

MEt3R: Measuring Multi-View Consistency in Generated Images

Supplementary Material

The supplementary materials are structured as follows. First, we provide detailed information about our multi-view latent diffusion model (MV-LDM) in Sec. A. Then, we compare MEt3R with the FWS variants in Sec. B, followed by a discussion on MEt3R metric in Sec. C. Finally, we present additional details on the multi-view generation baselines in Sec. D and their corresponding runtime statistics in Sec. E. Please also note our **supplementary video**, showcasing evaluations in motion.

A. Multi-View Latent Diffusion Model

This section presents further details for MV-LDM, including the architectural components, training, and sampling details.

A.1. Architecture.

Like CAT3D [11], our architecture is based on a multi-view 2D UNet shared across multiple input views with 3D self-attention at each UNet block. We initialize the UNet weights with Stable Diffusion 2.1 [29] and replace each attention layer with a 3D self-attention layer from MVDream [33] where each token from one view attends to all tokens from the other views. This accounts for 1.1B parameters for the multi-view UNet and 83.7M for the VAE. Due to memory and resource limitations, we fix the total number of concurrent views to 5, including the target and the conditioning views. Figure 10 shows the architecture of MV-LDM. We apply a VAE encoder and map the input images ($H \times W \times 3$) into latent representation ($\frac{H}{8} \times \frac{W}{8} \times 4$). For the low-resolution latent maps, we generate the ray encodings of shape ($\frac{H}{8} \times \frac{W}{8} \times 6$), which consists of a 3-dimensional origin and a 3-dimensional direction vector in relative camera space and concatenate it along the channel dimension.

A.2. Training and Evaluation with MEt3R

Dataset. We use RealEstate10K [51], which consists of 80K video sequences accounting for 10 million frames. During training, we randomly select a video sequence and the corresponding conditioning and target views that satisfy the following criteria:

- Sample 2 conditioning views (left and right) at frame number f_L and f_R with frame distance $d_c = f_R - f_L$ satisfying $50 \leq d_c \leq 180$.
- Sample 3 target views with distance d_t from the conditioning view that satisfies $f_L - 100 \leq d_t \leq f_R + 100$.

Afterward, we transform the absolute poses into relative poses with respect to the first conditioning view.

Training. The training procedure follows DDPM [14], sampling a noise level t , applying that to all given latent images and training the network to predict the noise present in the image. We randomly select the conditioning views N between 1 or 2 and the target views M between 3 and 4, respectively, to allow for single and few-view novel view generation. We linearly vary the beta schedule from 0.0001 to 0.02 for the forward diffusion process and train MV-LDM for a total of 1.65M iterations with an effective batch size of 24 at resolution 256^2 . We use AdamW [23] optimizer with a constant learning rate of $2e^{-5}$. During sampling, the network can receive a combination of existing and pure noise images with camera ray encodings to perform conditional generation. The backward diffusion process is done with ϵ -parameterization defined as the output of the model ϵ_θ :

$$\epsilon_{pred} = \epsilon_\theta(\mathbf{z}_t, \mathbf{c}_t, t), \quad (5)$$

where $\epsilon_{pred} = (\epsilon_{pred}^i)_{i=1}^M$ is the predicted noise latent, $\mathbf{z}_t = (\mathbf{z}_t^i)_{i=1}^M$ is the noisy latent, $\mathbf{c}_t = (\mathbf{c}_t^j)_{j=1}^N$ is the clean latent at the timestep t , whereas M and N are the number of target and conditioning views, respectively. The predicted noise is used to make a step in the direction of a sample in the target distribution under the DDIM [34] formulation. For classifier-free guidance, we randomly drop the clean conditioning views with a probability of 10%, and during sampling, we apply a guidance scale of 3 similar to CAT3D [11].

For training, we apply the standard diffusion loss on the predicted mean noise as the mean-squared error (MSE) against the ground truth noise:

$$\mathcal{L} = \|\epsilon - \epsilon_\theta(\mathbf{z}_t, \mathbf{c}_t, t)\|_2^2, \quad (6)$$

where $\epsilon = (\epsilon^i)_{i=1}^M$ and $\epsilon^i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is the ground truth noise for each target view.

Training evolution of MEt3R. Figure 11 shows the trend in 3D consistency in terms of MEt3R over training iterations, showing consistent improvements with longer training. There is a significant improvement in the initial 100k, and afterward, it saturates near 1M iterations.

Anchored vs. autoregressive sampling. We further test MEt3R with two sampling strategies, i.e., (1) autoregressively generating new target views and new anchors, conditioned on the previous anchor, and (2) using anchored sampling where we generate anchors first and then the rest as described in Sec. 4. Fig. 12 shows the average MEt3R plot

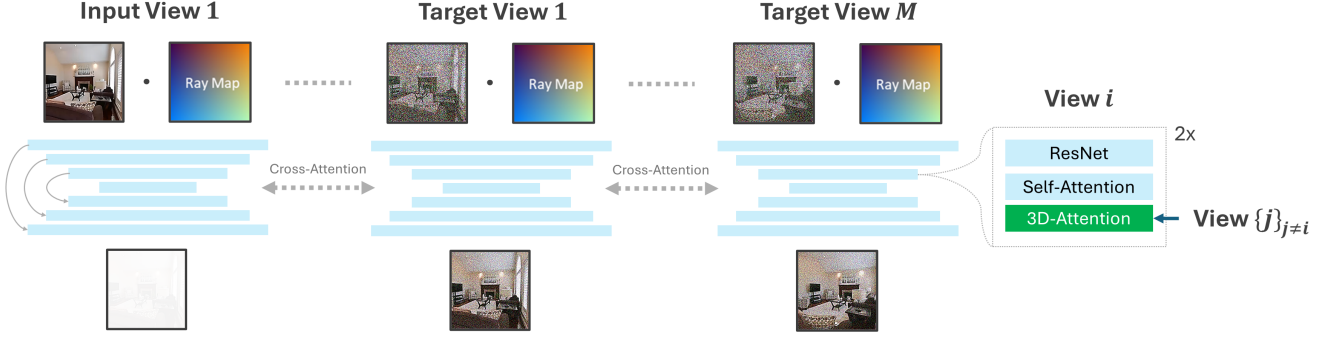


Figure 10. **MV-LDM**. Architecture overview of MV-LDM, which consists of a shared 2D UNet initialized from Stable Diffusion 2.1 [29] across multiple input views with cross-view attentions (3D attention) in between for modeling multi-view prior.

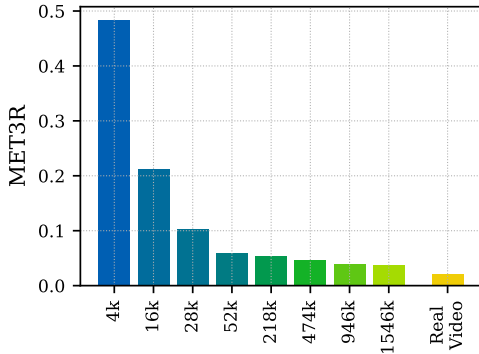


Figure 11. **MET3R at different training iterations**. As we continue to train MV-LDM, we see a consistent improvement in 3D consistency, which is an expected behavior. Furthermore, in the beginning, the improvements are large, which slows down and saturates in the later iterations.

per image-pair, showing the improvements with anchored sampling. We observe many diverging peaks attributed to several anchor-to-anchor transitions and accumulating errors for autoregressive sampling. For anchored sampling, the anchors are generated together first, followed by generating the rest. This limits the error accumulation and allows for fewer anchor-to-anchor transitions. We refer to Fig. 13 for a visual illustration of anchored and autoregressive sampling schemes.

MET3R on multiple scales. In Tab. 2, we investigate the effect of image resolutions on MET3R compared to SED [48]. We find that SED is highly sensitive to variation in image resolution with a significant increase at 128^2 . This is expected since SED computes the geometric distance of each correspondence from their epipolar line in the 2D-pixel space. Meanwhile, MET3R is more robust, attributed to the measurement in the feature space (c.f. Sec. 3), thus maintaining only minor differences in the scores. Although

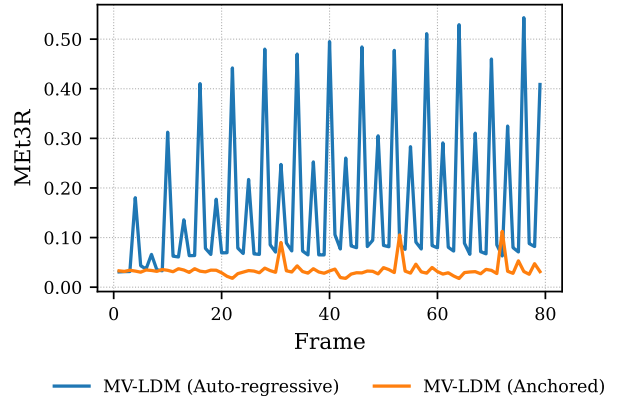


Figure 12. **Anchored vs. autoregressive**. Per-image-pair MET3R on 2 different sampling strategies. For autoregressive sampling, we see significant and periodic spikes becoming larger as we progress and show the effect of compounding error, i.e., sequentially generating new frames and anchors conditioned on the previously generated ones. As illustrated in Fig. 13, autoregressive sampling produces several anchor-to-anchor transitions causing these periodic spikes. On the other hand, anchored generation limits the effect of compounding error by generating all anchors in parallel.

the differences are small, we still recommend using a similar resolution for all baselines for a fair comparison.

B. Comparison of FWS Variants

Other metrics based on flow warping score (FWS) have been introduced to measure consistency, which uses optical flow, e.g., RAFT [37]. Given a pair of images, it first computes optical flow, which is used to warp one image into the other. Then, metrics such as SSIM, LPIPS, PSNR, and RMSE are computed to quantify multi-view consistency.

In Tab. 3, we evaluate both multi-view and video generation baselines on FWS and MET3R. We find that most variants, including PSNR, SSIM, and RMSE, rank DFM better

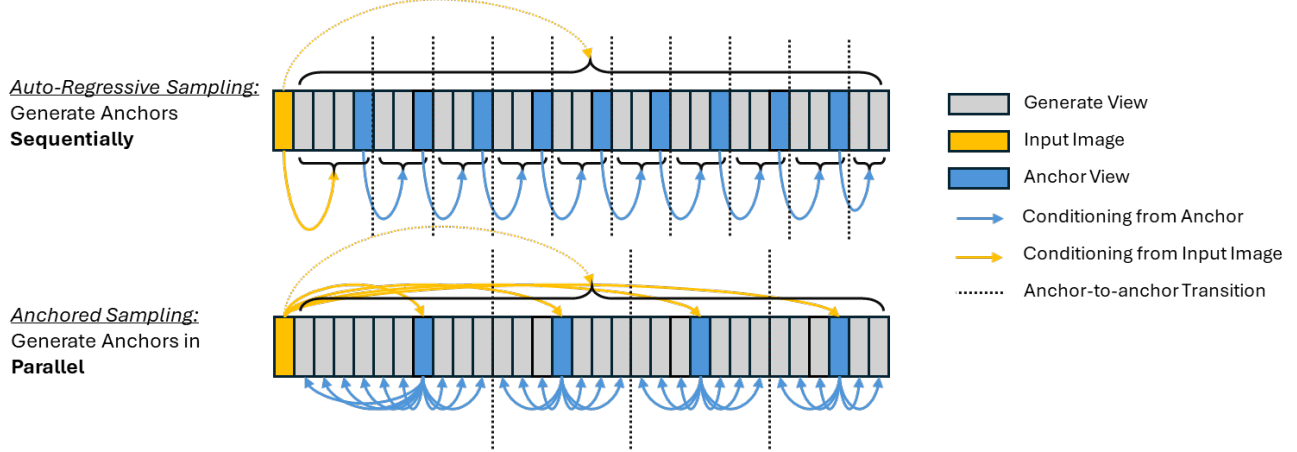


Figure 13. **Anchored vs. Autoregressive sampling schemes.** An illustration of the differences in the sampling schemes. In autoregressive sampling, we start from the initial input image and generate a set of target frames. The next set of frames is conditioned both on the input image and on the last frame (anchor) of the previously generated set. With this sampling strategy, we see several anchor-to-anchor transitions and results in large inconsistencies as visible in Fig. 12. Whereas using anchored generation i.e., generate anchors first and then sample the remaining conditioned on the closest anchor and the input image. With this strategy, we observe significantly fewer anchor-to-anchor transitions, limited error accumulation, and relatively stable and lower MEt3R across the image pairs.

	256 ²		224 ²		128 ²	
	MEt3R	SED	Diff _{MEt3R}	Diff _{SED}	Diff _{MEt3R}	Diff _{SED}
GenWarp [32]	0.120	1.398	-2.58%	-74.84%	+6.61%	-80.37%
PhotoNVS [48]	0.069	0.479	-3.89%	-26.23%	+1.60%	-42.50%
MV-LDM (Ours)	<u>0.036</u>	<u>0.405</u>	-3.88%	+2.29%	+16.66%	+34.23%
DFM [38]	0.026	0.346	-9.35%	-52.84%	+5.73%	-70.20%
Real Video	0.022	0.181	-2.46%	+18.99%	+47.87%	+148.17%

Table 2. **MET3R vs. SED on multiple resolutions.** We show differences in SED [48] and MEt3R for the baseline multi-view generation models over changing image resolution against the base resolution of 256² in percentage. MEt3R is more robust to variations in the input resolution since it measures in feature space, unlike SED, which measures in pixel space (c.f. Sec. 3). Here, SEDs total scale is less than one order of magnitude larger than MEt3Rs, while its variations are more than one order of magnitude larger in most cases.

than real video among multi-view generation methods because of their sensitivity to blur, pixel-level perturbation, and noise (c.f. Sec. 5.3 and 5.4). Meanwhile, MEt3R and FWS (LPIPS) ignore such perturbations and rely on feature and perceptual similarity, respectively. However, as shown in Fig. 14, FWS (LPIPS) suffers at higher frame distances between input pairs, where DFM, GenWarp, and MV-LDM can score better than real videos.

C. Additional MEt3R Architectural Details

This section presents additional details on the MEt3R pipeline, including the projection of both point maps to the first view and a description of the overlap mask used.

	MEt3R ↓	FWS			
		PSNR ↑	SSIM ↑	LPIPS ↓	RMSE ↓
GenWarp [32]	0.120	21.41	0.716	0.200	0.097
PhotoNVS [48]	0.069	25.10	0.779	0.137	0.060
MV-LDM (Ours)	0.036	28.46	0.851	0.095	0.044
DFM [38]	<u>0.026</u>	39.56	0.948	<u>0.082</u>	0.011
Real Video	0.022	<u>33.52</u>	<u>0.924</u>	0.075	<u>0.026</u>
I2VGen-XL [49]	0.050	28.62	0.844	0.107	0.044
Ruyi-Mini-7B [36]	0.047	28.01	0.831	0.133	0.043
SVD [2]	<u>0.032</u>	29.93	0.890	0.079	0.038
Real Video	0.022	33.60	0.925	0.074	0.026

Table 3. **Comparison of flow warping scores (FWS) with MEt3R.** We report the results on multi-view and video generation methods, with FWS variants including PSNR, SSIM, LPIPS, and RMSE.

Projection matrix. Figure 15 shows a side-by-side comparison of different projections we obtain using 1): fixed focal length and 2): Adjusting focal length based on the scale of canonical point map. We compute the canonical point map $\mathbf{X}_{\text{canon}}$ as the weighted sum of the point maps pair \mathbf{X}_i and \mathbf{X}_{i+1} using their corresponding confidences \mathbf{C}_i and \mathbf{C}_{i+1} from DUST3R [42] as,

$$\mathbf{X}_{\text{canon}} = \frac{\mathbf{C}_i \odot \mathbf{X}_i + \mathbf{C}_{i+1} \odot \mathbf{X}_{i+1}}{\mathbf{C}_i + \mathbf{C}_{i+1}} \quad (7)$$

Then, we extract the x , y , and z coordinate maps from $\mathbf{X}_{\text{canon}}$ as $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \in \mathbb{R}^{H \times W}$. Moreover, DUST3R already implements this in their codebase, which we incorporate in MEt3R as shown in Alg. 1. The computed focal length f_x and f_y , along with the principal point offsets c_x and c_y , are used to form the projection matrix.

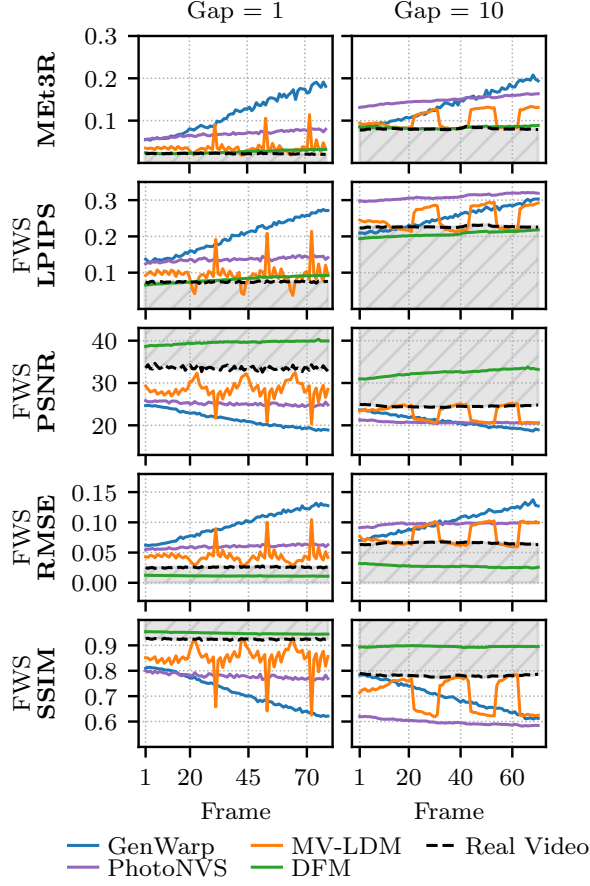


Figure 14. **MET3R comparison to existing Flow Warping Score (FWS).** For a gap (frame distance) of 10 between the image pair, MET3R is more robust and does not violate the lower bound, unlike FWS.

Algorithm 1 Computing focal length given 2D grid of pixel positions and 3D canonical point maps

Input: 2D pixel position $\mathbf{U}, \mathbf{V} \in \mathbb{R}^{H \times W}$, 3D position $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \in \mathbb{R}^{H \times W}$

Output: f_x, f_y
 1: $\mathbf{Q}_x = \frac{\mathbf{U} \odot \mathbf{Z}}{\mathbf{X}}$ $\triangleright \odot$ is the Hadamard Product
 2: $\mathbf{Q}_y = \frac{\mathbf{V} \odot \mathbf{Z}}{\mathbf{Y}}$
 3: $f_x = \text{median}(\mathbf{Q}_x)$ \triangleright Across spatial dimension
 4: $f_y = \text{median}(\mathbf{Q}_y)$

Overlap mask. We normalize MET3R with an overlap mask \mathbf{M} as formulated in Eq. 4 which is a crucial step. During rasterization, we set the background values to a large negative value η for each channel and subsequently build the mask using the background values for each projected



Figure 15. **Fixed vs. adjusted projection matrix.** With fixed focal length, the projection area varies across different scales of DUST3R [42] point maps. We automatically adjust the focal length for each example pair to allow maximal projection and, therefore, more pixels for evaluating feature similarity.

view, i.e.,

$$m_{ij}^k = \begin{cases} 0 & \text{if } p_{ij}^k = \eta \\ 1 & \text{otherwise} \end{cases} \quad (8)$$

where $m_{ij}^k = [\mathbf{M}^k]_{ij}$ is the mask for k^{th} view, $p_{ij}^k = [\mathbf{P}^k]_{ij}$ are the pixel values after projection and rasterization. We set $\eta = -10000$, and we perform pixel-wise multiplication of both masks \mathbf{M}^i and \mathbf{M}^{i+1} to get the overlap mask \mathbf{M} :

$$\mathbf{M} = \mathbf{M}^i \odot \mathbf{M}^{i+1} \quad (9)$$

Figure 16 shows MET3R without normalizing against the overlap mask \mathbf{M} in Eq. 4. Instead, we take the average of the similarity scores for all pixels. Compared to MET3R (c.f. Fig. 4), the lower bound gets significantly larger with a large offset, while DFM [38] gets worse than all other baselines. Meanwhile, PhotoNVS [48] gets almost similar to GenWarp [32]. This contradicts both the theoretical expectation and the visual judgment about the 3D consistency of the baselines. In addition, the standard deviations for all baselines are large and correspond to noisy scores for individual image pairs across the test sequences. However, some key features, such as spikes from anchor-to-anchor transitions in MV-LDM and the gradual increase in MET3R due to decreasing 3D consistency, are still visible.

D. Additional Details on Multi-View Generation Models

In the following, we present additional details on the multi-view generation baselines.

GenWarp. GenWarp [32] employs a two-step approach, i.e., project and in-paint. With a monocular depth estimator,

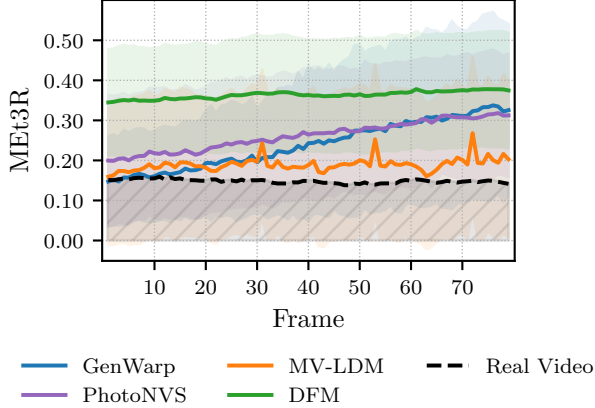


Figure 16. **MEt3R without overlap mask.** Per-image-pair MEt3R without normalizing against the overlap mask. Under this setting, DFM [38] is worse than all other baselines in 3D consistency, even though it has a strong inductive bias, which forces its results to be 3D consistent at the expense of blur. Whereas PhotoNVS [48] and GenWarp [32] are similar, both of which increase gradually, whereas MV-LDM stays relatively low with visible spikes due to anchor-to-anchor transitions.

it predicts depth maps for the input image and un-projects the RGB in 3D space. The 3D points are rendered onto a target view, followed by inpainting with an image-to-image diffusion model. GenWarp generates only one view at a time. For every novel view, we condition the model on the fixed input view for every novel view, as an autoregressive approach diverges very quickly due to error accumulation.

PhotoNVS. Just like GenWarp, PhotoNVS [48] also generates a single view at a time given a conditioning image. However, by employing a score-based diffusion UNet architecture for both views with cross-view attention in-between, it can always condition on the last generated frame in an autoregressive fashion, improving multi-view consistency across a full sequence.

DFM. DFM [38] incorporates a neural radiance field into the architecture of an image diffusion model such that novel views are 3D consistent by design. By employing pixel-NeRF [47], DFM generates the 3D representation given a set of conditioning views. Starting from a single view, it generates an extrapolated target view that acts as additional conditioning in all subsequent sampling steps.

E. Runtime

In Tab. 4, we compare the runtimes of the evaluated methods for generating 80 frames of a video sequence on an NVIDIA RTX4090 GPU with 24GB VRAM. GenWarp

	GenWarp	PhotoNVS	DFM	MV-LDM
Runtime (s)	70	7840	1020	<u>100</u>

Table 4. **Runtime comparison.** We report the runtime in seconds for all the baselines for generating a full video sequence comprising 80 frames. MV-LDM and GenWarp [32] achieve the fastest sampling, followed by DFM [38] and then PhotoNVS [48].

achieves the fastest sampling time, as high-quality but inconsistent novel views can already be obtained with 20 DDIM steps. Although MV-LDM generates multiple views at a time, which improves 3D consistency and uses 70 DDIM steps to achieve good image quality, it is only slightly slower than the single-view generation of GenWarp. Both DFM and PhotoNVS are an order of magnitude slower due to slow volumetric NeRF rendering and many denoising steps, respectively. Our proposed metric MEt3R can be evaluated in only *95ms* per image pair.



Figure 17. **Examples of generated multi-views and videos.** From Top → Down is the increasing frame number with columns for each method. Note that the first row is the input image, the first four columns are the results of multi-view generation models with explicit camera control, whereas the last three columns are generated videos from video diffusion models without any camera control.

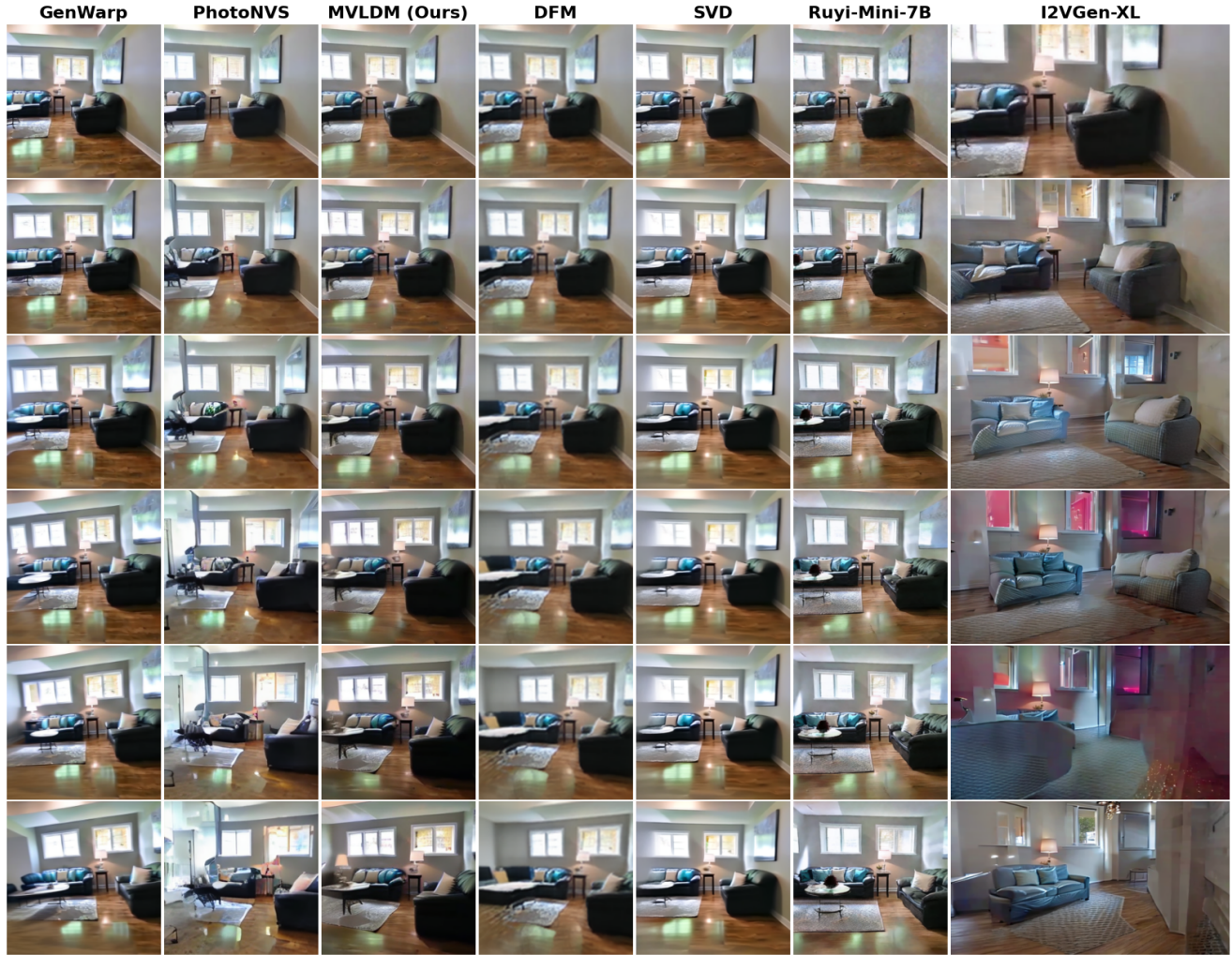


Figure 18. **Examples of generated multi-views and videos.** From Top \rightarrow Down is the increasing frame number with columns for each method. Note that the first row is the input image, the first four columns are the results of multi-view generation models with explicit camera control, whereas the last three columns are generated videos from video diffusion models without any camera control.

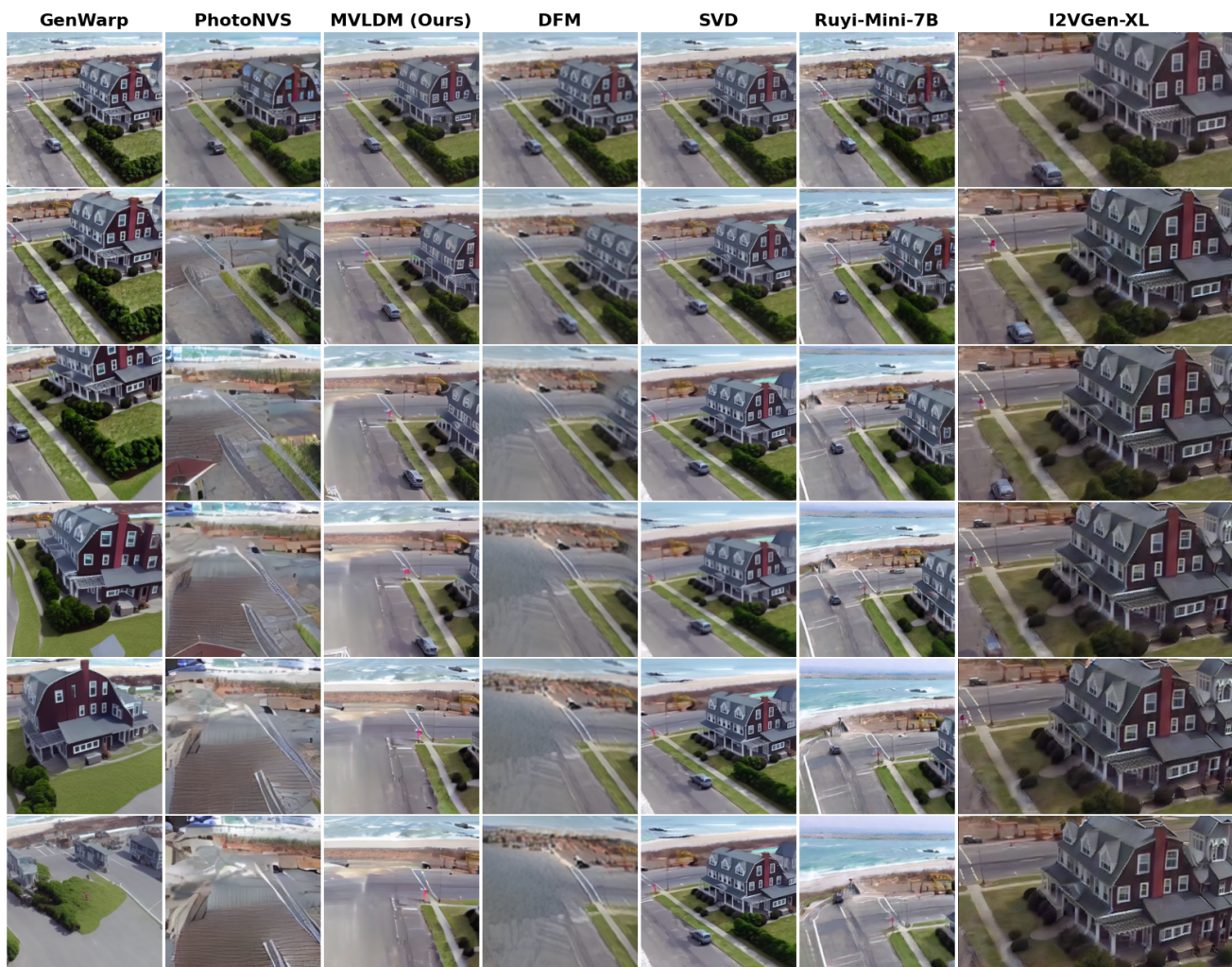
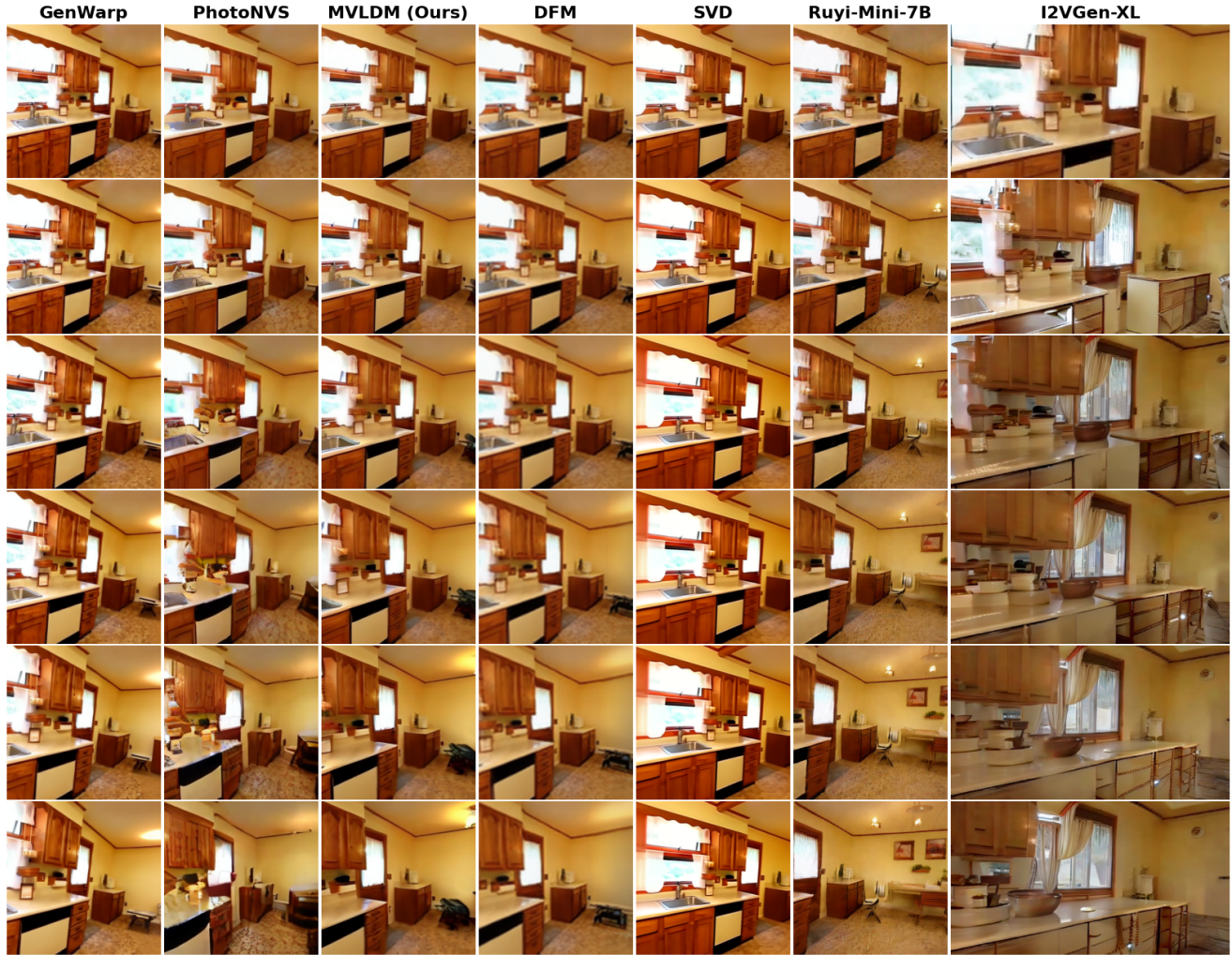


Figure 19. **Examples of generated multi-views and videos.** From Top \rightarrow Down is the increasing frame number with columns for each method. Note that the first row is the input image, the first four columns are the results of multi-view generation models with explicit camera control, whereas the last three columns are generated videos from video diffusion models without any camera control.



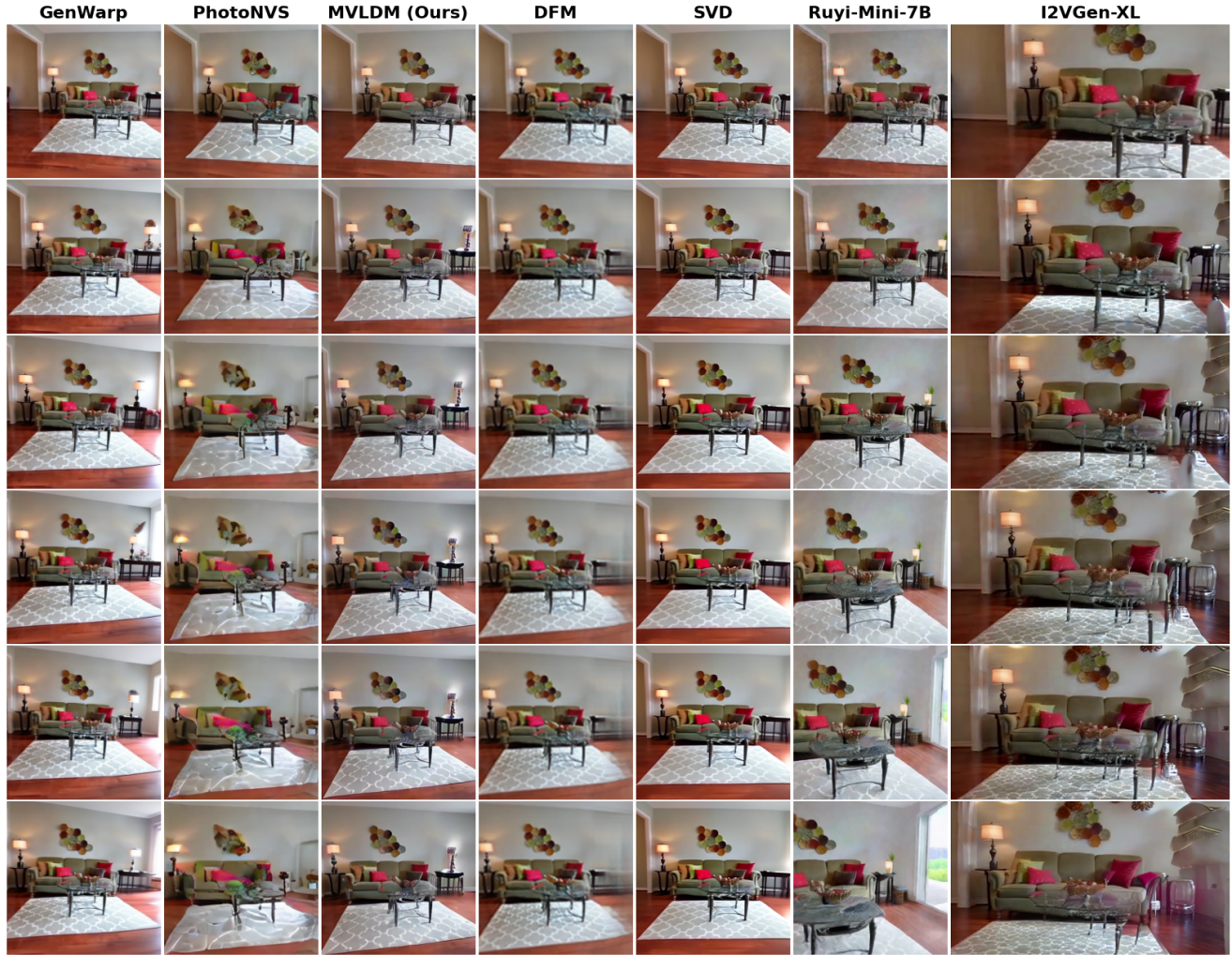


Figure 21. **Examples of generated multi-views and videos.** From Top → Down is the increasing frame number with columns for each method. Note that the first row is the input image, the first four columns are the results of multi-view generation models with explicit camera control, whereas the last three columns are generated videos from video diffusion models without any camera control.



Figure 22. **Qualitative results on GSO.** A 360° rendering of the GSO object *Elephant*.



Figure 23. **Qualitative results on GSO.** A 360° rendering of the GSO object *Alarm*.



Figure 24. **Qualitative results on GSO.** A 360° rendering of the GSO object *Blocks*.



Figure 25. **Qualitative results on GSO.** A 360° rendering of the GSO object *Cream*.