MVSplat360: Feed-Forward 360 Scene Synthesis from Sparse Views

Yuedong Chen[1] **Chuanxia Zheng**[2] **Haofei Xu**[3,4] **Bohan Zhuang**[1]
Andrea Vedaldi[2] **Tat-Jen Cham**[5] **Jianfei Cai**[1]

[1]Monash University [2]VGG, University of Oxford [3]ETH Zurich
[4]University of Tübingen, Tübingen AI Center [5]Nanyang Technological University

Abstract

We introduce MVSplat360, a feed-forward approach for 360° novel view synthesis (NVS) of diverse real-world scenes, using only sparse observations. This setting is inherently ill-posed due to minimal overlap among input views and insufficient visual information provided, making it challenging for conventional methods to achieve high-quality results. Our MVSplat360 addresses this by effectively combining geometry-aware 3D reconstruction with temporally consistent video generation. Specifically, it refactors a feed-forward 3D Gaussian Splatting (3DGS) model to render features directly into the latent space of a pre-trained Stable Video Diffusion (SVD) model, where these features then act as pose and visual cues to guide the denoising process and produce photorealistic 3D-consistent views. Our model is end-to-end trainable and supports rendering arbitrary views with as few as 5 sparse input views. To evaluate MVSplat360's performance, we introduce a new benchmark using the challenging DL3DV-10K dataset, where MVSplat360 achieves superior visual quality compared to state-of-the-art methods on wide-sweeping or even 360° NVS tasks. Experiments on the existing benchmark RealEstate10K also confirm the effectiveness of our model. Readers are highly recommended to view the video results at `donydchen.github.io/mvsplat360`.

1 Introduction

The rapid advancement in 3D reconstruction and NVS has been facilitated by the emergence of differentiable rendering [29, 31, 41, 40, 21]. These methods, while fundamental and impressive, are primarily tailored for per-scene optimization, requiring hundreds or even thousands of images to comprehensively capture every aspect of the scene. Consequently, the optimization process for each scene can be time-consuming, and collecting thousands of images is impractical for casual users.

In contrast, we consider the problem of novel view synthesis in diverse real-world scenes using a limited number of source views through a feed-forward network. In particular, this work investigates *the feasibility of rendering wide-sweeping or even 360° novel views using extremely sparse observations*, like fewer than 5 images. This task is inherently challenging due to the complexity of scenes, where the limited views do not contain sufficient information to recover the whole 3D scene. Consequently, there is a necessity to ensemble visible information under minimal overlap accurately and generate missing details reasonably.

This represents a new problem setting in sparse-view feed-forward NVS. Existing feed-forward methods typically focus on two distinct scenarios: 360° NVS with extremely sparse observations, but only at *object-level* [20, 52, 66, 59, 27, 72, 63, 56, 62, 50, 46, 47, 18], or generating reasonable results for *scene-level* synthesis, but only for nearby viewpoints [53, 7, 19, 43, 9, 13, 6, 60, 10, 70, 42, 61]. In contrast, we argue that the time is ripe to unify these previously distinct research directions.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

 https://doi.org/10.52202/079017-3399

Figure 1: **Examples of our MVSplat360**. Given sparse and wide-baseline observations of diverse in-the-wild scenes, MVSplat360 can directly render 360° novel views (inward or outward facing) or other natural camera trajectory views in a *feed-forward* manner, without any per-scene optimization.

Our goal should be to develop systems capable of synthesizing wide-sweeping or even 360° novel views of large, real-world scenes with complex geometry and significant occlusion. Specifically, this work explores synthesising 360° novel views from fewer than 5 input images. We show that in this challenging setting, existing feed-forward scene synthesis approaches [9, 13, 6, 60, 10, 57] struggle to succeed. This failure arises from two main factors: i) the limited overlap among input views causes many contents to appear in only a few views or even a single one, posing significant challenges for 3D reconstruction; ii) the extremely sparse observations lack sufficient information to capture the comprehensive details of the whole scene, resulting in regions unobserved from novel viewpoints.

In this paper, we propose a simple yet effective framework to address these limitations and introduce the first benchmark for feed-forward 360° scene synthesis from sparse input views. Our key idea is to leverage prior knowledge from a large-scale pre-trained latent diffusion model (LDM) [35] to "imagine" plausible unobserved and disoccluded regions in novel views, which are inherently highly ambiguous. Unlike existing 360° object-level NVS approaches [27, 63, 49, 56, 25, 72, 50], large-scale real-world scenes comprise multiple 3D assets with *complex arrangements, heavy occlusions*, and *varying rendering trajectories*, which makes it particularly challenging to condition solely on camera poses, as also verified by concurrent work ViewCrafter [68].

To develop a performant framework for scene-level synthesis, we opt to treat the LDM as a refinement module, while relying on a 3D reconstruction model to process the complex geometric information. Broadly, we build upon the feed-forward 3DGS [21] model, MVSplat [10], to obtain coarse novel views by matching and fusing multi-view information with the cross-view transformer and cost volume. Although these results are imperfect, exhibiting visual artifacts and missing regions (see Fig. 1), they represent the reasonable geometric structure of the scene, as they are rendered from 3D representation. Furthermore, we choose Stable Video Diffusion (SVD) [3] over other image-based LDM as the refinement module, since its strong temporal consistency capabilities align better with the view-consistent requirement of the NVS task, as also observed by concurrent work 3DGS-Enhancer [28]. Conditioning SVD with the 3DGS rendered outputs, our MVSplat360 produces visually appealing novel views that are multi-view consistent and geometrically accurate (see Fig. 1).

Importantly, the original MVSplat outputs only RGB images, which is not the optimal condition for the generator, and is difficult to optimize jointly with the SVD denoising module. To tackle this, we

propose a simple Gaussian feature rendering with multi-channels, supervised with an introduced latent space alignment loss. Despite a seemingly minor change, the additional feature condition for SVD leads to a significant impact: It bypasses the SVD's frozen image encoder, allowing the gradients from SVD to backpropagate to enhance the geometry backbone and lead to improved visual quality, especially on the new challenging DL3DV-10K dataset. While related work Reconfusion [58], CAT3D [15] and latentSplat [57] also combine the 3D representation with 2D generators, the former two focus more on per-scene optimisation, while the latter only shows 360° NVS at the object level.

We conduct a series of experiments, mainly on two datasets. First, we establish a new benchmark on DL3DV-10K dataset [23], creating a new training and testing split for feed-forward wide-sweeping and 360° NVS. In this challenging setting, our MVSplat360 achieves photorealistic 360° NVS from sparse observations and demonstrates significantly better visual quality, where the previous scene-level feed-forward methods [9, 6, 60, 10] fail to achieve plausible results. Second, we deploy MVSplat360 on the existing RealEstate10K [74] benchmark. Following latentSplat [57], we estimate both interpolation and extrapolation NVS, and report state-of-the-art performance.

Our main contributions can be summarized as follows. 1) We introduce a crucial and pressing problem for novel view synthesis, *i.e.*, how to do wide-sweeping or even 360° NVS from sparse and widely-displaced observations of diverse in-the-wild scenes (*not objects*) in a feed-forward manner (*no any per-scene optimization*). 2) We propose an effective solution that nicely integrates the latest feed-forward 3DGS and the pre-trained Stable Video Diffusion (SVD) model with meticulous integration designs, where the former is for reconstructing coarse geometry and the latter is for refining the noisy and incomplete coarse reconstruction. 3) Extensive results on the challenging DL3DV-10K and RealEstate10K datasets demonstrate the superior performance of our MVSplat360.

2 Related Work

Sparse view per-scene reconstruction and synthesis. Differentiable rendering methods, such as NeRF [31] and 3DGS [21], are mainly designed for very dense views (*e.g.*, 100) as inputs for per-scene optimization, which is impractical to collect for casual users in real applications. To bypass the requirement for dense views, various regularization terms have been proposed in per-scene optimization [33, 11, 69, 48]. Recently, ZeroNVS [38], Reconfusion [58] and concurrent submissions, including CAT3D [15], ReconX [24], ViewCrafter [68], LM-Gaussian [67], 3DGS-Enhancer [28], have leveraged large-scale diffusion models for generating pseudo dense views of a 3D scene, which are then input into a per-scene reconstruction pipeline. However, these methods are inherently slow for reconstructing unseen scenes due to the necessity of per-scene optimisation.

Feed-forward scene reconstruction and synthesis. To mitigate these limitations, early approaches like Light Field Networks [40] use ray querying to predict novel views. Subsequent methods [44, 43] employ epipolar attention for multi-view geometry estimation. Later, pixelNeRF [66] devised pixel-aligned features for NeRF reconstruction [31], leading to a range of subsequent methods that incorporate feature matching fusion [7, 9], Transformers [36, 13, 32] and 3D volume representation [7, 60]. Recently, 3D Gaussian Splatting [21] has been implemented into feed-forward networks, such as pixelSplat [6], MVSplat [10], Splatter Image [46], Flash3D [45] and latentSplat [57]. While these methods were successful in novel view synthesis from sparse views, they fail to achieve this in a wide-sweeping or 360° setting. Concurrent yet unpublished submissions, DepthSplat [61] and Long-LRM [75], also show promising results in the 360° setting, but their frameworks have limited generation capabilities, necessitating the use of denser inputs, *e.g.*, 12 or 32 views.

Camera trajectory controllable synthesis. Generative models have achieved remarkable results for image/video synthesis [14, 73, 8, 35, 5, 17, 39, 3], but they lack precise control over the viewpoint of generated images. To address this, several approaches fine-tune large-scale pre-trained diffusion models with explicit image and pose conditions [27, 26, 25, 4, 72, 64, 56, 50]. However, these methods mainly show 360° NVS results on single objects, leaving the complex scene synthesis problem unsolved. Natural scenes comprise multiple objects with intricate occlusion relationships, presenting greater challenges that are not easily addressed by these single-object NVS models. Besides, camera trajectories can be highly irregular and varied when roving around such complex scenes. Although related works [55, 22, 38, 15] have explored training or fine-tuning diffusion models with camera control for scene synthesis, they often struggle with precise camera pose control [55, 22], and still rely on per-scene optimization for 3D reconstruction [38, 15].

 https://doi.org/10.52202/079017-3399

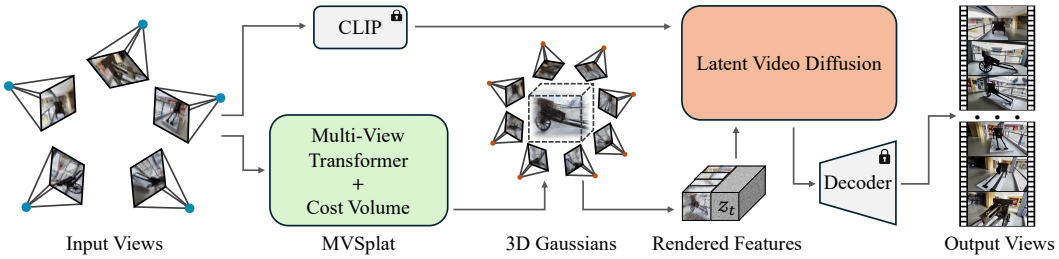

Figure 2: **Overview of our MVSplat360**. (a) Given sparse posed images as input, we first match and fuse the multi-view information using a multi-view Transformer and cost volume-based encoder. (b) Next, a 3DGS representation is constructed to represent the coarse geometry of the entire scene. (c) Considering such coarse reconstruction is imperfect, we further adapt a pre-trained SVD, using features rendered from the 3DGS representation as conditions to achieve 360° novel view synthesis.

3 Methodology

Given N sparse views $\mathcal{I} = \{\boldsymbol{I}^i\}_{i=1}^N$ and the corresponding camera poses $\mathcal{P} = \{\boldsymbol{P}^i\}_{i=1}^N$, with $\boldsymbol{P}^i = (\mathbf{K}^i, \mathbf{R}^i, \mathbf{T}^i)$ comprising intrinsic $\mathbf{K}^i$, rotation $\mathbf{R}^i$ and translation $\mathbf{T}^i$, our goal is to learn a model Φ that synthesizes wide-sweeping or even 360° novel view synthesis (NVS).

We opt to go beyond per-scene optimisation [1, 2, 58, 15], and to deal with a more general feed-forward network capable of achieving 360° NVS for unseen scenes, *yet without the need of additional per-scene training*. This requires effectively matching information between sparse views in 3D space, as well as generating sufficient content based on only partial observations. To achieve that, our MVSplat360 framework, illustrated in Fig. 2, comprises two main components: a multi-view geometry reconstruction module (Section 3.1) and a multi-frame consistent appearance refinement network (Section 3.2). The former is responsible for matching and fusing multi-view information from sparse observations to create a coarse geometry reconstruction, whereas the latter is designed to refine the appearance with a pre-trained latent video diffusion model. While similar two-step approaches have been explored in recent related works, *e.g.*, [4, 58, 38, 15], we are the first (to the best of our knowledge) to explore it on wide-sweeping or even 360° NVS for large-scale scenes from sparse views (as few as 5), *in a feed-forward manner*.

3.1 Multi-View Coarse Geometry Reconstruction

The first module is built upon a feed-forward 3DGS reconstruction model, *i.e.*, MVSplat [10] as in our implementation. Specifically, given sparse-view observations $\mathcal{I} = \{\boldsymbol{I}^i\}_{i=1}^N$ and their corresponding camera poses $\mathcal{P} = \{\boldsymbol{P}^i\}_{i=1}^N$, the model learns to predict 3D Gaussian parameters $\{(\boldsymbol{\mu}_i, \alpha_i, \boldsymbol{\Sigma}_i, \boldsymbol{c}_i)\}_{i=1}^{H\times W\times N}$, which can then be splatted to obtain a set of RGB images $\tilde{\mathcal{I}}^{\text{tgt}}$ using the target camera poses $\mathcal{P}^{\text{tgt}}$. To ensure better integration with the following diffusion module, we predict an additional Gaussian feature $\hat{\boldsymbol{f}}_i$, in parallel with other parameters, which can be rasterized to the corresponding latent features $\hat{\mathcal{F}}^{\text{tgt}}$. Furthermore, we also improve the view selection strategy to improve the model's robustness in handling widely displaced inputs.

Coarse geometry reconstruction. Our backbone comprises multi-view feature extraction, cost volume construction, depth estimation, and 3D Gaussian parameter predictions. First, a cross-view transformer encoder is applied to fuse multi-view information and obtain cross-view aware features $\mathcal{F} = \{\boldsymbol{F}^i\}_{i=1}^N$. Then, N cost volumes $\mathcal{C} = \{\boldsymbol{C}^i\}_{i=1}^N$ are constructed by matching feature correlations between cross-views. Specifically, it uniformly divides the depth into L layers in the near and far depth ranges, *i.e.* $\mathcal{D} = \{D_m\}_{m=1}^L$, and then warps the features from one view j to another view i via $\boldsymbol{F}_{D_m}^{j\to i} = \mathcal{W}(\boldsymbol{F}^j, \boldsymbol{P}^i, \boldsymbol{P}^j, D_m)$. The cost volume $\boldsymbol{C}^i = [\boldsymbol{C}_{D_1}^i, \boldsymbol{C}_{D_2}^i, \ldots, \boldsymbol{C}_{D_L}^i]$ is then collected by L correlations, where each correlation is expressed as $\boldsymbol{C}_{D_m}^i = \frac{\boldsymbol{F}_{D_m}^{j\to i}\cdot\boldsymbol{F}^i}{\sqrt{C}}$, with C denoting channel dimension. Finally, the per-view estimated depth d is obtained by applying the `softmax` operation on the cost volumes in the depth dimension. After that, the Gaussian mean is computed by $\boldsymbol{\mu} = \mathbf{K}^{-1}\boldsymbol{u}d + \Delta$, where $\mathbf{K}$ is the camera intrinsic, $\boldsymbol{u} = (u_x, u_y, 1)$ denotes each pixel, and $\Delta \in \mathbb{R}^3$ is the predicted offset, along with opacity $a \in [0, 1]$, covariance $\Sigma \in \mathbb{R}^{3\times 3}$, and color $\boldsymbol{c} \in \mathbb{R}^{3(S+1)^2}$,

where S is the order of the spherical harmonics [10]. Once the model predicts a set of 3D Gaussian parameters $\{(\boldsymbol{\mu}_i, \alpha_i, \boldsymbol{\Sigma}_i, \boldsymbol{c}_i)\}_{i=1}^{H\times W\times N}$, the target view $\tilde{\mathcal{I}}^{\text{tgt}}$ can be rendered through rasterization.

Gaussian feature rendering. Given sparse-view observations, MVSplat [10] tends to render images with noticeable artifacts in wide-sweeping novel viewpoints (see Fig. 1), resulting in suboptimal conditioning for the subsequent SVD. Since the rendered images must first be encoded into the latent space using a frozen encoder (see Section 3.2), enhancing the backbone with gradients from SVD would be computationally expensive. To address this issue, we propose directly rasterising features $\hat{\mathcal{F}}$ into *the latent space* of SVD, by predicting an additional parameter $\hat{\boldsymbol{f}}_i$ for each 3D Gaussian. This operation offers two advantages: (i) The latent feature includes multi-channel information, providing a more comprehensive representation of the scene; (ii) The entire framework is end-to-end connected by conditioning SVD on the rendered latent features instead of the image-encoded ones. It enables the SVD loss to optimize the Gaussian features, further enhancing the reconstruction backbone.

Observed and novel viewpoints selection. To enable 360° scene synthesis, it is crucial to choose the correct camera viewpoints, so that they can cover most contents in diverse and complex scenes [15]. It is impractical to assume a circular orbital camera trajectory like those object-level 360° view synthesis [27, 46, 72, 47], whereas it is suboptimal to randomly choose a video sequence like existing scene-level nearby viewpoint synthesis [46, 6, 10]. To this end, we propose to choose views *evenly distributed* within a set of targeted viewpoints as input. Specifically, for a given set of candidate views, we apply farthest point sampling over the camera locations to identify the input views and randomly choose from the rest as target views. The number of candidate views gradually increases throughout the training, stably improving the model's capability toward handling 360° scene synthesis.

View interaction within the local group. Recalling that the 3D reconstruction backbone MVSplat is primarily designed for nearby viewpoints, with key components like multi-view transformers and cost volume assuming sufficient overlap among input views. However, in the more challenging 360° settings, the widely displaced input views lead to minimal overlap between specific view pairs, hindering the effectiveness of the backbone. To mitigate this limitation, we refactor our backbone to use cross-view attention and construct the cost volume *only within a local group* of input views based on camera locations, reducing memory consumption and ensuring stable model convergence.

3.2 Multi-Frame Appearance Refinement

Video diffusion model. MVSplat360 utilizes an off-the-shelf multi-frame diffusion model, *i.e.* Stable Video Diffusion (SVD) [3], to refine the visual appearance of the aforementioned coarse reconstruction. SVD is pre-trained on large-scale video datasets and has strong prior knowledge of temporal consistency. It adheres to the original formulation used by Stable Diffusion (SD) [35] that conducts the denoising processing in the latent space. In particular, given a target sequence of $x^{1:M}$ with M images, they are initially embedded into the latent space by a frozen encoder $\mathcal{E}$, yielding $z_0^{1:M} = \mathcal{E}(x^{1:M})$, and then perturbed by adding Gaussian noise $\epsilon \sim \mathcal{N}(0, \boldsymbol{I})$ in a Markov process:

$$z_t^{1:M} = \sqrt{\bar{\alpha}_t} z_0^{1:M} + \sqrt{1-\bar{\alpha}_t}\epsilon_t^{1:M}, \tag{1}$$

while $\bar{\alpha}_0, \ldots, \bar{\alpha}_T$ is a pre-defined noise schedule within T steps. Given noise input $z_t^{1:M}$, the denoiser ϵ_θ is then trained by optimizing the following objective function:

$$\min_{\epsilon_\theta} \mathbb{E}_{(x^{1:M},y),t,\epsilon^{1:M}\sim\mathcal{N}(0,1)} \left[\|\epsilon^{1:M} - \epsilon_\theta(z_t^{1:M}, t, y)\|_2^2\right], \tag{2}$$

where $x^{1:M}$ is the target images, and y is the conditional inputs. After ϵ_θ is trained, the model can generate a video by performing iterative denoising from pure Gaussians $z_T^{1:M}$ conditioned on y.

Note that SVD is trained with the latest $\mathbf{v}$-prediction formulation [37], instead of the original ϵ-prediction in SD. Hence, the final loss is calculated in latent space using the mean squared error (MSE) between the ground truth and its prediction $||z_0^{1:M} - \hat{z}_t^{1:M}||_2^2$, where $\hat{z}_t^{1:M}$ is obtained by translating the velocity $\mathbf{v} = \Phi(z_t^{1:M}, t, y)$ to latent space, *i.e.*, $\hat{z}_t^{1:M} = \alpha_t z_t^{1:M} - \sigma_t \mathbf{v}$.

Multi-view hybrid conditions. To ensure an accurate understanding of the scene, the model requires the integration of both low-level perception (*e.g.*, depth and texture) and high-level understanding (*e.g.*, semantics and geometry). Following [27, 72, 50], we adopt a hybrid conditioning mechanism to fine-tune the SVD model for wide-sweeping NVS with sparse observations.

In one stream, a CLIP [34] image embedding token of the original visible views $\mathcal{I}$ is used as a global type and text prompt. At each UNet block, a cross-attention operation is applied to capture high-level

 https://doi.org/10.52202/079017-3399

semantics of the input images to the model. Since we have sparse views, we average these tokens to become one global token. In the other stream, the spatial conditions from the coarse geometry rendered features $\hat{\mathcal{F}}^{\text{tgt}} = \{\hat{\boldsymbol{F}}_i\}_{i=1}^{M}$ is channel-concatenated with the noised latent $z_t^{1:M}$. These spatially conditional features assist the model to capture the view information, and learn low-level perception to maintain the texture of the scenes. Compared to the concurrent work CAT3D [15], this coarse feature conditioning not only provides accurate pose information from the 3DGS rendering, but also offers reasonable visual information.

Color adjustment. While our MVSplat360 can achieve photorealistic NVS, the synthesized videos sometimes exhibit oversaturated colors (detailed in Appendix C). This may be visually acceptable for video generation, but it can decrease performance when evaluated on NVS task. To mitigate this, we apply post-processing by matching the color histogram between the SVD refined views $\hat{\mathcal{I}}^{\text{tgt}}$ and our 3DGS rendered views $\tilde{\mathcal{I}}^{\text{tgt}}$ before performing pixel-aligned measurements, *i.e.*, PSNR and SSIM.

3.3 Training Objectives

Our MVSplat360 predicts two sets of images, including the coarse one $\tilde{\mathcal{I}}^{\text{tgt}}$ from the 3DGS module and the refined one $\hat{\mathcal{I}}^{\text{tgt}}$ from the SVD module, where the former is mainly rendered to help supervise the geometry backbone. The entire model is end-to-end trainable, using three groups of loss functions, namely reconstruction loss, diffusion loss and latent space alignment loss.

In particular, the reconstruction loss is a linear combination of ℓ_2 and LPIPS [71], applied between the coarse outputs $\tilde{\mathcal{I}}^{\text{tgt}}$ and the corresponding ground truth $\mathcal{I}^{\text{tgt}}$. The other two loss functions are applied to the following SVD module, whose gradients will backpropagate to the 3DGS module but will not update those structural parameters, *i.e.*, $\boldsymbol{\mu}, \alpha, \boldsymbol{\Sigma}$. This is achieved by stopping the gradients from the structural parameters when rendering the latent features $\hat{\mathcal{F}}^{\text{tgt}}$, since keeping those gradient flows will lead to unstable training, as also observed by latentSplat [57]. We use the standard v-prediction formulation (detailed in Section 3.2) as the diffusion loss to fine-tune the denoising network of the SVD, keeping the first-stage encoder and decoder frozen. Since the SVD's released model is conditioned on image-encoded features while our diffusion module is on 3DGS-rendered ones, we find it beneficial to align these two spaces by regularizing with a latent space alignment loss, $\min_{g_\theta} \mathbb{E}_{\hat{\mathcal{F}} \sim g(\mathcal{I})} \|\mathcal{E}(\mathcal{I}^{\text{tgt}}) - \hat{\mathcal{F}}^{\text{tgt}}\|_2^2$, where g refers to the geometry backbone with trainable parameters θ and $\mathcal{E}$ is the frozen SVD encoder.

4 Experiments

4.1 Experimental Details

Datasets. To verify the effectiveness of MVSplat360 in synthesizing wide-sweeping and 360° novel views, we have established a challenging benchmark derived from DL3DV-10K [23]. It comprises 51.3 million frames from 10,510 real-world scenes, adhering to 65 point-of-interest (POI) [65] categories. For training, we use a subset in subfolders "3K" and "4K", resulting in ~2,000 scenes. We tested on the 140 benchmark scenes and filtered them out from the training set to ensure correctness. For each scene, we selected 5 input views using farthest point sampling based on camera locations and evaluated 56 views by equally sampling from the remaining, yielding a total of 7,840 test views. Additionally, since most DL3DV-10K scenes contain a two-round trajectory, we also report another setting by focusing only on half of the sequence, intending to cover the camera trajectory of one round. We denote the two settings as $n = 300$ and $n = 150$, where n refers to the frame distance span across all test views, as most scenes contain roughly 300 frames. We also assess our model on RealEstate10K [74], which contains real estate videos downloaded from YouTube. Consistent with existing works [6, 10], we train MVSplat360 on 67,477 scenes and test it on 7,289 scenes.

Metrics. To measure models from different perspectives, we follow [57] to report both the pixel-align metrics, *i.e.*, PSNR and SSIM [54], and the perceptual metrics, *i.e.*, LPIPS [71] and DISTS [12]. Since MVSplat360 aims to generate plausible contents for unobserved and disoccluded regions, we also reported the distribution metric, *i.e.*, Fréchet Inception Distance (FID) [16][1], which measures the similarity between distributions of the generated images and the real ones.

[1] `https://github.com/mseitzer/pytorch-fid`

Table 1: **Comparison with SoTA methods on DL3DV-10K**. Below, n is the frame distance span across all the tested novel views within each scene, which is set to 300 by default as most DL3DV-10K scenes contain roughly 300 extracted frames. Since most DL3DV-10K scenes contain a two-round trajectory, we also report another setting of $n = 150$ aiming for coverage of one round.

Method	$n = 300$					$n = 150$				
	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	FID↓	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	FID↓
pixelSplat [6]	14.83	0.401	0.576	0.383	142.83	16.05	0.453	0.521	0.348	134.70
MVSplat [10]	15.72	0.433	0.501	0.291	78.95	17.05	0.499	0.435	0.247	61.92
latentSplat [57]	16.68	0.469	0.439	0.234	37.68	17.79	0.527	0.391	0.206	34.55
MVSplat360	**16.81**	**0.514**	**0.418**	**0.175**	**17.01**	**17.81**	**0.562**	**0.352**	**0.151**	**18.89**

Implementation details. MVSplat360 is implemented with PyTorch and a CUDA-implemented 3DGS renderer. For coarse geometry reconstruction (Section 3.1), we set hyperparameters following MVSplat [10], except that we apply cross-view attention and build each cost volume within the nearest 2 views rather than all other views. For multi-frame appearance refinement (Section 3.2), we fine-tune from the 14-frame SVD [3] pre-trained model, but using rendered Gaussian features as conditions. We also remove the original "motion value" and "fps" conditions, since they are unrelated to our NVS task. We rescale the rendered feature to have a similar shape as the original image-encoded latent feature in the pre-trained model, which is critical for getting better details as it affects the decoder (more discussions are in Appendix A). We train SVD using 14 frames sampled along natural camera trajectories, captured by the initial videos. At inference, we directly feed 56 views to SVD but change all related temporal attention blocks to local attention with a window size of 14 to better align with the training. More implementation details can be found in Appendix B, and the codes are publicly available at `https://github.com/donydchen/mvsplat360`.

4.2 Results on the New DL3DV-10K Benchmark

We first assess the ability of MVSplat360 and baselines to synthesize wide-sweeping and 360° NVS in the newly constructed challenging benchmark with diverse scene categories.

Baselines. We perform a thorough comparison of MVSplat360 to the latest state-of-the-art (SoTA) 3DGS-based models, including pixelSplat [6], MVSplat [10] and latentSplat [57]. All models are trained on the same training split and evaluated on the publicly available 140 scenes.

Quantitative results. All models are trained to 100K steps and reported at Table 1, except for the latentSplat, which suffers from unstable training due to its GAN-based architecture. Hence, we report its best performance at around 60K training steps before the subsequent collapse. Our MVSplat360 outperforms all existing SoTA models in all metrics on the two settings $n = 300$ and $n = 150$. All models generally perform better on the $n = 150$ setting than on the $n = 300$ one since the latter spans larger viewpoints. It can be seen that the two generative models (latentSpalt and MVSplat360) generally perform better than the other two regression models, suggesting the importance of additional refinement in addressing feed-forward scene reconstruction.

Although our improvement on pixel-aligned metrics appears minor, this is expected since refinement via either interpolation (for disoccluded regions) or extrapolation (for unobserved regions) does not guarantee matching the ground truth at the pixel level. It mainly aims to provide a reasonable solution to refine the images and ensure they align with real-world image distribution. This is verified by the fact that our improvements on perceptual metrics are larger, and it is even more apparent on FID, which measures the distribution deviation. The superiority of our MVSplat360 stands out more from the qualitative results presented below.

Qualitative results. The qualitative comparisons are visualized in Fig. 3. MVSplat360 achieves remarkable visual results even under challenging conditions. pixelSplat [6] and MVSplat [10] exhibit obvious artifacts due to the issue of floating Gaussians. latentSplat [57] improves the results with an additional decoder and adversarial training. However, its resulting object geometry and image quality are still far from satisfactory, suggesting that the GAN-based framework cannot provide enough prior knowledge for refining 360° NVS in diverse real-world scenes. Readers are referred to our project page for video results with more comprehensive comparisons, where our MVSplat360 shows

 https://doi.org/10.52202/079017-3399

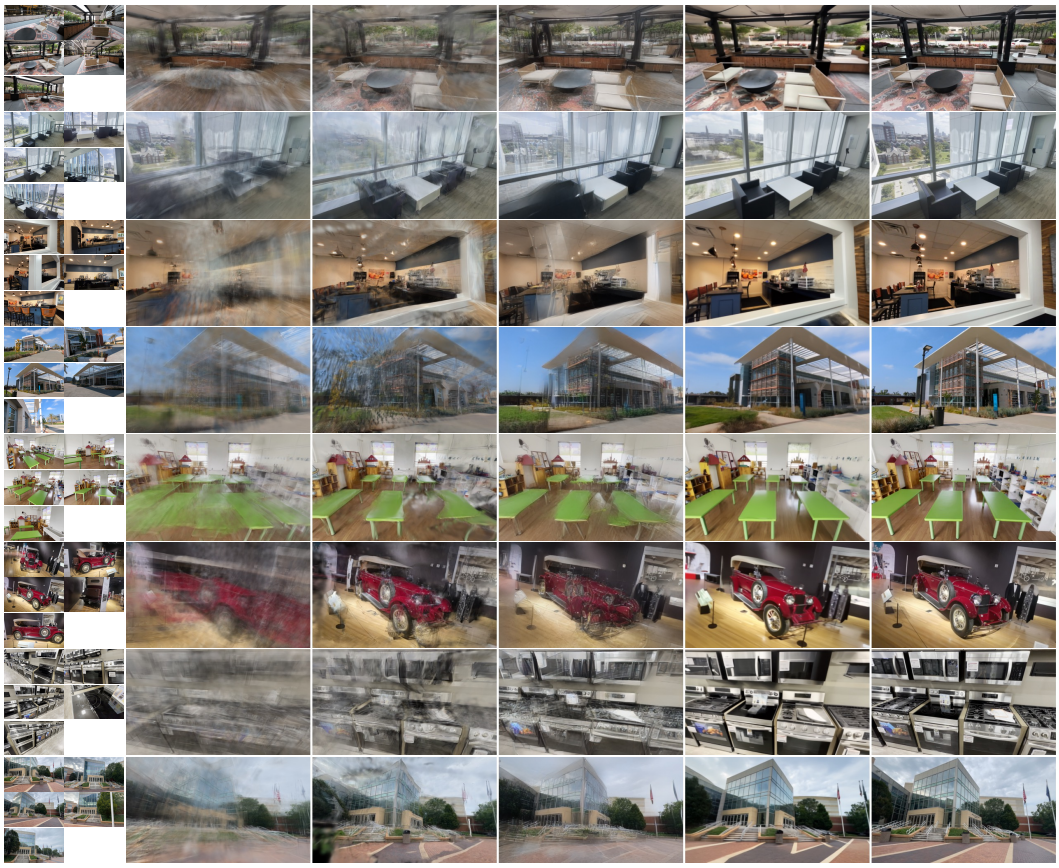

Figure 3: **Qualitative comparisons on DL3DV-10K**. MVSplat360 shows significant improvement compared to existing SoTA models. Here, we showcase with a rich mix of diversity and complexity, including indoor (bounded) *vs.* outdoor (unbounded), high *vs.* low texture frequency, more *vs.* less reflection, and more *vs.* less transparency. More results are provided in Appendix E.

Table 2: **Comparison with SoTA methods on RealEstate10K**. We report interpolation scores using the settings of [6, 10], where we retrain latentSplat [57] to maintain fair comparison (indicates with *). We report extrapolation scores by following [57].

Method	Interpolation			Extrapolation				
	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	DISTS↓	FID↓
PixelNeRF [66]	20.43	0.589	0.550	20.05	0.575	0.567	0.371	160.77
Du *et al.* [13]	24.78	0.820	0.213	21.83	0.790	0.242	0.144	11.34
pixelSplat [6]	25.89	0.858	0.142	21.84	0.777	0.216	0.130	5.78
latentSplat* [57]	25.53	0.851	0.139	22.62	0.777	0.196	0.109	2.79
MVSplat [10]	26.39	**0.869**	0.128	23.04	**0.812**	0.185	0.110	3.83
MVSplat360	**26.41**	**0.869**	**0.126**	**23.16**	0.810	**0.176**	**0.104**	**1.79**

multi-view consistent, high-quality visual results along complex camera trajectories, while others suffer from apparent artifacts.

4.3 Results on the Existing RealEstate10K Benchmark

We also assess MVSplat360 on the existing benchmark, following the existing "Interpolation" setting [6, 10] and "Extrapolation" setting [57]. We retrained latentSplat on Interpolation and MVSplat on Extrapolation for fair comparisons.

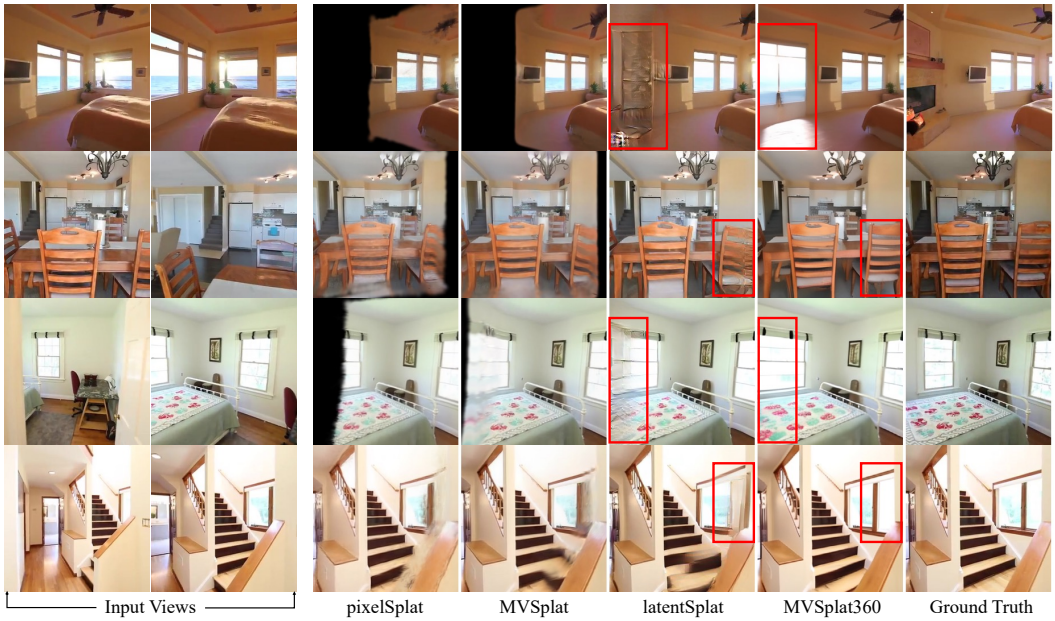

Figure 4: **Qualitative comparisons on RealEstate10K**. MVSplat360 shows reasonable generations for disoccluded and unobserved regions, while latentSplat [57] fills in content with artifacts.

Table 3: **MVSplat360 ablations**. All models are trained and evaluated on the DL3DV-10K dataset.

Models	SSIM↑	LPIPS↓	DISTS↓	FID↓
Baseline	0.433	0.501	0.291	78.95
+ SVD	0.399	0.556	0.248	38.05
+ ctx-attn	0.467	0.451	0.185	22.78
+ GS-feat.	**0.514**	**0.418**	**0.175**	**17.01**

(a) **Model components**. The baseline is the original MVSplat [10]. 'ctx-attn' refers to using multiple context views to enhance the cross attention, while 'GS-feat.' is the default model where SVD is conditioned with the 3DGS rendered features.

Views	SSIM↑	LPIPS↓	DISTS↓	FID↓
3 views	0.432	0.485	0.203	21.40
4 views	0.464	0.448	0.187	18.79
default	0.514	0.418	0.175	17.01
6 views	0.514	0.401	0.169	16.54
7 views	**0.526**	**0.390**	**0.166**	**16.26**

(b) **Number of input views**. The 'default' model is trained and tested with 5 views, while the others are directly evaluated with different numbers of input views during testing.

Quantitative results. Table 2 shows quantitative comparisons on RealEstate10K [74] of MVSplat360 and other approaches. Our MVSplat360 surpasses all previous state-of-the-art methods, mainly in terms of the perceptual metrics and the distribution metric. The former implies that our rendered views are more aligned with human perception, while the latter shows that our refined images correspond better to the dataset distribution. These observations can be further confirmed by visual assessment.

Qualitative results. The qualitative comparisons of the top four best models are in Fig. 4. PixelSplat [6] and MVSplat [10] fail to render any content for the unobserved regions due to the lack of generative capability. In contrast, latentSplat can perform extrapolation via its GAN-based decoder, improving the overall visual quality. However, we observed that the content generated by latentSplat is not visually reasonable. Our MVSplat360 generates more plausible content (see the "window" in 1st row and "chair" in 2nd row), thanks to the stronger generative capability of the diffusion model.

4.4 Ablations and Analysis

We conduct ablations on DL3DV-10K [23] to analyze MVSplat360 further. Results are reported in Table 3, and discussed in detail next.

Accessing model components. The baseline refers to MVSplat [10] since our model is built on top of it. (i) A natural extension is to render novel views from MVSplat and use them directly as conditions

https://doi.org/10.52202/079017-3399

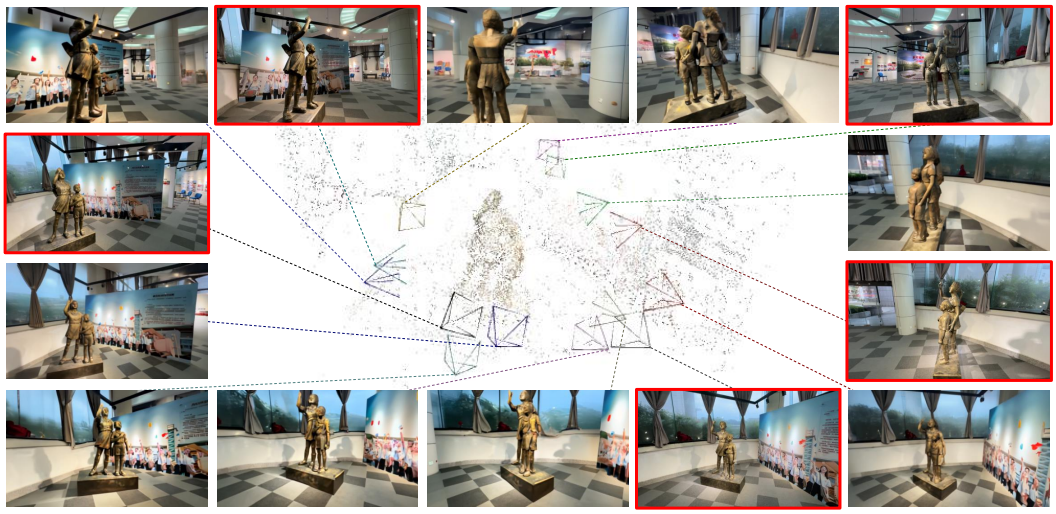

Figure 5: **SfM on input and rendered views**. Images with red borders are the input views, while others are rendered by our MVSplat360. The reasonably recovered camera poses and 3D point clouds via VGGSfM imply that our outputs are multi-view consistent and geometrically correct.

in the SVD denoising process. However, this straightforward approach performs slightly worse than the original MVSplat, likely because SVD struggles to infer pose and visual cues from the noisy image-encoded features. (ii) To better utilize the input context views, we average CLIP-embedded tokens from all views instead of just the first. This provides SVD with richer scene information via the cross-attention blocks, leading to a noticeable improvement. (iii) Lastly, we render high-dimensional features via the 3DGS rasterizer and concatenate them directly into the diffusion latent space, enabling gradients from the denoising UNet to backpropagate through the geometry backbone. This end-to-end training improves performance significantly and is used in our default model.

Accessing the number of input views. As shown in Table 4b, while our model is only trained with 5 input views, the performance can be gradually improved by adding more input views at testing. This is reasonable as more input views can provide more observable areas. On the contrary, reducing input views will inevitably result in worse performance. Surprisingly, even with 3 sparse views, our MVSplat360 still outperforms regression models (pixelSplat and MVSplat) that use 5 views.

Assessing the geometry accuracy. Our MVSplat360 builds on the video diffusion model SVD, which does ensure strong temporal/multi-frame consistency but does not inherently guarantee geometric accuracy. To confirm that our MVSplat360 produces geometrically accurate outputs, we run structure-from-motion (SfM) on both the input source views and the rendered novel views using VGGSfM [51]. As shown in Fig. 5, VGGSfM recovers reasonable camera poses and 3D point clouds, confirming that our novel views are both multi-view consistent and geometrically correct. This highlights how the 3DGS backbone's latent features provide essential geometric cues, enhancing 3D consistency in the final SVD-based outputs.

5 Conclusion

We present MVSplat360, a feed-forward model that synthesises 360° novel views of diverse real-world scenes from sparse input views. Our MVSplat360 leverages a feed-forward 3DGS model for recovering the coarse geometry and appearance from sparse observations of a 3D scene, which are then used to render latent features as the pose and visual cues to guide the following SVD in generating 3D consistent 360° novel views. To demonstrate its effectiveness, we construct a challenging benchmark for 360° NVS of real-world scenes. Experimental results show that MVSplat360 achieves superior visual quality compared to other SoTA feed-forward approaches.

Acknowledgements This research is supported by the Monash FIT Start-up Grant. Dr. Chuanxia Zheng is supported by EPSRC SYN3D EP/Z001811/1.

References

[1] Barron, J.T., Mildenhall, B., Tancik, M., Hedman, P., Martin-Brualla, R., Srinivasan, P.P.: Mip-nerf: A multiscale representation for anti-aliasing neural radiance fields. In: ICCV. pp. 5855–5864 (2021)

[2] Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: CVPR. pp. 5470–5479 (2022)

[3] Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorenz, D., Levi, Y., English, Z., Voleti, V., Letts, A., et al.: Stable video diffusion: Scaling latent video diffusion models to large datasets. arXiv preprint arXiv:2311.15127 (2023)

[4] Chan, E.R., Nagano, K., Chan, M.A., Bergman, A.W., Park, J.J., Levy, A., Aittala, M., Mello, S.D., Karras, T., Wetzstein, G.: GeNVS: Generative novel view synthesis with 3D-aware diffusion models. In: Proceedings of the International Conference on Computer Vision (ICCV) (2023)

[5] Chang, H., Zhang, H., Jiang, L., Liu, C., Freeman, W.T.: Maskgit: Masked generative image transformer. In: CVPR. pp. 11315–11325 (2022)

[6] Charatan, D., Li, S., Tagliasacchi, A., Sitzmann, V.: pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. arXiv preprint arXiv:2312.12337 (2023)

[7] Chen, A., Xu, Z., Zhao, F., Zhang, X., Xiang, F., Yu, J., Su, H.: Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In: ICCV. pp. 14124–14133 (2021)

[8] Chen, Y., Wu, Q., Zheng, C., Cham, T.J., Cai, J.: Sem2nerf: Converting single-view semantic masks to neural radiance fields. In: ECCV. pp. 730–748. Springer (2022)

[9] Chen, Y., Xu, H., Wu, Q., Zheng, C., Cham, T.J., Cai, J.: Explicit correspondence matching for generalizable neural radiance fields. arXiv preprint arXiv:2304.12294 (2023)

[10] Chen, Y., Xu, H., Zheng, C., Zhuang, B., Pollefeys, M., Geiger, A., Cham, T.J., Cai, J.: Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. arXiv preprint arXiv:2403.14627 (2024)

[11] Deng, K., Liu, A., Zhu, J.Y., Ramanan, D.: Depth-supervised nerf: Fewer views and faster training for free. In: CVPR (2022)

[12] Ding, K., Ma, K., Wang, S., Simoncelli, E.P.: Image quality assessment: Unifying structure and texture similarity. TPAMI **44**(5), 2567–2581 (2020)

[13] Du, Y., Smith, C., Tewari, A., Sitzmann, V.: Learning to render novel views from wide-baseline stereo pairs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4970–4980 (2023)

[14] Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: CVPR. pp. 12873–12883 (2021)

[15] Gao, R., Holynski, A., Henzler, P., Brussee, A., Martin-Brualla, R., Srinivasan, P., Barron, J.T., Poole, B.: Cat3d: Create anything in 3d with multi-view diffusion models. arXiv (2024)

[16] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. In: NeurIPS. pp. 6626–6637 (2017)

[17] Ho, J., Chan, W., Saharia, C., Whang, J., Gao, R., Gritsenko, A., Kingma, D.P., Poole, B., Norouzi, M., Fleet, D.J., et al.: Imagen video: High definition video generation with diffusion models. arXiv preprint arXiv:2210.02303 (2022)

[18] Jin, H., Jiang, H., Tan, H., Zhang, K., Bi, S., Zhang, T., Luan, F., Snavely, N., Xu, Z.: Lvsm: A large view synthesis model with minimal 3d inductive bias. arXiv preprint arXiv:2410.17242 (2024)

 https://doi.org/10.52202/079017-3399

[19] Johari, M.M., Lepoittevin, Y., Fleuret, F.: Geonerf: Generalizing nerf with geometry priors. In: CVPR. pp. 18365–18375 (2022)

[20] Kanazawa, A., Tulsiani, S., Efros, A.A., Malik, J.: Learning category-specific mesh reconstruction from image collections. In: ECCV. pp. 371–386 (2018)

[21] Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics (ToG) **42**(4), 1–14 (2023)

[22] Kumari, N., Su, G., Zhang, R., Park, T., Shechtman, E., Zhu, J.Y.: Customizing text-to-image diffusion with camera viewpoint control. arXiv preprint arXiv:2404.12333 (2024)

[23] Ling, L., Sheng, Y., Tu, Z., Zhao, W., Xin, C., Wan, K., Yu, L., Guo, Q., Yu, Z., Lu, Y., et al.: Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. arXiv preprint arXiv:2312.16256 (2023)

[24] Liu, F., Sun, W., Wang, H., Wang, Y., Sun, H., Ye, J., Zhang, J., Duan, Y.: Reconx: Reconstruct any scene from sparse views with video diffusion model. arXiv preprint arXiv:2408.16767 (2024)

[25] Liu, M., Shi, R., Chen, L., Zhang, Z., Xu, C., Wei, X., Chen, H., Zeng, C., Gu, J., Su, H.: One-2-3-45++: Fast single image to 3d objects with consistent multi-view generation and 3d diffusion. arXiv preprint arXiv:2311.07885 (2023)

[26] Liu, M., Xu, C., Jin, H., Chen, L., Xu, Z., Su, H., et al.: One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. arXiv preprint arXiv:2306.16928 (2023)

[27] Liu, R., Wu, R., Van Hoorick, B., Tokmakov, P., Zakharov, S., Vondrick, C.: Zero-1-to-3: Zero-shot one image to 3d object. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 9298–9309 (2023)

[28] Liu, X., Zhou, C., Huang, S.: 3dgs-enhancer: Enhancing unbounded 3d gaussian splatting with view-consistent 2d diffusion priors. arXiv preprint arXiv:2410.16266 (2024)

[29] Lombardi, S., Simon, T., Saragih, J., Schwartz, G., Lehrmann, A., Sheikh, Y.: Neural volumes: learning dynamic renderable volumes from images. TOG **38**(4), 1–14 (2019)

[30] Melas-Kyriazi, L., Laina, I., Rupprecht, C., Neverova, N., Vedaldi, A., Gafni, O., Kokkinos, F.: Im-3d: Iterative multiview diffusion and reconstruction for high-quality 3d generation. arXiv preprint arXiv:2402.08682 (2024)

[31] Mildenhall, B., Srinivasan, P., Tancik, M., Barron, J., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: ECCV (2020)

[32] Miyato, T., Jaeger, B., Welling, M., Geiger, A.: Gta: A geometry-aware attention mechanism for multi-view transformers. In: ICLR (2024)

[33] Niemeyer, M., Barron, J.T., Mildenhall, B., Sajjadi, M.S., Geiger, A., Radwan, N.: Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In: CVPR (2022)

[34] Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML. pp. 8748–8763. PMLR (2021)

[35] Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR. pp. 10684–10695 (2022)

[36] Sajjadi, M.S., Meyer, H., Pot, E., Bergmann, U., Greff, K., Radwan, N., Vora, S., Lučić, M., Duckworth, D., Dosovitskiy, A., et al.: Scene representation transformer: Geometry-free novel view synthesis through set-latent scene representations. In: CVPR. pp. 6229–6238 (2022)

[37] Salimans, T., Ho, J.: Progressive distillation for fast sampling of diffusion models. arXiv preprint arXiv:2202.00512 (2022)

[38] Sargent, K., Li, Z., Shah, T., Herrmann, C., Yu, H.X., Zhang, Y., Chan, E.R., Lagun, D., Fei-Fei, L., Sun, D., Wu, J.: ZeroNVS: Zero-shot 360-degree view synthesis from a single real image. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023)

[39] Singer, U., Polyak, A., Hayes, T., Yin, X., An, J., Zhang, S., Hu, Q., Yang, H., Ashual, O., Gafni, O., et al.: Make-a-video: Text-to-video generation without text-video data. In: ICLR (2022)

[40] Sitzmann, V., Rezchikov, S., Freeman, B., Tenenbaum, J., Durand, F.: Light field networks: Neural scene representations with single-evaluation rendering. NeurIPS **34**, 19313–19325 (2021)

[41] Sitzmann, V., Zollhöfer, M., Wetzstein, G.: Scene representation networks: Continuous 3d-structure-aware neural scene representations. NeurIPS **32** (2019)

[42] Smart, B., Zheng, C., Laina, I., Prisacariu, V.A.: Splatt3r: Zero-shot gaussian splatting from uncalibarated image pairs. arXiv preprint arXiv:2408.13912 (2024)

[43] Suhail, M., Esteves, C., Sigal, L., Makadia, A.: Generalizable patch-based neural rendering. In: ECCV. pp. 156–174. Springer (2022)

[44] Suhail, M., Esteves, C., Sigal, L., Makadia, A.: Light field neural rendering. In: CVPR (2023)

[45] Szymanowicz, S., Insafutdinov, E., Zheng, C., Campbell, D., Henriques, J.F., Rupprecht, C., Vedaldi, A.: Flash3d: Feed-forward generalisable 3d scene reconstruction from a single image. arXiv preprint arXiv:2406.04343 (2024)

[46] Szymanowicz, S., Rupprecht, C., Vedaldi, A.: Splatter image: Ultra-fast single-view 3d reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2024)

[47] Tang, J., Chen, Z., Chen, X., Wang, T., Zeng, G., Liu, Z.: LGM: Large multi-view Gaussian model for high-resolution 3D content creation. arXiv preprint arXiv:2402.05054 (2024)

[48] Truong, P., Rakotosaona, M.J., Manhardt, F., Tombari, F.: Sparf: Neural radiance fields from sparse and noisy poses. In: CVPR (2023)

[49] Tseng, H.Y., Li, Q., Kim, C., Alsisan, S., Huang, J.B., Kopf, J.: Consistent view synthesis with pose-guided diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 16773–16783 (2023)

[50] Voleti, V., Yao, C.H., Boss, M., Letts, A., Pankratz, D., Tochilkin, D., Laforte, C., Rombach, R., Jampani, V.: Sv3d: Novel multi-view synthesis and 3d generation from a single image using latent video diffusion. arXiv preprint arXiv:2403.12008 (2024)

[51] Wang, J., Karaev, N., Rupprecht, C., Novotny, D.: Vggsfm: Visual geometry grounded deep structure from motion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21686–21697 (2024)

[52] Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., Jiang, Y.G.: Pixel2mesh: Generating 3d mesh models from single rgb images. In: ECCV. pp. 52–67 (2018)

[53] Wang, Q., Wang, Z., Genova, K., Srinivasan, P.P., Zhou, H., Barron, J.T., Martin-Brualla, R., Snavely, N., Funkhouser, T.: Ibrnet: Learning multi-view image-based rendering. In: CVPR. pp. 4690–4699 (2021)

[54] Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. TIP **13**(4) (2004)

[55] Wang, Z., Yuan, Z., Wang, X., Chen, T., Xia, M., Luo, P., Shan, Y.: Motionctrl: A unified and flexible motion controller for video generation. arXiv preprint arXiv:2312.03641 (2023)

[56] Weng, H., Yang, T., Wang, J., Li, Y., Zhang, T., Chen, C., Zhang, L.: Consistent123: Improve consistency for one image to 3d object synthesis. arXiv preprint arXiv:2310.08092 (2023)

 https://doi.org/10.52202/079017-3399

[57] Wewer, C., Raj, K., Ilg, E., Schiele, B., Lenssen, J.E.: latentsplat: Autoencoding variational gaussians for fast generalizable 3d reconstruction. arXiv preprint arXiv:2403.16292 (2024)

[58] Wu, R., Mildenhall, B., Henzler, P., Park, K., Gao, R., Watson, D., Srinivasan, P.P., Verbin, D., Barron, J.T., Poole, B., Holynski, A.: Reconfusion: 3d reconstruction with diffusion priors. arXiv (2023)

[59] Wu, S., Li, R., Jakab, T., Rupprecht, C., Vedaldi, A.: Magicpony: Learning articulated 3d animals in the wild. In: CVPR. pp. 8792–8802 (2023)

[60] Xu, H., Chen, A., Chen, Y., Sakaridis, C., Zhang, Y., Pollefeys, M., Geiger, A., Yu, F.: Murf: Multi-baseline radiance fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2024)

[61] Xu, H., Peng, S., Wang, F., Blum, H., Barath, D., Geiger, A., Pollefeys, M.: Depthsplat: Connecting gaussian splatting and depth. arXiv preprint arXiv:2410.13862 (2024)

[62] Xu, Y., Shi, Z., Yifan, W., Chen, H., Yang, C., Peng, S., Shen, Y., Wetzstein, G.: Grm: Large gaussian reconstruction model for efficient 3d reconstruction and generation. arXiv preprint arXiv:2403.14621 (2024)

[63] Yang, J., Cheng, Z., Duan, Y., Ji, P., Li, H.: Consistnet: Enforcing 3d consistency for multi-view images diffusion. arXiv preprint arXiv:2310.10343 (2023)

[64] Ye, J., Wang, P., Li, K., Shi, Y., Wang, H.: Consistent-1-to-3: Consistent image to 3d view synthesis via geometry-aware diffusion models. In: Proceedings of the International Conference on 3D Vision (3DV) (2024)

[65] Ye, M., Yin, P., Lee, W.C., Lee, D.L.: Exploiting geographical influence for collaborative point-of-interest recommendation. In: Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval. pp. 325–334 (2011)

[66] Yu, A., Ye, V., Tancik, M., Kanazawa, A.: pixelnerf: Neural radiance fields from one or few images. In: CVPR. pp. 4578–4587 (2021)

[67] Yu, H., Long, X., Tan, P.: Lm-gaussian: Boost sparse-view 3d gaussian splatting with large model priors. arXiv preprint arXiv:2409.03456 (2024)

[68] Yu, W., Xing, J., Yuan, L., Hu, W., Li, X., Huang, Z., Gao, X., Wong, T.T., Shan, Y., Tian, Y.: Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. arXiv preprint arXiv:2409.02048 (2024)

[69] Yu, Z., Peng, S., Niemeyer, M., Sattler, T., Geiger, A.: Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. NeurIPS (2022)

[70] Zhang, K., Bi, S., Tan, H., Xiangli, Y., Zhao, N., Sunkavalli, K., Xu, Z.: GS-LRM: large reconstruction model for 3D Gaussian splatting. arXiv preprint arXiv:2404.19702 (2024)

[71] Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR. pp. 586–595 (2018)

[72] Zheng, C., Vedaldi, A.: Free3d: Consistent novel view synthesis without 3d representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2024)

[73] Zheng, C., Vuong, T.L., Cai, J., Phung, D.: Movq: Modulating quantized vectors for high-fidelity image generation. NeurIPS **35**, 23412–23425 (2022)

[74] Zhou, T., Tucker, R., Flynn, J., Fyffe, G., Snavely, N.: Stereo magnification: Learning view synthesis using multiplane images. ACM Trans. Graph **37** (2018)

[75] Ziwen, C., Tan, H., Zhang, K., Bi, S., Luan, F., Hong, Y., Fuxin, L., Xu, Z.: Long-lrm: Long-sequence large reconstruction model for wide-coverage gaussian splats. arXiv preprint arXiv:2410.12781 (2024)

A More Experiment Results

Assessing the robustness of SVD's first-stage autoencoder to input resolution. Since our MVSplat360 fine-tunes the denoising process in latent space, it is essential to ensure the autoencoder accurately maps between pixel and latent spaces. We observe that when the input resolution (256×480) differs significantly from the autoencoder's pre-trained resolution (768×1280), the autoencoding process causes noticeable information loss, leading to missing details (see Fig. A 2nd column) with an average PSNR of 26.77dB on the test set. Although fine-tuning the autoencoder might address this limitation, it risks overfitting due to our relatively small-scale training set (around 2000 scenes). Instead, we find that simply upscaling the input $2\times$ via bilinear interpolation before feeding them to the encoder helps preserve details effectively (see Fig. A 3rd column) and improves the PSNR to 32.52dB, ensuring the latent space remains accurate.

Original Images | Auto-encoded Images (w/o upscale) | Auto-encoded Images (w/ upscale)

Figure A: **Autoencoder using inputs with different resolutions**. The SVD's first-stage encoder and decoder are sensitive to image resolution. Hence, to ensure that images are encoded to the expected latent space, we upscale them 2 times via bilinear interpolation before feeding them to SVD.

B More Implementation Details

Following MVSplat [10], we set the near and far depth range used in the cost volume construction as 1 and 100, respectively. We render features from the 3D Guassians rasterizer with shape $4 \times h \times w$, where h and w refer to the image height and width, respectively, and the channel dimension is set to 4 to align with that of the SVD initial conditional vector. We bilinearly interpolate the latent features to a resolution of $1/4h \times 1/4w$, matching the encoded features from images of size $2h \times 2w$, as the encoder downsamples inputs by a factor of 8. This design ensures proper projection into the initial latent space, as detailed in Appendix A. The SVD decoder outputs are then bilinearly interpolated from $2h \times 2w$ back to $h \times w$. Due to resource limitations, we mainly experiment on the "images_8" branch of the DL3DV-10K dataset, which contains images with resolution 256×480. Our default model is trained with the Adam optimizer, and the learning rate is set to $1.e-5$ and decayed with the one-cycle strategy. All models are trained for 100,000 steps with an effective batch size of 8 on 1 to 8 A100 GPUs, and we apply the gradient accumulation technique whenever needed. During training, we sampled 5 views as input views and another 14 views as targeted rendering views, with the intention of better aligning with the following SVD module, which is trained on videos with 14 frames.

C Limitations and Discussions

Although MVSplat360 achieves plausibly consistent 360° NVS and significantly outperforms previous works in visual quality by leveraging SVD, it also inherits several limitations from SVD. For instance, the results may exhibit oversaturated colors (see Fig. B, left), limiting improvements on pixel-aligned metrics like PSNR and SSIM. This is likely because SVD is primarily trained on artistic videos with vibrant colors, and fine-tuning from its pre-trained weight can bias the outputs toward the original

https://doi.org/10.52202/079017-3399

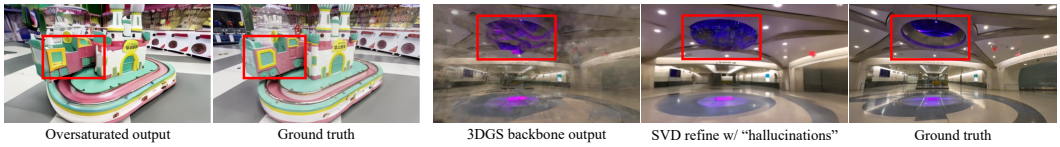

Figure B: **Limitations**. Left: Views may appear oversaturated; Right: Refined views can contain over-hallucinated contents. Both limitations stem from the diffusion module's prior knowledge.

training data distribution [50, 30]. Additionally, the outputs might exhibit hallucinations and contain contents not existing in the input views (see Fig. B, right). Lastly, the inference is slow due to the multiple sampling steps in the diffusion process. We expect these limitations to be mitigated as better video diffusion models and pre-trained weights become available in the future.

D Broader Social Impacts

Our MVSplat360 renders 360° novel views from sparse observations, making it a valuable tool for augmented reality applications. It can enhance immersive experiences in entertainment and media, including 3D videos and video games, and support historical reconstruction for educational purposes. However, MVSplat360 's powerful generative capabilities could be misused to create fake videos. Additionally, while the rendered views are high quality, they may not fully capture real-world details. Therefore, precautions are necessary when using the generated data in safety-critical applications, such as training autonomous driving models.

E More Visual Comparisons.

Below, we provide more visual comparisons with existing state-of-the-art models.

Figure C: **More qualitative comparisons on DL3DV-10K**. Our MVSplat360 significantly outperforms all existing models in various complex scenes.

Figure D: **More qualitative comparisons on DL3DV-10K**. Our MVSplat360 significantly outperforms all existing models in various complex scenes.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our contributions are explicitly stated in the abstract and introduction sections.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

https://doi.org/10.52202/079017-3399

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations in the supplementary material.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: This paper does not include theoretical results.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

Justification: We provide all the information needed to reproduce the main experimental results of the paper.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We will release our code.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.

 https://doi.org/10.52202/079017-3399

- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification:We elaborate the implementation details of our paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [NA]

   Justification: This paper does not include experiments reporting error bars.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We report it in the implementation details.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This paper conforms, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the positive and negative social impact in the supplementary document.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

https://doi.org/10.52202/079017-3399

Answer: [NA]

Justification: The paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We have correctly cited and credited assets used by our work.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: This paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

 https://doi.org/10.52202/079017-3399