VDPP: Video Depth Post-Processing for Speed and Scalability

Supplementary Material

Appendix

In this supplementary material, we provide additional implementation details in Appendix A. We then offer an in-depth qualitative analysis (Appendix B) against competing methods, visually validating VDPP's superior spatio-temporal coherence in Figure 1, 2. Furthermore, we present critical evidence of practical viability, starting with comprehensive performance metrics on limited resolutions in Table 1 (Appendix C). We follow this with dedicated validation on real-world LiDAR sensor data in Figure 3 (Appendix D), and conclude with a decisive memory efficiency profiling on the NVIDIA Jetson Orin Nano in Figure **??** and Figure 4 (Appendix E). Finally, we validate the reliability of TGSE as a temporal consistency metric through sensitivity analysis in Figure 5 (Appendix F).

A. Implementation Details

This section provides the comprehensive implementation details for our VDPP framework, covering the datasets, data processing, architectural choices, and training hyperparameters.

Training dataset details. For video training, we utilize three datasets with precise depth annotations: VKITTI [2], TartanAir [8], and PointOdyssey [11]. Initially, we employ an Image-to-Depth model [10] to generate depth estimations from the full-resolution datasets. Our post-processing network is then trained using these generated depth maps. For training, the datasets are processed into non-overlapping, consecutive 16-frame sequences. We apply random cropping to obtain 224x224 patches, which serve as the input resolution. To address the uneven data distribution and prevent domination by large-scale datasets (e.g., TartanAir), we employ a Weighted Random Sampler for uniform sampling.

Implementation details. Our model consists of several components. For the encoder, we utilized the small version of DINOv2 encoder [7] and trained it from scratch. For the decoder, we adopted and fine-tuned the small version head from Video Depth Anything [3]. All other modules were also trained from scratch. We train the model using the AdamW optimizer [6] with a base initial learning rate of 1e-6. We employ a CosineAnnealingWarmRestarts scheduler [5], configured with $T_0 = 10,000$, $T_{mult} = 2$, and $\eta_{min} = 1e-9$. The batch size is set to 16. The loss weights for α and β are set to 1.0 and 10.0, respectively. For downsampling, all depth maps are downsampled using bilinear interpolation with a ratio of $r = 0.5$. The model was trained for 430 hours on a single NVIDIA A6000 GPU, for a total of 630K iterations.

B. Quality Comparison with Depth Estimation Models Across All Metrics

In addition to the quantitative metrics presented in the main paper, this section provides an in-depth qualitative analysis to complement our findings in Figure 1 and Figure 2. We visually substantiate our core claim: VDPP resolves the spatial detail and temporal stability, overcoming the failure modes of both post-processing and end-to-end solutions.

B.1. Spatio-Temporal Coherence Analysis

The visualizations in Figure 1 are impressive. They are specifically chosen to represent common, yet challenging real-world scenarios where existing methods typically fail, forcing a compromise between spatial detail and temporal stability. We compare VDPP against NVDS [9] (post-processing) and VDA [3] (end-to-end) to visually substantiate our core claim: VDPP resolves this trade-off.

The first two rows (Basketball, Hanger) demonstrate the fundamental conflict between high-frequency detail and stability. The bottom two rows (Fire Pit, Train) test robustness in complex environments with transparency and fast motion.

- **In the Basketball scene,** the key challenge is capture in large-scale motion (change in depth-of-field). Both NVDS and VDA-S produce an overly 'flat' and temporally insensitive result in the slit-scan image (purple box), failing to register the player's significant movement away from the camera. VDPP, however, clearly captures this dynamic depth progression.
- **In the Hanger scene,** the slit-scan image (purple box) provides a definitive diagnosis. NVDS, a post-processing method, reveals its critical flaw: it introduces chaotic, high-frequency oscillations, resulting in severe temporal flickering even in static background regions. Conversely, VDA-S achieves temporal smoothness, but does so by sacrificing spatial fidelity. Its end-to-end approach

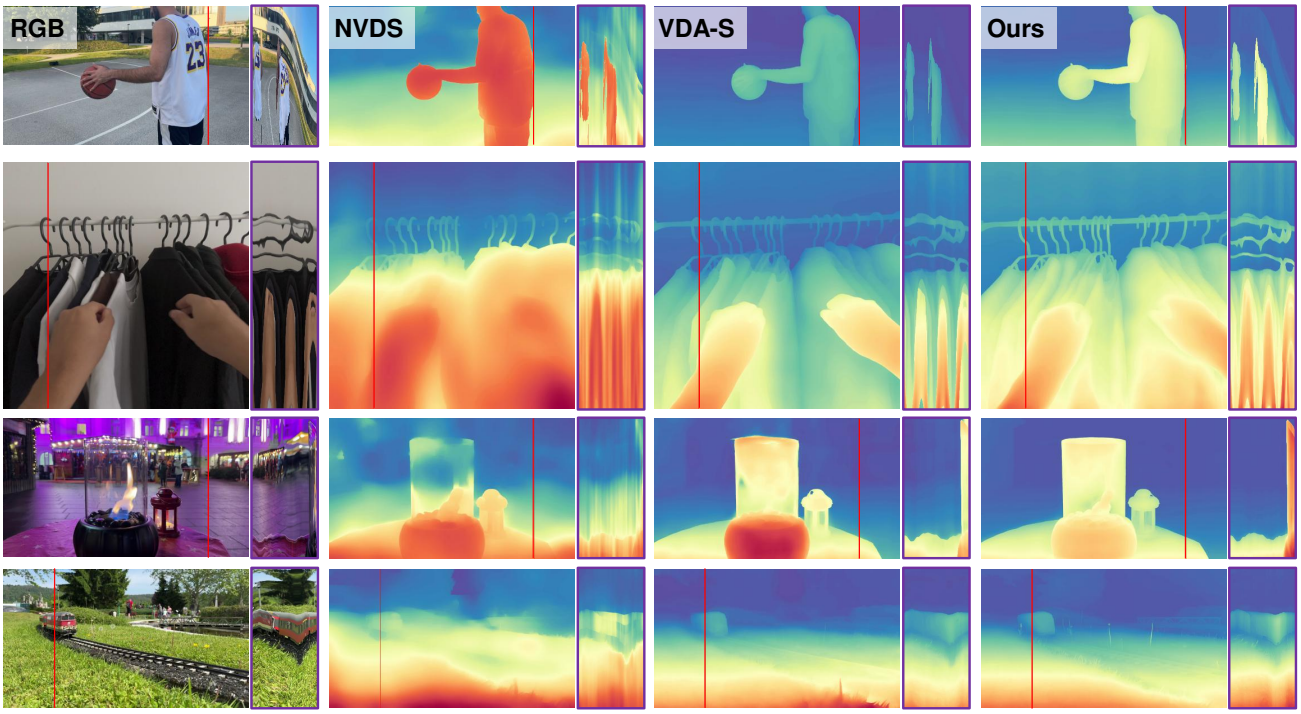

Figure 1. **Qualitative comparison of VDPP with NVDS (post-processing) and VDA (end-to-end).** We compare video depth estimation by accumulating results along the time axis. The temporal profile (purple box) is a slit-scan image generated by stacking the red pixel line from each frame sequentially over time. (Top row) A basketball player runs forward and then moves away. Both **NVDS** and **VDA** show minimal depth change, whereas **VDPP** correctly captures the motion. (Second row) A hand moves through clothes on a hanger. The slit-scan image reveals **NVDS** suffers from severe **flickering**, even in areas that should be static. **VDA** lacks fine-grained depth expression. (Third row) Flames flicker inside a transparent fire pit. **NVDS** again exhibits heavy flickering, and **VDA** fails to distinguish the depth between the front and back of the pit. **VDPP** clearly shows the depth difference with almost no flickering. (Bottom row) A train moves laterally across tracks. **NVDS** fails to estimate depth during the fast motion, while **VDA** loses the details of the tracks. In all scenarios, **VDPP** demonstrates superior **spatial accuracy** and **temporal coherence**.

appears to over-smooth the scene, blurring the individual clothing items into a single mass. VDPP uniquely achieves both goals: a stable temporal profile (a straight line in the static regions) combined with sharp, distinct spatial details.

- **In the Fire Pit scene,** the challenge is twofold: transparency and stochastic (non-rigid) motion from the flames. NVDS breaks down into uncontrolled flickering, unable to handle the non-rigid motion. VDA-S struggles with depth ambiguity, collapsing the depth between the front and back of the transparent pit into a single ambiguous plane. VDPP is the only method to correctly distinguish the distinct foreground and background depth layers while maintaining coherence.
- **In the Train scene,** the model is tested against fast lateral motion. NVDS's depth map almost completely disintegrates, demonstrating an inability to process the rapid pixel displacement. VDA-S maintains temporal stability but at a significant cost: it loses the critical geometric integrity of the tracks, blurring them into a non-descript surface. VDPP again excels, preserving the sharp spatial details of the individual tracks without introducing temporal artifacts or motion blur.

These visual results collectively serve as strong qualitative evidence. They confirm that VDPP is not making a simple compromise, but is fundamentally resolving the conflict between spatial accuracy and temporal coherence.

B.2. Spatial Fidelity and Fine Detail Analysis

While the previous section focused on temporal stability, this section provides a detailed analysis of spatial fidelity. End-to-end models like VDA often achieve temporal smoothness by over-smoothing spatial details, while post-processing methods like NVDS can fail to resolve fine structures. Figure 2 compares VDPP against NVDS, VDA-S, and the SOTA VDA-L in seven challenging scenarios focused on fine-grained detail.

- **In the Waterfall scene,** the challenge is capturing fast-

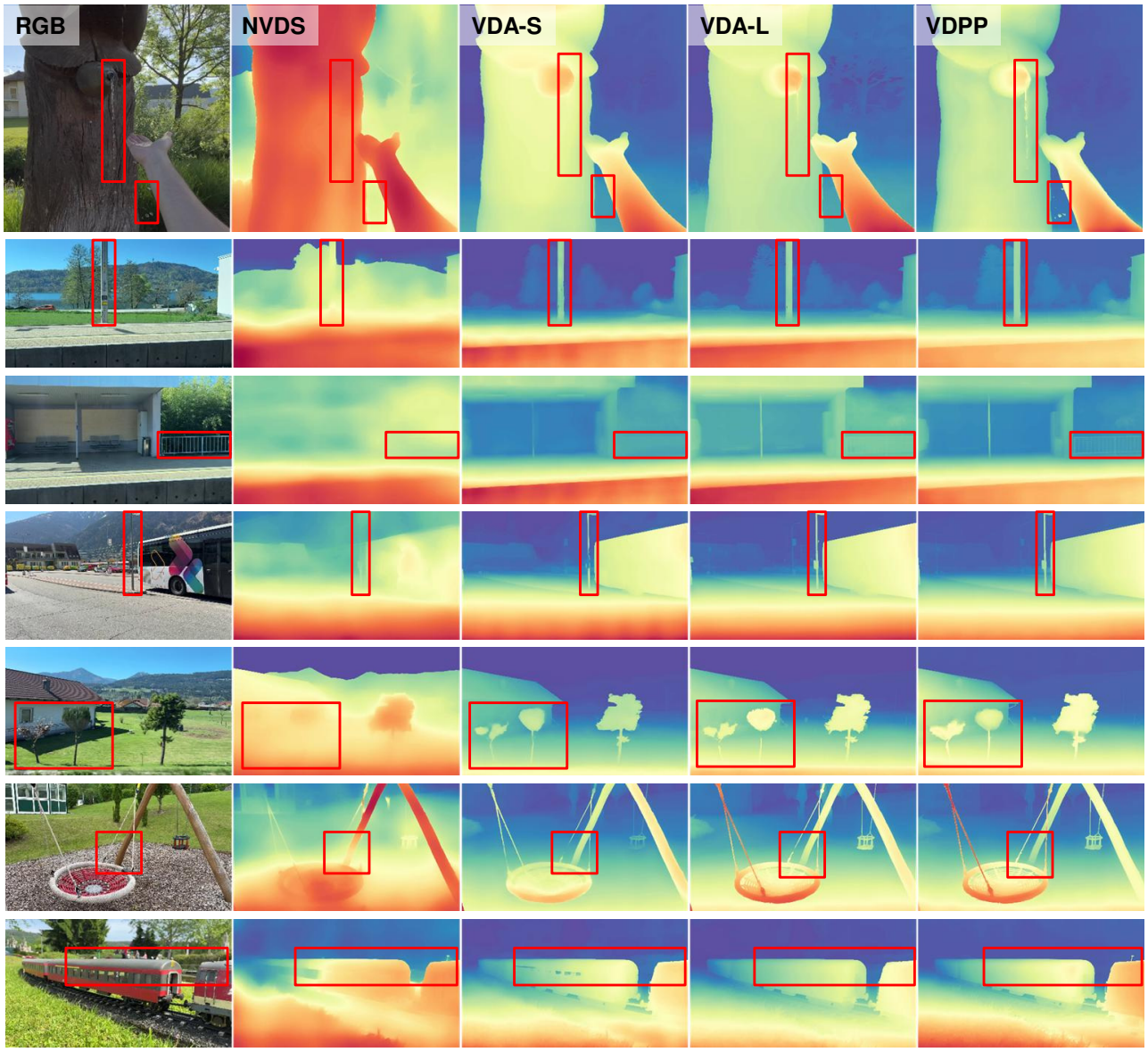

Figure 2. **Qualitative comparison of fine-grained spatial fidelity across seven challenging scenarios.** From left to right: RGB Input, NVDS, VDA-Small, VDA-Large (SOTA), and our VDPP. **(Top row)** The *Water falling* scene tests the ability to resolve fast-moving, non-rigid water and individual droplets. **(Second row)** The *Power Lines* scene highlights the challenge of preserving thin, static structures. **(Third row)** The *Fence* scene evaluates the separation of fine foreground posts from a complex background. **(Fourth row)** The *Bus Stop* tests robustness to dynamic background motion (passing buses) behind a thin object. **(Fifth row)** The *Street Trees* scene challenges the model to maintain structural integrity during complex camera and background motion. **(Sixth row)** The *Swing* scene focuses on resolving intricate details like thin ropes against a main structure. **(Bottom row)** The *Train* scene tests the difficult combination of transparent surfaces (windows) and fine details (antennas). In all examples, VDPP demonstrates a superior ability to preserve geometric integrity where other methods introduce artifacts, over-smoothing, or structural breaks.

moving, non-rigid water. Most methods fail to represent the stream's depth. Even the SOTA VDA-L, while capturing the main stream, fails to resolve the individual falling droplets. VDPP is the only method to clearly represent both the continuous stream and the distinct, fast-moving droplets, demonstrating its superior depth expression.

- **In the Power Lines Structure scene,** when zoomed in, VDPP is the only method to render a clean, complete power pole. NVDS fails to identify the structure, while both VDA-S and VDA-L produce results with "holes"

Table 1. **Performance and Speed Comparison on Sintel (384x384)**. This table combines speed (FPS, TPF) and performance (AbsRel, δ_1, TGSE) metrics. I2D refers to Image to Depth conversion time, and D2V refers to Depth to Video conversion time. The numbers in parentheses indicate the relative performance compared to the baselines(DPT, DAv2): green denotes improvement, and red denotes degradation. Among post-processing methods, **Bold** and underlined denote the best and next-best performance, respectively.

Method	Speed				Performance Metrics		
	FPS ↑	Total (ms)	I2D (ms)	D2V (ms)	AbsRel↓	δ_1 ↑	10^2·TGSE↓
End-to-end							
VDA-S	65.6	15.3	–	–	0.355	0.681	1.760
VDA-L	18.0	55.5	–	–	0.324	0.729	1.424
Image to Depth							
DPT-L	129	7.7	7.7	–	0.548	0.610	2.545
DAv2-B	141	7.1	7.1	–	0.358	0.678	1.809
DAv2-L	75	13.3	13.3	–	0.352	0.671	1.711
Post-processing							
DPT-L + NVDS	4	225.0	7.7	217.2	0.471 (-14.0%)	0.615 (+0.8%)	2.407 (-5.4%)
DAv2-B + NVDS	4	224.3	7.1	217.2	0.387 (+8.1%)	0.668 (-1.5%)	2.304 (+27.4%)
DAv2-L + NVDS	4	230.5	13.3	217.2	0.384 (+9.1%)	0.674 (+0.5%)	2.180 (+27.4%)
DPT-L + Ours	104	9.6	7.7	1.8	0.559 (+2.0%)	0.633 (+3.8%)	2.207 (-13.3%)
DAv2-B + Ours	**112**	**8.9**	7.1	1.8	**0.312** (-12.9%)	**0.731** (+7.8%)	1.756 (-3.0%)
DAv2-L + Ours	66	15.1	13.3	1.8	0.319 (-9.3%)	0.713 (+6.4%)	**1.704** (-0.4%)

where depth is missing, revealing instability even in static structures. VDPP maintains both spatial integrity and temporal stability.

- **In the Fence scene,** against a complex background of dense trees, VDPP is the only method to accurately capture the fine geometry of the individual fence posts and the gaps between them. All competitors, including VDA-L, render the general shape of the fence but completely fail to resolve these fine-grained structural details.
- **In the Bus Stop Sign scene,** the challenge is a static thin pole against a dynamic background of passing buses. This causes both NVDS and VDA-S to incorrectly blend the pole's depth with the background in several frames. VDA-L avoids blending but introduces significant depth artifacts not present in the original scene. VDPP remains robust, producing a clean and accurate representation of the sign throughout the sequence.
- **In the Street Trees scene,** viewed from a moving bus, the complex and dynamic parallax causes NVDS, VDA-S, and even VDA-L to fail. As a house passes in the background, their representations of the smaller tree trunks "break" or become disconnected. VDPP is the only method to handle this challenging dynamic scene, correctly preserving the geometric integrity of the trees.
- **In the Swing scene,** All competing methods, including VDA-L, fail to distinguish the thin ropes of the swing from its main structure, blending them into a single object. VDPP successfully preserves the depth separation and resolves these fine structures.
- **In the Train scene,** NVDS and VDA-S fail a classic challenge: transparent windows. They are unable to estimate depth for the window, likely due to the visible background. VDA-L correctly handles the transparent window but fails to capture finer details like the train's antenna. VDPP successfully resolves both the complex transparent surface and the fine-grained antenna detail.

These results confirm that VDPP's refinement process excels at preserving and enhancing spatial fidelity, overcoming the over-smoothing tendencies of VDA and the structural failures of NVDS.

C. Comparison on Small Resolution

To further validate VDPP's practical viability, we conducted a comprehensive benchmark at a fixed 384x384 resolution on the Sintel dataset [1]. This setup mimics common industrial applications where computational and memory budgets are strictly limited, forcing the use of downscaled inputs. The results, detailed in Table 1, decisively demonstrate VDPP's ability to resolve the performance trilemma, led by its state-of-the-art speed.

State-of-the-Art Speed: The primary advantage of VDPP is its exceptional processing speed. As shown in Table 1, our DAv2-B + Ours configuration achieves an extremely high 112 FPS. This is massively faster than all end-to-end competitors, running approximately 6.2x faster than VDA-L and 1.7x faster than VDA-S. Even our DAv2-L + Ours setup (66 FPS) surpasses VDA-S. This speed is en-

Figure 3. **Qualitative validation of VDPP on real-world LiDAR data from the KITTI dataset [4].** We compare the "Raw Inpainted" depth (from LiDAR depth completion) against our refined "VDPP" output across four driving sequences. The red boxes highlight VDPP's ability to **suppress artifacts** inherent to the inpainting process, such as horizontal streaking. The blue boxes demonstrate how VDPP **enhances depth expression**, clarifying object boundaries and improving geometric detail where the raw input was poorly defined. This validation confirms VDPP is not limited to refining single-image models but also serves as a powerful plug-and-play post-processor, significantly improving depth quality from real-world 3D sensor pipelines.

abled by our highly efficient D2V (Depth-to-Video) module, which adds negligible overhead.

Superioir Spatial Accuracy Despite High Speed: Crucially, this SOTA-level speed is not a trade-off. VDPP achieves this while simultaneously delivering state-of-the-art spatial accuracy. Both our DAv2-B + Ours (0.312 AbsRel) and DAv2-L + Ours (0.319 AbsRel) configurations are more accurate than the heavyweight VDA-L (0.324 AbsRel). This proves VDPP not only stabilizes the base models but significantly refines their geometric quality, a feat the competing post-processing method (NVDS) fails, as it severely degrades accuracy.

Competitive Temporal Coherence: Furthermore, this unprecedented combination of speed and accuracy is achieved without sacrificing temporal stability. Our temporal coherence (TGSE) scores are better than VDA-S and remain highly competitive with the VDA-L baseline. This confirms that our framework's gains in speed and accuracy do not introduce the flickering artifacts that plague other fast methods.

In summary, this 384x384 experiment confirms VDPP's exceptional value for practical deployment. It is the only framework that delivers SOTA speed, SOTA spatial accuracy, and competitive temporal coherence in a single, lightweight package, confirming its suitability for real-world industrial applications.

D. Experiment for Real Depth Sensors

As introduced in the main paper, one of VDPP's most significant advantages is its sensor-agnostic, depth-only architecture. This allows it to directly refine data from real-world depth sensors, a critical capability for robotics and autonomous systems that RGB-dependent methods lack. This section provides a more detailed validation of this capability using the KITTI dataset [4], as shown in Figure 3.

The "Raw Inpainted" column shows the input provided to our network. This is not the raw, sparse LiDAR point cloud itself, but rather a dense depth map created by applying a standard depth completion (inpainting) algorithm. While dense, this inpainted data suffers from two major problems: (1) significant artifacts from the completion process, such as horizontal streaking and noise, and (2) poor geometric definition, especially for distant or moving objects where the original point cloud was thinnest.

VDPP demonstrates its effectiveness as a powerful spatio-temporal refinement module for this noisy sensor data. It actively filters the inpainting noise while preserving the underlying geometric signal. In areas where the inpainted map shows blurred or poorly defined structures (like vehicle boundaries or distant scenery), VDPP refines the geometry, sharpens edges, and improves the depth separation between objects.

This validation on real sensor data confirms that VDPP is a truly scalable and practical solution. It can be seamlessly integrated into existing robotics stacks, not only to stabilize monocular depth estimation but also to serve as a high-performance refinement layer for noisy data from LiDAR or ToF sensors, significantly improving the quality and reliability of the perception system.

E. Memory Efficiency Validation on Edge Devices

To thoroughly evaluate VDPP's deployability on resource-constrained hardware, we conducted extensive memory profiling experiments on the NVIDIA Jetson Orin Nano (8GB), a widely-used edge computing platform for robotics and autonomous systems.

E.1. Experimental Setup

All experiments were conducted under identical conditions: Ubuntu 20.04, CUDA 11.4, with system idle memory baseline at 2.4 GB. We monitored memory usage using `jtop` and system resource monitors in real-time during inference from the Sintel dataset[1] at native resolution.

E.2. Successful Deployment Analysis

In sharp contrast, Figure **??** (main paper) captures the state of the system during sustained, real-time VDPP operation. This success is directly attributable to our geometric downsampling, which processes a compact manifold rather than full-resolution attention maps. By doing so, VDPP avoids the critical failure point identified in the failure case: the allocation of contiguous memory blocks that exceed device capacity. Crucially, it is worth noting that our deployment on the edge device does not rely on any additional model compression or optimization techniques, such as knowledge distillation or weight quantization. VDPP achieves practical inference speeds out-of-the-box. This demonstrates the inherent efficiency and 'plug-and-play' scalability of our framework, making it highly suitable for real-world applications where complex post-training optimizations are often costly or degrade baseline accuracy.

E.3. Failure Case

Figure 4 screenshots systematic failures of competing methods and shuts down the device. It Crashed at initialization when allocating attention matrices, estimated memory requirement (VDA-L, VDA-S). And VDPP with out downsampling also crashed at initialization, confirming downsampling is critical. Profiling reveals that full-resolution attention mechanisms allocate contiguous memory blocks that exceed device capacity, making them fundamentally incompatible with edge deployment regardless of optimization efforts.

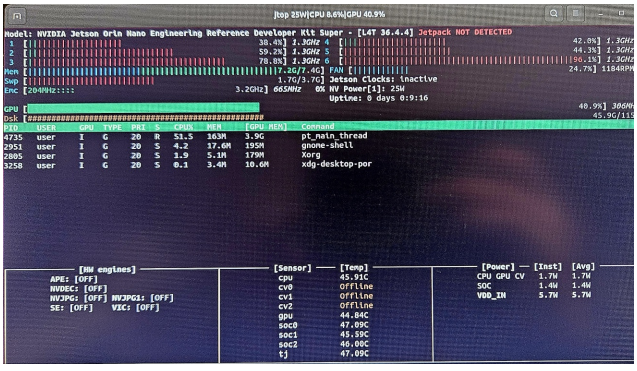

Figure 4. Failed deployment attempts of VDA-L, VDA-S, and NVDS on NVIDIA Jetson Orin Nano, showing Out-Of-Memory errors and system crashes.

These results demonstrate a categorical difference: VDPP enables a class of applications (edge-based video depth) that existing methods cannot support. The 2 GB memory headroom also allows co-deployment with other perception modules, critical for practical autonomous systems.

F. Analysis of the TGSE Metric

In this paper, we proposed a novel metric, **Temporal Gradient Squared Error (TGSE)**, to evaluate the temporal consistency of video depth estimation. We conducted a sensitivity analysis in a controlled environment to demonstrate that existing spatial accuracy metrics (e.g., AbsRel, δ_1) fail to properly capture temporal inconsistencies. This result highlights the necessity of a dedicated temporal metric and validates the reliability of TGSE.

F.1. Experimental Setup

We injected artificial temporal artifacts into the Ground Truth depth maps of all scenes in the Sintel dataset [1], which possesses perfect temporal consistency, to observe how the metrics respond. To simulate 'Flicker', the most common and detrimental artifact in video depth estimation, we applied **Frame-wise Multiplicative Scale Perturbation**.

For each frame t, we generated a perturbed depth $\tilde{D}^{(t)}$ by multiplying the Ground Truth depth $D_{GT}^{(t)}$ by a random scalar $s^{(t)}$:

$$\tilde{D}^{(t)} = D_{GT}^{(t)} \cdot s^{(t)}, \quad \text{where } s^{(t)} \sim \mathcal{U}(1-\lambda, 1+\lambda) \quad (1)$$

where $\mathcal{U}$ denotes a uniform distribution, and λ represents the perturbation level. We incrementally increased λ from 0.0 to 0.5 and measured the changes in TGSE and AbsRel.

F.2. Quantitative Analysis

An ideal temporal consistency metric should exhibit a monotonic increase in error value as the perturbation intensity (λ) increases. Figure 5 (Left) presents the results of this experiment.

Robustness of TGSE: Our TGSE (solid red line) maintains a strict monotonic increasing trend as the perturbation level rises. This demonstrates that TGSE accurately quantifies the intensity of flicker artifacts. The temporal gradient amplifies the variance of independent perturbation noise between frames by a factor of two, facilitating detection. Furthermore, the L2 Norm imposes a high penalty on large outliers, preventing score degradation due to stochastic effect.

Blindness of Spatial Metrics: In contrast, the standard spatial metric, AbsRel (dashed blue line), shows unpredictable fluctuations. Most critically, despite the perturbation level

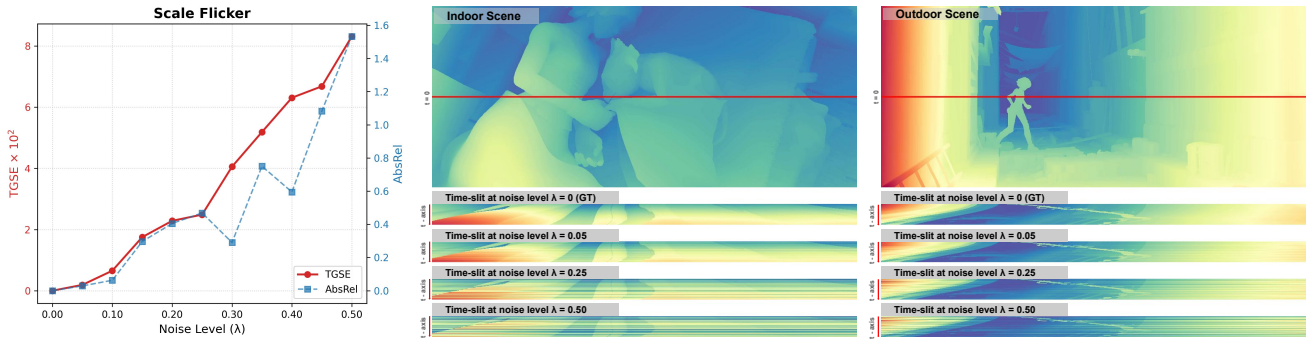

Figure 5. **Sensitivity and Qualitative Analysis of Scale Flicker.** We analyzed metric behavior and visual quality changes while increasing the intensity of Scale Perturbation (λ) from 0% to 50%. **Quantitative Sensitivity (Left):** The graph shows metric changes according to perturbation levels. Our **TGSE (red line)** shows a strict **monotonic sensitivity**, accurately reflecting artifact severity. Conversely, **AbsRel (blue line)** fluctuates irregularly and exhibits unreliable behavior, specifically showing **a sharp drop at perturbation level 0.3 despite the intensified flicker. Qualitative Temporal Profiles (Right):** Visualization of temporal profiles (slit-scans) for representative **Indoor (left)** and **Outdoor (right)** scenes from the Sintel dataset. The top of each block is the Reference Depth Map, with a red horizontal line indicating the slice position. The bottom stack shows temporal slices for Ground truth, $\lambda = 0.05$, $\lambda = 0.25$, and $\lambda = 0.5$ (perturbation levels) in order. As perturbation increases, temporal discontinuities in the form of **horizontal banding** become evident. TGSE strongly correlates with this visual collapse, while AbsRel fails to capture it consistently.

increasing from 0.25 to 0.30, the AbsRel value paradoxically drops sharply. This indicates that spatial metrics are vulnerable to stochastic alignment of per-frame pixels independently and may fail to correctly assess the degradation of temporal quality.

F.3. Qualitative Analysis

Figure 5 (Right) visualizes the temporal profile (slit-scan) of the experiment. As the perturbation level increases from 0 to 0.5, discontinuities in the form of **horizontal banding** become more pronounced in the temporal slice.

TGSE scores increase in strong correlation with this temporal degradation, whereas AbsRel fails to consistently reflect it. Therefore, a dedicated temporal metric like TGSE is essential alongside spatial metrics for the fair evaluation of video depth estimation models.

References

[1] Daniel J Butler, Jonas Wulff, Garrett B Stanley, and Michael J Black. A naturalistic open source movie for optical flow evaluation. In *European conference on computer vision*, pages 611–625. Springer, 2012. 4, 6

[2] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2. *arXiv preprint arXiv:2001.10773*, 2020. 1

[3] 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*, pages 22831–22840, 2025. 1

[4] 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. 5

[5] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 1

[6] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 1

[7] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 1

[8] 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. 2020. 1

[9] Yiran Wang, Min Shi, Jiaqi Li, Zihao Huang, Zhiguo Cao, Jianming Zhang, Ke Xian, and Guosheng Lin. Neural video depth stabilizer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9466–9476, 2023. 1

[10] 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. 1

[11] Yang Zheng, Adam W. Harley, Bokui Shen, Gordon Wetzstein, and Leonidas J. Guibas. Pointodyssey: A large-scale synthetic dataset for long-term point tracking. In *ICCV*, 2023. 1