360Anything: Geometry-Free Lifting of Images and Videos to 360°

Ziyi Wu[1,3], Daniel Watson[1], Andrea Tagliasacchi[†2,3], David J. Fleet[1,3], Marcus A. Brubaker[1], and Saurabh Saxena[1]

[1] Google DeepMind [2] Simon Fraser University [3] University of Toronto

Abstract. Lifting perspective images and videos to 360° panoramas enables immersive 3D world generation. Existing approaches often rely on explicit geometric alignment between the perspective and the equirectangular projection (ERP) space. Yet, this requires known camera metadata, obscuring the application to in-the-wild data where such calibration is typically absent or noisy. We propose `360Anything`, a geometry-free framework built upon pre-trained diffusion transformers. By treating the perspective input and the panorama target simply as token sequences, `360Anything` learns the perspective-to-equirectangular mapping in a purely data-driven way, eliminating the need for camera information. Our approach achieves state-of-the-art performance on both image and video perspective-to-360° generation, outperforming prior works that use ground-truth camera information. We also trace the root cause of the seam artifacts at ERP boundaries to zero-padding in the VAE encoder, and introduce *Circular Latent Encoding* to facilitate seamless generation. Finally, we show competitive results in zero-shot camera FoV and orientation estimation benchmarks, demonstrating `360Anything`'s deep geometric understanding and broader utility in computer vision tasks. Additional results are available at https://360anything.github.io.

Keywords: Panorama generation · Diffusion transformer · Outpainting

1 Introduction

Generating photorealistic 3D worlds is an exciting and challenging frontier in generative modeling, offering transformative potential across robotics, AR/VR, and gaming. Recent years have witnessed significant advancements in this domain [36,66,75,81], largely propelled by the dramatic progress in visual generative models [5,34,60,67,79]. However, standard generators produce *perspective* views, capturing a narrow view of the physical world, and limiting their utility in creating fully immersive 3D worlds. This limitation has spurred significant interest in 360° generative models [71,82,84,100], especially those for lifting perspective imagery to omnidirectional 360° panoramas [14,29,46,74].

[†] Work done at Google DeepMind.

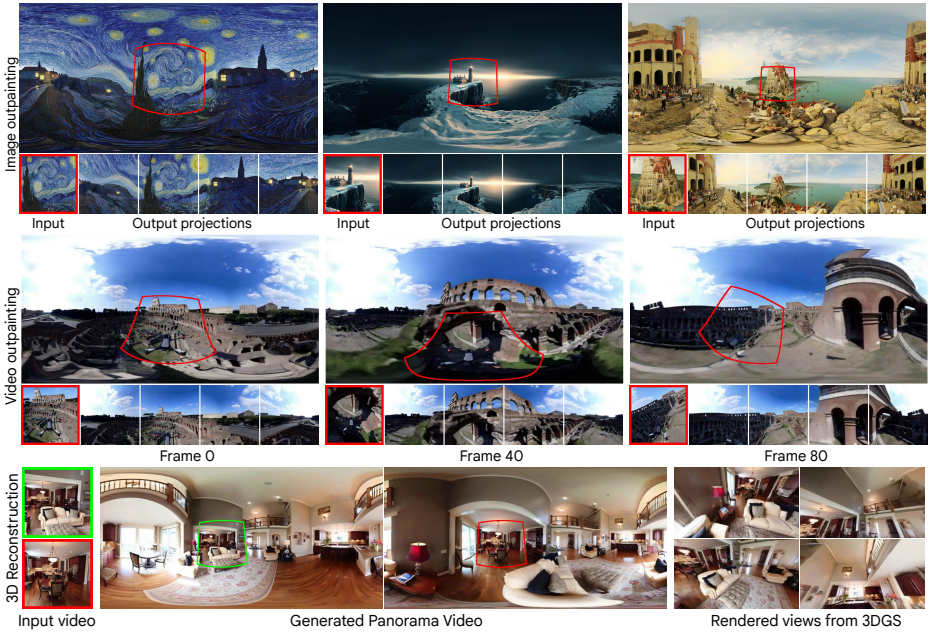


Fig. 1: `360Anything` lifts arbitrary perspective images (row 1) and videos (row 2) to seamless, gravity-aligned 360° panoramas. Model inputs and their projected regions are highlighted in **red** or **green**. Below each panorama, we show four perspective projections facing left, front, right, and back. Without using explicit camera information, `360Anything` handles images with varying Field-of-View and videos with large object and camera motion. The generated consistent panoramas enable 3D scene reconstruction via 3D Gaussian Splatting (row 3). Please see our project page for results in 360° viewers.

Despite recent progress, current approaches for perspective-to-panorama generation lack the robustness needed for "in-the-wild" inputs. To bridge the gap between perspective and panoramic spaces, prior works often rely on strong geometric inductive biases, such as *explicitly* projecting the perspective input to the target Equirectangular Projection (ERP) space to provide an aligned conditioning signal [46, 74, 89, 102]. However, this strategy requires known camera metadata, such as Field-of-View (FoV) and camera pose (yaw, pitch, roll) [46]. As a consequence, these models struggle with inputs where such metadata is absent, or are brittle when relying on noisy external estimators.

We posit that explicit geometric alignment is unnecessary for panorama generation. Instead, with sufficient data, a general-purpose architecture should be able to learn these relationships from data. To this end, our proposed framework utilizes a diffusion transformer (DiT) [54], and treats the perspective input and target panorama simply as token sequences. With attention on the concatenated sequence, the model learns their geometric relationship. This enables the model to effectively "place" the perspective input onto the 360° canvas and synthesize the remaining context, handling varying FoVs and camera poses as shown in Figure 1. Our pipeline thus eliminates the camera estimation step and makes the task fully end-to-end, which enjoys the benefit of scaling up model and data.

Beyond an end-to-end framework, we also address the common issue of seam artifacts at the boundary of a generated ERP image. Existing works rely on inference time tricks such as rotation augmentations to mitigate visible seams [46, 84, 89]. In contrast, we identify and eliminate the root cause of these artifacts during the training stage itself. Modern diffusion models often operate in the latent space of a convolution-based VAE [60], and VAEs utilize zero-padding in convolutional layers [27]. This introduces boundary artifacts in the latent representation of panorama data, which leads to seams in the generated panoramas. We propose a simple solution that uses *circular padding* when encoding VAE latents. This ensures that the latent representation has circular continuity, thereby eliminating the root cause of seam artifacts.

In summary, this work makes the following contributions:

1. We propose `360Anything`, a novel DiT-based architecture for "in-the-wild" perspective to canonical panorama generation that implicitly infers camera intrinsics and extrinsics, eliminating the need for camera calibration.
2. We identify VAE latent encoding as the root cause of seam artifacts in panorama generation and propose a simple remedy that mitigates the issue.
3. Despite not using camera metadata, `360Anything` achieves state-of-the-art performance for panoramic image and video generation, outperforming baselines that have access to extra camera information.
4. We evaluate the accuracy of our estimated FoV and camera poses, showing competitive results against supervised baselines. Furthermore, we can reconstruct consistent 3D scenes from our generated panoramic videos.

2 Related Work

Panorama Image Generation. Early approaches used GANs [10, 17, 19, 45, 48, 49, 69] or autoregressive generators [1, 8, 12, 104]. Recent methods have switched to diffusion models [22, 60] due to their state-of-the-art performance. One line of work generates panoramas from text prompts [14, 50, 71, 82]. They often design better panorama representations [6, 51] or panorama-aware operations [95, 99] to reuse knowledge in pre-trained perspective generators. Closer to `360Anything` are methods that outpaint panoramas from narrow field-of-view images [37, 42, 80, 94]. The majority of them project the conditioning perspective image to ERP space to be pixel-aligned with the target panorama, and then execute diffusion in ERP space [83, 89]. To handle the geometric properties of panoramas, they often inject strong inductive bias such as spherical convolutions [73, 88] and dual branch architectures [102]. A few works explore the cubemap representation to eliminate large distortions inherent in the ERP [25, 29]. However, existing methods either require known camera information to perform the projection, or assume the conditioning image has a fixed viewpoint and FoV. In contrast, our method treats the problem as a sequence-to-sequence task learned directly from data.

Panorama Video Generation. Some works directly fine-tune pre-trained video generators to produce panorama videos from text [43, 84, 90, 100]. This paper

instead tackles the task of panoramic outpainting from perspective videos [40, 47, 93]. Imagine360 [74] duplicates the denoising U-Net in AnimateDiff [18] to process panorama and perspective views separately, connected by spherical attention for information exchange. ViewPoint [13] proposes an improved cubemap representation to reduce the geometric distortion of ERP. Argus [46] further scales up training data to unconstrained YouTube videos [78]. Nevertheless, these methods (i) rely on *external tools* to estimate camera metadata of the conditioning video, and (ii) use *inference tricks* to eliminate seams in the generation. In contrast, `360Anything` learns to pose the input video in 4D space, and remove seams by identifying its root cause and adjusting the architecture.

Prior-Free Learning with Transformers. Recently, Transformers have dominated tasks that previously relied on inductive bias, including image generation [54, 92], editing [35, 87], and 3D understanding [61, 86]. While the majority of panorama generation methods still leverage a U-Net based architecture [29, 46], we identify this as a major limiting factor for the field. `360Anything` instead runs Transformers on a sequence of tokens without any geometric prior, while achieving state-of-the-art results across multiple tasks.

3 Method

Task formulation. Given a perspective video with T frames, $\mathrm{X}_{\text{pers}} \in \mathbb{R}^{T\times h\times w\times 3}$ (we treat image as a special case with $T{=}1$) and a caption $\boldsymbol{e}$, our goal is to outpaint a 360° panoramic video $\mathrm{Y}_{\text{equi}} \in \mathbb{R}^{T\times H\times W\times 3}$. In this work, we represent panorama data in the Equirectangular Projection (ERP) space.

Overview. Our method builds upon pre-trained latent diffusion transformers [54] (Sec. 3.1). We leverage a simple sequence concatenation approach to learn the perspective-to-equirectangular mapping and generate panoramas in a gravity-aligned canonical space (Sec. 3.2). Finally, we address the seam artifacts of generated panoramas by analyzing the panoramic latent space (Sec. 3.3). The overall architecture of `360Anything` is shown in Figure 2.

3.1 Background

We adopt the flow matching framework [41, 44], which learns a denoiser $\mathcal{G}_{\boldsymbol{\theta}}$ that maps from the standard normal distribution $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ to the distribution of panorama data $\mathrm{Y}_{\text{equi}} \sim p_{\text{data}}$. The forward diffusion process adds noise to clean data to obtain a noisy input $\mathrm{Y}^{\mathrm{t}}_{\text{equi}}$ at time $t \in [0, 1]$; i.e., $\mathrm{Y}^{\mathrm{t}}_{\text{equi}} = (1-t)\mathrm{Y}_{\text{equi}} + t\boldsymbol{\epsilon}$. The denoiser $\mathcal{G}_{\boldsymbol{\theta}}$, implemented as a neural network parameterized by $\boldsymbol{\theta}$, is trained to reverse this process with the following objective:

$$\min_{\boldsymbol{\theta}} \ \mathbb{E}_{t\sim p(t),\mathrm{Y}_{\text{equi}}\sim p_{\text{data}},\boldsymbol{\epsilon}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})} \|(\boldsymbol{\epsilon} - \mathrm{Y}_{\text{equi}}) - \mathcal{G}_{\boldsymbol{\theta}}(\mathrm{Y}^{\mathrm{t}}_{\text{equi}}, t, \boldsymbol{c})\|^2, \tag{1}$$

where $p(t)$ is the distribution of noise levels [11] and $\boldsymbol{c}$ refers to the auxiliary conditioning inputs, which in our case are the caption and the perspective input. We implement the denoiser $\mathcal{G}_{\boldsymbol{\theta}}$ as a diffusion transformer (DiT) [34, 54, 79].

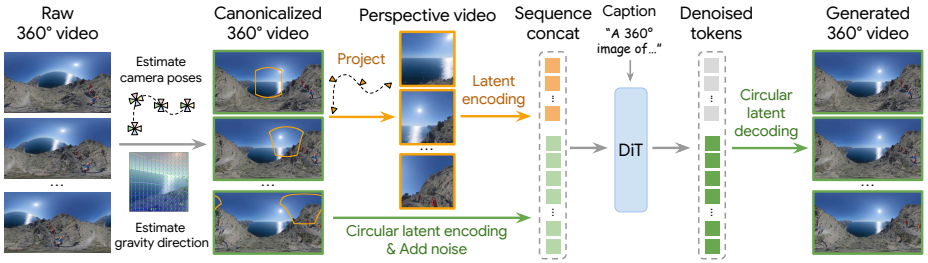


Fig. 2: 360Anything pipeline. Given a raw 360° training video with arbitrary camera orientations, we first estimate per-frame camera poses and rotate frames to align with the first frame. We then estimate the video's gravity direction and align it with the vertical axis. With such a *canonicalized* 360° video, we project it to a perspective video using randomly sampled camera intrinsics and poses (Sec. 4.2). We then encode both the conditioning and target videos to latent tokens. Critically, we employ *Circular Latent Encoding* for the target 360° video to avoid seam artifacts in the latent representation. The conditioning tokens (**orange**) and noisy target tokens (**green**) are concatenated along the *sequence dimension* and fed into a diffusion transformer (DiT). The denoised tokens can be decoded back to a 360° video via circular latent decoding.

To generate samples at a high resolution, modern diffusion models typically operate in the latent space of a pre-trained convolution-based VAE [60]. Following this practice, we encode panorama data $\mathrm{Y}_{\mathrm{equi}}$ to a latent representation $\mathrm{y}_{\mathrm{equi}}$ via an encoder $\mathcal{E}$, which can be decoded back to the pixel space with a decoder $\mathcal{D}$:

$$\mathrm{y}_{\mathrm{equi}} = \mathcal{E}(\mathrm{Y}_{\mathrm{equi}}), \quad \hat{\mathrm{Y}}_{\mathrm{equi}} = \mathcal{D}(\mathrm{y}_{\mathrm{equi}}). \tag{2}$$

The latent representation $\mathrm{y}_{\mathrm{equi}}$ is then patchified and flattened into a 1D sequence of tokens that is provided as input to the DiT.

3.2 Geometry-Free Scalable Panorama Generation

A core challenge in perspective-to-panorama generation lies in finding an effective solution for conditioning the model on the perspective input $\mathrm{X}_{\mathrm{pers}}$. Prior works [46, 74, 83, 89] typically project $\mathrm{X}_{\mathrm{pers}}$ into the ERP space to obtain $\mathrm{X}^{\mathrm{proj}}_{\mathrm{equi}}$, which is *pixel-aligned* with the generation target $\mathrm{Y}_{\mathrm{equi}}$. Then, they concatenate the latent of $\mathrm{X}^{\mathrm{proj}}_{\mathrm{equi}}$ and the noisy latent $\mathrm{y}^{\mathrm{t}}_{\mathrm{equi}}$ *channel-wise* as the input to the diffusion model. This approach imposes a strong geometric inductive bias by explicitly localizing the perspective input $\mathrm{X}_{\mathrm{pers}}$ on the panorama output $\mathrm{Y}_{\mathrm{equi}}$, drastically simplifying the task to image outpainting. However, pixel-aligned perspective-to-panorama projection requires precise camera Field-of-View (FoV) and orientation estimates [46]. For in-the-wild test data, this information is generally unavailable, and existing estimation methods can be noisy [39, 77, 85], resulting in accumulated errors and suboptimal performance. Consequently, when off-the-shelf camera estimators fail, channel-concatenation approaches break down completely due to the reliance on pixel-aligned conditioning; see Appendix B.1.

Sequence concatenation. We propose to relax this constraint by treating geometric alignment as a task we can learn from data. Instead of enforcing spatial

correspondence via projection into the ERP space, we employ a simple *sequence concatenation* mechanism inspired by recent image editing models [35, 87]. We directly encode the perspective input to latents: $x_{pers} = \mathcal{E}(X_{pers})$, and append it to the noisy latents y^t_{equi} as the DiT input: $\text{Concat}([x_{pers}, y^t_{equi}])$. The DiT thus runs global self-attention on the combined sequence of tokens. It learns to generate latents in the ERP image by reasoning their relationship to latents in the perspective image in a purely data-driven way.

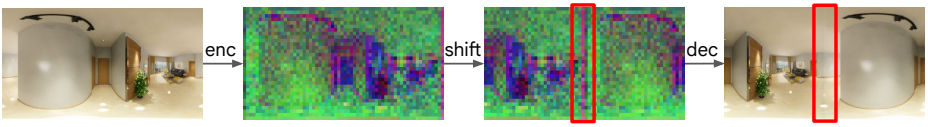
Generating canonical panoramas. Since we do not provide explicit camera pose to the model, the generated panorama Y_{equi} can be in any coordinate system. Prior works [13, 29] assume the conditioning view X_{pers} is always at the center of the ERP, hence generating panoramas with "unnatural" gravity directions (i.e., not pointing towards the bottom of the panorama). However, this requires the model to learn different spherical distortion patterns depending on the actual pose of the input X_{pers}. Our ablations show that this leads to degraded visual quality (see Tab. 7). Instead, we enforce a *Canonical Coordinate* constraint, for which the model is trained to generate panoramas in a standard, gravity-aligned upright orientation, regardless of the camera pose of the input X_{pers}. This requires the model to infer the camera pose of X_{pers} to "place" it on a canonical 360° canvas, and generate the rest of the panorama accordingly.

Implementing the canonical training objective requires ground-truth panoramas to be consistently aligned. This condition is naturally satisfied by our image datasets as they are predominantly synthetic renderings of 3D scenes [103]. Yet, our real-world video datasets [78] frequently exhibit arbitrary *non-canonical* orientations. Thus, we design a two-stage data pre-processing pipeline. We first apply COLMAP [64] to estimate per-frame camera pose, and rotate each frame to have zero rotation relative to the first frame. Then, we run GeoCalib [77] to estimate the global gravity direction of the stabilized video, and rotate the video to align the gravity direction with the vertical axis. This data pre-processing step ensures the model trains on consistent, gravity-aligned data, thereby generating canonical videos at test time. See Figure 8 in the Appendix for an example.

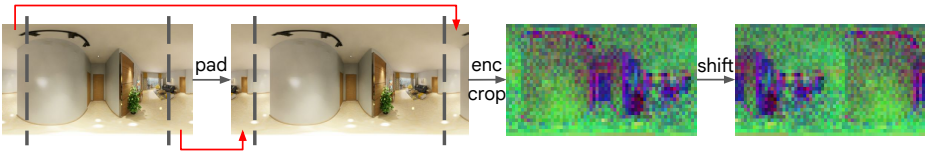
3.3 Seam-free Generation via Circular Latent Encoding

A common issue in panorama generation is "seam artifacts", where the left and right boundaries of the ERP image have visible discontinuities when concatenated (see Fig. 7). Prior works often attribute this to the generation process, employing *inference time* tricks such as rotated denoising (shifting the panorama cyclically across sampling steps) [89, 90] and circular padding in the VAE decoder [84].

We argue that the root cause of seams lies not in the inference stage, but in the *training stage*. Modern diffusion models are often applied in the latent space of a convolution-based VAE. When encoding a panorama image in the ERP format, the convolution layers perform zero-padding at the image boundaries, which introduces boundary artifacts in the feature maps [27]. As a result, even if the panorama image Y_{equi} is free from seams in pixel space, its latent representation y_{equi} contains a discontinuity (Fig. 3a). We posit that this discontinuity is the root cause of seam artifacts in the generated panorama.



(a) Naive VAE encoding of the panorama image leads to boundary discontinuities in the latent space.



(b) Our Circular Latent Encoding leads to seam-free latent representations.

Fig. 3: Illustration of Circular Latent Encoding. The top row (a) shows the seam artifact from naive VAE encoding. Shifting the encoded panorama latent by 180° shows a sharp discontinuity at the center, resulting in gray line-like artifacts when decoded back to image. The bottom row (b) illustrates our solution. Before encoding, we apply circular padding to the panorama image. After encoding, the latents in the padded regions are dropped. The shifted latent is now free from discontinuity, providing a seamless latent representation for diffusion training.

To eliminate the seam discontinuity in the encoded latent, we propose *Circular Latent Encoding*. Before encoding the latent of a panorama data, we crop w' (set as $W/8$ in our experiments) columns from its left and right regions, and pad them to the opposite side of the panorama to extend the boundary at both sides:

$$\mathrm{y}_{\mathrm{equi}}^{\mathrm{pad}} = \mathcal{E}(\mathrm{Concat}([\mathrm{Y}_{\mathrm{equi}}[-w':], \mathrm{Y}_{\mathrm{equi}}, \mathrm{Y}_{\mathrm{equi}}[:w']])). \tag{3}$$

After encoding, we drop the latent corresponding to the padded regions. This ensures the input sequence length to DiT is unchanged, thus introducing no overhead to training and inference. This simple technique produces a seam-free latent space (Fig. 3b), which serves as the correct target for model training.

4 Experiments

In this section, we conduct extensive experiments to answer the following questions: (i) How well does `360Anything` perform on perspective-to-360° image and video generation? (Sec. 4.1 and Sec. 4.2) (ii) How accurate are the camera FoV and orientation inferred by our model? (Sec. 4.3) (iii) What is the impact of each design choice in our framework? (Sec. 4.4)

4.1 Panoramic Image Outpainting

Implementation details. We fine-tune FLUX.1-dev [34], a state-of-the-art (SoTA) open weights text-to-image DiT. We use the Adam optimizer [31] with a learning rate of 5×10^{-5} and train with a batch size of 512 for 50k steps. At inference time, we use FLUX's default sampler [44] with 50 sampling steps and timestep shifting of 3.16. See Appendix A.2 for more implementation details.

Table 1: Quantitative results of perspective-to-360° image generation on Laval Indoor and SUN360 datasets. We borrow baseline results from CubeDiff [29] and report CubeDiff results under the single text description setting for a fair comparison. `360Anything` achieves a clear improvement across all metrics, only marginally lagging behind CubeDiff in terms of CLIP-FID on Laval Indoor.

Method	Laval Indoor					SUN360				
	FID↓	KID ($\times 10^2$)↓	CLIP-FID↓	FAED↓	CS↑	FID↓	KID ($\times 10^2$)↓	CLIP-FID↓	FAED↓	CS↑
OmniDreamer [1]	71.0	5.17	23.9	19.2	-	92.3	8.89	51.7	30.4	-
PanoDiffusion [89]	58.6	4.08	26.6	106.8	-	52.9	3.51	28.9	98.0	-
Diffusion360 [14]	33.1	2.07	16.9	23.7	26.38	45.4	3.73	18.5	12.6	22.89
CubeDiff [29]	<u>9.5</u>	<u>0.32</u>	**3.2**	<u>18.4</u>	<u>27.02</u>	<u>25.5</u>	<u>1.33</u>	8.1	<u>7.6</u>	25.00
`360Anything` **(ours)**	**8.0**	**0.22**	<u>4.6</u>	**9.8**	**29.21**	**22.4**	**1.27**	**7.3**	**3.8**	**28.07**

Training data and augmentations. For a fair comparison against CubeDiff [29], the prior state-of-the-art, we train on the same datasets with captions from Gemini 2.5 Flash [16]. To handle input images with diverse camera setup at test time, we uniformly sample FoV in [30°, 120°], pitch in [−60°, 60°], roll in [−15°, 15°], and use them to crop the conditioning perspective images for training. Following prior works, we also perform horizontal roll augmentation on the panorama image. We train the model to generate ERP images at 1024×2048 resolution.

Evaluation data and metrics. We follow the evaluation protocol proposed in CubeDiff and report results on the Laval Indoor [15] and SUN360 [91] datasets. To measure visual quality, we report Fréchet Inception Distance (FID) [21], Kernel Inception Distance (KID) [2], FID on CLIP [58] features (CLIP-FID), and FID on features of an auto-encoder fine-tuned on panorama images (FAED) [99]. Following prior works, FID, KID, CLIP-FID are computed on perspective crops from the generated ERP image, while FAED is computed directly on the generated ERP image. We also report CLIP-score (CS) [20] for text alignment.

Quantitative results. Table 1 compares `360Anything` with several perspective-to-panorama image generation baselines. In terms of visual quality, `360Anything` substantially outperforms all baselines on both datasets across FID, KID, and FAED metrics. Although it marginally lags behind CubeDiff in CLIP-FID on Laval Indoor (4.6 vs. 3.2), it outperforms it on SUN360 (7.3 vs. 8.1) which has more complex scene layouts and textures. Notably, we achieve a significant improvement in FAED, reducing the error by nearly 50% compared to the state-of-the-art. FAED is the only metric evaluated on the entire panorama, and it shows that `360Anything` generates 360° images with clearly better quality and geometry. Finally, our method also achieves the best CLIP-score, demonstrating its superior capability in adhering to prompts.

Qualitative results. We only compare with CubeDiff since other baselines lag behind by a *large* margin in all metrics. As CubeDiff is not open-sourced, we compare with samples from their website. CubeDiff leverages a cubemap representation with six faces, each with 90° FoV. It treats the conditioning image as the front face and denoises the other five faces. As shown in Figure 4 left, it sometimes generates *visible seams* between faces. In addition, when the input perspective image has an FoV smaller than 90°, CubeDiff has to stretch the object

Fig. 4: Qualitative results of perspective-to-360° image generation. We show multiple perspective views projected from the panorama, where the image with the **green border** is the conditioning image. Due to the use of a cubemap representation, CubeDiff sometimes generates seams between faces (left). In addition, CubeDiff always assumes the input image has a 90° FoV; yet when the actual FoV is smaller, it has to stretch the objects at the image boundary. This leads to distorted object structure, e.g., the balloons (middle) and the mushroom (right). In contrast, `360Anything` estimates the correct camera FoV and orientation of the input as shown by the green box on the panorama image, and thus produces much less distorted objects. Please check out our project page to view the generated panorama images interactively.

at the boundary, leading to distorted air-balloons (middle) and mushrooms (right). In contrast, `360Anything` accurately infers the camera parameters to place the input image in the ERP space, and produces objects with the correct structure.

4.2 Panoramic Video Outpainting

Implementation details. We fine-tune Wan2.1-14B [79], a SoTA open source text-to-video DiT. We use the Adam optimizer with a learning rate of 1×10^{-5} and train with a batch size of 64 for 20k steps. For model inference, we run 50 sampling steps with Wan's default sampler and a timestep shifting of 3.0.

Training data and augmentations. We take the same training data from Argus [46], run our video canonicalization pipeline, and caption using Gemini 2.5 Flash. We follow Argus to simulate camera trajectories to crop conditioning perspective videos in training. However, we note that simulated camera movement lacks diversity. Thus, we also incorporate camera trajectories extracted from real-world videos [59], which improves generalization to in-the-wild videos (see Fig. 13 in the Appendix). We train on 512×1024 resolution videos with 81 frames.

Evaluation data and metrics. We adopt the 101 testing videos from Argus [46]. The conditioning perspective videos come from two types of camera trajectories,

Table 2: Quantitative results of perspective-to-360° video generation. We follow Argus [46] to evaluate on two sets of camera trajectories. *Imag.*, *Aes.*, and *Motion* stand for the Imaging Quality, Aesthetic Quality, and Motion Smoothness metrics from VBench [26]. Since the exact eval split used in Argus is unavailable, we reproduce the eval split based on communication with the author (* denotes results on it). We report *both* the original and our reproduced results for Argus, which closely match to validate our reproduced eval set. Our method outperforms all baselines across all metrics.

Method	Real camera trajectory						Simulated camera trajectory					
	PSNR↑	LPIPS↓	FVD↓	Imag.↑	Aes.↑	Motion↑	PSNR↑	LPIPS↓	FVD↓	Imag.↑	Aes.↑	Motion↑
Imagine360* [74]	21.00	0.2636	1398.9	0.4556	0.4658	0.9666	20.45	0.2719	1532.1	0.4485	0.4536	0.9725
Argus [46]	21.83	0.2409	1228.6	0.4939	0.4828	0.9802	21.50	0.2602	1100.1	0.4812	0.4784	0.9805
Argus* [46]	22.35	0.2310	1020.7	0.4971	0.4863	0.9728	21.10	0.2653	1127.2	0.4762	0.4682	0.9852
ViewPoint* [13]	23.25	0.1364	844.3	0.5293	0.5150	0.9881	22.77	0.1326	957.8	0.5105	0.5045	0.9827
360Anything (ours)	**25.75**	**0.0468**	**483.4**	**0.5515**	**0.5427**	**0.9885**	**23.64**	**0.0846**	**432.9**	**0.5489**	**0.5394**	**0.9891**

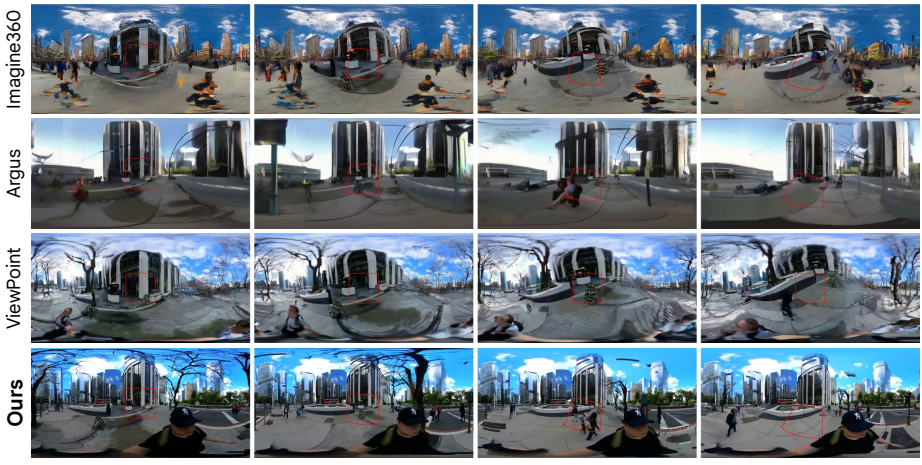


Fig. 5: Qualitative results of perspective-to-360° video generation. Regions corresponding to the input conditioning video are highlighted in **red**. Both Imagine360 and Argus exhibit low visual quality and distortions. ViewPoint always places the conditioning video at the center of the output, and thus generates a rotated image when the video contains large camera motion, leading to distortions (e.g., people and buildings). In contrast, `360Anything` generates stably *canonicalized* panorama videos, and accurately follows the text prompt to outpaint a person holding the camera. Please see our project page for better visual comparisons in the video format.

namely, simulated and extracted from real-world videos. To measure input preservation, we report PSNR and LPIPS [101] between ground-truth and generated panorama videos within regions covered by the perspective video. We also report FVD [76], Imaging Quality, Aesthetic Quality, and Motion Smoothness from VBench [26] to evaluate overall quality. Note that VBench metrics are computed on *perspective* crops (left, right, front, back) of the generated panorama videos.

Quantitative results. Table 2 compares `360Anything` with recent perspective-to-360° video generation methods. We outperform all baselines across all metrics on both subsets, often by a large margin. Surprisingly, while prior works construct

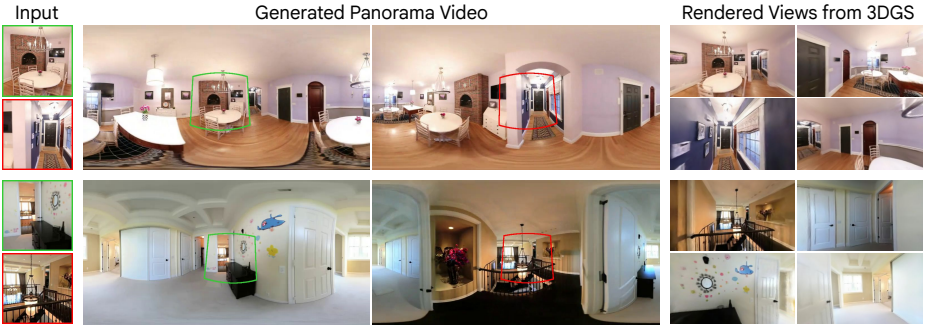


Fig. 6: Qualitative results of 3D scene reconstruction. Given an input monocular video (left), 360Anything outpaints the whole 360° viewpoint (middle), from which we can optimize a 3DGS (right). This allows fly-through exploration of the entire 3D scene. Please check out our project page to view 360° rendering of the 3DGS.

"pixel-aligned" input by unprojecting the conditioning perspective video, our method achieves better PSNR and LPIPS, meaning that it learns to better preserve the conditioning perspective video in the output. The significantly lower FVD indicates that our generated panorama videos exhibit more natural spherical distortion in the ERP format. In addition, 360Anything achieves higher VBench scores, demonstrating its superior visual and motion quality.

Qualitative results. As shown in Figure 5, 360Anything exhibits significantly higher visual quality than Imagine360 [74] and Argus [46]. Since ViewPoint [13] assumes the input video is always at the center of the ERP, it has to rotate the generated panorama given a tilted perspective video, leading to distortions (e.g., people and buildings). In contrast, our method generates canonicalized panorama frames that are *both* temporally consistent and distortion-free.

3D scene reconstruction. To demonstrate the 3D consistency of our generated videos, we distill the panoramic videos produced by 360Anything into a 3D Gaussian Splat (3DGS) [30]. We only test videos of static scenes such as indoor rooms [96, 105] as vanilla 3DGS cannot handle dynamic subjects. This process involves two main steps: (i) First, we employ rig-based COLMAP [63, 64] on the generated panoramic video to obtain camera poses. We project each video frame to cubemap faces and perform constrained bundle adjustment using a cubemap rig. (ii) Then, we train a vanilla 3DGS on the posed images. The scene reconstruction results are qualitatively shown in Figure 6. Given a monocular video with partial scene coverage, 360Anything outpaints the full 360° view that provides enough geometric cues for 3D reconstruction. This allows free exploration over the reconstructed scene, demonstrating its strong geometry consistency.

4.3 Single Image Camera Calibration

Experimental setup. To measure the camera parameters estimated by our model, we evaluate 360Anything (the image version) on several benchmarks on FoV and camera pose estimation. Given a testing image X_{pers}, we generate a

Table 3: Camera FoV estimation results (in degrees). We borrow baseline numbers from the MoGe paper [85]. Our zero-shot FoV estimation approach outperforms several supervised baselines, and lagging only slightly behind state-of-the-art methods DUSt3R and MoGe.

Method	NYUv2		ETH3D		iBims-1	
	Mean ↓	Med. ↓	Mean ↓	Med. ↓	Mean ↓	Med. ↓
Perspective [28]	5.38	4.39	13.6	11.9	10.6	9.30
WildCam [106]	3.82	3.20	7.70	5.81	9.48	9.08
LeReS [98]	19.4	19.6	8.26	7.19	18.4	17.5
UniDepth [56]	7.56	4.31	10.7	9.96	11.9	5.96
DUSt3R [86]	**2.57**	**1.86**	5.77	3.60	3.83	2.53
MoGe [85]	3.41	3.21	**2.50**	**1.54**	**2.81**	**1.89**
`360Anything`	3.90	3.17	5.68	4.22	5.21	4.04

Table 4: Camera pose estimation results (in degrees). Baseline numbers are from GeoCalib [77]. Our zero-shot approach outperforms most supervised baselines and lags behind current state-of-the-art, GeoCalib, by just ∼0.5 degrees.

Method	MegaDepth		LaMAR	
	Roll ↓	Pitch ↓	Roll ↓	Pitch ↓
MSCC [70]	0.90	5.73	1.44	3.02
ParamNet [28]	1.17	3.99	0.93	2.15
UVP [53]	0.51	4.59	0.38	1.34
GeoCalib [77]	**0.36**	**1.94**	**0.28**	**0.87**
`360Anything`	0.87	2.56	0.68	1.23

panorama image Y_{equi}, and infer the camera metadata via exhaustive search:

$$\min_{\text{fov,pitch,roll}} \|X_{pers} - \text{pano2pers}(Y_{equi}; \text{fov}, \text{pitch}, \text{roll})\|^2,$$

where pano2pers() is the panorama-to-perspective image projection function, and (fov, pitch, roll) are the estimated camera metadata.

Results on FoV estimation. We follow MoGe [85] and test our model on three real-world datasets: NYUv2 [68], ETH3D [65], and iBims-1 [33]. Table 3 compares `360Anything` with several baselines. It is worth mentioning that all baseline models are trained on large-scale datasets *specifically* for 3D understanding tasks, while `360Anything` is only trained for image-to-360° outpainting. In addition, over 90% of our training images are indoor scenes [103], thus creating a large *domain gap* with the outdoor ETH3D and iBims-1 datasets. Nevertheless, our zero-shot FoV estimation approach ranks among top-3 on most of the datasets. It achieves a low average estimation error of only 4.93 degrees, outperforming several supervised baselines while only lagging slightly (by $1-2$ degrees) behind recent methods DUSt3R [86] (4.06) and MoGe (2.91).

Results on camera orientation estimation. We follow GeoCalib [77] to test our model on MegaDepth [38] and LaMAR [62] datasets for estimating camera roll and pitch angles from a single image. As shown in Table 4, our zero-shot approach clearly outperforms most *supervised* baselines, only lagging behind the current state-of-the-art GeoCalib (∼0.5 degrees in both roll and pitch). These results demonstrate that our method learns to establish accurate correspondence between the conditioning perspective input and the generated panorama image.

4.4 Ablation Study

Circular latent encoding. Table 5 and Figure 7 compare different seam elimination techniques. We report the discontinuity score (DS) [9] to quantify seam artifacts. Our *circular latent encoding* (CLE) dramatically reduces DS

Table 5: Comparison of seam elimination techniques in perspective-to-360° image and video tasks. We report the discontinuity score (DS) to measure seams at the boundary of the panorama. *CLE* stands for our Circular Latent Encoding technique.

Method	Image			Video		
	Vanilla	Blended Decoding	**CLE (ours)**	Vanilla	Blended Decoding	**CLE (ours)**
DS↓	9.92	5.29	**3.87**	35.52	19.84	**13.28**

Fig. 7: Qualitative evaluation of various seam elimination techniques. For ease of visualization, we shift the generated panorama by 180° to show the concatenation of its left and right boundaries. Without any intervention (left), there are clear seams. Blended Decoding [46] (middle) "blurs" the seam to remove discontinuities; yet, it introduces gray line-like artifacts. Our technique (right) eliminates boundary artifacts entirely. We recommend zooming-in to evaluate these differences appropriately.

compared to the blended decoding approach proposed in Argus [46]. In addition, our method introduces no overhead to the generation process.

Camera augmentation. Training with randomly sampled camera FoVs and orientations enables `360Anything` to handle arbitrary perspective input at test time. However, the common evaluation protocol of perspective-to-360° image generation always uses conditioning images with 90° FoV and zero pitch and roll. We thus study whether disabling random camera sampling can further improve model performance. Surprisingly, Table 6 shows that such camera augmentation improves results on all metrics. We hypothesize that training with a wide distribution of camera setup forces the model to better understand perspective-equirectangular geometry, preventing overfitting to a single mapping.

Robustness to camera parameters. We condition `360Anything` on perspective images with different camera FoVs and orientations. As shown in Table 6, all metrics improve as FoV increases, as a larger FoV gives more information about the entire panorama image. Performance degrades slightly when changing pitch and roll angles, yet the degradations are all less than 1.0. We further compare with a variant of `360Anything` that uses channel-concatenation instead of sequence-concatenation for conditioning (dubbed *CC w/ GT Camera*). At test time, it uses *ground-truth* camera metadata to perform the perspective-to-equirectangular projection. Table 6 shows that it suffers from a similar performance drop as `360Anything`; yet our method does not rely on camera metadata. Overall, these results demonstrate the robustness of `360Anything`.

Table 6: Robustness to input view variations. We report the absolute FID/FAED for the standard view $(90°, 0°, 0°)$ and the relative change in metrics (Δ) for other input views. The last column shows the average degradation across all input view variations. *w/o Camera Aug.* stands for our method trained without camera augmentation, i.e., always train with $(90°, 0°, 0°)$ as the conditioning view. *CC w/ GT Camera* denotes the channel concatenation model that has access to ground-truth camera information. `360Anything` shows similar robustness to it even without camera metadata.

Metric	Method	Conditioning View (FoV, Pitch, Roll)							Avg.
		(90, 0, 0)	(30, 0, 0)	(60, 0, 0)	(90, 30, 0)	(90, -30, 0)	(90, 30, 5)	(90, -30, -5)	
FID ↓	w/o Camera Aug.	8.4	+2.4	+0.8	+8.9	+9.2	+8.9	+8.7	+6.48
	CC w/ GT Camera	7.7	+3.4	+0.8	**+0.5**	+0.9	**+0.5**	+0.9	+1.17
	`360Anything` **(ours)**	8.0	**+2.2**	+0.8	+0.6	+0.9	+0.6	**+0.8**	**+0.98**
FAED ↓	w/o Camera Aug.	10.4	+6.6	+1.7	+1.4	+1.8	+1.4	+1.6	+2.42
	CC w/ GT Camera	9.9	**+5.1**	+1.5	+0.4	+0.4	+0.3	**+0.2**	**+1.32**
	`360Anything` **(ours)**	9.8	+5.4	+1.5	**+0.3**	+0.4	+0.3	+0.4	+1.38

Table 7: Ablation on training video canonicalization. To save compute, models are trained at a lower resolution of 256×512. *Imag.*, *Aes.*, and *Motion* stand for Imaging Quality, Aesthetic Quality, and Motion Smoothness from VBench [26]. Training on canonical videos improves visual quality as indicated by FVD and VBench metrics.

Canonical	Real camera trajectory						Simulated camera trajectory					
	PSNR↑	LPIPS↓	FVD↓	Imag.↑	Aes.↑	Motion↑	PSNR↑	LPIPS↓	FVD↓	Imag.↑	Aes.↑	Motion↑
No	**26.56**	**0.0491**	559.5	0.4689	0.4939	0.9894	**24.66**	**0.0656**	527.0	0.4601	0.4917	0.9888
Yes (ours)	24.02	0.0521	**470.8**	**0.5437**	**0.5180**	**0.9899**	22.39	0.0880	**449.8**	**0.5387**	**0.5154**	**0.9903**

Training video canonicalization. We ablate the effect of training on canonicalized panorama videos in Table 7. When training on non-canonicalized videos, we always place the conditioning view at the center of the output image. This makes reconstructing the conditioning frame *much* easier, leading to better PSNR and LPIPS. However, it degrades the visual quality and fidelity significantly, as clearly indicated by FVD and VBench metrics. This is because the model has to generate panorama frames with varying gravity directions when the input video has non-zero roll and pitch angles, forcing it to learn different spherical distortion patterns. In contrast, we train `360Anything` to generate gravity-aligned upright panoramas, which greatly simplifies the generation task.

5 Conclusion

We present `360Anything`, a geometry-free framework for in-the-wild perspective-to-panorama generation. By shifting from explicit geometric unprojection to simple sequence concatenation within a DiT, we eliminate the dependency on camera information, allowing the model to learn geometric correspondence purely from data. Furthermore, we identify VAE encoder padding as the root cause of seam artifacts and introduce a novel and principled fix. Our approach not only achieves state-of-the-art results on panoramic image and video benchmarks, but also demonstrates robust zero-shot generalization to diverse, real-world media. We discuss the limitations and future directions in Appendix C.

References

1. Akimoto, N., Matsuo, Y., Aoki, Y.: Diverse plausible 360-degree image outpainting for efficient 3dcg background creation. In: CVPR (2022) 3, 8
2. Bińkowski, M., Sutherland, D.J., Arbel, M., Gretton, A.: Demystifying MMD GANs. In: ICLR (2018) 8, 23
3. Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorenz, D., Levi, Y., English, Z., Voleti, V., Letts, A., et al.: Stable video diffusion: Scaling latent video diffusion models to large datasets. arXiv preprint arXiv:2311.15127 (2023) 24, 25
4. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions. In: CVPR (2023) 22
5. Brooks, T., Peebles, B., Holmes, C., DePue, W., Guo, Y., Jing, L., Schnurr, D., Taylor, J., Luhman, T., Luhman, E., Ng, C., Wang, R., Ramesh, A.: Video generation models as world simulators. OpenAI technical reports (2024), https://openai.com/research/video-generation-models-as-world-simulators 1
6. Çapuk, H., Bond, A., Kızıl, M.B., Göçen, E., Erdem, E., Erdem, A.: TanDiT: Tangent-plane diffusion transformer for high-quality 360 {\deg} panorama generation. arXiv preprint arXiv:2506.21681 (2025) 3
7. Chen, B., Martí Monsó, D., Du, Y., Simchowitz, M., Tedrake, R., Sitzmann, V.: Diffusion Forcing: Next-token prediction meets full-sequence diffusion. NeurIPS (2024) 28
8. Chen, Z., Wang, G., Liu, Z.: Text2Light: Zero-shot text-driven hdr panorama generation. TOG (2022) 3
9. Christensen, A., Mojab, N., Patel, K., Ahuja, K., Akata, Z., Winther, O., Gonzalez-Franco, M., Colaco, A.: Geometry fidelity for spherical images. ECCV (2024) 12
10. Dastjerdi, M.R.K., Hold-Geoffroy, Y., Eisenmann, J., Khodadadeh, S., Lalonde, J.F.: Guided co-modulated gan for 360 field of view extrapolation. 3DV (2022) 3
11. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., et al.: Scaling rectified flow transformers for high-resolution image synthesis. In: ICML (2024) 4
12. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: CVPR (2021) 3
13. Fang, Z., Zhu, K., Liu, Z., Liu, Y., Zhai, W., Cao, Y., Zha, Z.J.: Panoramic video generation with pretrained diffusion models. NeurIPS (2025) 4, 6, 10, 11, 24
14. Feng, M., Liu, J., Cui, M., Xie, X.: Diffusion360: Seamless 360 degree panoramic image generation based on diffusion models. arXiv preprint arXiv:2311.13141 (2023) 1, 3, 8
15. Gardner, M.A., Sunkavalli, K., Yumer, E., Shen, X., Gambaretto, E., Gagné, C., Lalonde, J.F.: Learning to predict indoor illumination from a single image. arXiv preprint arXiv:1704.00090 (2017) 8, 23
16. Gemini Team Google: Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. arXiv preprint arXiv:2507.06261 (2025) 8, 21
17. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NeurIPS (2014) 3
18. Guo, Y., Yang, C., Rao, A., Wang, Y., Qiao, Y., Lin, D., Dai, B.: Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. In: ICLR (2024) 4, 24, 25

19. Hara, T., Harada, T.: Spherical image generation from a single normal field of view image by considering scene symmetry. In: AAAI (2021) 3
20. Hessel, J., Holtzman, A., Forbes, M., Bras, R.L., Choi, Y.: Clipscore: A reference-free evaluation metric for image captioning. In: EMNLP (2021) 8, 23
21. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. NeurIPS (2017) 8, 23
22. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: NeurIPS (2020) 3
23. Ho, J., Salimans, T.: Classifier-free diffusion guidance. arXiv preprint arXiv:2207.12598 (2022) 22
24. Huang, X., Li, Z., He, G., Zhou, M., Shechtman, E.: Self Forcing: Bridging the train-test gap in autoregressive video diffusion. NeurIPS (2025) 28
25. Huang, Y., Zhou, Y., Wang, J., Huang, K., Liu, X.: DreamCube: 3d panorama generation via multi-plane synchronization. In: ICCV (2025) 3
26. Huang, Z., He, Y., Yu, J., Zhang, F., Si, C., Jiang, Y., Zhang, Y., Wu, T., Jin, Q., Chanpaisit, N., et al.: VBench: Comprehensive benchmark suite for video generative models. In: CVPR (2024) 10, 14, 23
27. Islam, M.A., Jia, S., Bruce, N.D.: How much position information do convolutional neural networks encode? In: ICLR (2020) 3, 6
28. Jin, L., Zhang, J., Hold-Geoffroy, Y., Wang, O., Matzen, K., Sticha, M., Fouhey, D.F.: Perspective Fields for Single Image Camera Calibration. CVPR (2023) 12
29. Kalischek, N., Oechsle, M., Manhardt, F., Henzler, P., Schindler, K., Tombari, F.: CubeDiff: Repurposing diffusion-based image models for panorama generation. In: ICLR (2025) 1, 3, 4, 6, 8, 21
30. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM TOG (2023) 11, 26
31. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) 7, 22
32. Kocabas, M., Huang, C.H.P., Tesch, J., Müller, L., Hilliges, O., Black, M.J.: SPEC: Seeing people in the wild with an estimated camera. In: ICCV (2021) 21
33. Koch, T., Liebel, L., Körner, M., Fraundorfer, F.: Comparison of monocular depth estimation methods using geometrically relevant metrics on the ibims-1 dataset. CVIU (2020) 12
34. Labs, B.F.: Flux. https://github.com/black-forest-labs/flux (2024) 1, 4, 7, 21
35. Labs, B.F., Batifol, S., Blattmann, A., Boesel, F., Consul, S., Diagne, C., Dockhorn, T., English, J., English, Z., Esser, P., Kulal, S., Lacey, K., Levi, Y., Li, C., Lorenz, D., Müller, J., Podell, D., Rombach, R., Saini, H., Sauer, A., Smith, L.: FLUX.1 Kontext: Flow matching for in-context image generation and editing in latent space (2025) 4, 6, 22
36. Labs, W.: Marble. https://marble.worldlabs.ai/ (2025) 1
37. Li, J., Bansal, M.: PanoGen: Text-conditioned panoramic environment generation for vision-and-language navigation. NeurIPS (2023) 3
38. Li, Z., Snavely, N.: MegaDepth: Learning single-view depth prediction from internet photos. In: CVPR (2018) 12
39. Li, Z., Tucker, R., Cole, F., Wang, Q., Jin, L., Ye, V., Kanazawa, A., Holynski, A., Snavely, N.: MegaSaM: Accurate, fast and robust structure and motion from casual dynamic videos. In: CVPR (2025) 5, 25, 29

40. Liang, R., He, K., Gojcic, Z., Gilitschenski, I., Fidler, S., Vijaykumar, N., Wang, Z.: LuxDiT: Lighting estimation with video diffusion transformer. NeurIPS (2025) 4
41. Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. In: ICLR (2023) 4
42. Liu, A., Li, Z., Chen, Z., Li, N., Xu, Y., Plummer, B.A.: PanoFree: Tuning-free holistic multi-view image generation with cross-view self-guidance. In: ECCV (2024) 3
43. Liu, J., Lin, S., Li, Y., Yang, M.H.: DynamicScaler: Seamless and scalable video generation for panoramic scenes. In: CVPR (2025) 3
44. Liu, X., Gong, C., et al.: Flow straight and fast: Learning to generate and transfer data with rectified flow. In: ICLR (2023) 4, 7, 22
45. Lu, Z., Hu, K., Wang, C., Bai, L., Wang, Z.: Autoregressive omni-aware outpainting for open-vocabulary 360-degree image generation. In: AAAI (2024) 3
46. Luo, R., Wallingford, M., Farhadi, A., Snavely, N., Ma, W.C.: Beyond the Frame: Generating 360° panoramic videos from perspective videos. In: ICCV (2025) 1, 2, 3, 4, 5, 9, 10, 11, 13, 21, 24
47. Ma, J., Lu, E., Paiss, R., Zada, S., Holynski, A., Dekel, T., Curless, B., Rubinstein, M., Cole, F.: VidPanos: Generative panoramic videos from casual panning videos. In: SIGGRAPH Asia 2024 Conference Papers (2024) 4
48. May, C., Aliaga, D.: CubeGAN: omnidirectional image synthesis using generative adversarial networks. In: Computer Graphics Forum (2023) 3
49. May, C., Aliaga, D.: EpipolarGAN: Omnidirectional image synthesis with explicit camera control. In: ECCV (2024) 3
50. Ni, J., Zhang, C.B., Zhang, Q., Zhang, J.: What makes for text to 360-degree panorama generation with stable diffusion? In: ICCV (2025) 3
51. Park, M., Kang, T., Yun, J., Hwang, S., Choo, J.: SphereDiff: Tuning-free omnidirectional panoramic image and video generation via spherical latent representation. arXiv preprint arXiv:2504.14396 (2025) 3
52. Parmar, G., Zhang, R., Zhu, J.Y.: On aliased resizing and surprising subtleties in gan evaluation. In: CVPR (2022) 23
53. Pautrat, R., Liu, S., Hruby, P., Pollefeys, M., Barath, D.: Vanishing Point Estimation in Uncalibrated Images with Prior Gravity Direction. ICCV (2023) 12
54. Peebles, W., Xie, S.: Scalable diffusion models with transformers. ICCV (2023) 2, 4
55. Persson, E.: Texture from Humus. https://www.humus.name/index.php?page=Textures (accessed 09/2024) 21
56. Piccinelli, L., Yang, Y.H., Sakaridis, C., Segu, M., Li, S., Van Gool, L., Yu, F.: UniDepth: Universal monocular metric depth estimation. In: CVPR (2024) 12
57. polyhaven.com: HDRIs. https://polyhaven.com/hdris (accessed 09/2024) 21
58. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML (2021) 8, 23
59. Rockwell, C., Tung, J., Lin, T.Y., Liu, M.Y., Fouhey, D.F., Lin, C.H.: Dynamic camera poses and where to find them. In: CVPR (2025) 9, 21
60. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: CVPR (2022) 1, 3, 5
61. Rombach, R., Esser, P., Ommer, B.: Geometry-free view synthesis: Transformers and no 3d priors. In: ICCV (2021) 4

62. Sarlin, P.E., Dusmanu, M., Schönberger, J.L., Speciale, P., Gruber, L., Larsson, V., Miksik, O., Pollefeys, M.: LaMAR: Benchmarking localization and mapping for augmented reality. In: ECCV (2022) 12
63. Schönberger, J.L.: Colmap rig-based reconstruction. https://colmap.github.io/rigs.html, accessed: 2025-11-06 11
64. Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. In: CVPR (2016) 6, 11, 21
65. Schops, T., Sattler, T., Pollefeys, M.: Bad slam: Bundle adjusted direct rgb-d slam. In: CVPR (2019) 12
66. Schwarz, K., Rozumny, D., Rota Bulo, S., Porzi, L., Kontschieder, P.: A recipe for generating 3d worlds from a single image. In: ICCV (2025) 1
67. Sharma, A., Yu, A., Razavi, A., Toor, A., Pierson, A., Gupta, A., Waters, A., Tanis, D., Erhan, D., Lau, E., Shaw, E., Barth-Maron, G., Shaw, G., Zhang, H., Nandwani, H., Moraldo, H., Kim, H., Blok, I., Bauer, J., Donahue, J., Chung, J., Mathewson, K., David, K., Espeholt, L., van Zee, M., McGill, M., Narasimhan, M., Wang, M., Bińkowski, M., Babaeizadeh, M., Saffar, M.T., Pezzotti, N., Kindermans, P.J., Rane, P., Hornung, R., Riachi, R., Villegas, R., Qian, R., Dieleman, S., Zhang, S., Cabi, S., Luo, S., Fruchter, S., Nørly, S., Srinivasan, S., Pfaff, T., Hume, T., Verma, V., Hua, W., Zhu, W., Yan, X., Wang, X., Kim, Y., Du, Y., Chen, Y.: Veo (2024), https://deepmind.google/technologies/veo/ 1, 28
68. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: ECCV (2012) 12
69. Somanath, G., Kurz, D.: Hdr environment map estimation for real-time augmented reality. In: CVPR (2021) 3
70. Song, X., Kang, H., Moteki, A., Suzuki, G., Kobayashi, Y., Tan, Z.: MSCC: Multi-Scale Transformers for Camera Calibration. In: WACV (2024) 12
71. Stan, G.B.M., Wofk, D., Fox, S., Redden, A., Saxton, W., Yu, J., Aflalo, E., Tseng, S.Y., Nonato, F., Muller, M., et al.: LDM3D: Latent diffusion model for 3d. arXiv preprint arXiv:2305.10853 (2023) 1, 3
72. Su, J., Ahmed, M., Lu, Y., Pan, S., Bo, W., Liu, Y.: Roformer: Enhanced transformer with rotary position embedding. Neurocomputing (2024) 22
73. Sun, X., Ma, S., Li, S., Xu, M., Xia, J., Jiang, L., Deng, X., Wang, J.: Spherical-nested diffusion model for panoramic image outpainting. In: ICML (2025) 3
74. Tan, J., Yang, S., Wu, T., He, J., Guo, Y., Liu, Z., Lin, D.: Imagine360: Immersive 360 video generation from perspective anchor. NeurIPS (2025) 1, 2, 4, 5, 10, 11, 24
75. Team, H., Wang, Z., Liu, Y., Wu, J., Gu, Z., Wang, H., Zuo, X., Huang, T., Li, W., Zhang, S., et al.: HunyuanWorld 1.0: Generating immersive, explorable, and interactive 3d worlds from words or pixels. ArXiv:2507.21809 (2025) 1
76. Unterthiner, T., van Steenkiste, S., Kurach, K., Marinier, R., Michalski, M., Gelly, S.: Towards accurate generative models of video: A new metric & challenges. arXiv preprint arXiv:1812.01717 (2018) 10, 23
77. Veicht, A., Sarlin, P.E., Lindenberger, P., Pollefeys, M.: GeoCalib: Learning single-image calibration with geometric optimization. ECCV (2024) 5, 6, 12, 21
78. Wallingford, M., Bhattad, A., Kusupati, A., Ramanujan, V., Deitke, M., Kembhavi, A., Mottaghi, R., Ma, W.C., Farhadi, A.: From an image to a scene: Learning to imagine the world from a million 360 videos. NeurIPS (2024) 4, 6, 21
79. Wang, A., Ai, B., Wen, B., Mao, C., Xie, C.W., Chen, D., Yu, F., Zhao, H., Yang, J., Zeng, J., et al.: Wan: Open and advanced large-scale video generative models. arXiv preprint arXiv:2503.20314 (2025) 1, 4, 9, 22, 24, 25
80. Wang, C., Li, X., Qi, L., Lin, X., Bai, J., Zhou, Q., Tong, Y.: Conditional panoramic image generation via masked autoregressive modeling. NeurIPS (2025) 3

81. Wang, D., Jung, H., Monnier, T., Sohn, K., Zou, C., Xiang, X., Yeh, Y.Y., Liu, D., Huang, Z., Nguyen-Phuoc, T., Fan, Y., Oprea, S., Wang, Z., Shapovalov, R., Sarafianos, N., Groueix, T., Toisoul, A., Dhar, P., Chu, X., Chen, M., Park, G.Y., Gupta, M., Azziz, Y., Ranjan, R., Vedaldi, A.: WorldGen: From text to traversable and interactive 3d worlds (2025) 1
82. Wang, H., Xiang, X., Fan, Y., Xue, J.H.: Customizing 360-degree panoramas through text-to-image diffusion models. In: WACV (2024) 1, 3
83. Wang, J., Chen, Z., Ling, J., Xie, R., Song, L.: 360-degree panorama generation from few unregistered nfov images. In: ACM MM (2023) 3, 5
84. Wang, Q., Li, W., Mou, C., Cheng, X., Zhang, J.: 360DVD: Controllable panorama video generation with 360-degree video diffusion model. In: CVPR (2024) 1, 3, 6
85. Wang, R., Xu, S., Dai, C., Xiang, J., Deng, Y., Tong, X., Yang, J.: MoGe: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. In: CVPR (2025) 5, 12
86. Wang, S., Leroy, V., Cabon, Y., Chidlovskii, B., Revaud, J.: DUSt3R: Geometric 3d vision made easy. In: CVPR (2024) 4, 12
87. Wu, C., Li, J., Zhou, J., Lin, J., Gao, K., Yan, K., Yin, S.m., Bai, S., Xu, X., Chen, Y., et al.: Qwen-Image technical report. arXiv preprint arXiv:2508.02324 (2025) 4, 6, 22
88. Wu, T., Li, X., Qi, Z., Hu, D., Wang, X., Shan, Y., Li, X.: SphereDiffusion: Spherical geometry-aware distortion resilient diffusion model. In: AAAI (2024) 3
89. Wu, T., Zheng, C., Cham, T.J.: PanoDiffusion: 360-degree panorama outpainting via diffusion. In: ICLR (2024) 2, 3, 5, 6, 8
90. Xia, Y., Weng, S., Yang, S., Liu, J., Zhu, C., Teng, M., Jia, Z., Jiang, H., Shi, B.: PanoWan: Lifting diffusion video generation models to 360 {\deg} with latitude/longitude-aware mechanisms. NeurIPS (2025) 3, 6
91. Xiao, J., Ehinger, K.A., Oliva, A., Torralba, A.: Recognizing scene viewpoint using panoramic place representation. In: CVPR (2018) 8, 23
92. Xiao, S., Wang, Y., Zhou, J., Yuan, H., Xing, X., Yan, R., Li, C., Wang, S., Huang, T., Liu, Z.: OmniGen: Unified image generation. In: CVPR (2025) 4
93. Xie, K., Sabour, A., Huang, J., Paschalidou, D., Klar, G., Iqbal, U., Fidler, S., Zeng, X.: VideoPanda: Video panoramic diffusion with multi-view attention. arXiv preprint arXiv:2504.11389 (2025) 4
94. Yang, L., Duan, H., Zhu, Y., Liu, X., Liu, L., Xu, Z., Ma, G., Min, X., Zhai, G., Le Callet, P.: Omni2: Unifying omnidirectional image generation and editing in an omni model. In: ACM MM (2025) 3
95. Ye, W., Ji, C., Chen, Z., Gao, J., Huang, X., Zhang, S.H., Ouyang, W., He, T., Zhao, C., Zhang, G.: DiffPano: Scalable and consistent text to panorama generation with spherical epipolar-aware diffusion. NeurIPS (2024) 3
96. Yeshwanth, C., Liu, Y.C., Nießner, M., Dai, A.: ScanNet++: A high-fidelity dataset of 3d indoor scenes. In: ICCV (2023) 11, 21
97. Yin, T., Zhang, Q., Zhang, R., Freeman, W.T., Durand, F., Shechtman, E., Huang, X.: From slow bidirectional to fast autoregressive video diffusion models. In: CVPR (2025) 28
98. Yin, W., Zhang, J., Wang, O., Niklaus, S., Mai, L., Chen, S., Shen, C.: Learning to recover 3d scene shape from a single image. In: CVPR (2021) 12
99. Zhang, C., Wu, Q., Gambardella, C.C., Huang, X., Phung, D., Ouyang, W., Cai, J.: Taming stable diffusion for text to 360 panorama image generation. In: CVPR (2024) 3, 8, 23

100. Zhang, M., Chen, Y., Xu, R., Wang, C., Yang, J., Meng, W., Guo, J., Zhao, H., Zhang, X.: PanoDiT: Panoramic videos generation with diffusion transformer. In: AAAI (2025) 1, 3
101. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR (2018) 10, 23
102. Zheng, D., Zhang, C., Wu, X.M., Li, C., Lv, C., Hu, J.F., Zheng, W.S.: Panorama generation from nfov image done right. In: CVPR (2025) 2, 3
103. Zheng, J., Zhang, J., Li, J., Tang, R., Gao, S., Zhou, Z.: Structured3D: A large photo-realistic dataset for structured 3d modeling. In: ECCV (2020) 6, 12, 21
104. Zhou, T., Zhang, X., Tang, Y.: PanoLlama: Generating endless and coherent panoramas with next-token-prediction llms. In: ICCV (2025) 3
105. Zhou, T., Tucker, R., Flynn, J., Fyffe, G., Snavely, N.: Stereo magnification: Learning view synthesis using multiplane images. ACM TOG (2018) 11, 26
106. Zhu, S., Kumar, A., Hu, M., Liu, X.: Tame a wild camera: In-the-wild monocular camera calibration. NeurIPS (2023) 12

A Detailed Experimental Setup

In this section, we provide full details on the datasets, baselines, evaluation settings, and the training and inference implementation details of our model.

A.1 Training Data

Image data. To facilitate a fair comparison with the previous state-of-the-art method, CubeDiff [29], we follow them to use the same panorama image datasets as training data. These include Polyhaven [57], Humus [55], Structured3D [103], and Pano360 [32]. For Structured3D, we use all three subsets, namely, empty, simple, and full. As a result, around 90% of data are synthetic rendering of indoor rooms from Structured3D. We then use Gemini 2.5 Flash [16] for captioning.
Data augmentation: to handle input images with diverse camera setup at test time, we uniformly sample FoV in [30°, 120°], pitch in [−60°, 60°], roll in [−15°, 15°], and use them to crop the conditioning perspective images for training. We also perform horizontal roll augmentation on the panorama image.

Video data. We use the panorama videos from the 360-1M dataset [78] featuring YouTube videos. Specifically, we take the filtered subset from Argus [46], which removes videos with non-panorama format, very low motion, and bad visual quality. We then run our two-step video canonicalization pipeline (see Figure 8). First, we run COLMAP [64] with rig support to estimate per-frame camera pose, and then rotate each frame to eliminate *inter-frame camera rotation* (i.e., stabilize the video). Then, we run GeoCalib [77] to convert the video to a gravity-aligned upright pose. Since the video is stabilized, all frames should share the same gravity direction. We thus predict the direction for all frames and then average the predictions after removing outliers (i.e., values more than 3 standard deviations from the mean). GeoCalib is only trained on perspective images; thus we project each panorama video frame to eight perspective images (elevation=0 with uniform azimuth), run GeoCalib on each image, and take an average after the same outlier removal technique. Finally, we rotate the video so that its gravity direction is aligned with the vertical axis. Similar to the image data, we use Gemini 2.5 Flash to caption the video (downsampled to 1 FPS). We apply the same pipeline to both the coarse and high-quality subset from Argus.
Data augmentation: to handle in-the-wild perspective video, we follow Argus to simulate camera trajectories with randomly sampled linear motion plus noise. However, we note that simple linear motion is not diverse enough, and models trained on it fail to generalize to videos with complex motion (see Fig. 13). Thus, we also incorporate camera trajectories extracted from real-world videos [59, 96] during training. We randomly sample from simulated (80%) and real-world (20%) trajectories to crop perspective videos as model conditioning.

A.2 Implementation details

Image model. We fine-tune FLUX.1-dev [34], a popular open weights text-to-image diffusion transformer model. The target panorama and conditioning

Fig. 8: Visualization of the video canonicalization pipeline. *Top*: Raw panorama frames exhibit varying elevation angles, causing the horizon to fluctuate relative to the reference line (**red** dashed). *Middle*: After stabilization, inter-frame rotation is removed, resulting in a temporally consistent horizon height across all frames. *Bottom*: After aligning the gravity direction to the vertical axis, the horizon is rectified to a straight line parallel to the image boundaries, ensuring an upright orientation. Please refer to our project page for better comparisons in video format.

perspective images are separately encoded with the VAE, flattened to 1D sequence of tokens, and concatenated along the sequence dimension as model input. We use the same spatial index (x, y coordinate of the token) to apply 3D RoPE [72] to conditioning and target tokens. To distinguish between them in the concatenated sequence, we offset the time dimension index by 1 when applying 3D RoPE to perspective tokens following [35, 87]. We fine-tune the entire model using the Adam optimizer [31] with a batch size of 512 for 50k steps. The learning rate linearly increases from 0 to 5×10^{-5} in the first 1k steps, and then stays constant. A gradient clipping of 1.0 is applied to stabilize training. To apply classifier-free guidance (CFG) [23], we randomly drop the text embedding of the caption and the conditioning image with a 10% probability during training. The model is trained on panorama images in the Equirectangular Projection (ERP) format with a resolution of 1024×2048.

Inference. We use FLUX's default rectified flow sampler [44] with 50 sampling steps. FLUX computes timestep shifting based on the number of tokens. Images at 1024×2048 resolution surpasses its maximum number of tokens (4096), we thus uses its cutoff shifting value of $\exp(1.15) \approx 3.16$. We tried larger value but observed degradation in the result. We apply CFG on both text and image similar to [4], with a scale of 2.0 on text and 1.5 on image.

Video model. We fine-tune Wan2.1-14B [79], a popular open source text-to-video diffusion transformer model. Most of the model design is the same as the image model. The only difference is that, in 3D RoPE, we offset the time dimension index of perspective tokens by 0.1 rather than 1 to avoid confusion with tokens from subsequent frames. We fine-tune the entire model using the Adam optimizer with a learning rate of 1×10^{-5}, and a batch size of 64. The same warmup schedule, gradient clipping, and CFG dropping are applied. The model is first trained on ERP videos with 81 frames, 256×512 resolution from the

coarse subset for 10k steps, and then on ERP videos with 81 frames, 512×1024 resolution from the high-quality subset for another 10k steps.
Inference. We run 50 sampling steps with Wan's default sampler and timestep shifting of 3.0, which outperforms 2.0 and 5.0 in our ablation. We use a CFG weights of 3.0 for text and 2.0 for conditioning on perspective video.

A.3 Evaluation Setup

Perspective-to-360° image generation. We evaluate on the Laval Indoor [15] and SUN360 [91] datasets. To measure visual quality, we report Fréchet Inception Distance (FID) [21], Kernel Inception Distance (KID) [2], FID on CLIP [58] features (CLIP-FID), and FID on features of an auto-encoder fine-tuned on panorama images (FAED) [99]. In line with CubeDiff, FID, KID, and CLIP-FID are computed on 10 perspective crops (10 azimuth angles randomly sampled from [90°, 270°] to avoid overlap with the input view whose elevation=0°) from the generated panorama using the clean-fid package [52]. Meanwhile, FAED is computed directly on the entire panorama as it measures the overall geometry of the ERP. We adopt the implementation from PanFusion [99][1]. We also report CLIP-score (CS) [20] between captions and the ERP images for text alignment.
Perspective-to-360° video generation. We follow Argus and use a hold out set of 101 videos as evaluation data. However, as the exact eval split from Argus is unavailable, we reproduce the eval set based on offline communication with the authors. We tested Argus on this reproduced eval set and ensured that the metrics are comparable to those reported in the paper. Following Argus, we use two types of camera trajectories, namely, *simulated* and extracted from *real-world* videos, to obtain conditioning videos. To measure fidelity, we report PSNR and LPIPS [101] between ground-truth and generated panorama videos within regions covered by the perspective video. Concretely, it is computed by projecting a mask to the ERP space using ground-truth camera information at each step, and then take a union over all steps. We also report FVD [76] on the ERP videos to measure the overall geometry and visual quality. Finally, we adopt Imaging Quality, Aesthetic Quality, and Motion Smoothness from VBench [26] to evaluate overall generation quality. VBench metrics are computed on perspective projections (front, left, right, back) of the generated panorama videos.

A.4 Baselines

Perspective-to-360° image generation. Since we follow the evaluation setup of the previous state-of-the-art (CubeDiff), we borrow the number of baselines from their paper. Here, we only discuss CubeDiff. CubeDiff leverages a cubemap representation, which projects the panorama image to six perspective views (six faces of a cube), each with 90° FoV. At inference time, it inputs the conditioning image as the front face, and denoises the other five faces jointly. This means

[1] https://github.com/chengzhag/PanFusion

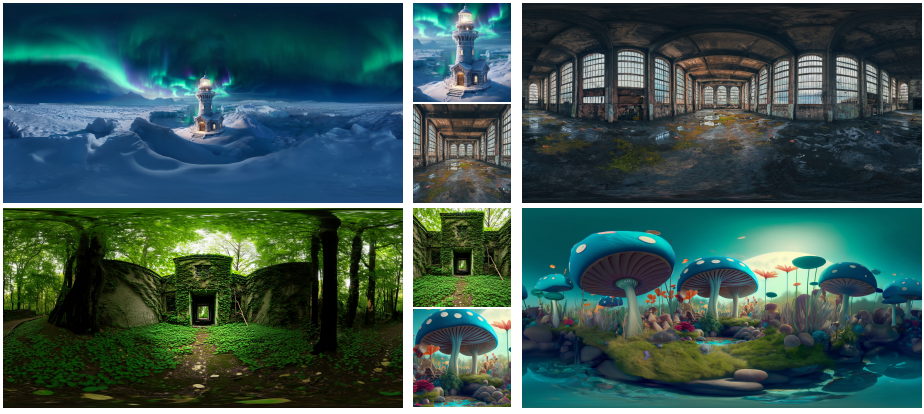


Fig. 9: More perspective-to-360° image generation results from `360Anything`. We test on out-of-distribution images, such as AI-generated ones. The conditioning perspective images are shown at the middle.

they always assumes the input image has a 90° FoV and lies at the *center* of the generated panorama, preventing them from adapting to in-the-wild images with arbitrary FoV and camera orientation.

Perspective-to-360° video generation. We compare with three recent works with released code. Notably, if the method requires camera metadata to project the perspective video to the ERP space, we always use ground-truth camera metadata. This provides additional information compared to `360Anything`.

- *Imagine360* [74] leverages a dual-branch architecture based on AnimateDiff [18] to process perspective and ERP data, connected with spherical attention. We leverage their official code.
- *Argus* [46] projects the perspective video to the ERP space, and fine-tunes SVD [3] to outpaint the entire panorama video. We leverage their official code.
- *ViewPoint* [13] is inspired by the cubemap representation, and designs a viewpoint map representation with less spherical distortion compared to ERP. It then fine-tunes Wan2.1 [79] on this new representation. However, similar to CubeDiff, ViewPoint also places the conditioning video at the front view, leading to severely rotated panoramas when the input video has large camera motion. We leverage their official code.

B More Experimental Results

B.1 Failure Cases of Channel-Concatenation Baselines

A popular line of work in perspective-to-360° generation first projects the perspective input to the ERP space, and then concatenates it with the noisy target latent channel-wise as model input. This requires external models to estimate the FoV and camera orientations for the projection, which is tedious from a user perspective. Moreover, channel-concatenation models may suffer from mistakes

Fig. 10: Qualitative results of 3D scene reconstruction. Given an input monocular video (left), 360Anything outpaints the whole 360° viewpoint (middle), from which we can optimize a 3DGS (right). This allows fly-through exploration of the entire 3D scene.

made by off-the-shelf camera estimators. As shown in Figure 14, when the input video has complex camera trajectories or lighting conditions, even SoTA camera estimators like MegaSaM [39] fail. In the first two examples, the predicted roll angles drift significantly, leading to severely tilted objects in the conditioning view. In the last example, the predicted FoV is too small, making the projected view uninformative. Argus is unable to correct the out-of-distribution conditioning input, and generates broken results. In contrast, 360Anything gets rid of explicit camera information with a sequence-concatenation mechanism, and can still generate reasonable panorama videos from these challenging videos.

B.2 Perspective-to-360° Image Generation

We present results on out-of-distribution perspective images in Figure 9. The conditioning images are generated by text-to-image models. Despite mainly trained on indoor synthetic data, 360Anything still generalizes to these OOD samples with high visual quality and correct overall structure.

B.3 Perspective-to-360° Video Generation

Panoramic video outpainting. We show more qualitative comparisons in Figure 15. In the first two examples, the conditioning perspective videos have large elevation changes (moves up and down). Imagine360 and Argus handle this by projecting the perspective view to the ERP space as model conditioning. Yet, they still generate videos with low visual quality due to the use of poor video backbones [3, 18]. ViewPoint uses the same Wan [79] backbone as us. However, it always treats the input frames as the front-view of the cubemap representation, and thus has to generate significantly rotated panoramas. This leads to severely distorted humans and objects. Thanks to the canonicalization training objective, 360Anything produces videos that are always upright, maintaining the correct scene geometry. In the last example, the conditioning video contains a partially observed hand. All baselines fail to outpaint the entire person, and only generate

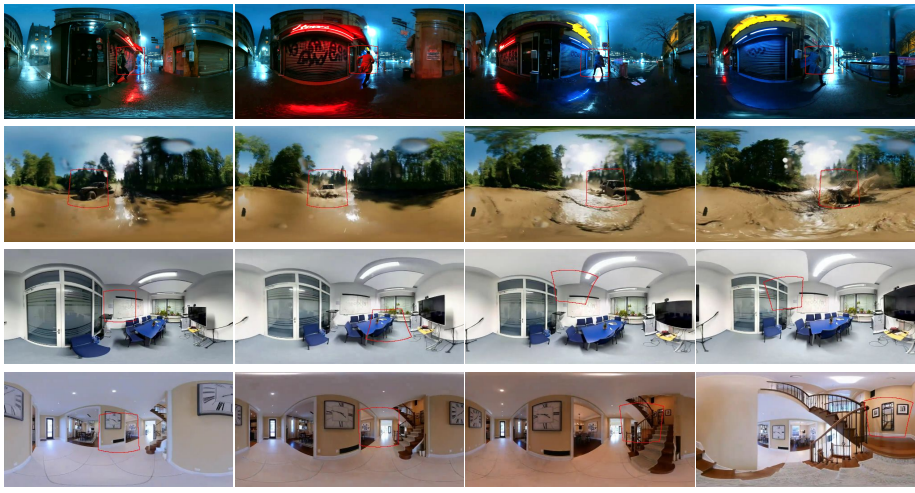


Fig. 11: Panorama video generation given large motion videos. Regions corresponding to the input conditioning video are highlighted in **red**. We test on perspective videos with large object or camera motion.

Fig. 12: Panorama video generation on AI generated videos. Regions corresponding to the input conditioning video are highlighted in **red**. We test on perspective videos generated by other video models.

a small part of an arm. In contrast, `360Anything` outpaints the entire person following the text prompt, and still maintains it when the hand moves out of the input view. This shows the strong world knowledge of our method.

3D scene reconstruction. We show more qualitative results in Figure 10. Given a narrow field-of-view video from RealEstate10K [105], `360Anything` synthesizes the entire 360° view of the room. We can then train a 3D Gaussian Splatting model [30] on the generated panoramas for novel view synthesis. This demonstrates the high 3D consistency of our generated videos.

Fig. 13: Ablation on real-world camera trajectories. 360Anything uses both simulated and real-world camera trajectories to crop perspective videos in model training. Models without this setup generate panorama frames with changing gravity directions, fail to produce canonicalized videos and may suffer from broken structure.

Results on large motion videos. We stress test our model on perspective videos with large object or camera motion. Figure 11 shows that 360Anything is still able to produce temporally consistent videos that are in a stable canonical pose. This shows that our model understands complex geometry and is able to find correspondence across the 4D world.

Results on OOD videos. We test our model on perspective videos generated by other video models, including Wan, Sora, Veo, and Runway Gen-4.5. Figure 12 shows that 360Anything still generates panorama videos with high quality. Notice how the reflection of billboards on water is well handled in the first example, even though it is not present in the input. In the second example, the dust caused by the car persists after the car drives by. In the third example, the model generates the statue underwater when the camera is above the water surface. The last example features stylized black-and-white footage.

Ablation on real-world camera trajectories. We tested the model on in-the-wild perspective videos with large camera motion. Large camera motion makes maintaining a canonicalized panorama output more challenging. As shown in Figure 13, models trained only with simulated camera trajectories fail to produce frames with changing gravity directions. In contrast, 360Anything with real-world camera trajectories generates stably canonicalized panorama videos.

C Limitations and Future Work

360Anything is fine-tuned from a *pre-trained* video diffusion model, and thus we are bounded by the capacity of the base model. For example, it is challenging to outpaint scenes involving complex physics. In addition, we also inherit the bias in its training data. For example, the model sometimes generates panorama

videos with black borders or undesired objects (e.g., a tripod or a human's hand) at the bottom of the video since they are common in YouTube 360° videos.

Due to the high resolution of panorama data (an ERP video has 8× number of pixels compared to a normal perspective video) and the limited compute, our current video model can only handle videos with 81 frames. A larger context window will enable larger-scale 3D world generation. An interesting future direction is combining `360Anything` with recent progress in long video generation that distills bi-directional DiTs to *causal* autoregressive DiTs [7,24,97]. To obtain higher-resolution panoramas, we tried existing video upsamplers designed for perspective videos [67]. However, this often re-introduces seams at the ERP boundary, and distorts the structure of the ERP space, calling for research in panorama upsampling techniques.

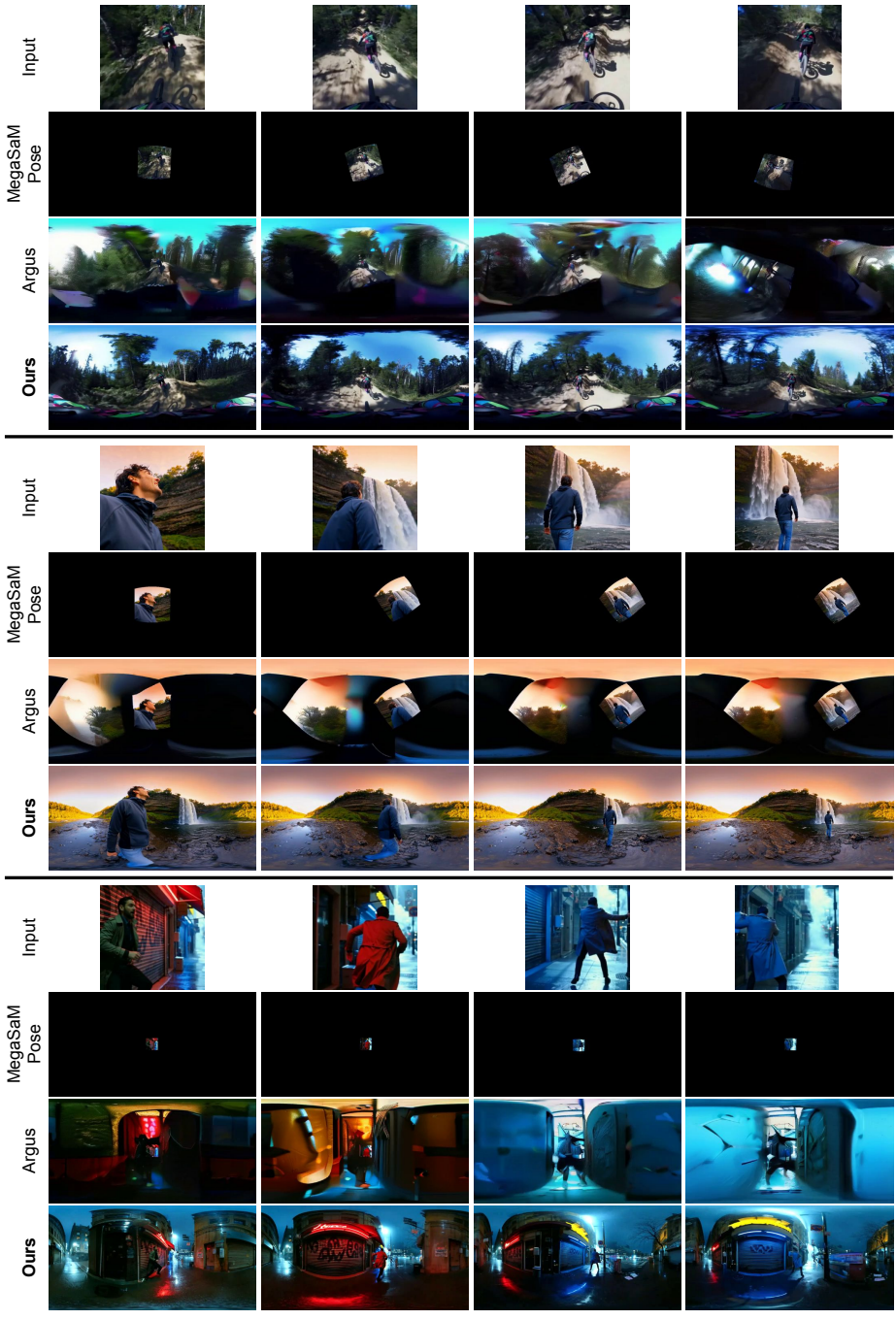


Fig. 14: Perspective-to-360° video generation on challenging input videos. MegaSaM [39] fails to predict the correct camera poses (first two examples) or FoV (last example), leading to degraded generation results from Argus. In contrast, our method runs end-to-end without a projection stage and thus generalizes well.

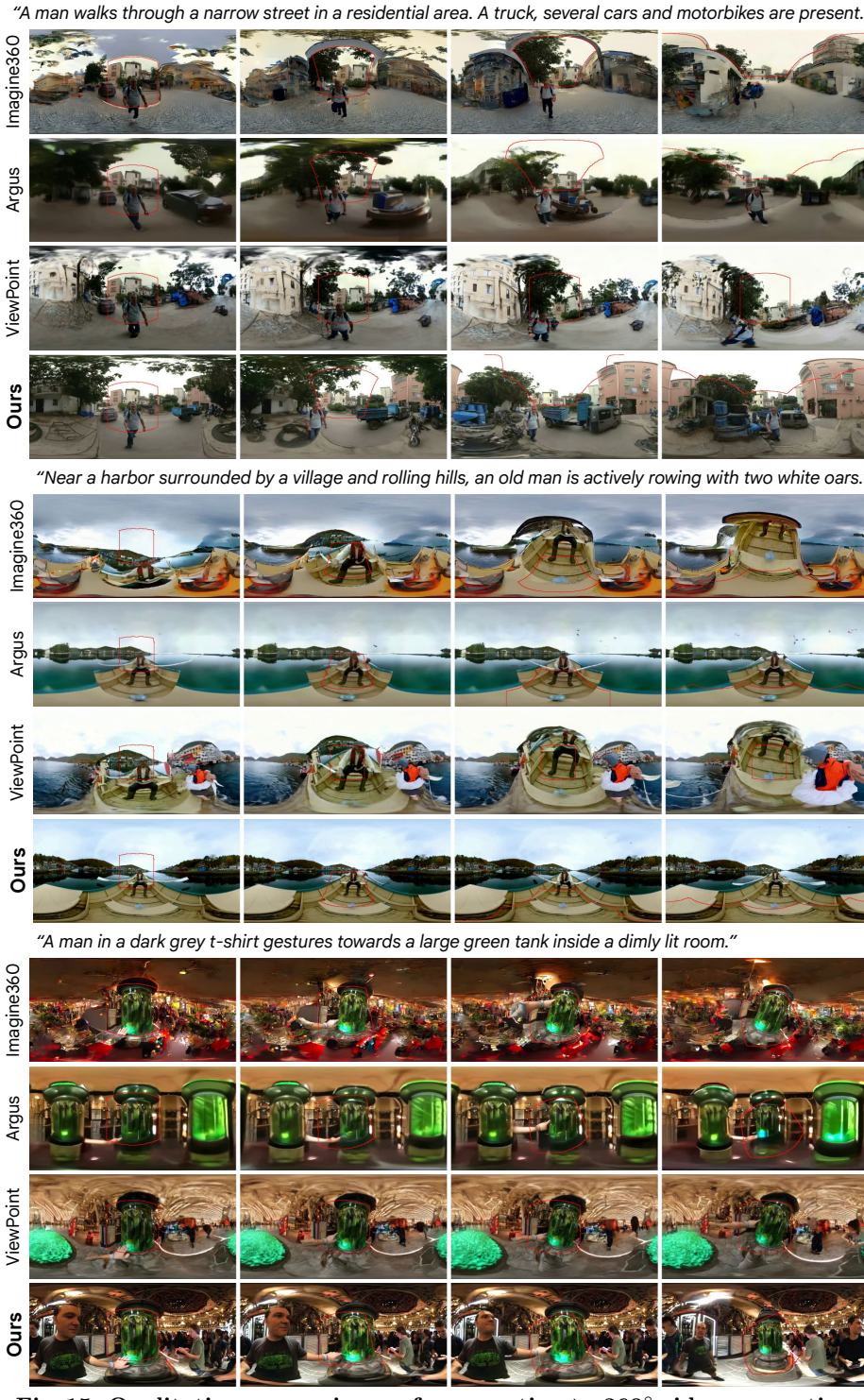


Fig. 15: Qualitative comparisons of perspective-to-360° video generation.