DEPTH PRO 2: SHARP AND TEMPORALLY CONSISTENT METRIC MONOCULAR DEPTH

Anonymous authors
Paper under double-blind review

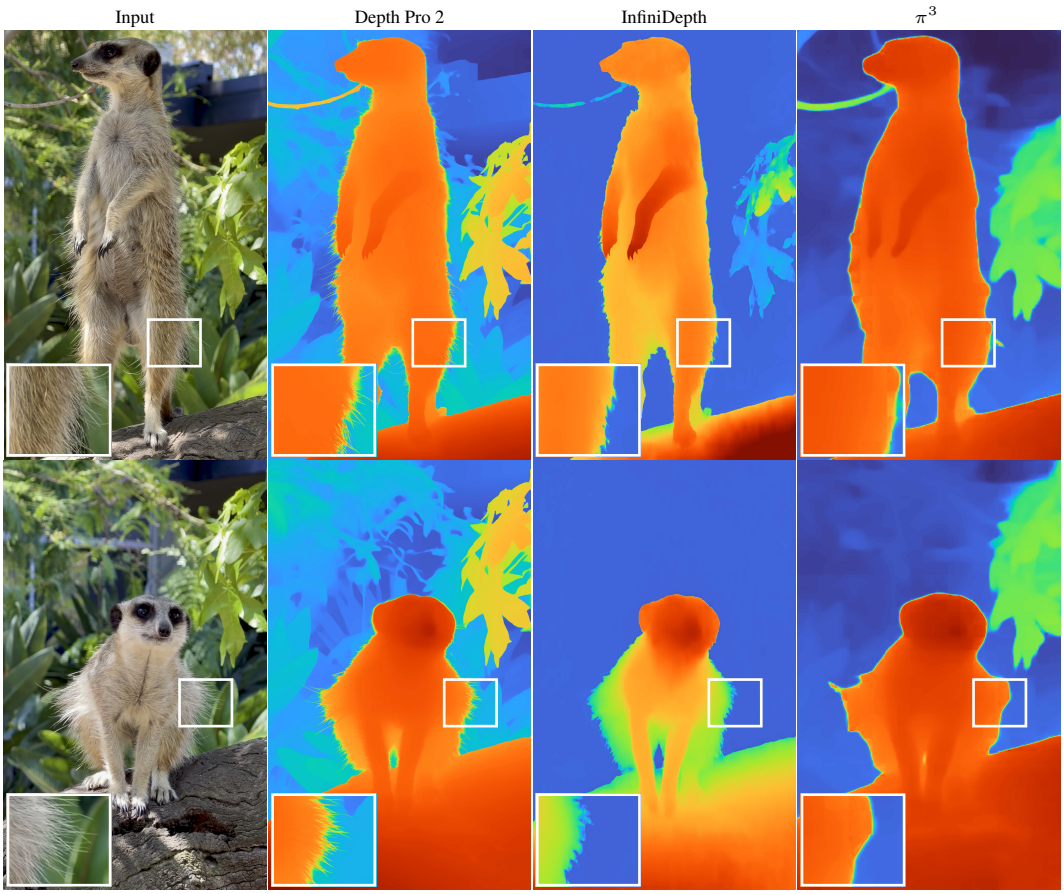

Figure 1: Left to right: two frames from a video (Pexels, 2026) with depth maps produced by Depth Pro 2, InfiniDepth (Yu et al., 2026), and π^3 (Wang et al., 2026b). Insets show fine details and consistent depth between frames for Depth Pro 2.

ABSTRACT

We present Depth Pro 2, an approach to estimating dense metric depth for images and video at full resolution. By analyzing prior approaches to monocular depth estimation, we find that they commonly trade off temporal consistency and boundary sharpness – multi-frame methods favor consistency, single-frame methods favor sharpness. To overcome this trade-off, we train a model that processes one or multiple images and predicts depth maps at native resolution. The network architecture is designed for simplicity and scalability, and is agnostic to whether the input is a single image or a video volume – the same network processes images and video. Depth Pro 2 produces depth maps that are metrically accurate, sharp, and temporally consistent all at once, outperforming prior work in all of these dimensions. We release a family of models that define a new Pareto frontier in terms of metric accuracy, sharpness, temporal consistency, and runtime.

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.

1 INTRODUCTION

More than a decade ago, the release of the Kinect sensor unleashed a wave of new research and applications (Zhang, 2012). Before its introduction, obtaining accurate dense depth in dynamic environments required specialized hardware and expertise. With the Kinect, countless research projects in computer vision, human-computer interaction (Shotton et al., 2011), and robotics (Henry et al., 2012) were spun up to leverage the cheap and comparably robust sensor. Replacing hardware depth sensors with software that delivers comparable or even higher metric accuracy, density, and temporal consistency could have an even bigger impact on downstream applications, as every existing camera could be turned into a depth camera, and all existing images could be lifted and processed in 3D. Research into depth estimation from monocular images has progressed towards this goal, from domain-specific depth estimation (Eigen et al., 2014; Saxena et al., 2009) to generalization across domains via relative depth (Ranftl et al., 2022), metric depth (Yin et al., 2023; Piccinelli et al., 2024), and pixelwise delineation of geometric detail (Li et al., 2024a; Bochkovskii et al., 2025).

Another line of work focuses on dense depth estimation for monocular video, processing multiple frames jointly in order to increase temporal consistency (Chen et al., 2025; Lin et al., 2026; Wang et al., 2026b). Reasoning over multiple frames at once, however, comes at the price of increased memory consumption and latency, especially when attending across frames with softmax attention, whose computational complexity is quadratic in the number of inputs (Vaswani et al., 2017). To limit memory footprint and latency while processing frames jointly for temporal coherence, methods commonly restrict per-frame resolution, which can compromise fine detail.

The trade-off made by these two lines of work is apparent in Figure 2, which scores recent works on boundary sharpness and temporal consistency. Current methods tend to be either temporally consistent or sharp, with multi-frame processing strongly associated with the former.

We present an approach that estimates sharp, metrically accurate, and temporally consistent depth from single images and videos, operating at native input resolution (*e.g.*, 12 MP). It achieves unprecedented sharpness and temporal consistency (Figure 1).[1] This is enabled through three key components. First, we design a simple architecture that allows lightweight multi-frame processing. Second, we process individual frames at native resolution, which enables sharp boundary prediction. Third, we optimize the data and training pipeline for single- and multi-image inputs at arbitrary resolution. Taken together, these contributions yield an approach that outperforms prior work on zero-shot metric depth accuracy, boundary sharpness, and temporal consistency.

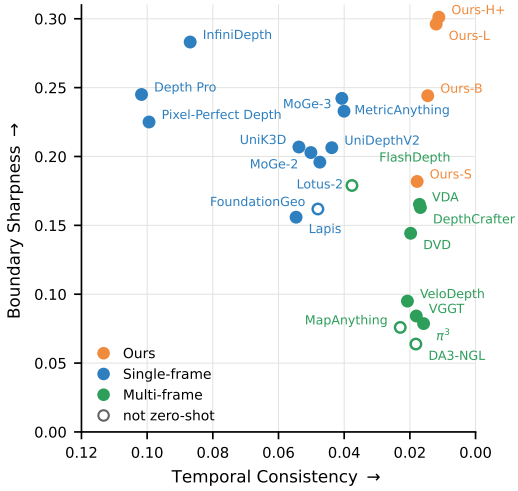

Figure 2: **Sharpness versus temporal consistency.** Depth Pro 2 outperforms prior work operating on a single frame or multiple frames on sharpness (Boundary F1 score (Bochkovskii et al., 2025)) and temporal consistency (Jitter, Sec. C.1). Results averaged over Sintel and Spring.

To support a broader range of applications that may have divergent priorities over sharpness, accuracy, and speed, we design a family of models from the same recipe through simple architecture scaling. Our largest model outperforms prior work on metric accuracy, sharpness, and temporal consistency, and processes 1080p video at 360 ms per frame and VGA video at 10 ms per frame on a B200 GPU. Our smallest model is an order of magnitude faster, processing 1080p video at 25 fps and VGA video at 1 ms per frame. Overall, the Depth Pro 2 family defines a new Pareto frontier across metric accuracy, sharpness, temporal consistency, and speed. We will release all models.

[1] Please see the anonymous website, anonymous2721692.github.io/curly-octo-dollop/ for videos and comparisons.

2 RELATED WORK

The frontier of monocular depth estimation has advanced significantly since Saxena et al. (2009) along multiple axes, including (1) *zero-shot*: generalizing to domains never seen during training; (2) *metric*: recovering metric depth despite the inherent scale ambiguity; (3) *sharp*: resolving fine details around thin structures; and (4) *consistent*: remaining temporally stable when applied to video. Below, we review key works along each axis.

Zero-shot metric depth. Early works (Eigen et al., 2014; Eigen & Fergus, 2015) regress metric depth on a per-domain basis and do not generalize well beyond their training distribution. Subsequent work has tackled two central challenges. (1) *Training data.* Limited sensor coverage and costly capture restricted both diversity and scale. Li & Snavely (2018) obtain large-scale relative depth supervision from Internet photo collections using SfM and MVS, while Ranftl et al. (2022) train across heterogeneous datasets with scale-and-shift-invariant (SSI) losses to improve cross-domain generalization. More recent methods scale supervision further through pseudo-labeling and synthetic data (Yang et al., 2024a;b; Lin et al., 2026; Wang et al., 2025c). In contrast, most of our training data is synthetic, with layouts and assets created at metric scale, and we train our model in a single stage without teacher models or pseudo-labels. (2) *Metric scale.* Scale is inherently ambiguous from a single image and can vary by orders of magnitude across scenes. Yin et al. (2023) address this by predicting depth in a camera-normalized canonical space, while Piccinelli et al. (2024) regress metric 3D points conditioned on a dense camera representation. Depth Pro predicts focal-length-normalized inverse depth together with focal length, and combines metric and SSI supervision (Bochkovskii et al., 2025). Our approach is closest to Depth Pro.

Sharp depth. A line of work processes images in tiles at multiple scales to obtain sharper depth predictions (Miangoleh et al., 2021; Li et al., 2024a;b; Bochkovskii et al., 2025). Kong et al. (2026) lift point estimates into sparse voxels and apply sparse 3D convolutions to iteratively refine local geometry and sharpness. Yu et al. (2026) represent depth as a neural implicit field that can be queried at arbitrary resolution, but require sparse metric depth as additional input to predict dense metric depth. Ke et al. (2024); Pavlovic et al. (2026); He et al. (2026) leverage latent diffusion priors to produce depth with sharp boundaries, while Xu et al. (2025) perform diffusion directly in pixel space to reduce flying pixels at depth discontinuities. In contrast, our model contains no module designed specifically for boundary sharpness; instead, we rely on a vision transformer backbone with RoPE (Heo et al., 2024) — specifically, a pretrained DINOv3 backbone (Siméoni et al., 2026) — to produce well-localized features when given high-resolution input.

Consistent depth. Many recent approaches improve video-depth consistency by adding temporal modules to single-frame depth models or by leveraging video diffusion priors. The former includes plug-and-play stabilization (Wang et al., 2023), spatiotemporal heads (Chen et al., 2025), and recurrent or propagation modules for streaming input (Piccinelli et al., 2026; Chou et al., 2025; Wang et al., 2025b). The latter finetunes pretrained video diffusion models for temporally consistent depth prediction (Hu et al., 2025; Shao et al., 2025; Yang et al., 2025b), or adapts them into deterministic single-step regressors (Zhang et al., 2026). Instead of relying on custom modules or computationally heavy video diffusion priors, we simply adapt a single-image ViT for multi-view processing through cross-attention across tokens from other views.

Another line of work takes inspiration from classic 3D reconstruction, and optimizes depth jointly with camera poses (Wang et al., 2024; 2025a; 2026a;b; Lin et al., 2026). To improve the consistency of reconstructed geometry, these methods are commonly optimized for handling large sets of views, which in turn limits their ability to process images at high resolution, a key requirement for sharp predictions. In contrast, our model estimates metric depth from a flexible number of input frames and processes them at native resolution to produce significantly sharper occlusion boundaries.

Reusing single-view attention across views. A common way to extend image-pretrained transformers to multiple images is to let existing self-attention layers attend over tokens from all views, reusing the pretrained weights without adding parameters. Variants of this idea have been used in video transformers (Carreira & Zisserman, 2017; Bertasius et al., 2021; Arnab et al., 2021), multi-view recognition (Chen et al., 2021; Lee et al., 2025), and generation (Khachatryan et al., 2023; Shi et al., 2024). Most closely related, Depth Anything 3 (Lin et al., 2026) rearranges tokens so that pretrained DINOv2 attention operates across views for feed-forward reconstruction. We take inspi-

ration from this line of work to adapt our ViT encoder to process multiple frames jointly. Unlike these approaches, we design our network to operate at native resolution.

3 METHOD

3.1 BOUNDARY SHARPNESS VERSUS TEMPORAL CONSISTENCY

To understand the current trade-offs in monocular depth estimation, we evaluate representative state-of-the-art methods for metric accuracy, boundary sharpness, and temporal consistency. We include leading metric-depth methods (Wang et al., 2025c; Kong et al., 2026; Piccinelli et al., 2025b;a; Liu et al., 2026b), sharp single-image and diffusion-based methods (Bochkovskii et al., 2025; Yu et al., 2026; Ma et al., 2026; He et al., 2026; Xu et al., 2025; Pavlovic et al., 2026), feed-forward reconstruction models (Wang et al., 2026b; 2025a; Keetha et al., 2026; Lin et al., 2026), and video-depth methods (Hu et al., 2025; Chou et al., 2025; Piccinelli et al., 2026). We evaluate depth accuracy (δ_1), boundary sharpness (Boundary F1), and temporal consistency (Jitter, Appendix C.1), using the strongest variant of each model. For sharpness and temporal consistency, we use Sintel (Butler et al., 2012) and Spring (Mehl et al., 2023), which provide high-resolution videos with spatiotemporally dense ground truth.

We find no clear relationship between architecture and metric accuracy, but a strong trade-off between boundary sharpness and temporal consistency. Figure 2 shows that single-frame methods are generally sharper, while multi-frame methods are more temporally consistent. We attribute this trade-off to competing memory demands. Multi-frame methods use additional context to align predictions across frames, but attending over multiple frames substantially increases memory consumption (Wang et al., 2025a; 2026a). Single-frame methods can instead devote more memory to higher spatial resolution, which benefits boundary sharpness (Bochkovskii et al., 2025; Yu et al., 2026). Thus temporal consistency and boundary sharpness compete for the same memory budget.

3.2 ARCHITECTURE

Overview. The architecture of our depth estimation network is depicted in Fig. 3. It consists of a multi-view transformer that ingests a single image or multiple frames of a video, processes them jointly, and decodes them to an inverse depth map per input image or frame. Akin to prior work, our depth estimation network predicts inverse depth normalized by focal length, and we combine it with a known or estimated focal length to predict metric depth (Bochkovskii et al., 2025). A separate focal length estimator processes the input image(s), and predicts a focal length per input image. Whenever the ground-truth focal length is provided, *e.g.* through EXIF data, we use it instead of running the focal length estimator.

Lightweight multi-frame depth estimation network. We design our depth estimation network to be lightweight and to flexibly handle a variable number of images, starting from a ViT variant pretrained for single-image processing (Dosovitskiy et al., 2021; Siméoni et al., 2026). Following the insight that strong temporal consistency requires an architecture that aligns predictions across frames, our model shares information across jointly processed frames via cross-attention, alternating blocks of within-frame self-attention and cross-frame attention (Wang et al., 2024) (See Fig. 3, ①). Since early blocks in multi-view transformers tend to encode low-level features and only later blocks approximate feature matching (Starý et al., 2025), we restrict the first $1/3$ out of L transformer blocks to self-attention operating only over tokens from the same frame for higher efficiency. For the remaining $2L/3$ blocks, we alternate between within- and cross-frame attention. For each token, we only encode the spatial position with the frame and omit any encoding of frame order, making our network invariant to permutations of frames. This allows flexible processing of any number of frames and directly reusing the attention weights from single-image pretrained encoder checkpoints (Lin et al., 2026) without adding transformer blocks that need to be trained from scratch for multi-view processing.

Native resolution processing. In Section 3.1 we observed that boundary sharpness correlates with the resolution at which frames are processed. To maximize boundary sharpness, we design our method to operate at full resolution – the native resolution of the input image or video. To that end, we initialize our multi-view transformer from DINOv3 weights, as DINOv3 is trained specifically to

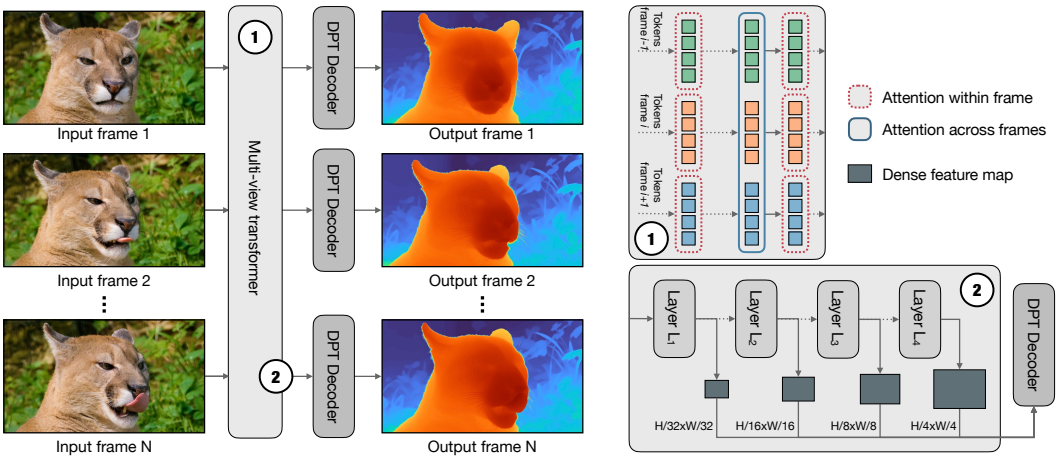

Figure 3: **Architecture.** Our depth estimation network consists of a multi-view transformer that ingests one or multiple input frames and processes them jointly, and a DPT decoder, which predicts an inverse depth map per input frame.

produce dense, high-resolution features (Siméoni et al., 2026). An input image is split into patches of size 16, and each patch embedded into tokens. We encode each token's spatial position through a 2D rotary embedding over their coordinates. Instead of normalizing coordinates per axis (Siméoni et al., 2026), which would discard any information over the image's aspect ratio, we divide both axes by the longer side. The position (x, y) of a patch (i, j) within a grid of size $h \times w$ is thus

$$y' = 2(i + \tfrac{1}{2})/\max(h, w) - 1, \quad x' = 2(j + \tfrac{1}{2})/\max(h, w) - 1$$

DPT decoder. A DPT decoder (Ranftl et al., 2021) decodes features extracted per frame to inverse depth maps, shown in Fig. 3, ②. Following DPT, we extract tokens after several transformer blocks, assemble them into feature maps of different resolution, and project them to a common channel dimension. Depending on the size of the multi-view transformer, we adjust the positions at which we extract features, and scale the number of channel dimensions the decoder operates on. We refer to Table 5 for the exact specification. To upsample feature maps within the decoder, we replace bilinear upsampling with transposed convolutions (Bochkovskii et al., 2025).

Focal length estimation. Our approach requires focal length to recover metric scale. When not provided, we train a separate field-of-view (FoV) estimator that serves all four of our depth models. The estimator is a DINOv3 ViT-L (Siméoni et al., 2026), finetuned end-to-end on Flickr photographs (Thomee et al., 2016) with EXIF focal lengths to predict FoV at a 256×256 resolution. For multi-frame input, we take the median predicted focal length over frames. Our FoV estimator is the most accurate on the zero-shot benchmark of Bochkovskii et al. (2025) (mean δ_{25} 87.9 vs. 71.2 for Depth Pro; Appendix B.4, Table 12). Appendix A.4 provides training and architecture details.

3.3 Training

Depth Pro 2 is trained on a diverse set of 19 datasets with a strong focus on synthetic ground truth, with the full list in Table 8. To preserve the sharpness of ground-truth depth maps, we avoid any interpolation of ground-truth depth during training, as averaging over depth discontinuities is a leading cause for flying pixels (Bochkovskii et al., 2025). To nevertheless expose our model to a broad set of image sizes and aspect ratios, we crop and sample images at varying strides. During training, we use ground-truth camera intrinsics (focal length) and adjust accordingly depending on how we crop or subsample. The efficacy of processing and training at native resolution (and avoiding interpolation) is confirmed through controlled experiments in the supplement.

To operate on images and videos, Depth Pro 2 is trained on clips with varying length N that are processed jointly. Images are treated as videos of length 1. For video and multi-view datasets, a clip contains up to 32 frames/views. We draw the source of each clip by temperature-scaled weights (T=2) (Arivazhagan et al., 2019), capping small sources to avoid overfitting.

Batching. To efficiently train with varying resolutions and clip lengths, we organize batches by tokens, and fill each GPU to a fixed budget of $b = 73,728$ tokens, which corresponds to *e.g.* 18 frames at 1024×1024. When one clip leaves budget unused, we pack several clips of equal resolution onto the same GPU. Different GPUs may train at different resolutions within a single step.

Training objective. Our loss is the unweighted sum of four terms:

$$\mathcal{L} = \mathcal{L}_{mae} + \mathcal{L}_{ssi} + \mathcal{L}_{mge} + \mathcal{L}_{mvc}. \tag{1}$$

$\mathcal{L}_{mae}$ is a trimmed $\mathcal{L}_1$ loss on focal-normalized inverse depth. It anchors metric depth, and we apply it only to sources with metric ground truth. $\mathcal{L}_{ssi}$ is the same trimmed $\mathcal{L}_1$ loss after normalizing prediction and ground truth independently to zero median and unit mean absolute deviation (Ranftl et al., 2022). We apply it to all sources. Both trimmed losses discard the τ fraction of pixels with the largest loss, with $\tau = 1\%$ on synthetic data and 10% on real data. $\mathcal{L}_{mge}$ matches first-order gradients of the normalized depth maps over a Gaussian pyramid of $M = 3$ scales (Bochkovskii et al., 2025). We apply it only to synthetic sources, where boundaries are exact. $\mathcal{L}_{mvc}$ enforces consistency across frames and is described below.

Multi-view self-consistency. To encourage consistency of predictions across frames, we derive a multi-view self-consistency loss $\mathcal{L}_{mvc}$, which we apply to random subsets of pixels. We take inspiration from the consistency loss of Luo et al. (2020), which penalizes inconsistencies across frames by warping points via their predicted depth and penalizing deviations from the predicted optical flow. To increase robustness, we replace optical flow estimation with correspondences derived from ground-truth poses. Let p be a pixel in frame i, $\hat{Z}(p)$ its predicted depth, and π the projection from world space to image space. Warping the pixel to frame j via the predicted depth and ground-truth camera poses yields the projected point $(p', z') = \pi_j(\pi_i^{-1}((p_i, \hat{Z}(p_i))))$. Assuming a static scene, the warped depth z' should equal the prediction at the projected location p'. We penalize deviations via

$$\mathcal{L}_{mvc} = \left|1/z' - 1/\hat{\mathbf{Z}}(p')\right|_{1^\tau}. \tag{2}$$

To mask out occluded pixels and moving objects, we warp pixels via their ground-truth depth and camera poses and discard all pixels whose warped depth and ground-truth depth differ by more than 2%. This allows applying the multi-view loss $\mathcal{L}_{mvc}$ to dynamic scenes as well.

4 Experiments

This section reports key results. The appendices contain further quantitative and qualitative results, controlled experiments, a runtime analysis, and training and evaluation details. It also contains evaluations on bokeh rendering and novel view synthesis, two downstream applications for monocular depth estimation.

To assess the efficacy of our method, we evaluate its performance in a zero-shot setting on metric depth accuracy, boundary sharpness, and temporal consistency, comparing against a comprehensive set of baselines. When comparing to prior work, we only include our strongest model, Depth Pro 2-H$^+$. We provide comparisons within the Depth Pro 2 family in Table 6 of Appendix A.1, and detail our evaluation protocols in Appendix C.

Figure 4 provides a qualitative comparison to prior work. To qualitatively assess temporal stability and sharpness, we encourage the reader to see the videos on the anonymous website, anonymous2721692.github.io/curly-octo-dollop/.

Zero-shot metric depth. We evaluate our model for zero-shot metric depth accuracy against single-image and multi-image baselines that predict dense *metric* depth. We report results on iBims (Koch et al., 2018), Middlebury (Scharstein et al., 2014), Booster (Ramirez et al., 2022), HAMMER (Jung et al., 2022), ETH3D (Schöps et al., 2017), KITTI (Geiger et al., 2012), NuScenes (Caesar et al., 2020), DIODE (Vasiljevic et al., 2019), Sintel (Butler et al., 2012), Spring (Mehl et al., 2023), and DyDToF (Sun et al., 2023). Ten datasets are adopted from the evaluation sets of Depth Pro (Bochkovskii et al., 2025) and MoGe-3 (Kong et al., 2026). We add DyDToF, a synthetic video dataset with dense metric ground truth and sharp boundaries. These datasets have been excluded from the training data for most methods and are therefore ideal for zero-shot benchmarking. We do not include datasets captured with Kinect sensors due to noisy, biased, and incomplete ground truth.

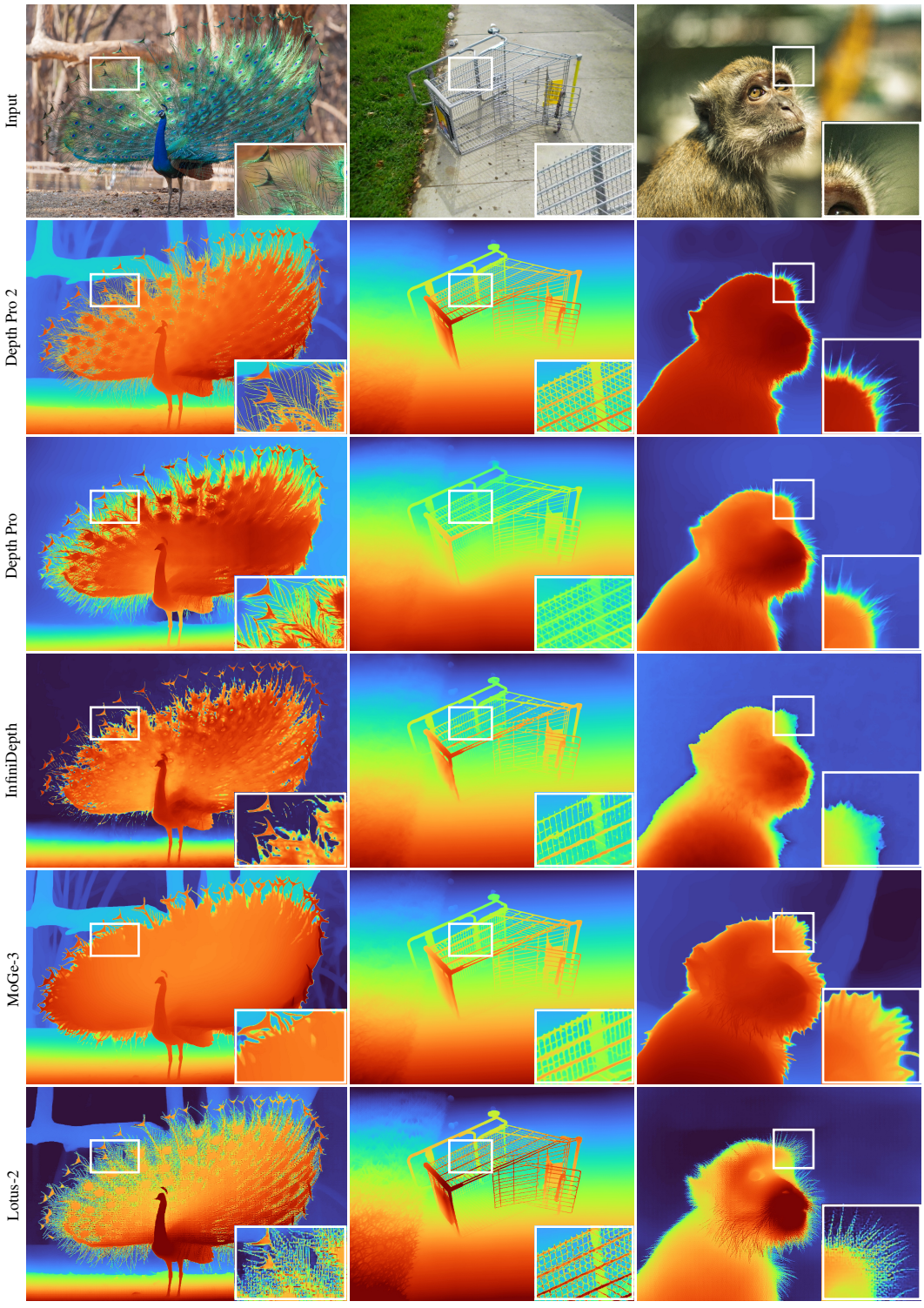

Figure 4: **Qualitative results.** Top to bottom: input images, followed by results from Depth Pro 2 and representative baselines: Depth Pro (Bochkovskii et al., 2025), InfiniDepth (Yu et al., 2026), MoGe-3 (Kong et al., 2026), and Lotus-2 (He et al., 2026). Images from Pexels (Pexels, 2026) (left), DIS-5K (Qin et al., 2022) (middle), and AM-2K (Li et al., 2022) (right).

Table 1: **Zero-shot metric depth accuracy, single image.** All methods receive a single frame. On video and multi-view datasets, each of the first 50 frames of a sequence is scored as a separate image. We report δ_1 (higher is better) under ground-truth and predicted intrinsics, rank methods per condition and dataset, and average the ranks over datasets (last column). Non-zero-shot results are gray. MetricAnything uses GT intrinsics only.

Method	Real, indoor								Real, indoor+outdoor				Real, outdoor				Synthetic						Avg. rank↓	
	iBims		M.bury		Booster		HAMMER		ETH3D		DIODE		KITTI		nuSc		Sintel		Spring		DyDToF			
	GT	pred	GT	pred	GT	pred	GT	pred	GT	pred	GT	pred	GT	pred	GT	pred	GT	pred	GT	pred	GT	pred	GT	pred
MapAnything (Keetha et al., 2026)	74.8	58.0	34.6	42.3	41.8	50.1	13.0	29.9	54.1	51.5	28.2	18.2	91.6	90.6	57.3	34.4	20.4	18.4	96.7	96.7	7.4	7.1	10.2	8.8
DepthPro (Bochkovskii et al., 2025)	88.1	82.9	72.2	75.1	56.7	59.0	71.7	55.2	46.3	41.0	45.4	40.6	84.8	82.7	56.5	60.5	36.5	30.1	12.5	16.7	67.2	27.4	7.6	6.4
MetricAnything‡ (Ma et al., 2026)	86.3	–	51.9	–	60.6	–	38.1	–	74.4	–	60.4	–	5.7	–	87.6	–	25.6	–	22.5	–	73.3	–	7.6	–
FoundationGeo (Liu et al., 2026b)	93.1	70.0	57.3	12.9	50.0	56.2	68.7	72.4	83.2	77.3	67.3	55.1	96.3	96.7	87.2	79.7	11.7	21.0	33.4	27.8	68.1	1.3	6.9	6.8
MoGe-3-G (Kong et al., 2026)	92.0	72.8	58.4	41.2	56.1	43.8	67.1	67.6	90.3	80.3	65.7	50.9	94.2	56.0	89.3	88.1	13.2	24.6	22.1	30.1	75.0	13.1	6.2	6.5
MoGe-2-L (Wang et al., 2025c)	89.0	77.8	57.1	46.2	57.7	43.9	75.3	66.9	92.1	**90.9**	67.2	51.3	91.5	42.7	87.0	85.6	22.2	36.1	**27.2**	17.7	71.0	8.1	5.6	6.1
DA3-NGL (Lin et al., 2026)	**96.1**	86.4	**73.9**	48.8	71.6	72.0	73.9	78.4	83.7	80.2	68.4	59.4	79.3	50.6	83.6	49.6	21.9	19.0	17.2	8.5	59.1	23.4	5.6	5.6
DAGE (Ngo et al., 2026)	89.9	84.2	69.6	69.2	**76.4**	**74.8**	76.1	74.8	90.4	84.0	63.5	60.8	95.8	89.1	74.8	83.2	22.2	15.7	91.3	93.1	**82.3**	47.8	4.7	4.2
UniDepthV2-L (Piccinelli et al., 2025b)	94.5	**94.6**	55.7	55.3	73.6	66.8	48.0	41.3	87.5	73.9	62.6	54.3	97.4	**97.6**	92.2	91.9	36.4	37.0	25.4	22.8	80.1	41.1	4.5	4.0
UniK3D-L (Piccinelli et al., 2025a)	95.3	94.4	59.3	51.6	73.5	73.8	58.1	61.1	92.0	84.9	72.2	**67.6**	98.0	97.0	**92.4**	**92.1**	31.7	31.4	19.2	19.8	76.4	61.1	3.6	3.0
Depth Pro 2-H⁺	94.4	81.5	73.8	**84.1**	65.9	58.1	**89.4**	**87.9**	**97.8**	86.1	**74.2**	65.2	**98.2**	91.9	90.9	64.7	**60.2**	**38.5**	20.5	**38.9**	80.5	**64.8**	**2.4**	**2.9**

Table 2: **Zero-shot metric depth accuracy, video.** All methods receive clips up to 50 frames. Single-image methods process frames independently. We report δ_1 and average rank, as in Table 1.

Method	Real, outdoor		Synthetic						Avg. rank↓	
	KITTI		Sintel		Spring		DyDToF			
	GT	pred	GT	pred	GT	pred	GT	pred	GT	pred
MapAnything (Keetha et al., 2026)	94.1	94.7	21.2	18.6	97.0	97.0	8.4	7.5	10.0	8.7
FoundationGeo (Liu et al., 2026b)	96.3	96.7	11.7	21.0	33.4	27.8	68.1	1.3	9.0	8.0
DA3-NGL (Lin et al., 2026)	95.7	25.8	30.3	20.3	16.9	11.1	77.4	32.1	5.0	9.0
VeloDepth (Piccinelli et al., 2026)	84.2	74.0	27.5	30.1	22.0	25.0	69.3	57.8	8.2	5.0
DepthPro (Bochkovskii et al., 2025)	84.8	82.7	36.5	30.1	12.5	16.7	67.2	27.4	7.8	6.8
MetricAnything‡ (Ma et al., 2026)	5.7	–	25.6	–	22.5	–	73.3	–	7.8	–
MoGe-3-G (Kong et al., 2026)	94.2	56.0	13.2	24.6	22.1	30.1	75.0	13.1	7.2	6.5
MoGe-2-L (Wang et al., 2025c)	91.5	42.7	22.2	36.1	27.2	17.7	71.0	8.1	7.0	7.0
DAGE (Ngo et al., 2026)	96.7	84.6	28.5	22.3	96.8	97.4	82.2	65.0	4.0	5.3
UniK3D-L (Piccinelli et al., 2025a)	98.0	97.0	31.7	31.4	19.2	19.8	76.4	61.1	4.5	3.5
UniDepthV2-L (Piccinelli et al., 2025b)	97.4	**97.6**	36.4	37.0	25.4	22.8	80.1	41.1	3.0	3.0
Depth Pro 2-H⁺	**98.1**	96.8	**51.2**	**39.4**	**51.8**	**35.5**	**86.3**	**80.6**	**1.0**	**1.5**

Table 3: **Zero-shot boundary accuracy.** We report the boundary $F1$ scores (higher is better) from Bochkovskii et al. (2025) along with average ranks across datasets. Table 3(a) methods receive a single frame (DepthCrafter, VeloDepth, and VDA are excluded here as they do not support single-image input) and Table 3(b) methods receive 50-frame clips. Non-zero-shot results are gray.

(a) Single image

Method	Real, indoor		Synthetic			Avg. rank↓
	iBims	M.bury	Sintel	Spring	DyDToF	
VGGT-Ω-1B (Wang et al., 2026a)	6.4	0.2	15.4	5.5	11.0	–
π^3 (Wang et al., 2026b)	6.4	0.2	10.8	4.0	7.1	19.4
MapAnything (Keetha et al., 2026)	9.2	0.4	9.9	5.0	7.9	19.2
VGGT-1B (Wang et al., 2025a)	7.1	0.5	12.3	4.6	8.7	17.8
DA3-NGL (Lin et al., 2026)	8.0	3.6	7.9	3.4	7.6	17.8
ChronoDepth (Shao et al., 2025)	9.7	3.6	16.2	3.9	14.3	15.6
FoundationGeo (Liu et al., 2026b)	13.6	0.8	23.9	8.5	15.7	15.2
DVD (Zhang et al., 2026)	15.9	1.3	19.5	7.0	18.3	14.0
Lapis *(log)* (Liu et al., 2026a)	13.2	8.9	22.7	8.5	11.1	12.4
DAGE (Ngo et al., 2026)	19.0	1.6	31.9	12.6	22.2	11.5
FlashDepth-L (Chou et al., 2025)	13.7	10.8	22.2	13.5	18.5	11.5
UniDepthV2-L (Piccinelli et al., 2025b)	15.9	2.9	34.3	7.0	23.9	11.2
UniK3D-L (Piccinelli et al., 2025a)	17.6	3.0	33.5	7.9	24.6	9.6
Lotus-2 (He et al., 2026)	17.6	5.7	28.5	10.7	24.4	9.0
MoGe-2-L (Wang et al., 2025c)	20.6	2.9	32.2	8.4	23.0	9.0
Marigold V2 *(log)* (Pavlovic et al., 2026)	15.0	16.2	27.6	13.9	22.6	8.4
Pixel-Perfect Depth *(log)* (Xu et al., 2025)	20.4	6.1	34.9	10.1	27.7	5.8
DepthPro (Bochkovskii et al., 2025)	19.5	7.9	36.3	12.7	24.8	5.4
MoGe-3-G (Kong et al., 2026)	**22.2**	6.2	37.1	9.5	27.5	5.0
MetricAnything (Ma et al., 2026)	19.2	9.6	34.4	14.0	29.4	4.6
InfiniDepth (Yu et al., 2026)	20.4	17.6	**44.8**	11.8	30.5	3.0
Depth Pro 2-H⁺	20.5	**22.0**	40.2	**16.3**	**31.6**	**1.6**

(b) Video

Method	Synthetic			Avg. rank↓
	Sintel	Spring	DyDToF	
VGGT-Ω-1B (Wang et al., 2026a)	15.7	5.6	10.2	–
DA3-NGL (Lin et al., 2026)	9.0	3.8	8.0	23.5
MapAnything (Keetha et al., 2026)	10.1	5.1	8.4	22.0
π^3 (Wang et al., 2026b)	11.6	4.2	6.8	21.0
VGGT-1B (Wang et al., 2025a)	12.2	4.7	8.1	19.7
VeloDepth (Piccinelli et al., 2026)	15.0	4.0	9.9	19.3
ChronoDepth (Shao et al., 2025)	15.6	3.8	12.6	18.7
FlashDepth-L (Chou et al., 2025)	22.2	13.5	18.5	15.5
FoundationGeo (Liu et al., 2026b)	23.9	8.5	15.7	15.5
Lapis *(log)* (Liu et al., 2026a)	22.7	8.5	11.1	15.3
DepthCrafter (Hu et al., 2025)	24.7	7.8	15.1	14.7
DVD (Zhang et al., 2026)	19.9	8.9	18.2	14.3
VDA-L (Chen et al., 2025)	23.9	9.1	18.6	12.0
MoGe-2-L (Wang et al., 2025c)	32.2	8.4	23.0	10.7
DAGE (Ngo et al., 2026)	32.9	12.7	21.5	10.5
UniDepthV2-L (Piccinelli et al., 2025b)	34.3	7.0	23.9	10.3
UniK3D-L (Piccinelli et al., 2025a)	33.5	7.9	24.6	9.3
Marigold V2 *(log)* (Pavlovic et al., 2026)	27.6	13.9	22.6	8.7
Lotus-2 (He et al., 2026)	28.5	10.7	24.4	8.3
MoGe-3-G (Kong et al., 2026)	37.1	9.5	27.5	5.3
Pixel-Perfect Depth *(log)* (Xu et al., 2025)	34.9	10.1	27.7	5.3
DepthPro (Bochkovskii et al., 2025)	36.3	12.7	24.8	4.7
MetricAnything (Ma et al., 2026)	34.4	14.0	29.4	3.7
InfiniDepth (Yu et al., 2026)	**44.8**	11.8	30.5	2.7
Depth Pro 2-H⁺	43.4	**16.8**	**34.4**	**1.3**

For a fair comparison of single-image and video methods, our evaluation is split into a single-image setting (Table 1) and a video setting (Table 2). In the single-image setting (Table 1), all methods are run and evaluated per frame. In the video setting (Table 2), we use the datasets that provide video frames (KITTI, Sintel, Spring, and DyDToF) and every method receives the same clips of

50 frames. Single-image methods process each frame independently. All of our subsequent zero-shot evaluations follow this split as well. ETH3D is a multi-view stereo benchmark with wide-baseline views of a static scene rather than video, so it appears in the single-image setting only. We evaluate all models under two conditions: first with ground-truth camera intrinsics and second with predicted camera intrinsics, if the model is capable of estimating focal length. In both the single-image (Table 1) and the video setting (Table 2), Depth Pro 2 achieves the lowest average rank both with and without ground-truth camera intrinsics.

Zero-shot boundaries. Following Bochkovskii et al. (2025), we evaluate zero-shot boundary sharpness using the $F1$ score in Table 3. We report scores on datasets from Table 1 that have high-quality *dense* depth annotations. Depth Pro 2 has the lowest average rank across datasets.

Temporal consistency. We evaluate our model's ability to predict temporally consistent depth. Table 4 reports jitter on the video datasets for which dense high-quality optical flow and occlusion masks are available. For the definition of the metric, further results using the Temporal Alignment Error, and a discussion of its limitations, we refer to Sec. C.1 and Tab. 11 in the appendices.

The results in Tables 3 and 4 confirm our earlier findings from Sec 3.1. Multi-frame approaches perform well in temporal consistency but poorly on boundary sharpness, while the opposite holds for single-frame approaches. Depth Pro 2 overcomes this trade-off and sets a new state of the art for both boundary sharpness and temporal consistency.

Applications. Appendix D reports depth-guided bokeh rendering under a fixed pipeline, with the renderer held constant and only the depth it receives varied, in Table 17 and Figure 7. It also reports single-image novel view synthesis (Khan et al., 2023) under a fixed renderer in Figure 9.

Table 4: **Temporal consistency along point tracks.** We apply one affine alignment per video and pool jitter over unoccluded ground-truth point tracks within each dataset, see Appendix. Scaled by 10^2 for readability. Lower is better.

Method	Sintel	Spring	Avg. rank↓
VGGT-Ω-1B (Wang et al., 2026a)	1.47	0.97	–
Pixel-Perfect Depth *(log)* (Xu et al., 2025)	10.01	9.88	23.5
DepthPro (Bochkovskii et al., 2025)	11.28	9.06	23.0
InfiniDepth (Yu et al., 2026)	7.68	9.70	22.5
Lapis *(log)* (Liu et al., 2026a)	6.78	4.15	20.0
UniK3D-L (Piccinelli et al., 2025a)	5.84	4.92	18.5
Marigold V2 *(log)* (Pavlovic et al., 2026)	7.10	3.03	18.0
MoGe-2-L (Wang et al., 2025c)	6.28	3.75	18.0
Lotus-2 (He et al., 2026)	6.43	3.06	17.0
UniDepthV2-L (Piccinelli et al., 2025b)	5.32	3.43	16.0
FoundationGeo (Liu et al., 2026b)	6.76	2.84	15.5
ChronoDepth (Shao et al., 2025)	4.05	3.07	14.0
FlashDepth-L (Chou et al., 2025)	5.35	2.18	13.0
MetricAnything (Ma et al., 2026)	5.26	2.88	13.0
MoGe-3-G (Kong et al., 2026)	5.11	2.90	13.0
MapAnything (Keetha et al., 2026)	3.46	1.12	9.0
DA3-L (Lin et al., 2026)	2.56	1.34	7.5
DVD (Zhang et al., 2026)	2.55	1.40	7.2
DAGE (Ngo et al., 2026)	2.62	1.00	6.0
VGGT-1B (Wang et al., 2025a)	2.59	1.02	6.0
DepthCrafter (Hu et al., 2025)	2.28	1.08	5.0
VeloDepth (Piccinelli et al., 2026)	3.42	**0.73**	5.0
π^3 (Wang et al., 2026b)	2.12	1.04	4.0
VDA-L (Chen et al., 2025)	2.55	0.87	3.8
Depth Pro 2-H$^+$	**1.38**	0.85	**1.5**

Runtime analysis. On a single B200 GPU and at a 32 frames window context, the Depth Pro 2-H$^+$ model runs in 360 ms per frame for 1080p video, while running in about 10 ms for VGA video. Appendix A.1 provides details and reports runtime and peak memory for the entire Depth Pro 2 family of models.

5 Conclusion

We present a family of models on the frontier of estimating sharp, consistent metric depth. By analyzing representative approaches to monocular depth estimation for their ability to produce temporally coherent, sharp depth maps from video, we observe a divide between methods operating on single images and those operating on multiple frames. We find that processing multiple frames jointly enables temporal consistency, and processing them at full resolution allows for recovering sharp depth boundaries. Building on these insights, we present a simple architecture that processes multiple frames jointly at native resolution, and a boundary-sensitive training recipe, which together yield a method that predicts metric depth from single images and videos with unprecedented accuracy, sharpness, and temporal consistency.

AI USE STATEMENT

We used generative AI tools as (a) a coding assistant during implementation and experiment tooling, and (b) a writing assistant for helping build tables and figures and revising the manuscript. All AI-assisted work was reviewed and verified by the (human) authors. The authors take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of AI.

REFERENCES

Adel Ahmadyan, Liangkai Zhang, Artsiom Ablavatski, Jianing Wei, and Matthias Grundmann. Objectron: A large scale dataset of object-centric videos in the wild with pose annotations. In *CVPR*, 2021.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, Wolfgang Macherey, Zhifeng Chen, and Yonghui Wu. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv:1907.05019*, 2019.

Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. ViViT: A video vision transformer. In *ICCV*, 2021.

Eduardo Arnold, Jamie Wynn, Sara Vicente, Guillermo Garcia-Hernando, Áron Monszpart, Victor Prisacariu, Daniyar Turmukhambetov, and Eric Brachmann. Map-free visual relocalization: Metric pose relative to a single image. In *ECCV*, 2022.

Prithviraj Banerjee, Sindi Shkodrani, Pierre Moulon, Shreyas Hampali, Shangchen Han, Fan Zhang, Linguang Zhang, Jade Fountain, Edward Miller, Selen Basol, Richard A. Newcombe, Robert Wang, Jakob Julian Engel, and Tomas Hodan. HOT3D: hand and object tracking in 3d from egocentric multi-view videos. In *CVPR*, 2025.

Gilad Baruch, Zhuoyuan Chen, Afshin Dehghan, Tal Dimry, Yuri Feigin, Peter Fu, Thomas Gebauer, Brandon Joffe, Daniel Kurz, Arik Schwartz, and Elad Shulman. ARKitScenes - a diverse real-world dataset for 3D indoor scene understanding using mobile RGB-D data. In *NeurIPS Datasets and Benchmarks*, 2021.

Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, 2021.

Michael J. Black, Priyanka Patel, Joachim Tesch, and Jinlong Yang. BEDLAM: A synthetic dataset of bodies exhibiting detailed lifelike animated motion. In *CVPR*, 2023.

Aleksei Bochkovskii, Amaël Delaunoy, Hugo Germain, Marcel Santos, Yichao Zhou, Stephan R. Richter, and Vladlen Koltun. Depth Pro: Sharp monocular metric depth in less than a second. In *ICLR*, 2025.

Daniel J. Butler, Jonas Wulff, Garrett B. Stanley, and Michael J. Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012.

Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual KITTI 2. *arXiv:2001.10773*, 2020.

Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020.

Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the Kinetics dataset. In *CVPR*, 2017.

Shuo Chen, Tan Yu, and Ping Li. MVT: Multi-view vision transformer for 3D object recognition. In *BMVC*, 2021.

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 *CVPR*, 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 *ICCV*, 2025.

Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *CVPR*, 2017.

Afshin Dehghan, Gilad Baruch, Zhuoyuan Chen, Yuri Feigin, Peter Fu, Thomas Gebauer, Daniel Kurz, Tal Dimry, Brandon Joffe, Arik Schwartz, and Elad Shulman. Arkitscenes: A diverse real-world dataset for 3d indoor scene understanding using mobile RGB-D data. In *NeurIPS Datasets and Benchmarks*, 2021.

Keyan Ding, Kede Ma, Shiqi Wang, and Eero P. Simoncelli. Image quality assessment: Unifying structure and texture similarity. *TPAMI*, 2022.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015.

David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NeurIPS*, 2014.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *CVPR*, 2012.

Jose L. Gómez, Manuel Silva, Antonio Seoane, Agnès Borrás, Mario Noriega, Germán Ros, Jose A. Iglesias-Guitian, and Antonio M. López. All for one, and one for all: UrbanSyn dataset, the third musketeer of synthetic driving scenes. *Neurocomputing*, 2025.

Jing He, Haodong Li, Mingzhi Sheng, and Ying-Cong Chen. Lotus-2: Advancing geometric dense prediction with powerful image generative model. *arXiv:2512.01030*, 2026.

Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox. RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments. *IJRR*, 2012.

Byeongho Heo, Song Park, Dongyoon Han, and Sangdoo Yun. Rotary position embedding for vision transformer. In *ECCV*, 2024.

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 *CVPR*, 2025.

Hanwen Jiang, Zexiang Xu, Desai Xie, Ziwen Chen, Haian Jin, Fujun Luan, Zhixin Shu, Kai Zhang, Sai Bi, Xin Sun, Jiuxiang Gu, Qixing Huang, Georgios Pavlakos, and Hao Tan. MegaSynth: Scaling up 3D scene reconstruction with synthesized data. In *CVPR*, 2025.

HyunJun Jung, Patrick Ruhkamp, Guangyao Zhai, Nikolas Brasch, Yitong Li, Yannick Verdie, Jifei Song, Yiren Zhou, Anil Armagan, Slobodan Ilic, Ales Leonardis, and Benjamin Busam. Is my depth ground-truth good enough? HAMMER–highly accurate multi-modal dataset for DEnse 3D scene regression. *arXiv:2205.04565*, 2022.

Nikita Karaev, Ignacio Rocco, Benjamin Graham, Natalia Neverova, Andrea Vedaldi, and Christian Rupprecht. DynamicStereo: Consistent dynamic depth from stereo videos. In *CVPR*, 2023.

Bingxin Ke, Anton Obukhov, Shengyu Huang, Nando Metzger, Rodrigo Caye Daudt, and Konrad Schindler. Repurposing diffusion-based image generators for monocular depth estimation. In *CVPR*, 2024.

Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, Jonathon Luiten, Manuel Lopez-Antequera, Samuel Rota Bulò, Christian Richardt, Deva Ramanan, Sebastian Scherer, and Peter Kontschieder. MapAnything: Universal feed-forward metric 3D reconstruction. In *3DV*, 2026.

Levon Khachatryan, Andranik Movsisyan, Vahram Tadevosyan, Roberto Henschel, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Text2Video-Zero: Text-to-image diffusion models are zero-shot video generators. In *ICCV*, 2023.

Numair Khan, Lei Xiao, and Douglas Lanman. Tiled multiplane images for practical 3D photography. In *ICCV*, 2023.

Tobias Koch, Lukas Liebel, Friedrich Fraundorfer, and Marco Körner. Evaluation of CNN-based single-image depth estimation methods. In *ECCV Workshops*, 2018.

Lingyu Kong, Ruicheng Li, Ruicheng Wang, Sicheng Xu, Chengtang Yao, Jianfeng Xiang, and Jiaolong Yang. MoGe-3: Fine-detail monocular geometry estimation with self-guided sparse volumetric refinement. *arXiv:2607.17967*, 2026.

Hoang-An Le, Thomas Mensink, Partha Das, Sezer Karaoglu, and Theo Gevers. EDEN: Multimodal synthetic dataset of enclosed GarDEN scenes. In *WACV*, 2021.

Han-Hung Lee, Yiming Zhang, and Angel Chang. Duoduo CLIP: Efficient 3D understanding with multi-view images. In *ICLR*, 2025.

Jizhizi Li, Jing Zhang, Stephen J. Maybank, and Dacheng Tao. Bridging composite and real: Towards end-to-end deep image matting. *IJCV*, 2022.

Yixuan Li, Lihan Jiang, Linning Xu, Yuanbo Xiangli, Zhenzhi Wang, Dahua Lin, and Bo Dai. MatrixCity: A large-scale city dataset for city-scale neural rendering and beyond. In *ICCV*, 2023.

Zhengqi Li and Noah Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *CVPR*, 2018.

Zhenyu Li, Shariq Farooq Bhat, and Peter Wonka. PatchFusion: An end-to-end tile-based framework for high-resolution monocular metric depth estimation. In *CVPR*, 2024a.

Zhenyu Li, Shariq Farooq Bhat, and Peter Wonka. Patchrefiner: Leveraging synthetic data for real-domain high-resolution monocular metric depth estimation. In *ECCV*, 2024b.

Haotong Lin, Sili Chen, Jun Hao Liew, Donny Y. Chen, Zhenyu Li, Guang Shi, Jiashi Feng, and Bingyi Kang. Depth Anything 3: Recovering the visual space from any views. In *ICLR*, 2026.

Bingde Liu, Wu Ran, Jinglei Zhang, Huanhuan Yuan, and Chao Ma. Efficient and high-quality depth estimation via pixel-space diffusion with linear attention. In *ECCV*, 2026a.

Muxin Liu, Xiaoyang Lyu, Tianhe Ren, Peng Dai, Xiaoshan Wu, Zhiyue Zhang, Jiaqi Zhang, Jiehong Lin, Shaoshuai Shi, and Xiaojuan Qi. FoundationGeo: Learning spatial pixel-wise fields for monocular metric geometry. In *ECCV*, 2026b.

Xuan Luo, Jia-Bin Huang, Richard Szeliski, Kevin Matzen, and Johannes Kopf. Consistent video depth estimation. *ACM TOG*, 2020.

Baorui Ma, Jiahui Yang, Donglin Di, Xuancheng Zhang, Jianxun Cui, Hao Li, Yan Xie, and Wei Chen. MetricAnything: Scaling metric depth pretraining with noisy heterogeneous sources. In *ECCV*, 2026.

Lingni Ma, Yuting Ye, Fangzhou Hong, Vladimir Guzov, Yifeng Jiang, Rowan Postyeni, Luis Pesqueira, Alexander Gamino, Vijay Baiyya, Hyo Jin Kim, Kevin Bailey, David Soriano Fosas, C. Karen Liu, Ziwei Liu, Jakob J. Engel, Renzo De Nardi, and Richard A. Newcombe. Nymeria: A massive collection of multimodal egocentric daily motion in the wild. In *ECCV*, 2024.

Parv Maheshwari, Jay Karhade, Yogesh Chawla, Isaiah Adu, Florian Heisen, Andrew Porco, Andrew Jong, Yifei Liu, Santosh Pitla, Sebastian A. Scherer, and Wenshan Wang. AnyThermal: Towards learning universal representations for thermal perception. *arXiv:2602.06203*, 2026.

Nikolaus Mayer, Eddy Ilg, Philip Häusser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, 2016.

Lukas Mehl, Jenny Schmalfuss, Azin Jahedi, Yaroslava Nalivayko, and Andrés Bruhn. Spring: A high-resolution high-detail dataset and benchmark for scene flow, optical flow and stereo. In *CVPR*, 2023.

S. Mahdi H. Miangoleh, Sebastian Dille, Long Mai, Sylvain Paris, and Yağız Aksoy. Boosting monocular depth estimation models to high-resolution via content-adaptive multi-resolution merging. In *CVPR*, 2021.

Tuan Duc Ngo, Jiahui Huang, Seoung Wug Oh, Kevin Blackburn-Matzen, Evangelos Kalogerakis, Chuang Gan, and Joon-Young Lee. DAGE: Dual-stream architecture for efficient and fine-grained geometry estimation. In *CVPR*, 2026.

Emanuele Palazzolo, Jens Behley, Philipp Lottes, Philippe Giguère, and Cyrill Stachniss. ReFusion: 3D reconstruction in dynamic environments for RGB-D cameras exploiting residuals. In *IROS*, 2019.

Xiaqing Pan, Nicholas Charron, Yongqian Yang, Scott Peters, Thomas Whelan, Chen Kong, Omkar M. Parkhi, Richard A. Newcombe, and Carl Yuheng Ren. Aria digital twin: A new benchmark dataset for egocentric 3d machine perception. In *ICCV*, 2023.

Manthan Patel, Fan Yang, Yuheng Qiu, Cesar Cadena, Sebastian Scherer, Marco Hutter, and Wenshan Wang. TartanGround: A large-scale dataset for ground robot perception and navigation. In *IROS*, 2025.

Igor Pavlovic, Thiemo Wandel, Anton Obukhov, Luca Bartolomei, Andrey Davydov, Fabio Tosi, Matteo Poggi, Sabine Süsstrunk, and Dengxin Dai. Marigold V2: Revisiting diffusion transformers for monocular depth estimation. In *SIGGRAPH Asia*, 2026.

Juewen Peng, Zhiguo Cao, Xianrui Luo, Hao Lu, Ke Xian, and Jianming Zhang. BokehMe: When neural rendering meets classical rendering. In *CVPR*, 2022.

Pexels. Pexels: Free stock videos. Online, 2026. URL https://www.pexels.com/videos/.

Luigi Piccinelli, Yung-Hsu Yang, Christos Sakaridis, Mattia Segu, Siyuan Li, Luc Van Gool, and Fisher Yu. UniDepth: Universal monocular metric depth estimation. In *CVPR*, 2024.

Luigi Piccinelli, Christos Sakaridis, Mattia Segu, Yung-Hsu Yang, Siyuan Li, Wim Abbeloos, and Luc Van Gool. UniK3D: Universal camera monocular 3D estimation. In *CVPR*, 2025a.

Luigi Piccinelli, Christos Sakaridis, Yung-Hsu Yang, Mattia Segu, Siyuan Li, Wim Abbeloos, and Luc Van Gool. UniDepthV2: Universal monocular metric depth estimation made simpler. *TPAMI*, 2025b.

Luigi Piccinelli, Thiemo Wandel, Christos Sakaridis, Wim Abbeloos, and Luc Van Gool. Video depth propagation. In *3DV*, 2026.

Xuebin Qin, Hang Dai, Xiaobin Hu, Deng-Ping Fan, Ling Shao, and Luc Van Gool. Highly accurate dichotomous image segmentation. In *ECCV*, 2022.

Pierluigi Zama Ramirez, Fabio Tosi, Matteo Poggi, Samuele Salti, Stefano Mattoccia, and Luigi Di Stefano. Open challenges in deep stereo: the Booster dataset. In *CVPR*, 2022.

René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, 2021.

René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *TPAMI*, 2022.

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 *ICCV*, 2021.

Ashutosh Saxena, Min Sun, and Andrew Y. Ng. Make3D: Learning 3D scene structure from a single still image. *TPAMI*, 2009.

Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *GCPR*, 2014.

Thomas Schöps, Johannes L. Schönberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *CVPR*, 2017.

Tim Seizinger, Florin-Alexandru Vasluianu, Marcos Conde, Zongwei Wu, and Radu Timofte. Bokehlicious: Photorealistic bokeh rendering with controllable apertures. In *ICCV*, 2025.

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 *CVPR*, 2025.

Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. MVDream: Multi-view diffusion for 3D generation. In *ICLR*, 2024.

Jamie Shotton, Andrew W. Fitzgibbon, Mat Cook, Toby Sharp, Mark Finocchio, Richard Moore, Alex Kipman, and Andrew Blake. Real-time human pose recognition in parts from single depth images. In *CVPR*, 2011.

Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. DINOv3. *TMLR*, 2026.

Michal Starý, Julien Gaubil, Ayush Tewari, and Vincent Sitzmann. Understanding multi-view transformers. *arXiv:2510.24907*, 2025.

Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Sheng Zhao, Shuyang Cheng, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *CVPR*, 2020.

Zhanghao Sun, Wei Ye, Jinhui Xiong, Gyeongmin Choe, Jialiang Wang, Shuochen Su, and Rakesh Ranjan. Consistent direct time-of-flight video depth super-resolution. In *CVPR*, 2023.

Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. YFCC100M: The new data in multimedia research. *Communications of the ACM*, 59(2), 2016.

Javier Tirado-Garín and Javier Civera. AnyCalib: On-manifold learning for model-agnostic single-view camera calibration. In *ICCV*, 2025.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

Alexander Veicht, Paul-Edouard Sarlin, Philipp Lindenberger, and Marc Pollefeys. GeoCalib: Learning single-image calibration with geometric optimization. In *ECCV*, 2024.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. VGGT: Visual geometry grounded transformer. In *CVPR*, 2025a.

Jianyuan Wang, Minghao Chen, Shangzhan Zhang, Nikita Karaev, Johannes Schönberger, Patrick Labatut, Piotr Bojanowski, David Novotny, Andrea Vedaldi, and Christian Rupprecht. VGGT-Ω. In *CVPR*, 2026a.

Qiang Wang, Shizhen Zheng, Qingsong Yan, Fei Deng, Kaiyong Zhao, and Xiaowen Chu. IRS: A large naturalistic indoor robotics stereo dataset to train deep models for disparity and surface normal estimation. In *ICME*, 2021.

Qianqian Wang, Yifei Zhang, Aleksander Holynski, Alexei A. Efros, and Angjoo Kanazawa. Continuous 3D perception model with persistent state. In *CVPR*, 2025b.

Ruicheng Wang, Sicheng Xu, Yue Dong, Yu Deng, Jianfeng Xiang, Zelong Lv, Guangzhong Sun, Xin Tong, and Jiaolong Yang. MoGe-2: Accurate monocular geometry with metric scale and sharp details. In *NeurIPS*, 2025c.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3D vision made easy. In *CVPR*, 2024.

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 *IROS*, 2020.

Yifan Wang, Jianjun Zhou, Haoyi Zhu, Wenzheng Chang, Yang Zhou, Zizun Li, Junyi Chen, Jiangmiao Pang, Chunhua Shen, and Tong He. π^3: Permutation-equivariant visual geometry learning. In *ICLR*, 2026b.

Yiran Wang, Min Shi, Jiaqi Li, Zihao Huang, Zhiguo Cao, Jianming Zhang, Ke Xian, and Guosheng Lin. Neural video depth stabilizer. In *ICCV*, 2023.

Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *NeurIPS Datasets and Benchmarks*, 2021.

Gangwei Xu, Haotong Lin, Hongcheng Luo, Xianqi Wang, Jingfeng Yao, Lianghui Zhu, Yuechuan Pu, Cheng Chi, Haiyang Sun, Bing Wang, Guang Chen, Hangjun Ye, Sida Peng, and Xin Yang. Pixel-perfect depth with semantics-prompted diffusion transformers. In *NeurIPS*, 2025.

Charig Yang, Samiul Alam, Shakhrul Iman Siam, Michael J. Proulx, Lambert Mathias, Kiran K. Somasundaram, Luis Pesqueira, James Fort, Sheroze Sheriffdeen, Omkar M. Parkhi, Carl Yuheng Ren, Mi Zhang, Yuning Chai, Richard A. Newcombe, and Hyo Jin Kim. Reading recognition in the wild. In *NeurIPS*, 2025a.

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 *ICLR*, 2025b.

Lihe Yang, Bingyi Kang, Zilong Huang, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything: Unleashing the power of large-scale unlabeled data. In *CVPR*, 2024a.

Lihe Yang, Bingyi Kang, Zilong Huang, Zhen Zhao, Xiaogang Xu, Jiashi Feng, and Hengshuang Zhao. Depth anything V2. In *NeurIPS*, 2024b.

Wei Yin, Chi Zhang, Hao Chen, Zhipeng Cai, Gang Yu, Kaixuan Wang, Xiaozhi Chen, and Chunhua Shen. Metric3D: Towards zero-shot metric 3D prediction from a single image. In *ICCV*, 2023.

Hao Yu, Haotong Lin, Jiawei Wang, Jiaxin Li, Yida Wang, Xueyang Zhang, Yue Wang, Xiaowei Zhou, Ruizhen Hu, and Sida Peng. InfiniDepth: Arbitrary-resolution and fine-grained depth estimation with neural implicit fields. In *CVPR*, 2026.

Hongfei Zhang, Harold Haodong Chen, Chenfei Liao, Jing He, Zixin Zhang, Haodong Li, Yihao Liang, Kanghao Chen, Bin Ren, Xu Zheng, Shuai Yang, Kun Zhou, Yinchuan Li, Nicu Sebe, and Ying-Cong Chen. DVD: Deterministic video depth estimation with generative priors. In *ICML Workshops*, 2026.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018.

Zhengyou Zhang. Microsoft Kinect sensor and its effect. *IEEE MultiMedia*, 2012.

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.

Xuan Zhu, Jijun Xiang, Xianqi Wang, Longliang Liu, Yu Wang, Hong Zhang, Fei Guo, and Xin Yang. SVDC: Consistent direct time-of-flight video depth completion with frequency selective fusion. In *CVPR*, 2025.

Jannik Zürn, Paul Gladkov, Sofía Dudas, Fergal Cotter, Sofi Toteva, Jamie Shotton, Vasiliki Simaiaki, and Nikhil Mohan. Wayvescenes101: A dataset and benchmark for novel view synthesis in autonomous driving. *arXiv:2407.08280*, 2024.

SUPPLEMENT

The supplement is organized as follows. Appendix A provides additional model and training details, Appendix B presents additional results, Appendix C specifies the evaluation protocol used throughout the paper, and Appendix D evaluates predicted depth in two downstream applications: bokeh rendering and novel-view synthesis.

A MODEL AND TRAINING DETAILS

A.1 ARCHITECTURE VARIANTS

Our model family comprises four variants, each built on a different pretrained DINOv3 backbone. Table 5 details the corresponding backbone and DPT decoder configurations.

Table 5: **Architecture specifications of the Depth Pro 2 model family.** Cross-frame blocks lists the backbone blocks that use cross-frame attention. Reassembled blocks lists the blocks whose features are passed to the DPT decoder. Block indices are zero-based. All variants use a patch size of 16×16.

		Backbone				DPT decoder			
Model	Total params	Params	L	D	Cross-frame blocks	Params	Reassembled blocks	D_s $(s = 4, 8, 16, 32)$	$\hat{D}$
Depth Pro 2-S	24.4M	21.6M	12	384	$5, 7, 9, 11$	2.8M	$2, 5, 8, 11$	$48, 96, 192, 384$	64
Depth Pro 2-B	96.8M	85.6M	12	768	$5, 7, 9, 11$	11.2M	$2, 5, 8, 11$	$96, 192, 384, 768$	128
Depth Pro 2-L	335.1M	303.1M	24	1024	$9, 11, \ldots, 23$	32.0M	$5, 11, 17, 23$	$256, 512, 1024, 1024$	256
Depth Pro 2-H$^+$	873.3M	840.5M	32	1280	$11, 13, \ldots, 31$	32.8M	$7, 15, 23, 31$	$256, 512, 1024, 1024$	256

Table 6 shows the scaling effect of model capacity within the Depth Pro 2 family. Note we exclude focal length estimator in the parameter sets and use GT intrinsics for a grounded comparison.

Table 6: **Scaling within the Depth Pro 2 family.** Each column averages one metric over the datasets of Table 2 for δ_1, Table 3(b) for $F1$, and Table 4 for jitter. All are scored on 50-frame clips, with ground-truth camera intrinsics for metric depth. $F1$ is scaled by $100\times$ for readability.

Model	Params	Metric $\delta_1 \uparrow$	Sharpness $F1 \uparrow$	Consistency jitter$\downarrow$
Depth Pro 2-S	24M	55.6	18.6	1.78
Depth Pro 2-B	97M	63.8	25.3	1.46
Depth Pro 2-L	335M	61.5	30.7	1.20
Depth Pro 2-H$^+$	873M	**71.8**	**31.6**	**1.12**

Runtime and memory. Table 7 reports network performance within the Depth Pro 2 family. Because our model runs at the native resolution of its input, we detail how speed scales across model sizes and video resolutions. We exclude warmup and compilation time. Every variant runs over a 32 frame window. At 4K, Depth Pro 2-H$^+$ requires 80.3 GiB and runs in 5.5 sec per frame, while Depth Pro 2-S requires 42.7 GiB and runs in about 600 ms.

A.2 TRAINING IMPLEMENTATION DETAILS

Training optimization. We train all our models for 100K steps on 128 B200 GPUs. We use AdamW at a peak learning rate of 5×10^{-5}, cosine annealed after a 5% warmup.

Section 3.3 describes our native-resolution training strategy. Here, we provide additional implementation details on token-budget batching and the composition of the training data.

Token-budget batching. Each GPU has a maximum budget of 73,728 patch tokens during training. Half of the steps use the full native frame. Otherwise, we draw an aspect ratio relative to the native aspect and a short-side limit, then crop to 60–100% of the largest feasible window. Where native resolution permits, we choose spatial stride two with probability 0.25, taking every second pixel in

Table 7: **Runtime and peak memory within the Depth Pro 2 family.** We measure on one B200 GPU at bf16 precision with `torch.compile`. We report sustained throughput (FPS) and peak memory (GiB) per resolution on 32-frame clips for all models. † we split features into chunks at decoding stage to avoid OOM at 4K resolution. Preprocessing time is excluded.

	Throughput (FPS) ↑				Peak memory (GiB) ↓			
Model	VGA	720p	1080p	4K	VGA	720p	1080p	4K
Depth Pro 2-S	843.0	117.0	25.5	1.7	1.6	4.9	10.7	42.7
Depth Pro 2-B	439.0	58.8	12.7	0.8	2.7	8.0	17.5	69.5
Depth Pro 2-L	168.0	22.1	4.8	0.3†	5.3	14.8	32.1	79.3†
Depth Pro 2-H⁺	96.7	12.8	2.8	0.2†	6.3	15.8	33.1	80.3†

both the image and depth. We do not interpolate ground truth to keep fidelity. Each clip uses as many frames as fit in the token budget, up to a separate cap of 32 frames. Remaining capacity is filled with more clips from the same source. The short side of the input video is 608/1024/2048 pixels at 10th/50th/90th percentiles. The sampled aspect ratios are 1.00/1.07/1.50 at the same percentiles.

Loss application and normalization. We apply $\mathcal{L}_{ssi}$ to all data sources. For metric synthetic and real sensor sources, we apply $\mathcal{L}_{mae}$. We in addition apply $\mathcal{L}_{mge}$ on dense synthetic depth, and $\mathcal{L}_{mvc}$ on multi-frame sources with camera poses.

For $\mathcal{L}_{ssi}$ and $\mathcal{L}_{mae}$, we first compute average over valid pixels within each frame, then average over frames per GPU. For $\mathcal{L}_{mge}$, we pool valid pixels across frames separately at each pyramid level and weight the levels equally. For $\mathcal{L}_{mvc}$, we average over sampled valid pixel pairs from frame pairs. Gradients are averaged equally across GPUs.

Training sources. Table 8 reports each source's type, numbers of scenes, total frames, and maximum resolution. Our internal data consists of both synthetically generated data and real-world captures. The synthetic portion features animated humans, pets, and objects in fully rendered indoor and outdoor environments under varied lighting, and supplies pixel-dense metric depth with known intrinsics and extrinsics at resolution up to 4096x3072. It includes both static scenes and dynamic scenes, rendered from multiple viewpoints and with moving cameras. The real-world portion is handheld smartphone video shot indoors and outdoors, carrying metric time-of-flight depth and camera poses.

Table 8: **Training data mixture.** Please refer to the text for a description of our internal datasets.

Dataset	Type	Scenes	Frames	Max resolution
Internal	Synthetic	964,332	24,150,620	4096x3072
Internal	Real	2,001	3,046,117	1920×1440
BEDLAM (Black et al., 2023)	Synthetic	9,454	1,452,165	1280x720
Dynamic Replica (Karaev et al., 2023)	Synthetic	477	155,100	1280x720
EDEN (Le et al., 2021)	Synthetic	694	348,062	640x480
Hypersim (Roberts et al., 2021)	Synthetic	695	68,324	1024x768
IRS (Wang et al., 2021)	Synthetic	92,976	92,976	960x540
MatrixCity (Li et al., 2023)	Synthetic	1,934	279,097	1000×1000
MegaSynth (Jiang et al., 2025)	Synthetic	64,200	3,081,600	512×512
PointOdyssey (Zheng et al., 2023)	Synthetic	938	93,798	960x540
SceneFlow (FlyingThings3D) (Mayer et al., 2016)	Synthetic	407	4,067	960x540
SceneFlow (Monkaa) (Mayer et al., 2016)	Synthetic	82	8,182	960x540
SceneFlow (Driving) (Mayer et al., 2016)	Synthetic	44	4,400	960x540
TartanAir V2 (Wang et al., 2020)	Synthetic	1,046	10,551,809	640×640
TartanAir (Wang et al., 2020)	Synthetic	2,727	272,697	640x480
TartanGround (Patel et al., 2025)	Synthetic	816	10,604,597	640×640
UrbanSyn (Gómez et al., 2025)	Synthetic	7,515	7,515	2048x1024
Virtual KITTI 2 (Cabon et al., 2020)	Synthetic	96	41,164	1242x375
Waymo Open Dataset (Sun et al., 2020)	Real	958	922,894	1920x1280

A.3 Long-video inference

Offline long-video inference. When a video is too long for one forward pass, we follow Video Depth Anything (Chen et al., 2025) and run the model on overlapping sliding windows. Each new window's inverse depth is aligned to the already predicted result by a least-squares scale and shift on the overlapping frames, then linearly interpolated between them. We include qualitative results over longer videos on an anonymous website anonymous2721692.github.io/curly-octo-dollop/.

A.4 Focal-length estimation

Architecture. We initialize our FoV estimator with a pretrained DINOv3 ViT-L/16, and finetune end-to-end on ImageNet-normalized inputs. We average the output patch tokens, then pass them through a single linear layer with two scalar outputs, representing horizontal and vertical FoV. Finally, a scaled sigmoid nonlinearity bounds each output to $[1°, 179°]$.

Training. We train on a large corpus of $\sim$92M images from Flickr, discarding those with invalid or missing EXIF data. We evaluate on the six photograph benchmarks of Depth Pro (Bochkovskii et al., 2025), which are held out during training.

Augmentation and input size. During training, we apply random aspect-ratio center cropping ($p = 0.75$; width:height in $\{0.56, 0.66, 0.75, 1, 1.33, 1.5, 1.77\}$), and a random size of at least 25% of the image along each side. Cropping narrows the FoV while keeping f_{px} fixed. We then resize the image to 256×256. Since this resizing operation changes the pixel aspect ratio, one focal length no longer describes the image. We therefore scale focal length separately along each axis, and supervise the horizontal and vertical FoV of the cropped image. Finally, we apply a horizontal flip ($p = 0.5$), along with standard photometric augmentations (color removal or jitter, blur, noise, JPEG/WebP recompression) each with probability 0.05 to 0.1.

Objective and optimization. Our objective is an L_1 loss on both angles in units of degrees. We train with AdamW, a learning rate of 5×10^{-5}, zero weight decay, and a constant schedule, with bf16 mixed precision. We use a batch size of 1920 (120×16 B200 GPUs), and train for 190k steps, which equates to roughly 4 passes over the data.

Inference. At test time, we resize inputs to 256×256 without cropping. Because crop augmentations can only narrow an image's FoV, wide angles are rare in training. The long side spans the wider angle, so we keep only the short-side FoV as the prediction. We then convert it to f_{px} and assume square pixels. For multi-frame inputs (*e.g.* video clips), we take the median predicted focal length over frames.

B Additional results

B.1 Additional qualitative examples

Figures 5 and 6 show further qualitative comparisons to representative baselines.

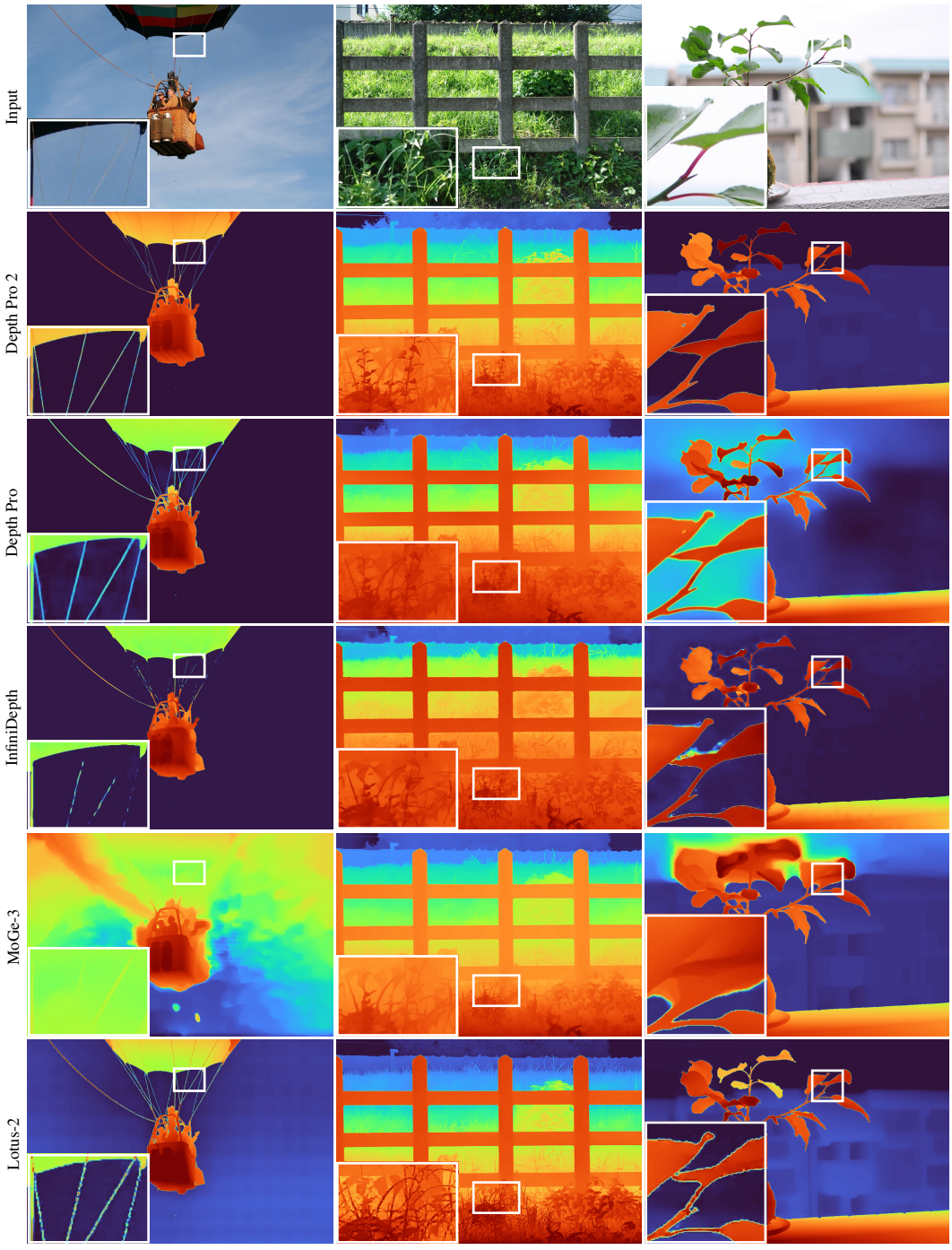

Figure 5: **Additional qualitative examples.** As in Figure 4, we compare our method to a selected few from Table 3. Insets zoom into the highlighted region. Every method runs at its own shipped default resolution.

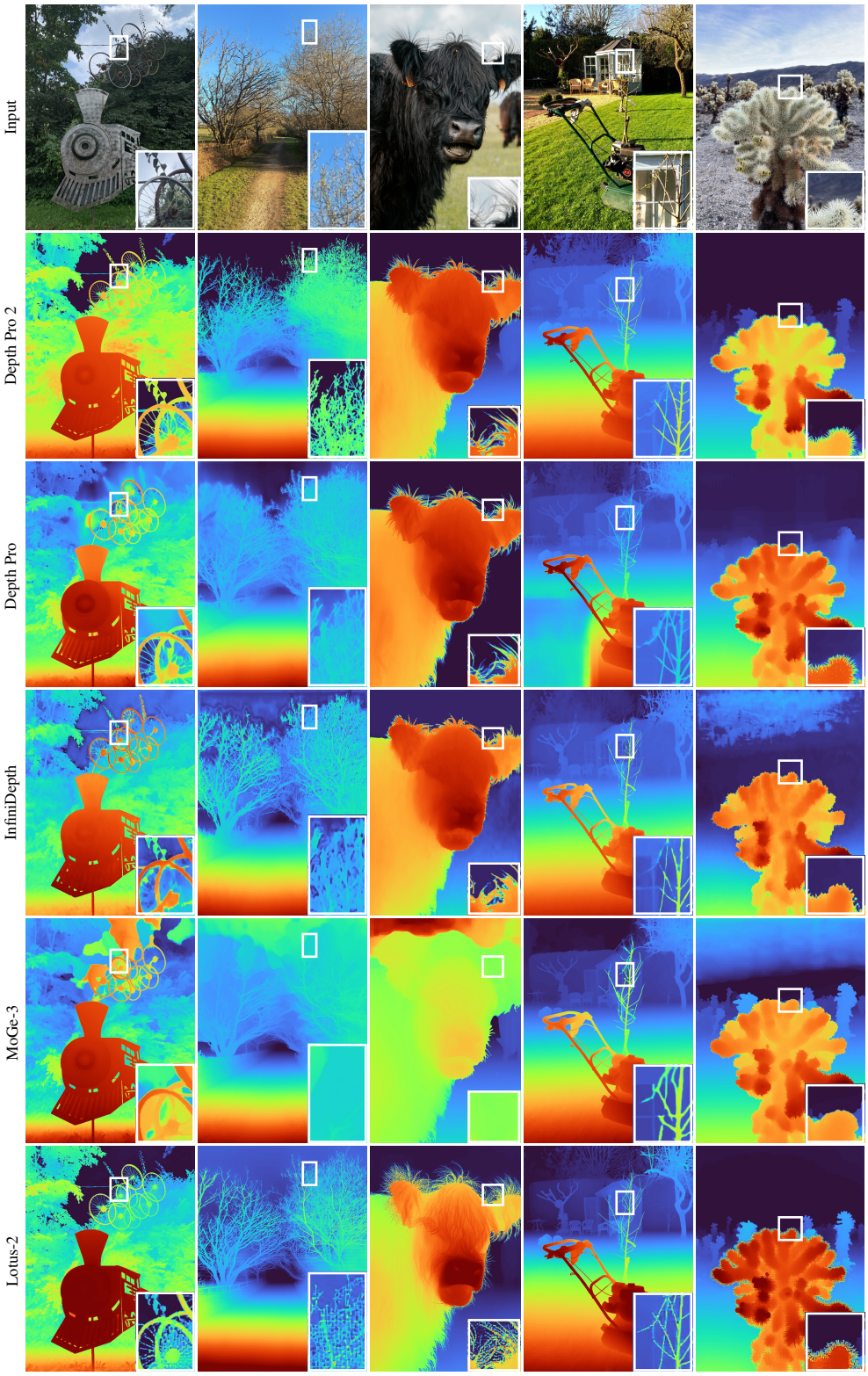

Figure 6: **Additional qualitative examples.** As in Figure 4, we compare our method to a selected few from Table 3. Insets zoom into the highlighted region. Every method runs at its own shipped default resolution.

B.2 Relative-depth accuracy

In addition to metric depth models, we extend baselines to the family of relative depth prediction models. We apply affine alignments to the ground truth and report relevant metrics.

Table 9 and 10 report δ_1 from single images and videos after the per-frame alignment described in Section C.4. Depth Pro 2-H$^+$ obtains the best average rank when evaluated on single images, and the second-best average rank when evaluated on videos, indicating its capability in both absolute to relative depth estimation.

B.3 Temporal alignment error

Table 11 reports Temporal Alignment Error (TAE) on Spring (Mehl et al., 2023), Sintel (Butler et al., 2012), Scannet (Dai et al., 2017), Bonn (Palazzolo et al., 2019) and KITTI (Geiger et al., 2012). TAE reprojects predicted depth between consecutive frames using the ground-truth camera intrinsics and poses, and measures the resulting absolute relative error (Yang et al., 2025b). Unlike our track-based *jitter* metric (Sec C.1, Table 4), this camera-induced correspondence does not account for independently moving objects and becomes ambiguous at occlusions. Consequently, ground-truth depth can have non-zero TAE. We show in Table 11 that predictions can obtain lower error than the ground truth. Ground-truth depth can also have non-zero TAE due to imprecise depth annotations (*e.g.*, from a Kinect sensor), meaning methods may be penalized for better depth coverage. In contrast, our proposed jitter metric is defined relative to the ground-truth. TAE is also biased toward smoother predictions. On 3/5 of the datasets in Table 11, we can improve Depth Pro 2-H$^+$'s TAE by simply blurring its outputs. On Sintel and Spring, the depth annotations are sufficiently dense and high-quality that we compute a mask to exclude dynamic content and occlusions, whereas the depth of ScanNet, Bonn, and KITTI are sparse and/or low-quality.

B.4 Focal-length estimation accuracy

Table 12 compares our FoV estimator against the FoV head of Depth Pro, two dedicated camera-calibration methods, AnyCalib (Tirado-Garín & Civera, 2025) and GeoCalib (Veicht et al., 2024), and three depth models that predict their own camera, UniDepth-V2 (Piccinelli et al., 2025b), UniK3D Piccinelli et al., 2025a, and Depth Anything 3 (Lin et al., 2026). Our model ranks first on all six datasets, with a mean δ_{25} of 87.9 against 71.2 for Depth Pro.

Table 9: **Zero-shot relative depth accuracy, single image.** We report δ_1 score (higher is better) under per-frame affine alignment as described in Section C.4. We evaluate 14 datasets, including datasets that have separately appeared in Table 1, 3, and 11. Every method sees one frame at a time, including video methods. We exclude three video methods: DepthCrafter (Hu et al., 2025) and Video Depth Anything cannot run on a single frame, and VeloDepth (Piccinelli et al., 2026) reduces to UniK3D (Piccinelli et al., 2025a).

	Real, indoor							Real, indoor+outdoor		Real, outdoor		Synthetic			
Method	iBims	M.bury	Booster	HAMMER	ScanNet	Bonn	TUM-D	ETH3D	DIODE	KITTI	nuSc	Sintel	Spring	DyDToF	Avg. rank↓
VGGT-Ω-1B (Wang et al., 2026a)	97.4	98.8	98.0	98.9	98.2	98.0	97.5	99.7	96.5	97.3	89.5	94.4	95.5	94.6	–
ChronoDepth (Shao et al., 2025)	83.0	89.6	86.1	82.8	87.4	83.6	77.8	86.4	82.3	77.4	68.4	66.0	58.0	83.3	20.1
DVD (Zhang et al., 2026)	96.8	99.4	98.4	96.8	95.9	96.9	87.1	97.7	94.0	92.7	84.1	74.7	77.0	93.3	17.1
Lapis *(log)* (Liu et al., 2026a)	96.3	99.0	98.4	98.7	96.2	97.2	84.0	97.8	92.7	96.4	84.8	79.0	82.6	82.1	16.3
FlashDepth-L (Chou et al., 2025)	97.7	99.2	96.9	97.1	96.3	97.7	93.1	96.4	93.5	93.7	83.5	78.2	89.3	93.9	15.9
MapAnything (Keetha et al., 2026)	96.9	98.7	97.9	97.8	97.7	97.5	90.6	98.3	94.7	95.5	88.1	75.3	97.0	85.6	15.7
VGGT-1B (Wang et al., 2025a)	97.1	98.8	97.3	98.0	98.6	97.8	91.3	98.2	95.0	96.1	86.4	76.8	**90.6**	91.5	14.7
DA3-NGL (Lin et al., 2026)	97.5	99.0	99.1	98.8	96.5	97.7	90.8	98.3	94.8	95.2	87.0	76.7	83.5	90.3	14.5
Pixel-Perfect Depth *(log)* (Xu et al., 2025)	97.9	99.4	99.0	99.4	97.6	95.6	89.1	98.3	96.0	96.2	87.6	80.4	85.6	93.9	11.2
DepthPro (Bochkovskii et al., 2025)	98.0	99.6	99.5	99.2	98.2	97.9	92.1	97.7	95.7	97.4	87.1	83.2	75.8	92.8	10.8
π^3 (Wang et al., 2026b)	97.7	99.1	97.9	99.1	98.8	**98.6**	93.3	98.8	96.2	96.9	88.7	80.6	88.0	93.2	10.4
Lotus-2 (He et al., 2026)	98.1	99.7	98.9	99.3	97.8	98.0	92.0	98.4	94.9	95.1	86.2	79.8	82.0	96.0	10.2
DAGE (Ngo et al., 2026)	98.2	99.5	98.5	98.5	98.8	98.0	92.5	99.1	96.4	97.8	89.2	80.1	97.4	93.0	9.8
InfiniDepth (Yu et al., 2026)	98.1	99.7	98.4	98.0	97.6	97.9	93.8	98.8	95.9	96.9	87.6	82.2	77.2	96.7	9.4
MetricAnything (Ma et al., 2026)	98.5	99.7	98.9	99.1	99.2	97.3	93.7	98.7	96.6	97.3	89.8	78.1	82.4	95.5	9.0
Marigold V2 *(log)* (Pavlovic et al., 2026)	**98.6**	99.3	94.3	**99.5**	98.3	97.8	93.7	98.4	96.9	97.6	85.5	83.3	84.2	96.9	7.9
MoGe-2-L (Wang et al., 2025c)	98.3	99.7	99.5	99.1	98.5	97.9	86.5	99.1	96.4	98.0	89.4	81.4	85.7	95.2	7.1
UniDepthV2-L (Piccinelli et al., 2025b)	98.2	99.7	98.9	99.4	98.5	97.8	93.5	99.4	96.2	97.9	92.5	83.8	82.7	96.1	6.2
FoundationGeo (Liu et al., 2026b)	98.1	99.6	99.3	99.2	98.6	98.5	**96.1**	99.1	96.5	98.3	89.2	82.6	96.7	94.0	5.7
UniK3D-L (Piccinelli et al., 2025a)	98.3	99.7	99.0	99.4	98.5	97.8	93.9	99.3	96.7	98.2	**92.7**	83.6	84.2	96.4	4.9
MoGe-3-G (Kong et al., 2026)	**98.6**	99.8	99.0	**99.5**	97.4	98.1	88.5	99.5	96.5	98.2	90.2	85.4	90.2	**97.4**	4.4
Depth Pro 2-H$^+$	**98.6**	**99.9**	**99.7**	99.2	**98.8**	98.0	93.5	**99.6**	**97.6**	**98.4**	90.1	**86.0**	89.8	95.2	**3.1**

Table 10: **Zero-shot relative depth accuracy, video.** The same δ_1 score on video datasets. Every method receives a clip of up to 50 frames as input in one forward pass. Video methods report different numbers here compared to Table 9 due to different input setup.

	Real, indoor			Real, outdoor	Synthetic			
Method	ScanNet	Bonn	TUM-D	KITTI	Sintel	Spring	DyDToF	Avg. rank↓
VGGT-Ω-1B (Wang et al., 2026a)	98.7	97.9	97.6	98.7	96.2	96.6	98.4	–
ChronoDepth (Shao et al., 2025)	87.9	83.3	74.8	73.6	67.5	56.1	80.3	22.3
Lapis *(log)* (Liu et al., 2026a)	96.2	97.2	84.0	96.4	79.0	82.6	82.1	18.1
VeloDepth (Piccinelli et al., 2026)	95.3	94.2	93.6	91.9	77.6	81.4	90.8	18.1
FlashDepth-L (Chou et al., 2025)	96.3	97.7	93.1	93.7	78.2	89.3	93.9	16.7
MapAnything (Keetha et al., 2026)	98.4	97.6	91.9	97.0	77.7	97.2	88.2	16.5
DepthCrafter (Hu et al., 2025)	89.3	98.3	85.5	91.6	80.6	81.7	88.8	16.0
Pixel-Perfect Depth *(log)* (Xu et al., 2025)	97.5	95.1	89.1	96.2	80.4	85.6	93.9	15.1
MetricAnything (Ma et al., 2026)	99.2	97.3	93.7	97.3	78.1	82.4	95.5	14.2
DVD (Zhang et al., 2026)	98.1	98.0	88.1	97.3	78.1	79.7	95.5	14.0
VDA-L (Chen et al., 2025)	98.0	97.7	93.9	95.0	79.4	83.9	91.6	13.6
DepthPro (Bochkovskii et al., 2025)	98.2	97.9	92.1	97.4	83.2	75.8	92.8	13.0
Lotus-2 (He et al., 2026)	97.8	98.0	92.0	95.1	79.8	82.0	96.0	12.9
VGGT-1B (Wang et al., 2025a)	98.7	97.8	91.9	97.3	80.3	**91.0**	97.0	11.2
InfiniDepth (Yu et al., 2026)	97.6	97.9	93.8	96.9	82.2	77.2	96.7	11.0
MoGe-2-L (Wang et al., 2025c)	98.5	97.9	86.5	98.0	81.4	85.7	95.2	10.6
UniDepthV2-L (Piccinelli et al., 2025b)	98.5	97.8	93.5	97.9	83.8	82.7	96.1	9.2
Marigold V2 *(log)* (Pavlovic et al., 2026)	98.3	97.8	93.7	97.6	83.3	84.2	96.9	7.7
MoGe-3-G (Kong et al., 2026)	97.4	98.1	88.5	98.2	85.4	90.2	97.4	7.6
DA3-NGL (Lin et al., 2026)	98.3	98.1	93.4	97.6	82.3	87.9	97.1	7.5
UniK3D-L (Piccinelli et al., 2025a)	98.5	97.8	93.9	98.2	83.6	84.2	96.4	7.5
DAGE (Ngo et al., 2026)	99.3	98.4	93.2	98.2	82.2	97.7	**98.6**	6.0
FoundationGeo (Liu et al., 2026b)	98.6	**98.5**	**96.1**	98.3	82.6	96.7	94.0	5.0
π^3 (Wang et al., 2026b)	99.2	98.2	95.2	98.5	85.0	90.7	97.9	**2.8**
Depth Pro 2-H$^+$	**99.2**	98.1	95.3	**98.7**	**88.8**	90.7	95.7	3.3

Table 11: **Temporal alignment error (TAE)** after a single affine alignment per video. Lower is better. TAE measures how well a prediction can be explained by the camera motion annotations. On Sintel and Spring, the depth annotations are accurate and *dense*, enabling us to compute TAE only over unoccluded, static content. For datasets with sparse and/or low-quality depth annotations, we argue that TAE is an unreliable metric since it favors blurrier predictions, and assigns non-zero error to the ground-truth depth. The ground-truth and blurred-depth reference rows are excluded from the average rank, as are any non-zero-shot results (which appear greyed-out). We discuss in Sec B.3.

Method	Sintel	Spring	ScanNet	Bonn	KITTI	Avg. rank↓
Ground-truth depth	0.13	0.06	0.44	2.64	5.91	–
Depth Pro 2-H$^+$, blurred (σ = 2% width)	8.74	0.91	0.42	3.15	7.24	–
VGGT-Ω-1B (Wang et al., 2026a)	2.26	0.93	0.58	3.68	10.51	–
DepthPro (Bochkovskii et al., 2025)	14.80	10.31	2.09	5.50	11.38	22.2
InfiniDepth (Yu et al., 2026)	9.87	11.28	2.60	5.28	11.69	22.1
Pixel-Perfect Depth *(log)* (Xu et al., 2025)	11.28	10.20	2.60	5.10	13.33	22.1
Lapis *(log)* (Liu et al., 2026a)	8.19	4.17	2.96	5.05	10.31	18.0
MoGe-2-L (Wang et al., 2025c)	7.48	3.64	2.13	5.07	10.60	17.8
FoundationGeo (Liu et al., 2026b)	8.25	2.98	2.04	4.94	11.24	17.0
MoGe-3-G (Kong et al., 2026)	7.34	3.12	2.72	4.97	10.53	17.0
Marigold V2 *(log)* (Pavlovic et al., 2026)	8.76	3.08	1.35	4.61	11.05	16.6
FlashDepth-L (Chou et al., 2025)	10.95	2.29	1.53	4.40	10.40	14.4
MapAnything (Keetha et al., 2026)	13.72	1.13	1.09	4.97	9.90	13.9
UniK3D-L (Piccinelli et al., 2025a)	7.29	4.73	1.20	4.44	9.87	13.2
UniDepthV2-L (Piccinelli et al., 2025b)	7.07	3.36	1.33	4.48	9.72	12.6
Lotus-2 (He et al., 2026)	7.92	3.12	1.07	4.86	9.46	12.5
MetricAnything (Ma et al., 2026)	5.95	2.91	0.95	4.34	10.75	11.5
ChronoDepth (Shao et al., 2025)	4.33	3.02	0.99	7.78	**7.55**	11.2
DVD (Zhang et al., 2026)	4.05	1.48	0.48	4.39	11.32	10.2
DA3-L (Lin et al., 2026)	5.07	1.25	0.83	4.20	10.41	9.0
DepthCrafter (Hu et al., 2025)	4.55	0.97	0.66	4.30	10.75	9.0
π^3 (Wang et al., 2026b)	3.70	1.04	**0.39**	4.21	10.37	5.6
VGGT-1B (Wang et al., 2025a)	4.59	0.91	0.46	3.57	9.88	4.9
DAGE (Ngo et al., 2026)	3.28	0.96	0.46	4.00	10.20	4.7
VDA-L (Chen et al., 2025)	3.95	0.77	0.55	3.84	9.64	4.0
VeloDepth (Piccinelli et al., 2026)	6.96	**0.63**	0.43	**3.49**	7.88	**3.4**
Depth Pro 2-H$^+$	**2.04**	0.90	0.51	4.30	10.96	7.1

Table 12: **Focal length estimation on photographs.** We evaluate focal length estimation accuracy on 6 datasets (Bochkovskii et al., 2025). δ_{25} and δ_{50} are the percentages of images whose relative focal-length error is below 25% and 50%; higher is better. Mean and rank use δ_{25} over all six datasets. Rank is the mean per-dataset rank for δ_{25}, where lower is better.

	DDDP		FiveK		PPR10K		RAISE		SPAQ		ZOOM		Mean	Rank
Method	δ_{25}	δ_{50}	δ_{25}	δ_{50}	δ_{25}	δ_{50}	δ_{25}	δ_{50}	δ_{25}	δ_{50}	δ_{25}	δ_{50}	δ_{25} ↑	↓
Dedicated focal estimators														
Depth Pro 2	**90.8**	**96.6**	**89.4**	**98.2**	**89.6**	**98.0**	**94.8**	**99.2**	**79.2**	**91.8**	**83.3**	**98.6**	**87.9**	**1.00**
DepthPro (Bochkovskii et al., 2025)	63.2	87.1	75.0	93.4	64.6	87.4	83.0	97.6	70.8	86.8	70.7	91.7	71.2	2.33
AnyCalib (Tirado-Garín & Civera, 2025)	36.3	86.5	54.4	78.8	30.0	72.6	77.2	91.2	67.8	81.8	58.6	77.6	54.1	4.17
GeoCalib (Veicht et al., 2024)	33.1	74.1	48.2	74.2	40.0	78.6	56.8	83.2	53.6	73.0	39.4	66.7	45.2	6.17
Depth models with a camera head														
UniDepthV2-L (Piccinelli et al., 2025b)	13.1	47.3	48.4	76.8	52.2	86.8	74.0	90.0	65.1	83.5	33.9	64.9	47.8	6.00
UniK3D-L (Piccinelli et al., 2025a)	27.8	77.6	52.6	83.8	67.8	92.2	74.4	93.0	75.2	87.6	45.1	76.4	57.2	3.67
DA3-NGL (Lin et al., 2026)	27.1	66.4	55.9	81.8	52.8	84.8	66.6	89.8	69.2	86.0	44.8	68.4	52.7	4.67

B.5 DIAGNOSTIC STUDIES

We take the trained Depth Pro 2-H+ and study the inference-time behavior.

Table 13: **Cross-frame attention vs temporal consistency.** We evaluate Depth Pro 2-H+ on 32-frame clips from DynamicReplica, while randomly dropping cross-frame attention between pairs of frames. All blocks follow the same dropout pattern throughout the forward pass. We report metrics over three draws. Jitter rises steadily as more connections are removed, while accuracy and boundary F1 hold until cross-frame attention is fully disabled. Bold: best.

Connections dropped	Metric δ_1 (%) ↑	Relative δ_1 (%) ↑	Boundary F1 (%) ↑	Jitter ↓
0 %	**99.6**	**99.6**	**58.3**	**0.23**
20 %	**99.6** ± 0.1	**99.6** ± 0.1	58.2 ± 0.1	0.27 ± 0.01
40 %	**99.6** ± 0.1	**99.6** ± 0.1	58.1 ± 0.1	0.33 ± 0.02
60 %	**99.6** ± 0.1	**99.6** ± 0.1	57.8 ± 0.1	0.46 ± 0.01
80 %	**99.6** ± 0.1	**99.6** ± 0.1	57.2 ± 0.1	0.75 ± 0.02
100 %	98.8	99.3	53.9	0.83

Cross-frame attention vs. temporal consistency. To probe the causal relationship between the cross-frame attention mechanism and temporal consistency, we randomly drop connections between frames with a certain probability. The dropout pattern is consistent across cross-frame attention blocks throughout a forward pass. We repeat three times with different dropout patterns and report the mean and standard deviation. We use a held-out set of DynamicReplica sequences for this evaluation. We report metric depth accuracy, relative depth accuracy, boundary F1, and jitter in Table 13. As we remove more connections, jitter rises monotonically from 0.23 to 0.83. This suggests the cross-frame attention mechanism is key to our model for temporally consistent depth estimation. In addition, sharpness also degrades as more connections are dropped, suggesting the model uses information from other frames to predict sharp boundary as well.

Input resolution vs. sharpness. To quantify the correlation between resolution and sharpness, we vary the input resolution to see how that impacts metrics. We resize the shorter side of every input image to fixed sizes, i.e., 360px, 540px, 720px, or remain at native resolution. For this analysis, we evaluate on Waymo, ARKitScenes, and DynamicReplica and report metric depth accuracy, relative depth accuracy, boundary sharpness, and temporal consistency, as shown in Table 14. We find that from 360px to native, the boundary F1 score more than doubles, from 24.2 to 58.3. Jitter also improves as resolution increases from 0.31 down to 0.23. Metric and relative depth accuracy also see slight but consistent improvement through scaling. The shows a strong positive correlation between input resolution and boundary sharpness, and shows it is best to always run inference at native resolution, when runtime and memory permit.

Long-video inference. For long videos that cannot fit into one forward pass, we run our model on sliding windows that overlap, as described in Appendix A.3. The window size and the overlap are corresponding hyperparameters. We provide an empirical study with the DynamicReplica dataset, whose sequences go up to 300 frames. We compare processing all 300 frames at once to running sliding windows with different window size and different overlap. In Table 15, we report full metrics from common sliding window configurations. Running all 300 frames at once achieves the best depth accuracies and the sharpest boundary, however, the temporal consistency is slightly worse than the best sliding window strategy, potentially due to cross-frame attention over 300 frames being out of training distribution, while the other metrics tend to be less sensitive to temporal context. Sliding window, on the other hand, is much more cost effective. It is important to have overlap between sliding windows, as zero overlap degrades temporal consistency (0.281 to 0.303, against 0.270 for full). With an overlap of 4 frames, we can already outperform full in consistency over 16-frame sliding windows, while producing at 4x throughput and consuming one third of the memory. In addition, our model is in general robust to different sliding window configurations.

B.6 CONTROLLED EXPERIMENTS.

Evaluation setup. We perform a series of controlled experiments, evaluated on held-out validation sequences: Waymo (Sun et al., 2020) (real, outdoor; 42 sequences at 1280x1920 or 886x1920) and

Table 14: **Input resolution vs. sharpness.** We evaluate Depth Pro 2-H+ at different input resolution, processing 32-frame windows from DynamicReplica. Native resolution means 720 px for DynamicReplica, 886px or 1280px for Waymo, and 1440px for ARKitScenes. Boundary F1 score more than doubles as input resolution increases. Bold: best.

	Metric δ_1 (%) ↑		Relative δ_1 (%) ↑		Boundary F1 (%) ↑	Jitter ↓	FPS ↑	GiB ↓
Input short side	Waymo	ARKitScenes	Waymo	ARKitScenes	DynamicReplica			
360	94.3	90.4	94.8	99.0	24.2	0.31	**80.3**	**16.4**
540	95.0	91.3	95.5	**99.2**	38.3	0.27	35.2	30.8
720	95.0	**92.5**	95.6	**99.2**	**58.3**	**0.23**	16.3	43.9
Native	**95.4**	92.4	**95.8**	**99.2**	**58.3**	**0.23**	16.2	43.9

Table 15: **Offline long-video inference.** We evaluate Depth Pro 2-H+ on 300-frame sequences from DynamicReplica, over different configurations of sliding windows that overlap, including processing all 300 frames in one go. Sliding window achieves similar often better temporal consistency compared to processing all frames at once, at only a fraction of the cost. A minimal overlap of 4 frames is key to maintaining temporal consistency. Bold: best.

Window	Overlap	Metric δ_1 (%) ↑	Relative δ_1 (%) ↑	Boundary F1 (%) ↑	Jitter ↓	FPS ↑	GiB ↓
Full	–	**99.6**	**99.7**	**58.7**	0.270	3.2	93.8
16	0	**99.6**	99.6	57.9	0.303	**16.0**	**31.0**
	2	99.5	99.6	58.0	0.274	14.7	**31.0**
	4	99.5	99.6	58.0	0.269	13.2	**31.0**
	8	99.5	99.6	58.0	0.263	9.0	**31.0**
32	0	**99.6**	99.6	58.3	0.286	14.5	32.3
	4	99.4	99.6	58.4	0.263	12.8	32.3
	8	99.5	99.6	58.4	0.261	10.9	32.3
	16	99.4	99.6	58.4	0.258	8.0	32.3
48	0	**99.6**	99.6	58.5	0.281	12.6	33.5
	8	**99.6**	99.6	58.6	0.261	9.9	33.5
	12	99.5	99.6	58.6	0.260	9.8	33.5
	24	99.5	99.6	58.6	**0.257**	6.6	33.5

ARKitScenes (Baruch et al., 2021) (real, indoor; 92 sequences at 1440x1920) for depth accuracy, and the synthetic DynamicReplica (Karaev et al., 2023) (44 at sequences at 720x1280) for boundary sharpness and temporal consistency. ARKitScenes is not trained on except in condition D1. From each sequence, we take one window of 32 consecutive frames and run it at native resolution in a single forward pass. We report metric δ_1 using ground-truth focal length, relative δ_1 after a per-frame affine fit in disparity (inverse depth) space, the boundary F1 of Bochkovskii et al. (2025), and jitter (Eq. 3) along point tracks obtained by chaining DynamicReplica's ground-truth optical flow. None of these datasets is used in the zero-shot evaluation.

Baseline. The baseline condition (A1) is a ViT-L based model trained for 100K steps using 8 GPUs. We train the baseline 3 times each with a different random seed. We report the mean and standard deviation. Results are in Table 16. We describe our findings below.

Scaling model size and batch size. For the same model, increasing the number of GPUs from 8 (A1) to 128 (B3), and thus the batch size, improves nearly all metrics. At 128 GPUs, performance improves monotonically with model size from Depth Pro 2-S (B1) to Depth Pro 2-H+ (B4) on every metric.

Training data. To assess the effect of the training data mix, we conduct an experiment in which we add more real-world data (D1), remove all real-world data (D2), remove all external data (D3), and remove all internal data (D4).

For D1, we add real-world datasets. Specifically, we add Objectron (Ahmadyan et al., 2021), Argoverse2 (Wilson et al., 2021), MapFree (Arnold et al., 2022), ARKitScenes (Dehghan et al., 2021), WayveScenes101 (Zürn et al., 2024), TartanRGBT (Maheshwari et al., 2026), the Aria captures

Table 16: **Controlled experiments.** Results on held-out validation sequences from Waymo (Sun et al., 2020), ARKitScenes (Baruch et al., 2021), and DynamicReplica (Karaev et al., 2023). Baseline A1 at 100k steps and A2 at 50k. The baseline is averaged across 3 runs with different random seeds, showing mean ± standard deviation σ. Differences larger than 1σ is tinted green (better) or red (worse). δ_1 and F1 are in %.

		Metric δ_1 (%) ↑		Relative δ_1 (%) ↑		Boundary F1 (%) ↑	Jitter ↓
		Waymo	ARKitScenes	Waymo	ARKitScenes	DynamicReplica	
100k steps							
A1	Baseline: Depth Pro 2-L, 8 GPUs	91.1 ±0.6	77.0 ±3.8	92.8 ±0.1	97.9 ±0.1	50.9 ±0.1	0.34 ±0.01
	Model size and compute						
B1	Depth Pro 2-S, 128 GPUs	−7.4	−11.9	−3.0	−3.9	−11.8	+0.08
B2	Depth Pro 2-B, 128 GPUs	+0.8	−8.1	−0.1	−1.5	−0.3	−0.02
B3	Depth Pro 2-L, 128 GPUs	+2.8	−0.4	+1.7	+0.8	+5.6	−0.09
B4	Depth Pro 2-H$^+$, 128 GPUs	+4.3	+15.4	+3.0	+1.3	+7.4	−0.11
	Training data						
D1	additional real-world data	+0.2	+14.3	+0.5	+0.7	−1.1	+0.02
D2	synthetic data only	−5.5	−12.3	−3.0	−0.5	+0.4	−0.01
D3	internal data only	−16.8	−7.2	−7.0	−0.4	−4.3	+0.06
D4	external data only	+3.2	−45.2	+2.0	−0.8	+4.5	−0.06
	Training crops						
E1	any aspect ratio, stride 1	+0.4	+0.6	+0.2	0.0	−0.4	0.00
E2	square, native scale	−46.6	−16.2	−1.1	−0.2	−2.1	+0.05
E3	square, resized to 1536^2	−48.5	−6.3	−1.4	+0.1	−5.8	0.00
	Backbone						
H1	DINOv2	−0.2	−12.8	+0.1	−0.4	−0.2	−0.02
H2	frozen DINOv3	−13.1	−29.4	−4.4	−6.7	−7.1	+0.32
	Training losses						
I1	multi-view consistency loss off	+0.9	+1.2	+0.4	+0.1	+0.2	+0.05
50k steps							
A2	Baseline: Depth Pro 2-L, 8 GPUs	88.3 ±1.6	71.4 ±1.4	91.7 ±0.3	97.1 ±0.1	46.3 ±0.5	0.41 ±0.03
	Training losses						
J2	Depth Pro's 1st-order boundary loss	0.0	+0.5	−0.1	+0.5	−4.1	0.00
J3	Depth Pro's full boundary losses	−26.9	−9.7	−1.4	0.0	−1.7	+0.03
J4	confidence-weighted losses	+0.6	−2.7	−0.1	−0.3	−9.1	+0.06

(ADT (Pan et al., 2023), HOT3D (Banerjee et al., 2025), Nymeria (Ma et al., 2024), Reading In the Wild (Yang et al., 2025a)), and additional internal captures.

We find that adding real-world data (D1) improves accuracy, but yields smoother boundaries, which we attribute to the commonly lower quality of ground truth around occlusion boundaries due to sensor measurements. Removing real-world data entirely (D2) hence leads to even sharper boundaries than the baseline data mix produces, but degrades depth accuracy.

Training on internal data only (D3) leads to worse performance on all metrics. Removing all internal data (D4) increases accuracy on Waymo, and boundary sharpness and jitter on DynamicReplica. It strongly degrades accuracy on ARKitScenes. This suggests that the internal data improves generalization but comes at the cost of reduced boundary sharpness and temporal consistency.

Training resolution. Our baseline (A1) trains on image crops of any aspect ratio at native scale and subsamples some with a stride of 2, adding lower-resolution views without interpolating ground truth. Removing the strided subsampling (E1) slightly improves accuracy, but also degrades boundary sharpness.

Training on only square crops at native scale (E2) keeps most of the sharpness, yet metric δ_1 on Waymo drops by 46.6%, suggesting the variety of aspect ratios and fields of views is crucial for the model to recover metric scale.

Further resizing the square crops to a fixed resolution of 1536^2, similar to *e.g.* Depth Pro (Bochkovskii et al., 2025) (E3) further reduces accuracy and boundary sharpness, and confirms the importance of processing at native resolution.

Backbone. Replacing the pretrained DINOv3 with DINOv2 ViT-L/14 (H1) slightly improves temporal consistency but lowers metric δ_1 on ARKitScenes by 12.8 points. Freezing DINOv3 (H2) degrades every metric.

Training losses. At 50K steps (A2), turning off the multi-view consistency loss (I1) degrades temporal consistency by +0.05. Replacing our gradient loss with the first-order boundary loss of Depth Pro (J2), the same gradient error at 6 scales instead of 3, lowers boundary F1 by 4.1 points, so 3 scales suffice. The full set of Depth Pro boundary losses (J3), which adds second-order and squared gradient terms, does not improve sharpness either and costs 26.9 points of metric δ_1 on Waymo. Confidence-weighting our inverse-depth losses (J4), as feed-forward reconstruction methods do, lowers boundary F1 by 9.1 points. We posit that boundaries are where predictions are most uncertain, and letting the model discount its loss there discourages it from learning to resolve them.

C Evaluation protocol

C.1 Temporal consistency.

We evaluate temporal stability along point tracks computed on synthetic data. Let track k be one such point, seen at pixel $p_k(i)$ in frame i. We measure the residual $e_k(i)$ between predicted and true log-depth, and define its change from one frame to the next as jitter_k:

$$e_k(i) = \log \hat{\mathbf{Z}}_i(p_k(i)) - \log \mathbf{Z}_i(p_k(i)), \qquad \mathrm{jitter}_k = \mathrm{mean}_i \big| e_k(i{+}1) - e_k(i) \big|, \tag{3}$$

jitter_k is identical to TEPE from (Karaev et al., 2023; Zhu et al., 2025), but computed in log-space so values are comparable across varying scene scales.

C.2 Evaluation details

Zero-shot status. Results are shown in gray when the corresponding checkpoint was trained on that evaluation dataset. Such entries are excluded when assigning per-dataset ranks and when computing average ranks across datasets. A method with non-zero-shot entries is therefore averaged over only the datasets on which it is zero-shot.

Single- and multi-frame input. Our model can process both single-frame and multi-frame input. For a fair evaluation between single image and multi-image approaches, we split the evaluation of metric depth accuracy and boundary sharpness into separate settings. In the single-image setting, every method receives one frame at a time; on video and multi-view datasets, each of the first 50 frames of a sequence is predicted and scored as a separate image, and models estimate the focal length independently for every frame.

For boundary sharpness and temporal consistency, we follow the same single-image and video split. The depth boundary protocol of Bochkovskii et al. (2025) is scale-invariant, which lets us compare against methods that predict only relative depth. Our temporal-consistency metrics are scale-invariant as well, so relative-depth estimators are admitted there too.

Baseline evaluation. Unless noted otherwise, we evaluate baseline checkpoints using the protocols described below. Our scores may differ from published results because studies use different camera intrinsics, alignment procedures, sampled frames or sequences, and evaluation resolutions.

DIODE evaluation split. Lin et al. (2026) and Piccinelli et al. (2025a) evaluate on the indoor split of DIODE (Vasiljevic et al., 2019) only. We evaluate on the full validation split, indoor and outdoor, where the outdoor scenes are harder and lower the scores. We apply the ground-truth validity mask of Pixel-Perfect Depth and Marigold V2, following their DIODE protocol.

In addition, we follow Bochkovskii et al. (2025) for their evaluation split of nuScenes and Kong et al. (2026) for their evaluation split of HAMMER. As a result, both are treated as a single-image dataset in our evaluation.

C.3 Baseline configurations

Common input protocol. Prior work evaluates video depth using different frame-sampling schemes, clip lengths, and alignment conventions, making published results difficult to compare directly. For example, VGGT-Ω evaluates 10 randomly sampled frames from each sequence (Wang et al., 2026a, §4), whereas Video Depth Anything evaluates up to 500 frames per video, also reports 90- or 110-frame settings for comparison with DepthCrafter, and aligns predictions in inverse-depth

space (Chen et al., 2025). We therefore evaluate every method on identical source frames: one contiguous clip per sequence, sampled at stride one, using 50 frames for each video sequence.

Clip processing. Each method consumes the common input clip using its official inference procedure. Methods that cannot process the entire clip in one pass retain their released windowing strategy. For example, Video Depth Anything processes the input in 32-frame chunks, whereas methods that support the complete clip receive all 50 frames in one pass.

Input resolution. We retain each method's official inference resolution because changing it can affect metric accuracy, boundary sharpness, temporal consistency, and runtime. For example, Depth Pro resizes every frame to 1536×1536, whereas DepthCrafter uses a dataset-specific resolution. For methods whose official preprocessing crops the input (*e.g.*, VGGT-Ω), we first pad each frame and remove the corresponding padding from the prediction, ensuring that every method is evaluated on the same image region.

C.4 Intrinsics, resizing, and alignment

Camera intrinsics. For metric-depth evaluation, we report separate results using ground-truth and predicted camera intrinsics. A method is included in the predicted-intrinsics setting only if it estimates focal length. Metric-depth predictions are evaluated without scale or shift alignment.

Output resizing. Because methods differ in their internal preprocessing, their predictions may not match the spatial resolution of the ground truth. In such cases, we resize the prediction to the ground-truth resolution using nearest-neighbor interpolation, thereby avoiding additional smoothing at depth discontinuities.

Affine alignment. The boundary-sharpness and temporal-consistency evaluations also include relative-depth methods, whose predictions must be aligned to the ground truth. For boundary evaluation, we estimate a separate affine transformation for each frame of a relative-depth method. Boundary F1 scores only pixel pairs for which both ground-truth depths are valid. For temporal evaluation, we estimate a single affine transformation per video for relative-depth methods. Aligning each frame independently would re-anchor the prediction at every time step and suppress the temporal scale variation that the temporal consistency metrics intend to measure.

For most methods, we fit scale and shift in inverse depth space by trimmed least-squares, retaining the 80% of sampled valid pixels with the smallest squared residuals (20% trimming). For methods whose native output is log depth (Lapis, Pixel-Perfect Depth, Marigold V2), we apply the same trimming rule in log-depth space, following their respective evaluation protocols. These cases are marked with "(log)" in Tables 3(a), 3(b), and 4.

D Applications

D.1 Bokeh

Evaluation protocol. We evaluate depth-guided bokeh rendering, where depth errors near object boundaries produce visible foreground–background blending artifacts. We hold the renderer fixed and vary only its input depth, thereby evaluating the downstream utility of each estimator rather than comparing end-to-end bokeh-generation systems.

We use the 220 test scenes from RealBokeh (Seizinger et al., 2025), each containing an $f/22$ source image and a corresponding real $f/2$ photograph. For each source image, we estimate metric depth and provide inverse metric depth to a frozen BokehMe renderer (Peng et al., 2022). For Depth Pro 2, we convert focal-normalized disparity using its predicted focal length; baselines are converted according to their native metric representations. We apply no depth renormalization or per-image alignment. Non-metric methods are excluded because their missing scale and shift cannot be recovered without scene-specific fitting. We also exclude MetricAnything, which requires a supplied focal length to produce metric depth, whereas no camera intrinsics are provided to any method in this study.

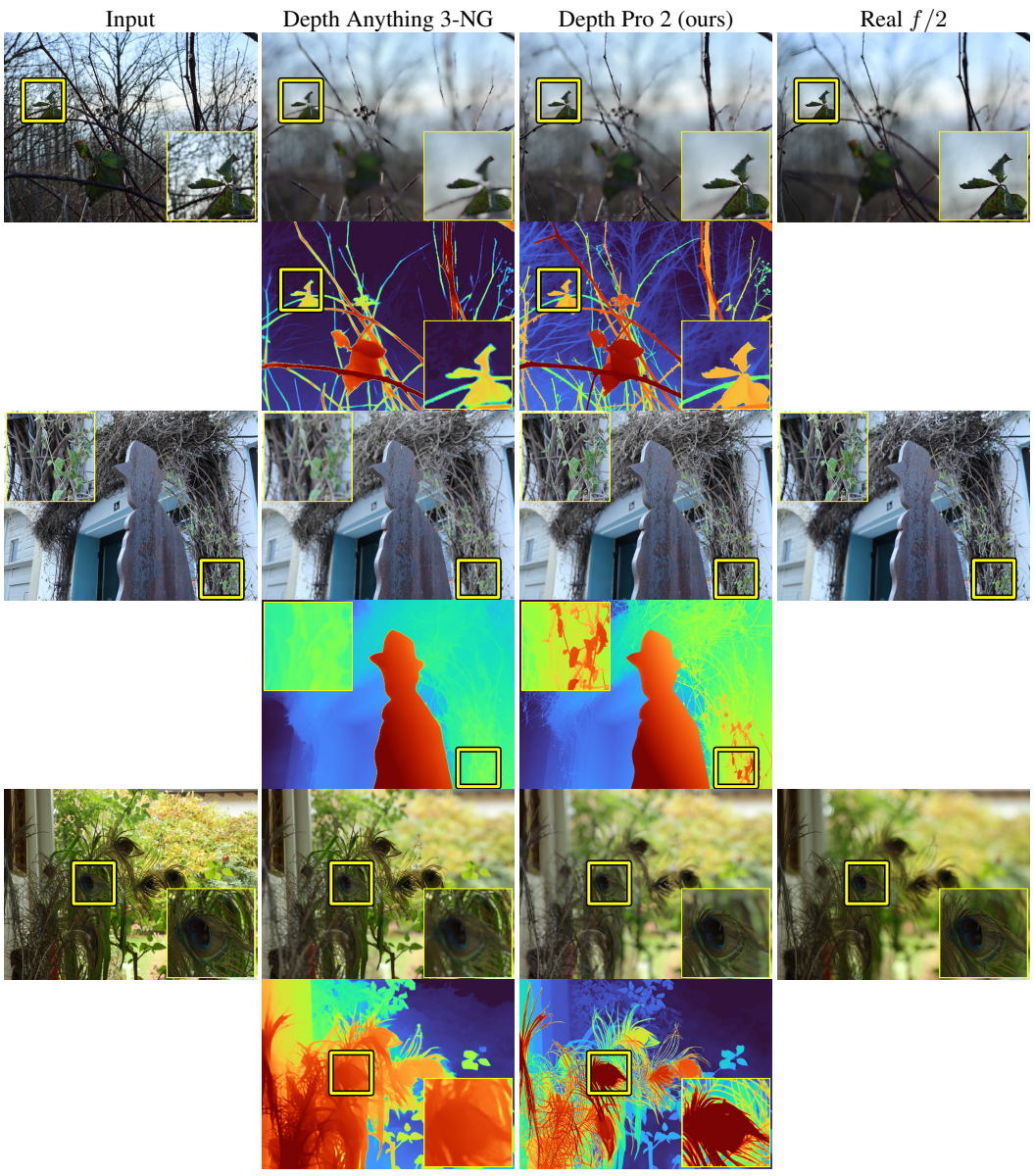

Figure 7: **Depth-guided bokeh rendering on RealBokeh.** Each scene shows the rendered result followed by predicted disparity. Every panel contains a $4\times$ enlargement of its boxed region.



Figure 8: **Depth-guided bokeh on video.** Rows show the input and the outputs rendered from each depth estimator; columns show four time steps of one handheld clip. Each panel includes a $4\times$ nearest-neighbor close-up of the boxed region.

Table 17: **Depth-guided bokeh rendering on RealBokeh.** Metric depth estimates from each method are provided to the same frozen BokehMe renderer without per-image alignment or fitting. Results are averaged over 220 test scenes and ordered by Boundary LPIPS. $\sim$ denotes an unreported inference resolution inferred from the method's configuration constraints. † Run at the native input resolution.

Depth source	Inference resolution	PSNR ↑	SSIM ↑	LPIPS ↓	Boundary LPIPS ↓	DISTS ↓
Depth Anything 3-NG †	2002×1498	22.45	0.8311	0.3412	0.6327	0.2902
UniK3D-L	$\sim 896 \times 672$	23.01	0.8407	0.3199	0.5571	0.2484
MoGe-2-L	966×728	23.27	0.8381	0.3195	0.5375	0.2259
UniDepthV2-L	$\sim 896 \times 672$	23.22	0.8446	0.3099	0.5364	0.2356
MoGe-3-L	966×728	23.28	0.8399	0.3166	0.5351	0.2222
Depth Pro	1536×1536	23.24	0.8431	0.3123	0.5336	0.2311
MoGe-3-G	966×728	23.29	0.8414	0.3133	0.5250	0.2225
Depth Anything 3-NG	504×378	23.44	0.8478	0.3061	0.5047	0.2296
Depth Pro 2-H$^+$	2000×1500	**23.89**	**0.8585**	**0.2900**	**0.4712**	**0.2154**

All images are rendered and evaluated at 2000×1500 resolution. The renderer and all its parameters are fixed across methods, including one global disparity multiplier selected on a disjoint validation set of 60 scenes. No parameters are fitted on the test set.

Boundary LPIPS. In addition to full-image PSNR, SSIM, LPIPS (Zhang et al., 2018), and DISTS (Ding et al., 2022), we measure LPIPS within a 16-pixel band around the capture's defocus boundaries. This band is derived solely from the $f/22$ source and $f/2$ target images and is therefore identical across methods. We first exposure-align the target luma to the source using a global affine least-squares fit. For each image, we compute local high-frequency energy as $E = \mathrm{box}_{15}((\nabla^2 \ell)^2)$. Pixels whose source energy exceeds its 60th percentile are classified as in focus when $\rho = E_{\mathrm{tgt}}/(E_{\mathrm{tgt}} + E_{\mathrm{src}}) \geq 0.25$. We close the resulting mask with a 9-pixel elliptical element, retain its largest 8-connected component, and fill its holes. Boundary LPIPS is the mean spatial LPIPS within eight pixels of the resulting boundary on either side, obtained using an exact Euclidean distance transform.

Results. Depth Pro 2-H$^+$ performs best across all five metrics. Relative to Depth Anything 3-NG, the strongest baseline, it improves PSNR by 0.32 dB and reduces global and Boundary LPIPS by 4.1% and 3.9%, respectively. Relative to Depth Pro, it improves PSNR by 0.52 dB and reduces the two LPIPS measures by 6.1% and 9.1%. These results indicate that the metric depth and camera calibration produced by Depth Pro 2 provide more effective guidance near defocus boundaries. This single-image benchmark does not assess temporal consistency.

Video. We additionally apply the frozen renderer frame by frame to a 197-frame handheld clip from Pexels (2026). Figure 8 compares Depth Pro 2 with Depth Anything 3-NG (ran at 1540 px resolution), qualitatively showing that Depth Pro 2 preserves sharp structures consistently over time.

D.2 Novel-view synthesis

Protocol. We use the official Tiled Multiplane Image renderer (Khan et al., 2023) with its released weights and the 4 planes setup. We convert our method's prediction to disparity map and feed into the renderer. We clamp the disparity to $[0.01, 100]$, then min-max normalize it to $[0, 1]$ per image, As we are mainly showcasing our model's utility in high-resolution novel view synthesis, we selected images from DIS5K (Qin et al., 2022), a dataset of high-resolution photographs with thin structures that are sensitive to depth-boundary errors. When applicable, we downsize each input image with bilinear interpolation to at most 4096 px on the long side, and skip images whose native long side is below 2048 px. The camera camera uses fixed intrinsics ($f = 0.58W$, with principal point at the image center). We keep the input image, rendering parameters, and camera fixed across methods and only change the disparity predictor.

Qualitative comparison. As shown in Figure 9, the baselines render airplane struts, whiskers, railings, and chains as jagged, disconnected, or missing. Depth Pro 2 preserves these structures sharply.

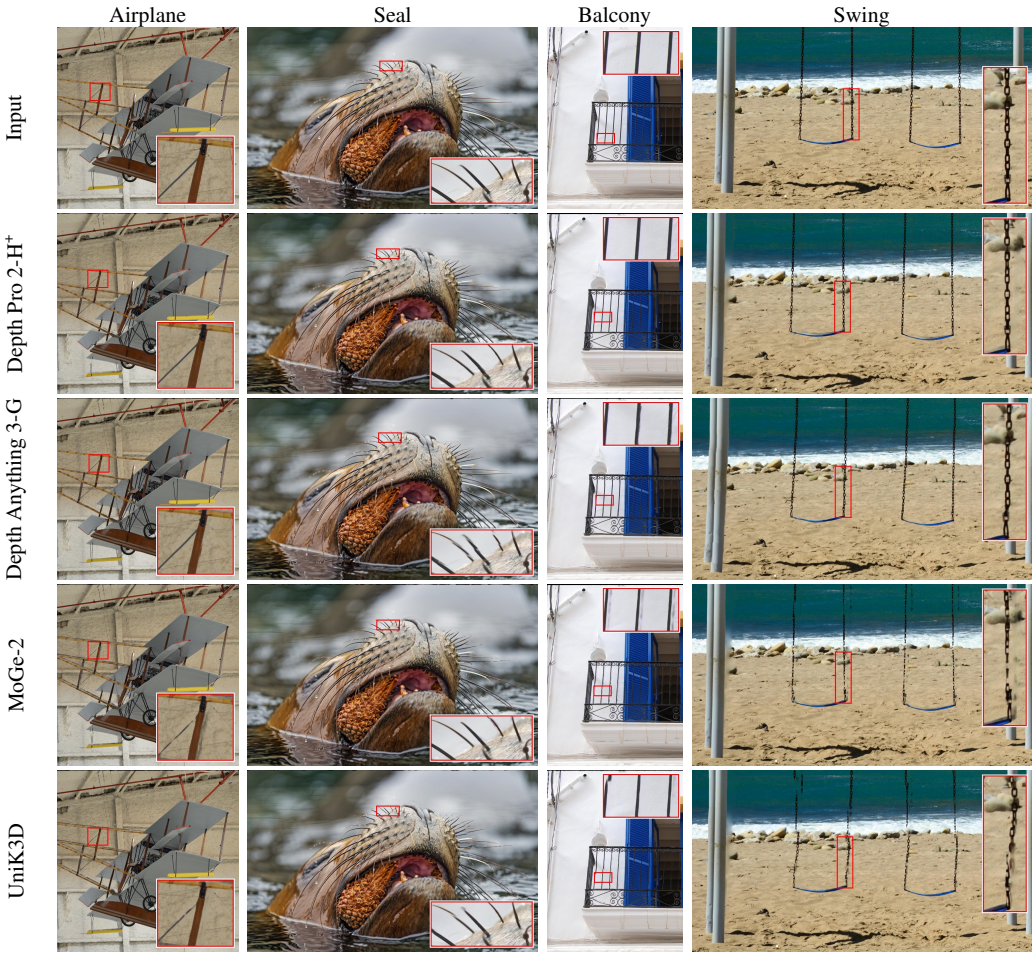

Figure 9: **High-resolution novel-view synthesis.** We show input images on top, results from Depth Pro 2-H$^+$ (2nd row), and results by representative baselines: Depth Anything 3-G (Lin et al., 2026), MoGe-2 (Wang et al., 2025c), and UniK3D (Piccinelli et al., 2025a). Depth is estimated for four high-resolution scenes from DIS5K (Qin et al., 2022), and every method's novel view is rendered with the same Tiled Multiplane Image renderer (Khan et al., 2023), at up to 4096 px on the long side. Depth Anything 3, MoGe-2, and UniK3D render thin structures jagged or broken. Depth Pro 2-H$^+$ renders sharp details.