DANDELION: TOWARDS REAL-TIME AND CONSISTENT LONG VIDEO DEPTH ESTIMATION

Anonymous authors
Paper under double-blind review

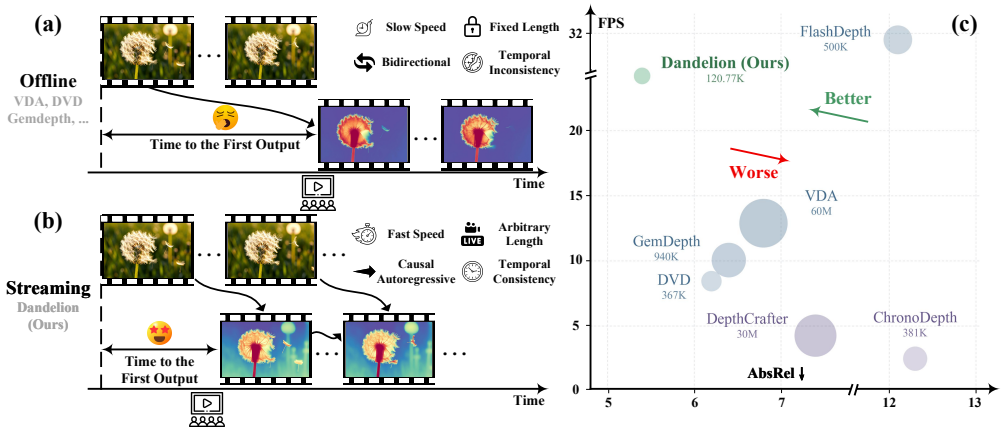

Figure 1: Compared with the previous offline method (**a**), our streaming framework, Dandelion (**b**), substantially reduces first-prediction latency while achieving real-time inference and superior depth accuracy using less depth training data (indicated by circle size) (**c**).

ABSTRACT

Recently, video depth estimation methods have successfully exploited the priors of pre-trained video generative models to enhance geometric perception, but the costly inference limits low-latency deployment. In this paper, we present **Dandelion**, the first generative streaming video depth estimation model, which reduces first-prediction latency and enables real-time estimation. However, directly adapting traditional frame-wise streaming strategies remains suboptimal due to the repeated per-frame inference and the lack of bidirectional temporal context. To address these issues, through systematic analysis, we introduce the streaming chunk strategy and tailor a dedicated chunk length to achieve both high depth accuracy and low-latency inference. Moreover, a stable reference is indispensable for temporal consistency. To achieve this, we design a two-stage training protocol, which first establishes a geometric anchor from GT depth history and extends its robustness throughout autoregressive rollout. To further accelerate inference, we recognize that details in distant depth history are less critical to current estimates. Thus, we design Multi-scale History Attention to compress distant context. With these advances, Dandelion achieves both state-of-the-art (SoTA) zero-shot performance and superior efficiency. Using only 0.2% of the depth supervision used by prior methods (VDA), Dandelion achieves a 12.9% performance gain and supports real-time depth estimation over long videos at 24 FPS.

1 INTRODUCTION

Depth estimation is fundamental to understanding dynamic 3D scenes, enabling a broad range of applications (Wang et al. (2026a); Wang et al. (2026c)), including 3D editing (Wang et al. (2026b)) and robotics. Although recent monocular depth estimation methods have achieved remarkable progress, extending them from images to videos remains a challenge (Xu et al. (2025b)).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Existing single-frame methods treat each frame in isolation. Directly extending these methods (Wang et al. (2024);Xu et al. (2025a)) to videos can lead to flickering and scale ambiguity. To alleviate these issues, recent approaches incorporate dedicated temporal modules (Chou et al. (2025)). However, they typically capture interactions within short video clips, limiting geometric consistency across clips. For long-video inference, they have to employ overlapping windows with affine alignment (Zhang et al. (2026)). Although these strategies can improve temporal coherence, they not only introduce cumulative alignment errors but also incur additional computational overhead. Moreover, the aligned depth output is released only after the entire video has been processed, resulting in substantial First-Prediction Latency (FPL), as shown in Fig. 1 (a).

To address the above issue, as shown in Fig. 1 (b), an intuitive direction is to adopt a streaming framework that simultaneously processes incoming observations and produces predictions. We formulate the task as causal autoregressive prediction, where each estimate is conditioned on the current observation and previously available context. However, traditional streaming models (Chou et al. (2025); Lan et al. (2026)) process a video frame by frame. Through systematic analysis, we find that they not only struggle to utilize local bidirectional context, resulting in performance degradation, but also exhibit limited efficiency due to a huge Number of Function Evaluations (NFE, one evaluation per frame). In this paper, we introduce streaming chunks to adapt video depth estimation to streaming inference, enabling bidirectional temporal modeling within each chunk and establishing causal dependencies across previous chunks. We have explored the effect of streaming chunk length and finally adopted a moderate chunk length to achieve the best performance and acceptable latency.

Moreover, successive depth predictions may exhibit scale shifts or discontinuities without a stable geometric anchor. To build this anchor, we develop a two-stage training protocol. In the first stage, we take GT depth as history to provide a reliable geometric context, where the model can learn a stable reference. To extend the robustness of this global geometric anchor, we replace the GT history with predicted depth results to support autoregressive depth estimation in the second stage.

To further improve inference speed, we explore the temporal propagation of geometric information. We find that distant depth predictions primarily provide depth trends for current estimates, while contributing relatively little to their fine-grained structure. This suggests that distant depth history can be compressed to reduce computational overhead. We therefore design Multi-scale History Attention to preserve recent geometric details while representing distant history at progressively coarser scales. This further reduces the high cost of history processing and accelerates inference.

Together, these improvements enable the effective transfer of rich spatio-temporal priors from pre-trained video generative models to streaming video depth estimation, yielding high-fidelity predictions (a 12.9% performance gain versus previous SoTA) with sharp structural details, enabling real-time inference (24 FPS) without requiring extra post-processing, and maintaining geometric consistency. Above all, as shown in Fig. 1, we denote the improved real-time video depth estimation model as Dandelion. Our main contributions are summarized as follows:

- We present Dandelion, the first generative streaming video depth estimation model based on causal autoregressive prediction. Dandelion enables real-time inference with low first-prediction latency while maintaining temporal consistency over long videos.
- To address the overhead of repeated per-frame inference and limited local temporal context, we introduce streaming chunks that combine bidirectional modeling within chunks with causal dependencies across chunks. A tailored chunk length enables accurate depth estimation with low latency.
- To provide a stable geometric reference for temporal consistency, we design a two-stage training protocol that establishes a geometric anchor and extends its guidance robustness throughout autoregressive rollout.
- Motivated by our observations of geometric information propagation, we design Multi-scale History Attention. It compresses the predicted depth context to reduce redundant history processing and accelerate inference.

2 Related Works

Monocular Depth Estimation aims to recover scene geometry from a single image. Recent advances have been driven by two complementary directions: scaling up training data and leveraging

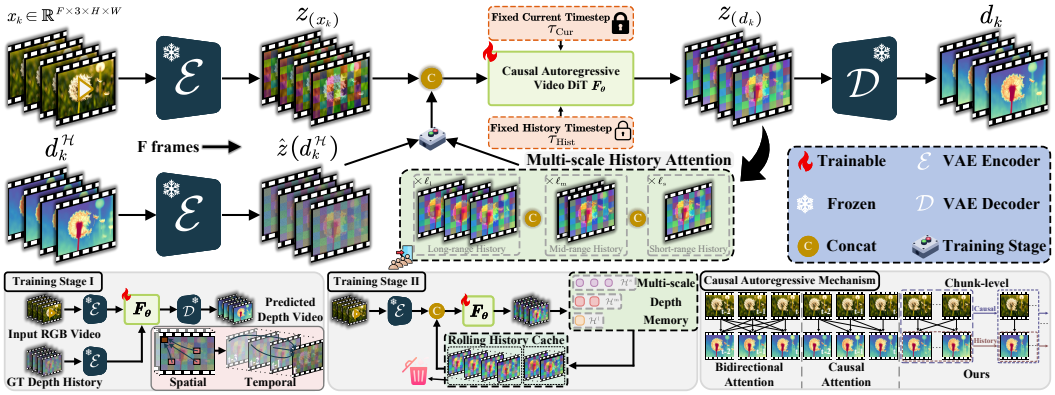

Figure 2: **Training Pipeline of Dandelion.** Sec. 3.1 details the pipeline. Dandelion is built upon a two-stage training protocol and Multi-scale History Attention, which are detailed in Sec. 3.3 and Sec. 3.4.

large-scale pretrained foundation models. Scaling-based approaches, exemplified by Depth Anything V2 (Yang et al. (2024)), significantly improved zero-shot generalization through large-scale labeled and unlabeled supervision. Inspired by the success of generative foundation models, Jasmine (Wang et al. (2026a)), Lotus (He et al. (2025)), and FE2E (Wang et al. (2026c)) exploit the rich visual priors to restore fine-grained structural details.

Video Depth Estimation methods can be broadly categorized into discriminative and generative paradigms. Discriminative approaches directly regress depth from RGB observations and improve temporal consistency by introducing lightweight temporal modules. VDA (Chen et al. (2025)) and STream3R (Lan et al. (2026)) achieve efficient inference while maintaining competitive performance. More recently, generative approaches repurpose pretrained video diffusion models to exploit richer spatio-temporal priors. DVD (Zhang et al. (2026)) adapts pretrained video diffusion models to single-pass depth regressors.

Online Video Depth Estimation aims to produce geometry predictions under strict latency constraints. Existing approaches primarily rely on lightweight architectures to satisfy these constraints. For example, STream3R (Lan et al. (2026)) performs causal inference for 3D reconstruction, while FlashDepth (Chou et al. (2025)) adapts single-image depth models to support real-time streaming at high resolution. Unlike these discriminative approaches, Dandelion reformulates video depth estimation as a causal autoregressive prediction task, achieving low-latency and real-time inference.

3 METHOD

3.1 PRELIMINARIES

Problem Formulation. As shown in Fig. 2 upper part, given an RGB video stream $x_k \in \mathbb{R}^{F\times 3\times H\times W}$ and corresponding depth GT $d_k^{\mathcal{H}} \in \mathbb{R}^{F\times 1\times H\times W}$, where F denotes the frame count, we adopt a frozen variational autoencoder (VAE) to encode RGB and depth videos into the latent space:

$$z_{(x_k)} = \mathcal{E}(x_k) \in \mathbb{R}^{f\times c\times h\times w}, \quad z_{d_k^{\mathcal{H}}} = \mathcal{E}(d_k^{\mathcal{H}}) \in \mathbb{R}^{f\times c\times h\times w}. \tag{1}$$

where f denotes the latent temporal length, and c, h, and w denote the channel and downsampled spatial dimensions. Following Karras et al. (2022), in the first training stage, we sample Gaussian noise ϵ and perturb the $d_k^{\mathcal{H}}$ using the predefined noise scale σ_t as detailed in Wan 2.1 (Wan et al. (2025)):

$$\hat{z}(d_k^{\mathcal{H}}) = z(d_k^{\mathcal{H}}) + \sigma_t^2\epsilon, \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{2}$$

After that, we concatenate the RGB latent sequence $z_{(x_k)}$ with $\hat{z}(d_k^{\mathcal{H}})$ and feed them into a pretrained video diffusion model F_θ to recover the geometric structure in depth latent $z_{(d_k)}$. Inspired by DVD (Zhang et al. (2026)), we improve upon their fixed-timestep strategy by incorporating the current observation τ_{Cur} and depth history τ_{Hist} into F_θ to enable single-step depth prediction. The

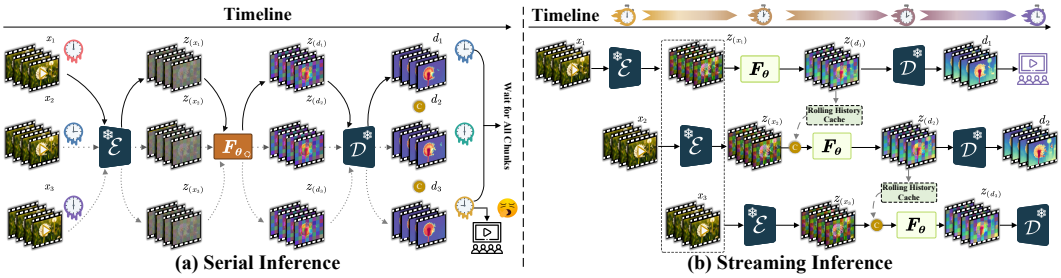

Figure 3: **Comparison of Inference pipelines.** Clock orientations indicate when each clip is processed, and the ( ) denote progressively later inputs. (**a**) Serial inference processes chunks sequentially along a single schedule from the upper-left clock to the lower-right clock (...). (**b**) The 3 parallel rows show the streaming inference that concurrently processes successive chunks. The dashed box shows that Dandelion can process $(z(x_1), \mathcal{E}(x_2), x_3)$ at the same time ( ). ( ) indicates that the first prediction is available to the user.

depth latent $z_{(d_k)}$ is then processed by multi-scale history attention in the second training stage, as detailed in Sec. 3.4. Finally, the depth results d_k are reconstructed via the frozen VAE decoder $d_k = \mathcal{D}(z_{(d_k)})$.

Causal Attention vs. Bidirectional attention. As shown in Fig. 2 lower-right part, bidirectional attention allows each frame to interact with all frames in the input sequence. However, it requires the complete input sequence to be available in advance, thereby restricting conventional video models to fixed-length videos. To support variable-length inference, previous work (Chen et al. (2024)) enforces causal dependencies through attention masks. It restricts each frame to interacting only with itself and preceding frames, enabling autoregressive inference as new frames arrive. The visibility masks for these two attention mechanisms can be formulated as

$$\mathrm{Att}^{\mathrm{bi}}_{\{c_p,c_q\}} = 1 \qquad \mathrm{Att}^{\mathrm{causal}}_{\{c_p,c_q\}} = \begin{cases} 1, & \text{if} \quad c_p \le c_q, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

where c_p and c_q denote the frame indices. It reduces the computational complexity (Huang et al. (2025)) to accelerate inference.

3.2 Streaming Perception Framework

Serial VS. Streaming Inference. Conventional video depth estimation can be typically categorized into two schemes: offline methods can only predict an entire video but are restricted to a fixed sequence length. The serial approach, as illustrated in Fig. 3 (a), processes video clips serially through the inference pipeline. However, the aligned depth video is released only after the entire sequence has been processed, leading to substantial first-prediction latency.

As illustrated in Fig. 3 (b), we perform a streaming pipeline, which means a module does not wait for the current clip to complete the remaining pipeline before accepting the next one. Once x_1 has been encoded, VAE immediately starts encoding x_2, while F_θ starts to process $z_{(x_1)}$. Without any expensive post-processing, the depth video clips can be directly released, which substantially reduces the latency.

Let $t_\mathcal{E}$, t_{F_θ}, and $t_\mathcal{D}$ denote the execution times of the encoder, Video DiT, and decoder, respectively. The runtime of the previous serial method, which processes K clips sequentially, can be approximated as:

$$T_{\text{serial}} \approx K\,(t_\mathcal{E} + t_{F_\theta} + t_\mathcal{D}) + T_{\text{overlap}}, \tag{4}$$

where T_{overlap} denotes the time required for subsequent alignments. In contrast, Dandelion requires

$$T_{\text{stream}} \approx t_\mathcal{E} + t_{F_\theta} + t_\mathcal{D} + (K-1)\max\left(t_\mathcal{E}, t_{F_\theta}, t_\mathcal{D}\right). \tag{5}$$

More importantly, its time-to-first-output is independent of the entire video length: $T_{\text{first}} \approx t_\mathcal{E} + t_{F_\theta} + t_\mathcal{D}$. To quantify the efficiency advantage, let $S = t_\mathcal{E} + t_{F_\theta} + t_\mathcal{D}$ and $M = \max(t_\mathcal{E}, t_{F_\theta}, t_\mathcal{D})$. Assuming no additional alignment overhead ($T_{\text{overlap}} = 0$), the ratio between the two schemes is $T_{\text{serial}}/T_{\text{stream}} \approx KS/(S + (K-1)M)$. It satisfies

$$3K/(K+2) \ge T_{\text{serial}}/T_{\text{stream}} > 1. \tag{6}$$

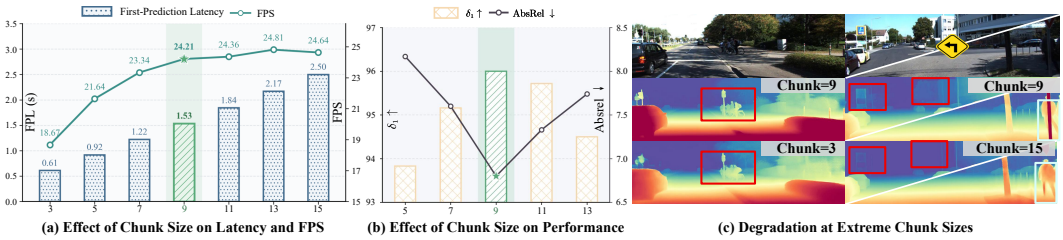

Figure 4: **Effect of streaming chunk size.** (**a**) and (**b**) analyse the effects of chunk size on inference efficiency and depth estimation performance, respectively. The corresponding analysis is detailed in Sec. 3.2. In (**c**), short chunks (l=3) provide insufficient bidirectional context, leading to degraded local geometry of the rider, whereas long chunks (l=15) may hinder adaptation of the geometric anchor to viewpoint changes (), resulting in inaccurate depth estimates for the pole.

In our experiments, the streaming implementation achieves an approximately 2× speedup over the serial baseline.

Streaming Chunk. We build Dandelion on a pretrained video diffusion model that utilizes a VAE to encode four video frames into a single latent frame. In this work, we define a chunk as several consecutive latents. Within each streaming chunk, bidirectional temporal attention provides local context for depth estimation, while causal attention is applied across chunks. As shown in Fig. 4, we systematically analyze the effects of chunk length on depth accuracy, inference speed, and first-prediction latency. We find that increasing chunk length initially improves inference speed because it reduces the total NFE required for a fixed-length video. However, in Fig. 4 (a), when chunk length (l) is ≥ 9, inference speed gains become marginal while first-prediction latency continues to increase. Moreover, in Fig. 4 (b), among the evaluated settings, $l = 9$ yields the best depth estimation performance. As shown in Fig. 4 (c), we think this is because short chunks limit the local bidirectional context, whereas long chunks disturb the geometric anchor due to substantial motion or scene changes. Therefore, we finally set $l = 9$ to achieve the best accuracy and efficiency.

3.3 Two-Stage Training Strategy

Accurate depth prediction for individual chunks is not sufficient for streaming video depth estimation. When new frames are processed, successive predictions must remain consistent with the historical geometric reference; otherwise, locally plausible depth maps may still exhibit scale shifts, structural drift, or discontinuities (Fig. 5 w/o T.T.S). To build this anchor, we decompose the training into two stages.

In the first stage (Fig. 2 lower-left), we construct the history memory from GT depth latents $d_k^{\mathcal{H}}$ and perturb them with moderate noise. In this process, GT depth provides a perfect geometric temporal-spatial anchor and the noise perturbation prevents over-reliance on clean depth context. In this way, the pre-trained DiT, F_θ, learns a reliable depth reference.

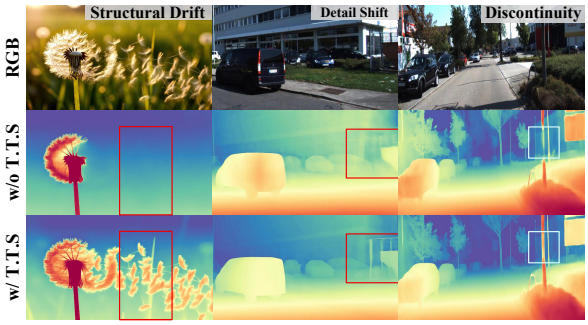

Figure 5: **Visualization of the Two-Stage Training Strategy (T.T.S.).** w/o T.T.S. and w/ T.T.S. denote single-stage training and our two-stage strategy, respectively. The results of (w/o T.T.S.) show a collapsed background, blurred contours, displaced locations, and a poorly defined depth boundary of the pole.

Based on this, the second stage (Fig. 2 lower-middle) replaces GT depth with imperfect rollout predictions to strengthen the geometric anchor's robustness. Moreover, truncating gradients avoids the prohibitive cost of full-sequence autoregressive training. The results in Fig. 5 w/ T.T.S clearly shows the performance gain.

3.4 Multi-Scale History Attention

Exploration of Temporal Propagation. As illustrated in Fig. 6 (a), we intuitively find that the distant observations ($t-20, t-30$) can only provide limited fine-grained geometric information for the current prediction (t). To quantify this, we measure the inter-frame *Detail Relevance* through scene information. The yellow curve in Fig. 6 (b) shows a narrow recent range with strong correspondence, an intermediate range with diminishing correspondence, and a distant range with negligible relevance. We further measure the *Scale Similarity* through their depth distributions. Although the occlusions and scene transitions may introduce slight fluctuations, its exponential moving average (EMA) is largely stable (green curve in Fig. 6 (b)).

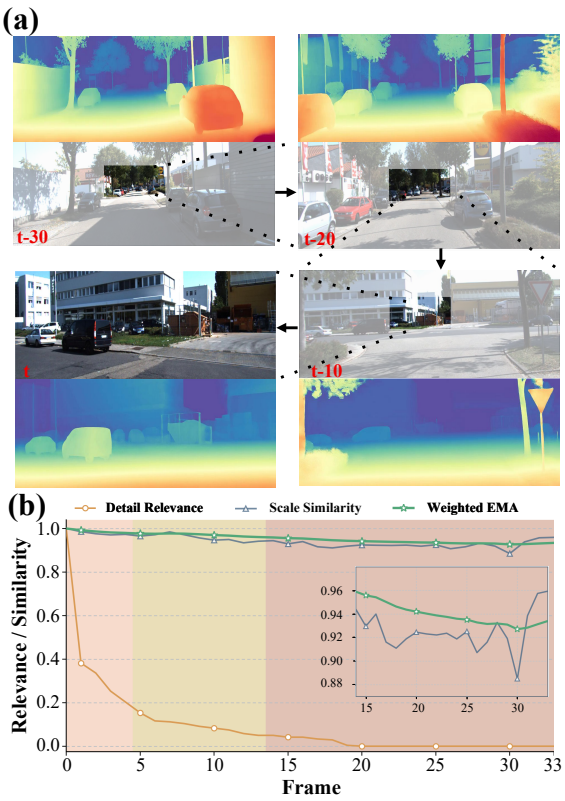

Figure 6: **Temporal characteristics of depth history.** (**a**) Even the recent frame ($t-10$) contains only a small portion of the current view (t), while the distant frame ($t-30$) provides negligible geometric information for the current depth prediction. (**b**) Detail relevance, scale similarity, and EMA similarity vs. temporal distance. The colored background denotes recent, intermediate, and distant history. The inset enlarges the small variations. The similarity calculation is detailed in the Sec A.2.

Multi-scale Depth Memory. Based on the above observation, we construct a fixed-capacity multi-scale depth memory that filters out fine-grained details from distant history while retaining its scale information. Specifically, before processing $z_{(x_k)}$, Dandelion maintains a rolling history cache $\mathcal{B}_k = [\mathbf{b}_{k,1}, \mathbf{b}_{k,2}, \ldots, \mathbf{b}_{k,L}]$, $L = \ell_\mathrm{l} + \ell_\mathrm{m} + \ell_\mathrm{s}$. Each $\mathbf{b}_{k,r}$ is a predicted depth latent. The $\ell_\mathrm{s}, \ell_\mathrm{m}, \ell_\mathrm{l}$ denote the latent numbers assigned to short-, mid-, and long-range history, respectively. According to their temporal distance from the current chunk, the cache is divided into three consecutive regions: $\mathcal{H}_k^\mathrm{l} = [\mathbf{b}_{k,1}, \ldots, \mathbf{b}_{k,\ell\mathrm{l}}]$, $\mathcal{H}_k^\mathrm{m} = [\mathbf{b}_{k,\ell_\mathrm{l}+1}, \ldots, \mathbf{b}_{k,\ell\mathrm{l}+\ell_\mathrm{m}}]$, and $\mathcal{H}_k^\mathrm{s} = [\mathbf{b}_{k,\ell\mathrm{l}+\ell_\mathrm{m}+1}, \ldots, \mathbf{b}_{k,L}]$. To preserve global geometric information, we use scale-specific 3D convolutional patch embeddings. For the short-, mid-, and long-range histories ($i = \mathrm{s}, \mathrm{m}, \mathrm{l}$), we assign scale indices $j = 0, 1, 2$, respectively. The temporal and spatial compression factors are then defined as $p_j^t = 2^j$ and $p_j^s = 2^{j+1}$. The resulting memory tokens are defined as $\mathbf{M}_k^i = \mathrm{Flatten}\left(\mathrm{Conv3D}_i\left(\mathcal{H}_k^i\right)\right)$. Each $\mathrm{Conv3D}_i$ layer is randomly initialized and jointly trained with the model. In this setting, for a history region containing ℓ_i latent frames with spatial resolution $h \times w$, the resulting number of memory tokens is approximately

$$N_i \approx \ell_i hw / [p_j^t \left(p_j^s\right)^2]. \tag{7}$$

As shown in Fig. 2, the three memory groups are then concatenated with the current RGB tokens and sent to the video DiT to generate the depth latent $z_{(d_k)}$:

$$\mathbf{T}_k = \left[\mathbf{M}_k^\mathrm{long}; \mathbf{M}_k^\mathrm{mid}; \mathbf{M}_k^\mathrm{short}; \mathbf{C}_k^\mathrm{rgb}\right]. \tag{8}$$

The total number of tokens is: $N_{\mathbf{T}_k} = N_\mathrm{long} + N_\mathrm{mid} + N_\mathrm{short} + N_\mathrm{rgb}$. For $\ell_\mathrm{l}, \ell_\mathrm{m}, \ell_\mathrm{s}$=6,2,1, our multi-scale representation reduces the total number of input tokens from 21,600 to 12,460, a 42.3% reduction. As shown in Fig. 10, our multi-scale depth memory maintains a stable inference speed as the history context increases.

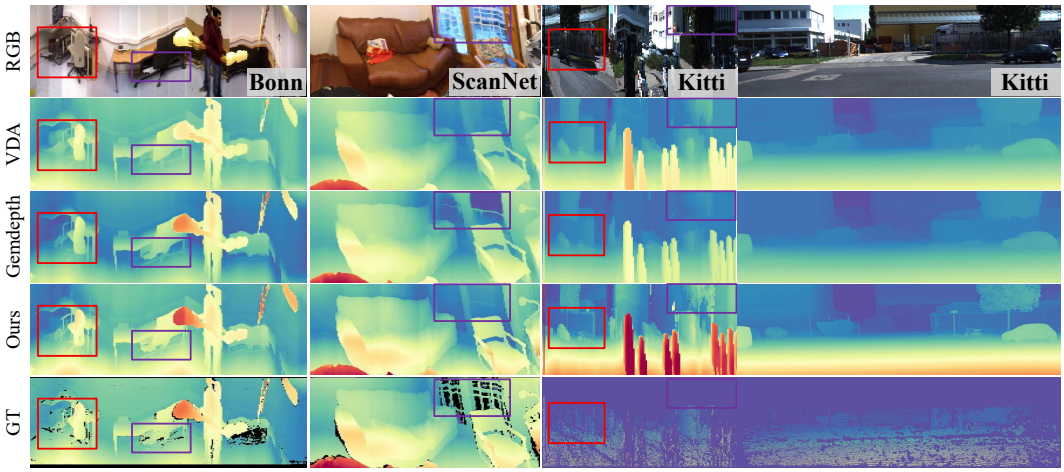

Figure 7: Qualitative results on indoor and outdoor scenes. Details are shown in Sec. 4.2.

Table 1: Comparison of zero-shot depth estimation results on short and long videos. The best results in each column are shown in **bold**, and the second-best are underlined. "-" indicates training on the corresponding evaluation dataset. OOM denotes an out-of-memory error during inference. All of these results were evaluated on an NVIDIA RTX 4090 GPU.

Method	Online Inference	ScanNet				KITTI				Bonn			
		AbsRel ↓	δ_1 ↑	AbsRel ↓	δ_1 ↑	AbsRel ↓	δ_1 ↑	AbsRel ↓	δ_1 ↑	AbsRel ↓	δ_1 ↑	AbsRel ↓	δ_1 ↑
Offline / Serial / Stream		**Short**		**Long**		**Short**		**Long**		**Short**		**Long**	
VDA	✗	7.8	94.9	8.5	92.0	8.2	93.7	8.6	93.1	6.8	95.3	7.3	94.0
ChronoDepth	✗	16.2	82.5	20.1	72.1	13.9	77.1	17.2	83.5	12.3	90.4	15.7	84.3
DepthCrafter	✗	12.6	85.1	17.3	76.2	10.6	89.2	13.5	79.4	7.4	94.7	12.8	86.1
DVD	✗	6.9	95.7	7.9	96.3	7.7	94.5	8.1	94.5	6.2	96.1	6.8	96.0
GemDepth	✗	6.7	96.1	8.0	95.1	7.9	94.2	7.8	93.9	6.4	95.9	7.0	95.5
FlashDepth	✓	10.1	90.3	13.8	86.7	13.4	80.6	16.9	73.8	12.1	90.5	7.6	94.6
StreamVGGT	✓	-	-	-	-	12.2	85.5	OOM	OOM	10.3	88.5	OOM	OOM
Stream3R	✓	-	-	-	-	10.6	89.6	OOM	OOM	7.5	94.1	OOM	OOM
VDA-S	✓	9.3	90.4	11.6	91.7	11.9	91.4	11.4	90.1	7.7	93.2	11.0	95.8
Dandelion (Ours)	✓	**6.2**	**97.9**	**7.0**	**97.7**	**6.8**	**96.0**	**7.3**	**96.3**	**5.4**	**97.8**	**6.2**	**97.5**

Table 2: Quantitative evaluation of temporal consistency.

Method	VDA-S	DepthCrafter	VDA	DVD	GemDepth	Dandelion (Ours)
TAE Metric ↓	0.972	0.673	0.596	0.568	0.523	**0.452**

4 Experimental Results

4.1 Experimental Setting

Implementation Details. Dandelion is built upon Wan-2.1 (Wan et al. (2025)) and jointly trained on synthetic image and video data. For the training data, we follow DVD (Zhang et al. (2026)) but use less training data. Training is performed on two RTX Pro 6000 GPUs for approximately five days. We use the AdamW optimizer with a learning rate of 1×10^{-4} and a batch size of 64. Additional implementation details are provided in Appendix A.1.

Evaluation. We benchmark Dandelion on both zero-shot video and image datasets. For video evaluation, we use KITTI (Geiger et al. (2013)), ScanNet (Dai et al. (2017)), and Bonn (Palazzolo et al. (2019)); for image evaluation, we report results on KITTI, NYUv2 (Silberman et al. (2012)), and DIODE (Vasiljevic et al. (2019)). We measure geometric accuracy using AbsRel and δ_1, and evaluate temporal consistency with Temporal Alignment Error (TAE).

Baselines. We evaluate Dandelion against several video depth estimation methods from two paradigms: (1) offline: VDA (Chen et al. (2025)), ChronoDepth (Shao et al. (2025)), DVD (Zhang et al. (2026)), DepthCrafter (Hu et al. (2025)), and GemDepth (Liu et al. (2026)); and (2) Serial:

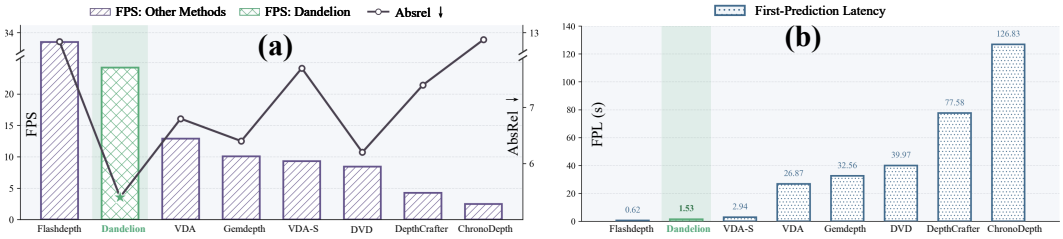

Figure 8: (**a**) compares inference speed and depth estimation performance; (**b**) reports the first-prediction latency of each method.

FlashDepth (Chou et al. (2025)), StreamVGGT (Zhuo et al. (2026)), and Stream3R (Lan et al. (2026)). VDA also provides a streaming version, which we refer to as VDA-S.

4.2 Comparative Results

Quantitative Comparison. Tab. 1 reports zero-shot depth estimation on short and long videos, evaluated using 110 and 300 frames per sequence, respectively. Among the evaluated methods, only FlashDepth, StreamVGGT, Stream3R, and Dandelion support online inference. The other online baselines either exhibit a substantial accuracy gap or fail in long-video evaluations due to OOM. In contrast, Dandelion combines real-time online inference over long videos with the best performance. For short videos, Dandelion reduces AbsRel from 6.2 for DVD to 5.4 on Bonn, achieving a 12.9% improvement over the second-best method. For long videos, Dandelion reduces the previous best AbsRel from 7.9 to 7.0 on ScanNet, yielding an 11.4% improvement.

Qualitative Comparison. To further vividly demonstrate temporal consistency, we visualize temporal depth profiles by extracting depth values along a fixed vertical line from each frame and stacking them over time. As shown in Fig. 7, the red boxes highlight the preservation of local geometric structures, while the purple boxes emphasize temporal continuity. Dandelion maintains clear object boundaries and fine-grained depth variations throughout the sequence, whereas competing methods tend to lose local structures. More importantly, their temporal profiles contain noticeable discontinuities, indicating unstable predictions across adjacent frames. In contrast, our profiles evolve smoothly over time while preserving high-frequency geometric details, closely following the temporal structure of the GT. These results explain the strong results in Tab. 1 and Tab. 2.

Temporal Consistency. Tab. 2 reports the TAE results on ScanNet. Dandelion achieves the lowest TAE of 0.452, corresponding to a relative reduction of 13.6% over GemDepth and 24.2% over VDA. The lower alignment error indicates more stable predictions across adjacent frames.

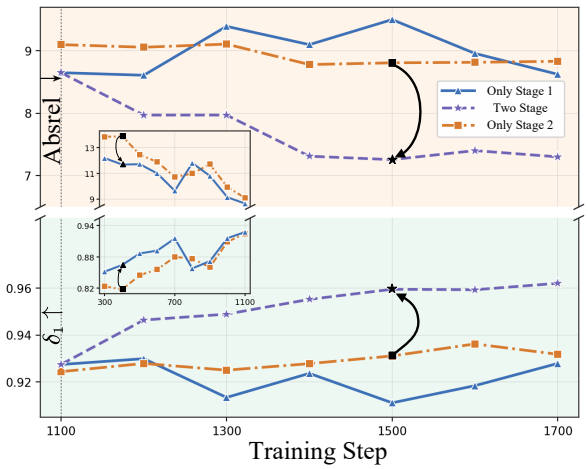

Figure 9: Performance curves showing improved accuracy with two-stage training on the KITTI dataset. The inset shows the trends in AbsRel and δ_1 during the early training period.

4.3 Comparison of Inference Efficiency

Fig. 8 (a) compares inference throughput and AbsRel across methods. Dandelion achieves the lowest AbsRel of 5.4 at approximately 24 FPS, exceeding the throughput of every method except FlashDepth. Although FlashDepth reaches approximately 33 FPS, its AbsRel is 124.1% higher than that of Dandelion. Fig. 8 (b) shows that Dandelion reduces the noticeable first-predict latency. These results demonstrate that Dandelion combines high inference speed and short first-prediction latency with accurate depth estimation. The measurement protocol and implementation details underlying the reported FPL values are provided in Appendix A.2.

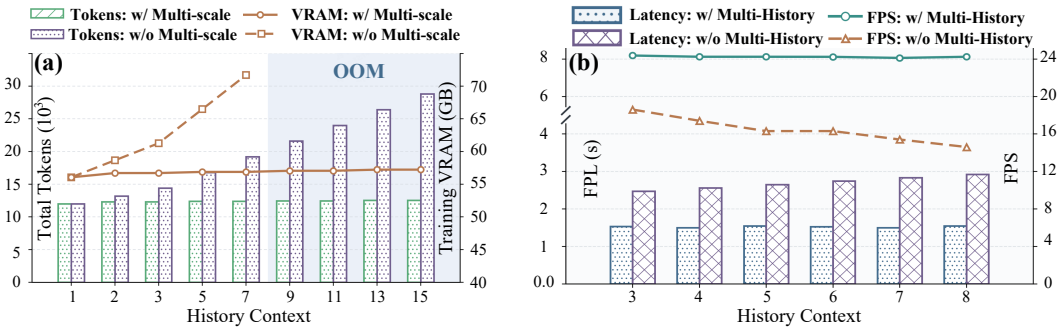

Figure 10: (**a**) Total input tokens/Training VRAM under different history lengths. (**b**) Inference throughput and FPL under different history lengths.

4.4 Ablation Studies

Effect of the Streaming Chunk. The necessity of streaming chunks is shown in Sec. 3.2.

Effect of the Two-stage Training Strategy. We compare Dandelion trained with Stage 1 alone, Stage 2 alone, and our two-stage training strategy. The early-step insets in Fig. 9 show that Stage 1 alone attains better initial performance, while our two-stage strategy consistently outperforms both single-stage training strategies later in training. At step 1,500, Dandelion trained with the two-stage strategy achieves a 17% lower AbsRel than Dandelion trained with Stage 2 alone, as indicated by the black arrow.

Efficiency of the Multi-scale History Attention. Fig. 10 (a) analyzes the token efficiency of multi-scale history attention. As the history length increases, uniformly encoding depth latents leads to near-linear growth in the number of input tokens, whereas our multi-scale representation maintains a controllable budget while preserving a longer temporal context. Fig. 10 (b) shows that the first-prediction latency remains around 1.5 seconds and the inference speed stays near 24 FPS as the history length increases. Meanwhile, the uniformly encoded setting reduces inference speed from 18 to 13.8 FPS and nearly doubles the FPL.

Effect of Multi-Scale Depth Memory. We ablate the three temporal scales of the depth history. As shown in Tab. 3, removing the entire history increases AbsRel from 7.3 to 8.5 and decreases δ_1 from 96.3 to 94.4, demonstrating the importance of historical depth for geometric prediction. Among the individual scales, removing the short-range history $\mathcal{H}^s$ results in the largest performance drop, followed by the mid-range history $\mathcal{H}^m$ and long-range history $\mathcal{H}^l$.

Table 3: Ablation study of the multi-scale history configuration on the KITTI dataset.

History Configuration	AbsRel (↓)	δ_1 (↑)
w/o History	8.5	94.4
w/o $\mathcal{H}^s$	8.2	94.9
w/o $\mathcal{H}^m$	8.0	95.3
w/o $\mathcal{H}^l$	7.8	95.5
Full History	**7.3**	**96.3**

5 Conclusion

In this work, we presented **Dandelion**, a generative framework for real-time streaming video depth estimation. Dandelion reformulates a pre-trained video generative model into a causal autoregressive depth estimator. Its streaming chunk strategy preserves bidirectional temporal context within each chunk, with a carefully selected chunk length balancing depth accuracy, inference speed, and first-prediction latency. The two-stage training protocol establishes a coherent geometric reference from ground-truth depth history and strengthens its robustness to model-generated histories during autoregressive rollout. Multi-scale History Attention further compresses distant history to reduce computational overhead while retaining geometric guidance. Extensive experiments demonstrate SoTA zero-shot depth estimation and real-time streaming inference at 24 FPS, using only 0.2% of the task-specific training data used by VDA. These results highlight the potential of large-scale video generative priors for accurate, temporally coherent, and efficient streaming depth estimation.

REFERENCES

Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024.

Sili Chen, Hengkai Guo, Shengnan Zhu, Feihu Zhang, Zilong Huang, Jiashi Feng, and Bingyi Kang. Video depth anything: Consistent depth estimation for super-long videos. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 22831–22840, 2025.

Gene Chou, Wenqi Xian, Guandao Yang, Mohamed Abdelfattah, Bharath Hariharan, Noah Snavely, Ning Yu, and Paul Debevec. Flashdepth: Real-time streaming video depth estimation at 2k resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9638–9648, 2025.

Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5828–5839, 2017.

Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4340–4349, 2016.

Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The international journal of robotics research*, 32(11):1231–1237, 2013.

Jing He, Haodong Li, Wei Yin, Yixun Liang, Leheng Li, Kaiqiang Zhou, Hongbo Zhang, Bingbing Liu, and YingCong Chen. Lotus: Diffusion-based visual foundation model for high-quality dense prediction. In *International Conference on Learning Representations*, volume 2025, pp. 89454–89467, 2025.

Wenbo Hu, Xiangjun Gao, Xiaoyu Li, Sijie Zhao, Xiaodong Cun, Yong Zhang, Long Quan, and Ying Shan. Depthcrafter: Generating consistent long depth sequences for open-world videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2005–2015, 2025.

Xun Huang, Zhengqi Li, Guande He, Mingyuan Zhou, and Eli Shechtman. Self forcing: Bridging the train-test gap in autoregressive video diffusion. *Advances in Neural Information Processing Systems*, 38:167283–167308, 2025.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

Yushi Lan, Yihang Luo, Fangzhou Hong, Shangchen Zhou, Honghua Chen, Zhaoyang Lyu, Bo Dai, Shuai Yang, Chen Change Loy, and Xingang Pan. Stream3r: Scalable sequential 3D reconstruction with causal transformer. In *International Conference on Learning Representations*, 2026.

Yuecheng Liu, Junda Cheng, Longliang Liu, Wenjing Liao, Hanrui Cheng, Yuzhou Wang, and Xin Yang. Gemdepth: Geometry-embedded features for 3d-consistent video depth. In *Forty-third International Conference on Machine Learning*, 2026.

Emanuele Palazzolo, Jens Behley, Philipp Lottes, Philippe Giguere, and Cyrill Stachniss. Refusion: 3d reconstruction in dynamic environments for rgb-d cameras exploiting residuals. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7855–7862. IEEE, 2019.

Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M Susskind. Hypersim: A photorealistic synthetic dataset for holistic indoor scene understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10912–10922, 2021.

Jiahao Shao, Yuanbo Yang, Hongyu Zhou, Youmin Zhang, Yujun Shen, Vitor Guizilini, Yue Wang, Matteo Poggi, and Yiyi Liao. Learning temporally consistent video depth from video diffusion priors. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 22841–22852, 2025.

Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pp. 746–760. Springer, 2012.

Igor Vasiljevic, Nick Kolkin, Shanyi Zhang, Ruotian Luo, Haochen Wang, Falcon Z Dai, Andrea F Daniele, Mohammadreza Mostajabi, Steven Basart, Matthew R Walter, et al. Diode: A dense indoor and outdoor depth dataset. *arXiv preprint arXiv:1908.00463*, 2019.

Team Wan, Ang Wang, Baole Ai, Bin Wen, Chaojie Mao, Chen-Wei Xie, Di Chen, Feiwu Yu, Haiming Zhao, Jianxiao Yang, et al. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. doi: 10.48550/arXiv.2503.20314. URL https://arxiv.org/abs/2503.20314.

Jiyuan Wang, Chunyu Lin, Lang Nie, Shujun Huang, Yao Zhao, Xing Pan, and Rui Ai. Weatherdepth: Curriculum contrastive learning for self-supervised depth estimation under adverse weather conditions. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4976–4982. IEEE, 2024.

Jiyuan Wang, Chunyu Lin, Guan Chen, Lang Nie, Jing He, Haodong Li, Kang Liao, and Yao Zhao. Jasmine: Harnessing diffusion prior for self-supervised depth estimation. *Advances in Neural Information Processing Systems*, 38:79901–79930, 2026a.

Jiyuan Wang, Chunyu Lin, Lei Sun, Zhi Cao, Yuyang Yin, Lang Nie, Zhenlong Yuan, Xiangxiang Chu, Yunchao Wei, Kang Liao, et al. Geometry-guided reinforcement learning for multi-view consistent 3d scene editing. *arXiv preprint arXiv:2603.03143*, 2026b.

Jiyuan Wang, Chunyu Lin, Lei Sun, Rongying Liu, Lang Nie, Mingxing Li, Kang Liao, and Xiangxiang Chu. Fe2e: From editor to dense geometry estimator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19844–19853, 2026c.

Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4909–4916. IEEE, 2020.

Gangwei Xu, Haotong Lin, Hongcheng Luo, Xianqi Wang, Jingfeng Yao, Lianghui Zhu, Yuechuan Pu, Cheng Chi_, Haiyang Sun, Bing Wang, et al. Pixel-perfect depth with semantics-prompted diffusion transformers. *Advances in Neural Information Processing Systems*, 38:174731–174755, 2025a.

Tian-Xing Xu, Xiangjun Gao, Wenbo Hu, Xiaoyu Li, Song-Hai Zhang, and Ying Shan. Geometrycrafter: Consistent geometry estimation for open-world videos with diffusion priors. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1–13. IEEE, 2025b.

Honghui Yang, Di Huang, Wei Yin, Chunhua Shen, Haifeng Liu, Xiaofei He, Binbin Lin, Wanli Ouyang, and Tong He. Depth any video with scalable synthetic data. In *International Conference on Learning Representations*, volume 2025, pp. 97335–97349, 2025.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything v2. *Advances in Neural Information Processing Systems*, 37:21875–21911, 2024.

Hongfei Zhang, Harold Haodong Chen, Chenfei Liao, Jing He, Zixin Zhang, Haodong Li, Yihao Liang, Kanghao Chen, Bin Ren, Xu Zheng, et al. Dvd: Deterministic video depth estimation with generative priors. In *ICML 2026 Workshop on Foundations of Deep Generative Models: Understanding Memorization, Generalization, and Reasoning*, 2026.

Dong Zhuo, Wenzhao Zheng, Jiahe Guo, Yuqi Wu, Jie Zhou, and Jiwen Lu. Streaming visual geometry transformer. In *International Conference on Learning Representations*, volume 2026, pp. 88055–88072, 2026.

A APPENDIX

A.1 MORE DETAILS OF IMPLEMENTATION

Training Dataset Details. As summarized in Tab. 4, Dandelion is trained on Hypersim (Roberts et al. (2021)), Virtual KITTI (Gaidon et al. (2016)) and TartanAir (Wang et al. (2020)).

Table 4: An overview of the datasets for training.

Dataset	Scenes	Image Pairs	Resolution
Hypersim (Roberts et al. (2021))	349	39.65K	480 ×640
Virtual KITTI (Gaidon et al. (2016))	4	67.16K	352 ×1216
TartanAir (Wang et al. (2020))	16	13.96K	480 ×640

For data augmentation, RGB and depth pairs are horizontally flipped in a synchronized manner, while training clips are randomly sampled from video sequences. Each sample is resized or cropped according to the resolution specified for its source dataset. This heterogeneous training configuration exposes the model to diverse scene domains, aspect ratios, and spatial resolutions.

Training Details. For reproducibility, we provide the complete training configuration of Dandelion in this section. The two-stage protocol is implemented with LoRA-based adaptation, together with stage-specific optimization schedules and depth-history construction. Tab. 5 summarizes the backbone, hardware configuration, numerical precision, LoRA settings, optimization hyperparameters, and historical-context setup used in each stage. We share the same backbone with DVD (Zhang et al. (2026)). Specifically, "Warm-up Chunk = 2" means that we process the two chunks preceding the current training chunk to populate the cache with historical depth predictions, ensuring that the current chunk is conditioned on a populated history rather than an empty cache.

Table 5: The detailed hyperparameters of the two-stage training protocol.

Configuration	Hyperparameter	Value	Hyperparameter	Value
	Stage-1		**Stage-2**	
Backbone	Pre-trained Model	Wan-2.1	Pre-trained model	Wan-2.1
Timestep	τ_{Cur}	0.5	τ_{Hist}	0
Training Setting	Numerical Precision	BFloat16	Numerical Precision	BFloat16
LoRA	Rank	512	Rank	512
	Alpha	512	Alpha	512
	Global Batch Size	64	Global Batch Size	64
Optimization	Optimizer	AdamW	Optimizer	AdamW
	Learning Rate	1e-4	Learning Rate	2e-5
	Learning Rate Schedule	Constant	Learning Rate Schedule	Constant
	Training Steps	1100	Training Steps	600
Depth History	History Source	GT Depth	History Source	Autoregressive Depth
	Chunk Length	9	Chunk Length	9
	Warm-up Chunk	-	Warm-up Chunk	2

Although our model is trained in a streaming manner, we supervise Dandelion using the spatial and temporal loss adopted from DVD (Zhang et al. (2026)). To evaluate temporal consistency, we adopt the Temporal Alignment Error (TAE) metric in (Yang et al. (2025)).

$$\text{TAE} = \frac{1}{2\,(F-1)} \sum_{k=1}^{F-1} \text{AbsRel}\left(f\left(\hat{x}_d^k, p^k\right), \hat{x}_d^{k+1}\right) + \text{AbsRel}\left(f\left(\hat{x}_d^{k+1}, p_-^{k+1}\right), \hat{x}_d^k\right), \tag{9}$$

where F denotes the number of frames. f denotes the function that uses the transformation matrix p^k to map the k-th estimated depth frame to the (k+1)-th frame. p_-^{k+1} denotes the inverse transformation matrix. We follow the evaluation protocol of DVD (Zhang et al. (2026)) regarding the source of camera transformations, depth reprojection, occlusion handling, valid-pixel masking, and depth-scale alignment.

Zero-shot Depth Estimation Performance on Single Images. Despite being designed for streaming video inference, Dandelion also generalizes effectively to single images, yielding the best results on DIODE and KITTI while remaining competitive on NYUv2. As shown in Tab. 6, these

Table 6: Zero-shot single-image depth estimation results.

Method	DIODE		KITTI		NYUv2	
	AbsRel ↓	δ_1 ↑	AbsRel ↓	δ_1 ↑	AbsRel ↓	δ_1 ↑
VDA (Chen et al. (2025))	24.8	73.4	8.6	93.0	**5.2**	**97.5**
DepthCrafter (Hu et al. (2025))	33.9	67.0	10.5	88.6	15.2	87.9
GemDepth (Liu et al. (2026))	25.0	71.7	8.4	92.7	6.1	95.6
DVD (Zhang et al. (2026))	<u>23.2</u>	<u>73.6</u>	<u>8.3</u>	<u>94.1</u>	5.8	96.2
Dandelion (Ours)	**22.2**	**74.2**	**7.8**	**94.6**	<u>5.6</u>	<u>96.7</u>

results demonstrate that the proposed causal streaming framework and multi-scale attention preserve strong geometric priors for individual frames while sustaining accurate depth prediction over extended video sequences. Serial methods progressively update their temporal states as new frames arrive; evaluating them on an isolated image would therefore reduce their inference to the initial step without any accumulated temporal context, which does not reflect their intended operating setting. We thus evaluate these methods in the video experiments and restrict the single-image comparison to offline video depth estimators.

A.2 Details of Evaluation and Visualization

Inference Scheduling and Runtime Measurement. We measure inference runtime on a single NVIDIA RTX 4090. Dandelion achieves a measured throughput of approximately 24 FPS. In the serial baseline, the encoder starts processing the next chunk only after the decoder has finished processing the current chunk, and temporal alignment is applied as post-processing. We compare the measured total completion times of the serial and streaming pipelines on the same ten-chunk sequence under identical input and precision settings, including post-processing where applicable. Dandelion achieves an approximately $2\times$ end-to-end speedup, reflecting the combined benefits of overlapping pipeline stages and eliminating alignment post-processing.

Analysis of First-Prediction Latency. We evaluate first-prediction latency (FPL) with preloaded video inputs in Sec. 4.3. FPL measures the time from input submission to the inference pipeline until the first complete depth frame becomes available for display. This setting excludes video acquisition time (all models are the same).

Most existing video depth estimation methods (*e.g.*, VDA, DVD Zhang et al. (2026)) rely on computationally intensive temporal post-processing to improve temporal consistency. However, they have to release the results after processing the complete video (rather than clip-wise depth predictions), leading to a substantial first-prediction latency. In contrast, Dandelion performs chunk-wise prediction in latent space and incrementally releases the decoded depth frames. Each decoded output is released immediately, without waiting for the remaining frames in the chunk to be decoded. This output schedule reduces the waiting time before the first depth frame becomes available and enables smooth delivery during inference.

It is worth noting that, even when video acquisition time is included, Dandelion's first-prediction latency is approximately 2.6 s. This does not change the latency ranking in Fig. 6: Dandelion remains second only to FlashDepth. In comparison, VDA-S requires 2.94 s to produce its first prediction even without accounting for video acquisition time. We consider this latency a practical trade-off for streaming applications that tolerate a short buffering delay, particularly given Dandelion's substantially better depth accuracy than FlashDepth.

Temporal Roles of Multi-scale Depth History. The temporal profiles in Fig. 11 further reveal their distinct roles. Without $\mathcal{H}^s$, local contour trajectories become discontinuous and abrupt depth changes emerge across adjacent chunks, indicating its importance in preserving fine-grained boundaries and short-term geometric continuity. Removing $\mathcal{H}^m$ produces less smooth and occasionally piecewise depth transitions over intermediate temporal ranges, showing its role in connecting local geometry with longer-term scene structure. In contrast, removing $\mathcal{H}^l$ largely preserves local contours but introduces broad shifts in the depth distribution over time, reflecting a less stable long-term geometric reference. The full history configuration achieves the best quantitative results and the most coherent temporal profiles, confirming the complementary roles of the three history scales.

Table 7: Extremely long video depth estimation results on Bonn.

Method	**Bonn**(1000frames)	
	AbsRel ↓	δ_1 ↑
VDA (Chen et al. (2025))	9.8	95.8
DVD (Zhang et al. (2026))	<u>7.8</u>	<u>96.2</u>
Dandelion (Ours)	**5.5**	**97.1**

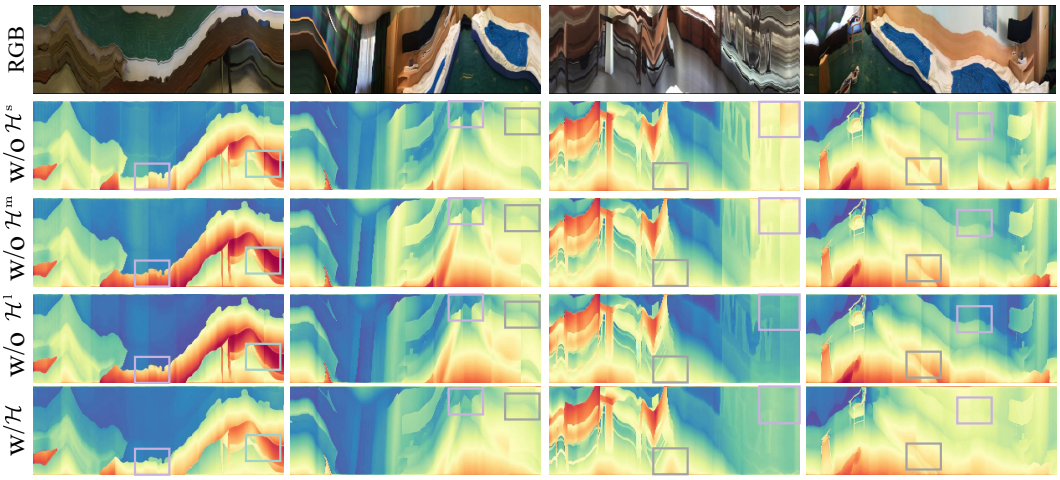

Figure 11: **Qualitative results of multi-scale history ablation.** Removing individual history scales leads to distinct degradation in temporal geometry. The colored boxes highlight the temporally inconsistent depth estimates.

Temporal Similarity Metrics. We characterize the relationship between the current frame t and a historical frame $t - \Delta$ using two complementary metrics. *Detail Relevance* measures their image overlap. Let Ω_t denote the valid image domain of the current frame, and let $M_{t,\Delta}(p)$ be a binary mask indicating whether the scene content at pixel p is also visible in the historical frame. We define

$$R_{\text{detail}}(t, \Delta) = \frac{1}{|\Omega_t|} \sum_{p \in \Omega_t} M_{t,\Delta}(p). \tag{10}$$

A larger value indicates that the historical frame covers a greater fraction of the current view, providing a proxy for the availability of direct visual correspondences. *Scale Similarity* compares the global depth statistics of the two frames. Let P_t and $P_{t-\Delta}$ denote the normalized probability distributions of their valid GT depth values, evaluated using a common depth range and consistent depth units. We convert their Kullback–Leibler (KL) divergence into a similarity score:

$$S_{\text{scale}}(t, \Delta) = \exp[-D_{\text{KL}}(P_t \| P_{t-\Delta})]\,. \tag{11}$$

The score lies in $(0, 1]$, with larger values indicating closer agreement between the depth distributions and a value of 1 corresponding to identical distributions. These metrics capture shared visible content and global depth-distribution similarity, respectively.

A.3 Additional Results

Quantitative Results. As shown in Tab. 7, Dandelion still performs the best on ultra long videos (1000 frames). This indicates that our method can better handel error accumulation and maintain scale stability.

Qualitative Results. Figs. 13 and 14 present additional comparisons with VDA (Chen et al. (2025)) and GemDepth (Liu et al. (2026)) on short and long videos, spanning indoor, outdoor, and highly dynamic scenes. Dandelion preserves fine-grained structures and sharp depth boundaries while maintaining coherent geometry over time.

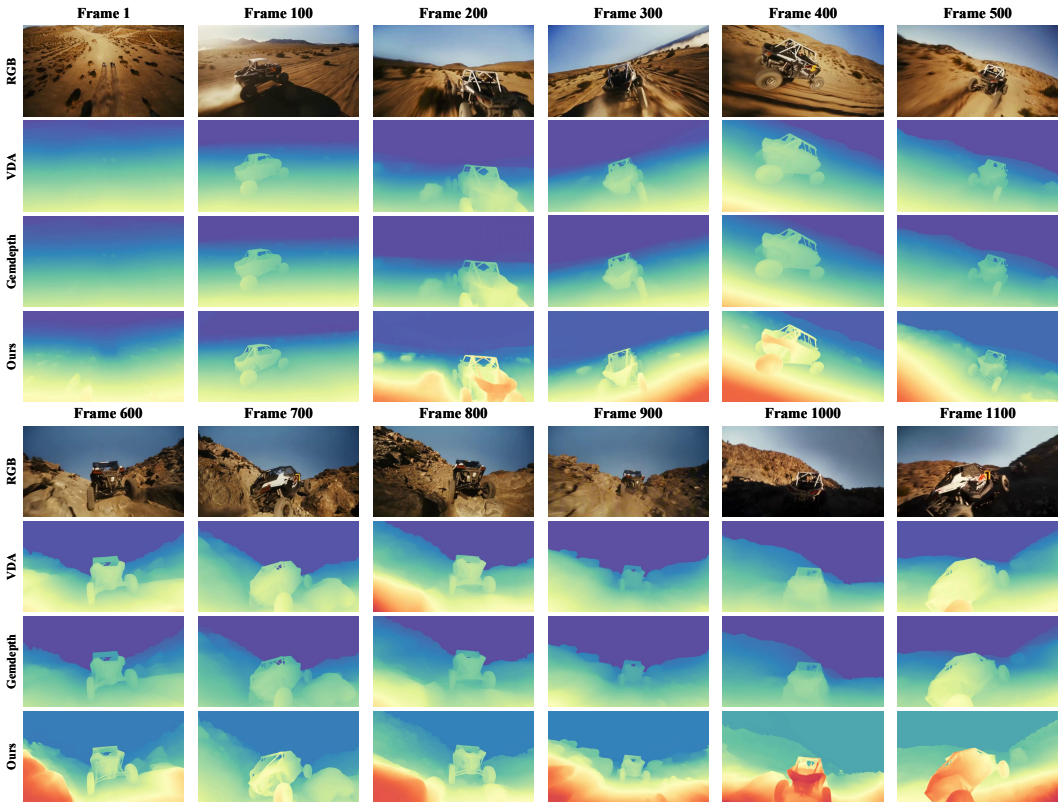

Figure 12: **Challenging cases under rapid motion and blur.** Rapid motion and severe blur reduce the reliability of temporal cues, leading to reduced temporal stability across the compared methods. Dandelion nevertheless preserves clearer object contours and finer local geometry than VDA (Chen et al. (2025)) and GemDepth (Liu et al. (2026)).

A.4 Limitations and Future Work

Robustness to Unreliable Depth History. Our fixed-capacity multi-scale depth history is continuously updated with new predictions but retains predefined temporal coverage. Severe motion blur or rapid camera motion may reduce the reliability of historical cues and impair temporal consistency. Table. 3 and Fig. 11 analyze the contributions of different temporal scales, while Fig. 12 presents challenging cases. These comparisons suggest a shared sensitivity across methods, although Dandelion preserves clearer contours and finer local geometry than VDA Chen et al. (2025) and GemDepth Liu et al. (2026). Future work may explore content-adaptive temporal coverage and confidence-aware cache updates to better handle varying scene dynamics.

Long-Term Geometric Consistency in Super-long Videos. Our method maintains a fixed-capacity rolling depth history to limit the computational and memory costs of streaming inference. Following the evaluation protocol of prior work, we first evaluate the performance on 300-frame videos. To reduce the concern of error accumulation and scale stability over ultra-long videos, we further quantitatively and qualitatively evaluate 1,000-frame videos (Fig. 14, never done by other works before). However, error accumulation and scale stability over **super-longer** time horizons still require systematic quantitative evaluation. Future work will examine how depth errors and scale drift evolve over sequences of 10k+ frames or more, and explore persistent geometric memory and online drift correction to support stable inference over longer durations under limited computational and memory budgets.

Data Scale and Diversity. Dandelion achieves strong zero-shot generalization while using only 0.2% of the training data adopted by prior methods. Although this result demonstrates its data efficiency, increasing the amount and diversity of training data may further improve geometric ac-

curacy and robustness in challenging scenes. Future work will investigate how Dandelion benefits from training on larger and more diverse datasets.

B STATEMENT

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we describe the streaming inference framework, two-stage training strategy, and multi-scale history mechanism in Sec. 3. Appendix A.1 documents the training datasets and data preprocessing, with dataset statistics and stage-specific training configurations summarized in Tables 4 and 5, respectively. Experimental settings and evaluation metrics are described in Sec. 4.1 and Appendix A.1. Appendix A.3 further specifies inference scheduling and the measurement procedures for runtime and first-prediction latency.

AI USE STATEMENT

In this work, we have not used generative AI tools to help develop theoretical models or conceptual frameworks, to formulate mathematical claims, or to provide critical inputs for proving mathematical claims. We have used generative AI tools to assist in the writing of paper, refine hypotheses, provide feedback on research experiments, support qualitative data analysis, and clean/reformat datasets. We have reviewed all AI-assisted work.

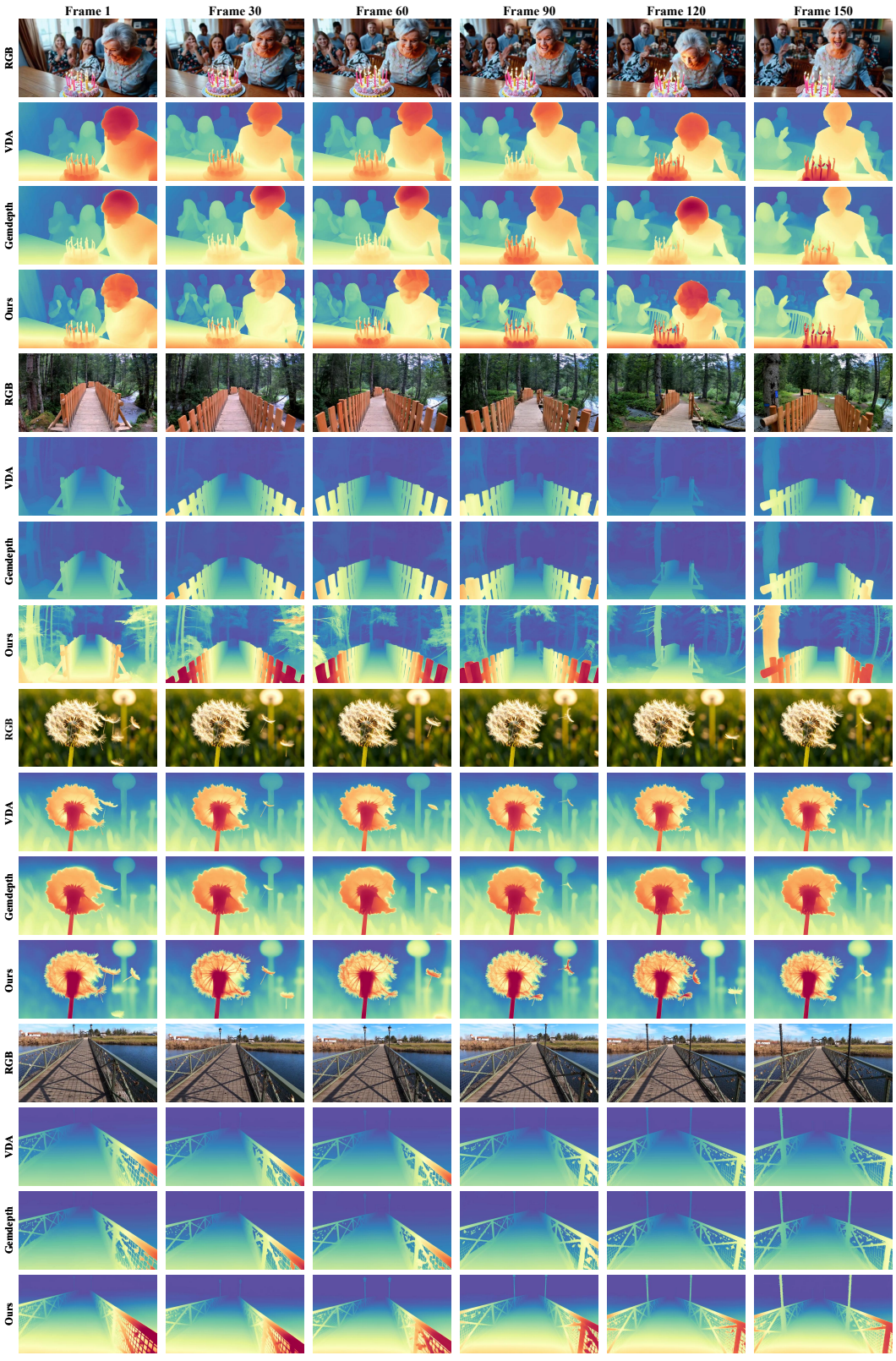

Figure 13: Additional qualitative comparisons on short videos.

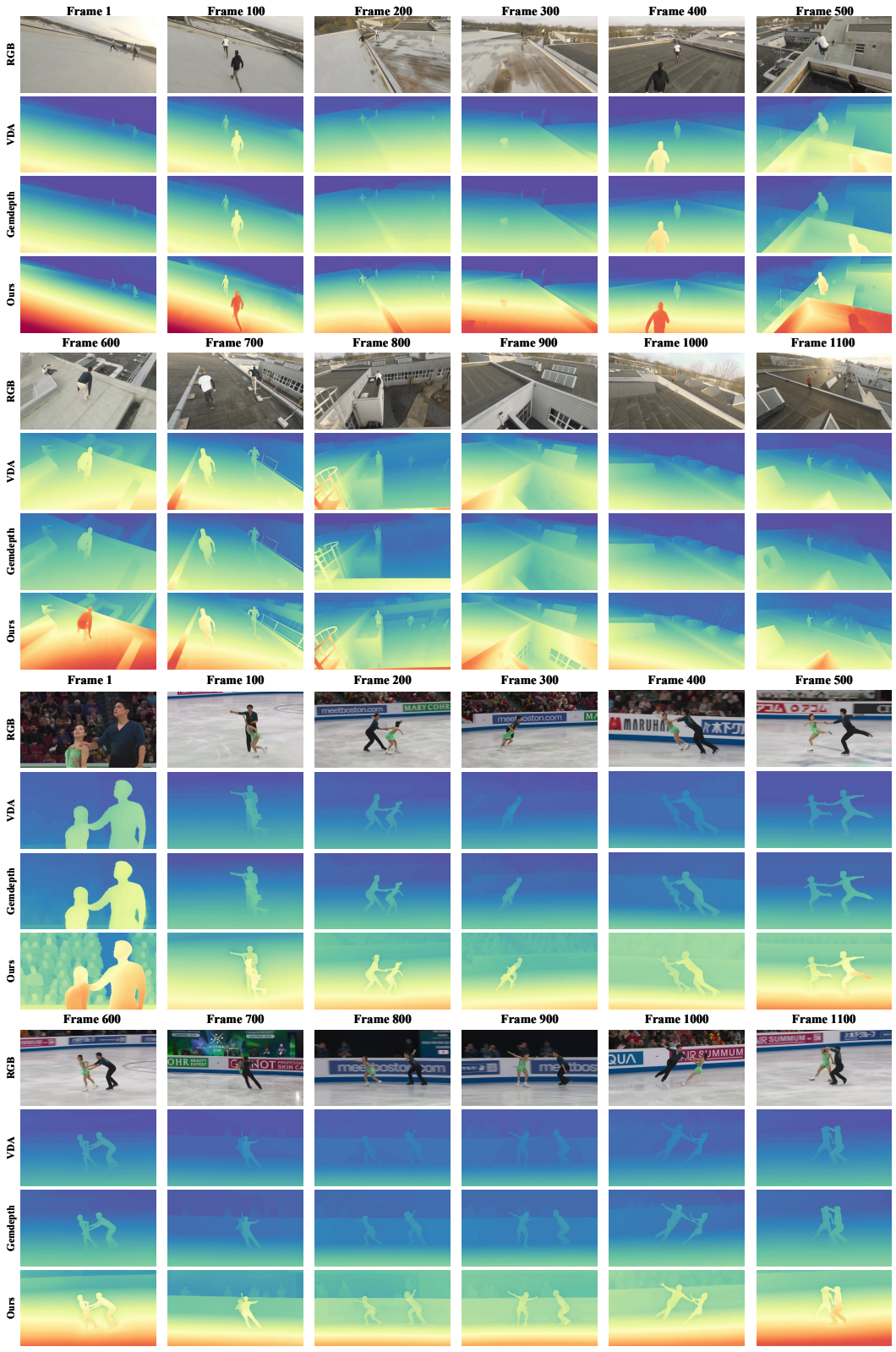

Figure 14: Additional qualitative comparisons on long videos.