Diffusion Models for 4D Novel View Synthesis
Supplementary Material

This supplementary material expands upon the main paper, 'Diffusion Models for 4D Novel View Synthesis', by providing in-depth details and analyses. We first delve into the scale calibration process for the RealEstate10K dataset and the scene filtering methodology (Section A). We also discuss the effect of multi-guidance (Section B). Next, we provide a comprehensive overview of the 4DiM architecture, including its UViT backbone, transformer blocks, and camera ray conditioning (Section C). We then detail the training methodology and compute cost (Section D) and elaborate on the evaluation metrics used (Section E). Section F offers details of the implementation of baseline models from previous work, and Section G presents a qualitative comparison of 360° scene generation between 4DiM and ZeroNVS. Finally, we showcase the capabilities of 4DiM with an extensive collection of generated samples (Section H). For a more complete collection of samples, please also visit `https://anonymous-4d-diffusion.github.io`.

A CALIBRATED REALESTATE10K

To calibrate RealEstate10K, we predict a single length scale per scene with the help of a recent zero-shot model for monocular depth (Saxena et al., 2023) that predicts metric depth from RGB and FOV. For each image we compute metric depth using the FOV provided by COLMAP. The length-scale is then obtained by regressing the 3D points obtained by COLMAP to metric depth at image locations to which the COLMAP points project. An L1 loss provides robustness to outliers in the depth map estimated by the metric depth models. The mean and variance of the per-frame scales yields a per-sequence estimator. We use the variance as a simple measure of confidence to identify scenes for which the scale estimate may not be reliable, discarding ∼30% of scenes with the highest variance. Table 6 shows the importance of filtering out scenes with less reliable calibration. The calibrated dataset will be made public to facilitate quantitative comparisons.

	FID (↓)	FDD (↓)	FVD (↓)	$TSED_{t=2}$ (↑)	$SfMD_{pos}$(↓)	$SfMD_{rot}$(↓)	LPIPS (↓)	PSNR (↑)	SSIM (↑)
cRE10k test									
4DiM-R (no filtering)	31.81	313.8	237.6	0.9815 (1.000)	1.035 (1.049)	0.3328 (0.3529)	0.3107	16.49	0.4465
4DiM-R (w/ filtering)	**31.23**	**306.3**	**195.1**	**0.9974** (1.000)	**1.023** (1.075)	**0.3029** (0.3413)	**0.2630**	**18.09**	**0.5309**
ScanNet++ zero-shot									
4DiM-R (no filtering)	24.35	**232.0**	143.1	n/a (0.9815)	1.881 (1.774)	1.639 (1.512)	0.1990	20.15	0.6436
4DiM-R (w/ filtering)	**22.89**	243.2	**130.6**	0.9685 (0.9815)	**1.851** (1.917)	**1.541** (1.550)	**0.1809**	**21.25**	**0.6952**
cLLFF zero-shot									
4DiM-R (no filtering)	**62.55**	369.5	848.5	0.9167 (0.9962)	**0.9214** (0.8844)	0.2194 (0.1932)	0.5412	11.48	0.1389
4DiM-R (w/ filtering)	63.48	**353.2**	**841.6**	**0.9659** (0.9962)	0.9265 (0.8951)	**0.2011** (0.1934)	**0.5403**	**11.55**	**0.1444**

Table 6: **Data filtering ablation.** We show the importance of discarding scenes where scale calibration is least reliable. Beyond the clear impact on fidelity, metric scale alignment itself improves by a wide margin. This is quantitatively shown through the reference-based image metrics (LPIPS, PSNR, SSIM) which favor content alignment, as discussed in Section 4.

B MULTI-GUIDANCE

To assess the effectiveness of multi-guidance, we conducted a two-stage evaluation. First, we evaluated samples generated with various 'standard' guidance weights, where all conditioning variables (image, pose, and timestamp) were treated uniformly (illustrated in blue in Figure 7). Through qualitative and quantitative analysis, we identified the optimal standard guidance weight. Next, while holding this image guidance weight constant, we systematically varied the pose or timestamp guidance weights (shown in red). This procedure was applied to two tasks: single-image 3D generation using the cRE10K dataset, and video extrapolation from the first two frames using the DAVIS dataset. As demonstrated in Figure 7, our findings indicate that for video extrapolation, a slightly stronger timestamp guidance enhances performance in both dynamics and FID. For 3D generation, a slightly stronger pose guidance improves pose alignment (measured by TSED) with minimal impact on FID.

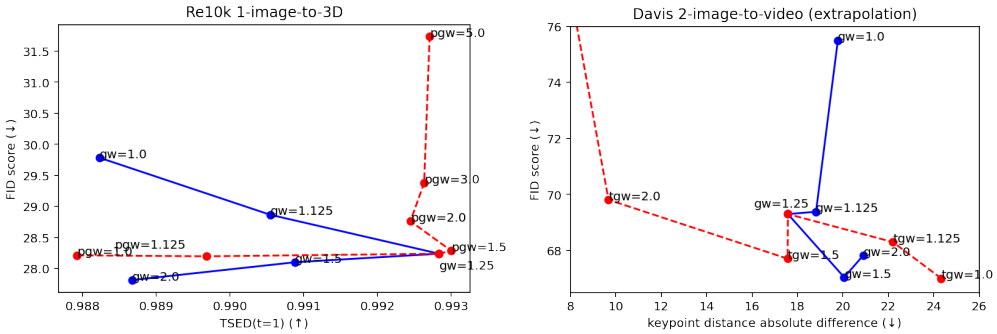

Figure 7: **Multi-guidance ablation.** Left: Pareto plot between FID and TSED for single image to 3D task. Right: Pareto plot between FID and keypoint distance for video extrapolation task.

C Neural Architecture

UViT backbone. As hinted by Figure 2, 4DiM uses a UViT architecture following Hoogeboom et al. (2023). Compared to the commonly employed UNet architecture (Ronneberger et al., 2015), UViT uses a transformer backbone at the bottleneck resolution with no convolutions, leading to improved accelerator utilization. Information across frames mixes exclusively in temporal attention blocks, following (Ho et al., 2022b). We find that this is key to keep computational costs within reasonable limits compared to, say, 3D attention (Shi et al., 2023). Because the temporal attention blocks have a limited sequence length (32 frames), we insert them at all UViT resolutions. To limit memory usage, we use per-frame self-attention blocks only at the 16x16 bottleneck resolution where the sequence length is the product of the current resolution's dimensions.

Transformer block design. Our transformer blocks combine helpful choices from several prior work. We use parallel attention and MLP blocks following Dehghani et al. (2023), as we found in our early experiments that this leads to slightly better hardware utilization while achieving almost identical sample quality. We also inject conditioning information in a similar fashion to Peebles & Xie (2023), though we use fewer conditioning blocks due to the use of the parallel attention + MLP. We additionally emply query-key normalization following the findings of Gilmer et al. for improved training stability, but with RMSNorm (Zhang & Sennrich, 2019) for simplicity.

Conditioning. We use the same positional encodings for diffusion noise levels and relative timestamps as (Saharia et al., 2022b). For relative poses, we follow 3DiM (Watson et al., 2022), and condition generation with per-pixel ray origins and directions, as originally proposed by SRT (Sajjadi et al., 2022) in a regressive setting. Xiong et al. (2023) has compared this choice to conditioning via encoded extrinsics and focal lengths a-la Zero-1-to-3 (Liu et al., 2023b), finding it advantageous. While a thorough study of different encodings is missing in the literature, we hypothesize that rays are a natural choice as they encode camera intrinsics in a way that is independent of the target resolution, yet gives the network precise, pixel-level information about what contents of the scene are visible and which are outside the field of view and therefore require extrapolation by the model. Unlike Plucker coordinates, rays additionally preserve camera positions which is key to deal with occlusions in the underlying 3D scene.

D Compute, Training Time, and Sharding

We train 8-frame 4DiM models for $\sim$1M steps. Using 64 TPU v5 chips, we achieve a throughput of approximately 1 step per second with a batch size of 128. The model has $\sim$2.6B parameters, and available HBM is maximized with various strategies: first, we use bfloat16 activations (but still use float32 weights to avoid instabilities). We also use FSDP (i.e., zero-redundancy sharding (Rajbhandari et al., 2020) with delayed and rematerialized all-gathers). We then finetune this model to its final 32-frame version. This strategy allows us to leverage large batch-size pre-training, which is not possible with too many frames because the amount of activations stored for backpropagation scales linearly respect the number of frames per training example. The model is finetuned with the same number of chips, albeit only for 50,000 steps and at batch size of 32. This allows us to shard the frames of the video and use more than one chip per example at batch size 1. When using temporal

attention, we simply all-gather the keys and values and keep the queries sharded over frames (one of the key insights from Ring Attention (Liu et al., 2023a)), though we don't decompose the computation of attention further as we find we have sufficient HBM to fully parallelize over all-gathered keys and values. FSDP is also enabled on the frame axis to maximize HBM savings so the number of shards is truly the number of chips (as opposed to the number of batch-parallel towers). This is possible because frame sharding requires an all-reduce by mean on the loss over the frame axis. Identically to zero-redundancy sharding, this can be broken down into a reduce-scatter followed by an all-gather.

E Further details on proposed metrics

We follow Yu et al. (2023a) and compute TSED only for contiguous pairs in each view trajectory, and discard sequences where we find less than 10 two-view keypoint matches. We use a threshold of 2.0 in all our experiments. For our proposed SfM distances, in order to get a scale-invariant metric, we first align the camera positions predicted by COLMAP by relativizing the predicted and original poses with respect to the first conditioning frame. This should resolve rotation ambiguity. Then, we resolve scale ambiguity by analytically solving for the least squares optimization problem that aligns the original positions to re-scaled COLMAP camera positions. Finally, we report the *relative error* in positions under the L_2 norm, i.e., we normalize by the norm of the original camera positions.

F Further details on baselines from prior work

For PNVS, the conditioning image is the largest square center crop of the original conditioning frame, resized to 256×256 pixels. This is the same preprocessing procedure used by all our 4DiM models. PNVS generates output frames sequentially (one at a time), using 2000 denoising steps for each frame. All 4DiM samples are produced with 256 denoising steps.

For MotionCtrl, we use the checkpoint based on Stable Diffusion, which generates 14 frames conditioning on one image and 14 poses. We set the sampling hyper-parameter speed to 1, use 128 ddim denoising steps, and keep the other hyper-parameters as default values in the released script from MotionCtrl. To avoid any loss of quality due to an out-of-distribution resolution or aspect ratio, we use the resolution 576×1024, i.e. the same resolution as MotionCtrl was trained on. RE10K images already have the desired aspect ratio. LLFF images have resolution 756×1008; we are thus forced to crop them to 567×1008 so they have the same aspect ratio MotionCtrl was trained with. For comparability with 4DiM and PNVS, we *postprocess* MotionCtrl samples at 567×1008 by taking the largest center crop and then resizing to 256×256. Note that, in order to preserve the aspect ratio of LLFF, the resulting square outputs are for a smaller region than 4DiM and PNVS, hence the gray padding for MotionCtrl LLFF samples in Figure 3.

For ZeroNVS, we use its diffusion model to evaluate novel view synthesis on RE10K and LLFF. Both input and output images are 256×256. We use default hyper-parameters as in the open sourced configuration, except the 'scene scale' parameter used for dealing with length scale ambiguity. We sweep over scale in $[0.7, 1.0, 1.2, 1.5, 2.0, 3.0, 5.0]$ and choose 3.0 for RE10K and 1.5 for LLFF.

G Qualitative comparison of 360° generation vs ZeroNVS

We employ the ZeroNVS pipeline with score distillation sampling (SDS) to generate 360° views conditioned on a single image. For each scene, we use the default scene-generation configuration as open sourced by ZeroNVS, as adjusting *camera_distance* and *elevation_angle* yield worse and incoherent generation. We adjust the *scale* parameter for each scene, and use $1.0, 3.0, 3.0$ for the three scenes in Figure 8. Due to the computationally intensive nature of 360° generation with ZeroNVS (2-3 hours per scene), exhaustive hyperparameter tuning is impractical. We also observe another limitation: volume rendering will fail if the camera positions are not allowed to vary (or even if *camera_distance* is too small, e.g., 0.5), i.e., it is not feasiable to keep the camera position fixed and only employ rotation. This occurs because there is no parallax and it is therefore not possible to infer depth/density (rotation alone only warps pixels).

Figure 8 presents a comparison between 4DiM and ZeroNVS outputs, showing the superior sharpness and richer content achieved by 4DiM.

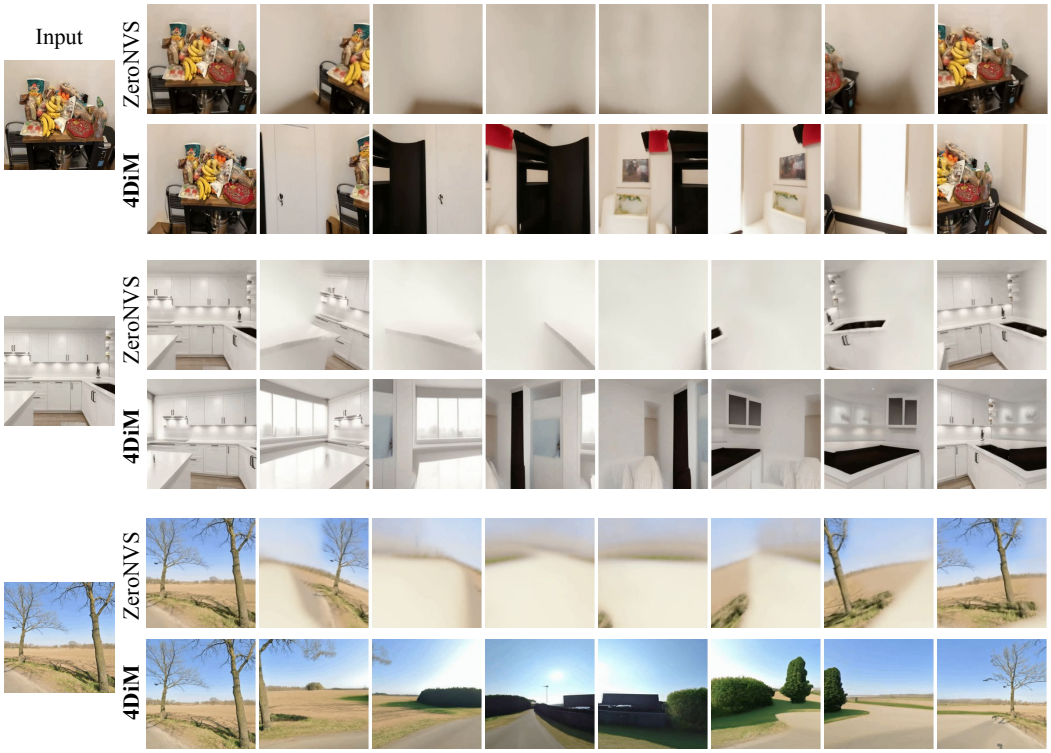

Figure 8: Qualitative comparison of 360° generation between 4DiM and ZeroNVS. 4DiM achieves superior sharpness and richer content.

H SAMPLES

We include more 4DiM samples below, though we strongly encourage the reader to browse our website: https://anonymous-4d-diffusion.github.io

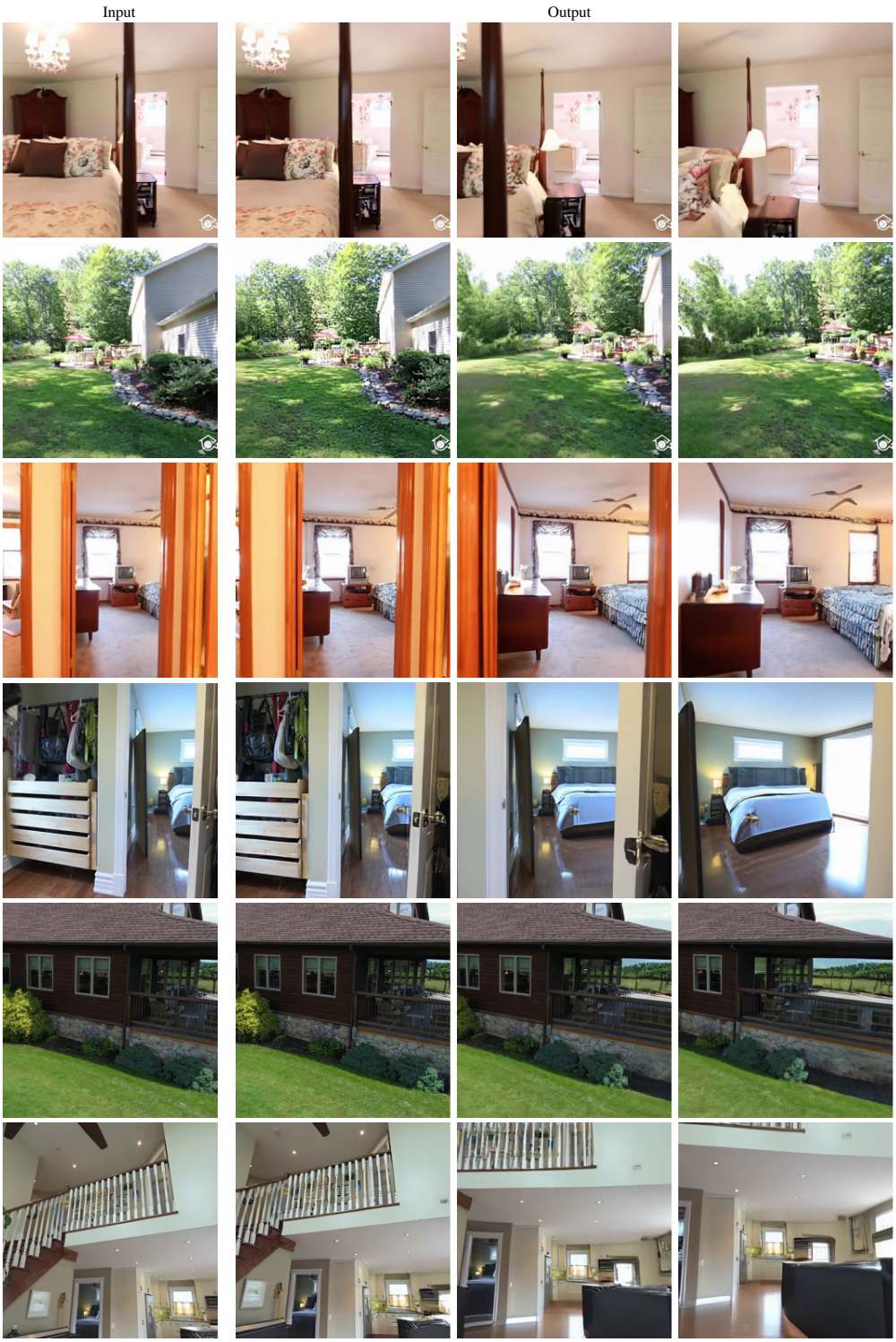

Figure 9: More 4DiM samples from the RealEstate10K dataset.

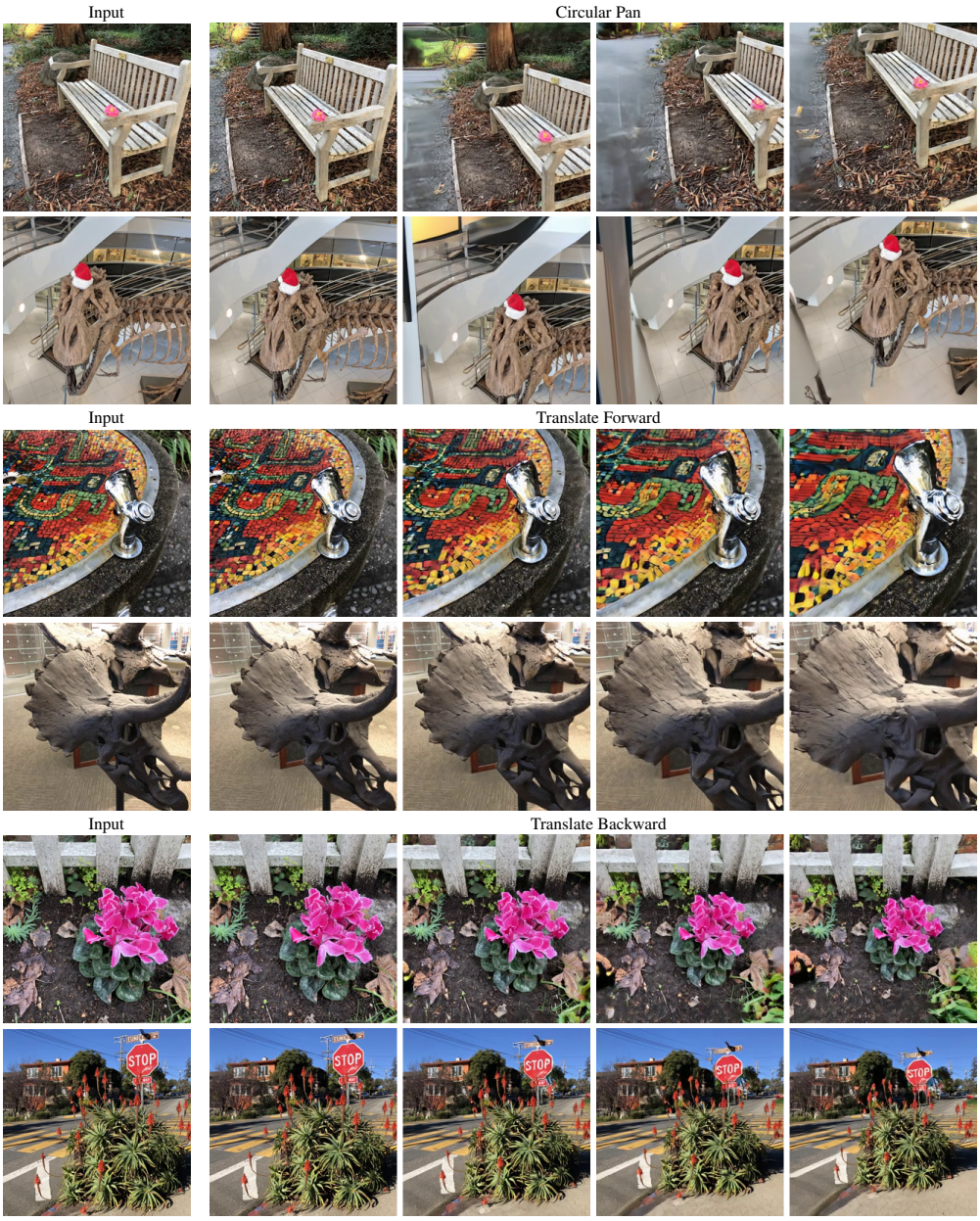

Figure 10: More 4DiM samples with custom trajectories from the LLFF dataset.

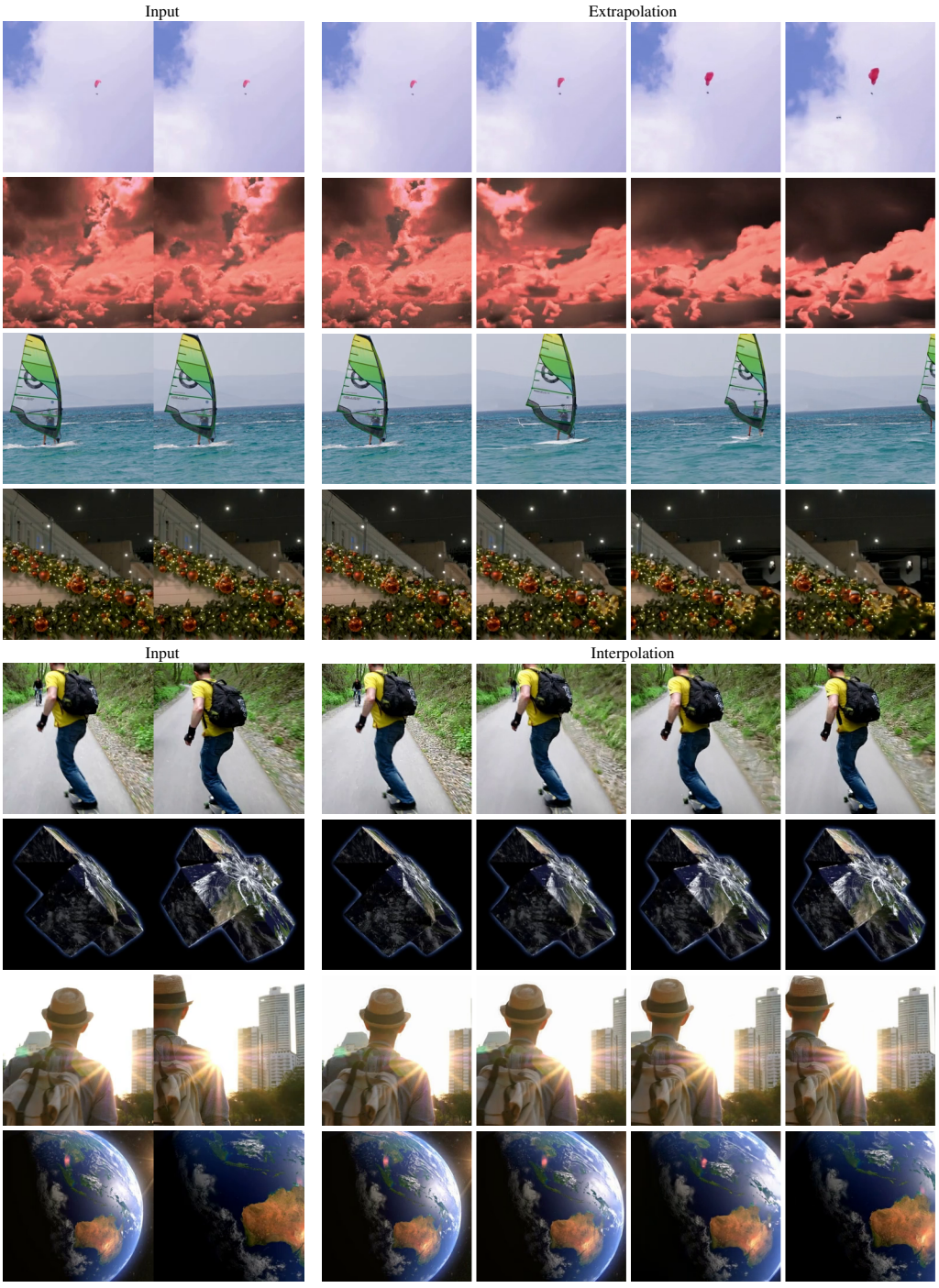

Figure 11: More 4DiM samples of videos generated from 2 input images.

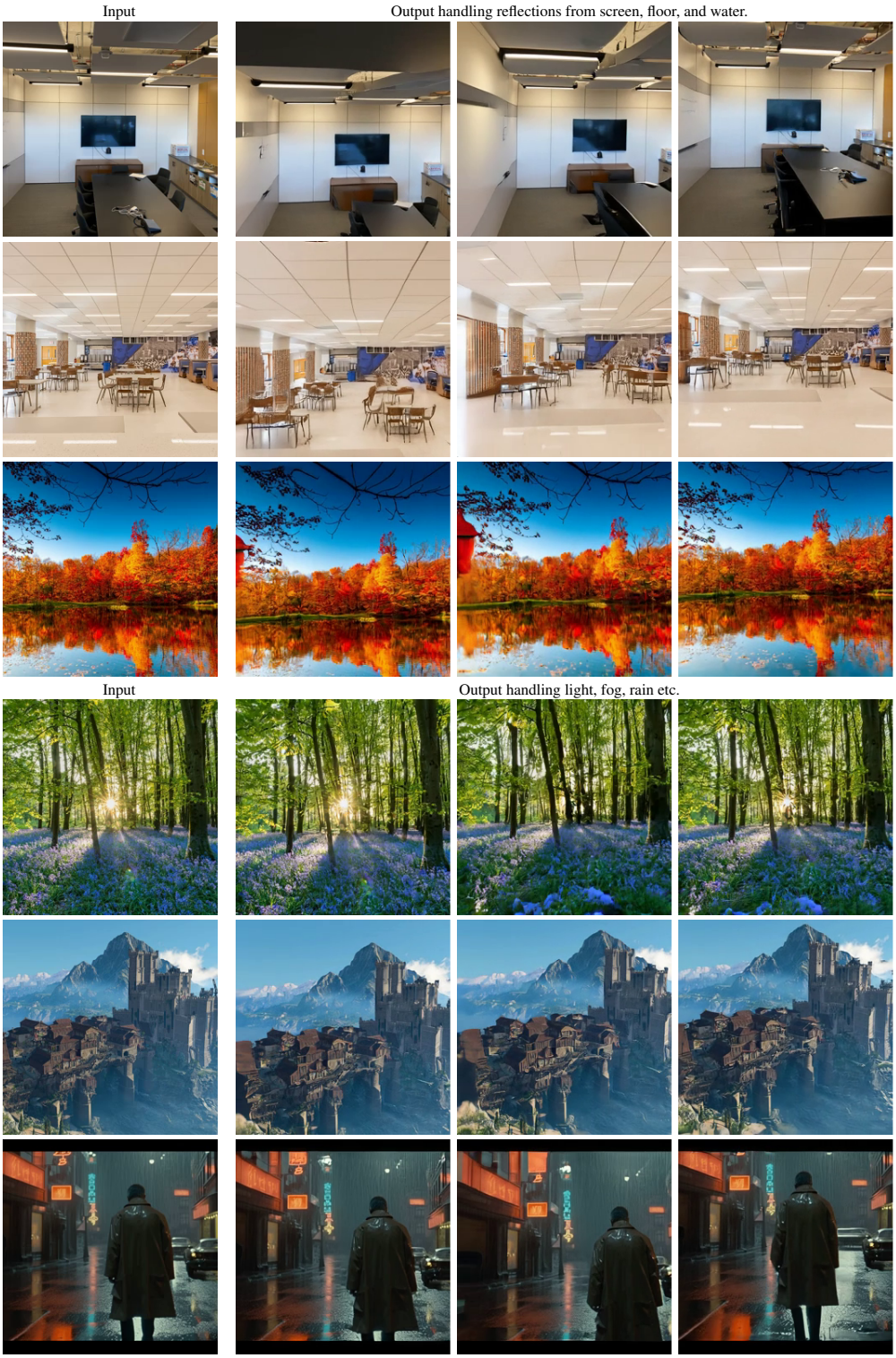

Figure 12: 4DiM can handle challenging scenarios such as reflection, light rays, fog, rain, etc.