ConsistNet: Enforcing 3D Consistency for Multi-view Images Diffusion

Jiayu Yang[1,2], Ziang Cheng[1,2], Yunfei Duan[1], Pan Ji[1], Hongdong Li[2]
[1]Tencent, [2]Australian National University
{jiayu.yang, ziang.cheng, hongdong.li}@anu.edu.au,
kownseduan@global.tencent.com, panji@tencent.com

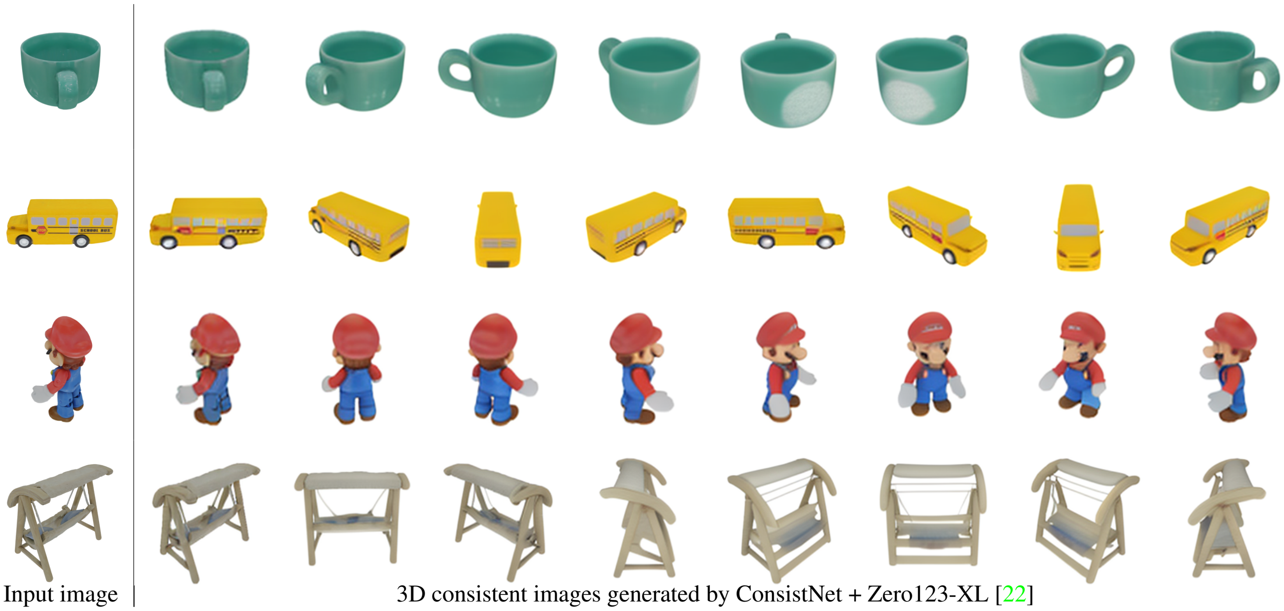
Input image
3D consistent images generated by ConsistNet + Zero123-XL [22]

Figure 1. We present ConsistNet, a plug-in module for image diffusion models like Zero123-XL [7] to generate multi-view consistent images. The ConsistNet is designed to be lightweight and efficient, allowing Zero123-XL [7] to generate 16 multi-view consistent images in **11 seconds**, which is **10x** faster than recent competing method [23].

Abstract

Given a single image of a 3D object, this paper proposes a novel method (named ConsistNet) that can generate multiple images of the same object, as if they are captured from different viewpoints, while the 3D (multi-view) consistencies among those multiple generated images are effectively exploited. Central to our method is a lightweight multi-view consistency block that enables information exchange across multiple single-view diffusion processes based on the underlying multi-view geometry principles. ConsistNet is an extension to the standard latent diffusion model and it consists of two submodules: (a) a view aggregation module that unprojects multi-view features into global 3D volumes and infers consistency, and (b) a ray aggregation module that samples and aggregates 3D consistent features back to each view to enforce consistency. Our approach departs from previous methods in multi-view image generation, in that it can be easily dropped in pre-trained LDMs without requiring explicit pixel correspondences or depth prediction. Experiments show that our method effectively learns 3D consistency over a frozen Zero123-XL backbone and can generate 16 surrounding views of the object within 11 seconds on a single A100 GPU. Our code will be made available on https://github.com/JiayuYANG/ConsistNet.

1. Introduction

Recent advances in the Latent Diffusion Models (LDM) [1, 28] for image generation have brought about remarkable success in generating high-quality images with compelling details. However, when applied to generate multiple-view images of the same object, vanilla LDMs are unable to ensure the 3D-consistencies among the generated multiple images. This is primarily due to the lack of mechanisms to enforce such 3D consistency information among the images taken from different viewpoints.

3D-consistent multi-view image diffusion models provide not only theoretical values, but hold major practical relevance, e.g., for 3D asset generation in VR/AR and video gaming applications. Such diffusion models can either serve as a multi-view consistent image prior for 3D generation via the Score Distillation Sampling (SDS) loss [26], or allow direct reconstruction of 3D assets from once sampled images.

A recent work, Zero123 [22] and Zero123-XL [7], stands out as a promising approach for this purpose. It leverages the power of CLIP image embedding combined with camera embedding to produce semantically coherent images that are also viewpoint-aware. However, such semantic multi-view consistent heuristics are rather weak, in the sense that the multiple-view images generated by Zero123-XL do not necessarily adhere to any shared 3D structure. In other words, the much-desired multi-view geometry consistency is not explicitly enforced in any effective manner.

To address this challenge, we introduce a novel latent diffusion model. Instead of using a single diffusion model, we run multiple diffusion models in parallel, each dedicated to a specific viewpoint. We propose a plug-in multi-view consistency block, namely, ConsistNet. This block ensures that the multiple images generated satisfy the underlying multi-view geometry principles (e.g., see Fig. 1).

In our method, the base diffusion models are pre-trained and remain frozen. The only trainable component is the ConsistNet block. This block can be plugged into every decoding layer of the denoising UNet. Its primary function is to gather multi-view feature maps and produce a residual feature map at every viewpoint that reflects 3D consistency. This residual map is then added back into the corresponding decoder layers to enforce 3D consistency.

Our method, albeit based on the Zero123-XL backbone, surpasses it in terms of 3D consistency. Through extensive experiments, we have achieved marked improvement in 3D consistency, and our model exhibits commendable generality when exposed to unseen data.

2. Related Work

Diffusion Model Denoising diffusion models have been applied to various tasks in computer vision, *e.g.*, image enhancement [6, 12, 45], style transfer and content editing [2, 15, 21, 30, 48], image [9, 31] video [1, 14, 16, 35] and 3D shape generation.

Classic diffusion models [17, 37] generate images by reversing a Markov process, where random noises are progressively added to clean images until the eventual distribution is Gaussian. Song *et al.* [36] proposed DDIM sampling that uses an alternative non-Markovian formulation that significantly reduces the number of denoising steps. A notable example of diffusion models is Latent diffusion models (LDM) [28], where a variational autoencoder is first trained to compress natural images to a compact latent space where the diffusion process later takes place.

There also exist methods for fine-tuning a pre-trained diffusion model, which allow the diffusion model to learn new concepts from a small dataset efficiently without the need to re-train the entire model (*e.g*. [11, 18, 30]). ControlNet [46] is yet another example that uses a zero-initialised ResNet block to piggyback a pre-trained diffusion model to enrich its expressive power. ControlNet has demonstrated promising results in controlling single image generation with various forms of inputs, including depth map, normal map, sketch image, and human pose. The role our new ConsistNet plays with respect to a pre-trained Zero123-XL is similar to what the ControlNet plays to a regular diffusion network.

Sparse and Single View Novel View Synthesis A closely related task to multi-view consistent image generation is the task of Novel View Synthesis (NVS). Traditional NVS methods require a large number of input images from diverse viewpoints, and novel views are estimated from interpolating or extrapolating those real input images (*e.g*. [4, 13, 20, 25, 32]. However, their performances are critically dependent on the viewpoint coverage and density of the input images.

In contrast, recent generative models are capable of hallucinating plausible novel images that are from the input views [3, 39, 49]. While these methods can often maintain a certain degree of appearance consistency across multiple views (*e.g*. the epipolar constraint), there is no guarantee that these multi-view 3D consistencies are geometrically correct. How to ensure the multi-view relationship is valid is precisely the main motivation of the present paper. Our method adopts the idea of a cross-view attention mechanism for leveraging multi-view consistency, an idea also used in previous work [5, 27, 41, 42]). We, however, use this idea in a novel way, namely, first un-project the latent image features to a consistent 3D feature volume, then re-project the consistency information back to the 2D image.

3D Consistent Image Generation Several methods have been proposed for multi-view consistent image generation using diffusion models. MVDream [33] fine-tunes a pre-trained image diffusion model on a multi-view dataset with a trainable attention module on the batch (multi-view) dimension. However, the attention module itself does not incorporate multi-view geometry, and the viewpoints are fixed. MVDiffusion [38] achieves multi-view consistency by attending multi-view features with camera projection. However, it requires knowledge of scene geometry.

Concurrent to our work is SyncDreamer [23], which uses a 3D-aware feature attention mechanism to synchronise features across views at every denoising step. It pre-processes latent images from multiple views into a 3D volume and uses the volume to guide U-Net to improve consistency. Differently, our ConsistNet block is designed to efficiently infer and improve 3D consistency operating within the U-Net itself, acting as a plug-in module. Specifically, SyncDreamer aggregates all multi-view latent images into a uniform spatial feature volume *outside* the denoising UNet, whereas, our view aggregation is performed on every layer *within* the UNet decoder. SyncDreamer proceeds to train each view's denoising step independently. By contrast, we allow all views to be mutually dependent and train the UNet to learn a joint multi-view distribution, which is an arguably more realistic assumption. Moreover, our method is not tethered to a fixed elevation angle. Compared to SyncDreamer, our model achieved 10x faster inference time under the same generation quality, is not limited to specific viewpoints, and is flexible on the number of views to generate.

3. Method

3.1. Latent Diffusion Model

The Latent Diffusion Model (LDM) is the backbone of our method. An LDM comprises two parts: a variational autoencoder (VAE) that compresses natural images into a computationally compact latent space $\mathcal{Z}$, and a denoising UNet that predicts the noise of a noisy latent representation.

During training, random noises $\epsilon \sim \mathcal{N}$ are progressively added to $\mathbf{x}_0 \in \mathcal{Z}$ in a Markov chain of $t = 1...N$ steps, and that $q(\mathbf{x}_t|\mathbf{x}_{t-1})$ is a Gaussian. The denoising UNet ϵ_θ is trained to approximate the reverse process $q(\mathbf{x}_{t-1}|\mathbf{x}_t)$ by minimising a lower bound loss term, in the form of

$$L_t = \sum_{t=2}^{T} \mathbb{E}_q D_{KL}\big(q(\mathbf{x}_{t-1}|\mathbf{x}_0, \mathbf{x}_t)\|p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t)\big). \quad (1)$$

The above minimisation problem can be implemented as predicting the noise at each time step, leading to the following training loss

$$L = \mathbb{E}_{\mathbf{x}_0,\epsilon,t}\|\epsilon - \epsilon_\theta(\mathbf{x}_t, t)\|_2^2. \quad (2)$$

Our goal is to extend the diffusion-based image generation process to the multi-view setting, where an independent random noise is progressively added to each multi-view image $\mathbf{x}_t^1, ...\mathbf{x}_t^N$ during the noising process. The resulting multi-view training loss is

$$L = \mathbb{E}_{\mathbf{x}_0^{1..N},\epsilon,i,t}\|\epsilon - \epsilon_\theta^{1..N}(\mathbf{x}_t^{1..N}, t)\|_2^2. \quad (3)$$

Our multi-view diffusion model is realised by calling multiple single view LDMs in parallel. To incorporate view consistency, we introduced a 3D aware plug-in module called ConsistNet block, that aggregates intermediate feature maps through cross-view projection.

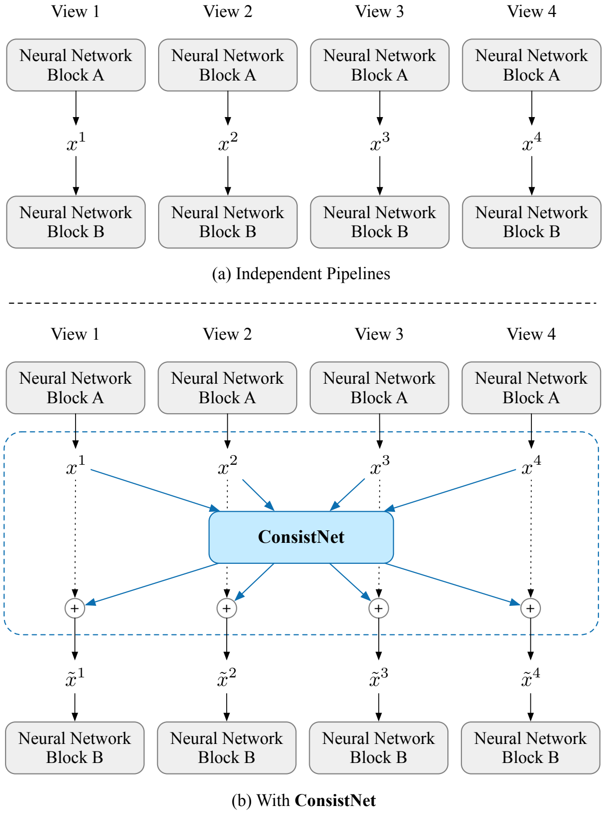

Figure 2. Enforcing 3D consistency among individual network pipelines using ConsistNet. Trainable modules are marked in blue. (a) Network pipelines running independently do not have information over each other. (b) ConsistNet inter-connect pipelines and enforce 3D consistency.

3.2. ConsistNet Block

At the core of our method is an add-on block to pre-trained LDMs, called ConsistNet, that exchanges information between parallel LDMs running at different viewpoints based on underlying multi-view geometry principles, see Fig. 2. The ConsistNet block is trained end-to-end using the same loss function as defined in (3).

Denote by $i = 1...N$ the viewpoint where every image is generated, the ConsistNet block $\mathcal{M}$ is a residual attention block that connects all N LDMs. It gathers a feature map x_t^i from every LDM, and adds back to it a 3D-consistent feature map via residual connection,

$$x_t^i \leftarrow x_t^i + \mathcal{M}(i, \{x_t^j | j = 1...N\}). \quad (4)$$

The ConsistNet block comprises two submodules: (i) a view aggregation module that un-projects feature maps x_t^i into world feature volumes then uses a view aggregation network to infer consistency, and (ii) a ray aggregation module that samples 3D consistent features back to each view and uses a ray aggregation network to enforce consistency.

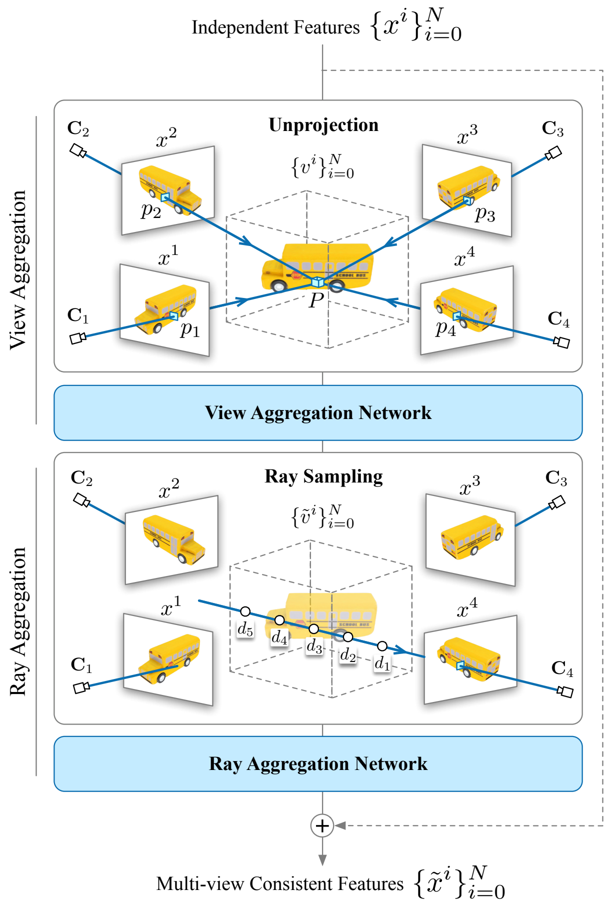

Figure 3. ConsistNet Block. Our ConsistNet block consists of two sub-modules: (a) a view aggregation module that un-projects image features to 3D and infers consistency by a view aggregation network, and (b) a ray aggregation module that samples 3D consistent features back to each view and uses a ray aggregation network to enforce consistency. Trainable modules are marked in blue.

View Aggregation. We start by unprojecting every feature map x^i onto a 3D volume in unified world coordinates, see Fig. 3. This yields a volume v^i for every viewpoint, defined as

$$v^i = \Pi_i^{-1}(x^i) \oplus \text{PosEncode}(v^i_{cam}), \tag{5}$$

where Π^{-1} is the inverse camera projection by bi-linear interpolation, $\oplus$ denotes feature concatenation, and v^i_{cam} is a fixed camera parameter volume that encodes the view direction and projection depth at each voxel. We use the sinusoidal position encoding for camera volume.

We use self-attention and 3D convolutions to implement the view aggregation network. All N volumes are voxel-aligned, and are attended to each other by applying a multi-headed self-attention layer in the N views dimension. More formally,

$$\bar{v}^i[P] = Attention(\{v^i[P]|i = 1...N\}), \tag{6}$$

where P denotes voxels. We further process the volume $\bar{v}^i$ with a few layers of 3D convolution.

Ray Aggregation. The volume $\bar{v}^i$ gathers shared information between parallel multi-view LDMs, which is then delivered back to each LDM via a ray aggregation module, see Fig. 3.

We first warp the world volumes $\bar{v}^i$ back to its corresponding camera frustum by uniformly sample depth along the viewing ray of each pixel between minimum and maximum depth, use tri-linear interpolation to fetch the feature, and append the warped volume with its camera depth encoding for depth-wise attention.

$$\tilde{v}^i = Attention\big(\text{Warp}_i(\bar{v}^i) \oplus \text{PosEncode}(d^i_{cam})\big). \tag{7}$$

The warped volume is then projected to a 2D map $\tilde{x}^i$ by aggregating it along the depth dimension. We implement this projection as the ray aggregation network, by a cross-attention layer using the feature map x^i as query,

$$\tilde{x}^i[r] = CrossAttention(x^i, \{\tilde{v}^i[d]|d \in [d_{\text{near}}, d_{\text{far}}]\}), \tag{8}$$

where r is the ray traced back from camera i, and d_{near} and d_{far} are the near and far depth of camera frustum.

To allow fast training of ConsistNet on a pre-trained LDM, we add $\tilde{x}^i$ back to the original feature map x^i via a residual layer whose weights are initialised to zero following ControlNet [46]. The residual layer is implemented as an MLP that processes the feature map in a per-pixel manner without convolution. Hence, the only trainable components are the two aggregation networks and the final MLP.

4. ConsistNet for Multi-view Diffusion

We use Zero123-XL [7] as backbone to showcase how ConsistNet can enforce 3D multi-view consistency in large pretrained LDM models. Zero123-XL is a specifically fine-tuned version of image variation LDM model, which follows the structure of Stable Diffusion [28], see Fig. 4. The denoising U-Net [29] contains an encoder and a decoder, both of which have 12 blocks, and a mid block in between. Unlike Stable-Diffusion where the denoising U-Net is conditioned on the CLIP language embedding of text input, Zero123-XL is conditioned on the CLIP image embedding of the input image as well as the relative camera rotation from input view. Moreover, the reference image is encoded to a reference latent image and is concatenated with the noisy latent image as input to U-Net on every denoising step.

We instantiate several parallel pretrained Zero123-XL models, with each model dedicated to a specific viewpoint. Without the ConsistNet block, these diffusion processes would operate independently of each other, lacking the 3D consistency between shared views. To introduce 3D consistency, we insert a ConsistNet block to each decoder layer of

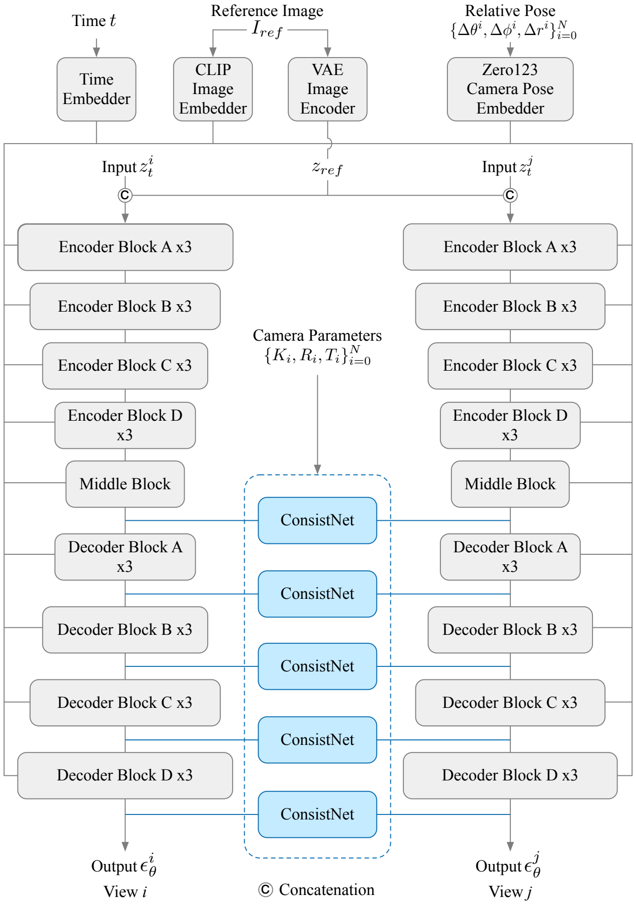

Figure 4. ConsistNet plugged into the U-Net of Zero123 [22]. Trainable modules are marked in blue. We plug ConsistNet block after every decoder block of Zero123-XL's U-Net to enforce 3D consistency.

the UNet, therefore allowing the feature maps to be mutually attended in a coarse to fine manner. An illustration is given in Fig 4.

5. Experiments

5.1. Datasets

Objaverse Dataset [8] is a large-scale dataset containing 800K+ annotated 3D mesh objects. We use this dataset for training and validation. We first filter out samples containing multiple objects by counting the number of bounding boxes in the scene. We render 16 256 × 256 resolution images per object from uniformly distributed viewpoints surrounding the object. For each object, we randomly choose a positive elevation angle up to 30 degrees for all views. We render under a mixture of global lighting and point lighting from camera centre to produce desired shading effects (*e.g.* normal direction and specular highlights) without introducing strong shadows.

Google Scanned Objects Dataset [10] is a open-source collection of over one thousand 3D-scanned household items. We use this entire dataset for evaluation only. We evaluate our model trained on Objaverse dataset on this dataset without any fine-tuning. We render objects from this dataset under the same setting as Objaverse dataset and manually set three options of elevation angles, 0, 15, and 30 degrees, to evaluate model performance on different elevation angles.

5.2. Implementation Details

Our ConsistNet blocks are plugged into a pre-trained and frozen Zero123-XL backbone. We train for 85k iterations with an AdamW optimiser [24] and a learning rate of 3×10^{-5}. The training takes 46 hours on 8 A100 40G GPUs. We directly test our trained model on entire Google Scanned Objects [10] dataset without any fine-tuning. We use DDIM sampler [36] with 50 denoising steps and generate 16 multi-view images for each object. We use a single A100 40GB GPU for all evaluations. We implement ConsistNet in HuggingFace Diffusers [40] framework.

5.3. Metrics

We use the following evaluation metrics to quantitatively evaluate the performance of our model compared to existing methods. Perceptual Loss (LPIPS) [47] measures the perceptual distance between two images by comparing the deep features extracted by deep neural networks given each image as input. Two pre-trained networks, AlexNet [19] and VGG [34], are employed to compute perceptual loss, denoted as LPIPS_Alex and LPIPS_VGG accordingly. Structural Similarity (SSIM) [43, 44] measures the structural similarity between two images, considering both colour and texture information. We report SSIM score [43] and multiscale SSIM score [44], denoted as SSIM and MS-SSIM accordingly. We also report Peak Signal-to-Noise Ratio (PSNR).

5.4. Compared methods

We compare our pipeline with several baseline methods. Zero123-XL serves as a baseline that is trained to produce multi-view images conditioned solely base on the CLIP embedding of input image and viewing angles. Additionally, we include the results from DreamFusion [26] using Zero123-XL as guidance. While DreamFusion [26] is multi-view consistent by construction, it requires training of a NeRF [25] through Score Distillation Sampling loss, resulting in long generation time. We also include a concurrent method, Syncdreamer [23], in our comparison. We use its official inference code and provided pre-trained model for evaluation.

5.5. Quantitative Results

For each object in the Google Scanned Objects dataset, we use one of its rendered images as reference image in-

Elevation	Model	LPIPS_Alex ↓	LPIPS_VGG ↓	SSIM ↑	MS-SSIM ↑	PSNR ↑	Runtime ↓
0	Zero123-XL [7]	0.19	0.14	0.85	0.72	18.25	5s
	DreamFusion [26]+Zero123-XL [7]	0.22	0.16	0.87	0.72	18.53	18min
	SyncDreamer [23]	0.24	0.17	0.85	0.66	17.41	2min
	Ours	**0.15**	**0.11**	**0.89**	**0.82**	**22.75**	11s
15	Zero123-XL [7]	0.23	0.17	0.83	0.65	16.59	5s
	DreamFusion [26]+Zero123-XL [7]	0.23	0.16	0.85	0.69	17.76	18min
	SyncDreamer [23]	0.17	0.14	0.86	0.76	18.93	2min
	Ours	**0.12**	**0.09**	**0.90**	**0.86**	**23.93**	11s
30	Zero123-XL [7]	0.27	0.18	0.83	0.61	16.10	5s
	DreamFusion [26]+Zero123-XL [7]	0.14	0.12	0.88	0.84	21.53	18min
	SyncDreamer [23]	**0.10**	**0.09**	**0.90**	**0.88**	**23.81**	2min
	Ours	0.11	**0.09**	**0.90**	0.86	23.67	11s

Table 1. **Google Scanned Objects Dataset.** Performance at different elevation angle. Our model performs comparably to SyncDreamer on elevation angle 30 with 10x faster speed, and can generalise well on 0 and 15 degree elevations.

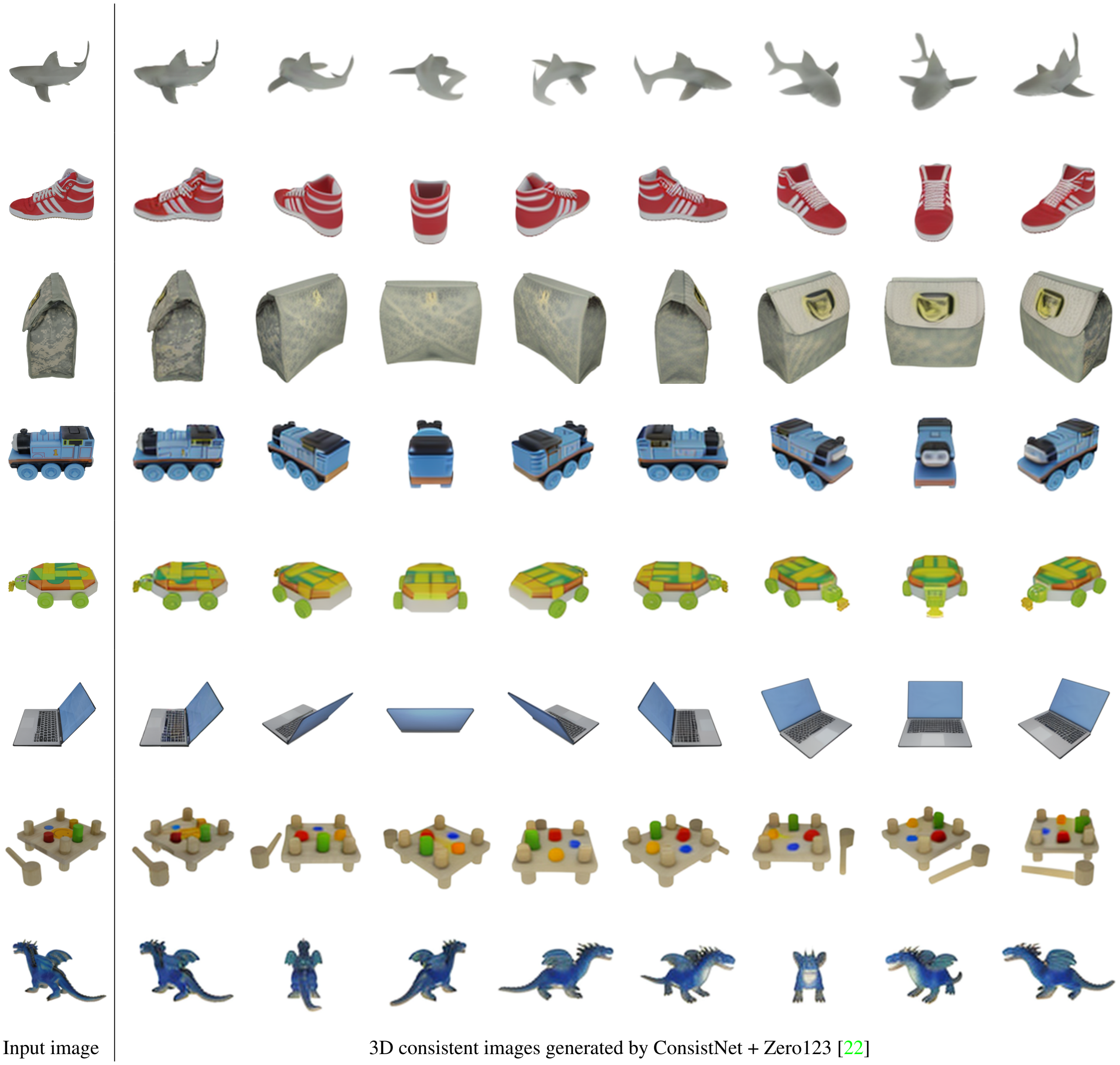

Figure 5. **Google Scan Objects dataset.** More qualitative results generated by our method.

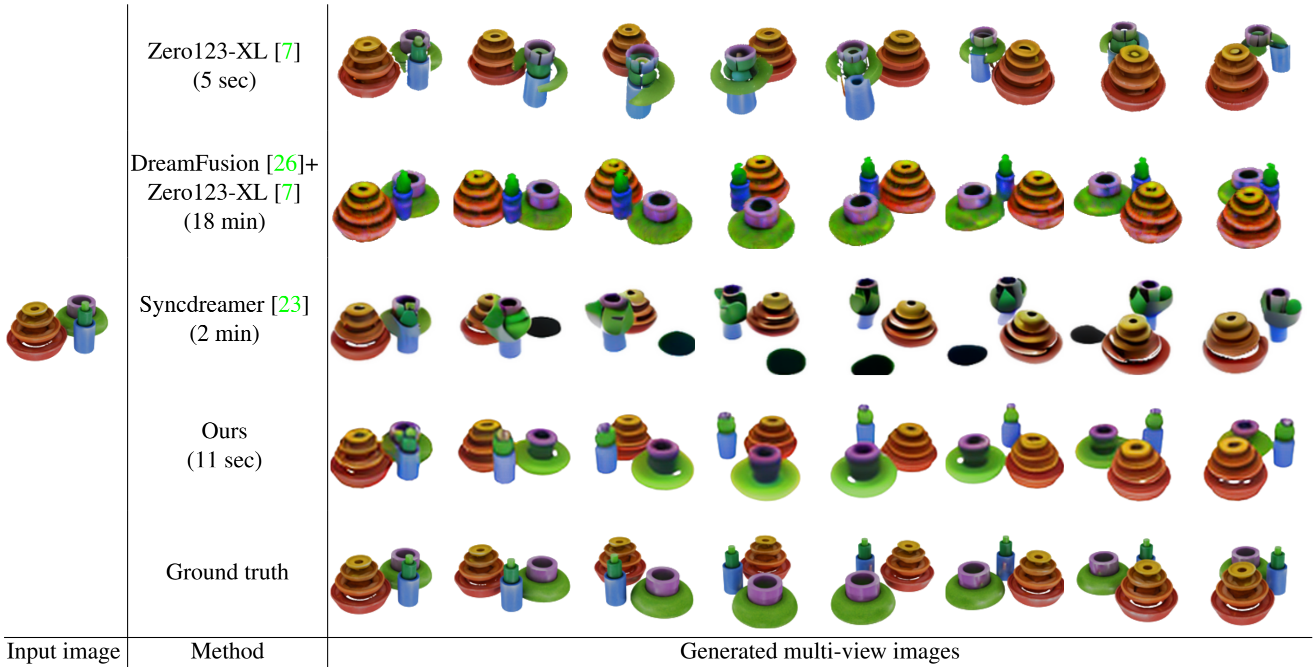

Figure 6. **Google Scan Objects dataset.** Comparison of baseline methods and our approach with complex geometry and colors. Our method is able to infer the correct geometry and produce consistent multi-view images.

put to generate 16 uniformly sampled surrounding views of the object. We evaluate under three choices of elevation angle, 0 degree, 15 degree and 30 degree, to better reflect the model performance. Results are shown in Tab. 1. On elevation 0 and 15, our model largely outperforms all existing works and concurrent work SyncDreamer [23] on all metrics. On elevation 30, our model performs comparably to SyncDreamer. Comparing with our base model Zero123-XL [22], plugging in our module improves its generation quality on all elevation angles.

5.6. Qualitative Results

We show qualitative results generated by our method in Fig. 1 and Fig. 5. Our model can generalise well to unseen data. Moreover, we select two objects from Google Scanned Objects with complex geometry and diverse colour to qualitatively compare with existing methods. Results are shown in Fig. 6. Our model improves 3D consistency of our base model Zero123-XL. We also show qualitative results from our model in Fig. 7 using internet images and images generated by existing text-to-image models. Our model can generalise well to various image sources.

6. Ablation Study

Plug-in location We now examine how the plug-in location of ConsistNet blocks affects image generation quality. We survey a few plug-in location options, including encoder/decoder of the U-Net, and before/after each U-Net block. Results are shown in Tab. 2. Plugging ConsistNet blocks on U-Net decoder only achieved the best performance, while plug-in on encoder only does not perform as well.

Plug-in location	LPIPS_Alex ↓	SSIM ↑	PSNR ↑	Runtime ↓
Encoder Only	0.14	0.87	23.14	11s
Enc. & Dec.	0.11	0.91	23.69	19s
Decoder Only	0.11	0.90	23.67	11s
Before Block	0.12	0.88	23.56	11s
After Block	0.11	0.90	23.67	11s

Table 2. **Google Scanned Objects Dataset.** Analysis of plug-in location w.r.t generation quality and runtime efficiency. We choose to plug-in ConsistNet after each U-Net decoder block to achieve the best trade-off between generation quality and efficiency.

Base Model	LPIPS_Alex ↓	SSIM ↑	PSNR ↑	Runtime ↓
Zero123 [22]	0.14	0.81	21.88	11s
Zero123-XL [7]	0.11	0.90	23.67	11s

Table 3. **Google Scanned Objects Dataset.** Choice of base model. ConsistNet perform better when plug-in into the better Zero123-XL base model.

Base Model We now examine how the base model affect image generation quality. We compare the original Zero123 baseline with the Zero123-XL when using as the base model of ConsistNet and results are listed in Tab. 3. ConsistNet perform better when plug-in into the better Zero123-XL base model.

Parameter sensitivity

We conduct experiments to analyze the effect of the number of views and inference steps on the final image generation quality. Results are listed in Tab. 4. Reducing generation views to 8 does not impact generation quality. Further reducing to 4 views result in insufficient overlap between views that affect generation quality. 50 denoising steps are sufficient to achieve the best performance.

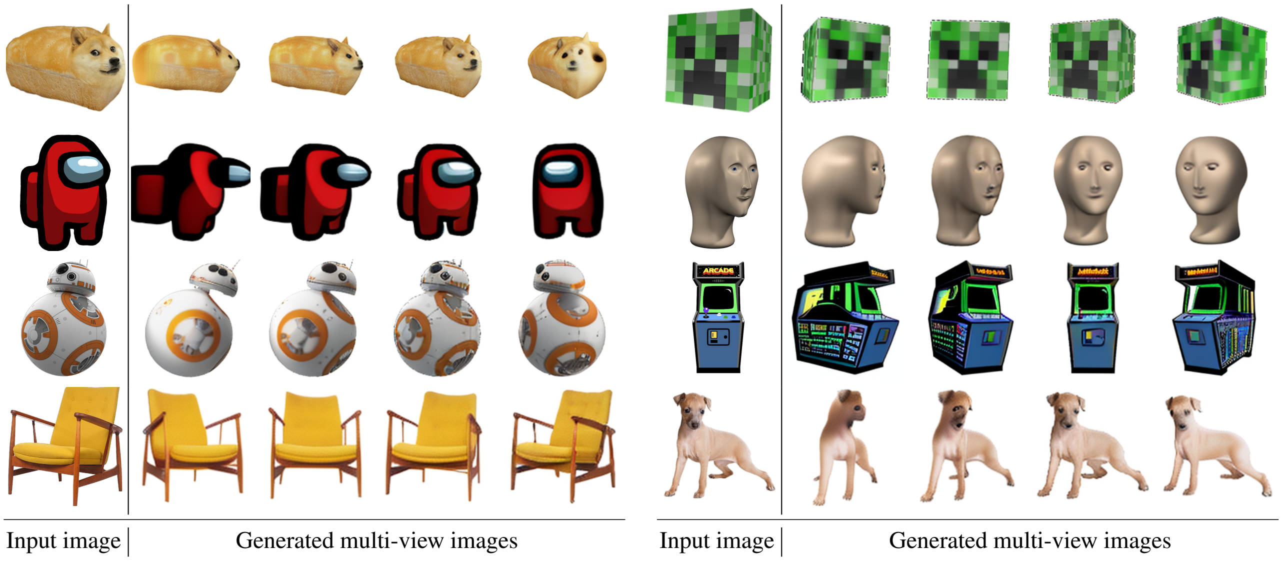

Figure 7. Using ConsistNet on Internet images and AI generated images. Our model can generalize well to various inputs.

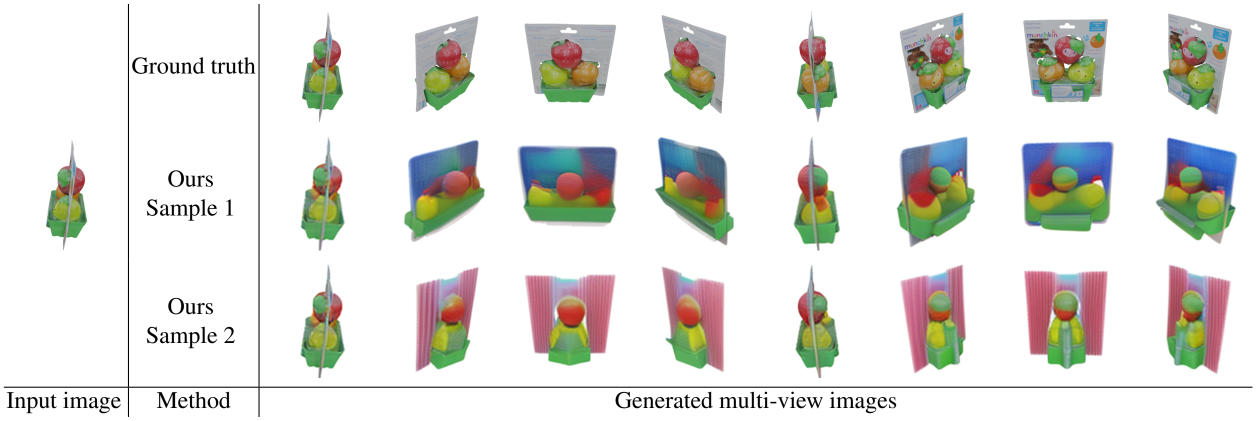

Figure 8. **Google Scan Objects dataset.** Comparison of ground-truth multi-view images with two generated variants, showing the inherent ambiguity and diverse possibilities when extrapolating from a single reference image.

Parameter		LPIPS_Alex ↓	SSIM ↑	PSNR ↑	Runtime ↓
Views	4	0.13	0.88	23.39	6s
	8	0.11	0.90	23.51	8s
	16	0.11	0.90	23.67	11s
Steps	50	0.11	0.90	23.67	11s
	100	0.11	0.91	23.68	21s
	200	0.11	0.91	23.68	40s

Table 4. **Google Scanned Objects Dataset.** Parameter sensitivity test. Views denotes the number of multi-view images generated. Steps denotes the denoising steps.

6.1. Discussion.

The task of generating unseen multi-view images of an object from a single reference image is severely ill-posed. In general, there are infinite numbers of possible solutions given only a single reference image. Fig. 8 illustrate such a solution ambiguity. Consequently, using so-called 'quantitative evaluation' by simply comparing the generated views with the ground-truth views as the performance metric is not well suited. Designing better metrics for this task is an important future task.

7. Conclusion

We have proposed ConsistNet, a multi-view consistency plug-in block for latent diffusion models to improve 3D consistency without requiring explicit pixel correspondences or depth prediction. Experiments show that our models effectively improve 3D consistency of a frozen Zero123-XL backbone and can generalise well to unseen data. In the future, we plan to further improve the computational efficiency of the model and develop a 3D reconstruction plug-in module to generate a 3D mesh along the multi-view image denoising process.

References

[1] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22563–22575, 2023. 1, 2

[2] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023. 2

[3] Eric R. Chan, Koki Nagano, Matthew A. Chan, Alexander W. Bergman, Jeong Joon Park, Axel Levy, Miika Aittala, Shalini De Mello, Tero Karras, and Gordon Wetzstein. GeNVS: Generative novel view synthesis with 3D-aware diffusion models. In *arXiv*, 2023. 2

[4] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14124–14133, 2021. 2

[5] Yuedong Chen, Haofei Xu, Qianyi Wu, Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. Explicit correspondence matching for generalizable neural radiance fields. *arXiv preprint arXiv:2304.12294*, 2023. 2

[6] Zheng Chen, Yulun Zhang, Ding Liu, Bin Xia, Jinjin Gu, Linghe Kong, and Xin Yuan. Hierarchical integration diffusion model for realistic image deblurring. *arXiv preprint arXiv:2305.12966*, 2023. 2

[7] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. *arXiv preprint arXiv:2307.05663*, 2023. 1, 2, 4, 6, 7

[8] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023. 5

[9] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 2

[10] Anthony G. Francis, Brandon Kinman, Krista Ann Reymann, Laura Downs, Nathan Koenig, Ryan M. Hickman, Thomas B. McHugh, and Vincent Olivier Vanhoucke, editors. *Google Scanned Objects: A High-Quality Dataset of 3D Scanned Household Items*, 2022. 5

[11] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022. 2

[12] Sicheng Gao, Xuhui Liu, Bohan Zeng, Sheng Xu, Yanjing Li, Xiaoyan Luo, Jianzhuang Liu, Xiantong Zhen, and Baochang Zhang. Implicit diffusion models for continuous super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10021–10030, 2023. 2

[13] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The lumigraph. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 453–464. 2023. 2

[14] Yuwei Guo, Ceyuan Yang, Anyi Rao, Yaohui Wang, Yu Qiao, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725*, 2023. 2

[15] Amir Hertz, Kfir Aberman, and Daniel Cohen-Or. Delta denoising score. 2023. 2

[16] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022. 2

[17] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2

[18] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 2

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 5

[20] Marc Levoy and Pat Hanrahan. Light field rendering. In *Seminal Graphics Papers: Pushing the Boundaries, Volume 2*, pages 441–452. 2023. 2

[21] Jun Hao Liew, Hanshu Yan, Jianfeng Zhang, Zhongcong Xu, and Jiashi Feng. Magicedit: High-fidelity and temporally coherent video editing. *arXiv preprint arXiv:2308.14749*, 2023. 2

[22] 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. *arXiv preprint arXiv:2303.11328*, 2023. 1, 2, 5, 6, 7

[23] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Learning to generate multiview-consistent images from a single-view image. *arXiv preprint arXiv:2309.03453*, 2023. 1, 2, 5, 6, 7

[24] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[25] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 2, 5

[26] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion, 2022. 2, 5, 6, 7

[27] Yufan Ren, Tong Zhang, Marc Pollefeys, Sabine Süsstrunk, and Fangjinhua Wang. Volrecon: Volume rendering of signed ray distance functions for generalizable multi-view

reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16685–16695, 2023. 2

[28] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1, 2, 4

[29] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. 4

[30] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. 2022. 2

[31] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems*, 35:36479–36494, 2022. 2

[32] Yujiao Shi, Hongdong Li, and Xin Yu. Self-supervised visibility learning for novel view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9675–9684, 2021. 2

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

[34] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5

[35] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. *arXiv preprint arXiv:2209.14792*, 2022. 2

[36] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020. 2, 5

[37] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 2

[38] Shitao Tang, Fuyang Zhang, Jiacheng Chen, Peng Wang, and Yasutaka Furukawa. Mvdiffusion: Enabling holistic multi-view image generation with correspondence-aware diffusion. *arXiv preprint arXiv:2307.01097*, 2023. 2

[39] Hung-Yu Tseng, Qinbo Li, Changil Kim, Suhib Alsisan, Jia-Bin Huang, and Johannes Kopf. Consistent view synthesis with pose-guided diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16773–16783, 2023. 2

[40] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, and Thomas Wolf. Diffusers: State-of-the-art diffusion models. https://github.com/huggingface/diffusers, 2022. 5

[41] Peihao Wang, Xuxi Chen, Tianlong Chen, Subhashini Venugopalan, Zhangyang Wang, et al. Is attention all nerf needs? *arXiv preprint arXiv:2207.13298*, 2022. 2

[42] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul P Srinivasan, Howard Zhou, Jonathan T Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2021. 2

[43] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 5

[44] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402. Ieee, 2003. 5

[45] Jay Whang, Mauricio Delbracio, Hossein Talebi, Chitwan Saharia, Alexandros G Dimakis, and Peyman Milanfar. Deblurring via stochastic refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16293–16303, 2022. 2

[46] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023. 2, 4

[47] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 5

[48] Yuxin Zhang, Nisha Huang, Fan Tang, Haibin Huang, Chongyang Ma, Weiming Dong, and Changsheng Xu. Inversion-based style transfer with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10146–10156, 2023. 2

[49] Zi-Xin Zou, Weihao Cheng, Yan-Pei Cao, Shi-Sheng Huang, Ying Shan, and Song-Hai Zhang. Sparse3d: Distilling multiview-consistent diffusion for object reconstruction from sparse views. *arXiv preprint arXiv:2308.14078*, 2023. 2