You See it, You Got it: Learning 3D Creation on Pose-Free Videos at Scale

Supplementary Material

1. Broader Impact and Limitations

Broader Impact: Our model facilitates open-world 3D content creation from large-scale video data, eliminating the need for costly 3D annotations. This can make 3D generation more accessible to industries like gaming, virtual reality, and digital media. By leveraging visual data from the rapidly growing Internet videos, it accelerates 3D creation in real-world applications. However, careful consideration of ethical issues, such as potential misuse in generating misleading or harmful content, is crucial. Ensuring that the data used is curated responsibly to avoid bias and privacy concerns is vital for safe deployment.

Limitations: While our model excels at long-sequence generation, it comes with some limitations regarding: 1) Inference Speed: The model requires several minutes for inference, making it challenging for real-time applications. Future work should aim to improve inference speed for real-time generation. 2) Focus on 3D Generation: The current model focuses only on 3D generation, avoiding the modeling of object motion. Future research could extend the model to simultaneously generate 3D and 4D content for dynamic scenes. 3) Model Scalability: While the data scaling approach is effective, the scalability of the model itself has not been explored. Expanding the model's architecture could enhance its capability to handle more complex and diverse 3D content.

2. Video Data Curation

Our WebVi3D dataset is sourced from Internet videos through an automated four-step data curation pipeline. In this section, we provide details on this process.

Step 1: Temporal-Spatial Downsampling. To enhance data curation efficiency, we downsample each video both temporally and spatially. Temporally, we retain one frame for every two by downsampling with a factor of two. Spatially, we adjust the downsampling factor according to the original resolution to ensure consistent visual appearance across different video aspect ratios. The final resolution is standardized to 480p in our experiment.

Step 2: Semantic-Based Dynamic Recognition We perform content recognition on each frame to identify potential dynamic regions. Following [18], we utilize the off-the-shelf instance segmentation model Mask R-CNN [11] to generate coarse motion masks $\mathcal{M}_m$ for potential dynamic objects, including humans, animals, and sports activities. If motion masks are present in more than half of the video frames, the sequence is deemed likely to contain dynamic regions and excluded from further processing.

Step 3: Flow-based Dynamic Filtering After filtering out videos with common dynamic objects, we implement a precise strategy to identify and exclude videos containing potential dynamic regions. Following [18], we use the pretrained RAFT [30] to compute the optical flow between consecutive frames. Based on the optical flow, we calculate the Sampson Distance, which measures the distance of each pixel to its corresponding epipolar line. Pixels exceeding a predefined threshold are marked to create a motion mask $\mathcal{M}_s$. The number of pixels in $\mathcal{M}_s$ serves as an indicator of the likelihood of motion in the current frame.

However, relying solely on this metric is unreliable, as most data are captured in real shots, where dynamic objects of interest are often concentrated near the center of the imaging plane. These moving regions may not occupy a significant portion of the frame. Therefore, we also consider the spatial location of the dynamic mask and propose a dynamic score $\mathcal{S}$ to evaluate the motion probability for each frame. Let H, W denote the height and width of an image, respectively. We define the central region as starting at $W' = 0.25 \times W, H' = 0.25 \times H$. The proportions of the mask occupying the entire image, Θ_i, and the central area Θ_c are calculated as:

$$\Theta_i = \frac{\Sigma_{u,v=0}^{W,H} \mathcal{M}_s(u,v)}{H \times W}, \Theta_c = \frac{\Sigma_{u,v=W',H'}^{W-W',H-H'} \mathcal{M}_s(u,v)}{H/2 \times W/2}. \quad (1)$$

The dynamic score $\mathcal{S}$ can be formulated as:

$$\mathcal{S}_i = \begin{cases} 2, & \Theta_i \geq 0.12 \ \ \& \ \ \Theta_c \geq 0.35 \\ 1.5, & \Theta_i \geq 0.12 \ \ \& \ \ 0.2 \leq \Theta_c < 0.35 \\ 1, & \Theta_i < 0.12 \ \ \& \ \ 0.2 \leq \Theta_c < 0.35 \\ 0.5, & \Theta_i < 0.12 \ \ \& \ \ \Theta_c < 0.2 \end{cases}. \quad (2)$$

This strategy targets the dynamic regions near the image center, enhancing data filtering accuracy. The final dynamic score $\mathcal{S}$ for the entire sequence is calculated as:

$$\mathcal{S} = \Sigma_{i=0}^{N} \mathcal{S}_i, \tag{3}$$

where N represents the total number of extracted frames. If $\mathcal{S} >= 0.25 \times N$, the sequence is classified as dynamic and subsequently excluded.

Step 4: Tracking-Based Small Viewpoint Filtering. The previous steps produced videos with static scenes. We require videos that contain multi-view images captured from a wider camera viewpoint. To achieve this, we track the motion trajectory of key points across frames and calculate the radius of the minimum outer tangent circle for each trajectory. Videos with a substantial number of radii below a defined threshold are classified as having small camera trajectories and are excluded. This procedure includes keypoint extraction, trajectory tracking, and circle fitting using RANSAC (Random Sample Consensus) [8].

Keypoint Extraction. To reduce computational complexity, we downsample the extracted video frames by selecting every fourth frame. SuperPoint [6] is then used to extract keypoints $\mathbf{K} \in \mathbb{R}^{N\times 2}$ from the first frame, where $N = 100$ represents the number of detected keypoints used to initialize tracking.

Trajectory Tracking. Keypoints are tracked across all frames using the pretrained CoTracker [14], which generates trajectories and visibility over time as:

$$\mathbf{T}_{\text{pred}}, \mathbf{V}_{\text{pred}} = \text{CoTracker}(\mathbf{I}, \text{queries} = \mathbf{K}). \tag{4}$$

Here, $\mathbf{I}$ denotes the input frames, $\mathbf{T}_{\text{pred}} \in \mathbb{R}^{1\times T\times N\times 2}$ represents the tracked positions of each keypoint over time, and $\mathbf{V}_{\text{pred}} \in \mathbb{R}^{1\times T\times N\times 1}$ indicates the visibility of each point.

Circle Fitting. For each tracked keypoint, a circle fitting method is applied to its trajectory, selecting only frames where the keypoint is visible ($\mathbf{V}_{\text{pred}} = 1$). Let $\mathbf{T}_{\text{visible}} \in \mathbb{R}^{M\times 2}$ be the filtered points, where M is the number of visible points. We then use the RANSAC-based circle fitting algorithm on $\mathbf{T}_{\text{visible}}$ to determine the circle's center $\mathbf{c} = (c_x, c_y)$ and radius r:

$$\mathbf{c}, r = \text{RANSAC}(\mathbf{T}_{\text{visible}}). \tag{5}$$

The RANSAC algorithm selects random subsets of three points to define candidate circles, computes the inliers, and optimizes for the circle with the highest inlier count and smallest radius. Finally, we count the number of circles with a radius smaller than a specified threshold, $r \leq 20$:

$$\text{count} = \sum_{i=1}^{N} \mathbb{I}(r_i \leq 20), \tag{6}$$

where $\mathbb{I}$ is the indicator function. The mean radius is also computed to provide an overall measure of circular motion. If the number of small-radius circles exceeds 40 and the average circular motion is less than 5, we classify this video as having small camera trajectories.

User Study. To verify the effectiveness of our data curation pipeline, we conducted a user study with a randomly selected set of 10,000 video clips before filtering. We require our users to evaluate videos based on two aspects: *static content* and *large-baseline trajectories*. Only videos meeting both criteria are classified as 3D-aware videos. Among these, 1,163 videos met our criteria for 3D-aware videos, accounting for 11.6% of the total validation set. After applying our data screening pipeline, we randomly selected 10,000 video clips for annotation. In this filtered set, 8,859 videos were identified as 3D-aware, yielding a ratio of 88.6%, represents a 77% improvement compared to the previous set. These results demonstrate the efficacy of our pipeline in filtering 3D-aware videos from large-scale Internet videos.

3. Technical Implementations

3.1. Details of Visual Conditional 3D Generation

Global Recovery of Metric Depth. As described in the main manuscript, offline depth estimation methods often suffer from *scale ambiguity* and *geometric estimation errors*. To ensure a reliable depth map for subsequent 3D reconstruction, we first perform pixel-wise depth scale alignment, followed by global recovery of the metric dense depth.

Here, we provide a detailed explanation of how to adapt sparse scales to the entire depth map. Denoting the recovered positions as the sparse guidance $\hat{d}_n^*$, we utilize LWLR to recover the whole depth map. Let (u, v) represent 2D positions, their depth $\hat{\boldsymbol{D}}_n$ can be fitted to the sparse guided points by minimizing the squared locally weighted distance, which is reweighted by the diagonal weight matrix as $\boldsymbol{W}_{u,v}$

$$\boldsymbol{W}_{u,v} = diag(w_1, w_2, ..., w_m), w_i = \frac{1}{\sqrt{2\pi}} \exp(-\frac{\text{dist}_i^2}{2b^2}), \tag{7}$$

where b is the bandwidth of Gaussian kernel, and dist is the Euclidean distance between the guided point and the underestimate target point. Denote $\boldsymbol{X}$ as the homogeneous representation of $\hat{\boldsymbol{D}}_n$. The scale map $\boldsymbol{S}_{scale}$ and shift map $\boldsymbol{S}_{shift}$ of target points can be computed by iterating over

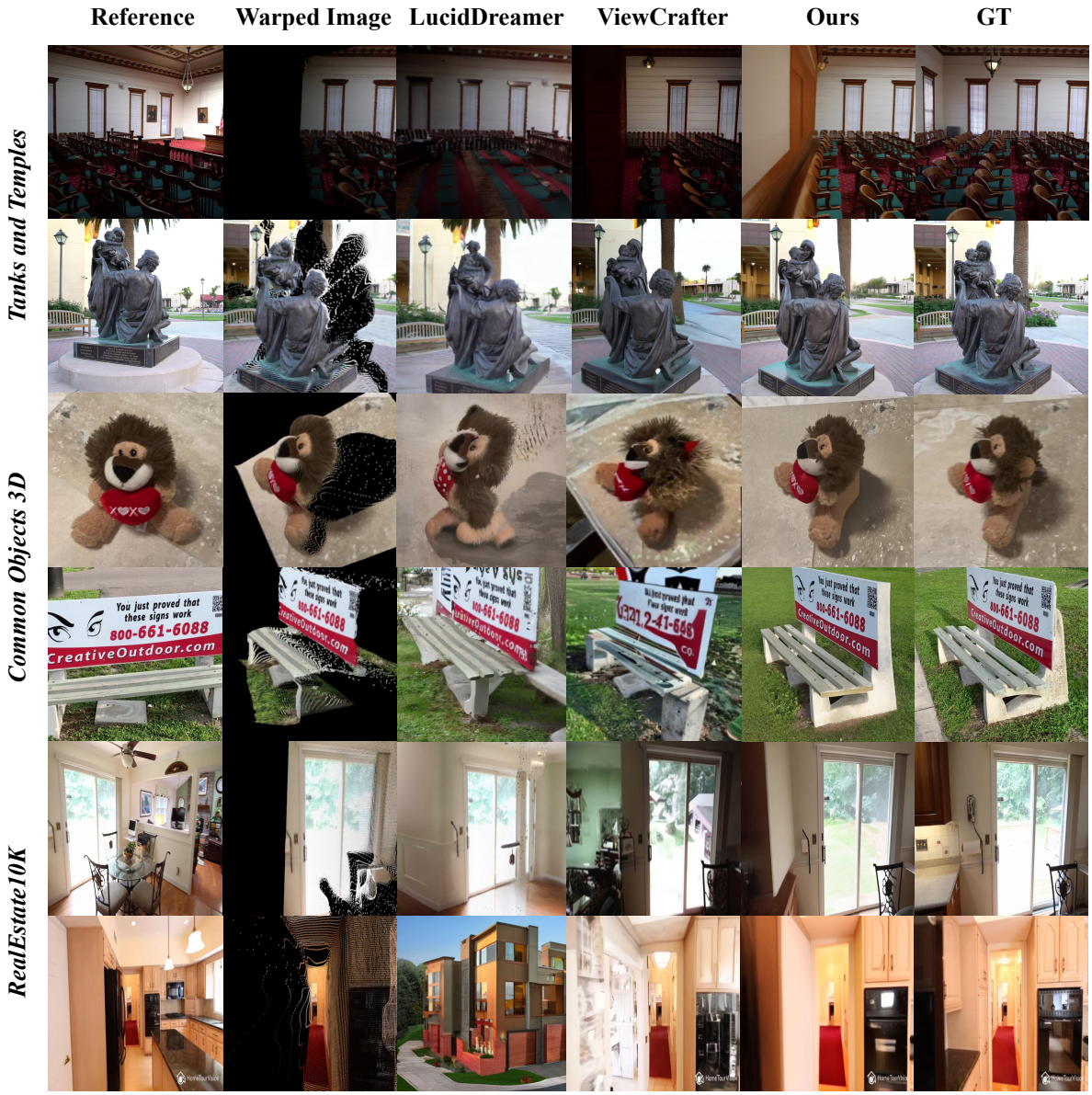


Figure 1. **Single-view to 3D**. Compared with LucidDreamer [3] and ViewCrafter [35], which are also conditioned on warped images, our model can consistently generate high-fidelity views with detailed texture and structural information.

each location in the entire image, formulated as:

$$\begin{aligned} &\min_{\boldsymbol{\beta}_{u,v}} (\hat{d}_n^* - \boldsymbol{X}\boldsymbol{\beta}_{u,v})^\top \boldsymbol{W}_{u,v} (\hat{d}_n^* - \boldsymbol{X}\boldsymbol{\beta}_{u,v}) + \lambda \boldsymbol{S}_{shift}^2, \\ &\hat{\boldsymbol{\beta}}_{u,v} = (\boldsymbol{X}^\top \boldsymbol{W}_{u,v} \boldsymbol{X} + \lambda)^{-1} \boldsymbol{X}^\top \boldsymbol{W}_{u,v} \hat{d}_n^*, \\ &\boldsymbol{\beta_{u,v}} = [\boldsymbol{S}_{scale}, \boldsymbol{S}_{shift}]_{u,v}^\top, \\ &\boldsymbol{D}_n = \hat{d}_n^* \oplus \boldsymbol{S}_{scale} \odot \hat{\boldsymbol{D}}_n + \boldsymbol{S}_{shift}, \end{aligned} \tag{8}$$

where $\boldsymbol{D}_n$ is the scaled whole depth map, $\oplus$ is the concatenation operator, and λ is the regularization hyperparameter l_2 used to simplify the solution. Additionally, the explicit constraint of the source frame with the target frames ensures that each novel view maintains contextual consistency with preceding generations.

Novel View Generation. After obtaining the aligned depth $\boldsymbol{D}_n$, we generate target visual hints through warping as $\hat{I}_j = \Pi_{n \to j}(\boldsymbol{D}_n)$. The warped images $\{\hat{I}_j\}_{j=n}^{n+m}$ contain unfilled regions, as indicated by the binary warping mask $\{M_j\}_{j=n}^{n+m}$, providing strong visual hints for **See3D** to perform novel view generation. To ensure strong multi-view consistency between the newly generated sequence and the previous content, we randomly select k anchor views

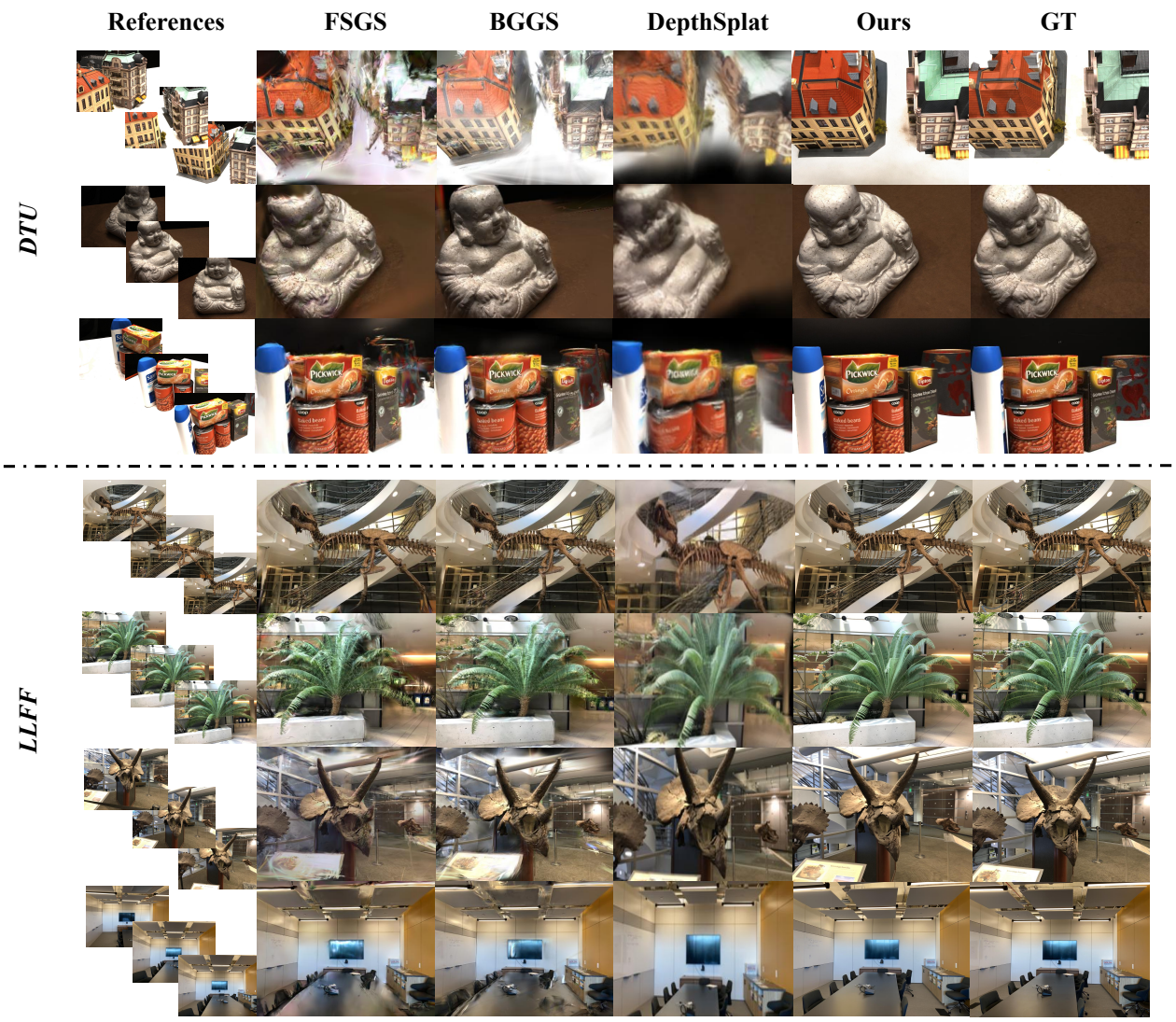


Figure 2. **Sparse-views to 3D**. Given 3 input views, our model generates clear, high-fidelity novel views that closely match the ground truth (GT), without artifacts or blurring. Note that the results from DepthSplat [34] are cropped and resized following the same data processing as the official source code.

$\{I_k\}, k \in [1, N]$ from the earlier generated frames to guide subsequent generation. The generation process is formulated as: $I_j = \textbf{See3D}(\hat{I}_j, M_j, \{I_0, I_k\})$. We iteratively perform depth estimation, alignment, warping, and generation until all predefined multi-view images are obtained.

3D Reconstruction. We reconstruct the 3D scene using 3D Gaussian Splatting (3DGS) [15]. The training objective is to minimize the sum of photometric loss and SSIM loss, consistent with the original 3DGS approach. Additionally, we introduce a perceptual loss (LPIPS [37]) to mitigate subtle *inter-frame* discrepancies in multi-view generated images during 3DGS reconstruction. LPIPS emphasizes higher-level semantic consistency between Gaussian-rendered and generated multi-view images, rather than focusing on minor high-frequency differences. Furthermore, the potential *inner-frame* diversity may lead to inconsistencies with the corresponding camera poses. Following [7], we implement joint pose-Gaussian optimization, treating camera parameters as learnable variables alongside Gaussian attributes, thereby reducing gaps between generated viewpoints and their corresponding camera poses.

3.2. Model Architecture

The main backbone of **See3D** model is based on the structure of 2D diffusion models but integrates 3D self-attention to connect the latents of multiple images, as shown in

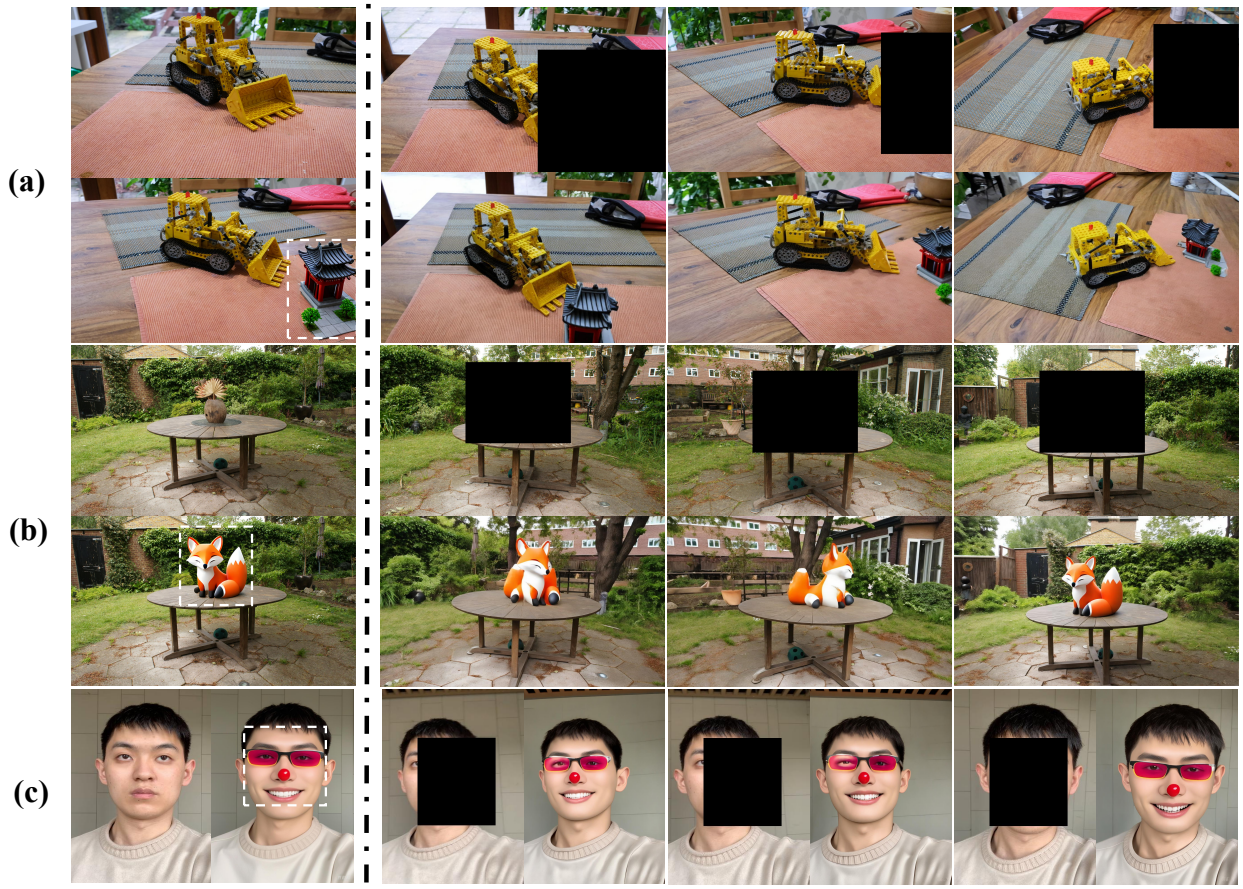


Figure 3. **Examples of Open-world 3D Editing**. (a) Occlusion-free Editing: An Asian-style attic is added, and novel views are generated realistically. (b) Full Replacement Editing: A vase is replaced with a toy fox, seamlessly integrated into the scene from various viewpoints. (c) Occluded Editing: Hidden regions in the masked areas are inferred and completed to produce novel views.

prior work [26]. Specifically, we adapt the existing 2D self-attention layers of the original 2D diffusion model into 3D self-attention by inflating different views within the self-attention layers. To incorporate visual conditions, we introduce the necessary convolutional kernels and biases using Zero-Initialize [27]. The model is initialized from a pretrained 2D diffusion model [21] and fine-tuned with all parameters, leveraging FlashAttention for acceleration. In accordance with prior work [25], switching from a scaled-linear noise schedule to a linear schedule is essential for achieving improved global consistency across multiple views. Additionally, we implement cross-attention between the latents of multiple views and per-token CLIP embeddings of reference images using a linear guidance mechanism [29]. For training, we randomly select a subset of frames from a video clip as reference images, with the remaining frames serving as target images. The number of reference images is randomly chosen to accommodate different downstream tasks. The multi-view diffusion model is optimized by calculating the loss only on the target images, as outlined in Eq. 1. of the main manuscript.

3.3. Training Details

Brightness Control. We observe that the *visual-condition* effectively guides camera movement but cannot control brightness changes, posing a significant limitation. Determining the light source position is particularly challenging with limited observations from single or sparse views. In our real-world test data, camera movement often causes random highlighting or darkening in some regions of scenes, which has a significant impact on pixel-level metrics like PSNR. This issue highlights a key problem: the inability to control brightness undermines the reliability of pixel-level metrics, as brightness variations affect these metrics more than the actual quality of the generated content. To achieve illumination control, 1) we preprocess the training data by

converting corrupted images into HSV format, which represents hue, saturation, and brightness. 2) We define a $w \times h$ window and calculate the average brightness difference within this window between the ground truth image and the corrupted data. Using this difference, we apply a scaling factor to the brightness channel of the corrupted data while preserving hue and saturation, before converting the image back to RGB. This ensures brightness adjustment in the *visual-condition* without leaking color or content from the ground truth.

During training, we randomly drop this preprocessing with a probability of 0.5, enabling the model to infer lighting changes on its own during inference when brightness control is not required. In our evaluation experiments, brightness scaling is applied to the unmasked regions of warped images to align with ground truth, reducing the impact of brightness, and thus yielding a higher correlation between the generated content and pixel-level metrics. Meanwhile, keeping hue and saturation unchanged to avoid content or color leakage. Additionally, the model enables user-controlled brightness adjustments for specific regions in multi-view generation by modifying the *visual-condition* as needed.

Training Configuration. We initialize the **See3D** model from MVDream [26] and employ a progressive training strategy. First, the model is trained at a resolution of 512 $\times$ 512 with a sequence length of 5. This phase involves 120,000 iterations, using 1 reference view and 4 target views. Due to the relatively small sequence length, a larger batch size of 560 is used to enhance stability and accelerate convergence. Next, the sequence length is increased to 16, and the model is trained for 200,000 iterations with 1 or 3 reference views and 15 or 13 target views, maintaining the resolution of 512 $\times$ 512. In this phase, the batch size is reduced to 228. Finally, a multi-view super-resolution model is trained using the same network structure. It takes the multi-view predictions from **See3D** as input and outputs target images with multi-view consistency at a resolution of 1024 $\times$ 1024, using a batch size of 114. In all stages, all parameters of the diffusion model are fine-tuned with a learning rate of 1e-5. Additionally, we render some multi-views or extract clips from datasets such as Objaverse [5], CO3D [23], RealEstate10k [38] , MVImgNet [36], and DL3DV [17] datasets, forming a supplemental 3D dataset with fewer than 0.5M samples, please refer to Section 5.3 for details on analysis and ablation. During training, this supplemental data is randomly sampled and incorporated into our WebVi3D dataset ($\sim$16M). To enhance training efficiency, we utilize FlashAttention [4] alongside DeepSpeed with ZeRO stage-2 optimizer [22] and bf16 precision. We also implement classifier-free guidance (CFG) [12] by randomly dropping visual conditions with a probability of 0.1. The **See3D** model is trained on 114 $\times$ NVIDIA-A100-SXM4-40GB GPUs over approximately 25 days using a progressive training scheme. During inference, a DDIM sampler [28] with classifier-free guidance is employed.

3.4. Definition of $f(t)$ and W_t

Definition for $f(t)$. In Eq. 2 of the main manuscript, C_t is formulated as $C_t = \sqrt{\bar{\alpha}_{t'}}(1-M)\mathbf{X}_0 + \sqrt{1-\bar{\alpha}_{t'}}\boldsymbol{\epsilon}$, where $\alpha_{t'}$ is a composite function that depends on α and t', with $t' = f(t)$ and $f(t) = \beta \cdot t$. In our experiments, we set the hyper-parameter $\beta = 0.2$, which controls the noise level added to C_t. A larger β increases the noise in C_t. As β approaches 1, C_t converges toward a Gaussian distribution, improving robustness but reducing the correlation between C_t and $\mathbf{X}_0$, thereby weakening camera control. Conversely, as β approaches 0, the distributions of C_t and $\mathbf{X}_0$ become more similar, improving controllability. However, for downstream tasks, a very small β creates a significant domain gap between task-specific visual cues and the video data, compromising robustness. Thus, β serves as a trade-off parameter, balancing camera control and robustness.

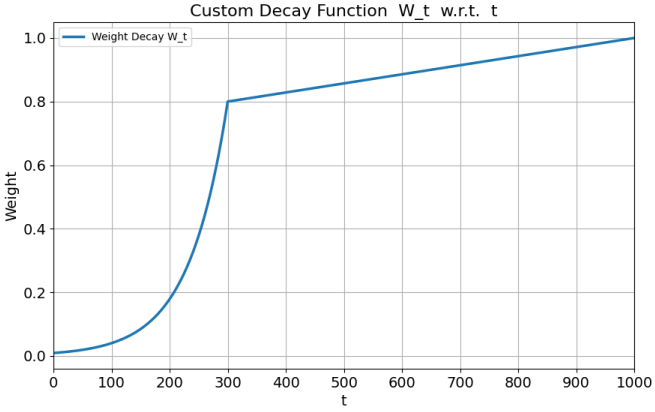


Figure 4. **Piecewise Function** W_t, showing linear decay for timesteps t between 300 and 1000, and a monotonically decreasing concave behavior for $t < 300$.

Formulation for W_t. Recapping Eq. 3 from the main manuscript, $V_t = [W_t * C_t + (1 - W_t) * X_t; M]$, where W_t is defined as a piecewise function of t.

$$W_t = \begin{cases} v_{\text{decay\_end}} \cdot e^{-b \cdot (t_{\text{decay\_end}} - t)}, & \text{if } t < t_{\text{decay\_end}}, \\ 1 - (1 - v_{\text{decay\_end}}) \cdot \frac{t_{\text{peak}} - t}{t_{\text{peak}} - t_{\text{decay\_end}}}, & \text{if } t \geq t_{\text{decay\_end}}, \end{cases}$$

where $t_{\text{peak}} = 1000$, $t_{\text{decay\_end}} = 300$, $v_{\text{decay\_end}} = 0.8$, and $b = 0.075$. To ensure that W_t remains within the range $[0, 1]$, it is clamped as: $W_t = \text{clamp}(W_t, 0, 1)$. As shown in Figure 4, 1) For t between 300 and 1000, W_t decreases linearly as t decreases; 2) For $t < 300$, W_t transitions to a monotonically decreasing concave function of t.

The rationale behind this design is to ensure that when C_t has significant noise, it exerts a stronger influence on V_t, thus affecting MVD generation. Conversely, as the noise in C_t diminishes, X_t rapidly replaces C_t, reducing the risk of information leakage from C_t and improving the robustness of task-specific visual cues. The formulation of W_t enables flexible parameter tuning, such as $v_{\text{decay\_end}}$ and b, to control its monotonic behavior. Smaller parameter values emphasize the impact of C_t on MVD, while larger values prioritize robustness.

4. More Experimental Results

Leveraging the developed web-scale dataset WebVi3D, our model supports both object- and scene-level 3D creation tasks, including single-view-to-3D, sparse-view-to-3D, and 3D editing. Additional experimental results for these tasks are presented below.

4.1. Single View to 3D

Table 1 presents a quantitative comparison of zero-shot novel view synthesis performance on the Tanks-and-Temples [16], RealEstate10K [38], and CO3D [23] datasets. Our method consistently outperforms all others on both easy and hard sets, achieving the best results in every evaluation metric. Qualitative results are shown in Figure 1. Compared to warping-based competitors such as LucidDreamer [3] and ViewCrafter [35], our approach more effectively captures both geometric structure and texture details, producing more realistic 3D scenes. These results highlight the robustness and versatility of our method in synthesizing high-quality novel views across diverse and challenging scenarios.

4.2. Sparse Views to 3D

Experimental Setting. We extend our model to the sparse-view reconstruction task, evaluating it on three datasets: LLFF [19], DTU [13], and Mip-NeRF 360 [1]. We compare our method against several few-shot 3D reconstruction baselines, including optimization-based method MuRF [33], FSGS [39], and BGGS [10]; diffusion-based methods CAT3D [9], ZeroNVS (modified to handle multi-view input) [24], and ReconFusion [32]; as well as the feed-forward method DepthSplat [34]. Following the evaluation protocols from [20, 32, 39], we use 3, 6, and 9 views as input. For few-shot reconstruction, dense multi-view images are generated from sparse views, similar to CAT3D [9], and 3DGS reconstruction is performed with pose optimization to render test views for evaluation. We report PSNR, SSIM, and LPIPS [37] to evaluate novel view synthesis performance.

Results. Qualitative and quantitative results are presented in main manuscript. The additional Quantitative comparisons using 3, 6, and 9 input views are presented in Table 2. The 3DGS model trained on multi-view images generated by **See3D** outperformed state-of-the-art models in novel view rendering, demonstrating its ability to provide consistent multi-view support for 3D reconstruction without additional constraints. Qualitative comparisons in Figure 2 reveal fewer floating artifacts in the NVS results. This indicates its ability to provide high-quality, consistent multi-view support for 3D reconstruction without imposing additional constraints. Compared to ReconFusion [32] and CAT3D [9], which also leverage diffusion priors for sparse-view reconstruction, our model exhibits effective scalability. Qualitative comparisons in Figure 5 reveal that NVS results produced by **See3D** exhibit fewer floating artifacts, suggesting its capability to generate more consistent and high-fidelity multi-view images.

4.3. 3D Editing

Our model, trained on large-scale videos, naturally supports open-world 3D editing without the need for additional fine-tuning. Our core idea is to mask the regions to be edited in the reference viewpoint from other viewpoints and validate the multi-view generation capability of See3D by assessing the consistency after editing. Specifically, We first select a region to be edited in the reference input image and re-project this region to other viewpoints using the given camera poses. We then randomly expand the masking region for other viewpoints (up to 30%) to ensure that the masking in these viewpoints covers the editing region in the reference viewpoint. During our editing experiments, we do not require highly consistent multi-view masking, it is sufficient to ensure that the regions to be edited are consistent across viewpoints and are completely covered. Figure 3 illustrates three distinct editing scenarios: a) *Occlusion-free Editing*. An Asian-style attic is placed next to a toy bulldozer in the original image, which serves as the reference view. Our model generates highly realistic images containing the Asian-style attic from various new viewpoints. b) *Full Replacement Editing*. The vase in the original image is completely replaced with a toy fox. Our model generates new scenes from different viewpoints, seamlessly incorporating the toy fox into the designated area with no residual traces of the vase. c) *Occluded Editing*. Given an occluded edited image as a reference view, our model can generate multiple novel views within the specified masked regions, inferring and filling in the hidden details of the occluded parts.

5. Additional Ablation Studies

5.1. Effectiveness of Visual-condition.

Excluding the benefits of data scaling, we investigate the effectiveness of our *visual-condition* on pose-free data. Pre-

Dataset	Easy set			Hard set		
Method	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑
Tanks-and-Temples						
LucidDreamer [3]	0.413	14.53	0.362	0.558	11.69	0.267
ZeroNVS [24]	0.482	14.71	0.380	0.569	12.05	0.309
MotionCtrl [31]	0.400	15.34	0.427	0.473	13.29	0.384
ViewCrafter	0.194	21.26	0.655	0.283	18.07	0.563
ViewCrafter*	0.221	20.39	0.648	0.289	17.86	0.584
Ours	0.167	25.01	0.756	0.214	22.52	0.714
RealEstate10K						
LucidDreamer [3]	0.315	16.35	0.579	0.400	14.13	0.511
ZeroNVS [24]	0.364	16.50	0.577	0.431	14.24	0.535
MotionCtrl [31]	0.341	16.31	0.604	0.386	16.29	0.587
ViewCrafter	0.145	21.81	0.796	0.178	22.04	0.798
ViewCrafter*	0.164	20.59	0.825	0.201	20.40	0.778
Ours	0.125	26.54	0.872	0.167	24.18	0.837
CO3D						
LucidDreamer [3]	0.429	15.11	0.451	0.517	12.69	0.374
ZeroNVS [24]	0.467	15.15	0.463	0.524	13.31	0.426
MotionCtrl [31]	0.393	16.87	0.529	0.443	15.46	0.502
ViewCrafter	0.243	21.38	0.687	0.324	18.96	0.641
ViewCrafter*	0.331	20.12	0.703	0.348	18.02	0.653
Ours	0.225	25.23	0.781	0.276	23.33	0.748

Table 1. **Zero-shot Novel View Synthesis (NVS) on Tanks-and-Temples[16], RealEstate10K[38] and CO3D[23] dataset.**

vious work [35] has demonstrated that warped images can serve as a pivot condition to guide the model to generate the target viewpoint. However, due to the reliance on the annotated camera to control the projection and unprojection, warp-based conditions are inherently unscalable. Therefore, we compare the model's ability to control cameras conditioned on pose-free *visual-condition* and conditioned on warped images. Specifically, we extract a subset of MVImageNet [36] for training and testing.

For each multi-view sequence in the training set, we select the point cloud of the first frame and render it into the subsequent 5 camera planes along the camera trajectory, based on the 3D annotations in the dataset. We obtain warped images and form pairs with the ground-truth multi-views to train an MVD model, referred to as MV-Posed. With the same experimental settings (training set, network architecture, batch size and predicted sequence length), we train an additional model without any 3D annotations, except for the modification of warp condition to the time-dependent *visual-condition* V_t described in Sec. 3.2, called MV-UnPoseT. Meanwhile, we employ randomly masked multiple views as condition to train the model as an additional baseline, called MV-UnPoseM.

The results are reported in Tab.3 and Fig.5, where the performance of MV-Posed and MV-UnPoseT is comparable. In contrast, MV-UnPoseM struggles to handle the gap between the warped image and masked images, in the case of geometric distortion and self-obscuration. These findings indicate that the *visual-condition* offers a viable alternative

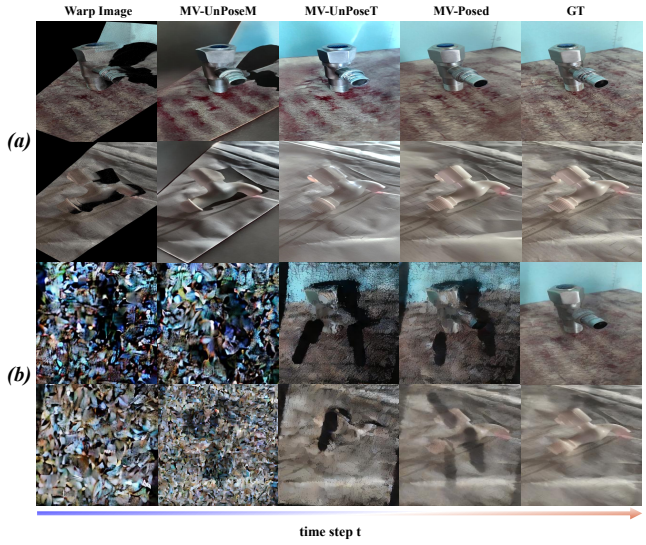


Figure 5. Top: Qualitative ablation of *visual-condition*; Bottom: As timestep decreases, visualize the trend of *visual-condition*.

Dataset	3-view			6-view			9-view		
Method	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓
LLFF									
Zip-NeRF* [2]	17.23	0.574	0.373	20.71	0.764	0.221	23.63	0.830	0.166
MuRF [33]	21.34	0.722	0.245	23.54	0.796	0.199	24.66	0.836	0.164
FSGS [39]	20.31	0.652	0.288	24.20	0.811	0.173	25.32	0.856	0.136
BGGS [10]	21.44	0.751	0.168	24.84	0.845	0.106	26.17	0.877	0.090
ZeroNVS* [24]	15.91	0.359	0.512	18.39	0.449	0.438	18.79	0.470	0.416
DepthSplat [34]	17.64	0.521	0.321	17.40	0.499	0.340	17.26	0.486	0.341
ReconFusion [32]	21.34	0.724	0.203	24.25	0.815	0.152	25.21	0.848	0.134
CAT3D [9]	21.58	0.731	0.181	24.71	0.833	0.121	25.63	0.860	0.107
Ours	23.23	0.768	0.135	25.32	0.820	0.104	26.19	0.844	0.098
DTU									
Zip-NeRF* [2]	9.18	0.601	0.383	8.84	0.589	0.370	9.23	0.592	0.364
MuRF [33]	21.31	0.885	0.127	23.74	0.921	0.095	25.28	0.936	0.084
FSGS [39]	17.34	0.818	0.169	21.55	0.880	0.127	24.33	0.911	0.106
BGGS [10]	20.71	0.862	0.111	24.31	0.917	0.073	26.70	0.947	0.052
ZeroNVS* [24]	16.71	0.716	0.223	17.70	0.737	0.205	17.92	0.745	0.200
DepthSplat [34]	15.59	0.525	0.373	15.061	0.523	0.406	14.87	0.478	0.451
ReconFusion [32]	20.74	0.875	0.124	23.62	0.904	0.105	24.62	0.921	0.094
CAT3D [9]	22.02	0.844	0.121	24.28	0.899	0.095	25.92	0.928	0.073
Ours	28.04	0.884	0.073	29.09	0.900	0.066	29.99	0.911	0.059
Mip-NeRF 360									
Zip-NeRF* [2]	12.77	0.271	0.705	13.61	0.284	0.663	14.30	0.312	0.633
DepthSplat [34]	13.85	0.254	0.621	13.82	0.260	0.636	14.48	0.288	0.602
ZeroNVS* [24]	14.44	0.316	0.680	15.51	0.337	0.663	15.99	0.350	0.655
ReconFusion [32]	15.50	0.358	0.585	16.93	0.401	0.544	18.19	0.432	0.511
CAT3D [9]	16.62	0.377	0.515	17.72	0.425	0.482	18.67	0.460	0.460
Ours	17.35	0.442	0.422	19.03	0.517	0.365	19.89	0.542	0.335

Table 2. **Quantitative Comparison of Sparse-view 3D Reconstruction**

Model	LPIPS ↓	PSNR ↑	SSIM ↑
MV-Posed	0.182	26.21	0.822
MV-UnPoseM	0.443	16.14	0.521
MV-UnPoseT	0.194	25.56	0.811

Table 3. **Ablation Study on Visual-condition**.

to 3D-reliant warped conditions. Despite a significant domain gap between V_t and warp images as shown in Fig.5, our model robustly handles this discrepancy, thanks to the time-dependent nature of the proposed condition.

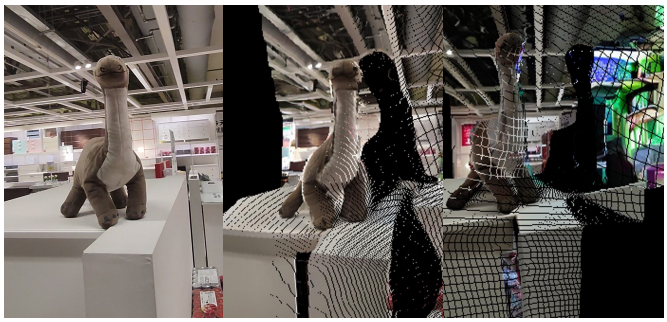
5.2. Effectiveness of Pixel-level Depth Alignment

We conducted additional ablation experiments to validate the effectiveness of the proposed pixel-level depth alignment. Specifically, we enabled and disabled pixel-level depth alignment when generating novel views through warping and visualized the warped results at a specific generation step. As shown in Figure 6, the left image shows the reference GT image, the middle image corresponds to warping with pixel-level aligned depth, and the right one depicts warping without pixel-level aligned depth. The results demonstrate that pixel-level depth alignment not only effectively restores the scale of the depth map but also significantly corrects errors in monocular depth estimation (e.g., the toy's neck and the tabletop). Consequently, integrating our proposed 3D generation pipeline improves generation quality.

5.3. Efficacy of Scaling up Data

In the main manuscript, we conducted an ablation study on the 3D dataset MVImageNet [36] to evaluate the effectiveness of the proposed *visual-condition*. Table 3 shows that: 1) When conditioned on purely masked images, the MV-UnPoseM model performed the worst, struggling with the domain gap issue. 2) When conditioned on pose-guided warped images, the MV-Posed model achieved the best results, benefiting from pose annotations. 3) Our MV-UnPoseT model, conditioned on the time-dependent *visual-condition*, demonstrated performance very close to that of the MV-Posed model.

Intuitively, models trained entirely on 3D data tend to achieve optimal performance at a specific data scale, establishing an upper bound at that scale. When the volume of video data matches that of 3D data, models trained on 3D



Reference GT with Pixel-level Align without Pixel-level Align

Figure 6. **Ablation on Pixel-level Depth Alignment.**

Model	LPIPS ↓	PSNR ↑	SSIM ↑
MV-UnPoseT	0.194	25.56	0.811
MV-UnPoseT-10%	0.187	25.95	0.817
MV-UnPoseT-20%	0.183	26.19	0.820
MV-UnPoseT-60%	0.181	26.14	0.819
MV-Posed	0.182	26.21	0.822

Table 4. **Ablation on Supplementary 3D Data.**

still set the performance ceiling. However, as video data is virtually unlimited, scaling up the dataset can intuitively raise this upper bound.

Following the same settings in Table 3, we further investigate the impact of supplementing multi-view data with 3D annotations on model performance. We conduct an ablation study using the MV-UnPoseT model, trained on unposed multi-view data with *visual-condition*. In this study, we progressively introduce 3D pose annotations at levels of 10%, 20%, 60%, and 100% into the training set. When the training data is entirely composed of 3D annotations, the model configuration is equivalent to the MV-Posed model. The results in Table 4 indicate that our MV-UnPoseT model, initially trained on unposed data, improves steadily as 3D annotations are introduced. For instance, with only 20% 3D data (MV-UnPoseT-20%), the model's performance closely approaches that of the fully 3D-annotated MV-Posed model. This suggests that even a small amount of 3D data in a largely unposed multi-view dataset can significantly boost model performance, approaching the models trained on fully annotated 3D datasets.

This insight is essential because unposed multi-view data is cost-effective and can be easily collected in large quantities. By incorporating a small volume of high-quality 3D data, we can achieve performance comparable to models trained on large, expensive 3D datasets. Therefore, in our proposed WebVi3D dataset (16M samples), we incorporated a small portion (0.5M samples) of 3D data to optimize model performance.

6. Additional Visualizations

Open-world 3D Generation with Long Sequences. We manually configured complex camera trajectories, including rotation, translation, zooming in, zooming out, focus distance adjustments and various random combinations, as shown in Figure 8 and Figure 7. Our model consistently generates high-quality, continuous novel views along these trajectories. Experimental visualizations demonstrate that the model effectively preserves spatial consistency and visual realism across long sequences. This highlights its robustness in handling intricate camera paths, including rapid transitions and diverse perspectives, making it highly applicable to open-world scenarios.

7. Acknowledgements

We thank Wenyuan Zhang and Yu-Shen Liu from Tsinghua University, as well as Yance Jiao, Hua Zhou, Liao Zhang, Yaohui Chen, Jinxin Xie, Yiwen Shao, and other colleagues from BAAI, for their valuable support and contributions to the See3D project.

Figure 7. **More Examples of Long-sequence Generation.**

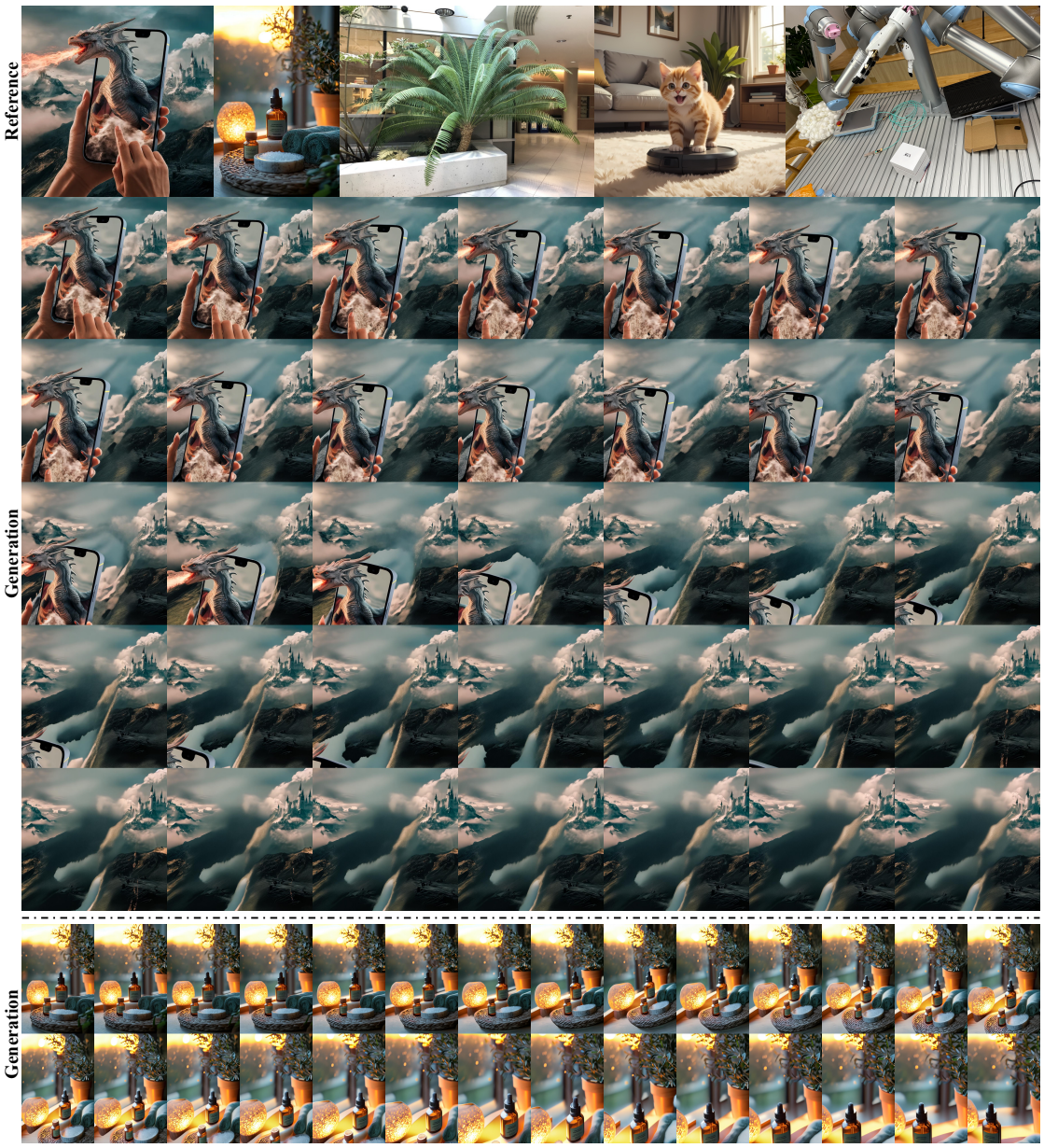


Figure 8. **Examples of Long-sequence Generation.** High-quality novel views generated along complex camera trajectories, maintaining spatial consistency and visual realism across extended sequences.

References

[1] Jonathan T. Barron, Ben Mildenhall, Dor Verbin, Pratul P. Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. CVPR, 2022. 7

[2] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Zip-nerf: Anti-aliased grid-based neural radiance fields. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 19697–19705, 2023. 9

[3] Jaeyoung Chung, Suyoung Lee, Hyeongjin Nam, Jaerin Lee, and Kyoung Mu Lee. Luciddreamer: Domain-free generation of 3d gaussian splatting scenes. arXiv preprint arXiv:2311.13384, 2023. 3, 7, 8

[4] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. Advances in Neural Information Processing Systems, 35:16344–16359, 2022. 6

[5] 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. 6

[6] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superpoint: Self-supervised interest point detection and description. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pages 224–236, 2018. 2

[7] Zhiwen Fan, Wenyan Cong, Kairun Wen, Kevin Wang, Jian Zhang, Xinghao Ding, Danfei Xu, Boris Ivanovic, Marco Pavone, Georgios Pavlakos, et al. Instantsplat: Unbounded sparse-view pose-free gaussian splatting in 40 seconds. arXiv preprint arXiv:2403.20309, 2, 2024. 4

[8] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, 24(6):381–395, 1981. 2

[9] Ruiqi Gao, Aleksander Holynski, Philipp Henzler, Arthur Brussee, Ricardo Martin-Brualla, Pratul Srinivasan, Jonathan T Barron, and Ben Poole. Cat3d: Create anything in 3d with multi-view diffusion models. arXiv preprint arXiv:2405.10314, 2024. 7, 9

[10] Liang Han, Junsheng Zhou, Yu-Shen Liu, and Zhizhong Han. Binocular-guided 3d gaussian splatting with view consistency for sparse view synthesis. arXiv preprint arXiv:2410.18822, 2024. 7, 9

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In Proceedings of the IEEE international conference on computer vision, pages 2961–2969, 2017. 1

[12] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598, 2022. 6

[13] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engin Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 406–413, 2014. 7

[14] Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. Cotracker: It is better to track together. In Proc. ECCV, 2024. 2

[15] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. ACM Trans. Graph., 42(4):139–1, 2023. 4

[16] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. ACM Transactions on Graphics (ToG), 36 (4):1–13, 2017. 7, 8

[17] Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 22160–22169, 2024. 6

[18] Yu-Lun Liu, Chen Gao, Andreas Meuleman, Hung-Yu Tseng, Ayush Saraf, Changil Kim, Yung-Yu Chuang, Johannes Kopf, and Jia-Bin Huang. Robust dynamic radiance fields. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 13–23, 2023. 1

[19] Ben Mildenhall, Pratul P Srinivasan, Rodrigo Ortiz-Cayon, Nima Khademi Kalantari, Ravi Ramamoorthi, Ren Ng, and Abhishek Kar. Local light field fusion: Practical view synthesis with prescriptive sampling guidelines. ACM Transactions on Graphics (ToG), 38(4):1–14, 2019. 7

[20] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 5480–5490, 2022. 7

[21] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. arXiv preprint arXiv:2307.01952, 2023. 5

[22] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pages 3505–3506, 2020. 6

[23] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In Proceedings of the IEEE/CVF international conference on computer vision, pages 10901–10911, 2021. 6, 7, 8

[24] Kyle Sargent, Zizhang Li, Tanmay Shah, Charles Herrmann, Hong-Xing Yu, Yunzhi Zhang, Eric Ryan Chan, Dmitry Lagun, Li Fei-Fei, Deqing Sun, et al. Zeronvs: Zero-shot 360-degree view synthesis from a single real image. arXiv preprint arXiv:2310.17994, 2023. 7, 8, 9

[25] Ruoxi Shi, Hansheng Chen, Zhuoyang Zhang, Minghua Liu, Chao Xu, Xinyue Wei, Linghao Chen, Chong Zeng, and Hao

Su. Zero123++: a single image to consistent multi-view diffusion base model. arXiv preprint arXiv:2310.15110, 2023. 5

[26] 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. 5, 6

[27] 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. 5

[28] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502, 2020. 6

[29] Kunpeng Song, Ligong Han, Bingchen Liu, Dimitris Metaxas, and Ahmed Elgammal. Diffusion guided domain adaptation of image generators. arXiv preprint arXiv:2212.04473, 2022. 5

[30] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16, pages 402–419. Springer, 2020. 1

[31] Zhouxia Wang, Ziyang Yuan, Xintao Wang, Yaowei Li, Tianshui Chen, Menghan Xia, Ping Luo, and Ying Shan. Motionctrl: A unified and flexible motion controller for video generation. In ACM SIGGRAPH 2024 Conference Papers, pages 1–11, 2024. 8

[32] Rundi Wu, Ben Mildenhall, Philipp Henzler, Keunhong Park, Ruiqi Gao, Daniel Watson, Pratul P Srinivasan, Dor Verbin, Jonathan T Barron, Ben Poole, et al. Reconfusion: 3d reconstruction with diffusion priors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 21551–21561, 2024. 7, 9

[33] Haofei Xu, Anpei Chen, Yuedong Chen, Christos Sakaridis, Yulun Zhang, Marc Pollefeys, Andreas Geiger, and Fisher Yu. Murf: Multi-baseline radiance fields. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 20041–20050, 2024. 7, 9

[34] Haofei Xu, Songyou Peng, Fangjinhua Wang, Hermann Blum, Daniel Barath, Andreas Geiger, and Marc Pollefeys. Depthsplat: Connecting gaussian splatting and depth. arXiv preprint arXiv:2410.13862, 2024. 4, 7, 9

[35] Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. arXiv preprint arXiv:2409.02048, 2024. 3, 7, 8

[36] Xianggang Yu, Mutian Xu, Yidan Zhang, Haolin Liu, Chongjie Ye, Yushuang Wu, Zizheng Yan, Chenming Zhu, Zhangyang Xiong, Tianyou Liang, et al. Mvimgnet: A large-scale dataset of multi-view images. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 9150–9161, 2023. 6, 8, 9

[37] 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. 4, 7

[38] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. ACM Trans. Graph. (Proc. SIGGRAPH), 37, 2018. 6, 7, 8

[39] Zehao Zhu, Zhiwen Fan, Yifan Jiang, and Zhangyang Wang. Fsgs: Real-time few-shot view synthesis using gaussian splatting. In European Conference on Computer Vision, pages 145–163. Springer, 2025. 7, 9