face issues.

4.2.2 Quantitative evaluation

In terms of statistical evaluation, our method performs best in CLIP-R-Precision and ranks second in FID in Table 1. This result aligns with the qualitative findings observed in Fig. 3. From a manual evaluation perspective, our method significantly alleviates the multi-head problem. In terms of multi-head alleviation metrics, our method

Fig. 3 Prompt: A barbie doll is dancing. On the left is the experimental methods, while the right side displays detailed images and the rendering results from various perspectives. Our method achieved the best results in terms of detail texture and object consistency

Table 1 HexaDream shows comparative performance in demultihead and fidelity

Combined method	Statistical evaluation		Artificial evaluation		
	FID↓	CLIP-R-Precision↑	Multi-head alleviation↑	Detail richness↑	Keypart fidelity↑
Dreamfusion [1]	157	0.284	0.134	0.305	0.184
PerpNeg [43]	**87**	0.387	0.576	0.340	0.256
RealFusion [2]	196	0.264	0.253	0.371	0.331
Magic123 [11]	113	0.778	0.473	**0.773**	0.620
HexaDream	96	**0.820**	**0.782**	0.653	**0.687**

exhibits approximately 30% improvement compared to Magic123. Additionally, our method achieves satisfactory results in the other two evaluation metrics.

Among the five evaluation metrics shown in Table 1, our method does not achieve the best performance in all metrics. However, it notably outperforms other comparison methods in CLIP-R, multi-head alleviation, and keypart fidelity. Particularly in the multi-head alleviation, our method shows a significant enhancement by improving by 20%, compared with the PerpNeg which optimizes multi-headed issues with negative prompt engineering. However, our method slightly lags behind PerpNeg [43] and Magic123 [11] in terms of diversity.

4.3 Ablation experiments

We conduct ablation experiments to assess the contributions of three modules (Module1: hexaview generation diffusion model; Module2: feature aggregation attention mechanism; Module3: dynamic-weighted HexaConstraint module) based on DreamFusion independently. However, DreamFusion predicts three additional views from one text-generated image and renders a 3D object from four counterparts, while our method renders on six orthogonal views, the difference in the number of rendered views makes it challenging to directly use DreamFusion as a baseline. Therefore, we make slight adjustments to DreamFusion by adding two supplementary views, changing the tetrahedral constraint to a hexahedral constraint for 3D object generation, and name DreamFusion+ as the benchmark.

We investigate the impact of three proposed modules on the experimental evaluation metrics and visualize the results using a radar chart as shown in Fig. 4. Figure 4(a) illustrates the evaluation metrics for DreamFusion+ and the experiment with all three modules added. It is evident that whether in terms of statistical metrics or manual evaluation, the experiment with all three modules performs significantly better than DreamFusion+. HexaDream shows an improvement of approximately 30 particularly on 3D generation diversity and quality.

Figure 4(b) depicts the evaluation metrics for DreamFusion+ with each of the three modules added separately. It is clear that adding Module 1 significantly improves multi-head alleviation (20.6%↑) while adding Module 3 has the best effect on enhancing the diversity and quality of generated 3D objects (12%↑). Looking at the other three experimental evaluation metrics, adding these modules separately leads to a substantial improvement, with relatively consistent enhancement effects compared to Module 3 individually. To elaborate further, Module 1 may contribute to reducing information insufficiency or distortion caused by a single viewpoint, effectively alleviating the multi-head problem by introducing Constraint from other viewpoints. Covering six main viewpoints, Module 3 ensures consistency in the geometric structure of the generated object. It also fully utilizes information observed from different viewpoints, resulting in richer and more realistic detailed textures. Module 2 achieves the best performance in detail richness owing to the feature aggregation.

To further study the loss distribution of rendered views and projected views for the three modules on six orthogonal views, we plot the line chart displayed in Fig. 5. The horizontal axis represents the six views, and the vertical axis represents the loss scores between rendered views and projected views. Overall, higher loss scores are concentrated in the top view and bottom view, possibly due to the imbalance in training data. However, after introducing Module 1 and Module 3, the loss for these two views decreases significantly, demonstrating that both multi-view Constraint and hexahedral supervision Constraint can significantly improve the generation of 3D objects, mitigating adverse effects caused by data imbalance during training thanks to the weight adjustment dynamically.

5 Limitations

In this study, we aim to address the challenge of insufficient image details and textures by increasing the number of views. By generating six main orthogonal views of objects through text-to-view conditional guidance, we successfully capture the overall structure of

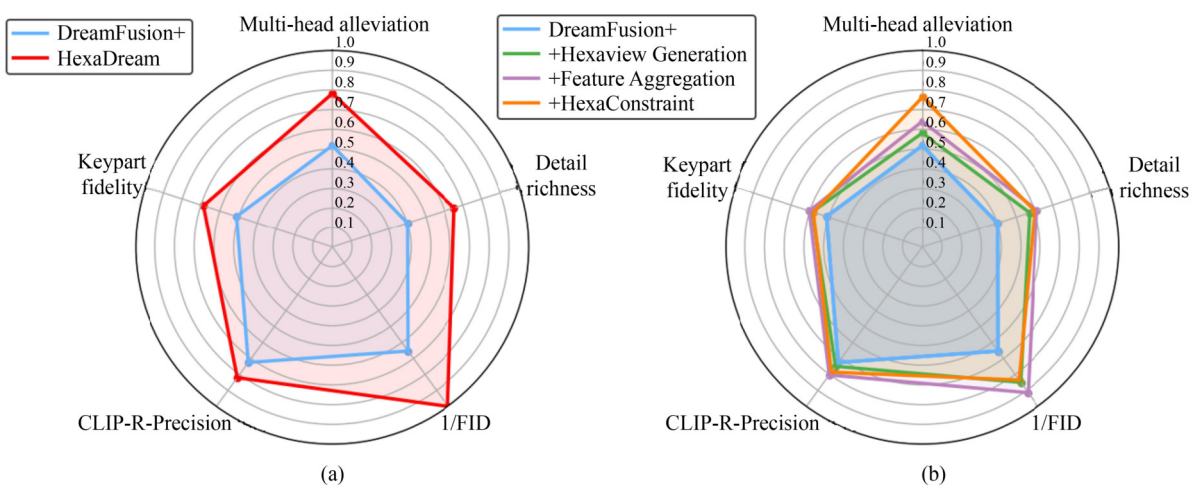

Fig. 4 HexaDream also achieves comparative performance in module level. To align with our hexa-constraint approach, we have augmented the DreamFusion methodology as DreamFusion+ by adding two additional views. (a) Evaluation metrics for DreamFusion+ and with all three modules added; (b) evaluation metrics for DreamFusion+ with each of the three modules added separately

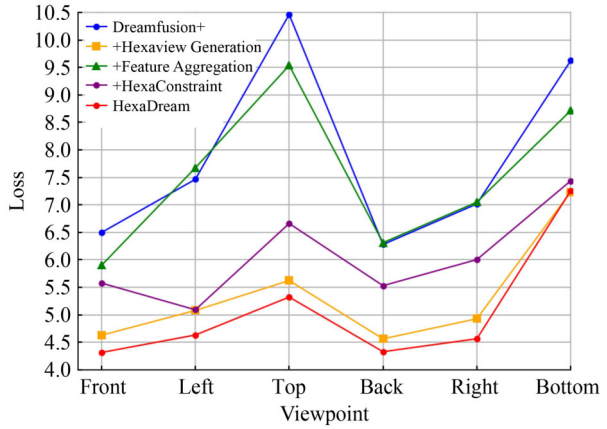

Fig. 5 The view loss of ablation experiments. To further explore the influence of view angle on 3D object synthesis, we calculate HexaConstraint loss on six orthogonal views seperately in ablation experiments

objects. However, the fixed selection of views may lack flexibility and result in memory waste when generating less important views. To overcome this limitation, future research can explore a more flexible approach that increases the depiction of key parts' perspectives using local views. This approach would allow us to maximize the capture of the overall object structure while avoiding unnecessary memory usage.

6 Conclusion

This study introduced a high-fidelity text-to-3D object model supervised by six orthogonal views, significantly improving challenges in generating realistic 3D objects with distorted or less-than-ideal fidelity in the text-to-3D object domain. The six-sided orthogonal view generation module, guided by text, produced images from six main perspectives, presenting the object comprehensively. Additionally, the feature extraction module extracted 2D point features from six views and, through feature aggregation with attention mechanisms, uniquely mapped these 2D point features into 3D space, enhancing the coupling between different views. Finally, using neural radiance fields, we rendered 3D objects and projected them onto six specific planes for differential supervision. We designed a method to dynamically adjust the weights of losses for each face, significantly accelerating the model's convergence speed. Experimental results demonstrated that our method achieved state-of-the-art performance across multiple metrics. Furthermore, we observed the framework had a significant impact on alleviating the generation of multiple heads.

Acknowledgements

This research was supported by the Hunan Province Graduate Student Innovation Project. We gratefully acknowledge the funding provided by Project XJQY2024040, Project XJZH2024038 and QL20220009.

Competing interests

The authors declare that they have no competing interests or financial conflicts to disclose.

Appendixes

A.1 Implementation details

In this section, we provide full implementation details which were omitted from the main text due to space constraint. The additional ablation of Stable-DreamFusion with/without HexaConstraint is displayed in Fig. A1. Besides, Fig. A2 demonstrates the insightful idea of how to make multi-view attention constraint. Detailed elaboration of the subjective evaluation procedure: For the user study: Detail Richness evaluates the level of detail in the generated 3D objects, including texture, material, and fine structures. Keypart Fidelity assesses the similarity of the generated 3D objects to real objects in key parts, e.g., facial features, animal limbs.

- **Participants** We involve 200 participants with diverse backgrounds in computer graphics and 3D modeling.
- **Procedure** Participants are asked to evaluate the generated 3D models based on three criteria: multi-head alleviation (MA), detail richness (DR), and key part fidelity (KF). Each participant reviews a randomized set of models without knowing which method generated to prevent bias.
- **Data collection platform** The evaluations are conducted using an online survey platform, ensuring anonymity and ease of data collection.
- **Screening** We include attention checks to ensure data quality and exclude responses that fail these checks.

Shading. We consider three different types of shading: albedo, diffuse, and textureless.

For albedo, we simply render the RGB color of each ray as given by our model:

$$I(u) = I_\rho(u) = \mathcal{R}(u;\sigma,c). \tag{A1}$$

For diffuse, we also compute the surface normal n as the normalized negative gradient of the density with respect to u. Then, given a point light l with color l_ρ and an ambient light with color l_a, we render

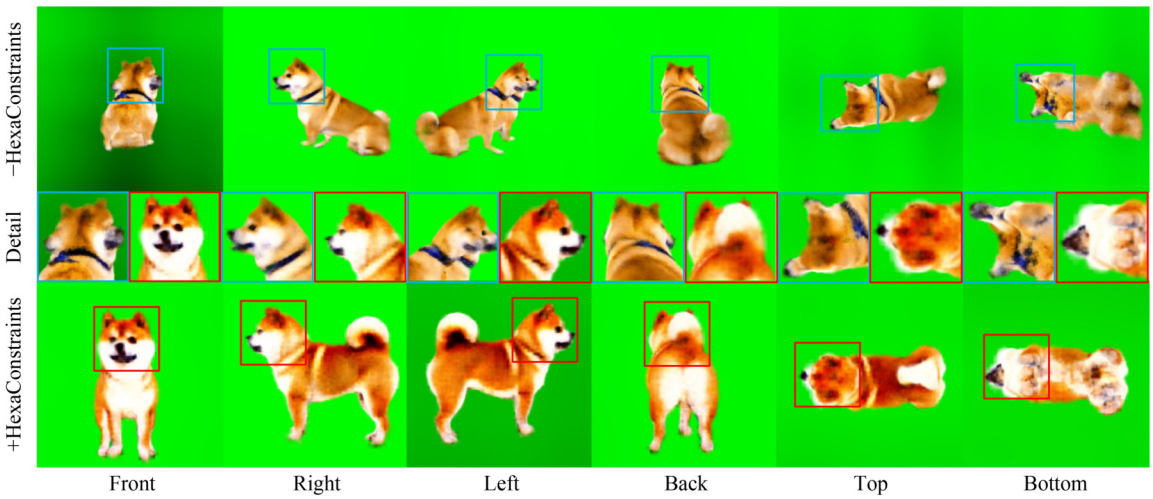

Fig. A1 Prompt "A Shiba" by Stable-DreamFusion with/without HexaConstraint. The HexaConstraint algorithm improves the accuracy of diffusion models to follow view instructions specified in text prompts during 3D scene training. It enhances the ability to observe finer texture details from hex-view supervision, which mitigates the Janus problem

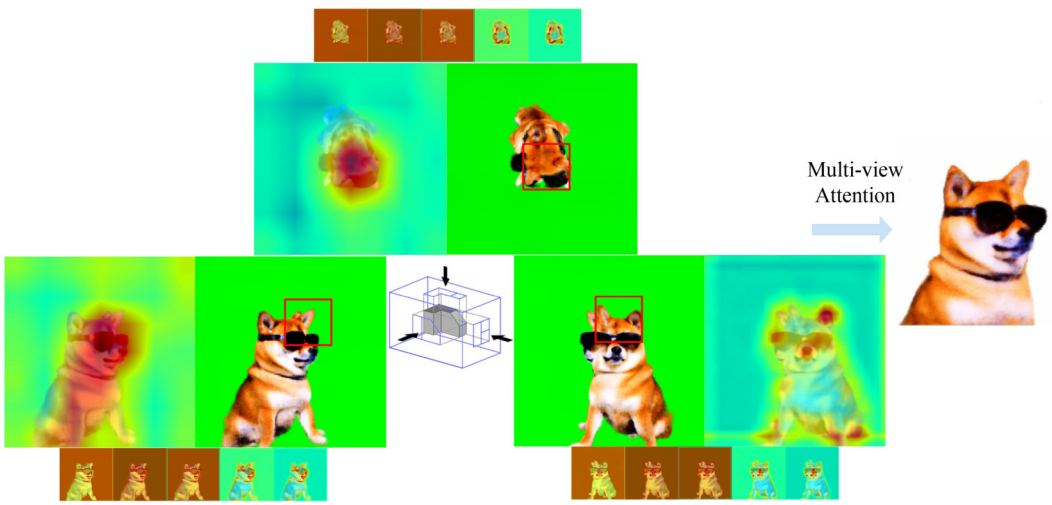

Fig. A2 Multi-view attention constraint

$$I(u) = I_\rho(u) \circ \left(l_\rho \circ \max\left(0, n \cdot \frac{l-u}{\|l-u\|} + l_a\right)\right). \tag{A2}$$

For textureless, we use the same equation with $I_\rho(u)$ replaced by white (1,1,1). For the reconstruction view, we only use albedo shading. For the random view (i.e., the view used for the prior objectives), we use albedo shading for the first 1,000 steps of training by setting l_a = 1.0 and l_ρ = 0.0. Afterwards we use l_a= 0.1 and l_ρ = 0.9, and select stochastically between albedo, diffuse, and textureless with probabilities 0.2, 0.4, and 0.4, respectively. We obtain the surface normal using finite differences:

$$n = \frac{1}{2 \cdot \epsilon} \begin{pmatrix} I(u+\epsilon_x) - I(u-\epsilon_x) \\ I(u+\epsilon_y) - I(u-\epsilon_y) \\ I(u+\epsilon_z) - I(u-\epsilon_z) \end{pmatrix}, \tag{A3}$$

where $\epsilon_x = (\epsilon, 0, 0), \epsilon_y = (0, \epsilon, 0)$, and $\epsilon_z = (0, 0, \epsilon)$.

Density bias. As in DreamFusion we add a small Gaussian blob of density to the origin of the scene in order to assist with the early stages of optimization. This density takes the form:

$$\sigma_{\text{init}}(\mu) = \lambda \cdot e^{-\|\mu\|^2/(2\nu^2)}, \tag{A4}$$

with $\lambda = 5$ and $\nu = 0.2$.

Camera. The fixed camera for reconstruction is placed at a distance of 1.8 from the origin, oriented toward the origin, at an elevation of 15° above the horizontal plane. For a small number of scenes in which the object of interest is clearly seen from overhead, the reconstruction camera is placed at an elevation of 40°.

The camera for the prior objectives is sampled randomly at each iteration. Its distance from the origin is sampled uniformly from 1.0 to 1.5. Its azimuthal angle is sampled uniformly at random from the 360° around the object. Its elevation is sampled uniformly in degree space from -10° to 90° with probability 0.5 and uniformly on the upper hemisphere with probability 0.5. The field of view is uniformly sampled between 40 and 70. The camera is oriented toward the origin. Additionally, every tenth iteration, we place the prior camera near the reconstruction camera: its location is sampled from the prior camera's location perturbed by Gaussian noise with mean 0 and

variance 1.

Lighting. We sample the position of the point light by adding a noise vector $\eta \sim \mathcal{N}(0,1)$ to the position of the prior camera.

View-Dependent Prompt. We add a view dependent suffix to our text prompt based on the location of the prior camera relative to the reconstruction camera. If the prior camera is placed at an elevation of above $60°$, the text prompt receives the suffix "overhead view." If it is at an elevation below $0°$, the text receives "bottom view." Otherwise, for azimuthal angles of $\pm 30°, \pm 30° - 90°$, or $\pm 90° - 180°$ in either direction of the reconstruction camera, it receives the suffices "front view," "side view," or "bottom view," respectively.

InstantNGP. Our InstantNGP parameterizes the density and albedo inside a bounding box around the origin with side length 0.75. It is a multi-resolution feature grid with 16 levels. With coarse-to-fine training, only the first eight (lowest- resolution) levels are used during the first half of training, while the others are masked with zeros. Each feature grid has dimensionality 2. The features from these grids are stacked and fed to a 3-layer MLP with 64 hidden units.

Rendering and diffusion prior. We render at resolution 96px. Since Stable Diffusion is designed for images with resolution 512px, we upsample renders to 512px be- fore passing them to the Stable Diffusion latent space encoder (i.e., the VAE). We add noise in latent space, sampling $t \sim \mathcal{U}(0.02, 0.98)$. We use classifier-free guidance strength 100. We find that results with classifier-free guidance strength above 30 produce good results; those below 30 lead to many more geometric deformities. Although we do not backpropagate through the Stable Diffusion UNet for L_{SDS}, we do backpropagate through the latent space encoder.

Optimization. We optimize using the Adam optimizer with learning rate $1e^{-3}$ for 5000 iterations. The optimization process takes approximately 45 minutes on a single A100 GPU.

Background model. For our background model, we use a two-layer MLP which takes the viewing direction as input. This model is purposefully weak, such that the model cannot trivially optimize its objectives by using the background.

Additional regularizers. We additionally employ two regularizers on our density field. The first is the orientation loss from Ref-NeRF, also used in DreamFusion, for which we use $\lambda_{\text{orient}} = 0.01$. The second is an entropy loss which encourages points to be either fully transparent or fully opaque: $L_{\text{entropy}} = (w\log 2(w) - (1-w)\log 2(1-w)$, where w is the cumulative sum of density weights computed as part of the NeRF rendering equation (Eq. (1)). Single-image textual inversion. Our single-image textual inversion step, which is a variant of textual inversion, entails optimizing a token e introduced into the diffusion model text encode to match an input image. The key to making this optimization successful given only a single image is the use of image augmentations. We optimize using these augmentations for a total of 3000 steps using the Adam optimizer with image size 512px, batch size 16, learning rate $5 \cdot 10^{-4}$, and weight decay $1 \cdot 10^{-2}$.

The embedding e can be initialized either randomly, manually (by selecting a token from the vocabulary that matches the object), or using an automated method.

One automated method that we find to be successful is to use CLIP (which is also the text encoder of the Stable Diffusion model) to infer a starting token to initialize the inversion procedure. For this automated procedure, we begin by considering the set of all tokens in the CLIP text tokenizer which are nouns, according to the WordNet. We use only nouns because we aim to reconstruct objects, not reproduce styles or visual properties. We then compute text embeddings for captions of the form "An image of a h token i" using each of these tokens. Separately, we compute the image embedding for the input image. Finally, we take the token whose caption is most similar to the image embedding as initialization for our textual inversion procedure.

We use the manual initialization method for the examples in the main paper and the automated initialization method for the examples in the supplemental material (i.e., those included below).

A.2 Performance evaluation

The training loss convergence for each viewpoint is reflected in Fig. A3(a). Ablation experiments are conducted to evaluate different view Constraint for text-to-3D generation including TetraConstraint, HexaConstraint, and TetradecConstraint as Fig. A3(b) shown. With the objective of identifying the most suiee Constraint for our specific task. It is obvious to see that TetradecConstraint outperforms the others in most metrics. This is attributed to the extensive application of view Constraint for supervision, which enhances the model's performance in 3D synthesis. Secondly, we evaluate the rates of

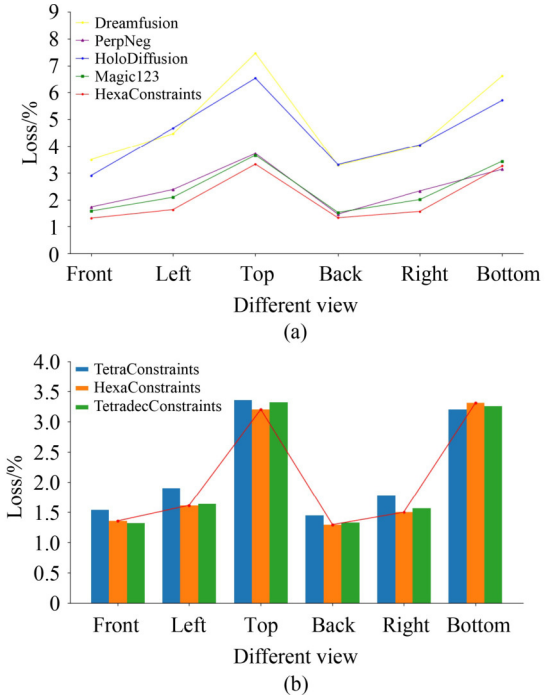

Fig. A3 Model training loss and multi-view constraint loss. (a) Comparative experiment showcases the loss of HexaConstraint and baseline models in different views. (b) The ablation experiment studies the trend of training loss between views in generating 3D targets with various number of views

multi-face alleviation, training speed, and memory consumption, respectively, as shown in Fig. A4. Most of these details follow DreamFusion as shown in Fig. A5, but a few are slightly modified. We present a detailed analysis of the performance variations resulting from the addition of different numbers of face constraints. These constraints are applied through plugin-style supervised learning on both DreamFusion (single-stage) and Magic123 (two-stage). The analysis reveals a fundamental conclusion: as the number of face

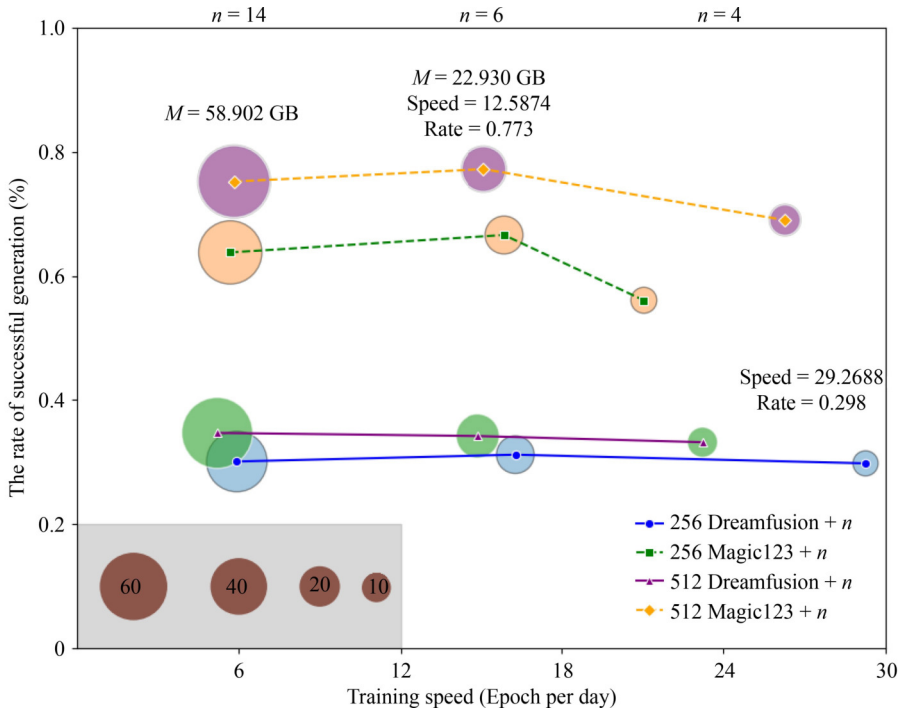

Fig. A4 Performance comparison of Dreamfusion and Magic123 models under the Constraint of different number of views. We comprehensively evaluate the varying relationship between multi-views Constraint and model performance by training speed, demultihead rate, and memory consumption (GB)

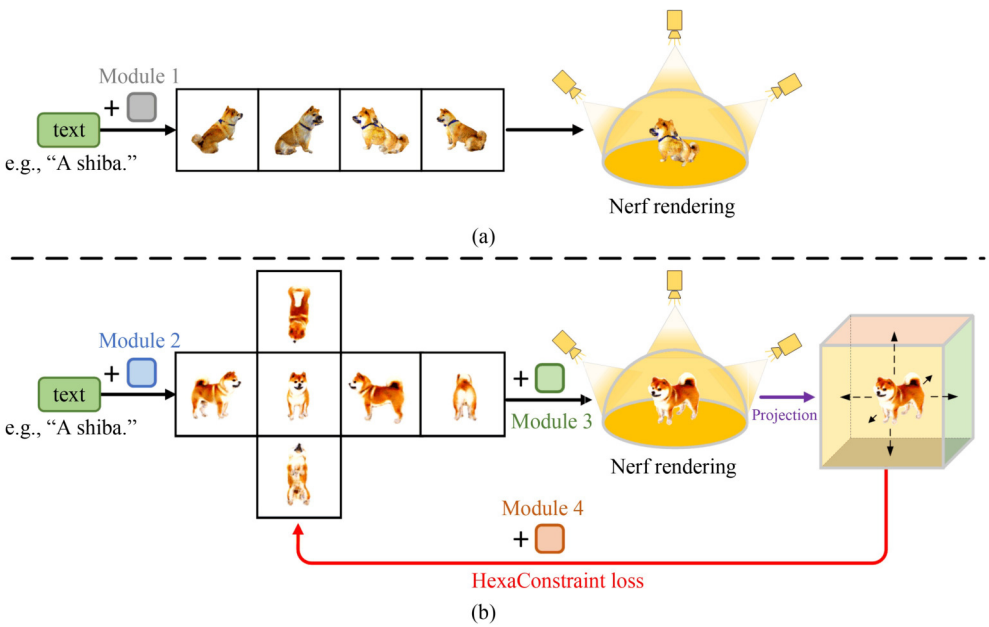

Fig. A5 Comparisons of DreamFusion [1] and HexaDream. Instead of adopting diffusion model (Module 1) in DreamFusion, we design a novel Hexaview generation diffusion model (Module 2) for rendering six-orthogonal-view images. Besides, we develop a feature aggregation module (Module 3) for mapping and aggregating the 2D features extracted from the rendered views into 3D space. After the NeRF rendering process, the 3D object is projected onto six-orthogonal-view for calculating the loss between the projection views and rendered views through the HexaConstraint module (Module 4), updating the NeRF parameters in the meantime. (a) DreamFusion; (b) HexaDream

constraints increases, the required GPU memory grows significantly. Moreover, the computation speed decreases notably with the addition of more constraints. Additionally, the impact of multi-head removal becomes more pronounced as the latent pixel space expands. Based on these findings, we offer the following recommendations: For those with access to an A100 GPU with 80GB of memory, we recommend using 14 face supervision constraints, provided that computation time is not a primary concern. For users with an A100 GPU offering approximately 40GB of memory, we suggest employing a six-face supervision scheme with a resolution of 512, balancing performance and memory consumption. For researchers using 3090 or 4090 GPUs, we recommend a four-face supervision scheme with a 256 resolution, which ensures that memory usage remains within a manageable range (10-20GB).

As the number of views and image resolution increases, there is a sharp rise in memory consumption. Transitioning from six views to fourteen views, the improvement in multi-head alleviation success rate is not significant, while the memory consumption for fourteen views is nearly triple that of six views. Therefore, in this paper, we opt for six views due to its excellent performance as a trade-off.

References

[1] Poole B, Jain A, Barron J T, Mildenhall B. DreamFusion: text-to-3D using 2D diffusion. In: Proceedings of the 11th International Conference on Learning Representations. 2023

[2] Melas-Kyriazi L, Laina I, Rupprecht C, Vedaldi A. RealFusion: 360° reconstruction of any object from a single image. In: Proceedings of 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023, 8446−8455

[3] Karnewar A, Vedaldi A, Novotny D, Mitra N J. HoloDiffusion: training a 3D diffusion model using 2D images. In: Proceedings of 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023, 18423−18433

[4] Shi Y, Wang P, Ye J, Long M, Li K, Yang X. MVDream: multi-view diffusion for 3D generation. 2023, arXiv preprint arXiv: 2308.16512

[5] Hu Z, Zhao M, Zhao C, Liang X, Li L, Zhao Z, Fan C, Zhou X, Yu X. EfficientDreamer: high-fidelity and stable 3D creation via orthogonal-view diffusion priors. In: Proceedings of 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024, 4949–4958

[6] Zhang Z, Chen H, Yin X, Deng J. EAWNet: an edge attention-wise objector for real-time visual internet of things. Wireless Communications and Mobile Computing, 2021, 2021(1): 7258649

[7] Zhang Z, Chen H, Yin X, Deng J, Li W. Dynamic selection of proper kernels for image deblurring: a multistrategy design. The Visual Computer, 2023, 39(4): 1375−1390

[8] Tang S, Zhang F, Chen J, Wang P, Furukawa Y. MVDiffusion: enabling holistic multi-view image generation with correspondence-aware diffusion. In: Proceedings of the 37th International Conference on Neural Information Processing Systems. 2023, 2229

[9] Cho J, Youwang K, Oh T H. Cross-attention of disentangled modalities for 3D human mesh recovery with transformers. In: Proceedings of the 17th European Conference on Computer Vision. 2022, 342−359

[10] Vilums R, Buikis A. Conservative averaging and finite difference methods for transient heat conduction in 3D fuse. WSEAS Transactions on Heat and Mass Transfer, 2008, 3(1): 111−124

[11] Qian G, Mai J, Hamdi A, Ren J, Siarohin A, Li B, Lee H Y, Skorokhodov I, Wonka P, Tulyakov S, Ghanem B. Magic123: one image to high-quality 3D object generation using both 2D and 3D diffusion priors. In: Proceedings of the 12th International Conference on Learning Representations. 2024

[12] 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. 2023, arXiv preprint arXiv: 2305.16213

[13] Saharia C, Chan W, Saxena S, Lit L, Whang J, Denton E, Ghasemipour S K S, Ayan B K, Mahdavi S S, Gontijo-Lopes R, Salimans T, Ho J, Fleet D J, Norouzi M. Photorealistic text-to-image diffusion models with deep language understanding. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. 2022, 2643

[14] Radford A, Kim J W, Hallacy C, Ramesh A, Goh G, Agarwal S, Sastry G, Askell A, Mishkin P, Clark J, Krueger G, Sutskever I. Learning transferable visual models from natural language supervision. In: Proceedings of the 38th International Conference on Machine Learning. 2021, 8748−8763

[15] Nichol A Q, Dhariwal P, Ramesh A, Shyam P, Mishkin P, McGrew B, Sutskever I, Chen M. GLIDE: towards photorealistic image generation and editing with text-guided diffusion models. In: Proceedings of the 39th International Conference on Machine Learning. 2022, 16784−16804

[16] Ramesh A, Dhariwal P, Nichol A, Chu C, Chen M. Hierarchical text-conditional image generation with CLIP latents. 2022, arXiv preprint arXiv: 2204.06125

[17] Yu J, Xu Y, Koh J Y, Luong T, Baid G, Wang Z, Vasudevan V, Ku A, Yang Y, Ayan B K, Hutchinson B C, Han W, Parekh Z, Li X, Zhang H, Baldridge J, Wu Y. Scaling autoregressive models for content-rich text-to-image generation. 2022, arXiv preprint arXiv:2206.10789

[18] Ding M, Zheng W, Hong W, Tang J. CogView2: faster and better text-to-image generation via hierarchical transformers. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. 2022, 1229

[19] Rombach R, Blattmann A, Lorenz D, Esser P, Ommer B. High-resolution image synthesis with latent diffusion models. In: Proceedings of 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022, 10674−10695

[20] Ruiz N, Li Y, Jampani V, Pritch Y, Rubinstein M, Aberman K. DreamBooth: fine tuning text-to-image diffusion models for subject-driven generation. In: Proceedings of 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023, 22500−22510

[21] Liu M, Shi R, Chen L, Zhang Z, Xu C, Wei X, Chen H,Zeng C, Gu J, Su H. One-2-3-45++: Fast single image to 3d objects with consistent multi-view generation and 3d diffusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2024, 10072–10083

[22] 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: Proceedings of

2023 IEEE/CVF International Conference on Computer Vision. 2023, 9264−9275

[23] Shi R, Chen H, Zhang Z, Liu M, Xu C, Wei X, Chen L, Zeng C, Su H. Zero123++: a single image to consistent multi-view diffusion base model. 2023, arXiv preprint arXiv: 2310.15110

[24] Chen Z, Zhang H. Learning implicit fields for generative shape modeling. In: Proceedings of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2019, 5932−5941

[25] Mescheder L, Oechsle M, Niemeyer M, Nowozin S, Geiger A. Occupancy networks: learning 3D reconstruction in function space. In: Proceedings of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2019, 4455−4465

[26] Henzler P, Mitra N, Ritschel T. Escaping plato's cave: 3D shape from adversarial rendering. In: Proceedings of 2019 IEEE/CVF International Conference on Computer Vision (ICCV). 2019, 9983−9992

[27] Lunz S, Li Y, Fitzgibbon A, Kushman N. Inverse graphics GAN: learning to generate 3D shapes from unstructured 2D data. 2020, arXiv preprint arXiv: 2002.12674

[28] Luo S, Hu W. Diffusion probabilistic models for 3D point cloud generation. In: Proceedings of 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2021, 2836−2844

[29] Zeng X, Vahdat A, Williams F, Gojcic Z, Litany O, Fidler S, Kreis K. LION: latent point diffusion models for 3D shape generation. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. 2022, 728

[30] Zhou L, Du Y, Wu J. 3D shape generation and completion through point-voxel diffusion. In: Proceedings of 2021 IEEE/CVF International Conference on Computer Vision (ICCV). 2021, 5806−5815

[31] Mildenhall B, Srinivasan P P, Tancik M, Barron J T, Ramamoorthi R, Ng R. NeRF: representing scenes as neural radiance fields for view synthesis. Communications of the ACM, 2021, 65(1): 99−106

[32] Meng Q, Chen A, Luo H, Wu M, Su H, Xu L, He X, Yu J. GNeRF: GAN-based neural radiance field without posed camera. In: Proceedings of 2021 IEEE/CVF International Conference on Computer Vision (ICCV). 2021, 6331−6341

[33] Wang Z, Wu S, Xie W, Chen M, Prisacariu V A. NeRF--: neural radiance fields without known camera parameters. 2021, arXiv: 2102.07064

[34] Deng K, Liu A, Zhu J Y, Ramanan D. Depth-supervised NeRF: fewer views and faster training for free. In: Proceedings of 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2022, 12872−12881

[35] Roessle B, Barron J T, Mildenhall B, Srinivasan P P, Nießner M. Dense depth priors for neural radiance fields from sparse input views. In: Proceedings of 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2022, 12882−12891

[36] Niemeyer M, Barron J T, Mildenhall B, Sajjadi M S M, Geiger A, Radwan N. RegNeRF: regularizing neural radiance fields for view synthesis from sparse inputs. In: Proceedings of 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2022, 5470−5480

[37] Deng, Congyue and Jiang, "Max" Chiyu and Qi, Charles R and Yan, Xinchen and Zhou, Yin and Guibas, Leonidas and Anguelov, Dragomir. NeRDi: single-view NeRF synthesis with language-guided diffusion as general image priors. In: Proceedings of 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).2023, 20637−20647

[38] Mohammad Khalid N, Xie T, Belilovsky E, Popa T. CLIP-Mesh: generating textured meshes from text using pretrained image-text models. In: Proceedings of SIGGRAPH Asia 2022 Conference Papers. 2022, 25

[39] Michel O, Bar-On R, Liu R, Benaim S, Hanocka R. Text2Mesh: text-driven neural stylization for meshes. In: Proceedings of 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022, 13482–13492

[40] Jain A, Mildenhall B, Barron J T, Abbeel P, Poole B. Zero-shot text-guided object generation with dream fields. In: Proceedings of 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022, 857−866

[41] Barron J T, Mildenhall B, Verbin D, Srinivasan P P, Hedman P. Mip-NeRF 360: unbounded anti-aliased neural radiance fields. In: Proceedings of 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022, 5460−5469

[42] 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: Proceedings of 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2023, 300−309

[43] Armandpour M, Sadeghian A, Zheng H, Sadeghian A, Zhou M. Re-imagine the negative prompt algorithm: transform 2D diffusion into 3D, alleviate Janus problem and beyond. 2023, arXiv preprint arXiv: 2304.04968

[44] Chang A X, Funkhouser T, Guibas L, Hanrahan P, Huang Q, Li Z, Savarese S, Savva M, Song S, Su H, Xiao J, Yi L, Yu F. ShapeNet: an information-rich 3D model repository. 2015, arXiv preprint arXiv: 1512.03012

[45] Yu L, Xiang W, Han K. Edit-DiffNeRF: editing 3D neural radiance fields using 2D diffusion model. 2023, arXiv preprint arXiv: 2306.09551

[46] Wu T, Zhang J, Fu X, Wang Y, Ren J, Pan L, Wu W, Yang L, Wang J, Qian C, Lin D, Liu Z. OmniObject3D: large-vocabulary 3D object dataset for realistic perception, reconstruction and generation. In: Proceedings of 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2023, 803−814

[47] Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, Zhou Y, Li W, Liu P J. Exploring the limits of transfer learning with a unified text-to-text transformer. The Journal of Machine Learning Research, 2020, 21(1): 140

[48] Schuhmann C, Beaumont R, Vencu R, Gordon C, Wightman R, Cherti M, Coombes T, Katta A, Mullis C, Wortsman M, Schramowski P, Kundurthy S, Crowson K, Schmidt L, Kaczmarczyk R, Jitsev J. LAION-5B: an open large-scale dataset for training next generation image-text models. In: Proceedings of the 36th International Conference on Neural Information Processing Systems. 2022, 1833

[49] Chong M J, Forsyth D. Effectively unbiased fid and inception score and where to find them. In: Proceedings of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020, 6069−6078

Zhi-Chao ZHANG is currently a PhD student at College of Computer Science and Technology, National University of Defense Technology, China. His research interests include 3D computer vision and 4D generative content.

Hui CHEN is currently a PhD student at College of Computer Science and Technology, National University of Defense Technology, China. Her research interests include computer vision and robotics.

Jin-Sheng DENG received his PhD degree in computer science from National University of Defense Technology, China. He is currently a professor at Military Intelligent Research Institute, Academy of Military Sciences, China. His research interests include computer vision, 3D vision, virtual reality, big data and intelligence, and natural language processing.

Ming XU received his PhD degree in computer science from National University of Defense Technology (NUDT), China. He is currently a professor at College of Computer Science and Technology, NUDT. His research interests include federated learning, privacy protection, and large language models.

Zheng-Bin PANG received his PhD degree in computer science from National University of Defense Technology (NUDT), China. He is currently a professor at College of Computer Science and Technology, NUDT. His research interests include parallel computing, path planning, and machine learning.