Dur360BEV: A Real-world 360-degree Single Camera Dataset and Benchmark for Bird-Eye View Mapping in Autonomous Driving

Wenke E[1], Chao Yuan[1], Li Li[1], Yixin Sun[1], Yona Falinie A. Gaus[1],
Amir Atapour-Abarghouei[1], Toby P. Breckon[1, 2]
Department of {[1]Computer Science, [2]Engineering}, Durham University, UK

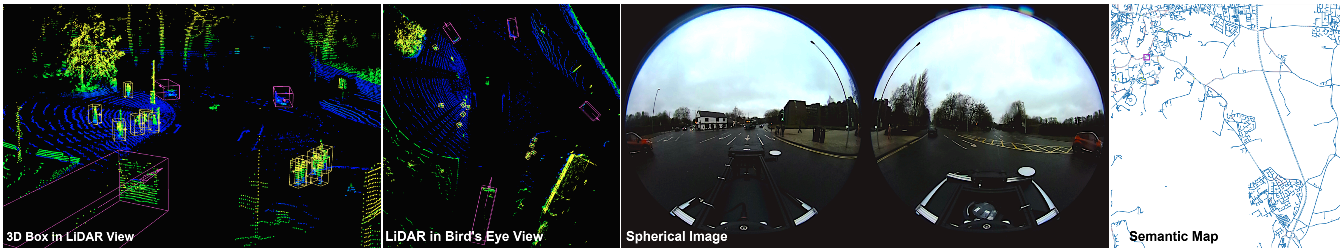

Fig. 1: An example from the Dur360BEV dataset where (from left-to-right) see 3D bounding box annotation for LiDAR, an exemplar LiDAR in Bird's Eye View (BEV), the dual-fisheye image from our spherical camera and our semantic map based on OpenStreetMap.

***Abstract*— We present Dur360BEV, a novel spherical camera autonomous driving dataset equipped with a high-resolution 128-channel 3D LiDAR and a RTK-refined GNSS/INS system, along with a benchmark architecture designed to generate Bird-Eye-View (BEV) maps using only a single spherical camera. This dataset and benchmark address the challenges of BEV generation in autonomous driving, particularly by reducing hardware complexity through the use of a single 360-degree camera instead of multiple perspective cameras. Within our benchmark architecture, we propose a novel spherical-image-to-BEV module that leverages spherical imagery and a refined sampling strategy to project features from 2D to 3D. Our approach also includes an innovative application of focal loss, specifically adapted to address the extreme class imbalance often encountered in BEV segmentation tasks, that demonstrates improved segmentation performance on the Dur360BEV dataset. The results show that our benchmark not only simplifies the sensor setup but also achieves competitive performance.**

Code + Dataset: https://github.com/Tom-E-Durham/Dur360BEV

I. INTRODUCTION

A spherical dual-fisheye camera, provides a full field of view (FoV) with dual fisheye lenses, capturing the entire environment in a single frame with just one device. This minimalist sensor setup offers a streamlined alternative to multi-camera systems. It is particularly well-suited for applications like autonomous driving, where a single spherical camera ensures full situational awareness while reducing hardware complexity, such as the need for multi-sensor calibration, synchronization, and connectivity [1], [2], [3], [4].

While this setup simplifies the hardware, it introduces new challenges for image processing due to the significant radial distortion inherent in fisheye lenses. Previous studies [5], [6] have attempted to adapt pre-trained models designed for perspective images to fisheye images through techniques like rectification [7] and data augmentation [8], but these methods often fall short in addressing the distortion. More recent work [9] demonstrates the effectiveness of convolutional neural networks (CNN) specifically designed for fisheye images, capturing detailed spatial information from a single spherical camera. However, utilizing this single-sensor data for generating accurate and reliable top-down views, such as bird's-eye view (BEV) maps, remains an area requiring further exploration.

BEV maps are essential for autonomous driving, as they provide a unified top-down representation of the environment that helps in sensor fusion [10], [11], motion forecasting [12], [13], and trajectory planning [14], [15]. These maps integrate raw sensor data into a format that is interpretable for downstream tasks, improving the system ability to predict vehicle motion [16], [17] and plan paths [18] effectively. Achieving accurate BEV maps requires the system to interpret spatial relationships within the scene and resolve issues related to distortion and occlusion, especially when relying solely on a single spherical camera.

While spherical imagery has been applied in previous research on tasks such as depth estimation [6], [9], [19], the research direction of generating BEV maps using only a single spherical camera has not been thoroughly investigated. Most existing approaches [20], [17], [21], [22], [23] depend on multiple-camera setups to mitigate challenges related to limited FoV. The reliance on additional sensors such as LiDAR [24] and radar [25] further complicates system integration and increases cost, highlighting a gap for approaches that focus on minimising hardware (sensor) complexity and cost while maintaining overall perception performance.

We address this gap by proposing the first approach to generate BEV maps from a single spherical camera in the context of autonomous driving. To achieve this, we have collected the first autonomous driving dataset specifically featuring a single spherical camera image. Furthermore, we introduce a benchmark architecture that enables the generation of BEV maps using only one camera, providing a streamlined and efficient solution for autonomous driving applications. Overall, our contributions can be summarized as follows:

- A novel large-scale real-world autonomous driving dataset comprising a (360°) spherical RGB camera, a high-fidelity 3D LiDAR (128 channels), and a GNSS/INS system. The first autonomous driving dataset with fully 3D bounding box annotation that features spherical camera modality.
- A benchmark for generating BEV maps from spherical images, with a novel spherical-image-to-BEV module that handles spherical distortions and maps 2D features onto a 3D sparse volume for accurate BEV representation.
- We introduce the use of focal loss, originally developed for object detection, as an innovative approach to address the extreme class imbalance in BEV segmentation. Our experiments demonstrate that this novel application of focal loss significantly improves segmentation performance, validating its effectiveness in the BEV domain.

II. Related Work

We consider prior work in two related topic areas: autonomous driving datasets (Section II-A) and vision based BEV model (Section II-B).

A. Autonomous Driving Datasets

For autonomous driving, real-world datasets are crucial and numerous have been published in recent years.

Real dataset *vs.* synthetic dataset. Autonomous driving datasets can typically be categorised into two types: real-world datasets [1], [29], [2], [3], [31], [4], [30] and synthetic datasets [27], [26], [28]. Acquiring a real-world outdoor dataset requires considerable effort, including sensor setup, route planning, and data post-processing. Conversely, synthetic datasets generated from simulators offer comparable information with advantages such as time efficiency, cost savings and flexible data configurations-allowing for customized camera setups, precise 3D location data, and detailed annotation information [32], [33], [34]. However, they often lack the realism and unpredictability of real-world data, which can lead to gaps in model robustness and overfitting to specific characteristics of the synthetic environment [35], [36]. To address these issues, secondary solutions like domain adaptation techniques are often employed to bridge the gap and make models trained on synthetic data applicable to real-world scenarios [37], [38], [39]. Additionally, the absence of sensor noise and artifacts in synthetic datasets may result in models that struggle when applied to noisy real-world data [40].

Perspective cameras *vs.* fisheye cameras. Cameras are essential sensors for autonomous driving and in both real and synthetic datasets, different types of cameras can be configured. For example, datasets such as KITTI [1], Waymo [2], and nuScenes [3] predominantly use perspective (pinhole) cameras to equip their vehicles. In nuScenes [3], five perspective cameras are used alongside a fisheye camera positioned at the rear of the vehicle. In contrast, datasets such as KITTI-360 [30] and WoodScape [29] rely exclusively on fisheye cameras, equipping their vehicles with multiple fisheye lenses. Fisheye lenses offer a significant advantage due to their wide FoV, allowing for 360-degree horizontal coverage with fewer cameras. For example, nuScenes [3] employs six cameras to achieve a 360-degree visual coverage, whereas WoodScape [29] requires four fisheye cameras, and KITTI-360 [30] manages with only two. However, no dataset to date has utilised a single spherical camera in place of all other cameras and achieved comprehensive 360-degree visual information specifically targeting BEV recovery.

B. Vision-based BEV Approaches

In autonomous driving, BEV map is useful for tasks such as object detection, path planning, and scene understanding. This process typically involves using multiple cameras positioned around the vehicle to capture the environment from various angles. The primary challenge is transforming 2D image data into a coherent 3D representation that can be accurately projected onto a BEV map. To address this challenge, several methods have been developed to lift 2D image features into 3D space before projecting them into a BEV map.

1) Multiple Perspective Camera Models: LSS [20] pioneered the concept of lifting 2D features into 3D space before splatting them into a BEV map. This method laid the foundation for subsequent advancements in BEV generation. FIERY [17] expanded upon this approach by integrating a multi-task framework, which uses uncertainty weighting to balance three critical sub-tasks: centerness, segmentation, and offset. SimpleBEV [25] further optimized the lifting strategy by introducing a bilinear-subsampling technique, which replaces the need for predicting depth distribution. PointBEV [23] improved upon this process by employing a coarse-to-fine mechanism, which reduces the indexing size during the lifting phase. These methods all rely on input from six surrounding-view cameras to generate feature maps, leading to more accurate and efficient BEV representations for downstream applications in autonomous driving.

2) Multiple Fisheye Camera Models: While the majority of BEV models have focused on perspective camera setups, there has been growing interest in leveraging fisheye cameras due to their wide field of view. F2BEV [28] is one of the few models that specifically addresses the challenges of generating BEV maps from multiple fisheye cameras. This approach is particularly advantageous for capturing a 360-degree view with fewer cameras, though it introduces additional complexities in handling the severe radial distortion inherent to fisheye lenses.

III. DUR360BEV Dataset

We introduce the first ever dataset that aims to use a single camera to solve real-world tasks in autonomous driving. Our dataset represents a shift towards the future of autonomous driving, where traditional systems often rely on an array of sensors, such as multiple cameras, LiDAR, and radar, working in tandem. By focusing on a single 360-degree camera, our approach not only simplifies the hardware setup but also reduces

Dataset	Real/Synthetic	Frames	FPS	Camera	LiDAR	GPS
SynWoodScape [26]	Synthetic	80K	10Hz	4 fisheye cams	No	No
OmniScape [27]	Synthetic	10K	N/A	2 fisheye cams	No	No
FB-SSEM [28]	Synthetic	20K	2Hz	4 fisheye cams	No	No
WoodScape [29]	Real	10K	N/A	4 fisheye cams	64-channel	GPS only
KITTI [1]	Real	15K	10Hz	1 stereo cam	64-channel	GPS only
KITTI-360 [30]	Real	78K	N/A	1 stereo + 2 fisheye cams	64-channel	GPS only
Waymo [2]	Real	198K	10Hz	5 perspective cams	32-channel	GPS only
nuScenes [3]	Real	40K	1Hz	5 perspective + 1 fisheye cams	32-channel	GPS+RTK refined
Lyft L5 [31]	Real	46	1Hz	7 perspective cams	32-channel	GPS only
DurLAR [4]	Real	0	N/A	1 stereo cam	128-channel	GPS only
Dur360BEV (ours)	**Real**	**32K**	**10Hz**	**1 spherical cam**	**128-channel**	**GPS+RTK refined**

TABLE I: Comparison between existing datasets (N.B. columns 'Frames' and 'FPS' in represent the frames labelled with 3D bounding box annotations and the frequency of these annotated frames in the dataset respectively; 'N/A' means that the information is not provided or the dataset has no annotation).

the complexity, cost, and power consumption of on-vehicle perception systems. Our Dur360BEV dataset comprises:

- **HD 360-degree camera imagery** in raw dual-fisheye format, where each pair of fisheye images are calibrated and can be in either equirectangular or cubemap formats.
- **Annotated dense LiDAR pointclouds** which has the resolution of 128×2048 and 3D bounding box annotation for vehicle, pedestrian and bicycles.
- **RTK-corrected GNSS/INS positioning** delivering exceptional accuracy, providing at most centimeter-level position data and high-precision vehicle attitude measurements, with 0.03° accuracy in pitch/roll and 0.15° in slip angle, ensuring not only highly reliable vehicle localization but also precise self-attitude assessment.
- **A High-Detail Semantic Map**, constructed using OpenStreetMap (OSM) in a geospatial database format, providing detailed environmental information surrounding the ego vehicle, as illustrated in Figure 1. The use of OSM ensures that the database remains flexible and up-to-date, benefiting from ongoing contributions by the OSM community.
- **Ground truth BEV segmentation map** which contains object and map tile information in the local environment around the ego vehicle.

A comparison between existing datasets is shown in Table I. Our Dur360BEV dataset has the highest resolution in terms of the LiDAR sensor, a relatively high annotated FPS and is the only autonomous dataset that provides single spherical camera images.

Sensor	Details
Camera	Spherical dual-fisheye camera (i.e., 360-degree camera, model: Ricoh Theta S), dual 1/2.3" 12M CMOS sensor, RGB image, 15Hz capture frequency, 1280x640 resolution, auto exposure, JPEG compressed, factory calibrated.
LiDAR	Ouster OS1-128 LiDAR sensor, 128 channel as vertical resolution, 2048 horizontal resolution, 10Hz capture frequency, 360 degree HFOV, -21.2 to 21.2 degree VFOV, 120m range @ $>50\%$ detection probability, 100m range @ $>90\%$ detection probability, 0.3cm range resolution.
GNSS/INS	OxTS RT3000v3 global navigation satellite and inertial navigation system, 100Hz capture frequency, 0.03 pitch/roll accuracy, 0.15 slip angle accuracy, centimeter level accuracy (with RTK corrections received via NTRIP).

TABLE II: Sensor details in Dur360BEV.

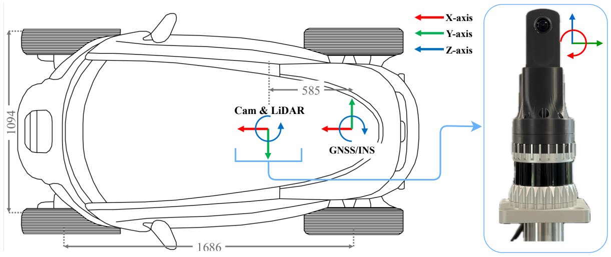

Fig. 3: Sensor placement. Left: the top view of the vehicle equipped with sensors. Right: our spherical camera on top of the LiDAR. Both figures show the coordinates space for each sensor.

A. Sensor Setup

The dataset is collected by a spherical camera, a high-resolution LiDAR and a GNSS/INS navigation system calibrated and equipped on a Renault Twizy vehicle. The details of the sensor is shown in Table II and the setup is illustrated in Figure 3.

B. Data Collection and Process

We collect data from various locations to ensure the dataset encompasses a diverse range of vehicles and traffic conditions. Specifically, we conduct data collection in four distinct areas of Durham, UK: the campus, highway, city center, and residential neighborhoods. These areas effectively represent the vast majority of driving environments across the UK. They include both straightforward scenarios, such as on highways where vehicle movements are relatively steady without parked cars on the roadside, and more challenging situations, such as in non-highway areas where vehicles might be parked in varying positions, and traffic patterns on the road become more complicated to predict due to additional traffic rules.

Data synchronisation: Given the inherent asynchrony in the data streams generated by different sensors operating at varied frequencies, we utilise the Robot Operating System (ROS Noetic) to achieve temporal alignment across different data sources based on their timestamps. The LiDAR, with its lower frame rate, serves as the reference sensor. We employ a synchronisation strategy that uses a queue size of 20 to handle incoming sensor messages and a slop parameter setting of 0.03 seconds to match messages from different sensors within this time window. This approach synchronises the dataset at 10 Hz, allowing for slight temporal discrepancies while ensuring accurate and coherent data integration, balancing the precision of alignment with the likelihood of successful message pairing.

3D bounding box annotations were labeled using a combination of automated and manual processes on the Xtreme1 open-source annotation platform [41]. The high-resolution LiDAR sensor used in our setup facilitates both automated detection and manual labeling, as the dense point cloud data makes it easier for the model to detect objects and for human annotators to accurately label them. Initially, an integrated LiDAR object detection model was employed on the platform, successfully identifying approximately 60% of the 3D bounding boxes for objects. Each bounding box annotation includes detailed data such as the 3D coordinates of the center, the rotation along the X, Y, and Z axes, the size of the bounding box in three dimensions, the sensor distance, and the annotated point amount. Following the automated process, an experienced data annotator meticulously reviewed and manually annotated the remaining objects within a $100m \times 100m$ square area centered around the ego vehicle. The dataset is at a frequency of 10 Hz, and objects are labelled into three distinct classes: vehicle, pedestrian, and bicycle.

C. Spherical Imagery

The proposed Dur360BEV dataset is the first to provide single spherical camera imagery specifically for autonomous driving tasks. Unlike previous datasets, such as nuScenes [3] and Waymo [2], which use multiple perspective cameras, or KITTI-360 [30] and WoodScape [29], which rely on numerous fisheye cameras to capture 360-degree horizontal imagery, our approach utilizes a single spherical camera to achieve comprehensive coverage. This reduces both the number of sensors required and the input data size for models in this domain. As shown in [25], while increasing input resolution in BEV model training on the nuScenes dataset can improve performance to some extent, it also significantly increases processing time, which is not ideal for real-time autonomous driving applications.

Our spherical imagery is crucial for calibrating the spherical camera with LiDAR data, enabling the accurate identification of corresponding pixels in an image based on the point cloud from the same frame. In practice, we find the best result by projecting 3D Cartesian coordinates (X,Y,Z) onto spherical dual-fisheye image coordinates (u,v) using a fourth-order polynomial transformation. First, the 3D coordinates are converted into spherical coordinates, where the azimuth angle $\theta = \arctan 2(Y,Z)$ and the polar angle $\phi = \arctan\left(\frac{\sqrt{Y^2+Z^2}}{X+\varepsilon}\right)$ are calculated. The polar angle ϕ is then mapped to a radius $r(\phi)$ using the piecewise function:

$$r(\phi) = a_4\phi^4 + a_3\phi^3 + a_2\phi^2 + a_1\phi + a_0, \tag{1}$$

where a_0, a_1, a_2, a_3, a_4 represent the coefficients of the polynomial that are determined through calibration to best fit the mapping between the spherical coordinates and the image plane. The 2D image coordinates $\mathbf{r} = \begin{pmatrix} x \\ y \end{pmatrix}$ are then computed as:

$$\mathbf{r} = r(\phi)\begin{pmatrix} \cos(\theta) \\ \sin(\theta) \end{pmatrix}. \tag{2}$$

Map the points from the front and back of the camera to their corresponding positions on the dual-fisheye image:

$$x = \begin{cases} \frac{x+1}{2}, & \text{if } X > 0, \\ \frac{x-1}{2}, & \text{if } X \leq 0. \end{cases} \tag{3}$$

Let H and W denote the height and width of the image, respectively. The pixel coordinates (u,v) are then given by the following expressions:

$$u = \frac{x+1}{2} \cdot W, \quad v = \frac{-y+1}{2} \cdot H, \tag{4}$$

avoiding any out-of-bounds coordinates:

$$u_{\text{dist}} = \text{clip}(u, 0, W-1), \quad v_{\text{dist}} = \text{clip}(v, 0, H-1). \tag{5}$$

IV. Methodology

To leverage the advantages of the Dur360BEV dataset, we propose a novel benchmark task that takes spherical images as input to generate BEV map of the scene. Our benchmark architecture can be divided into two parts: Spherical-image-to-BEV module (Section IV-A) and multi-task framework with focal loss (Section IV-B).

A. Spherical-Image-to-BEV module

As we replace the six input camera images used in previous work [20], [17], [21], [22], [25], [23] with a single spherical camera to simplify the hardware setup and reduce redundancy, the conventional image-to-feature module is no longer applicable to our dataset. To address this, we introduce a novel application of the spherical-image-to-BEV module. This new module is specifically designed to handle the unique challenges posed by spherical imagery.

Building upon the foundational ideas in [23], our approach begins by feeding an RGB spherical image, with dimensions $3 \times H \times W$, into a backbone network. This network outputs a feature map $I \in \mathbb{R}^{C \times H \times W}$, where $C, H, W \in \mathbb{N}$ represent the number of channels, height, and width of the feature map, respectively. Unlike traditional setups, which are tailored for perspective images from multiple cameras, our method is adapted to process the entire 360-degree field of view captured by a single spherical camera.

The backbone network extracts key features from the spherical image, which are then refined through a specifically tailored two-stage coarse-to-fine sampling strategy. This strategy is centered around what we call the Feature Pulling Process, which has been adapted to accommodate the distinct geometric properties of spherical images. By re-engineering the sampling geometry and refining the feature extraction process, we ensure that the module effectively captures and projects the spherical image features onto a BEV map, achieving accurate and reliable results.

The Feature Pulling Process begins by taking a set of predefined 2D BEV points and generating pillars, each composed of 3D points with dimensions $N_{\text{points}} \times 3$, where N_{points} represents the number of points sampled in this step. These 3D points are evenly spaced along the vertical axis in the BEV space. They are then projected onto the camera feature maps derived from the 360-degree imagery. Bilinear interpolation is applied to sample the corresponding 2D features, resulting in a high-dimensional feature volume with dimensions $N_{\text{points}} \times C$. This feature volume is then processed by a decoder, such as a sparse U-Net, which compresses the features onto the 2D BEV plane, generating initial BEV predictions.

Coarse sampling applies the Feature Pulling Process to a broader set of 2D BEV points, generating a sparse 3D volume with dimensions $N_{\text{coarse}} \times 3$. The resulting BEV predictions are used to identify high-confidence regions—those with the highest logit values—which are selected as anchor points.

Fine sampling then applies the Feature Pulling Process again based on the 3D points generated around the anchor points selected during the coarse sampling. This produces a refined feature volume with dimensions $N_{\text{fine}} \times C$, which focuses on enhancing the representation of critical regions. The outputs from the fine stage are combined with those from the coarse stage to produce a final, densified BEV map.

The coarse-to-fine sampling strategy plays a crucial role in efficiently generating BEV maps by focusing computational resources on high-confidence regions in our spherical images, thereby potentially alleviating class imbalance issues to some extent. However, to further enhance the ability of the model to handle severe class imbalance, we integrate focal loss [42] into our training process.

B. Multi-Task Framework and Loss Functions

Following the Multi-Task Framework [17], our benchmark architecture incorporates three specialized segmentation heads—*i.e.*, centerness, offset, and segmentation—each targeting a distinct aspect of the BEV map prediction. The centerness head predicts the likelihood of a location being the center of an object, the offset head estimates the spatial displacement from a predefined anchor point, and the segmentation head differentiates between foreground and background regions in the BEV map. For the centerness and offset tasks, we utilize a balanced mean squared error (MSE) loss and an absolute error loss, respectively. Given the significant class imbalance between foreground and background in BEV maps, we specifically apply focal loss [42] to the segmentation head to enhance model focus on difficult-to-classify regions.

Focal loss extends the standard cross-entropy (CE) loss, which is commonly used for binary classification. The cross-entropy loss is defined as $\text{CE}(p,y) = \text{CE}(p_t) = -\log(p_t)$, where $y \in 0,1$ denotes the ground-truth label, and $p \in [0,1]$ represents the predicted probability for the positive class ($y = 1$). For uniformity, we define p_t as:

$$p_t = \begin{cases} p, & \text{if } y = 1, \\ 1 - p, & \text{otherwise.} \end{cases} \tag{6}$$

One of the key challenges in training models for tasks like BEV segmentation is that standard cross-entropy loss tends to be dominated by straightforward examples, potentially overwhelming rare classes with small loss values. To better handle this, focal loss modifies the loss function by reducing the contribution of these simpler examples, thereby shifting the focus of training towards harder negatives. This is achieved by introducing a modulating factor $(1 - p_t)^\gamma$ with a tunable parameter $\gamma \geq 0$, leading to the focal loss formulation:

$$\text{FL}(p_\text{t}) = -(1 - p_\text{t})^\gamma \log(p_\text{t}). \tag{7}$$

This approach effectively reduces the influence of simpler classified examples, allowing the model to concentrate on learning from more challenging cases, which is crucial for handling imbalanced data in tasks like BEV segmentation.

V. EXPERIMENTS

In this section, we outline the experimental setup used to evaluate our proposed SI2BEV module. We conduct a comparative analysis of two sampling strategies: dense grid sampling [25] and a combination of sparse and dense sampling [23], specifically focusing on their performance within our SI2BEV module on the Dur360BEV dataset. Additionally, we investigate the impact of varying the gamma parameter in the focal loss on BEV task performance.

A. Experimental Setup

Dataset: Our experiments are conducted on the Dur360BEV dataset, a challenging real-world spherical image dataset tailored for autonomous driving applications. It consists of 16.4k point cloud frames, with 14.7k frames used for training and 1.6k for validation. The vehicle class is selected for training and evaluation.

Evaluation Protocol: Ground truth BEV maps are generated using the 3D bounding box annotations for the vehicle class. Pixels within these bounding boxes on the BEV map are labeled as positive, while all other pixels are labeled as negative. The evaluation metric is Intersection over Union (IoU), defined as the ratio of the overlap between predicted and ground truth positive regions to their union. Higher IoU values reflect better alignment and model performance.

Implementation Details: For all experiments, the proposed architecture is trained over 4k iterations, each with 5 batches, using the AdamW optimizer [43], with a learning rate of $\lambda = 5e^{-5}$, weight decay $\text{w} = 10^{-7}$, and a 1-cycle learning rate schedule [44]. We closely monitor the validation loss throughout the training process, which consistently decreases to converge at a stable point, as shown in Figure 4. This convergence was achieved within the set 4000 iterations, beyond which the residuals of the loss function showed minimal change. The results confirm that the model effectively learns and stabilizes within this iteration limit, demonstrating the efficiency and reliability of our approach.

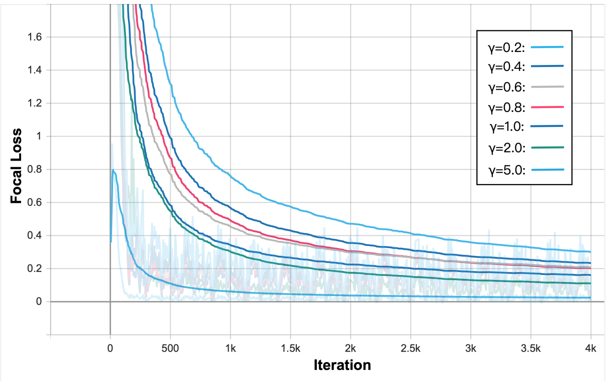

Fig. 4: Validation loss curves for different values of γ. From top to bottom: $\gamma = 0.2, 0.4, 0.8, 0.6, 1, 2, 5$. The curves illustrate how the choice of γ influences the convergence behavior during training.

B. Compare sampling strategies

We compare performance between the sparse/dense strategy proposed in our architecture and the dense grid sampling strategy used in SimpleBEV [25]. Both methods are designed to detect objects within a 100m×100m grid with a 50cm resolution resulting in a 200×200 BEV map.

For a fair comparison, both sampling strategies were trained under identical conditions. The comparison results are summarized in Table III. It is important to note that when $\gamma = 0$, the focal loss simplifies to a standard Binary Cross Entropy (BCE) loss.

As observed in Table III, focal loss significantly enhances performance across both sampling strategies. The Dense Grid strategy shows a notable improvement in IoU by +1.6 at the 100m range, while the Coarse/Fine strategy achieves a +1.1 increase in IoU. These results highlight the effectiveness of focal loss in addressing class imbalance, particularly over extensive spatial ranges, thereby improving the accuracy of BEV segmentation. When considering the model complexity, the Coarse/Fine sampling strategy with a $\gamma = 2$ setup demonstrates the best overall performance, balancing both IoU improvement and model efficiency, as shown in Table IV. The qualitative visualisation of the result for this optimal setup is presented in Figure 5.

VI. CONCLUSIONS

We introduce Dur360BEV, the first large-scale autonomous driving dataset to feature a spherical RGB camera, high-fidelity 128-channel 3D LiDAR, and fully 3D annotated bounding boxes. This dataset simplifies hardware complexity while maintaining rich environmental data, advancing the state-of-the-art in BEV recovery for autonomous driving.

We also develop a benchmark architecture with the Spherical-Image-to-BEV (SI2BEV) module, effectively addressing the challenges of spherical imagery to produce accurate BEV maps. Our experiments further demonstrate that the incorporation of focal loss significantly enhances BEV segmentation performance, particularly in addressing class imbalance inherent in 360-degree camera datasets. This underscores the importance of considering class imbalance when generating BEV maps, especially in challenging environments. Collectively, our contributions provide robust tools and methodologies that enhance the development of autonomous driving technologies using a simpler low-cost, low-power sensing option.

Strategy	γ	Backbone	IoU_{100}	IoU_{50}	IoU_{20}	Eff. Score
Dense Grid	5	RN-101	30.7	37.9	39.3	0.73
Dense Grid	2	RN-101	31.5	38.3	39.9	0.75
Dense Grid	1	RN-101	**32.7**	**40.4**	**42.0**	0.78
Dense Grid	0	RN-101	31.1	37.0	38.5	0.74
Coarse/Fine	5	EN-b4	26.9	34.3	36.8	3.20
Coarse/Fine	2	EN-b4	**32.6**	**40.3**	**41.6**	**3.88**
Coarse/Fine	1	EN-b4	28.9	36.3	39.4	3.44
Coarse/Fine	0	EN-b4	31.5	38.9	39.7	3.75
Coarse/Fine	0.8	EN-b4	29.5	36.7	37.6	3.51
Coarse/Fine	0.6	EN-b4	27.5	35.2	37.4	3.27
Coarse/Fine	0.4	EN-b4	31.0	38.3	40.0	3.69
Coarse/Fine	0.2	EN-b4	31.0	38.4	40.3	3.69

TABLE III: Comparison the BEV vehicle segmentation on Dur360BEV dataset between two sampling strategies. Computed on the validation split at different values of γ parameter in focal loss. 'EN-b4' and 'RN-101' stand for EfficientNet-b4 [45] and ResNet101 [46] respectively. $\text{IoU}_{100}, \text{IoU}_{50}, \text{IoU}_{20}$ represent the IoU scores for the BEV maps in range 100m, 50m and 20m respectively. The 'Eff. Score' represents the ratio of the IoU_{100} to the number of parameters (in millions). All the model are trained on Dur360BEV training split, with batch size=6, learning rate=5e-5 and iterations=4000 for fairness.

Strategy	C	IoU_{100}	C:IoU_{100}
Dense Grid ($\gamma = 1$)	42.04	32.7	0.777
Coarse/Fine ($\gamma = 2$)	8.40	32.6	3.881

TABLE IV: The model complexity is calculated by the ratio of the IoU to the number of model parameters (C) in millions.

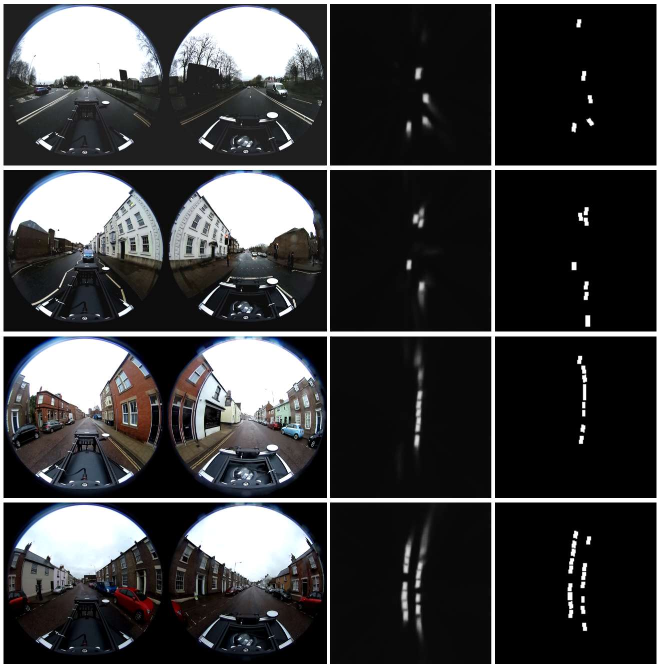

Fig. 5: The inference visualisation of the Coarse/Fine sampling strategy and focal loss with $\gamma = 2$ on Dur360BEV validation split. Left: Input image; Middle: Prediction; Right: Ground Truth Map.

References

[1] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 3354–3361.

[2] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, *et al.*, “Scalability in perception for autonomous driving: Waymo open dataset,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 2446–2454.

[3] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, “nuscenes: A multimodal dataset for autonomous driving,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631.

[4] L. Li, K. N. Ismail, H. P. H. Shum, and T. P. Breckon, “DurLAR: A high-fidelity 128-channel LiDAR dataset with panoramic ambient and reflectivity imagery for multi-modal autonomous driving applications,” in *2021 International Conference on 3D Vision (3DV)*. IEEE, 2021, pp. 1227–1237.

[5] E. Plaut, E. Ben Yaacov, and B. El Shlomo, “3d object detection from a single fisheye image without a single fisheye training image,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3659–3667.

[6] M. Rey-Area, M. Yuan, and C. Richardt, “360monodepth: High-resolution 360deg monocular depth estimation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3762–3772.

[7] Z. Xue, N. Xue, G.-S. Xia, and W. Shen, “Learning to calibrate straight lines for fisheye image rectification,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1643–1651.

[8] V. H. Duong, D. Q. Nguyen, T. Van Luong, H. Vu, and T. C. Nguyen, “Robust data augmentation and ensemble method for object detection in fisheye camera images,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 7017–7026.

[9] M. Li, X. Jin, X. Hu, J. Dai, S. Du, and Y. Li, “Mode: Multi-view omnidirectional depth estimation with 360^{o} cameras,” in *European Conference on Computer Vision*. Springer, 2022, pp. 197–213.

[10] Z. Liu, H. Tang, A. Amini, X. Yang, H. Mao, D. L. Rus, and S. Han, “Bevfusion: Multi-task multi-sensor fusion with unified bird’s-eye view representation,” in *2023 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2023, pp. 2774–2781.

[11] Y. Man, L.-Y. Gui, and Y.-X. Wang, “Bev-guided multi-modality fusion for driving perception,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 21 960–21 969.

[12] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. R. Qi, Y. Zhou, Z. Yang, A. Chouard, P. Sun, J. Ngiam, V. Vasudevan, A. McCauley, J. Shlens, and D. Anguelov, “Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 9710–9719.

[13] L. Zhang, P. Li, S. Liu, and S. Shen, “Simpl: A simple and efficient multi-agent motion prediction baseline for autonomous driving,” *IEEE Robotics and Automation Letters*, 2024.

[14] W. Zeng, W. Luo, S. Suo, A. Sadat, B. Yang, S. Casas, and R. Urtasun, “End-to-end interpretable neural motion planner,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8660–8669.

[15] H. Li, P. Chen, G. Yu, B. Zhou, Y. Li, and Y. Liao, “Trajectory planning for autonomous driving in unstructured scenarios based on deep learning and quadratic optimization,” *IEEE Transactions on Vehicular Technology*, 2023.

[16] P. Wu, S. Chen, and D. N. Metaxas, “Motionnet: Joint perception and motion prediction for autonomous driving based on bird’s eye view maps,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[17] A. Hu *et al.*, “FIERY: Future Instance Prediction in Bird’s-Eye View From Surround Monocular Cameras,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 273–15 282, accessed: Jun. 23, 2024.

[18] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, W. Wang, *et al.*, “Planning-oriented autonomous driving,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 853–17 862.

[19] Q. Feng, H. P. Shum, and S. Morishima, “360 depth estimation in the wild-the depth360 dataset and the segfuse network,” in *2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. IEEE, 2022, pp. 664–673.

[20] J. Philion and S. Fidler, “Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d,” in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*. Springer, 2020, pp. 194–210.

[21] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Y. Qiao, and J. Dai, “Bevformer: Learning bird’s-eye-view representation from multi-camera images via spatiotemporal transformers,” in *European conference on computer vision*. Springer, 2022, pp. 1–18.

[22] C. Yang, Y. Chen, H. Tian, C. Tao, X. Zhu, Z. Zhang, G. Huang, H. Li, Y. Qiao, L. Lu, *et al.*, “Bevformer v2: Adapting modern image backbones to bird’s-eye-view recognition via perspective supervision,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 830–17 839.

[23] L. Chambon, E. Zablocki, M. Chen, F. Bartoccioni, P. Pérez, and M. Cord, “PointBeV: A Sparse Approach for BeV Predictions,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 15 195–15 204.

[24] T. Liang, H. Xie, K. Yu, Z. Xia, Z. Lin, Y. Wang, T. Tang, B. Wang, and Z. Tang, “Bevfusion: A simple and robust lidar-camera fusion framework,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 10 421–10 434, 2022.

[25] A. W. Harley, Z. Fang, J. Li, R. Ambrus, and K. Fragkiadaki, “Simple-BEV: What Really Matters for Multi-Sensor BEV Perception?” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 2759–2765.

[26] A. R. Sekkat, Y. Dupuis, V. R. Kumar, H. Rashed, S. Yogamani, P. Vasseur, and P. Honeine, “SynWoodScape: Synthetic Surround-View Fisheye Camera Dataset for Autonomous Driving,” *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8502–8509, July 2022.

[27] A. R. Sekkat, Y. Dupuis, P. Vasseur, and P. Honeine, “The omniscape dataset,” in *2020 IEEE International conference on robotics and automation (ICRA)*. IEEE, 2020, pp. 1603–1608.

[28] E. U. Samani, F. Tao, H. R. Dasari, S. Ding, and A. G. Banerjee, “F2BEV: Bird’s Eye View Generation from Surround-View Fisheye Camera Images for Automated Driving,” in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2023, pp. 9367–9374.

[29] S. Yogamani, C. Hughes, J. Horgan, G. Sistu, S. Chennupati, M. Uricar, S. Milz, M. Simon, K. Amende, C. Witt, H. Rashed, S. Nayak, S. Mansoor, P. Varley, X. Perrotton, D. Odea, and P. Perez, “WoodScape: A Multi-Task, Multi-Camera Fisheye Dataset for Autonomous Driving,” in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2019, pp. 9307–9317.

[30] 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*, pp. 3292–3310, Mar. 2023.

[31] J. Houston, G. Zuidhof, L. Bergamini, Y. Ye, L. Chen, A. Jain, S. Omari, V. Iglovikov, and P. Ondruska, “One thousand and one hours: Self-driving motion prediction dataset,” in *Conference on Robot Learning*. PMLR, 2021, pp. 409–418.

[32] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “Carla: An open urban driving simulator,” in *Conference on robot learning*. PMLR, 2017, pp. 1–16.

[33] S. Shah, D. Dey, C. Lovett, and A. Kapoor, “Airsim: High-fidelity visual and physical simulation for autonomous vehicles,” in *Field and Service Robotics: Results of the 10th International Conference*. Springer, 2017, pp. 621–635.

[34] N. Koenig and A. Howard, “Design and use paradigms for gazebo, an open-source multi-robot simulator,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2004, pp. 2149–2154.

[35] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, “Playing for data: Ground truth from computer games,” *International Journal of Computer Vision*, vol. 125, no. 1-3, pp. 127–144, Dec. 2017.

[36] J. Tremblay *et al.*, “Training deep networks with synthetic data: Bridging the reality gap by domain randomization,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 1082–1090.

[37] H. D. III, “Frustratingly easy domain adaptation,” in *Proceedings of the*

47th Annual Meeting of the Association for Computational Linguistics (ACL). Association for Computational Linguistics, 2009, pp. 256–263.

[38] J. Hoffman, D. Wang, F. Yu, and T. Darrell, "Fcns in the wild: Pixel-level adversarial and constraint-based adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 3427–3436.

[39] Y. Wang, W. Liu, and Z. Wang, "Unsupervised domain adaptation for autonomous driving with synthesized data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2019, pp. 1238–1247.

[40] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "Virtual KITTI: Analyzing visual systems with synthetic scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 578–585.

[41] L. A. . D. Foundation, "Xtreme1 - the next gen platform for multisensory training data," 2023, software available from https://github.com/xtreme1-io/xtreme1/. [Online]. Available: https://xtreme1.io/

[42] T.-Y. Ross and G. Dollár, "Focal loss for dense object detection," in *proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2980–2988.

[43] I. Loshchilov, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[44] L. N. Smith and N. Topin, "Super-convergence: Very fast training of neural networks using large learning rates," in *Artificial intelligence and machine learning for multi-domain operations applications*, vol. 11006. SPIE, 2019, pp. 369–386.

[45] M. Tan, "Efficientnet: Rethinking model scaling for convolutional neural networks," *arXiv preprint arXiv:1905.11946*, 2019.

[46] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.