Beyond the Frame: Generating 360° Panoramic Videos from Perspective Videos

Rundong Luo[1] Matthew Wallingford[2] Ali Fahardi[2] Noah Snavely[1] Wei-Chiu Ma[1]

[1]Cornell University [2]University of Washington

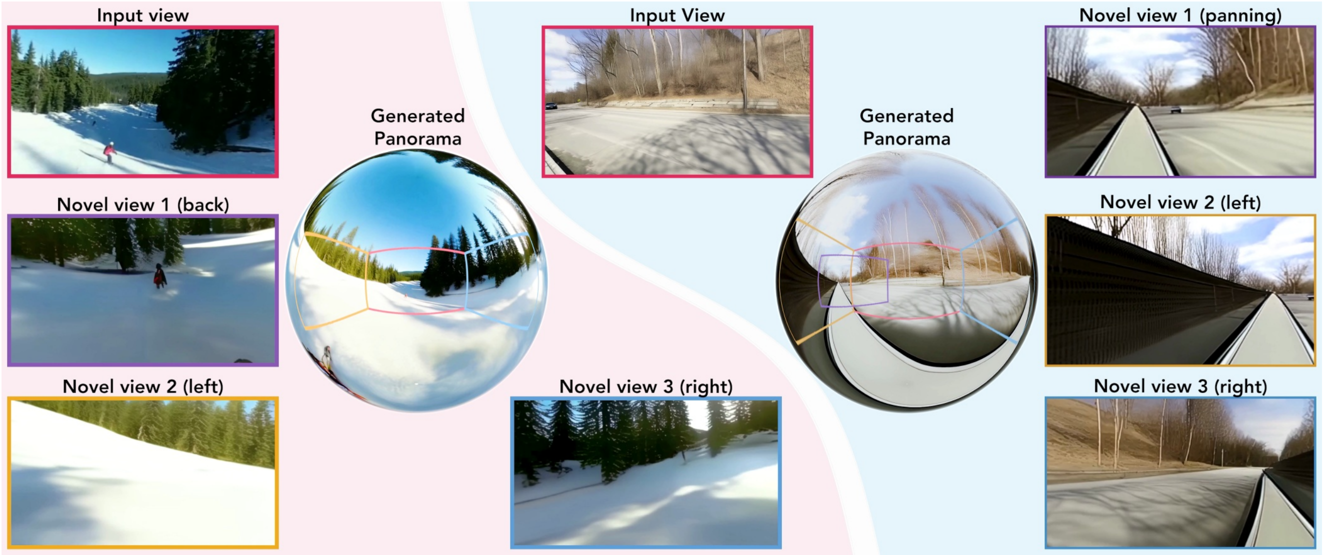


Figure 1. **360° videos generated by our model, Argus[†].** Starting from an input perspective video with arbitrary camera motion (red box), Argus generates a full 360° panoramic video (visualized as environmental maps), where the red box indicates the input view in the generated frame. The blue, orange, and purple boxes show sampled perspectives from the generated 360° video. *Best viewed in Adobe Acrobat Reader for the* ***embedded videos****.*

Abstract

360° videos have emerged as a promising medium to represent our dynamic visual world. Compared to the "tunnel vision" of standard cameras, their borderless field of view offers a more complete perspective of our surroundings. While existing video models excel at producing standard videos, their ability to generate full panoramic videos remains elusive. In this paper, we investigate the task of video-to-360° generation: given a perspective video as input, our goal is to generate a full panoramic video that is consistent with the original video. Unlike conventional video generation tasks, the output's field of view is significantly larger, and the model is required to have a deep understanding of both the spatial layout of the scene and the dynamics of objects to maintain spatio-temporal consistency. To address these challenges, we first leverage the abundant 360° videos available online and develop a high-quality data filtering pipeline to curate pairwise training data. We then carefully design a series of geometry- and motion-aware operations to facilitate the learning process and improve the quality of 360° video generation. Experimental results demonstrate that our model can generate realistic and coherent 360° videos from in-the-wild perspective video. In addition, we showcase its potential applications, including video stabilization, camera viewpoint control, and interactive visual question answering. View more high-resolution video results here[*].

1. Introduction

Remarkable advances in video generation have led to impressive capabilities, driven in part by large-scale video data from the web [5, 6, 22, 24, 39]. Models can now produce high-quality video clips based on an input image, allowing us to step into the world behind the pixels. While these models achieve impressive fidelity, they still provide us only

[†] Argus is named after a figure in Greek mythology with many eyes, symbolizing the ability to observe from multiple perspectives.

[*] This file contains embedded videos best viewed in Adobe Acrobat Reader. High-resolution results are available on our project page.

a narrow slice of the four-dimensional scene. Unlike the real world where we can freely look around and observe events as they unfold, current video models are restricted to a narrow, fixed perspective. Expanding video to the 360° medium, which more faithfully captures the visual world, enables better understanding of spatial layout and scene dynamics while improving spatio-temporal coherence. For example, standard video models commonly suffer from spatio-temporal inconsistency where content changes when looking back at previously observed parts of the scene. However, we find that generating 360° videos naturally resolves this problem as the entire scene is consistently visible.

To this end, we study the task of video-to-360° generation, aiming to generate a complete 360° video of a dynamic scene from a single-view perspective video. This task is difficult as it poses the following challenges: the input video only offers a narrow range of viewpoints, while the model must comprehend both the spatial layout of the scene and the dynamics of objects, then extrapolate to the entire scene. As illustrated in Figure 1, when the model observes a vehicle entering and then existing the frame (the red box), it must infer both the vehicle's previous and future trajectories and the progression of the surrounding scene. This prediction requires deep understanding of real-world constraints—for instance, that roads typically extend in a straight line, and vehicles maintain their lane at a constant pace.

One straightforward approach would be expanding the input video using existing video outpainting models [9, 12, 15, 49]. However, as we will show in Section 4, their generation quality degrades drastically as we extend further from the input viewpoint. This issue arises because current models are trained on videos with narrow field-of-view, which prevents them from learning complete scene dynamics.

To overcome these challenges, we leverage the relatively untapped data source of 360° videos. The growing popularity of 360° cameras has created a wealth of panoramic content spanning sports, travel, and everyday activities—providing valuable insights into how scenes and actions naturally unfold in our world. We formulate this task as a video outpainting problem from dynamic masks. Given a perspective video, our approach first estimates camera poses for each frame and projects them onto equirectangular maps within a shared coordinate system. We then condition a diffusion-based generation process on these maps and the input video. To facilitate model training, we propose three key techniques: camera motion simulation that models perspective video trajectories from 360° video, view-based frame alignment to ensure a fixed viewpoint in the generated panorama, and blended decoding to maintain boundary coherence. Our model, Argus, is the first to generate realistic and coherent 360° videos from standard perspective inputs.

Experimental results demonstrate that Argus outperforms existing methods in spatial coherence and visual quality. Our approach maintains consistency between the input and the generated content while producing realistic panoramic videos. The model generalizes effectively to various data sources, including online clips, self-recorded videos with complex dynamics, and model generated videos. Furthermore, Argus opens possibilities for several downstream applications, including video stabilization, camera viewpoint control, dynamic environmental mapping, and interactive visual question answering.

2. Related Works

Video Generation. Video generation aims to create high-quality, temporally consistent videos from multimodal inputs. Researchers have explored various architectures, including RNNs [2, 8, 13, 51], normalizing flows [4, 14], GANs [20, 30, 44, 45, 47], and transformers [17, 53, 54, 57]. However, these approaches suffer from resolution limitations and poor generalization, as they primarily train on small datasets designed for discriminative tasks. The recent success of diffusion models [23, 36] and access to larger, high-quality datasets have accelerated progress in video generation. While these approaches [5, 22, 24, 39, 58] produce remarkably realistic videos from text or image prompts, they remain constrained to narrow field-of-view outputs, preventing the generation of full 360° panoramic experiences.

Video Outpainting. While diffusion-based image outpainting from arbitrary mask regions has achieved satisfactory results by mask conditioning [36, 38] or inference process modifications [11, 31], video outpainting is limited to rectangular frame extensions [9, 12, 15, 49], constraining its application in panoramic content generation. Recently, VidPanos [32] introduced a method for synthesizing video panoramas from panning footage, but it focuses on dynamics within the observed regions and cannot extrapolate beyond initial viewpoints. Our approach overcomes these limitations by enabling flexible outpainting across dynamic, non-linear regions within a complete 360° panorama, generating immersive 360° scenes from single-view video inputs. This advancement expands video outpainting capabilities, enabling the generation of content that captures the full spatial and temporal dynamics of environments.

360° Panorama Generation. Generating 360° panoramic content presents unique challenges due to nonlinear distortions in equirectangular projections. These distortions warp objects and spatial layouts, complicating geometric appearance and creating boundary discontinuities. While current 360° image panorama generation methods [1, 26, 33, 43, 55, 60] produce satisfactory results, they struggle with video panoramas where temporal coherence and spatial consistency are crucial. For video panorama generation, Wang et al. [50] proposed a text-to-360° video generation framework, emphasizing text alignment rather than video-to-panorama transformation. Most relevant to our work is [42], where

Tan et al. independently developed a video-to-360° framework based on AnimateDiff [19]. However, their approach assumes pitch-only camera movements, uses limited training data, and confines evaluation to model-generated, subjectless or subject-centered scenes with minimal camera movement. We address these problems through geometry- and motion-aware modules and larger-scale training data. Our method generates realistic 360° panoramic videos from perspective inputs, outperforming existing approaches.

3. Video to 360°

Given a standard perspective video as input, our goal is to extrapolate beyond its limited field of view to generate a corresponding 360° panoramic video. The generated panorama must maintain both content consistency and temporal dynamics that align with the input frames.

Since the problem is heavily under-constrained, we propose to capitalize on a relatively untapped data source – 360° videos – to learn the priors. We start with the 360-1M dataset [48], which consist of approximately 1 million videos of varying quality, and systematically filter down to 283,863 video clips (see the supp. material for details). Then, we build upon a diffusion-based image-to-video architecture [5, 27, 36] and introduce a series of geometry- and motion-aware design tailored for video-to-360° generation (*e.g.*, camera motion simulation, view-based frame alignment, etc). As we will show in Section 4, these modifications are crucial for generating realistic panoramic videos.

3.1. Video-Conditioned 360° Diffusion

Our goal is to learn a diffusion mapping between an input perspective video $X_{\text{pers}} \in \mathbb{R}^{T\times 3\times H\times W}$ and an output 360° panoramic video $Y_{\text{equi}} \in \mathbb{R}^{T\times 3\times H'\times W'}$. We represent 360° video frames as equirectangular images and denote the number of frames by T. Following Latent Diffusion Models [5, 27, 36], our model consists of an encoder $\mathcal{E}$, a decoder $\mathcal{D}$, an image feature extractor $\mathcal{F}$, and a denoising U-Net f_θ, with f_θ as the only learnable component. We adopt the temporal VAE from Stable Video Diffusion [36] as our encoder and decoder, while the feature extractor is CLIP [35].

Since diffusion models require the input and the output to have the same dimensionality, we first convert the input perspective video X_{pers} into an equirectangular format X_{equi}, matching the dimensions of the output Y_{equi}. The unmapped areas are set to black . Next, we encode both equirectangular videos, X_{equi} and Y_{equi}, to continuous latents, $\mathbf{x}_{\text{equi}} = \mathcal{E}(X_{\text{equi}})$ and $\mathbf{y}_{\text{equi}} = \mathcal{E}(Y_{\text{equi}})$. Finally, we add time-dependent noise to $\mathbf{y}_{\text{equi}}$ to produce $\mathbf{y}_{\text{equi},t}$, concatenate it with a noise-augmented [23] version of $\mathbf{x}_{\text{equi}}$, and feed this combination into the denoising network f_θ to estimate the injected noise. The network f_θ is conditioned on the timestamp t and the image feature sequence $\mathcal{F}(X_{\text{pers}})$ through cross-attention [36]. In practice, projecting from perspective to equirectangular format requires prior knowledge of the camera's field of view and poses. While this information is known during training (determined when extracting perspective frames from 360° videos), it is unknown during inference. In Section 3.3, we will describe how we address this challenge.

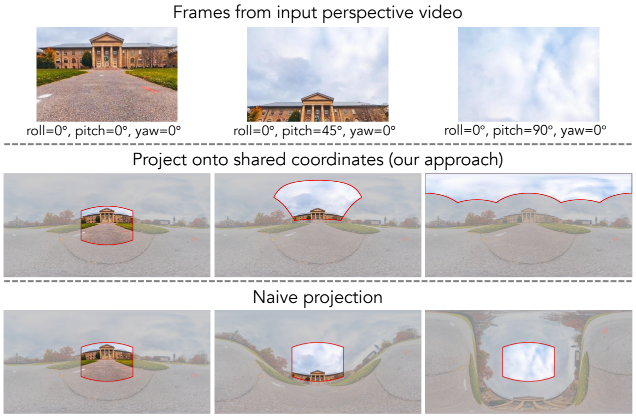

Frames from input perspective video
roll=0°, pitch=0°, yaw=0°
roll=0°, pitch=45°, yaw=0°
roll=0°, pitch=90°, yaw=0°
Project onto shared coordinates (our approach)
Naive projection

Figure 2. **View-based frame alignment.** Given input perspective video frames (first row), we project them onto shared coordinates to ensure a consistent viewing direction (second row). Without alignment, placing all video frames at the center (third row) forces the model to learn varying scene arrangements (*e.g.*, the sky appearing at different heights), complicating the learning process.

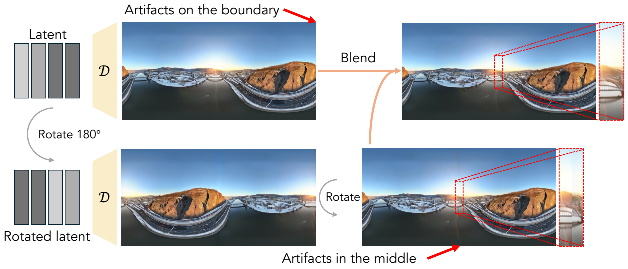


Figure 3. **Blended decoding.** We blend the video decoded from the original and 180°-rotated latents to ensure boundary consistency. Zoom in to see the artifacts on the bottom-right image.

3.2. Model Training

We train our denoising network f_θ with a score matching objective:

$$\min_{\theta} \mathbb{E}_{(\mathbf{x}_{\text{equi}},\mathbf{y}_{\text{equi}})\sim p_{\text{data}}(\mathcal{E}(X_{\text{equi}}),\mathcal{E}(Y_{\text{equi}})),t,\epsilon\sim\mathcal{N}(0,1)} \lambda(h)||\epsilon - f_\theta(\mathbf{y}_{\text{equi},t};t,\mathbf{x}_{\text{equi}},\mathcal{F}(X_{\text{pers}}))||_2^2. \quad (1)$$

Here, $\lambda(h) = (\frac{1}{2} - |\frac{1}{2} - h|)^2 + \delta$ is a re-weighting function that scales the loss of each pixel based on its height $h \in [0, 1]$ on the equirectangular map. Intuitively, it gives greater importance to regions near the equator (*i.e.*, h closer to $\frac{1}{2}$), as regions near the poles (*i.e.*, $h = 0$ or 1) are disproportionally enlarged in the equirectangular format. δ is a small offset to ensure that all regions contribute to the loss.

Method	Real camera trajectory						Simulated camera trajectory						Geometry
	PSNR↑	LPIPS↓	FVD↓	Imag.↑	Aes.↑	Motion↑	PSNR↑	LPIPS↓	FVD↓	Imag.↑	Aes.↑	Motion↑	Line cons.↑
PanoDiffusion [55]	16.44	0.4138	2649.0	**0.5055**	0.4486	0.9426	15.28	0.4469	2622.3	**0.4986**	0.4533	0.9384	0.6504
Argus (ours)	**21.83**	**0.2409**	**1228.6**	0.4939	**0.4828**	**0.9802**	**21.50**	**0.2602**	**1100.1**	0.4812	**0.4784**	**0.9805**	**0.8506**

Table 1. **Quantitative results for video-to-360° generation.** We finetune PanoDiffusion [55] on 360° video frames for fair comparison. *Imag.*, *Aes.*, and *Motion* stands for the Imaging Quality, Aesthetic Quality, and Motion Smoothness metrics from VBench [25]. *Line cons.* stands for our proposed *line consistency* metric. Simulated trajectories are generated by our camera motion simulation technique, and real-world trajectories are extracted from in-the-wild videos through calibration.

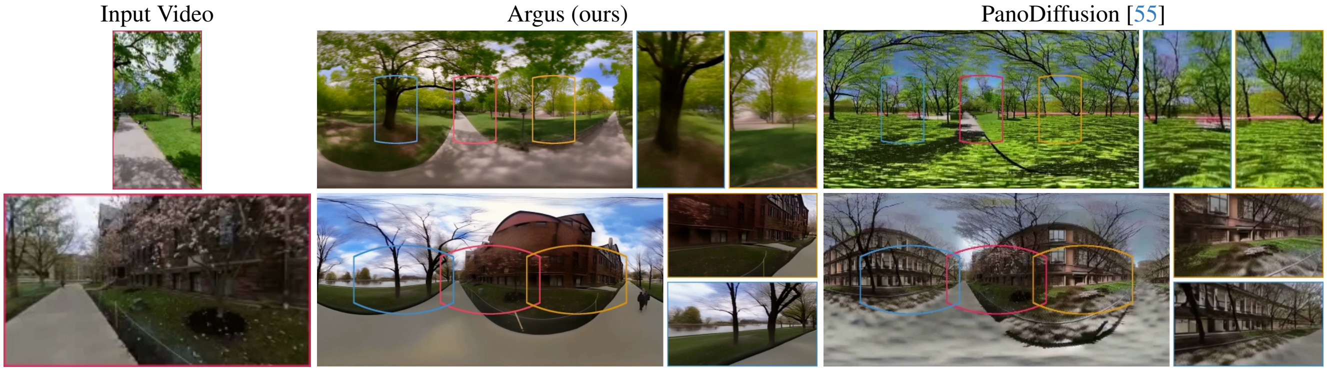


Figure 4. **Qualitative comparison with 360° image generation method PanoDiffusion (videos embedded).** The input region is highlighted in red, with orange and blue regions indicate extracted perspective views. Although PanoDiffusion can generate plausible 360° images from perspective inputs, the generated frames are temporally inconsistent.

We optimize our model using the EDM [27] diffusion framework, parameterizing the denoiser f_θ as:

$$f_\theta(\mathbf{y};\sigma) = c_{\text{skip}}(\sigma)\,\mathbf{y} + c_{\text{out}}(\sigma)\,F_\theta(c_{\text{in}}(\sigma)\,\mathbf{y}; c_{\text{noise}}(\sigma)), \quad (2)$$

where F_θ is the model to be trained, $\sigma = \sigma(t)$ indicates the noise schedule, and $c_{\text{in}}, c_{\text{out}}, c_{\text{skip}}, c_{\text{out}}$ are scaling functions. During training, the noise schedule σ is sampled from a log-Gaussian distribution. We refer readers to [27] for more details on the EDM framework.

Camera Movement Simulation. Our model aims to generate 360° videos from arbitrary perspective videos. However, naively sampling perspective views from 360° videos to train diffusion models would be ineffective due to the complex patterns of camera motion in real-world footage. We thus design a sampling strategy that allows us to approximate real-world camera motion and extract realistic training pairs of perspective and 360° videos.

Inspired by [18, 52], we introduce linear drift, oscillatory, and noise terms to mimic natural human motion [52]. Formally, camera movement is simulated as follows:

$$\begin{aligned}
\phi_{\text{roll}}(k) &= \mathcal{N}(0,\eta_r) + a_r \sin(\omega k + \tau_r), \\
\phi_{\text{pitch}}(k) &= \mathcal{N}(0,\eta_p) + a_p \sin(\omega k + \tau_p) + d_p k, \\
\phi_{\text{yaw}}(k) &= \mathcal{N}(0,\eta_y) + a_y \sin(\omega k + \tau_y) + d_y k + \phi_0,
\end{aligned} \quad (3)$$

where k is the frame index, ω is the oscillatory frequency, τ_r, τ_p, τ_y the initial phases, a_r, a_p, a_y the oscillatory amplitudes, η_r, η_p, η_y the noise strengths, d_p, d_y the drift rates, and ϕ_0 a random offset. The horizontal and vertical field of view are randomly chosen between $[30°, 120°]$. Additionally, since horizontal rotation preserves the 360° property, we augment the data with random circular shifts.

3.3. Model Inference

The above framework is sufficient for training our model on paired 360° and perspective videos. However, generating outputs, especially for in-the-wild videos, presents several challenges. First, as discussed in Section 3.1, projecting a perspective video into an equirectangular format typically requires knowledge of the camera's field of view and poses, yet in practice, the relative camera angles between frames are often unknown. Another challenge is the presence of boundary artifacts in equirectangular images: while the left and right edges are distant in image space, they are spatially adjacent in the scene. As a result, the model struggles to condition the right edge based on the left and vice versa, causing abrupt changes at the boundary.

View-Based Frame Alignment. To project the perspectives videos into equirectangular format, one straightforward solution is to always map perspectives frames to the center of equirectangular maps, as shown in Figure 2 (bottom row). While this approach sidesteps the need for camera pose estimation, it forces the diffusion model to implicitly learn the camera motion and handle complex distortions. For example, the model must detect when the camera is panning upward, as in Figure 2 (bottom row), and predict surrounding content

Method	FoV $= 60^\circ$			FoV $= 90^\circ$			FoV $= 120^\circ$		
	Imaging↑	Aesthetic↑	Motion↑	Imaging↑	Aesthetic↑	Motion↑	Imaging↑	Aesthetic↑	Motion↑
Be-Your-Outpainter [49]	0.4014	0.3461	0.9683	0.4469	0.4161	0.9649	0.4175	0.3951	0.9628
Follow-Your-Canvas [9]	0.4268	**0.4750**	0.9704	0.4267	0.4685	0.9679	0.4130	0.4513	0.9660
Argus (ours)	**0.4760**	0.4722	**0.9816**	**0.4773**	**0.4785**	**0.9796**	**0.4895**	**0.4796**	**0.9777**

Table 2. **Quantitative comparison with video outpainting methods**. *Imaging*, *Aesthetic*, and *Motion* stands for the Imaging Quality, Aesthetic Quality, and Motion Smoothness metrics from VBench [25].

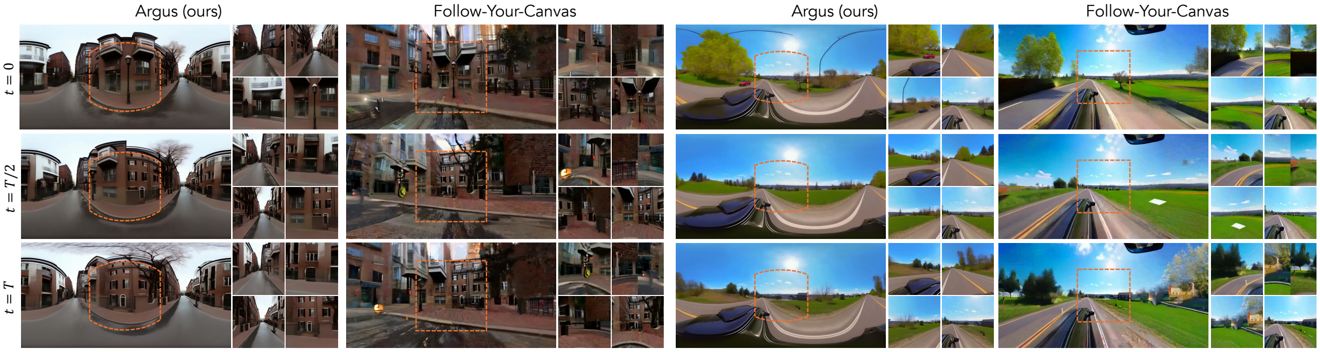


Figure 5. **Qualitative comparison with state-of-the-art video outpainting method.** The input region is highlighted in orange. For each generated 360° frame, four unwrapped perspective views are shown on the right. Video outpainting method struggles with satisfying 360° panoramic property and the generation quality declines as it extends further from the input viewpoint.

according to varying patterns of spherical distortion. Furthermore, the sky may appear in different locations within the 360° scene, further complicating the task. To address this challenge, we first estimate the relative camera poses of the input video using SLAM framework [28]. We then compute the Euler angles relative to the first frame and project them onto the equirectangular map. As shown in Figure 2 (middle row), this coordinate alignment ensures that each part of the equirectangular map corresponds to roughly the same scene region across frames, significantly improving consistency. For example, the sky appears consistently at the top, while the road remains at the bottom.

Blended Decoding. When generating 360° video frames, inconsistencies often emerge at the boundary where the left and right edges of the equirectangular image meet. To address this, we introduce blended decoding (Figure 3).

Previous techniques such as two-end alignment sampling [55] and circular padding [50] operate in the latent space, which cannot guarantee smooth boundary transitions after decoding, as the VAE is trained on standard perspective images or videos only. We propose blending in the pixel space instead. Specifically, we decode both the original latent and a 180°-rotated version, creating two outputs with identical content but differently positioned artifacts. We then compute a distance-based weighted average, assigning greater weight to pixels farther from the boundary:

$$Y_{k,i,j} = h_W(i)Y_{k,i,j} + (1 - h_W(i))Y'_{k,i,j}, \tag{4}$$

$$h_W(x) = 1 - 2\left|\frac{x}{W} - \frac{1}{2}\right|. \tag{5}$$

Here, i and j refer to the pixel coordinates. Y_k and Y'_k denote the equirectangular frames generated at 0° and 180° offsets for frame k. W represents the image width. This approach allows us to blend the two videos, effectively mitigating boundary artifacts. See Figure 3 for qualitative examples.

Long Video Generation. The method described above is limited to generating 360° panoramas from input perspective videos of exactly T frames. To accommodate longer input sequence, we extend our approach through context-aware training. Concretely, the model learns to predict the subsequent $T - S$ frames conditioned on S initial frames, which are fully observable in the conditioning equirectangular video. During training, we alternate between standard inputs (all T conditioning frames masked) and context-aware inputs (first S frames visible, remaining $T - S$ frames masked). For inference on extended sequences, we implement an iterative sampling process in which recent predictions serve as a context for subsequent iterations, allowing the generation of longer-length panoramic videos.

4. Experiments

In this section, we first present a quantitative evaluation of Argus, followed by qualitative examples of 360° generation from in-the-wild videos. Finally, we present a diverse set of downstream tasks that Argus can be applied to off-the-shelf.

4.1. Experimental Setup

Our model is initialized from the Stable Video Diffusion-I2V-XL model [5]. We train it in two phases: first at 384×768 resolution for 100K iterations, then finetuning on a high-quality

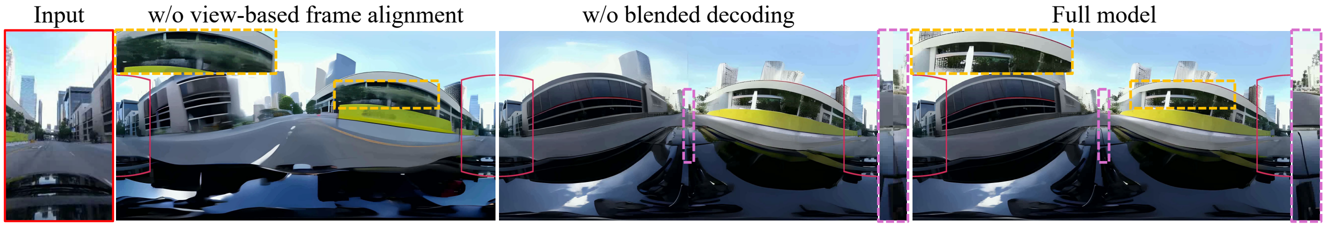


Figure 6. **Qualitative ablation studies.** The input region is marked in red. The 360° images are rotated 180° to illustrate the panoramic consistency. Compared to our full model, the variant without view-based frame alignment appears blurrier (orange box), while the variant without blended decoding shows artifacts in the center (pink box). Boxes are enlarged for ease of visualization.

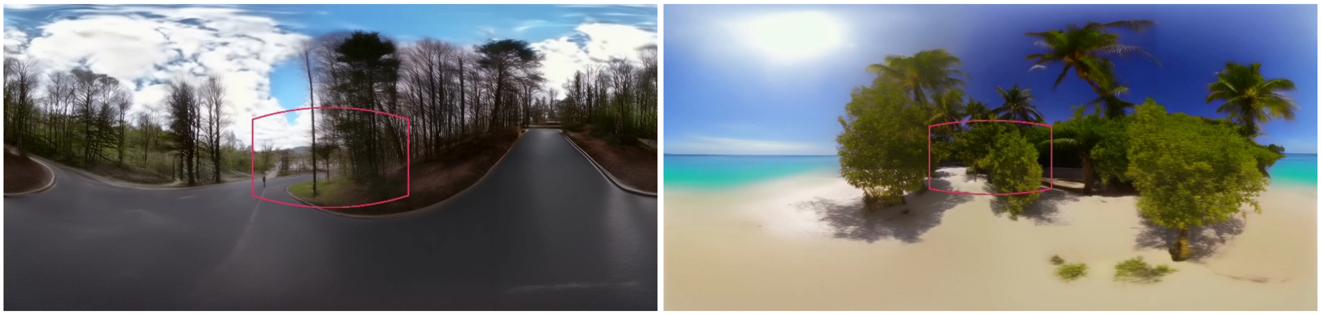


Figure 7. **Long-term 360° video generation in the wild.** The input video region is marked in red. Our generated results maintain semantic consistency across two rounds of generation. View the video results on our project page.

Variant	PSNR↑	LPIPS↓	FVD↓	Imaging↑	Aesthetic↑	Motion↑
w/o frame alignment	20.42	0.3194	1349.6	0.3816	0.4604	0.9783
w/o blended decoding	**22.09**	0.2675	**1226.3**	0.4574	0.4705	0.9795
Full model	21.83	**0.2409**	1228.6	**0.4939**	**0.4828**	**0.9802**
VAE Reconstruction	24.54	0.1663	121.8	0.5272	0.4929	0.9793

Table 3. **Ablation studies.** Our view-based frame alignment technique significantly improves overall performance, while blended decoding notably enhances boundary consistency despite its minimal effect on quantitative scores. Results of direct reconstruction using VAE are listed to represent the performance upper bound.

subset at 512×1024 resolution for additional 20K iterations, both with batch size 16. The finetuning phase adopts context-aware training and employs a noisier distribution to enhance training effectiveness at higher resolutions [10]. We set sequence length $T = 25$ and context length $S = 5$. We briefly describe our data, metrics, and baselines below, with complete details available in the supp. material.

Data. We evaluate our approach using a dataset of 101 360° videos, captured either with Insta360 cameras or from a hold-out set from YouTube. The 360°-perspective video pairs are created using two types of camera trajectories: (i) simulated trajectories generated by our camera motion simulation technique, and (ii) real-world trajectories extracted through calibration. Additionally, we collected 15 videos featuring linear structures, such as lanes and sidewalks, to evaluate geometric consistency in extrapolated views.

Metrics. We evaluate our results based on three key criteria: image quality, temporal coherency, and geometric consistency. For image quality, we use PSNR, LPIPS [61], Imaging Quality, and Aesthetic Quality metrics from VBench [25]. For temporal coherency, we employ FVD [46] and Motion Smoothness [25]. For geometric consistency, we introduce a *line consistency* metric to evaluate whether straight lines remain straight within extrapolated views. This metric is particularly important for assessing whether our model preserves fundamental geometric properties when generating novel views. To quantitatively measure this consistency, we follow [34] and use EA-score [62] to evaluate the angular and Euclidean distances between line pairs.

Baselines. Since no existing method is explicitly designed for the video-to-360° task, we adapt PanoDiffusion [55], a 360° image generation method, as our first baseline. Specifically, we re-trained their model on 360° video frames from our dataset without the depth branch. To improve consistency across frames, we applied identical initial noise across all frames during the sampling process [40]. We also compare Argus with video outpainting methods [9, 49]. Since these baselines support only rectangular input, we center square videos on the canvas and expand the vertical and horizontal field of view (FoV) to 180° and 360°, respectively. For evaluation, we extracted three perspective videos from each 360° test video, with FoVs of 60, 90, and 120 degrees.

4.2. Results and Analyses

Quantitative and Qualitative Results. We evaluate our model and baselines on our curated 360°-perspective video pairs. We use GT camera trajectories for all methods to isolate the impact of imperfect camera poses. As shown in

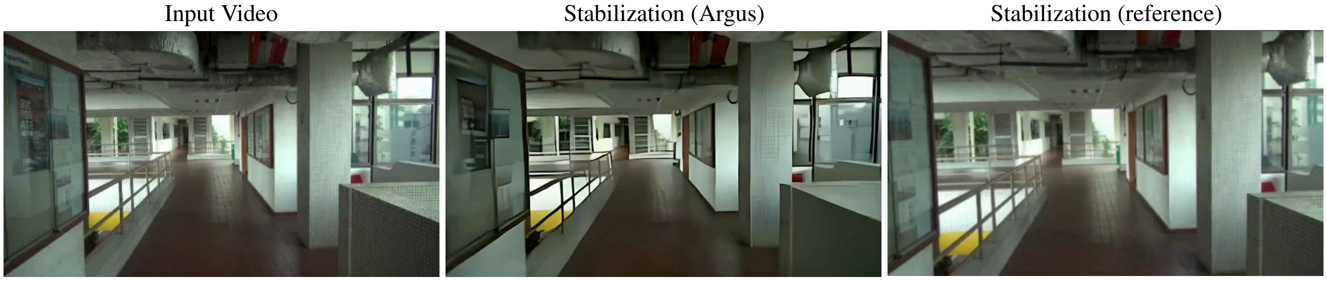


Figure 8. **Video stabilization results (videos embedded).** Columns from left to right: input frames, result from Argus, and reference result from [29]. Unlike cropping-based approaches, Argus maintains the full field of view due to its panoramic generation capability.

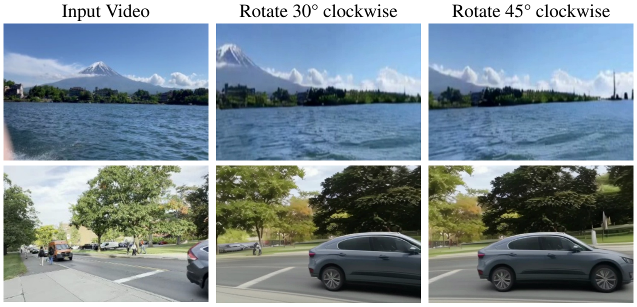


Figure 9. **Camera control in dynamic scenes (videos embedded).** Our model enables free camera rotation within dynamic scenes to capture elements beyond the initial viewpoint.

Table 1 and Figure 4, Argus significantly outperforms the adapted PanoDiffusion. While the adapted PanoDiffusion generates plausible individual 360° frames, it struggles with temporal consistency. Argus , in contrast, produces temporally smooth results, and is able to understand the geometric layout in the input and correctly extrapolate beyond. Comparing with video outpainting baselines, our method also achieves better visual quality and temporal coherency (see Table 2 and Figure 5). Video outpainting methods notably fail to preserve 360° panoramic properties, with generation quality deteriorating as the distance from the original viewpoint increases. In contrast, our model produces realistic panoramic videos throughout the entire field of view.

Ablation Studies. To verify the effectiveness of view-based frame alignment, we train a model in which perspective videos are always centered within the equirectangular map. During evaluation, we adjust the GT 360° videos accordingly. As shown in Table 3 and Figure 6, the absence of viewpoint alignment leads to degraded performance. This supports our hypothesis in Section 3.3 that without viewpoint alignment, the diffusion model must implicitly learn camera motion and manage complex distortions, making the task significantly more challenging. Table 3 also showcases the importance of blended decoding. For reference, we include results from direct reconstruction using the VAE, which represents the performance upper bound.

360° Video Generation In the Wild. Besides the curated 360°-perspective video pairs, we test our model on in-the-wild perspective videos featuring a diverse range of camera

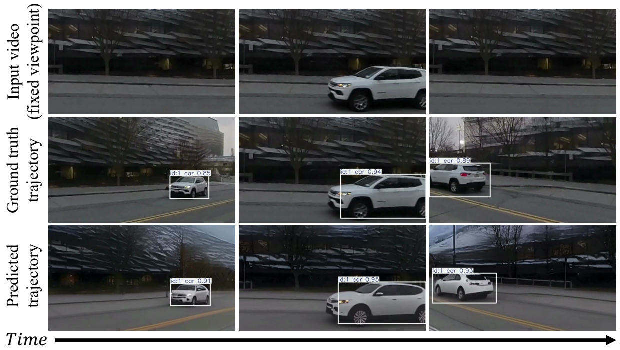
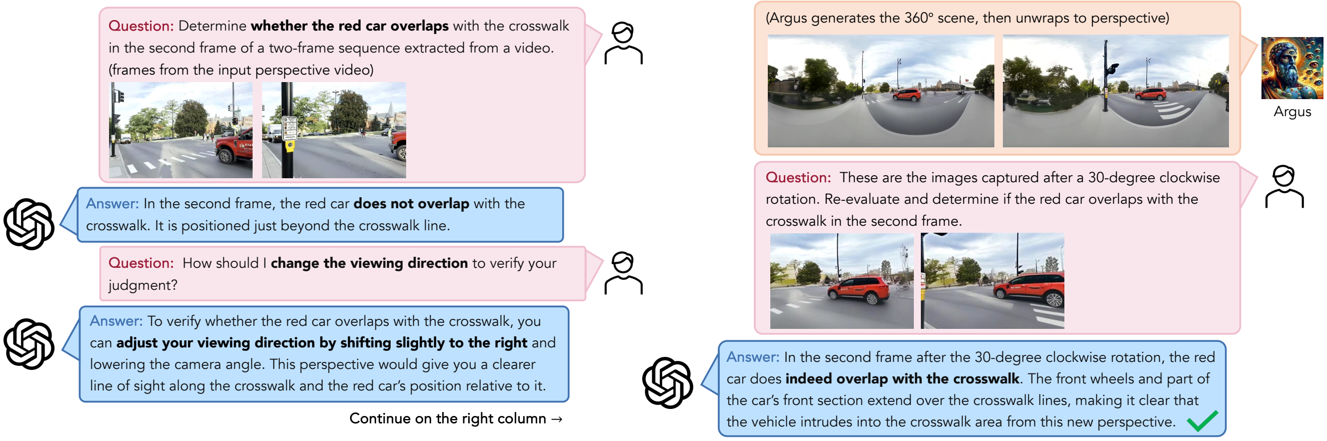


Figure 10. **Interpreting scene dynamics.** We capture a car driving scene with 360° camera and provide our model with a 60° FoV input of fixed viewing direction (top). The car's ground truth trajectory (middle) and Argus's predicted trajectory (bottom) shows strong alignment, demonstrating its ability to accurately predict object dynamics beyond the visible field of view.

motions and environments. We calibrate camera poses and employ iterative sampling for extended video generation. Our model is able to handle fixed orientation (Figure 7, left), mild motion (Figure 7, right), rapid motion (Figure 1), panning and vertical movement (project page), and even synthetic inputs from a text-to-video model (project page).

Interpreting Scene Dynamics. As we have alluded to in Figure 1, our model can understand the dynamics encoded in the input video (*e.g.*, the motion of the car) and extrapolate beyond. To better evaluate whether the generated dynamics are reasonable, we first capture a 360° video of a car driving by. We then crop a 60° horizontal FoV and input it into Argus. Finally, we apply tracking to both the generated 360° video and the original footage. As shown in Figure 10, the predicted trajectory closely aligns with the car's ground-truth motion. See our project page for more details.

Scene Generation Plausibility. Beyond line consistency, we evaluate the geometric plausibility of our generated 360° videos through 3D reconstruction. We unwrap panning perspective videos with yaw angles ranging from 45° to -45°, then calibrate using MegaSaM [28]. Testing on 48 smartphone videos, we compare predicted rotation angles with unwrapping angles and observe minimal average differences of $(\Delta\text{roll}, \Delta\text{pitch}, \Delta\text{yaw}) = (0.22°, 0.30°, 0.34°)$, confirming our generated content achieves high geometric realism.



Question: Determine **whether the red car overlaps** with the crosswalk in the second frame of a two-frame sequence extracted from a video. (frames from the input perspective video)

Answer: In the second frame, the red car **does not overlap** with the crosswalk. It is positioned just beyond the crosswalk line.

Question: How should I **change the viewing direction** to verify your judgment?

Answer: To verify whether the red car overlaps with the crosswalk, you can **adjust your viewing direction by shifting slightly to the right** and lowering the camera angle. This perspective would give you a clearer line of sight along the crosswalk and the red car's position relative to it.

Continue on the right column →

(Argus generates the 360° scene, then unwraps to perspective)

Question: These are the images captured after a 30-degree clockwise rotation. Re-evaluate and determine if the red car overlaps with the crosswalk in the second frame.

Answer: In the second frame after the 30-degree clockwise rotation, the red car does **indeed overlap with the crosswalk**. The front wheels and part of the car's front section extend over the crosswalk lines, making it clear that the vehicle intrudes into the crosswalk area from this new perspective. ✓

Figure 11. **Interactive visual question answering.** The first image sequence shows a red vehicle approaching a crosswalk, where the vision-language model (GPT-4o) fails to answer the question correctly because it lacks full scene comprehension. With Argus, we can freely rotate the camera, enabling better spatial understanding and accurately revealing the vehicle's overlap with the crosswalk.

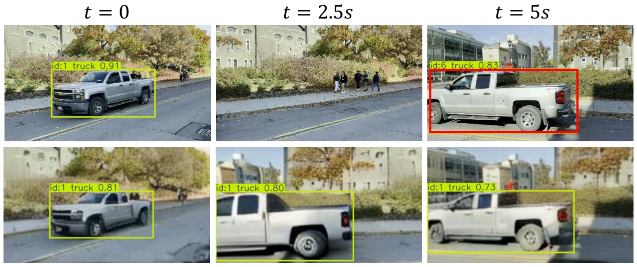


Figure 12. **Consistent object tracking.** Object detection results comparing input video (top) versus our unwrapped panorama (bottom). While the truck is identified as a separate entity when exiting and re-entering the input frame, it remains continuously visible in our generated panorama, resulting in consistent tracking.

4.3. Applications

This section showcases Argus's potential applications, including video stabilization, camera viewpoint control, dynamic environmental mapping, and interactive VQA.

Video Stabilization. Argus shows promising application to video stabilization without modifications. Traditional video stabilization techniques require cropping, resulting in a reduced field of view and visual information loss. In contrast, Argus enables video stabilization while maintaining a consistent field of view, as the generated panorama preserves scene information across frames. To achieve higher-resolution outputs, we crop regions with a smaller field of view from 360° videos and finetune on them. We test our approach using the video stabilization dataset from [29]. As shown in Figure 8, our method produces visually pleasing stabilization results while preserving a larger field of view than the reference results, effectively overcoming the limitations of cropping.

Camera Viewpoint Control. Argus enables viewpoint control in dynamic environments by unwrapping the generated 360° scene into perspective views. This capability allows exploration beyond the initial field of view (Figure 9) and facilitates tracking of fast-moving objects (Figure 12), enhancing immersion and supporting scene understanding tasks.

Dynamic Environmental Mapping. Argus enables realistic object relighting using the generated 360° panorama videos as dynamic environment maps. Figure 1 showcases metallic spheres rendered with these videos, exhibiting accurate reflections and lighting that validate practical applications.

Interactive VQA. Finally, we explore how the generated panorama video can help visual question answering in dynamic environments. Although generated videos might not provide a solid ground of facts, we show that by enabling free rotation of the camera, Argus allows for comprehensive spatial understanding by seeing the scene from multiple perspectives, based on the signals fully or partially available within the input perspectives. This flexibility supports interactive visual question answering, such as verifying if a vehicle overlaps with a crosswalk (Figure 11). This capability overcomes the limitation of fixed-viewpoint videos and enhances scene comprehension and opens new possibilities for video analysis applications.

5. Discussion

Limitations. Due to computational resource constraints, our current output resolution (512 × 1024) is lower than that of typical 4K real-world panoramas. The resolution further decreases when unwrapping back to perspective views. Additionally, while our model substantially improves upon the base SVD model in terms of object dynamics and temporal consistency (see supp. material for comparisons), it still exhibits shape inconsistencies and physics artifacts, similar to SVD and other SoTA video models such as COSMOS.

Conclusion. We present Argus, a video-to-360° generation model that creates full 360° panoramas from single-view perspective videos. Argus is trained on a relatively untapped data source, 360° videos. To enhance 360° video generation, we incorporate techniques such as camera movement simulation, blended decoding, and view-based frame alignment. Argus demonstrates strong performance across varied video sources, effectively capturing dynamic scenes with seamless spatial continuity. Our model offers promising potential for a broad range of downstream applications, marking a step forward in panoramic video generation.

Acknowledgment. The research is partially supported by a gift from Ai2, NVIDIA Academic Grant, and DARPA TIAMAT program No. HR00112490422. Its contents are solely the responsibility of the authors and do not necessarily represent the official views of DARPA.

References

[1] Naofumi Akimoto, Seito Kasai, Masaki Hayashi, and Yoshimitsu Aoki. 360-degree image completion by two-stage conditional gans. In *ICIP*, 2019. 2

[2] Mohammad Babaeizadeh, Chelsea Finn, Dumitru Erhan, Roy H Campbell, and Sergey Levine. Stochastic variational video prediction. In *ICLR*, 2018. 2

[3] Hana Bezalel, Dotan Ankri, Ruojin Cai, and Hadar Averbuch-Elor. Extreme rotation estimation in the wild. *arXiv:2411.07096*, 2024. 1

[4] Andreas Blattmann, Timo Milbich, Michael Dorkenwald, and Björn Ommer. ipoke: Poking a still image for controlled stochastic video synthesis. In *ICCV*, 2021. 2

[5] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv:2311.15127*, 2023. 1, 2, 3, 5

[6] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *CVPR*, 2023. 1

[7] Ruojin Cai, Bharath Hariharan, Noah Snavely, and Hadar Averbuch-Elor. Extreme rotation estimation using dense correlation volumes. In *CVPR*, 2021. 1

[8] Lluis Castrejon, Nicolas Ballas, and Aaron Courville. Improved conditional vrnns for video prediction. In *ICCV*, 2019. 2

[9] Qihua Chen, Yue Ma, Hongfa Wang, Junkun Yuan, Wenzhe Zhao, Qi Tian, Hongmei Wang, Shaobo Min, Qifeng Chen, and Wei Liu. Follow-your-canvas: Higher-resolution video outpainting with extensive content generation. *arXiv:2409.01055*, 2024. 2, 5, 6, 3

[10] Ting Chen. On the importance of noise scheduling for diffusion models. *arXiv:2301.10972*, 2023. 6, 2

[11] Ciprian Corneanu, Raghudeep Gadde, and Aleix M Martinez. Latentpaint: Image inpainting in latent space with diffusion models. In *WACV*, 2024. 2

[12] Loïc Dehan, Wiebe Van Ranst, Patrick Vandewalle, and Toon Goedemé. Complete and temporally consistent video outpainting. In *CVPR*, 2022. 2, 3

[13] Emily Denton and Rob Fergus. Stochastic video generation with a learned prior. In *ICML*, 2018. 2

[14] Michael Dorkenwald, Timo Milbich, Andreas Blattmann, Robin Rombach, Konstantinos G Derpanis, and Bjorn Ommer. Stochastic image-to-video synthesis using cinns. In *CVPR*, 2021. 2

[15] Fanda Fan, Chaoxu Guo, Litong Gong, Biao Wang, Tiezheng Ge, Yuning Jiang, Chunjie Luo, and Jianfeng Zhan. Hierarchical masked 3d diffusion model for video outpainting. In *ACM MM*, 2023. 2

[16] Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Image Analysis*, 2003. 1

[17] Songwei Ge, Thomas Hayes, Harry Yang, Xi Yin, Guan Pang, David Jacobs, Jia-Bin Huang, and Devi Parikh. Long video generation with time-agnostic vqgan and time-sensitive transformer. In *ECCV*, 2022. 2

[18] Matthias Grundmann, Vivek Kwatra, and Irfan Essa. Auto-directed video stabilization with robust l1 optimal camera paths. In *CVPR*, 2011. 4

[19] Yuwei Guo, Ceyuan Yang, Anyi Rao, Zhengyang Liang, Yaohui Wang, Yu Qiao, Maneesh Agrawala, Dahua Lin, and Bo Dai. Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. In *ICLR*, 2024. 3

[20] Sonam Gupta, Arti Keshari, and Sukhendu Das. Rv-gan: Recurrent gan for unconditional video generation. In *CVPR*, 2022. 2

[21] Jingwen He, Tianfan Xue, Dongyang Liu, Xinqi Lin, Peng Gao, Dahua Lin, Yu Qiao, Wanli Ouyang, and Ziwei Liu. Venhancer: Generative space-time enhancement for video generation. *arXiv:2407.07667*, 2024. 2

[22] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv:2210.02303*, 2022. 1, 2

[23] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *JMLR*, 2022. 2, 3

[24] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. In *NeurIPS*, 2022. 1, 2

[25] Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, et al. Vbench: Comprehensive benchmark suite for video generative models. In *CVPR*, 2024. 4, 5, 6, 2

[26] Nikolai Kalischek, Michael Oechsle, Fabian Manhardt, Philipp Henzler, Konrad Schindler, and Federico Tombari. Cubediff: Repurposing diffusion-based image models for panorama generation. In *ICLR*, 2025. 2

[27] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *NeurIPS*, 2022. 3, 4

[28] Zhengqi Li, Richard Tucker, Forrester Cole, Qianqian Wang, Linyi Jin, Vickie Ye, Angjoo Kanazawa, Aleksander Holynski, and Noah Snavely. Megasam: Accurate, fast, and robust structure and motion from casual dynamic videos. *arXiv:2412.04463*, 2024. 5, 7, 2

[29] Shuaicheng Liu, Lu Yuan, Ping Tan, and Jian Sun. Bundled camera paths for video stabilization. *ACM TOG*, 2013. 7, 8

[30] Pauline Luc, Aidan Clark, Sander Dieleman, Diego de Las Casas, Yotam Doron, Albin Cassirer, and Karen Simonyan. Transformation-based adversarial video prediction on large-scale data. *arXiv:2003.04035*, 2020. 2

[31] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *CVPR*, 2022. 2

[32] Jingwei Ma, Erika Lu, Roni Paiss, Shiran Zada, Aleksander Holynski, Tali Dekel, Brian Curless, Michael Rubinstein, and Forrester Cole. Vidpanos: Generative panoramic videos from casual panning videos. In *SIGGRAPH Asia*, 2024. 2

[33] Changgyoon Oh, Wonjune Cho, Yujeong Chae, Daehee Park, Lin Wang, and Kuk-Jin Yoon. Bips: Bi-modal indoor panorama synthesis via residual depth-aided adversarial learning. In *ECCV*, 2022. 2

[34] Shengyi Qian and David F Fouhey. Understanding 3d object interaction from a single image. In *CVPR*, 2023. 6, 2, 3

[35] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 3

[36] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 2, 3

[37] Runwayml. Stable-diffusion-inpainting, 2022. 3

[38] Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *SIGGRAPH*, 2022. 2

[39] 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. In *ICLR*, 2023. 1, 2

[40] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 6

[41] Tomás Soucek and Jakub Lokoc. Transnet v2: An effective deep network architecture for fast shot transition detection. In *ACM MM*, 2024. 1

[42] Jing Tan, Shuai Yang, Tong Wu, Jingwen He, Yuwei Guo, Ziwei Liu, and Dahua Lin. Imagine360: Immersive 360 video generation from perspective anchor. *arXiv:2412.03552*, 2024. 2

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

[44] Yu Tian, Jian Ren, Menglei Chai, Kyle Olszewski, Xi Peng, Dimitris N Metaxas, and Sergey Tulyakov. A good image generator is what you need for high-resolution video synthesis. In *ICLR*, 2021. 2

[45] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, and Jan Kautz. Mocogan: Decomposing motion and content for video generation. In *CVPR*, 2018. 2

[46] Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphaël Marinier, Marcin Michalski, and Sylvain Gelly. Fvd: A new metric for video generation. In *ICLR*, 2019. 6, 2

[47] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *NeurIPS*, 2016. 2

[48] Matthew Wallingford, Anand Bhattad, Aditya Kusupati, Vivek Ramanujan, Matt Deitke, Aniruddha Kembhavi, Roozbeh Mottaghi, Wei-Chiu Ma, and Ali Farhadi. From an image to a scene: Learning to imagine the world from a million 360° videos. In *NeurIPS*, 2024. 3, 1

[49] Fu-Yun Wang, Xiaoshi Wu, Zhaoyang Huang, Xiaoyu Shi, Dazhong Shen, Guanglu Song, Yu Liu, and Hongsheng Li. Be-your-outpainter: Mastering video outpainting through input-specific adaptation. In *ECCV*, 2024. 2, 5, 6, 3

[50] Qian Wang et al. 360dvd: Controllable panorama video generation with 360-degree video diffusion model. In *CVPR*, 2024. 2, 5

[51] Yunbo Wang, Mingsheng Long, Jianmin Wang, Zhifeng Gao, and Philip S Yu. Predrnn: Recurrent neural networks for predictive learning using spatiotemporal lstms. In *NeurIPS*, 2017. 2

[52] David A Winter. *Biomechanics and motor control of human movement*. John wiley & sons, 2009. 4

[53] Chenfei Wu, Lun Huang, Qianxi Zhang, Binyang Li, Lei Ji, Fan Yang, Guillermo Sapiro, and Nan Duan. Godiva: Generating open-domain videos from natural descriptions. *arXiv:2104.14806*, 2021. 2

[54] Chenfei Wu, Jian Liang, Lei Ji, Fan Yang, Yuejian Fang, Daxin Jiang, and Nan Duan. Nüwa: Visual synthesis pre-training for neural visual world creation. In *ECCV*, 2022. 2

[55] Tianhao Wu, Chuanxia Zheng, and Tat-Jen Cham. Panodiffusion: 360-degree panorama outpainting via diffusion. In *ICLR*, 2023. 2, 4, 5, 6, 3

[56] Jianxiong Xiao, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Recognizing scene viewpoint using panoramic place representation. In *CVPR*, 2012. 1

[57] Wilson Yan, Yunzhi Zhang, Pieter Abbeel, and Aravind Srinivas. Videogpt: Video generation using vq-vae and transformers. *arXiv:2104.10157*, 2021. 2

[58] Zhuoyi Yang, Jiayan Teng, Wendi Zheng, Ming Ding, Shiyu Huang, Jiazheng Xu, Yuanming Yang, Wenyi Hong, Xiaohan Zhang, Guanyu Feng, et al. Cogvideox: Text-to-video diffusion models with an expert transformer. *arXiv:2408.06072*, 2024. 2

[59] Maoyuan Ye, Jing Zhang, Shanshan Zhao, Juhua Liu, Bo Du, and Dacheng Tao. Dptext-detr: Towards better scene text detection with dynamic points in transformer. In *AAAI*, 2023. 1

[60] Xiaoding Yuan, Shitao Tang, Kejie Li, Alan Yuille, and Peng Wang. Camfreediff: Camera-free image to panorama generation with diffusion model. *arXiv:2407.07174*, 2024. 2

[61] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 6, 2

[62] Kai Zhao, Qi Han, Chang-Bin Zhang, Jun Xu, and Ming-Ming Cheng. Deep hough transform for semantic line detection. *TPAMI*, 2021. 6, 2, 3

Beyond the Frame: Generating 360° Panoramic Videos from Perspective Videos

Supplementary Material

1. Supplementary Material Overview

In this supplementary material, we provide additional dataset and implementation details. Accompanying this supplementary file is our project page.

2. Dataset Collection and Statistics

While 360° videos have been utilized on a small scale for various vision applications [3, 7, 56], their potential remains largely unexplored at greater magnitudes. In this section, we introduce a scalable data curation strategy for training a video-to-360° diffusion model. Then we show examples from our dataset and introduce its statistics to provide a rough understanding of our dataset.

2.1. Data Processing

We begin with the 360-1M dataset [48], which includes approximately 1 million 360° videos of varying quality. To establish a quality baseline, we retain only videos with more than 50 likes. Despite this initial filtering, the dataset still contains mislabeled 180° videos, standard perspective videos, static posters, static scenes, and unrealistic animations. To address this, we developed a scalable data processing pipeline:

1. **Format Filtering.** We sample frames from each video and detect horizontal lines in the center or vertical lines at the boundaries to verify the equirectangular format. Horizontal line detection removes up-down formatted 360° videos, while vertical line detection filters out perspective videos and posters.
2. **Intra-frame Filtering.** We compute LPIPS between the left and right halves to filter 180° videos and between the top and bottom halves to filter improperly formatted 360° videos.
3. **Inter-frame Filtering.** To ensure scene dynamics, we sample frames at random intervals and calculate the pixel variance. Static videos with minimal inter-frame variation are removed.

After coarse filtering, the videos are split into 10-second clips. We then apply fine-grained filtering using optical flow [16] to detect low-motion clips, TransNetv2 [41] to identify cuts, and DPText-DETR [59] to detect texts from unwrapped perspective views. Clips with excessive black pixels or low pixel variance are also excluded, as they indicate low visual complexity.

2.2. Dataset Statistics

The final dataset consists of 283,863 ten-second clips, distributed across 14 subject categories. The most prominent category, “Travel and Events,” accounts for 63,935 clips. From this dataset, we also build a high-quality selected after manual inspection of the video frames. This subset was used for high-quality fine-tuning. The distribution of categories in the dataset is shown in Figure 13, with examples of filtered and included clips in Figures 14 and 15.

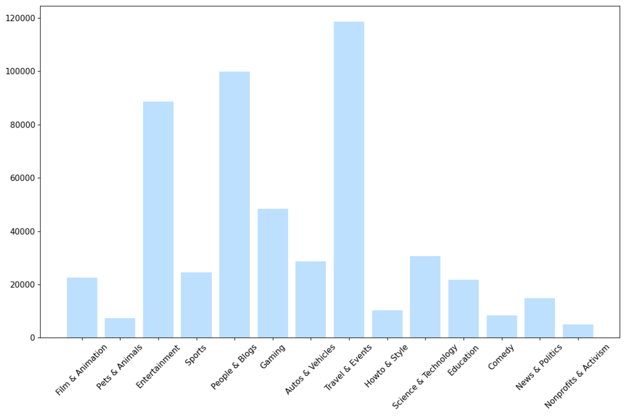


Figure 13. **Clip category distribution in our dataset.**

3. Implementation Details and Analyses

3.1. Perspective to Equirectangular Projection

We detail the mathematical process of mapping perspective video pixels to equirectangular maps. This includes equations for coordinate normalization, rotation, and spherical mapping.

To map a pixel coordinate (u, v) from an image with a given field of view, roll, pitch, and yaw to an equirectangular map, we first normalize the pixel coordinates to the normalized device coordinates (NDC). Assuming an image resolution of (W, H), the NDC coordinates (x_{ndc}, y_{ndc}) are given by

$$x_{ndc} = \frac{2u}{W} - 1, \quad y_{ndc} = \frac{2v}{H} - 1. \tag{6}$$

Given horizontal and vertical FOVs α and β, we compute a 3D direction vector (X, Y, Z) for the pixel in the camera's coordinate frame as follows:

$$X = x_{ndc} \cdot \tan\left(\frac{\alpha}{2}\right), \quad Y = y_{ndc} \cdot \tan\left(\frac{\beta}{2}\right), \quad Z = -1. \tag{7}$$

To reorient this vector from the camera frame to the equirectangular frame, we apply a series of rotations defined by the roll r, pitch p, and yaw y angles. Each angle defines a rotation matrix: R_r for roll,

$$R_r = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(r) & -\sin(r) \\ 0 & \sin(r) & \cos(r) \end{bmatrix}, \tag{8}$$

R_p for pitch,

$$R_p = \begin{bmatrix} \cos(p) & 0 & \sin(p) \\ 0 & 1 & 0 \\ -\sin(p) & 0 & \cos(p) \end{bmatrix}, \tag{9}$$

and R_y for yaw,

$$R_y = \begin{bmatrix} \cos(y) & -\sin(y) & 0 \\ \sin(y) & \cos(y) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{10}$$

The rotated vector (X', Y', Z') is obtained by applying these transformations in the order $R_y \cdot R_p \cdot R_r$:

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = R_y \cdot R_p \cdot R_r \cdot \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}. \tag{11}$$

We then convert (X', Y', Z') to spherical coordinates, where $\theta = \arctan 2(Y', X')$ and $\phi = \arcsin\left(\frac{Z'}{\sqrt{X'^2+Y'^2+Z'^2}}\right)$. Finally, the spherical coordinates are mapped to equirectangular pixel coordinates (u_{eq}, v_{eq}) for an equirectangular map of dimensions (W_{eq}, H_{eq}) by

$$u_{eq} = \frac{W_{eq}}{2\pi} \cdot (\theta + \pi), \quad v_{eq} = \frac{H_{eq}}{\pi} \cdot \left(\frac{\pi}{2} - \phi\right). \tag{12}$$

This yields the pixel location on the equirectangular map corresponding to the input pixel in the original image.

3.2. Training Details

Our model is initialized from the Stable Video Diffusion-I2V-XL model [5]. We implement a two-phase training strategy: initially at 384×768 resolution for 100K iterations, where we sample the noise scheduler parameter σ from a log-Gaussian distribution ($\log \sigma \sim \mathcal{N}(P_{\text{mean}}, P_{\text{std}}^2)$) and progressively increase the noise schedule from $(P_{\text{mean}}, P_{\text{std}}^2) = (-1, 1)$ to $(0, 1)$. In the second phase, we finetune the model at higher 512×1024 resolution on a high-quality subset for 20K iterations, employing context-aware training with a stronger noise schedule of $(P_{\text{mean}}, P_{\text{std}}) = (1, 1)$ as recommended by [10]. We set the sequence length $T = 25$ and context length $S = 5$. For both phases, we use the AdamW optimizer with a learning rate of 10^{-5} and a batch size of 16. The training required approximately six days on 16 A6000 GPUs for the first phase and four days on 8 A100 GPUs for the second phase.

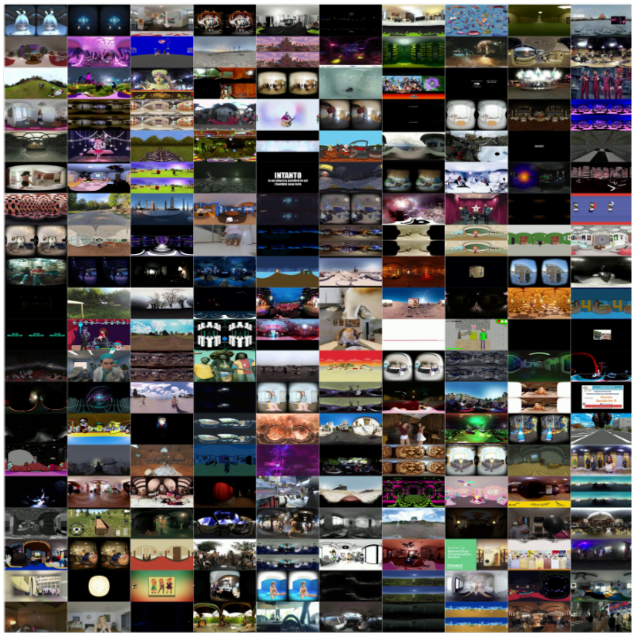


Figure 14. **Examples of videos discarded during data the data filtering pipeline.** We discard 180° videos, standard perspective videos, static posters, static scenes, and unrealistic animations from the initial noisy dataset.

3.3. Inference Details on In-the-Wild Videos

For in-the-wild input videos, we first employ MegaSaM [28] to estimate the camera intrinsics and poses, followed by generating the corresponding masked equirectangular video used to condition the network. After generation, we apply video super-resolution model [21] enhanced by our proposed blended decoding to increase the spatial resolution of the generated video by a factor of 2. Note that we do not apply super resolution modules in ablation studies and comparison with baseline methods.

3.4. Metrics

We evaluate our results based on three key criteria: image quality, temporal coherency, and geometric consistency. For image quality, we use PSNR, LPIPS [61], Imaging Quality, and Aesthetic Quality metrics from VBench [25]. For temporal coherency, we employ FVD [46] and the Motion Smoothness [25]. For geometric consistency, we introduce a *line consistency* metric to evaluate whether straight lines remain straight within extrapolated views. This metric is particularly important for assessing whether our model preserves fundamental geometric properties when generating novel views. To quantitatively measure this consistency, we follow [34] and use EA-score [62] to evaluate the angular and Euclidean distances between line pairs.

Specifically, FVD is calculated on the full 360° scene to evaluate overall distribution, while VBench metrics are

Figure 15. **Video frames sampled from our dataset.** We arrange the video frames to from a 360° image.

applied to four square 2D projections (front, back, left, right) extracted from the 360° video, as VBench is designed for perspective videos. PSNR and LPIPS are computed only within masked regions of visible directions and aggregated across frames, since other directions are extrapolated. Though this visible region remains under-constrained (visible areas at timestamp 0 may not appear at timestamp T), this approach provides more accurate evaluation than existing video outpainting methods [9, 12, 49] that calculate scores over the entire generated video.

Line Consistency. We introduce a line consistency metric to evaluate geometric fidelity across extrapolated viewpoints. This metric assesses whether straight lines in the original perspective remain consistent in neighboring views. Our approach uses real-world perspective videos that contain prominent linear structures, such as lanes and sidewalks.

Specifically, we first annotate lines in input views, then detect corresponding lines in neighboring views unwrapped from generated 360° videos using the Hough transform. Then, we compute the analytical solution of ground truth lines in neighboring views using homography and employ bipartite matching to pair these with detected lines. Finally, we follow [34] and report the EA-score [62], a score in $[0, 1]$ to measure the angle and euclidean distance between two lines, between the matched ground truth and detected lines. An example of our dataset and the line detection result in shown in Fig. 16.

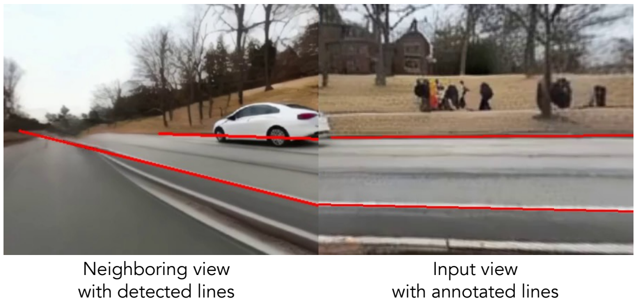


Figure 16. **Illustration of our line detection metric.** Given input view with annotated linear structures, we detect their extension in the neighboring views and measure their consistency.

3.5. Baseline Implementation Details

PanoDiffusion [55]. We reproduced this model due to the unavailability of their training code. We finetuned the image inpainting model [37] on the video frames of our dataset, omitting the depth branch due to the lack of depth information in the dataset. The model was trained for 50K iterations using the AdamW optimizer with a learning rate of 10^{-5} and a batch size of 128, running on 8 NVIDIA A6000 GPUs.

Be-Your-Outpainter [49] and Follow-Your-Canvas [9]. Video outpainting methods support only rectangular inputs, so we centered square videos on the canvas and expanded the vertical field of view to 180° and horizontal field of view 360°. For evaluation, we extracted three perspective videos from each 360° test video with FoVs of 60°, 90°, and 120°. Because these models require per-video optimization for each generation, they are very compute expensive, taking

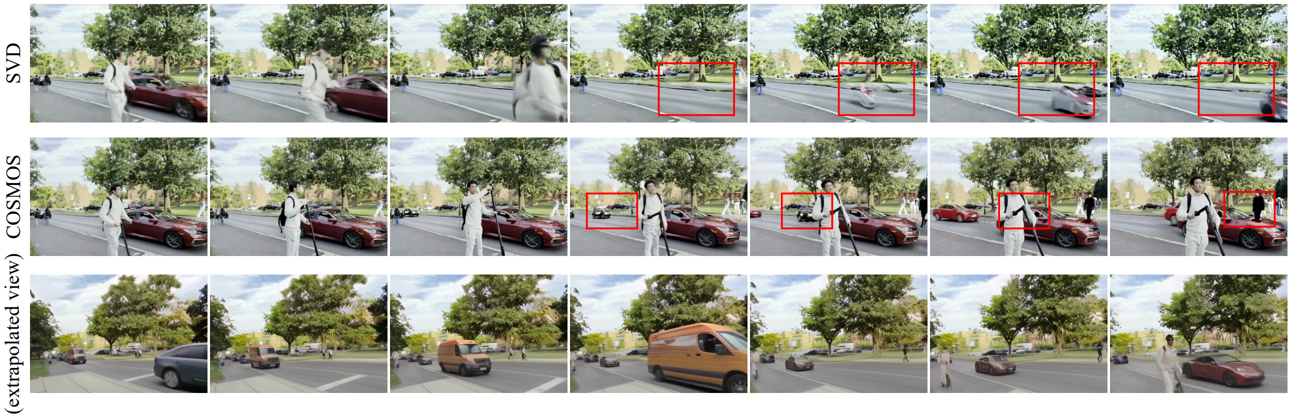


Figure 17. **Comparison with perspective video generation models.** Preserving shape consistency and dynamic plausibility remains an open challenge for video generation models. Specifically, our base model, SVD, exhibits noticeable appearance changes in the generated video (first row), while even state-of-the-art video models such as COSMOS demonstrate physical artifacts, where the black car on the back disappears (middle row).

about 14 and 11 minutes, respectively, on a single NVIDIA A6000 GPU for each generation. In contrast, our method does not introduce additional compute overhead upon SVD, taking around 90 seconds for each generation while achieving significantly better quality.

Limitations. Due to computational resource constraints, our current output resolution (512×1024) is lower than that of typical 4K real-world panoramas. The resolution further decreases when unwrapping back to perspective views. Additionally, while our model substantially improves upon the base SVD model in terms of object dynamics and temporal consistency, it still exhibits shape inconsistencies and physics artifacts, similar to SVD and other SoTA video models such as COSMOS, as shown in Figure 17.

4. Additional Qualitative Results

Additional comparison, application, and in-the-wild video generation results are available in our project page.