MVGBench: a Comprehensive Benchmark for Multi-view Generation Models

Xianghui Xie[1,2,3] Chuhang Zou Meher Gitika Karumuri Jan Eric Lenssen[3] Gerard Pons-Moll[1,2,3]

[1]University of Tübingen, Germany [2]Tübingen AI Center, Germany
[3]Max Planck Institute for Informatics, Saarland Informatic Campus, Germany
https://virtualhumans.mpi-inf.mpg.de/MVGBench/

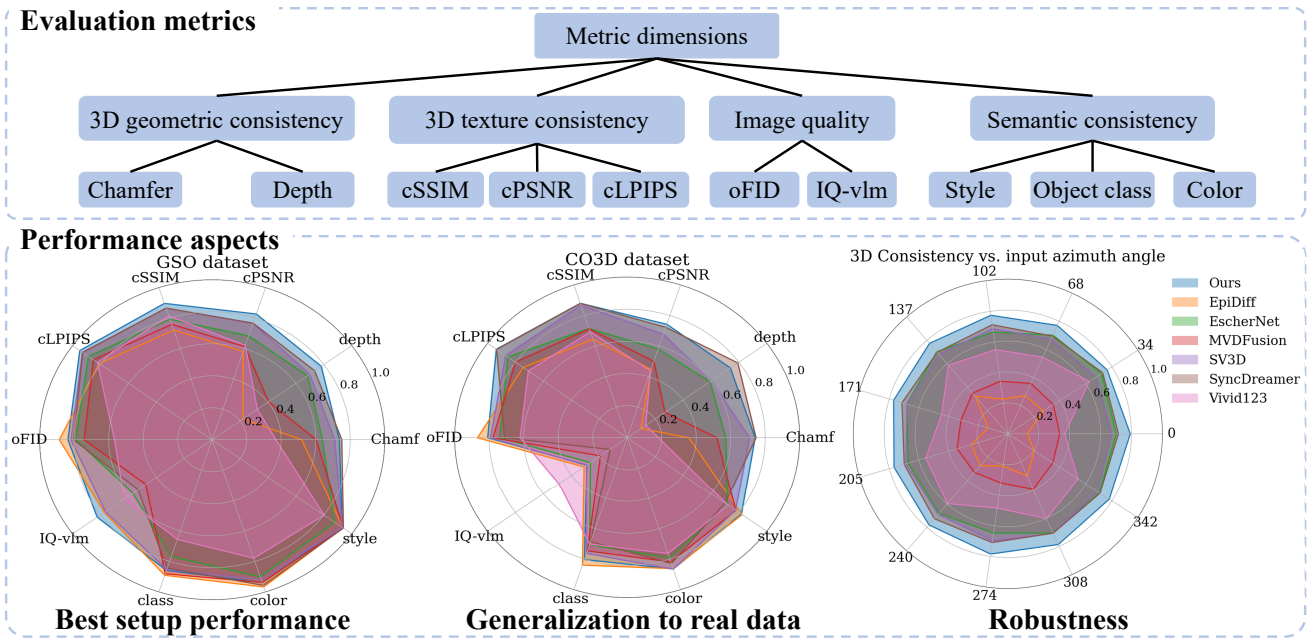

Figure 1. We present MVGBench, a comprehensive evaluation suite for multi-view image generation models (MVGs). We propose ten metrics to evaluate the 3D consistency in geometry and texture, image quality, and semantics of generated multi-view images. This suite allows us to fairly compare existing MVGs in three aspects: best setup performance, generalization, and robustness to input perturbations. We use our benchmark to systematically analyze different models and identify critical design choices, leading to a new model that achieves the best 3D consistency and robustness, with otherwise on-par performance. All values are normalized, and outermost is better.

Abstract

We propose MVGBench, a comprehensive benchmark for multi-view image generation models (MVGs) that evaluates 3D consistency in geometry and texture, image quality, and semantics (using vision language models). Recently, MVGs have been the main driving force in 3D object creation. However, existing metrics compare generated images against ground truth target views, which is not suitable for generative tasks where multiple solutions exist while differing from ground truth. Furthermore, different MVGs are trained on different view angles, synthetic data and specific lightings – robustness to these factors and generalization to real data are rarely evaluated thoroughly. Without a rigorous evaluation protocol, it is also unclear what design choices contribute to the progress of MVGs.

MVGBench evaluates three different aspects: best setup performance, generalization to real data and robustness. Instead of comparing against ground truth, we introduce a novel 3D self-consistency metric which compares 3D reconstructions from disjoint generated multi-views. We systematically compare 12 existing MVGs on 4 different curated real and synthetic datasets. With our analysis, we identify important limitations of existing methods specially in terms of robustness and generalization, and we find the most critical design choices. Using the discovered best practices,

we propose ViFiGen, a method that outperforms all evaluated MVGs on 3D consistency. Our benchmark suite and pretrained models will be publicly released.

1. Introduction

Powerful image generation models [7, 10, 46, 47, 53] have been foundation models for a variety of tasks such as low-level vision [55, 62, 73], image editing [13, 50, 78, 79] and 3D generation [20, 34, 43, 57]. Multi-view generation (MVG) models [28, 38, 54, 56] which are trained to generate images at target views are particularly important as they have been the driving force for the rapid development of 3D content creation. Based on MVGs [37, 54, 56], recent methods are able to create high quality 3D models from single images [36, 48, 70, 71] or text [26, 40, 49].

Given the fundamental importance and rapid development of MVGs, proper evaluation is however lagging behind. Prior works [28, 38, 54] compute 2D metrics such as PSNR and SSIM between the generated and ground truth novel view images (Fig. 2). This is problematic for two reasons: a) the generative model samples from a distribution of solutions, so there is no single correct ground truth view; and b) the generated images are evaluated independently without considering 3D consistency. Furthermore, each method is trained using images with different rendering setups, resulting in different optimal input and output settings. Simply comparing them to method-specific ground truth will yield incomparable numbers, as the size of objects in 2D renderings varies between methods.

Some works [60, 70, 71] compare the performance of MVGs by first lifting the multi-views to 3D, then aligning and calculating scores against 3D GT. The reliance on 3D GT makes it suboptimal to evaluate generative models and it is impossible to report numbers on real images where 3D GT is rarely available. Most works demonstrate selected examples, and only a few works conduct more rigorous evaluation via user studies, which is difficult to scale.

To this end, we propose MVGBench, a benchmark suite with comprehensive metrics and unified datasets for evaluating three important aspects of MVGs: *a) Best setup performance.* The actual performance of each method using input images from their optimal camera setup. *b) Generalization to real images.* We manually annotate real images with front view and elevation angles for a unified evaluation of generalization capabilities. *c) Robustness to input perturbations.* We render objects at different elevations, azimuths and lighting conditions as input images and evaluate the performance of each method under these settings.

A key contribution of our benchmark suite is a 3D consistency metric which does not compare against the ground truth and can faithfully assess methods that operate on different optimal setups. We evaluate 3D consistency by measuring the discrepancy between two 3D reconstructions obtained from two different subsets of the generated multi-view images. This design makes it suitable for evaluating generative models and also allows us to report quantitative results on real images in a more scalable way than user studies. In addition to 3D consistency, image quality and semantic consistency (class, color, and style) are also important for downstream applications, and we introduce metrics based on vision language models to evaluate these aspects.

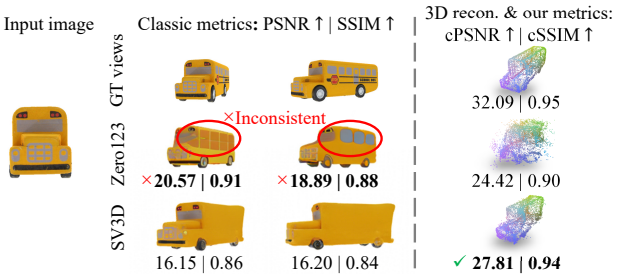

Figure 2. **Comparison of classic pair-wise metrics and our metrics**. Classic metrics compare generated images independently to paired ground truth views, which represent only one of many correct solutions in the ambiguous single-view generation task. In the example shown, despite the inconsistent generated multi-views, they assign a higher score to Zero123 [2] while our metrics correctly identify SV3D [54] as the more 3D consistent method.

We use our MVGBench to evaluate 12 state-of-the-art MVGs and analyse the key design choices of the best performing methods. We observe that there is a trade-off between 3D consistency and image quality in existing models, and video diffusion models generally achieve a better balance. However, a significant performance gap persists between synthetic and real images, and most methods lack robustness to input perturbations and struggle with fine-grained details. We further investigate design choices for 3D consistency and find that better camera embeddings and input image encoder can further improve performance. Leveraging best practices, we introduce ViFiGen, a video based multi-view generation method with a fine-grained input image encoder that outperforms all existing methods. Our evaluation suite and model will be publicly released.

In summary, our **contributions** are:

- We introduce the MVGBench suite, with comprehensive metrics and manually curated datasets to evaluate MVGs.
- We propose a novel 3D consistency metric that can fairly evaluate different MVGs on both synthetic and real data.
- We use MVGBench to systematically analyze and identify key problems of 12 state-of-the-art MVGs.
- We investigate the key design choices of the best performing MVGs and introduce ViFiGen that leverages the best practices and outperforms all existing baselines.

2. Related works

Multi-view image generation models (MVGs) [32, 37, 49, 56], typically fine-tuned from large-scale image [46, 47] or video [6, 66] generation models, have exhibited strong generalization capability [18, 70, 71] and significantly advanced 3D content creation [26, 36, 40, 48, 52, 69]. Zero123 [2] pioneered the repurposing of text to image generation model for camera-conditioned novel view synthesis. Follow-up works [39, 48, 49, 56, 80] further improve 3D consistency by simultaneously generating multiple views with advanced multi-view feature interaction mechanisms. SyncDreamer [38] proposes to synchronize multi-view features via 3D convolutions, while video-based models [9, 31, 54, 72] rely on dense spatial-temporal attention driven by video data. Epipolar attention is also a common way to enhance the consistency between novel views [17, 21, 67]. Beside feature interaction, different camera embeddings [14, 28, 67, 80] and input image encoding [42, 44, 59] are also adopted. Despite promising results, there is no unified evaluation protocol for MVGs, making it difficult to understand the actual progress in this field and contributions of different design choices. Our benchmark suite allows for a unified evaluation and analysis of MVGs.

Evaluation benchmarks or analysis such as [4, 5, 16, 51, 65] are essential to understanding the progress of a research field. The evaluation of generative models is less direct than traditional ground truth based evaluation and significant efforts have been made for evaluating image [15, 29, 68], video [3, 23, 33, 35], or 3D generation [64]. These benchmarks focus on semantic consistency [29], image or video quality [23], video dynamics [33] or alignment with human perceptions [15, 64, 68]. However, in the field of MVG, most works [2, 9, 17, 22, 25, 28, 30, 38, 54] still compare generated images against ground truth which is not meaningful and misses the important 3D consistency aspect, leading to inconsistent method rankings in different papers [22, 38, 54]. Flow Warping Score [35] and concurrent work MEt3R [3] measure consistency but focus on 3D scene generation. Some consistency metrics [58, 80] are proposed for object-level MVGs but are not well adopted. There is no unified and reliable evaluation protocol for object-level MVG models, let alone comprehensive analysis of this fast-evolving field with more than 20 papers per year. Our MVGBench provides a reliable 3D consistency metric and the first unified benchmark framework that allows fair comparison and systematic analysis.

3. MVGBench Evaluation Suite

We present MVGBench, a comprehensive evaluation suite for benchmarking multi-view generation models. We focus on evaluating models that generate images of a single or compositional object in the center while the background is masked out. Our evaluation suite consists of comprehensive evaluation metrics, including 3D consistency (Sec. 3.1), image quality, and semantic consistency (Sec. 3.2). We then curate several datasets to evaluate methods on three distinct aspects (Sec. 3.3). An overview of our metric dimensions and performance aspects can be found in Fig. 1. Please see metric implementation details in our Supp.

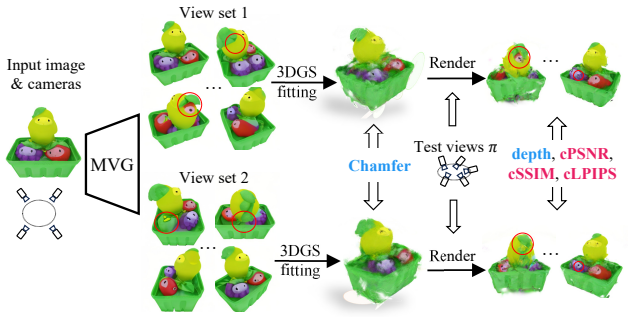

Figure 3. **3D consistency metrics for multi-view generation models**. After prompting the model to generate multi-views at target camera poses, we split the output views and fit 3D Gaussian Splatting (3DGS) separately into two view sets. We measure the **geometric** and **texture** consistency between two 3DGSs as the 3D consistency of the multi-view generation model.

3.1. 3D Consistency Metrics

Generated multi-view images should be 3D consistent to form a coherent 3D model [70, 71]. Previous methods evaluate this by first reconstructing 3D from multi-view images and then comparing to the 3D ground truth. However, this cannot be scaled to datasets without 3D ground truth. Moreover, generative models can produce images that differ from ground truth but are consistent, making this metric unreliable. Our key idea is to measure 3D consistency via self-consistency in 3D between generated images. An overview of our 3D consistency metrics is shown in Fig. 3.

Given a single RGB image $\mathbf{I}$ and N target view camera poses, we prompt the multi-view generation model to generate N images $\mathcal{I} = \{\mathbf{I}_1, ...\mathbf{I}_N\}$ at target views. We then divide these images into two subsets $\mathcal{I}_1 = \{\mathbf{I}_1^1, ...\mathbf{I}_n^1\}, \mathcal{I}_2 = \{\mathbf{I}_1^2, ...\mathbf{I}_n^2\}$ and fit two 3D Gaussian Splattings (3DGSs [27]) to them separately. We allow a small view overlap between $\mathcal{I}_1$ and $\mathcal{I}_2$ when N is small (detailed later). Let $\mathcal{G}_1, \mathcal{G}_2$ denote the optimized 3DGSs from two view sets $\mathcal{I}_1, \mathcal{I}_2$ respectively. In principle, the two 3DGSs will be very similar if the generated images are consistent with each other. Therefore, we measure the discrepancy between two 3DGSs as the indicator for the 3D geometric and texture consistency of the generated multi-views.

Geometric consistency metrics evaluate the consistency of the geometric structure between two 3DGSs fitted separately from two subsets of multi-views. We compute the `Chamfer` distance (CD) and rendered `depth` error between two 3DGSs. The Chamfer distance e_{cd} measures the

discrepancy in the overall shape structure, while the depth error e_d is more sensitive to edge inconsistencies.

For the **Chamfer distance**, instead of using the Gaussian centers that do not faithfully represent the actual surface of the shape, we resample the 3DGSs using the optimized covariance matrices. Let $\mu_1 \in \mathbb{R}^{M_1 \times 3}, \mathbf{\Sigma}_1 \in \mathbb{R}^{M_1 \times 3 \times 3}$ be the centers and covariances of the 3DGS $\mathcal{G}_1$ from images $\mathcal{I}_1$, the Chamfer distance between two 3DGSs is defined as:

$$e_{\text{cd}}(\mathcal{G}_1, \mathcal{G}_2) = d_{\text{CD}}(\mathbf{P}_1, \mathbf{P}_2) \quad (1)$$

$$\text{where } \mathbf{P}_1 \sim \mathcal{N}(\mu_1, \mathbf{\Sigma}_1), \mathbf{P}_2 \sim \mathcal{N}(\mu_2, \mathbf{\Sigma}_2) \quad (2)$$

here $d_{\text{CD}}(\cdot, \cdot)$ is the Chamfer distance between two sets of points. Two 3DGSs usually have different number of Gaussians. Hence we first sample five points from each Gaussian as $\mathbf{P}_1, \mathbf{P}_2$ and then downsample to 60k points to compute d_{CD}. We empirically found that this resampling produces more faithful distance values.

For the **depth error**, we render K depth maps of optimized 3DGSs following LGM [52], using the same camera views for two 3DGSs. We then compute the error between the two depth maps after masking out empty background:

$$e_d(\mathcal{G}_1, \mathcal{G}_2) = \frac{1}{K} \sum_{i=1}^{K} \frac{1}{|\mathbf{M}_i|} \sum \mathbf{M}_i |\pi_i^d(\mathcal{G}_1) - \pi_i^d(\mathcal{G}_2)| \quad (3)$$

where π_i^d denotes the depth rendering of view i and $\mathbf{M} = (\pi_i^d(\mathcal{G}_1) > 0) | (\pi_i^d(\mathcal{G}_2) > 0)$ is the union mask of the two rendered foregrounds.

Texture consistency metrics. We render each of the two optimized 3DGSs $\mathcal{G}_1, \mathcal{G}_2$ into K different views using same cameras as π_i^d. We then calculate the distance between the two renderings using PSNR, SSIM, and LPIPS:

$$e_m(\mathcal{G}_1, \mathcal{G}_2) = \frac{1}{K} \sum_{i=1}^{K} d_m(\pi_i(\mathcal{G}_1), \pi_i(\mathcal{G}_2)) \quad (4)$$

where π_i renders 3DGS $\mathcal{G}$ into an RGB image and $d_m(\cdot, \cdot)$ is the distance between two images defined by PSNR, SSIM or LPIPS. To distinguish them from classic names, we call these consistency metrics `cPSNR`, `cSSIM`, and `cLPIPS`.

Handling different input and output setups. One challenge in comparing different MVGs is that these methods render the training images at different camera focal lengths, distances, or elevations. This creates training data bias, and each model works optimally in different setups. Therefore, it would be *unfair* to render the same input image for all methods and compare the results, as this would favor methods trained on a similar camera setup. Using different testing inputs yields output objects of different sizes or view angles, and comparing against method-specific GT views (as done in prior evaluation) leads to incomparable numbers across methods. Our consistency metrics address this problem by design, as we can define arbitrary rendering views π_i for evaluation and use the same view across all methods. The only requirement is that the optimized 3DGSs are aligned in 3D space, as we discuss below.

For synthetic datasets where 3D is available, we normalize the 3D object within the unit cube and use the training camera setup (focal, distance) of each method to render its input image. When prompted with this image, the generated multi-views should represent a 3D object that aligns with the normalized 3D object in scale and rotation. Hence, the 3DGSs optimized with each methods' own cameras are also aligned, allowing us to define same test views π_i and compute scores that are comparable. For real images where we cannot recreate the input image at a specific camera setup, the optimized 3DGS will be misaligned since different methods generate different object sizes given the same input image. In this case, we use ICP with uniform scaling to align 3DGSs from different methods to a reference 3DGS from one MVG. We then use the same camera views for all methods to render images from the aligned 3DGS, leading to aligned consistency scores, as desired.

The number of output views each method is trained on also differ, affecting the accuracy of 3DGS fitting. To align the errors raised from 3DGS fitting given different views, we allow small overlap between the two view sets when the number of generated views is small, leading to a fair upper bound given different number of GT multi-views (Tab. 1).

Discussions. Our 3D consistency metric has two advantages over traditional evaluation metrics: (1) No comparison with 3D GT, which makes it more suitable for evaluating generative models and reporting quantitative results on real images. (2) Fair comparison of methods. Our metric allows each method to take input rendered in its own training data setting and report numbers that still align.

3.2. Semantic and Image Quality Metrics

The 3D consistency metric in Sec. 3.1 is suitable for evaluating the potential of generated images for 3D tasks. However, a list of pure white images is perfectly 3D-consistent but semantically meaningless. Therefore, it is also valuable to measure semantics which are orthogonal to 3D consistency but still important. To this end, we propose five metrics to measure image quality and semantic consistency.

Image quality. We assess the quality of generated multi-views using average object FID (`oFID`) and pretrained vision language models (VLMs). We compute oFID using CLIP features [44] which is more robust to diverse objects [24]. Let $\mathcal{F}_i^{\text{gt}}, \mathcal{F}_i$ be the CLIP features of the object i from N target views, oFID is the average FID of all objects in the test dataset $\mathcal{D}$: $\text{oFID} = \sum_{i=1}^{|\mathcal{D}|} \text{FID}(\mathcal{F}_i^{\text{gt}}, \mathcal{F}_i)$, where $\text{FID}(\cdot, \cdot)$ is the standard Fréchet distance. We show in Sec. 4.1 that our oFID aligns better with human preferences than the classic dataset level FID. Rendering at target views is difficult for real data as 3D GT is typically not available,

Figure 4. **Example results** on CO3D [45] (row 1), MVImgnet [75] (row 2), Omni3D [63] (row 3) and GSO [12] (row 4). Vivid123 [30] generates images that look good but are 3D inconsistent while SyncDreamer [38] images are 3D consistent but the image quality is worse. SV3D [54] and our model adopts video prior and achieves a better trade-off between consistency and quality. Our method leverages ConvNextV2 [59] instead of CLIP [44] to encode input image and preserves better details than SV3D.

we hence use datasets that capture each object in a video and randomly sample frames from this video to extract GT multi-view features.

We further prompt a pretrained VLM [8] to assess the overall image quality of the generated images and return a binary "yes" or "no" answer (yes means overall good). The score is the percentage of "yes" returned by the VLM, denoted as `IQ-vlm` (image quality via VLM).

Semantic consistency. We use the pretrained VLM [8] to assess whether the generated images are semantically consistent with the input image. We first prompt the VLM to annotate semantic attributes (object class, color, and appearance style) of each input using ground truth multi-view images, yielding the reference attributes. We then prompt the VLM with the generated images and ask if the reference attribute is presented in the image. Each attribute evaluation is a binary "yes/no" question with "yes" meaning the attribute is presented in the image. We denote these three metrics as `class`, `color`, and `style`. We show in Sec. 4.1 that our VLM-based metrics align well with human preferences and prompt template in Supp.

3.3. Evaluation Datasets

To have a common benchmark for comparing different methods, we curated several datasets and further processed them to evaluate three important performance aspects: best setup performance, generalization, and robustness.

Best setup performance. This setup is designed to compare the best performance of each method. Inputs are rendered from 3D models using each method's own training setup, ensuring that each method is run under optimal input conditions. We use Google Scanned Objects (GSO[12]) and OmniObject3D (Omni3D[63]) for this aspect. For GSO, we reuse 30 objects used in previous works (GSO30 [28, 38, 54]) and sample 70 more non-overlapping objects. For Omni3D, we randomly sample one instance from each category, resulting in 202 unique objects.

Generalization to real images. Existing methods typically show different qualitative examples of selected images, making it difficult to compare actual generalization abilities. Conducting user studies is more rigorous, but does not scale. Our metrics allow us to quantify the generalization performance on real images, which is more scalable. We use MVImgnet [75] and CO3D [45] for this evaluation. Both datasets capture multi-view images of real objects with unaligned camera poses, while most methods [17, 38, 54] require elevation angle as input. To this end, we manually select images that best fit as the frontal view and annotate their elevation angle. We select two instances per category for CO3D and one instance per category for MVImgnet, leading to 102 and 230 images, respectively.

Robustness. Another important aspect is to understand the robustness of the model under different input perturbations, which is barely done in previous evaluations. We consider three types of input perturbations that are difficult to undo via input image processing once the images are captured: elevation, azimuth, and light intensity. To do this, we render the GSO30 objects used by [28] under different conditions as input images. Note that in addition to the control factors we consider, we still use the camera focal length and distance from the training setup of each method to ensure that each method operates under optimal condition.

Most MVGs generate images at the same elevation as input, hence the optimized 3DGS is mostly accurate at similar

elevations. Errors in renderings far from this elevation are mainly caused by 3DGS fit rather than multi-view inconsistencies. This is problematic when evaluating robustness w.r.t. different input elevations, since it is not possible to have the same test views for all input elevations. We hence use test views with a 15 degree offset the from input elevation, and normalize the consistency scores using an upper bound defined by the ground truth multi-view images. See the appendix for the normalization formula.

4. Experiments

In this section, we first validate our proposed metrics and then present our evaluation analysis of 12 typical MVGs. We then further systematically investigate some key design choices of MVGs and propose a method that outperforms all existing baselines in terms of 3D consistency.

4.1. Validating MVGBench metrics

3D Consistency. A good 3D consistency metric should be invariant to the number of views used and the specific camera setups. Therefore, using ground truth views (perfectly 3D consistent), we report our 3D consistency metrics varying these aspects. As can be seen from Tab. 1, the deviation is less than 8% of the average score across all variants of our metric. Note that the score differs from the theoretical upper bound due to slight inaccuracies in 3DGS fitting. However, these numbers indicate that such inaccuracies are negligible and are not affected by specific method setups. Hence, we can conclude that our metric scores vary only due to the 3D inconsistency of multi-view images.

#views	camera	CD↓	depth↓	cPSNR↑	cSSIM↑	cLPIPS↓
16	[17]	1.993	6.941	30.281	0.924	0.034
18	[9]	2.119	7.974	30.688	0.934	0.030
20	[37]	2.133	6.977	30.448	0.925	0.031
20	[54]	2.091	6.350	30.800	0.932	0.029
relative std.		0.026	0.082	0.006	0.004	0.051

Table 1. Our 3D consistency metrics are **invariant to number of views and different camera rendering settings**.

VLM metrics. We propose four metrics based on a pre-trained VLM [8] to evaluate image quality and semantic consistency. We conduct a user study to verify if they align with human perception. We randomly select 400 generated images from 5 methods in GSO [12] and CO3D [45] datasets and ask 10 users to answer the exact same verification questions as the VLM (yes means the image passes the check). Notably, the average scores (percentage of yes) reported by the users strongly correlate with the one from the VLM, see Fig. 5. We can hence conclude that our proposed VLM metric is faithful to human perception.

oFID score. We propose oFID score that measures how well the generated image of a specific object matches the distribution of multi-views of that object. Hence, instead of computing FID against the full dataset of all objects, we compute FID per object instance, and then average. To verify that our proposed oFID aligns better with human perception, we perform user studies to compare the method rankings based on oFID, FID and humans. We ask users to rank methods pair-wise based on alignment with the input image. We select 10 pairs of methods and for each pair, 40 random examples are selected and evaluated by 10 users. We report the percentage of ranking matches between FID or our oFID and human rankings: 0.77 (oFID), 0.50 (dataset FID). Clearly, oFID is much better aligned with human rankings. Our oFID and human rankings are strongly correlated and have a Pearson and Spearman coefficient of 0.69 and 0.67. We show example user study questions in the appendix.

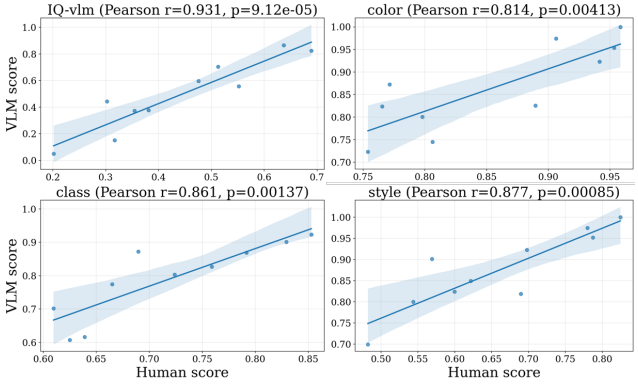

Figure 5. **Validating vision language model (VLM) based metrics.** Our VLM metrics strongly correlate with human perception (Pearson coefficient confidence interval: 0.95).

4.2. Evaluation results and observations

4.2.1. Evaluation setup

We evaluate open-sourced multi-view generation methods for object-level image generation. We exclude methods that only generate fixed few-views such as ImageDream [56], Zero123++ [48], Wonder3D [39] and [61]. Hence, we evaluate the following 12 SoTA methods: Zero123 [37], zero123-xl [11], Vivid123 [30], EpiDiff [21], Free3D [80], MVDFusion [17], ViewFusion [74], EscherNet [28], SyncDreamer [38], V3D [9], Hi3D [72], SV3D[54]. See the appendix for details about the individual input and output setups. We also add a new baseline method that leverages the best practices of MVGs, named ours (detailed in Sec. 4.3).

4.2.2. Observations

We present the summarized analysis in this section. The full evaluation tables can be found in the appendix.

Trade-off between 3D consistency and quality. We plot the 3D consistency (cPSNR) versus image quality (IQ-vlm) on our CO3D [45] and GSO [12] test set in Fig. 6. It can be seen that the top-right (best in both consistency and quality) region is unoccupied. Methods like Zero123 [37], Vivid123 [30] produce images that achieve good image

Method	CO3D [45] (real)			GSO [12] (synthetic)		
	CD ↓	cPSNR ↑	IQ-vlm ↑	CD ↓	cPSNR ↑	IQ-vlm ↑
Ours	**3.02**	**25.82**	0.29	3.15	**28.93**	0.82
SV3D-tune	3.32	24.70	0.29	3.24	27.95	0.82
SyncDreamer	3.04	25.30	0.12	**2.99**	26.83	0.53
SV3D	3.48	23.72	0.29	3.47	26.75	0.77
Hi3D	5.60	20.92	0.35	3.29	24.60	**0.87**
Eschernet	5.14	20.34	0.26	4.34	23.89	0.57
V3D	12.24	15.87	0.32	4.25	23.83	0.81
ViewFusion	10.45	16.39	0.33	5.33	22.34	0.63
MVDFusion	5.77	17.50	0.19	4.77	21.44	0.48
Free3d	11.15	14.42	0.32	6.03	20.26	0.73
EpiDiff	7.71	15.66	0.31	5.77	20.28	0.77
Vivid123	9.81	15.31	**0.49**	7.57	21.74	0.63
Zero123	12.06	13.16	0.38	10.99	17.37	0.73
Zero123-xl	12.58	12.97	0.34	15.40	17.10	0.72

Table 2. **Evaluation results on GSO and CO3D**. The performance gap between synthetic and real data is large, especially on the image quality aspect (IQ-vlm).

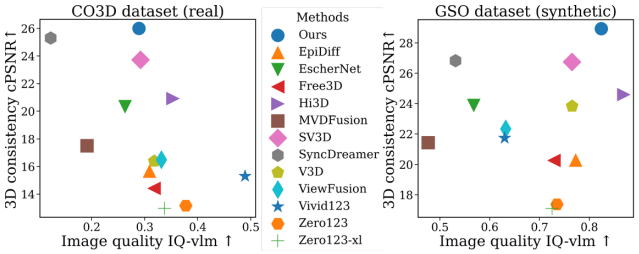

Figure 6. **Trade-off between 3D consistency and image quality**. No method can achieve the best performance in both dimensions.

quality, but are not 3D consistent. Conversely, methods like SyncDreamer [38], EscherNet [28] and MVDFusion [17] can generate 3D consistent images at the cost of lower quality. We observe that methods are either consistent but lack detail, or have detail that improve quality but difficult to be consistent. More recent video-based methods SV3D [54] and ours balance 3D consistency and quality better, see example images in Fig. 4. It is also visible in Fig. 1 where no method can reach the best score on all dimensions.

Large gap between synthetic and real images. We report the key metrics in 3D consistency (geometry+texture) and image quality (IQ-vlm) of all methods in Tab. 2. The performance on real images (CO3D [45]) is considerably inferior to the performance on synthetic images (GSO [12]), especially in the (IQ-vlm). We also show comparisons in Fig. 4 where the generated images are much worse for real images. More examples are presented in the appendix.

Methods are not robust to input perturbations. We plot the error versus different input light intensity, azimuth and elevation degrees, alongside representative examples in Fig. 7. Some methods are sensitive to dark light or specific azimuth and none of them are robust to elevations. We attribute this to limited pose variation in training. In contrast, our method trained on diverse renderings is more robust especially wrt. different lighting and azimuth angles.

Methods struggle to handle fine structures or textures. To identify the most challenging images for each method, we rank the input images based on 3D consistency score. Objects with complex and fine-grained geometric structures or textures such as bicycle, flowers, text boxes are the most challenging. One reason is that the autoencoder for latent diffusion [46] destroys high-frequency details and simply passing images through the autoencoder already degrades 3D consistency, see analysis in the appendix.

4.3. MVG design choices

What makes a MVG 3D consistent? With our benchmark suite, we can study this question from a fair and unified perspective. We classify different design choices into four categories: **a) Camera pose embedding.** Most methods [37, 54] adopt simple MLP based embedding while EscherNet proposes to use CaPE and Free3D adopts Plucker ray-based embedding. **b) Input image encoder.** Most methods [37, 38, 54] use CLIP [44] to encode input image while EscherNet [28] adopts ConvNextV2 [59] encoder to extract fine-grained features. We also consider the DINOv2 [42] encoder. **c) Interaction between target view features.** Recent methods generate multiple target views together and compute different attentions between target view features. Due to resource limits, we only compare the synchronized 3D convolution from SyncDreamer [38] and spatial-temporal attention from SV3D [54]. **d) How much training data is needed** to train a good MVG? The training data used in prior methods range from 40k to 800k objects but it is unclear how much is actually needed.

We adopt SV3D as our base model, which balances well between consistency and quality, and study the contribution of these different design choices. SV3D uses simple MLP to embed camera pose and the input image is encoded with CLIP. It generates 21 images together and uses the spatial-temporal attention pretrained for video generation.

Camera embedding. We add the camera positional encoding (CaP) from either EscherNet [28] or the ray conditional network (RCN) from Free3D [80] to SV3D. We fine tune the modified network for 26k steps. To rule out the effect of training data, we also fine tune SV3D in the same dataset and the results are reported in Tab. 3 b-e. It can be seen that both Cap (Tab. 3d) and RCN (Tab. 3c) are more effective than simple MLP based embedding (Tab. 3b). RCN is better than CaP and interestingly, combing both leads to a worse result. However, when replacing CLIP with convolution encoder, the difference bewteen CaP (Tab. 3h) and RCN (Tab. 3f) is small.

Input image encoder. We use SV3D combined with CaP from EscherNet as our base model and replace the CLIP input image encoder with ConvNextV2 or DINOv2 encoder. We choose the encoder that produces exactly same feature dimension as the CLIP encoder used in SV3D. Hence the network architecture is exactly the same except for the feature used for cross attention. We fine tune the model for 50k steps to adapt to the new input image feature and results are reported in Tab. 3. It can be seen that both DI-

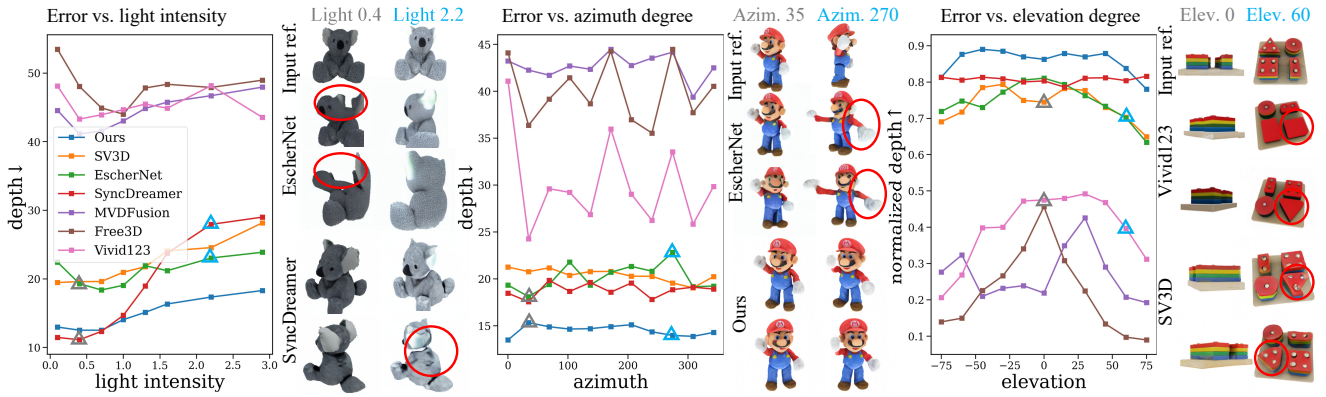

Figure 7. **Robustness w.r.t different light intensity, azimuth and elevation angles.** Some methods (EscherNet [28]) are sensitive to dark lighting while others (SyncDreamer [38]) are sensitive to strong lighting. Some methods (EscherNet, Vivid123 [30]) are also sensitive to the input azimuth angles and none of the methods are robust to higher elevation angles.

Model	CD↓	depth↓	cPSNR↑	cSSIM↑	cLPIPS↓
a. SV3D pretrained	3.472	19.651	26.751	0.865	0.070
b. SV3D fine-tuned	3.342	16.493	27.708	0.876	0.061
c. +RCN†	3.212	15.671	28.257	0.889	0.055
d. +CaP†	3.244	15.528	27.980	0.884	0.057
e. +RCN†+Cap†	3.246	15.369	27.809	0.884	0.058
f. +RCN†+Cov‡	**3.127**	**13.862**	28.615	0.890	0.052
g. +CaP†+DINOv2‡	3.146	14.301	28.507	0.891	0.053
h. +CaP†+Cov‡ (**Ours**)	3.154	14.204	**28.934**	**0.897**	0.052
i. +CaP†+Cov‡+sync♮	3.145	13.919	28.881	0.895	**0.051**

Table 3. **Investigating different design choices** added on top of SV3D [54]: camera embedding (†), input image encoder (‡), and multi-view feature syncronization (♮). Results on GSO.

	#Objs.	CD↓	depth↓	cPSNR↑	cSSIM↑	cLPIPS↓
GSO (syn.)	10k	3.354	16.919	27.765	0.881	0.061
	50k	3.186	13.504	28.053	0.881	0.056
	100k	**3.148**	**13.351**	28.142	0.885	0.053
	150k	3.154	14.204	**28.934**	**0.897**	**0.052**
CO3D (real)	10k	3.349	21.703	24.261	0.855	0.079
	50k	3.176	17.556	25.118	0.866	0.068
	100k	3.166	17.819	25.131	0.868	0.067
	150k	**3.099**	**16.942**	**25.994**	**0.880**	**0.062**

Table 4. **The effect of training data amount** on 3D consistency when replacing CLIP with the convolution encoder. More data improves mainly the generalization to real images (CO3D [45]).

NOv2 (Tab. 3g) and ConvNextV2 (Tab. 3h) are better than the CLIP encoder (Tab. 3b), while the difference between two vision encoders is small. We also show some results in Fig. 4 where our model with ConvNextV2 encoder can preserve better details from input than CLIP based SV3D. Given that ConvNextV2 is more efficient than DINO model, we propose using ConvNextV2 to replace the CLIP encoder.

Interaction between target views. SyncDreamer [38] computes a synchronized 3D feature volume and derive multi-view images from it, leading to strong 3D consistency (Tab. 2). We combine such 3D feature volume with the video based SV3D + CaP and ConvNextV2 encoding and results are shown in Tab. 3. Surprisingly, the performance gain with the additional synchronized 3D convolution is small. We hypothesize that the explicit 3D feature interaction from SyncDreamer is already implicitly achieved by the dense spatial-temporal attention of SV3D. Hence **we adopt SV3D+Cap+Cov as the model for `Ours`**.

Training data amount. Another important aspect to consider is the amount of data needed. Since it is not computationally feasible to retrain every single method with different amounts of data, we study the effect of data amount on performance when adding a new module to a pretrained model. Specifically, we start with SV3D+CaP and replace the CLIP image encoder with a convolution image encoder and fine tune it on different amount of objaverse objects. The 3D consistency results are shown in Tab. 4. It can be seen that 50K objects are sufficient to achieve good performance on synthetic images (GSO100) and improvement is small after that. By contrast, performance continues to improve with more data when testing on real images.

5. Conclusion

We present MVGBench, a comprehensive benchmark suite for evaluating multi-view generation models (MVGs). We unify 10 metric dimensions to evaluate the 3D consistency, image quality, and semantic consistency of generated images. Experiments show that our 3D consistency metric reports meaningful scores and fair comparisons of methods. Our quality and semantic consistency metrics align well with human perception: Pearson scores ranging from 0.69 to 0.92. We evaluate 12 SoTA MVGs and find that there is a trade-off between 3D consistency and image quality, and no method can achieve the best performance in all dimensions. We also observe that large performance gap persists between synthetic and real images, and most methods are not robust to different elevations, azimuths, or lightings.

We investigate four key design choices of MVGs, including camera embedding, input image encoding, attention mechanism, and training data amount. We find that the con-

volution encoder can preserve fine details of the input image, resulting in better 3D consistency. Explicit 3D convolution does not provide much improvement on top of video-based models. While 50k training objects are sufficient for synthetic input, more data is necessary to improve generalization to real data. We will publicly release our benchmark suite and fine-tuned models.

Acknowledgements. We thank RVH group members [1] for their helpful discussions. This work is funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - 409792180 (Emmy Noether Programme, project: Real Virtual Humans), and German Federal Ministry of Education and Research (BMBF): Tübingen AI Center, FKZ: 01IS18039A, and Amazon-MPI science hub. Gerard Pons-Moll is a Professor at the University of Tübingen endowed by the Carl Zeiss Foundation, at the Department of Computer Science and a member of the Machine Learning Cluster of Excellence, EXC number 2064/1 – Project number 390727645.

References

[1] http://virtualhumans.mpi-inf.mpg.de/people.html. 9

[2] Stability AI. Stable zero123: Quality 3d object generation from single images. https://stability.ai/news/stable-zero123-3d-generation, 2023. 2, 3, 4, 5

[3] Mohammad Asim, Christopher Wewer, Thomas Wimmer, Bernt Schiele, and Jan Eric Lenssen. Met3r: Measuring multi-view consistency in generated images. In *arXiv preprint*, 2025. 3

[4] Rodrigo Benenson, Mohamed Omran, Jan Hosang, and Bernt Schiele. Ten years of pedestrian detection, what have we learned?, 2014. 3

[5] Bharat Lal Bhatnagar, Xianghui Xie, Ilya Petrov, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. Behave: Dataset and method for tracking human object interactions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[6] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 3

[7] Jonathan Chang, Barret Zoph, Andrew Dai, Zihang Chen, and Quoc V Le. Muse: Text-to-image generation via masked generative transformers. *arXiv preprint arXiv:2301.00704*, 2023. 2

[8] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024. 5, 6, 2

[9] Zilong Chen, Yikai Wang, Feng Wang, Zhengyi Wang, and Huaping Liu. V3d: Video diffusion models are effective 3d generators. *arXiv preprint arXiv:2403.06738*, 2024. 3, 6, 2, 4, 5

[10] Zihang Chen, Ming Zhao, et al. Llamagen: Autoregressive large language models for visual image generation. *arXiv preprint arXiv:2406.06525*, 2024. 2

[11] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, Eli VanderBilt, Aniruddha Kembhavi, Carl Vondrick, Georgia Gkioxari, Kiana Ehsani, Ludwig Schmidt, and Ali Farhadi. Objaverse-XL: A universe of 10m+ 3d objects. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. 6, 3, 4, 5

[12] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B. McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2022. 5, 6, 7, 3, 4

[13] Zongcai Fei. Stable-edit: Text-based real image editing with stable diffusion models. https://github.com/feizc/Stable-Edit, 2023. 2

[14] Ruiqi Gao, Aleksander Hołyński, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul Srinivasan, Jonathan T. Barron, and Ben Poole. Cat3d: Create anything in 3d with multi-view diffusion models. *arXiv preprint arXiv:2405.10314*, 2024. 3

[15] Ruiqi Gao, Aleksander Hołyński, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul Srinivasan, Jonathan T. Barron, and Ben Poole. Mvreward: Better aligning and evaluating multi-view diffusion models with human preferences. *arXiv preprint arXiv:2412.06614*, 2024. 3

[16] Tomas Hodan, Frank Michel, Eric Brachmann, Wadim Kehl, Anders Glent Buch, Dirk Kraft, Bertram Drost, Joel Vidal, Stephan Ihrke, Xenophon Zabulis, Caner Sahin, Fabian Manhardt, Federico Tombari, Tae-Kyun Kim, Jiri Matas, and Carsten Rother. Bop: Benchmark for 6d object pose estimation, 2018. 3

[17] Hanzhe Hu, Zhizhuo Zhou, Varun Jampani, and Shubham Tulsiani. MVD-Fusion: Single-view 3D via Depth-consistent Multi-view Generation. 3, 5, 6, 7, 4

[18] Hanzhe Hu, Tianwei Yin, Fujun Luan, Yiwei Hu, Hao Tan, Zexiang Xu, Sai Bi, Shubham Tulsiani, and Kai Zhang. Turbo3d: Ultra-fast text-to-3d generation, 2024. 3

[19] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH 2024 Conference Papers*. Association for Computing Machinery, 2024. 1

[20] Yangyi Huang, Hongwei Yi, Yuliang Xiu, Tingting Liao, Jiaxiang Tang, Deng Cai, and Justus Thies. TeCH: Text-guided Reconstruction of Lifelike Clothed Humans. In *International Conference on 3D Vision (3DV)*, 2024. 2

[21] Zehuan Huang, Hao Wen, Junting Dong, Yaohui Wang, Yangguang Li, Xinyuan Chen, Yan-Pei Cao, Ding Liang, Yu Qiao, Bo Dai, and Lu Sheng. EpiDiff: Enhancing Multi-View Synthesis via Localized Epipolar-Constrained Diffusion. 3, 6, 4, 5

[22] Zehuan Huang, Yuanchen Guo, Haoran Wang, Ran Yi, Lizhuang Ma, Yan-Pei Cao, and Lu Sheng. Mv-adapter: Multi-view consistent image generation made easy. *arXiv preprint arXiv:2412.03632*, 2024. 3

[23] Ziqi Huang, Yilun Sheng, Yujian Liu, Yujie Lu, and Wenhu Chen. Vbench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 3

[24] Sadeep Jayasumana, Srikumar Ramalingam, Andreas Veit, Daniel Glasner, Ayan Chakrabarti, and Sanjiv Kumar. Rethinking fid: Towards a better evaluation metric for image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9307–9315, 2024. 4

[25] Yoonwoo Jeong, Jinwoo Lee, Chiheon Kim, Minsu Cho, and Doyup Lee. NVS-Adapter: Plug-and-Play Novel View Synthesis from a Single Image, 2024. 3

[26] Yash Kant, Aliaksandr Siarohin, Ziyi Wu, Michael Vasilkovsky, Guocheng Qian, Jian Ren, Riza Alp Guler, Bernard Ghanem, Sergey Tulyakov, and Igor Gilitschenski. SPAD: Spatially Aware Multi-View Diffusers. 2, 3

[27] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4), 2023. 3, 1

[28] Xin Kong, Shikun Liu, Xiaoyang Lyu, Marwan Taher, Xiaojuan Qi, and Andrew J Davison. EscherNet: A Generative Model for Scalable View Synthesis. 2, 3, 5, 6, 7, 8, 4

[29] Max Ku, Tianle Li, Kai Zhang, Yujie Lu, Xingyu Fu, Wenwen Zhuang, and Wenhu Chen. Imagenhub: Standardizing the evaluation of conditional image generation models. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR)*, 2024. 3

[30] Jeong-gi Kwak, Erqun Dong, Yuhe Jin, Hanseok Ko, Shweta Mahajan, and Kwang Moo Yi. ViVid-1-to-3: Novel View Synthesis with Video Diffusion Models. 3, 5, 6, 8, 4

[31] Lingen Li, Zhaoyang Zhang, Yaowei Li, Jiale Xu, Wenbo Hu, Xiaoyu Li, Weihao Cheng, Jinwei Gu, Tianfan Xue, and Ying Shan. Nvcomposer: Boosting generative novel view synthesis with multiple sparse and unposed images, 2024. 3

[32] Peng Li, Yuan Liu, Xiaoxiao Long, Feihu Zhang, Cheng Lin, Mengfei Li, Xingqun Qi, Shanghang Zhang, Wenhan Luo, Ping Tan, et al. Era3d: High-resolution multiview diffusion using efficient row-wise attention. *arXiv preprint arXiv:2405.11616*, 2024. 3

[33] Mingxiang Liao, Qixiang Ye, Wangmeng Zuo, Fang Wan, Tianyu Wang, Yuzhong Zhao, Jingdong Wang, Xinyu Zhang, et al. Evaluation of text-to-video generation models: A dynamics perspective. *Advances in Neural Information Processing Systems*, 37:109790–109816, 2024. 3

[34] Tingting Liao, Hongwei Yi, Yuliang Xiu, Jiaxiang Tang, Yangyi Huang, Justus Thies, and Michael J. Black. TADA! Text to Animatable Digital Avatars. In *International Conference on 3D Vision (3DV)*, 2024. 2

[35] Kunhao Liu, Fangneng Zhan, Yiwen Chen, Jiahui Zhang, Yingchen Yu, Abdulmotaleb El Saddik, Shijian Lu, and Eric Xing. Stylerf: Zero-shot 3d style transfer of neural radiance fields. 2023. 3

[36] Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Mukund T, Zexiang Xu, and Hao Su. One-2-3-45: Any Single Image to 3D Mesh in 45 Seconds without Per-Shape Optimization. In *Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2023. 2, 3

[37] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object, 2023. 2, 3, 6, 7

[38] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Generating multiview-consistent images from a single-view image. *arXiv preprint arXiv:2309.03453*, 2023. 2, 3, 5, 6, 7, 8, 4

[39] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. Wonder3d: Single image to 3d using cross-domain diffusion. *arXiv preprint arXiv:2310.15008*, 2023. 3, 6

[40] Yuanxun Lu, Jingyang Zhang, Shiwei Li, Tian Fang, David McKinnon, Yanghai Tsin, Long Quan, Xun Cao, and Yao

Yao. Direct2.5: Diverse text-to-3d generation via multi-view 2.5d diffusion. *Computer Vision and Pattern Recognition (CVPR)*, 2024. 2, 3

[41] Saswat Subhajyoti Mallick, Rahul Goel, Bernhard Kerbl, Markus Steinberger, Francisco Vicente Carrasco, and Fernando De La Torre. Taming 3dgs: High-quality radiance fields with limited resources. In *SIGGRAPH Asia 2024 Conference Papers*, New York, NY, USA, 2024. Association for Computing Machinery. 1

[42] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2024. 3, 7

[43] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv*, 2022. 2

[44] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*. PMLR, 2021. 3, 4, 5, 7

[45] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 5, 6, 7, 8, 3, 4

[46] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 2, 3, 7

[47] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. Stable diffusion 3.5: High-resolution image synthesis with latent diffusion models. *Stability AI*, 2024. 2, 3

[48] Ruoxi Shi, Hansheng Chen, Zhuoyang Zhang, Minghua Liu, Chao Xu, Xinyue Wei, Linghao Chen, Chong Zeng, and Hao Su. Zero123++: a single image to consistent multi-view diffusion base model, 2023. 2, 3, 6

[49] Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv:2308.16512*, 2023. 2, 3

[50] Yujun Shi, Chuhui Xue, Jiachun Pan, Wenqing Zhang, Vincent YF Tan, and Song Bai. Dragdiffusion: Harnessing diffusion models for interactive point-based image editing. *arXiv preprint arXiv:2306.14435*, 2023. 2

[51] Deqing Sun, Stefan Roth, and Michael J. Black. Secrets of optical flow estimation and their principles. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2432–2439, 2010. 3

[52] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. *arXiv preprint arXiv:2402.05054*, 2024. 3, 4, 2

[53] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. 2024. 2

[54] Vikram Voleti, Chun-Han Yao, Mark Boss, Adam Letts, David Pankratz, Dmitry Tochilkin, Christian Laforte, Robin Rombach, and Varun Jampani. SV3D: Novel Multi-view Synthesis and 3D Generation from a Single Image using Latent Video Diffusion, 2024. 2, 3, 5, 6, 7, 8, 4

[55] Jianyi Wang, Zongsheng Yue, Shangchen Zhou, Kelvin CK Chan, and Chen Change Loy. Exploiting diffusion prior for real-world image super-resolution. *arXiv preprint arXiv:2305.07015*, 2023. 2

[56] Peng Wang and Yichun Shi. Imagedream: Image-prompt multi-view diffusion for 3d generation. *arXiv preprint arXiv:2312.02201*, 2023. 2, 3, 6

[57] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *arXiv preprint arXiv:2305.16213*, 2023. 2

[58] Daniel Watson, William Chan, Ricardo Martin-Brualla, Jonathan Ho, Andrea Tagliasacchi, and Mohammad Norouzi. Novel View Synthesis with Diffusion Models, 2022. 3

[59] Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. Convnext v2: Co-designing and scaling convnets with masked autoencoders. *arXiv preprint arXiv:2301.00808*, 2023. 3, 5, 7

[60] Haoyu Wu, Meher Gitika Karumuri, Chuhang Zou, Seungbae Bang, Yuelong Li, Dimitris Samaras, and Sunil Hadap. Direct and explicit 3d generation from a single image, 2024. 2

[61] Haoyu Wu, Meher Gitika Karumuri, Chuhang Zou, Seungbae Bang, Yuelong Li, Dimitris Samaras, and Sunil Hadap. Direct and Explicit 3D Generation from a Single Image, 2024. 6

[62] Rongyuan Wu, Lingchen Sun, Zhiyuan Ma, and Lei Zhang. One-step effective diffusion network for real-world image super-resolution. *arXiv preprint arXiv:2406.08177*, 2024. 2

[63] Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Liang Pan Jiawei Ren, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, Dahua Lin, and Ziwei Liu. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 5, 3, 4

[64] Tong Wu, Guandao Yang, Zhibing Li, Kai Zhang, Ziwei Liu, Leonidas Guibas, Dahua Lin, and Gordon Wetzstein. Gpt-4v(ision) is a human-aligned evaluator for text-to-3d generation. In *CVPR*, 2024. 3

[65] Xianghui Xie, Xi Wang, Nikos Athanasiou, Bharat Lal Bhatnagar, Chun-Hao P. Huang, Kaichun Mo, Hao Chen, Xia Jia, Zerui Zhang, Liangxian Cui, Xiao Lin, Bingqiao Qian, Jie Xiao, Wenfei Yang, Hyeongjin Nam, Daniel Sungho Jung, Kihoon Kim, Kyoung Mu Lee, Otmar Hilliges, and Gerard Pons-Moll. Rhobin challenge: Reconstruction of human object interaction, 2024. 3

[66] Jinbo Xing, Menghan Xia, Yong Zhang, Haoxin Chen, Wangbo Yu, Hanyuan Liu, Xintao Wang, Tien-Tsin Wong, and Ying Shan. Dynamicrafter: Animating open-domain images with video diffusion priors. *arXiv preprint arXiv:2310.12190*, 2023. 3

[67] Dejia Xu, Weili Nie, Chao Liu, Sifei Liu, Jan Kautz, Zhangyang Wang, and Arash Vahdat. CamCo: Camera-Controllable 3D-Consistent Image-to-Video Generation, 2024. 3

[68] Jiazheng Xu, Yuxuan Du, Yilun Zhao, Yifan Xu, Lei Li, Maosong Sun, Zhiyuan Liu, and Yang Liu. Imagereward: Learning and evaluating human preferences for text-to-image generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023. 3

[69] Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191*, 2024. 3

[70] Yuxuan Xue, Xianghui Xie, Riccardo Marin, and Gerard. Pons-Moll. Gen-3Diffusion: Realistic Image-to-3D Generation via 2D & 3D Diffusion Synergy. 2024. 2, 3

[71] Yuxuan Xue, Xianghui Xie, Riccardo Marin, and Gerard Pons-Moll. Human 3diffusion: Realistic avatar creation via explicit 3d consistent diffusion models. In *Arxiv*, 2024. 2, 3

[72] Haibo Yang, Yang Chen, Yingwei Pan, Ting Yao, Zhineng Chen, Chong-Wah Ngo, and Tao Mei. Hi3d: Pursuing high-resolution image-to-3d generation with video diffusion models. In *ACM MM*, 2024. 3, 6, 4, 5

[73] Tao Yang, Rongyuan Wu, Peiran Ren, Xuansong Xie, and Lei Zhang. Pixel-aware stable diffusion for realistic image super-resolution and personalized stylization. *arXiv preprint arXiv:2308.14469*, 2023. 2

[74] Xianghui Yang, Yan Zuo, Sameera Ramasinghe, Loris Bazzani, and Gil Avraham. ViewFusion: Towards Multi-View Consistency via Interpolated Denoising. 6, 3, 4, 5

[75] Xianggang Yu, Mutian Xu, Yidan Zhang, Haolin Liu, Chongjie Ye, Yushuang Wu, Zizheng Yan, Tianyou Liang, Guanying Chen, Shuguang Cui, and Xiaoguang Han. Mvimgnet: A large-scale dataset of multi-view images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 5, 3

[76] Zehao Yu, Anpei Chen, Binbin Huang, Torsten Sattler, and Andreas Geiger. Mip-splatting: Alias-free 3d gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19447–19456, 2024. 1

[77] Zehao Yu, Torsten Sattler, and Andreas Geiger. Gaussian opacity fields: Efficient high-quality compact surface reconstruction in unbounded scenes. *arXiv:2404.10772*, 2024. 1

[78] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023. 2

[79] Qinsheng Zhang, Jiaming Song, Xun Huang, Yongxin Chen, and Ming yu Liu. Diffcollage: Parallel generation of large content with diffusion models. In *CVPR*, 2023. 2

[80] Chuanxia Zheng and Andrea Vedaldi. Free3D: Consistent Novel View Synthesis Without 3D Representation. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9720–9731, Seattle, WA, USA, 2024. IEEE. 3, 6, 7, 4, 5

MVGBench: a Comprehensive Benchmark for Multi-view Generation Models

Supplementary Material

In this supplementary, we first discuss the implementation details, including our MVGBench metrics (Sec. 6.1) and evaluation experiment setups (Sec. 6.2). We then show additional evaluation result and analysis in Sec. 7, and conclude with discussion of limitations.

6. Implementation Details

We discuss the details of our metrics and experiment setups. Our benchmark suite and pre-trained models will be publicly released.

6.1. MVGBench Metric Implementation

View sets split. For 3D consistency metric, we split the generated multi-view images into two subsets and fit 3DGS separately to them. We allow small overlap when the total number of generated views is small. There are three different number of output views for all the methods we evaluated: 16, 18, 21, see Tab. 5. The view indices for the two subsets are: 1). Output 16 views: [0, 2, 4, 5, 6, 8, 9, 10, 11, 12, 14], [1, 3, 5, 6, 7, 9, 11, 12, 13, 14, 15]. 2). Output 18 views: [0, 1, 2, 4, 6, 8, 10, 12, 14, 16], [0, 1, 2, 3, 5, 7, 9, 11, 13, 15, 17]. 3). Output 21 views: the first (input) view is excluded and rest is divided into two non-overlap views, namely [0, 2, 4, 6, 8, 10, 12, 14, 16, 18], [1, 3, 5, 7, 9, 11, 13, 15, 17, 19].

3DGS optimization. We use the original version of 3DGS [27] for optimization. We explored more advanced version of 3DGS but found that they are either less accurate for object level multi-views [19, 41, 76] or the runtime is too long [77]. We hence stick to the original 3DGS version and randomly sample 100k points from unit cube [-1, 1] to initialize the Gaussians and optimize for 10k steps. White background is used during optimization as all methods generate images with white background.

Test view rendering. We render the optimized 3DGSs into RGB and depth images to compute the depth, cPSNR, cSSIM, cLPIPS metrics. To produce comparable numbers, the test views have to be the same for two 3DGSs and across all methods, for the same test object. The test views should be diverse so that is does not favor output elevation angles specific to some methods while it should also be close to the views used to fit 3DGS, otherwise the calculated scores are dominated by 3DGS fitting error instead of multi-view inconsistency. To this end, we use two setups to choose the views for rendering: a). Random views sampled from a fixed range, and b). Fixed views that differ 15 elevation degrees from generated multi-views. For both setup $K = 16$ views are used for rendering, and each object might have different test views but the same views are always used across methods for fair comparison.

Random test views are used for best setup performance, robustness w.r.t to lighting and azimuth conditions. As existing methods generate multi-view images with elevation angle ranging from 0 to 30 degrees (Tab. 5), we uniformly sample elevation from range [-15, 45], azimuth from [0, 360], and camera distance from [1.5, 1.9]. The field of view (Fov) is fixed at 42 degree such that it does not favor any of the methods evaluated.

Fixed test views are used for generalization to real images and robustness w.r.t to different elevation degrees. In these setups, the output elevation differ a lot and it is difficult to define a common range where 3DGS fitting also works well and we can sample elevation from. To this end, we take 8 views with equal azimuth distance from 8.5 to 360 degree and the elevation is 15 degree higher than the elevation of generated multi-views. The other 8 views have the same azimuth angles but the elevation is 15 degree lower than the output multi-view elevation. The fov and camera distance are fixed to 42 degree and 3.2m. We choose these azimuth, fov, and distance to not favor any methods. Note that this will lead to consistency scores that are not comparable across different output elevations, which address next.

Normalization of the consistency scores. The exact scores of our consistency metrics depend on the views used to render test images and the raw numbers are not directly comparable if the views are different. This is the case when we want to evaluate the robustness of a method w.r.t to different elevation angles (see discussion above). We hence propose to normalize the raw numbers using the upper bound scores obtained from ground truth multi-view images. Let $e_{\text{gt}}, e_{\text{mvg}}$ be the raw consistency score defined in Sec. 3.1 using MVG and GT images of the same camera views. The normalized error $e_{\text{mvg, n}}$ is computed as:

$$e^i_{\text{mvg, n}} = \begin{cases} \frac{e_{\text{mvg}}}{e_{\text{gt}}} & \text{if type}(e) \in \{\text{cPSNR, cSSIM}\}, \\ 1 - \frac{e_{\text{mvg}}}{\max e_{\text{mvg}}} & \text{if type}(e) \in \{\text{CD, depth, cLPIPS}\}. \end{cases} \tag{5}$$

here $\max e_{\text{mvg}}$ is the maximum error for this metric among our evaluated methods. This yields a score between 0 and 1 and it is always the higher the better. This normalization is also used to visualize the bottom plots in Fig. 1.

Prompt templates for VLM based metrics. We propose four metrics based on the pretrained 73B InternVL2.5 VLM [8]. The same model is used to obtain the reference attributes (Sec. 3.2) via three-round prompts given multi-view images. The three sequential prompts are: 1). "Here are images of a daily object, what is the appearance style of this object? Ignore the background, focus on the appearance, style and design instead of describing the object type, return the appearance style only and in less than 5 words.", 2). "Which object it is? Just return the class name, do not repeat question. Use daily used common words. If there are multiple possibilities, return like this: classname 1 or classname2 or classname3...", 3). "What is the main color(s) of this object? simply answer the color(s), summarize to less than 4 colors."

We then use the answers from these prompts as the reference attributes and evaluate the semantic consistency using the following templates: 1). `class`: "Is [obj cls] presented in this image? just answer yes or no." 2). `color`: "Does the object (possibly [obj cls]) shown in this image have the color(s): [color]? just answer yes or no. 3). "Is the appearance style of the object (possibly [obj cls]): [style]? just answer yes or no."

We also use the same model to asses the image quality (IQ-vlm) which we find align well with human perception. The prompt template is: "Is this image an overall high-quality image with good overall structure, good visual quality, nice color harmony, clear object and free of strange artifacts and distortions? just answer yes or no.".

Runtime performance. The most compute expensive steps in our evaluation pipeline are 3DGS optimization and VLM assessment, which takes around 76s (two subsets) and 12s per input image to finish on L40s GPU. In total it takes around 2.7 hours to evaluate 100 objects which is still reasonable. More advanced techniques such as better 3DGS initialization [9] or VLM inference via API call could be adopted to speed up evaluation. We leave these for future works.

6.2. Experiment Setups

User study. We conduct user studies to verify our oFID score and VLM based metrics, each with 400 questions answered by 10 users. As 400 questions are too many for one single user study survey, we divide it into 8 smaller surveys, each with 50 questions. We then recruit 10 users to finish each survey and no overlap is allowed for different surveys. Hence in total we have 80 different users to participate one user study. This ensures sufficient diversity and statistically meaningful results. We show example questions from our user studies in Fig. 8 and Fig. 9.

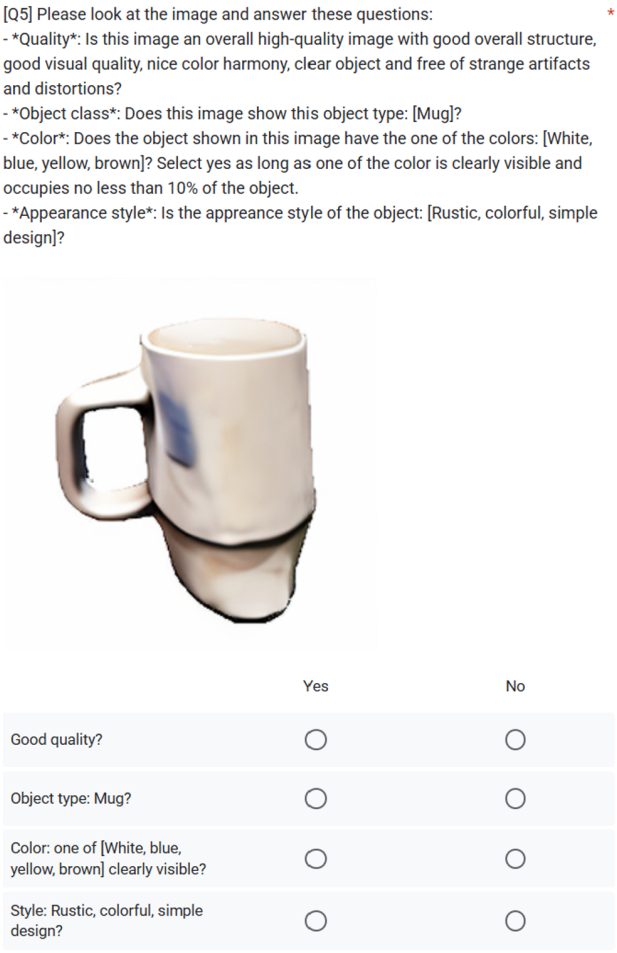

Figure 8. Example question from our user study on the alignment between our VLM based metrics and human preference.

Evaluation setup for existing methods. We show the input and output setups for all the methods we evaluate in the **best-setup performance** experiment in Tab. 5. We use ambience light of 1.0 and zero azimuth angle for this setup. The rendering setup is the same for **robustness evaluation** except for the attribute we want to evaluate (elevation, light intensity, and azimuth angle). For **generalization to real images**, we cannot control the rendering anymore hence we use the same input image crop for different methods, which has 0.2 margin from the object bounding box to image boarder. The number of output views of each method remain the same as in best-setup evaluation.

MVG design choice experiments. We use the 150k kiui objects filtered by LGM [52] as our training data. Following same camera parameters in SV3D [54], we render each object from 84 views and randomly pick 21 views at training time. We adopt the dynamic orbit rendering from SV3D which adds perturbations of azimuth and elevation angles to the equally distributed static orbit. We pre-compute the la-

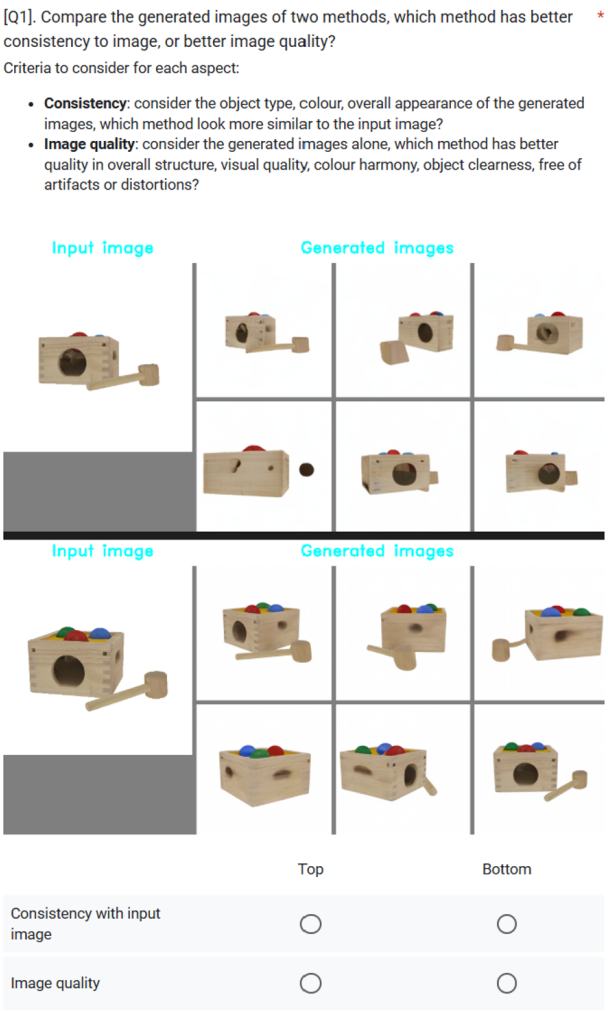

Figure 9. Example question from our user study on the alignment between our oFID score and human preference.

tent features of CLIP [44], SVD [46], DINOV2 [42] and ConvNextV2 [59] to speed up training. We use batch size of 64, learning rate of 2e-5 for all the experiments. The total training steps is 26k for camera embedding experiments and 50k for all other experiments.

7. Additional Results and Analysis

Full evaluation results. We show all scores of our MVGBench from all evaluated methods on four datasets in Tab. 6 (GSO [12]), Tab. 7 (Omni3D [63]), Tab. 8 (CO3D [45]), and Tab. 9 (MVImgnet [75]). It can be seen that our method achieves the best overall 3D consistency and on par performance on image quality and semantic consistency.

Methods struggle with fine-grained details. We rank the input images based on the 3D consistency score (cPSNR) from different methods and visualize the 10 common inputs that have the worst scores in Fig. 10. It can be seen

Methods	Fov	Elev.	Dist.(m)	#Out views
SyncDreamer based [17, 21, 38, 72, 80]	49.1	30	1.5	16
V3D [9]	60	0	2.0	18
SV3D [54], ours	33.8	12.5	2.35	21
Zero123 based [2, 11, 28, 30, 74]	49.1	0	1.85	21

Table 5. The input rendering (Fov, camera elevation degree and distance) and number of output views of each method for the best performance evaluation.

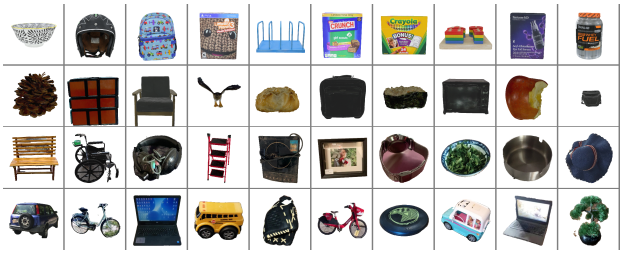

Figure 10. **The most challenging test images** from GSO[12], Omni3D[63], MVImgnet[75] and CO3D[45]. Methods produce the most 3D inconsistent images for these inputs due to their complex geometric structure or high frequency details.

that the common challenging images are the objects with complex and fine-grained geometric structures and textures such as bicycle, flowers, text boxes. Diving further into this problem, we find that the autoencoder used in all MVGs already destroys the high frequency structures after one single pass through the autoencoder. We show two examples in Fig. 11. To further understand the effect on 3D consistency, we send the ground truth multi-view images of 30 GSO objects [28] through the autoencoder of SV3D [54]. The consistency scores before and after the autoencoder are (CD / depth / cPSNR / cSSIM / cLPIPS): 2.58 / 9.82 / 31.94 / 0.94 / 0.03 (before), 2.69 / 9.05 / 30.29 / 0.92 / 0.04 (after). It can be clearly seen that the 3D texture consistency scores already degrade after single feedforward through the autoencoder. Interestingly, the depth error, sensitive to inconsistency in edges, decreases which indicates the images are indeed smoother with high-frequency details lost.

8. Limitations and Future Work

We present the first comprehensive benchmark to evaluate 3D consistency of object multi-view generation models. Despite robust to various settings, there are still limitations of our benchmark. First, our method cannot evaluate methods that generate very few views (<10) as the 3DGS fitting is very inaccurate and fitting error instead of multi-view inconsistency dominates our consistency scores. One possible solution is to replace 3DGS fitting with pre-trained models that can take few-views as input and directly regress 3DGS, such as LGM [52]. This however requires the model

Method	Geometry consistency		Texture consistency			Image quality		Semantic consistency		
	CD ↓	depth ↓	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	IQ-vlm ↑	class ↑	color ↑	style ↑
Ours	3.15	**14.20**	**28.93**	**0.90**	**0.05**	20.46	0.82	0.86	0.94	0.93
SyncDreamer[38]	**2.99**	17.29	26.83	0.87	0.07	22.72	0.53	0.84	0.96	0.94
SV3D-tune	3.34	16.49	27.71	0.88	0.06	19.06	0.80	0.89	0.95	**0.96**
SV3D[54]	3.47	19.65	26.75	0.86	0.07	21.31	0.77	0.85	0.92	0.93
Hi3D[72]	3.29	21.69	24.60	0.84	0.09	18.68	**0.87**	**0.89**	0.95	0.95
V3D[9]	4.25	28.08	23.84	0.81	0.12	21.20	0.77	0.86	0.96	0.91
EscherNet[28]	4.34	20.61	23.89	0.79	0.11	24.71	0.57	0.77	0.90	0.88
MVDFusion[17]	4.77	38.74	21.44	0.76	0.15	25.60	0.48	0.88	0.94	0.94
ViewFusion[74]	5.33	40.20	22.34	0.80	0.14	22.03	0.63	0.82	0.92	0.92
EpiDiff[21]	5.77	50.65	20.28	0.72	0.19	**16.53**	0.77	0.89	**0.97**	0.94
Free3D[80]	6.03	44.27	20.26	0.77	0.18	27.30	0.73	0.78	0.82	0.90
Vivid123[30]	7.57	43.97	21.74	0.81	0.18	38.91	0.63	0.66	0.78	0.80
Zero123[2]	10.99	63.72	17.37	0.67	0.29	21.35	0.73	0.82	0.90	0.93
Zero123-xl[11]	15.40	68.13	17.10	0.66	0.30	20.72	0.72	0.83	0.91	0.94

Table 6. Best setup performance on the GSO [12] dataset.

Method	Geometry consistency		Texture consistency			Image quality		Semantic consistency		
	CD ↓	depth ↓	PSNR ↑	SSIM ↑	LPIPS ↓	oFID ↓	IQ-vlm ↑	class ↑	color ↑	style ↑
Ours	2.98	**11.63**	**29.09**	**0.92**	**0.04**	15.47	**0.54**	0.77	0.85	**0.88**
SyncDreamer[38]	**2.93**	13.60	27.24	0.89	0.06	5.94	0.24	0.64	0.85	0.79
Hi3D[72]	3.13	17.63	25.25	0.88	0.08	16.07	0.55	0.74	0.87	0.84
SV3D-tune	3.16	14.11	27.69	0.91	0.05	15.00	0.51	0.79	0.88	0.89
SV3D[54]	3.46	19.46	26.02	0.88	0.07	17.60	0.50	0.69	0.85	0.85
V3D [9]	4.51	23.62	23.01	0.85	0.12	17.70	0.46	0.70	0.84	0.85
EscherNet[28]	5.01	23.25	21.87	0.77	0.14	22.39	0.41	0.60	0.80	0.79
MVDFusion[17]	5.67	47.96	19.04	0.76	0.19	26.89	0.21	0.69	0.83	0.82
EpiDiff[21]	6.78	57.31	18.37	0.73	0.21	**14.61**	0.52	**0.79**	**0.89**	0.88
ViewFusion[74]	7.88	54.32	17.90	0.73	0.24	16.96	0.44	0.68	0.85	0.87
Free3D[80]	8.02	52.58	16.97	0.72	0.25	23.78	0.50	0.61	0.77	0.79
Zero123-xl[11]	13.67	70.17	13.64	0.60	0.39	17.86	0.51	0.67	0.84	0.86
Zero123[2]	14.17	70.32	14.15	0.62	0.38	17.62	0.51	0.69	0.84	0.88
Vivid123[30]	14.31	56.07	17.80	0.76	0.26	27.98	0.50	0.56	0.74	0.74

Table 7. Best setup performance on the Omni3D [63] dataset.

Method	Geometry consistency		Texture consistency			Image quality		Semantic consistency		
	CD ↓	depth ↓	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	IQ-vlm ↑	class ↑	color ↑	style ↑
Ours	3.10	16.94	**25.99**	**0.88**	**0.06**	23.40	0.29	0.80	**0.86**	0.82
SyncDreamer[38]	**3.04**	**13.48**	25.30	0.88	0.06	30.96	0.12	0.69	0.83	0.70
SV3D-tune	3.43	19.99	24.32	0.85	0.08	21.71	0.26	0.82	0.84	0.83
SV3D[54]	3.48	25.80	23.72	0.87	0.13	24.19	0.29	0.76	0.87	0.78
EscherNet[28]	5.14	26.46	20.34	0.71	0.14	28.54	0.26	0.71	0.79	0.72
Hi3D[72]	5.60	31.09	20.92	0.81	0.12	25.51	0.35	0.75	0.82	0.75
MVDFusion[17]	5.77	47.43	17.50	0.71	0.20	27.16	0.19	0.75	0.82	0.78
EpiDiff[21]	7.71	58.58	15.66	0.64	0.26	**20.58**	0.31	0.84	0.86	0.82
ViewFusion[74]	7.75	49.76	16.49	0.77	0.29	22.10	0.33	0.82	0.85	0.82
Vivid123[30]	9.81	56.38	15.31	0.69	0.29	35.89	**0.49**	0.70	0.76	0.72
V3D[9]	10.45	58.71	16.39	0.71	0.26	28.76	0.32	0.72	0.85	0.77
Free3D[80]	11.15	60.95	14.42	0.76	0.33	32.84	0.32	0.71	0.75	0.75
Zero123[2]	12.06	64.74	13.16	0.55	0.38	21.22	0.38	0.84	0.87	**0.86**
Zero123-xl[11]	12.58	66.99	12.97	0.54	0.38	20.83	0.34	**0.85**	0.86	0.84

Table 8. Evaluation results on the CO3D[45] dataset with manually selected front view and annotated elevation angles.

to be robust to diverse camera setups which is still an ongoing research. Second, we curated four datasets which cov-

Method	Geometry consistency		Texture consistency			Image quality		Semantic consistency		
	CD ↓	depth ↓	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	IQ-vlm ↑	class ↑	color ↑	style ↑
Ours	3.04	17.58	**26.43**	**0.88**	**0.06**	22.10	**0.37**	**0.74**	**0.88**	0.84
SyncDreamer[38]	**2.87**	**15.35**	25.44	0.88	0.06	30.64	0.17	0.59	0.84	0.79
SV3D-tune	3.37	21.94	24.97	0.85	0.08	22.04	0.34	0.72	0.88	0.82
SV3D[54]	3.39	26.17	23.99	0.83	0.09	22.07	0.33	0.71	0.87	0.79
EscherNet[28]	5.35	29.55	20.31	0.72	0.15	25.78	0.32	0.66	0.82	0.78
Hi3D[72]	6.24	33.72	21.42	0.81	0.12	25.77	0.41	0.63	0.83	0.78
MVDFusion[17]	6.29	50.54	17.27	0.70	0.22	28.45	0.26	0.62	0.80	0.78
EpiDiff[21]	8.05	61.91	15.85	0.64	0.27	**20.21**	0.42	0.74	0.86	0.84
ViewFusion[74]	8.26	54.27	16.14	0.63	0.28	21.77	0.39	0.72	0.84	0.84
Free3D[80]	10.64	60.00	14.77	0.65	0.33	33.58	0.38	0.60	0.70	0.75
Vivid123[30]	10.67	58.06	15.81	0.69	0.30	35.84	**0.56**	0.56	0.68	0.75
V3D [9]	10.74	65.40	16.30	0.71	0.26	27.89	0.40	0.65	0.79	0.79
Zero123-xl[11]	12.04	67.08	13.45	0.55	0.38	20.51	0.41	0.74	0.85	**0.87**
Zero123[2]	12.11	66.76	13.61	0.56	0.38	20.83	0.42	0.74	0.85	0.86

Table 9. Evaluation results on the MVImgNet [75] dataset with manually selected front view and annotated elevation angles.

Figure 11. Degradation of image quality after passing through the autoencoder of SV3D [54]. Clearly the high frequency details are destroyed by the autoencoder.

ers mainly daily objects, and most of them are indoor. It would be interesting to also consider outdoor objects such as buildings, statues or complex compositional shapes such as human-human or human-object interactions. Last but not least, we evaluate the robustness w.r.t lighting, elevation and azimuth angles. Real life objects have much more attributes that can affect the performance, such as the material, shading condition, specific object categories. One can do more comprehensive analysis could be done using our proposed metrics to understand the progress of SoTA methods. We leave these for future works.