WideDepth: Millimeter-Accurate Benchmark for Fisheye Depth Estimation

Ilia Indyk[1], Ignat Penshin[1], Ivan Sosin[1], Maxim Monastyrny[1], Aleksei Valenkov[1] and Ilya Makarov[2]

Abstract— **Fisheye cameras are increasingly adopted in robotics for near-field manipulation, navigation, and immersive perception, yet indoor depth benchmarks with accurate ground truth are still missing. To address this, we introduce WideDepth — the first indoor dataset for fisheye depth estimation, featuring 101 scenes containing 5K high-resolution stereo pairs labeled with millimeter-level ground truth depth and disparity. Our dataset also includes paired pinhole and fisheye samples across varying fields of view and baselines in both horizontal and vertical stereo setups. We further propose a method to adapt pinhole-trained stereo models to fisheye images and introduce a novel stereo fisheye image generation pipeline based on high-resolution LiDAR scans. Leveraging these methods, we thoroughly evaluate state-of-the-art monocular depth, stereo matching, and depth completion models on our benchmark. Additionally, we provide 18K LiDAR-derived sparse depth training samples, achieving up to a 62% performance boost on fisheye data when fine-tuning pinhole-based stereo models. In summary, the high precision and versatility of our benchmark set a strong foundation for advancing research in fisheye depth estimation and robotics perception. We will publicly release the datasets and code upon paper acceptance.**

I. Introduction

Depth estimation is crucial in autonomous driving, augmented reality (AR), and robotics. Common methods include monocular depth from RGB images, stereo matching, and LiDAR depth completion. However, wide field of view (FOV) depth estimation remains underexplored.

Wide FOV is important for robotics in tight spaces and autonomous vehicles needing 360-degree awareness. Multiple pinhole cameras lead to synchronization and computational issues, while fisheye cameras offer better coverage but pose challenges due to nonlinear geometry and information loss from rectification.

Research on wide FOV and distortion effects remains limited, especially for indoor fisheye depth estimation. Most existing datasets are synthetic or outdoor, lacking real indoor data. To bridge this gap, we introduce a benchmark based on high-precision laser scanning, providing dense point clouds and generated fisheye and pinhole images. As shown in Table I, our benchmark is not only the first in the indoor domain, but also surpasses existing fisheye datasets across multiple parameters.

Notably, stereo matching in the fisheye domain differs significantly from traditional methods. Existing pinhole-based approaches cannot be directly applied, as disparity is interpreted differently in fisheye geometry. However, since usually the primary goal is to obtain metric depth rather than disparity, we developed a method to convert fisheye disparity values to metric depth and vice versa.

The main contributions of our paper are:

1) We present the first indoor, millimeter-precise fisheye depth benchmark comprising 5K samples across 101 scenes with a wide range of camera parameters. Additionally, we evaluate the performance of state-of-the-art pretrained models on this dataset.
2) We propose a CUDA-accelerated pipeline that generates stereo fisheye RGB, depth, and disparity ground truth from high-resolution LiDAR scans using the Double Sphere model. This enables scalable benchmark creation without storing large panoramas and allows adapting pinhole-trained models to fisheye without retraining.
3) We compile a training dataset of 18K outdoor stereo fisheye pairs with LiDAR depth and demonstrate that additional domain adaptation through fine-tuning of pinhole-based models can further enhance metrics.

II. Related Work

A. Depth Estimation Datasets

NYU-Depth V2 [1] is widely used for indoor depth estimation and instance segmentation, aiding mobile robotics. Captured with Microsoft Kinect, its depth maps have gaps due to structured light limitations. SUN RGB-D [2] offers similar data with annotated 2D polygons and 3D cuboids for 10,000 images.

Matterport3D [3] provides 10,800 panoramic views from 194,400 RGB-D images of 90 large-scale indoor scenes, with camera poses, depth data, and semantic segmentations.

For robotics, IRS [4] focuses on disparity and surface normal estimation across 103,316 diverse indoor samples. Middlebury [5], despite having a small number of samples, remains a relevant stereo dataset. The Booster [6] dataset is also noteworthy, designed for evaluating stereo matching with a focus on challenging glass and mirror surfaces, providing high-resolution pairs and precise ground truth.

B. Fisheye Datasets

High precision is essential for indoor environments, where manipulation tasks demand sub-centimeter accuracy. Collecting depth data with such precision is challenging, especially for near-field perception, making simulation data a promising alternative. For example, SynWoodScape [7] and OmniScape

[1] Ilia Indyk, Ignat Penshin, Ivan Sosin, Maxim Monastyrny and Aleksei Valenkov are with Robotics Center, Moscow, Russia

[2] Ilya Makarov is with the AIRI, Moscow, Russia

TABLE I

OUR BENCHMARK OFFERS UNPARALLELED PRECISION OF DEPTH MAPS, LEVERAGING VARYING FOV AND HIGH-RESOLUTION IMAGES, SURPASSING ALL EXISTING FISHEYE DATASETS. OUR TRAINING DATASET IS THE FIRST-EVER WITH A VERTICAL STEREO SETUP.

Parameter	OmniScape	SynWoodScape	Oxford RobotCar	KITTI-360	WoodScape	WideDepth train (ours)	WideDepth benchmark (ours)
Real/Synthetic	Synthetic	Synthetic	Real	Real	Real	Real	Real
Domain	Outdoor	Outdoor	Outdoor	Outdoor	Outdoor	Outdoor	Indoor
Fisheye resolution	1024×1024	1280×966	1024×1024	1400×1400	1280×966	1920×1080	2048×1152
Fisheye HFOV	185	190	180	180	190	180	120..195
Precision@10m	-	-	±30 mm	±20 mm	±20 mm	±20 mm	±1 mm
Horizontal Stereo	✓	✗	✗	✗	✗	✗	✓
Vertical Stereo	✗	✗	✗	✗	✗	✓	✓
Pinhole	✗	✗	✓	✓	✗	✗	✓
Dense depth map	✓	✓	✗	✗	✗	✗	✓

[8] datasets use simulators, but their scenes are limited to outdoor settings. Additionally, the sim-to-real gap complicates accurate performance evaluation on synthetic data.

The Oxford RobotCar dataset [9] provides extensive fisheye camera data from vehicles, but without depth information. KITTI-360 [10] includes fish-eye and pinhole images for outdoor scenes, but its sparse LiDAR depth data only partially covers the wide field of view, limiting its depth estimation utility.

WoodScape [11], inspired by Robert Wood's 1906 fisheye camera, was among the first to focus on fisheye imagery for tasks like semantic segmentation, object detection, and motion segmentation. However, its lack of depth data remains a significant drawback.

C. Fisheye Projection Models

Various projection models have been developed to handle depth estimation in super wide FOV applications. The **equirectangular** projection represents 360° spherical data as a 2D image, mapping latitude and longitude to vertical and horizontal axes. It maintains epipolar line consistency along the vertical axis, offers a continuous full-scene view, and features a regular grid structure, making it well-suited for stereo depth estimation with minimal preprocessing.

The **cubemap** projection divides the scene into six faces, preserving local detail but introducing discontinuities at face boundaries, which can complicate stereo matching. A notable example of cubemap-based depth estimation is OmniVidar [12]. The **Cassini** projection was presented in the MODE paper [13] for depth estimation in horizontal panoramas, maps spherical epipolar lines to sinusoidal curves, simplifying epipolar constraints in such cases. While these projections provide structured representations, their geometric transformations impact depth consistency and processing complexity.

Other projections, such as **fisheye** and **pinhole**, present significant limitations. The **fisheye** projection captures an ultra-wide FOV but introduces radial distortion, curving epipolar lines and complicating stereo matching. The **pinhole** model, effective for narrow FOVs, lacks the coverage necessary for wide-angle depth estimation. All the considered projections were tested on the proposed benchmark as shown in Fig. 1.

III. METHODS

This chapter proposes methods to generate a variety of benchmark data, including different image projections as well as FOV and image distortion. Since the original benchmark data is a set of high-quality colored point clouds, in addition to the images, Depth2Disparity and Disparity2Depth conversions are proposed for equirectangular projection. Equirectangular and cubemap projections were compared for depth estimation on super-wide FOV.

A. Projection and Warping with the DS-Model

The Double Sphere (DS) camera model proposed by Usenko *et al.* [14] is well-suited for cameras with fisheye lenses, offering a closed-form inverse and eliminating the need for computationally expensive trigonometric operations. In the DS model, a 3D point is initially projected onto two unit spheres with their centers offset by ξ. The point is then mapped onto the image plane using a pinhole model adjusted by $\alpha/(1-\alpha)$. The model is defined by the parameter set $\mathbf{i} = [f_x, f_y, c_x, c_y, \xi, \alpha]^T$.

To warp fisheye images into pinhole or equirectangular views, we first simulate a 3D grid for the target projection and then reproject the points onto the image plane using the Double Sphere model. Camera intrinsics and extrinsics are estimated with the Kalibr toolbox [14], [15].

For the set of virtual cameras in the benchmark, the calibrated camera parameters were used as a reference. These cameras are integrated into SensorBox, which is detailed in Sec. IV-B. To capture a diverse set of data, including camera distortion, the intrinsic parameters of the virtual cameras, $\mathbf{i_{virt}}$, were made functionally dependent on the FOV:

$$f_{x,y}^{virt}(\text{FOV}) = f_{x,y} \cdot \left(n \cdot \frac{180^\circ}{\text{FOV}} \right), \tag{1}$$

$$\xi^{virt}(\text{FOV}) = \xi \cdot \left(1 - m \cdot \frac{\text{FOV}}{180^\circ} \right), \tag{2}$$

$$\alpha^{virt}(\text{FOV}) = \alpha + m \cdot (1-\alpha) \cdot \frac{\text{FOV}}{180^\circ}, \tag{3}$$

The scaling factors $m = 0.2$ and $n = 1.25$ were empirically determined to ensure the preservation of the maximum available density of LiDAR 3D points while maintaining

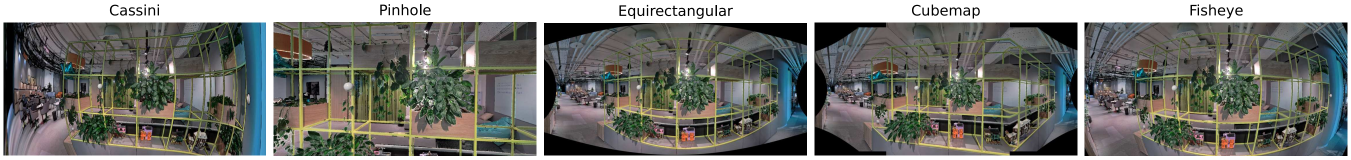

Fig. 1. Benchmark scene presented in the considered projections: Cassini, Pinhole, Equirectangular, Cubemap, Fisheye.

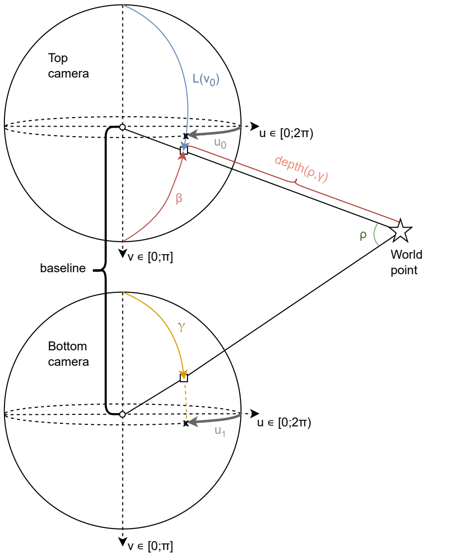

Fig. 2. Intuition for the proposed Depth-Disparity Conversion formulations for a vertical stereo pair in spherical projection.

image quality. Specifically, these values were selected to mitigate the occurrence of artifacts, such as blank regions in the projected image. The parameter $f_{x,y}^{virt}$ directly influences the final scale of the projected image. The parameters ξ^{virt} and α^{virt} exhibit a functional dependence on the FOV to explicitly model the increasing impact of distortion as FOV expands.

B. Disparity-Depth Conversions

Stereo Depth Estimation can be performed on a set of virtual vertical stereo pairs in equirectangular projection warped from fisheye. The problem is that Stereo Depth Estimation outputs a disparity map, while initial lidar data can only be naively converted into a depth map. To evaluate the quality of the predicted disparity, we implemented the Disparity2Depth and Depth2Disparity Conversion methods for vertical stereo pairs in equirectangular projection. The geometric intuition for both methods is shown in Fig. 2.

Disparity2Depth Conversion on equirectangular projection is based on spherical geometry. Given the disparity map from a stereo pair, the depth for each pixel is calculated by the following steps.

1. Latitude Calculation: The latitude L of each pixel is determined from its vertical coordinate v in the image:

$$L(v) = \frac{v}{H} \cdot \pi, \tag{4}$$

where $L \in [0; \pi]$, H is the height of the image.

2. Disparity Angle: The disparity angle ρ for each pixel $\mathbf{u}$ is derived from its disparity value $disp_{\mathbf{u}}$. $\rho \in [0; \pi]$ corresponds to the vertex of a 3D point in a spherical triangle:

$$\rho(\mathbf{u}) = \frac{disp_{\mathbf{u}}}{H} \cdot \pi \tag{5}$$

3. Angles β, γ: The angle β is calculated based on the latitude L. The angle γ is dependent on ρ and β in a spherical triangle:

$$\beta = \pi - L \tag{6}$$

$$\gamma = \pi - \rho - \beta \tag{7}$$

4. Depth Calculation: Finally, the $depth(\rho, \gamma)$ for each pixel is determined using the law of sines:

$$depth(\rho, \gamma) = \frac{B \cdot \sin(\gamma)}{\sin(\rho)} \tag{8}$$

where B is a constant factor depending on the stereo camera baseline and setup.

This method ensures an efficient conversion from disparity to depth on equirectangular projections by leveraging the geometry of spherical images. It was validated on Helvipad [16], an omnidirectional dataset with sparse GT depth. While Helvipad is not a fisheye dataset, it is still suitable to evaluate our conversion method. Using CREStereo [17] with our approach, we achieved decent results (MAE = 1.92, RMSE = 3.40), comparable to those reported in the original paper. This demonstrates the robustness of our method, even in challenging scenarios.

Depth2Disparity Conversion on equirectangular projection. The method is predicated on the sine theorem for the aforementioned in Fig. 2 triangle. The $depth(\rho, \gamma)$ from Eq. (8) is a known value for this method. It will be written $depth_{\mathbf{u}}$. In the following section the transformation to obtain the $disp_{\mathbf{u}}$ value for each pixel $\mathbf{u}$ is presented:

$$disp_{\mathbf{u}} = \left(\frac{H}{\pi}\right) \cdot \arctan\left(\frac{sin(L_{\mathbf{u}})}{\frac{depth_{\mathbf{u}}}{B} + cos(L_{\mathbf{u}})}\right) \tag{9}$$

This method provides an efficient way to convert depth maps to disparity maps for vertical stereo pairs in equirectangular projection, using the spherical coordinate system.

Computational efficiency. Our implementation of both the Disparity2Depth and Depth2Disparity methods is optimized for computational efficiency through CUDA acceleration, utilizing matrix operations to maximize parallel processing performance. This design enables real-time or near-real-time execution, making it well-suited for high-resolution depth estimation tasks.

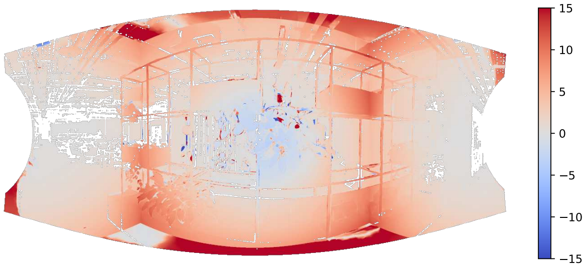

Fig. 3. Subtraction of absolute cubemap and equirectangular disparity errors. The prevalence of positive values (red) indicates a larger cubemap error, which grows up at the boundaries.

TABLE II
COMPARISON OF PROJECTION METHODS ON OUR WIDEDEPTH BENCHMARK. FOR ALL METRICS THE LOWER THE BETTER.

Projection	EPE (px)	Q^{50}_{EPE}(px)	Q^{95}_{EPE}(px)	bad-1 (%)	bad-2 (%)	bad-3 (%)
Cubemap	4.500	3.111	13.743	78.93	62.92	51.13
Equirectangular	1.065	0.351	3.056	28.27	10.37	5.16

C. Projection Model Choice

Equirectangular and **cubemap** projections were discussed in Sec. II-C. These projections offer distinct advantages and challenges in the context of wide FOV depth estimation.

To quantitatively compare these projections, we computed the disparity errors for both models using the reference disparity from our benchmark dataset as GT. Both projections are processed using the same stereo matching model, CREStereo [17], ensuring that observed differences arise solely from the projection method itself. The difference in absolute errors is visualized in Fig. 3.

A numerical evaluation of disparity errors is provided in Table II. The equirectangular projection consistently outperforms the cubemap representation across all error metrics, demonstrating lower End-Point Error (EPE) and reduced bad-pixel percentages.

Overall, while the cubemap projection retains local detail, it suffers from depth consistency issues and increased processing complexity due to face separation and subsequent stitching. The equirectangular projection remains a more structured and effective representation for depth estimation in wide FOV scenarios. Due to these advantages, we use this projection in our experiments.

IV. OUR PROPOSED DATASETS

We introduce two datasets: benchmark and training set. The benchmark, our primary contribution, is smaller but features diverse camera parameters and highly precise GT depth maps. In contrast, our training dataset offers sparse depth maps but includes more samples, enabling us to assess the effect of domain adaptation through fine-tuning.

In this chapter, we detail the requirements and generation processes of both datasets.

A. Benchmark Design

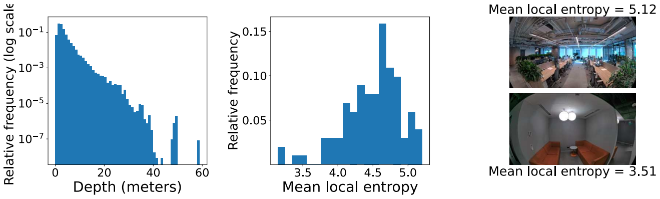

Fig. 4. **Left:** Depth distribution in our benchmark follows typical indoor patterns with a long right tail due to corridors. **Center:** The distribution of mean local entropy is skewed toward high values, reflecting the abundance of fine structural details and texture complexity. **Right:** Example of samples with high (top) and low (bottom) entropy; high-entropy images contain small details like plant leaves and ceiling-mounted communication elements.

To evaluate model robustness, a benchmark should include diverse scenes and conditions. Unlike autonomous driving datasets with high-frame-rate vehicle-mounted cameras producing near-duplicate samples, we curate scenes based on fine details, geometric complexity, lighting, and depth estimation use cases. For manipulation, isolated objects are key, while navigation benefits from corridors, offices, and cluttered environments (e.g., kitchens). Captured indoors, our benchmark includes the following lighting categories: natural light, office lamps, and mixed conditions. To enhance diversity, we varied capture heights: 1.65m for AR/humanoid robots, 2.5m for CCTV, and 0.5m for mobile platforms. Higher viewpoints emphasize large planes, while lower ones capture finer details like chair legs and wires.

The key idea is to project a dense, colored point cloud into 2D (see details in Sec. IV-C), enabling the use of virtual cameras with flexible configurations. For stereo pairs, we utilize a range of baselines tailored to specific applications: from near-field tasks like robotic arm cameras (20 mm) and consumer stereo cameras such as ZED X (65 mm and 120 mm) to setups with extremely large baselines (200 mm and 300 mm). For fisheye cameras, we selected horizontal FOVs of 120, 140, 165, and 195 degrees to evaluate model performance as the angle gradually increases. The combination of five baselines and four angles results in 20 vertical and 20 horizontal stereo pairs per scene. For pinhole cameras, the FOV is fixed at 90 degrees, generating 5 vertical and 5 horizontal pairs per scene based on baseline variations.

Figure 4 presents key dataset statistics. The depth range histogram shows a long right tail, which can challenge near-range indoor models. Overall, the primary depth bins are: 0–1 m (6.9%), 2–5 m (74.7%), 5–10 m (9.6%), and 10+ m (1.7%), aligning with typical indoor datasets. To evaluate scene complexity, we calculated local entropy over 11 neighboring pixels. The majority of samples in our benchmark exhibit high mean local entropy, indicating diverse objects and rich edge details, which pose a greater challenge for depth estimation methods.

We generate stereo fisheye pairs directly from merged LiDAR scans, avoiding the need for panoramic intermediates. With the Double Sphere camera model and a CUDA-accelerated pipeline, our method efficiently synthesizes fisheye images with arbitrary baselines and focal lengths, preserving realistic occlusions and geometric fidelity. This design provides broad coverage of configurations and scalable image generation for diverse benchmarking scenarios.

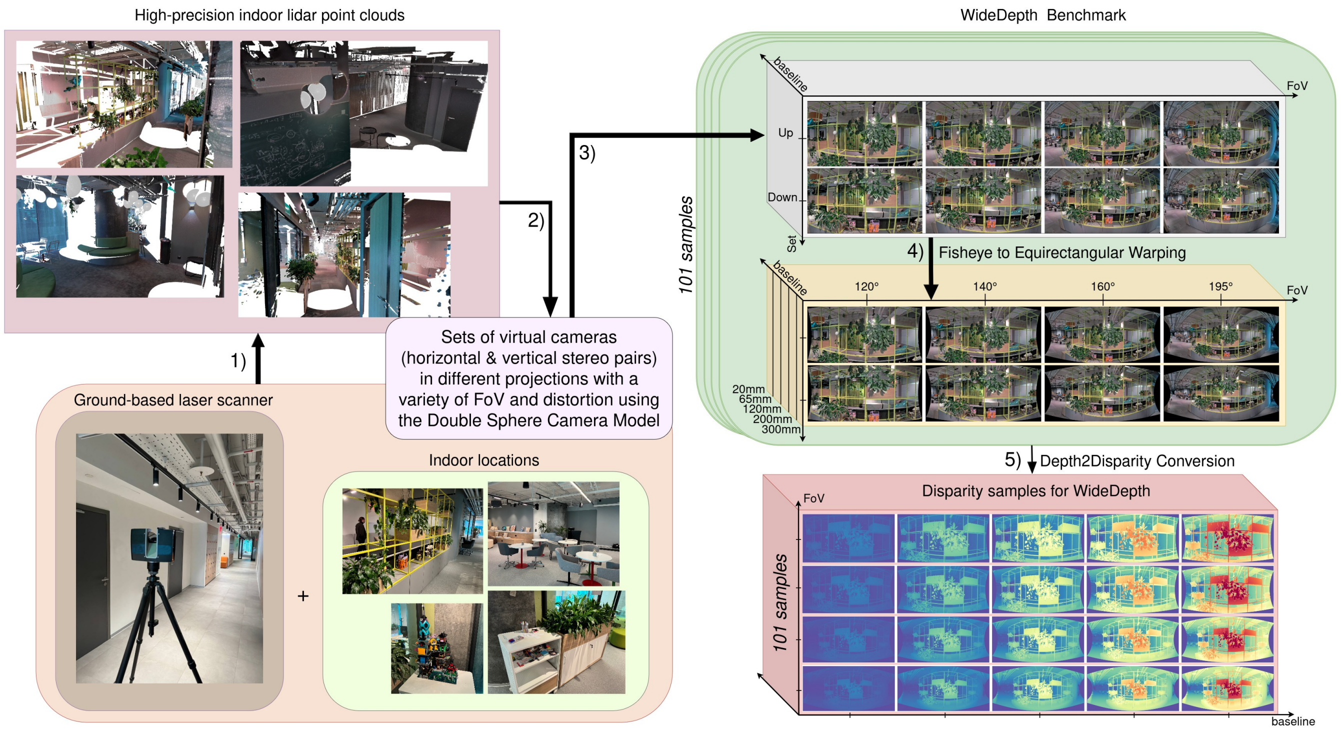
High-precision indoor lidar point clouds
WideDepth Benchmark
1)
2)
3)
Sets of virtual cameras (horizontal & vertical stereo pairs) in different projections with a variety of FoV and distortion using the Double Sphere Camera Model
Ground-based laser scanner
+
Indoor locations
101 samples
baseline
FoV
Up
Down
Set
4) Fisheye to Equirectangular Warping
baseline
120°
140°
160°
195°
FoV
20mm
65mm
120mm
200mm
300mm
5) Depth2Disparity Conversion
Disparity samples for WideDepth
FoV
101 samples
baseline

Fig. 5. WideDepth: A high-resolution benchmark for fisheye depth estimation in indoor environments. From lidar-generated point clouds (1), we create virtual fisheye cameras with diverse FOVs, distortions, and origins using the Double Sphere Model (2). Cameras capture multiple perspectives and baselines (3), with each fisheye view warped to equirectangular projection to maintain epipolar constraints (4). Finally, depth data is converted to disparity for stereo model evaluation (5). WideDepth includes 101 samples spanning FOVs from 120° to 195°, pushing the limits of mono and stereo depth estimation with superwide-FOV cameras.

B. Hardware

To obtain a high-precision colored point cloud for benchmark, we used a ground-based laser scanner with a range error of 1-2 mm and a resolution up to 165 MP. Each scan generates about 20 million points over a 360° view but requires 10 minutes per scan, limiting large-scale data collection. For train dataset we developed the portable SensorBox, integrating pair of ZED X One fisheye cameras (30 Hz) in vertical stereo setup and two Livox Mid 360 LiDARs (10 Hz) positioned such that their fields of view partially overlap, maximizing coverage of the scene. Further details, including LiDAR specifications and SensorBox setup, can be found in the accompanying video clip.

C. Datasets Acquisition

Benchmark. We performed laser scanning in the office space. The distance between scanning points varied from 1.5 to 5 meters, with closer spacing in rooms containing furniture or small objects and wider spacing in long corridors. Spherical markers, positioned around the scanner, facilitated alignment of neighboring point clouds. To generate 2D depth maps, point clouds were sampled and projected based on the virtual camera's direction, baseline, and FOV. This pipeline can be seen on the Fig. 5. Stereo pairs were created by shifting the camera parallel to the scan center, which can introduce occlusions or empty regions, especially with wider baselines. To address this, we used a bundle of three overlapping scans, allowing occlusions in the *central* scan to be filled with points from *adjacent* scans. This method was applied to all but the first and last scans, resulting in a total of 101 scenes. Windows and mirrors are handled through semi-manual point cloud cleaning. To maintain indoor depth distribution, we clip the depth of outdoor objects visible through windows to a closer value. Also, reflective surfaces are masked with zero depth values.

Train dataset. To prevent overfitting to benchmark indoor environment, our train dataset was captured entirely outdoors, handheld, across city areas, parks, and streets during daylight under cloudy conditions. Stereo pairs were rectified and warped to an equirectangular projection as described in Sec. III-A. Point clouds from two hardware-synced LiDARs were merged into a single cloud and projected onto the upper camera frame using the DS camera model. For the training set, we prioritized dataset size over GT label precision and density as a trade-off. Synchronization between LiDARs and the camera was achieved through timestamp alignment within the ROS framework. In addition to *depth* GT, we generated *disparity* GT using our proposed Depth2Disparity conversion method described in Sec. III-B. After projecting the point clouds into the camera frame, we observed a mean depth point density of 8.1% across the dataset.

V. Experiments

In this chapter, we evaluate SOTA models on our benchmark, analyzing how increasing FOV impacts monocular

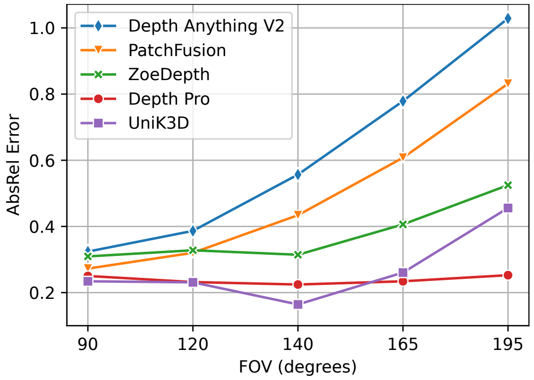

Fig. 6. All observed monocular depth models perform well on pinhole images, but most degrade with a wider field of view. This deterioration at higher FOV highlights the unresolved performance issue, making our benchmark highly valuable.

depth and depth completion models. We expect monocular models to degrade more noticeably than depth completion models, which rely on depth guidance. We also evaluate stereo models across varying FOV-baseline combinations and fine-tune a pretrained stereo model on our fisheye dataset to assess domain adaptation effects.

A. Monocular and Depth Completion

Monocular Depth. For comparison, we selected SOTA off-the-shelf models that offer acceptable inference times and memory consumption for practical applications, with an additional key criterion being their ability to produce metric depth. Based on these requirements, we chose Depth Anything V2-Large [18], PatchFusion [19], ZoeDepth [20], Depth Pro [21] and UniK3D-Large [22] models.

Following the standard depth estimation metrics summarized in [23], we analyzed how quality changes as the FOV expanded from 120 to 195 degrees, the narrowest to widest fisheye in our dataset. Table III shows that all monocular models degrade at wider FOVs, but to varying extents. We further examined model performance at intermediate FOV values. As depicted in Fig. 6, nearly all models exhibit gradual performance degradation as FOV increases. Since UniK3D was explicitly designed to be camera-universal, it achieves its highest accuracy at moderate fisheye ranges (120–140°) and surpasses other tested monocular models in this regime. This makes it the most reliable choice for applications targeting wide yet not extreme FOVs. However, even this model shows decline at extreme FoVs ($\geq 165^\circ$) when operated in camera-free mode without intrinsics.

This supports our initial hypothesis that wider angles make it more challenging for models to achieve metric depth due to domain shift and distorted geometry.

Depth Completion. Among state-of-the-art models for depth completion, we selected two larger models, NLSPN [24] and CompletionFormer [25], as well as a lighter model, CostDCNet [26], which offers high quality with low latency. All three models were trained on the same NYUv2 [1] dataset. As shown in Table III, although all models experience a decrease in quality according to the AbsRel metric, certain metrics actually improve in the larger models. This may be attributed to complex models — those incorporating mechanisms like attention and higher capacity — being better equipped to interpret scenes at wide viewing angles.

TABLE III

PERCENTAGE CHANGE IN METRICS FOR FOV 195 COMPARED TO FOV 120. GREEN INDICATES IMPROVED RESULTS WITH A WIDER ANGLE, WHILE RED - DECLINE IN PERFORMANCE AT A WIDER ANGLE.

Model	AbsRel ↓	MAE ↓	RMSE ↓	$\delta_{1.25}$ ↑	$\delta_{1.25^2}$ ↑	$\delta_{1.25^3}$ ↑
NLSPN [24]	+5.92%	-1.26%	-3.05%	-0.03%	+0.04%	+0.04%
CompletionFormer [25]	+6.73%	-2.05%	+1.97%	-0.01%	+0.05%	+0.04%
CostDCNet [26]	+38.5%	-13.9%	+0.67%	+0.06%	-0.07%	-0.16%
DepthAnythingV2 [18]	+166%	+126%	+106%	-86.8%	-81.5%	-53.9%
PatchFusion [19]	+160%	+123%	+91.3%	-87.4%	-68%	-27.9%
UniK3D [22]	+97.3%	+147.2%	+119.6%	-73.7%	-56.2%	-38.9%
ZoeDepth [20]	+60.1%	+47.3%	+21.3%	-51.7%	-25%	-6.5%
DepthPro [21]	+9.0%	+3.3%	+8.3%	-4.1%	-4.5%	-2.31%

In summary, even robust, large-scale models like Depth Anything V2-Large degrade in performance on fisheye images as FOV increases. As expected, depth completion models are less affected by wide angles but still show significant declines in the AbsRel metric, one of the most important indicators, even among high-performing SOTA models with attention mechanisms.

B. Stereo Matching and Fine-tuning

Stereo models vary widely in size, so we selected models from different weight categories, including FADNet++ [27], CREStereo [17], BGNet [28], IGEV [29], GMStereo [30], and StereoBase [31]. All models received input data processed as described in Sec. III, including equirectangular projection, cropping of empty areas, and a 90-degree counter-clockwise rotation. Evaluation followed the commonly adopted stereo metrics summarized in [32]. We also measured each model's latency in half precision on an NVIDIA GeForce RTX 3080 Ti at 1024 × 512 resolution.

Quantitative results in Table IV show that model rankings closely match pinhole benchmarks, suggesting that fisheye stereo performance can be inferred from pinhole-based ratings. For wide-angle applications, StereoBase proves ideal, though at the cost of high latency. In Fig. 7 this model demonstrates consistent performance with our method, even in challenging cases. The relative error map highlights common stereo matching errors (e.g., on translucent and reflective surfaces) without fisheye-specific issues, showcasing the robustness of our adaptation approach.

We also analyzed the impact of baseline and FOV on StereoBase, finding that baseline variations affect performance more significantly than FOV. A 65 mm baseline yields optimal metrics, while larger baselines degrade quality due to disparity distribution shift. Conversely, higher FOV consistently enhances performance, reinforcing the importance of wide angles for capturing broader indoor environments.

Fine-tuning. We selected BGNet, a lightweight model optimized for real-time inference on embedded devices (e.g., Jetson Orin), making it well-suited for practical use. A SceneFlow-pretrained model was fine-tuned on our WideDepth training set and evaluated on the WideDepth benchmark under the same conditions as other pretrained models.

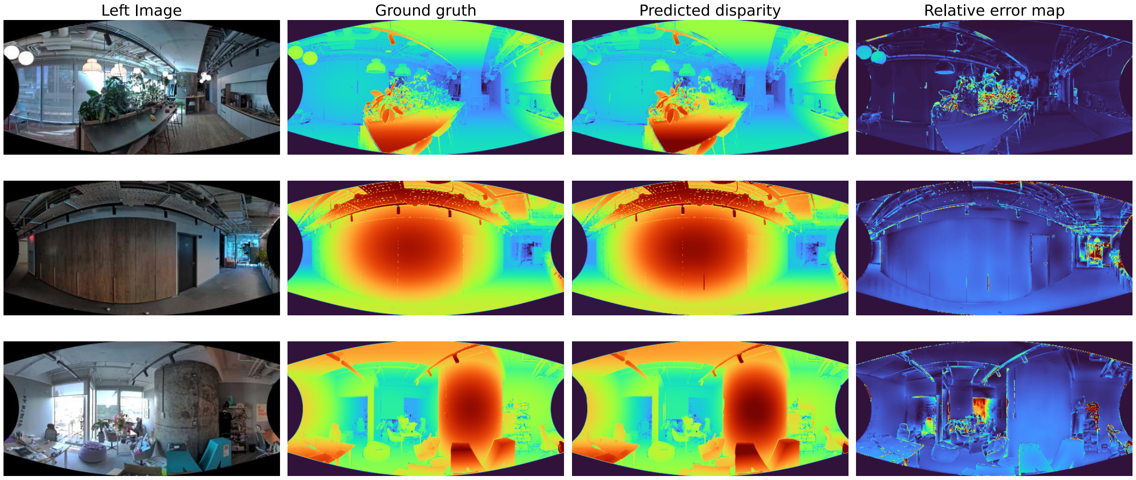

Fig. 7. Qualitative results for stereo with FOV 195 using the StereoBase model. The model shows no degradation from geometric distortion, demonstrating the success of our approach in adapting pinhole models to fisheye data.

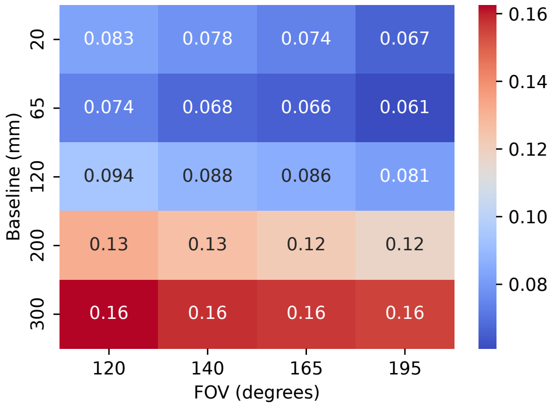

Fig. 8. Impact of baseline and FOV variations on RelEPE (lower is better) using StereoBase model. Results indicate much greater sensitivity to unfamiliar baselines than high FOVs, demonstrating our approach's effectiveness for fisheye stereo.

TABLE IV

STEREO MODEL PERFORMANCE AT 195° FOV AND 65 MM BASELINE; LOWER VALUES ARE BETTER. FINE-TUNING LIGHTWEIGHT BGNET ON OUR TRAIN DATASET BOOSTS IT TO MATCH HEAVIER SOTA MODELS, SHOWING THE VALUE OF DOMAIN ADAPTATION

Model	Latency (ms)	EPE (px)	Q^{50}_{EPE} (px)	Q^{95}_{EPE} (px)	bad-1 (%)	bad-2 (%)	bad-3 (%)
FADNet++ [27]	40	2.089	0.643	6.373	29.63	15.59	10.46
GMStereo [30]	186	1.950	0.700	6.495	30.06	15.16	10.31
IGEV [29]	552	**1.605**	0.533	**4.526**	24.87	12.58	**8.37**
StereoBase [31]	665	1.612	0.529	4.554	**24.75**	**12.49**	**8.37**
CREStereo [17]	284	1.759	**0.528**	6.238	24.78	13.02	9.02
BGNet [28]	19	3.411	1.410	13.595	53.30	33.32	24.34
BGNet [28] (fine-tuned)	19	1.767 (-48%)	0.606 (-57%)	5.736 (-58%)	27.63 (-48%)	13.76 (-59%)	9.29 (-62%)

Fine-tuning was conducted for 15 epochs with batch size 8, half precision, and the AdaBelief [33] optimizer. A one-cycle scheduler [34] adjusted the learning rate, peaking at $5 \cdot 10^{-5}$ with a warmup over 10% of iterations. To increase data diversity, we applied asymmetric chromatic augmentation with 50% probability.

Table IV shows substantial performance gains after fine-tuning on our dataset, matching or surpassing heavier models. This underscores that while robust fisheye performance can be achieved with pinhole-trained models, as we proposed, dedicated domain adaptation to fisheye data can further improves results.

VI. Limitations and Future work

LiDAR scanning struggles to accurately capture transparent and reflective surfaces, so we masked these areas in our benchmark. However, these challenging cases present valuable research opportunities, and we aim to address depth estimation in such conditions in future work. In addition, we plan to expand our benchmark beyond depth estimation by incorporating other vision tasks, such as semantic segmentation, to enhance its applicability. Regarding the experiments, our GPU constraints restricted fine-tuning to lightweight architectures, leaving the effect of domain adaptation on larger models with distorted geometry uncertain.

VII. Conclusion

In this article, we introduced WideDepth, the first indoor fisheye depth benchmark with 101 scenes and over 5K stereo pairs featuring diverse fields of view, baselines, and high-precision depth and disparity maps. Additionally, we provided an outdoor stereo fisheye dataset containing 18K LiDAR-labeled samples, which has demonstrated strong effectiveness for model adaptation to fisheye geometry.

Since existing disparity-to-depth methods do not directly apply to fisheye stereo, we proposed a conversion method tailored for fisheye cases. To leverage a broad range of existing stereo models, we introduced an approach that enables pinhole-trained models to be used on fisheye images without architectural changes. Finally, we evaluated 14 SOTA models across various tasks on our benchmark, including monocular depth estimation, stereo matching, and depth completion.

We believe WideDepth will significantly advance fisheye depth estimation research, bridging the gap with pinhole-based methods.

References

[1] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *ECCV*, 2012.

[2] S. Song, S. P. Lichtenberg, and J. Xiao, "SUN RGB-D: A RGB-D scene understanding benchmark suite," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 567–576. [Online]. Available: https://api.semanticscholar.org/CorpusID:6242669

[3] A. X. Chang, A. Dai, T. A. Funkhouser, M. Halber, M. Nießner, M. Savva, S. Song, A. Zeng, and Y. Zhang, "Matterport3D: Learning from RGB-D data in indoor environments," in *2017 International Conference on 3D Vision (3DV)*, 2017, pp. 667–676. [Online]. Available: https://api.semanticscholar.org/CorpusID:21435690

[4] Q. Wang, Z. Shizhen, Q. Yan, F. Deng, K. Zhao, and X. Chu, "IRS: A large naturalistic indoor robotics stereo dataset to train deep models for disparity and surface normal estimation," in *2021 IEEE International Conference on Multimedia and Expo (ICME)*, 2021, pp. 1–6. [Online]. Available: https://api.semanticscholar.org/CorpusID:236273594

[5] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nesic, X. Wang, and P. Westling, "High-resolution stereo datasets with subpixel-accurate ground truth," in *German Conference on Pattern Recognition*, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:14915763

[6] P. Z. Ramirez, F. Tosi, M. Poggi, S. Salti, S. Mattoccia, and L. di Stefano, "Open challenges in deep stereo: the booster dataset," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 21 136–21 146. [Online]. Available: https://api.semanticscholar.org/CorpusID:249538677

[7] A. R. Sekkat, Y. Dupuis, V. R. Kumar, H. Rashed, S. K. Yogamani, P. Vasseur, and P. Honeine, "SynWoodScape: Synthetic surround-view fisheye camera dataset for autonomous driving," *IEEE Robotics and Automation Letters*, vol. 7, pp. 8502–8509, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:247362954

[8] A. R. Sekkat, Y. Dupuis, P. Vasseur, and P. Honeine, "The OmniScape dataset," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 1603–1608. [Online]. Available: https://api.semanticscholar.org/CorpusID:221847109

[9] W. P. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The Oxford RobotCar dataset," *The International Journal of Robotics Research*, vol. 36, pp. 15–3, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:22556995

[10] Y. Liao, J. Xie, and A. Geiger, "KITTI-360: A novel dataset and benchmarks for urban scene understanding in 2D and 3D," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, pp. 3292–3310, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:238198653

[11] S. K. Yogamani, C. Hughes, J. Horgan, G. Sistu, P. Varley, D. O'Dea, M. Uřiář, S. Milz, M. Simon, K. Amende, C. Witt, H. Rashed, S. Chennupati, S. Nayak, S. Mansoor, X. Perroton, and P. Pérez, "WoodScape: A multi-task, multi-camera fisheye dataset for autonomous driving," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 9307–9317. [Online]. Available: https://api.semanticscholar.org/CorpusID:146121106

[12] S. Xie, D. Wang, and Y.-H. Liu, "OmniVidar: omnidirectional depth estimation from multi-fisheye images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 21 529–21 538.

[13] M. Li, X. Jin, X. Hu, J. Dai, S. Du, and Y. Li, "MODE: Multi-view omnidirectional depth estimation with 360 cameras," in *European Conference on Computer Vision*. Springer, 2022, pp. 197–213.

[14] V. Usenko, N. Demmel, and D. Cremers, "The double sphere camera model," in *2018 International Conference on 3D Vision (3DV)*. IEEE, 2018, pp. 552–560.

[15] J. Rehder, J. Nikolic, T. Schneider, T. Hinzmann, and R. Siegwart, "Extending kalibr: Calibrating the extrinsics of multiple IMUs and of individual axes," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 4304–4311.

[16] M. Zayene, J. Endres, A. Havolli, C. Corbière, S. Cherkaoui, A. Kontouli, and A. Alahi, "Helvipad: A real-world dataset for omnidirectional stereo depth estimation," *arXiv preprint arXiv:2411.18335*, 2024.

[17] J. Li, P. Wang, P. Xiong, T. Cai, Z. Yan, L. Yang, J. Liu, H. Fan, and S. Liu, "Practical stereo matching via cascaded recurrent network with adaptive correlation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 16 263–16 272.

[18] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao, "Depth anything v2," *ArXiv*, vol. abs/2406.09414, 2024. [Online]. Available: https://api.semanticscholar.org/CorpusID:270440448

[19] Z. L. et al., "PatchFusion: An end-to-end tile-based framework for high-resolution monocular metric depth estimation," in *CVPR'24*, 2024, pp. 10 016–10 025. [Online]. Available: https://api.semanticscholar.org/CorpusID:265659202

[20] S. B. et al., "ZoeDepth: Zero-shot transfer by combining relative and metric depth," *ArXiv*, vol. abs/2302.12288, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:257205739

[21] A. Bochkovskii, A. Delaunoy, H. Germain, M. Santos, Y. Zhou, S. R. Richter, and V. Koltun, "Depth Pro: Sharp monocular metric depth in less than a second," 2024. [Online]. Available: https://arxiv.org/abs/2410.02073

[22] L. Piccinelli, C. Sakaridis, M. Segu, Y.-H. Yang, S. Li, W. Abbeloos, and L. Van Gool, "UniK3D: Universal camera monocular 3d estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[23] A. Masoumian, H. A. Rashwan, J. Cristiano, M. S. Asif, and D. Puig, "Monocular depth estimation using deep learning: A review," *Sensors*, vol. 22, no. 14, p. 5353, 2022, section 4 summarizes/defines AbsRel, SqRel, RMSE, RMSE(log), and δ accuracy thresholds. [Online]. Available: https://www.mdpi.com/1424-8220/22/14/5353

[24] J. Park, K. Joo, Z. Hu, C.-K. Liu, and I. S. Kweon, "Non-local spatial propagation network for depth completion," 2020. [Online]. Available: https://arxiv.org/abs/2007.10042

[25] Y. Zhang, X. Guo, M. Poggi, Z. Zhu, G. Huang, and S. Mattoccia, "CompletionFormer: Depth completion with convolutions and vision transformers," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 18 527–18 536. [Online]. Available: https://api.semanticscholar.org/CorpusID:258309598

[26] J. Kam, S. K. J. Kim, J. Park, and S. Lee, "CostDCNet: Cost volume based depth completion for a single RGB-D image," in *European Conference on Computer Vision*, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:253513199

[27] Q. Wang, S. Shi, S. gang Zheng, K. Zhao, and X. Chu, "FADNet++: Real-time and accurate disparity estimation with configurable networks," *ArXiv*, vol. abs/2110.02582, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:238407735

[28] B. Xu, Y. Xu, X. Yang, W. Jia, and Y. Guo, "Bilateral grid learning for stereo matching networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1–10.

[29] G. Xu, X. Wang, X. Ding, and X. Yang, "Iterative geometry encoding volume for stereo matching," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 21 919–21 928.

[30] H. Xu, J. Zhang, J. Cai, H. Rezatofighi, F. Yu, D. Tao, and A. Geiger, "Unifying flow, stereo and depth estimation," 2023. [Online]. Available: https://arxiv.org/abs/2211.05783

[31] X. Guo, C. Zhang, J. Lu, Y. Wang, Y. Duan, T. Yang, Z. Zhu, and L. Chen, "OpenStereo: A comprehensive benchmark for stereo matching and strong baseline," 2024. [Online]. Available: https://arxiv.org/abs/2312.00343

[32] F. Tosi, L. Bartolomei, and M. Poggi, "A survey on deep stereo matching in the twenties," *International Journal of Computer Vision*, vol. 133, pp. 4245–4276, 2025, appendix C defines EPE, RMSE, bad-τ, and KITTI D1 outlier metrics. [Online]. Available: https://link.springer.com/article/10.1007/s11263-024-02331-0

[33] J. Zhuang, T. M. Tang, Y. Ding, S. C. Tatikonda, N. C. Dvornek, X. Papademetris, and J. S. Duncan, "AdaBelief optimizer: Adapting stepsizes by the belief in observed gradients," *ArXiv*, vol. abs/2010.07468, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:222377595

[34] L. N. Smith and N. Topin, "Super-convergence: very fast training of neural networks using large learning rates," in *Defense + Commercial Sensing*, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:260552651