Scaling4D: Pushing the Frontier of Video Novel View Synthesis through Large-Scale Monocular Videos

Supplementary Material

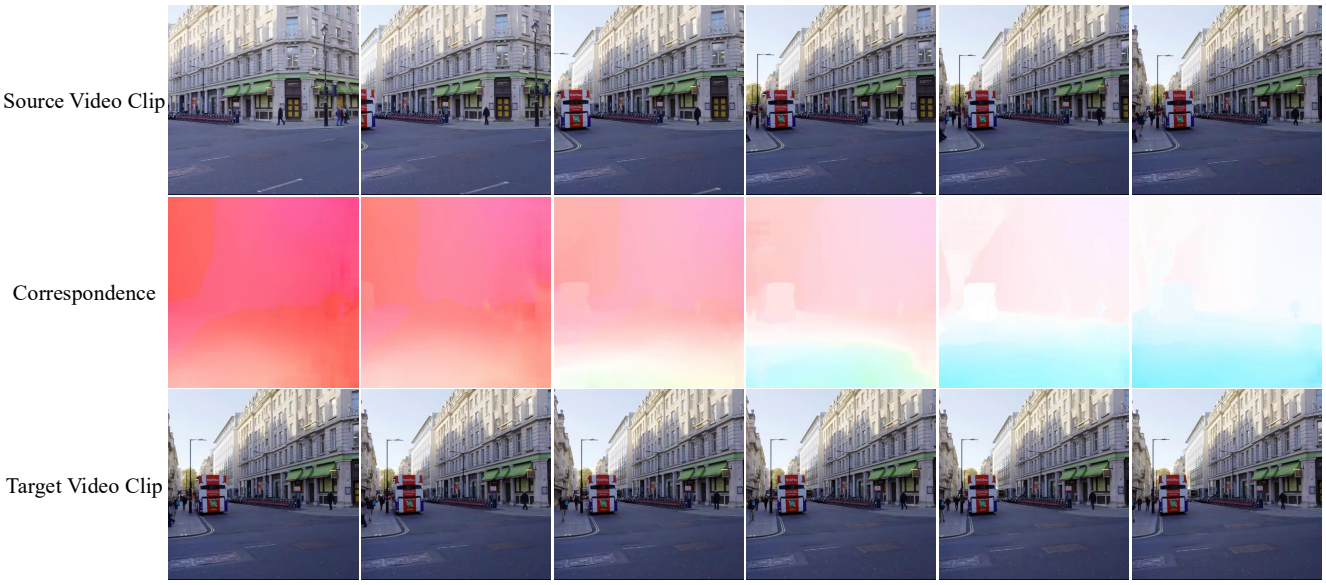

Figure 1. An example of training data construction.

This supplementary material provides additional details for Scaling4D, including the construction pipeline of our training dataset, qualitative evaluations for both the ablation studies and data scaling, alongside further discussions on architectural choices and system robustness.

A. Training Data Construction

Our training data is sourced from large-scale, in-the-wild monocular videos. For each video, we construct a training sample comprising a source video clip ($\mathbf{I}^s$), a target video clip ($\mathbf{I}^t$), and the correspondence ($\mathbf{C}^r$) between them, as shown in Fig. 1. The detailed procedure is outlined below.

A.1. Sampling Source and Target Video Clips

Given a monocular video from our dataset, we first extract its frames to form a tensor of shape $[T, C, H, W]$, where T is the total number of frames. From this sequence, we independently sample two temporal clips, $\mathbf{I}^s$ and $\mathbf{I}^t$, both with a fixed length N. This sampling procedure is specifically designed to generate pairs of clips that are temporally proximate yet varied, involving the following steps:

1. **Select Center Frames:** We first define a valid range for selecting the center frames of our clips, leaving adequate margins at the beginning and end of the video to allow for bidirectional sequence extension. Two center-frame indices, m_s for the source and m_t for the target, are then randomly chosen from this valid range. To ensure temporal proximity, m_t is selected from a local window centered around m_s.
2. **Form Initial Continuous Sequences:** For each center frame (m_s and m_t), we form an initial continuous sequence of frames. The length of this sequence, L_{base}, is randomly determined to be within a specific proportion of the final target sequence length N. This proportion is bounded by two hyperparameters:
   - `base_length_ratio_min`: The minimum ratio of the base sequence length to the target length N (e.g., 0.75).
   - `base_length_ratio_max`: The maximum ratio of the base sequence length to the target length N (e.g., 1.0).

   Consequently, the base sequence length is bounded by $L_{base} \in [\lceil N \cdot \texttt{ratio}_{\min} \rceil, \lfloor N \cdot \texttt{ratio}_{\max} \rfloor]$. The sequence is formed by extending outwards from the center frame, while strictly ensuring the frame indices remain within the video's valid boundaries $[0, T-1]$.
3. **Pad to Target Length:** The initial sequence, with length L_{base}, is typically shorter than the required final length N. To bridge this gap of $N - L_{base}$ frames, we randomly sample *with replacement* from the initial sequence and

append these additional frames to it.

4. **Finalize Clips:** Finally, all selected frame indices for both the source and target clips are sorted in ascending order to maintain strict chronological consistency within each clip. These sorted index lists are then used to gather the corresponding frames from the original full video tensor, yielding the final source clip $\mathbf{I}^s \in \mathbb{R}^{N\times C\times H\times W}$ and target clip $\mathbf{I}^t \in \mathbb{R}^{N\times C\times H\times W}$.

This strategy effectively leverages short, continuous segments from a single monocular video to create diverse training pairs, simulating natural variations in viewpoint or time while strictly maintaining content coherence.

A.2. Computing Correspondence with Optical Flow

With the pair of sampled video clips, $\mathbf{I}^s$ and $\mathbf{I}^t$, we compute the dense correspondence $\mathbf{C}^r$ between them. This correspondence serves as the foundational control signal for our model. The computation process is executed as follows:

1. **Optical Flow Estimation:** We employ a pre-trained optical flow model, RAFT [2], which is highly effective for estimating dense motion between pairs of images. Specifically, the model takes the source clip $\mathbf{I}^s$ and target clip $\mathbf{I}^t$ as inputs.
2. **Generate Flow Fields:** The RAFT model processes the clips frame-by-frame, comparing each frame i from $\mathbf{I}^s$ with the corresponding frame i from $\mathbf{I}^t$. For each pair of frames, it outputs a 2D flow field, resulting in a sequence of flow fields for all N frames.
3. **Construct Correspondence Tensor:** These flow fields are stacked to form the final correspondence tensor $\mathbf{C}^r \in \mathbb{R}^{N\times 2\times H\times W}$. This tensor encapsulates the pixel-level spatial mapping from the source video clip to the target video clip. Specifically, for each frame i, the tensor $\mathbf{C}^r_i$ provides the (u, v) displacement vectors required to warp pixels from $\mathbf{I}^s_i$ to their corresponding physical locations in $\mathbf{I}^t_i$.

The computed correspondence $\mathbf{C}^r$, alongside the source clip $\mathbf{I}^s$ and its warped version $\mathbf{I}^r$ (obtained by backward-warping $\mathbf{I}^s$ using $\mathbf{C}^r$), forms a comprehensive control signal sequence that guides our video generation model during training.

A.3. Filtering Unreliable Correspondence

Optical flow estimation can occasionally degrade under challenging in-the-wild conditions, such as extreme non-rigid motion or transient dynamics. To ensure the high quality of our training data, we employ cycle-flow consistency to filter out inaccurate correspondence pairs.

Let $C^r_{t\to t+1}(\mathbf{x})$ and $C^r_{t+1\to t}(\mathbf{x})$ denote the forward and backward flows between consecutive frames (I_t, I_{t+1}), respectively. We compute the cycle residual at pixel location $\mathbf{x}$ as:

$$\mathbf{r}_t(\mathbf{x}) = C^r_{t\to t+1}(\mathbf{x}) + C^r_{t+1\to t}\big(\mathbf{x} + C^r_{t\to t+1}(\mathbf{x})\big), \quad (1)$$

where $C^r_{t+1\to t}(\cdot)$ is evaluated at generally non-integer coordinates via bilinear sampling.

We score each video based on the average cycle residual over time and space:

$$S_{\text{cycle}} = \frac{1}{T-1}\sum_{t=1}^{T-1} \mathbb{E}_{\mathbf{x}}\left[\|\mathbf{r}_t(\mathbf{x})\|_2^2\right], \quad (2)$$

where $\mathbb{E}_{\mathbf{x}}[\cdot]$ denotes the spatial average. Videos with a cycle consistency score exceeding a predefined empirical threshold τ are discarded. Furthermore, to proactively enhance the model's robustness to complex real-world conditions, we intentionally construct challenging scenarios within our synthetic data pipeline. This exposes the model to diverse, hard-to-estimate motion patterns during training, ensuring stable performance even when inference correspondences are imperfect.

B. Further Discussions on Method Design

B.1. VNVS Block Design and Alternatives

Our proposed VNVS Block is formulated in the style of MM-DiT conditioning. It functions as a lightweight, shape-preserving residual modulation that maintains the base network's core interface, maximizes the reuse of pretrained weights, and completely avoids the introduction of computationally heavy cross-attention layers.

During the architectural exploration phase, we systematically compared alternative conditioning mechanisms under identical training configurations. Empirical observations indicate that additive fusion yields significantly faster early-stage convergence than channel concatenation (e.g., reaching 28.4% of the target training loss at step 200 in our preliminary experiments).

We also explored ControlNet-style conditioning structures. However, duplicating the diffusion backbone substantially increases the parameter count (typically by $\sim 2\times$) and incurs prohibitive memory overhead. This makes it computationally impractical for our target high-resolution spatiotemporal modeling tasks. Consequently, we adopted the additive residual modulation, a design choice that effectively injects scalable correspondence control while maintaining an exceptionally minimal and efficient architectural footprint.

B.2. Details of Baseline Re-implementations

To ensure a rigorous and standardized evaluation protocol under strictly identical test environments in our main text, we carefully re-implemented the representative baselines, GEN3C [1] and TrajectoryCrafter [3]. Table 1 presents a quantitative comparison between these re-implementations and their official open-source checkpoints (denoted with *) on the single-view test set.

Method	FID ↓	FVD ↓	CLIP-T ↑	CLIP-F ↑	CLIP-V ↑	RotErr ↓	TransErr ↓
GEN3C [1]	69.35	442.70	27.43	98.49	90.14	6.98	299.68
GEN3C* [1]	77.12	483.62	27.01	98.09	89.52	8.05	406.51
TrajectoryCrafter [3]	68.94	425.20	27.35	98.37	90.41	6.65	320.38
TrajectoryCrafter* [3]	73.05	470.37	27.12	98.43	89.95	7.40	454.09

Table 1. Quantitative evaluation of the baseline implementations. We report the results of both our re-implemented versions and the official open-source model weights (*). The strong performance of our re-implementations serves as a robust foundation for fair comparisons in the main text.

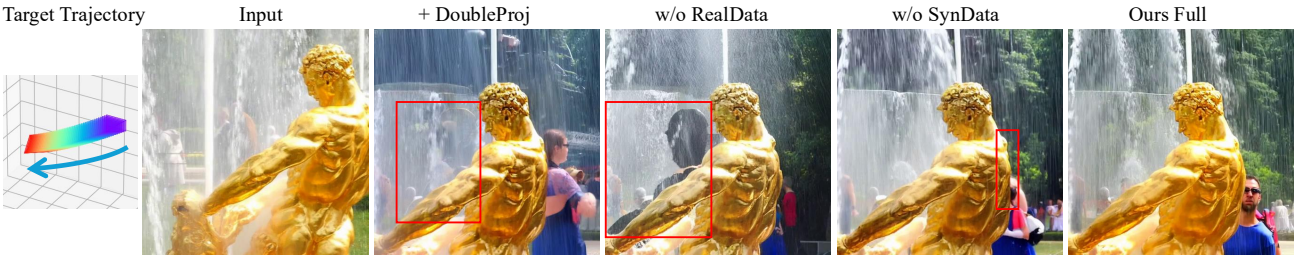

Figure 2. The qualitative evaluations of our ablation studies.

As the results demonstrate, our re-implemented models perform comparably to, or even favorably against, the official weights. For example, our GEN3C implementation achieves a superior FVD of 442.70, and our TrajectoryCrafter implementation achieves a refined RotErr of 6.65. By employing these highly optimized versions, we establish a solid and robust baseline comparison framework for evaluating Scaling4D.

C. Qualitative Evaluations of Ablation Study

In this section, we provide a qualitative evaluation of our ablation study, with visual results presented in Fig. 2. This analysis offers intuitive insights into the specific contributions of different data components to the overall synthesis performance.

Impact of Double Reprojection Data. As shown in the third column of Fig. 2, training the model with additional double reprojection data (Ours + DoubleProj) introduces noticeable artifacts in regions that are originally occluded in the input view. For example, the area on the ground behind the statue exhibits unnatural textural distortions. This visual observation is highly consistent with our quantitative findings: while pose precision might receive a marginal boost, it comes at the steep cost of overall visual quality degradation.

Impact of Real Data. The fourth column (Ours w/o RealData) demonstrates the model's performance when trained exclusively on synthetic data. The rendered novel view of the fountain scene clearly lacks physical realism, particularly in capturing the complex dynamics of the splashing water. This limitation arises because our current synthetic dataset lacks fine-grained simulations for such intricate fluid movements, underscoring that large-scale real-world data is indispensable for achieving high-fidelity generation and covering the vast diversity of natural phenomena.

Impact of Synthetic Data. In the fifth column (Ours w/o SynData), where the model is trained entirely without synthetic geometric priors, we observe a slight but perceptible decline in camera pose precision. A close inspection of the bronze statue's shoulder reveals subtle misalignments and structural distortions compared to the pristine outputs of our full model. This visual evidence validates that synthetic data effectively enforces strict camera control precision, leading to geometrically highly consistent novel views.

D. Qualitative Evaluations of Scalability

In addition to the quantitative scaling laws discussed in the main text, we provide a qualitative evaluation to intuitively demonstrate the profound impact of data volume on the model's generative capabilities. As illustrated in Fig. 3, there is a clear, monotonic improvement in the structural and textural quality of the generated novel views as the training data volume increases.

When the model is trained on a highly constrained dataset (e.g., 1k samples), the resulting synthesis suffers from severe artifacts and a complete lack of structural integrity. For instance, the generated novel view of the building is blurry, heavily distorted, and fails to reconstruct coherent geometric details. However, as the data volume aggressively scales up from 1k to 3000k, we observe a dra-

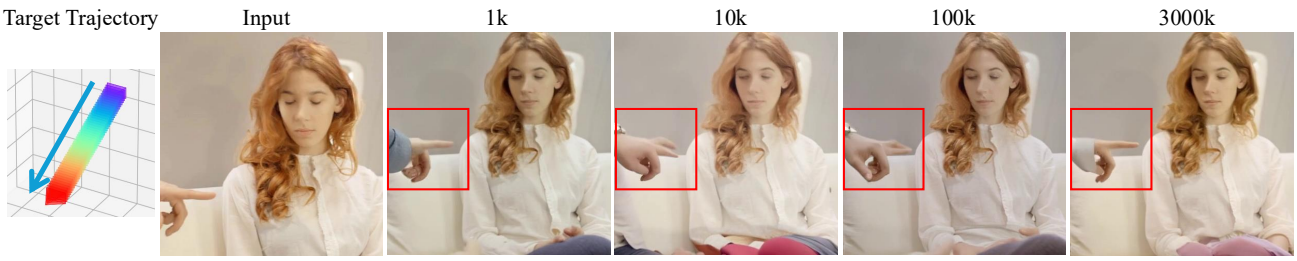

Figure 3. The qualitative evaluations of scalability on data volume.

Figure 4. Qualitative analysis of generation robustness under challenging depth estimation conditions, including low texture, harsh lighting, sensor noise, dynamic objects, and occlusion boundary artifacts.

matic enhancement in both visual fidelity and structural accuracy.

With 10k samples, the basic building layout begins to emerge, though it remains visually coarse. At 100k samples, the model produces a significantly sharper and highly recognizable image with plausible facade details. Finally, at 3000k samples, the model yields the most realistic and geometrically precise result, accurately rendering the building's complex architecture, lighting, and textures. This visual progression strongly corroborates our quantitative findings, confirming that Scaling4D possesses immense capacity to leverage massive datasets for superior generative performance.

E. Analysis of Robustness to Depth Errors

To comprehensively evaluate the model's behavior under sub-optimal input conditions, Fig. 4 visualizes the synthesis performance of Scaling4D across typical failure modes of off-the-shelf depth estimators. These challenging stress-test scenarios include regions with low texture, harsh lighting conditions, sensor noise, dynamic foreground objects, and occlusion boundary artifacts.

The visual results clearly indicate that the generated novel views remain structurally stable and visually pleasing despite these severe input imperfections. We attribute this strong robustness directly to our flow-based training paradigm. By naturally exposing the network to diverse optical flow misalignments and boundary noise during the large-scale learning process, the model implicitly learns to accommodate and correct noisy spatial control signals. Consequently, it achieves highly resilient and consistent view synthesis during inference, effectively decoupling generation quality from the strict dependency on flawless depth maps.

References

[1] Xuanchi Ren, Tianchang Shen, Jiahui Huang, Huan Ling, Yifan Lu, Merlin Nimier-David, Thomas Müller, Alexander Keller, Sanja Fidler, and Jun Gao. Gen3c: 3d-informed world-consistent video generation with precise camera control. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025. 2, 3

[2] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision (ECCV)*, pages 402–419. Springer, 2020. 2

[3] Mark Yu, Wenbo Hu, Jinbo Xing, and Ying Shan. Trajectorycrafter: Redirecting camera trajectory for monocular videos via diffusion models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 100–111, 2025. 2, 3