JRDB-PanoTrack: An Open-world Panoptic Segmentation and Tracking Robotic Dataset in Crowded Human Environments

Duy Tho Le[1*], Chenhui Gou[1*], Stavya Datta[1], Hengcan Shi[1†],
Ian Reid[2,3], Jianfei Cai[1], Hamid Rezatofighi[1]
[1]Monash University, [2]MBZUAI, [3]University of Adelaide,
{tho.le1, chenhui.gou, hengcan.shi}@monash.edu
[*]Equal contribution, [†]Corresponding author
https://jrdb.erc.monash.edu/dataset/panotrack

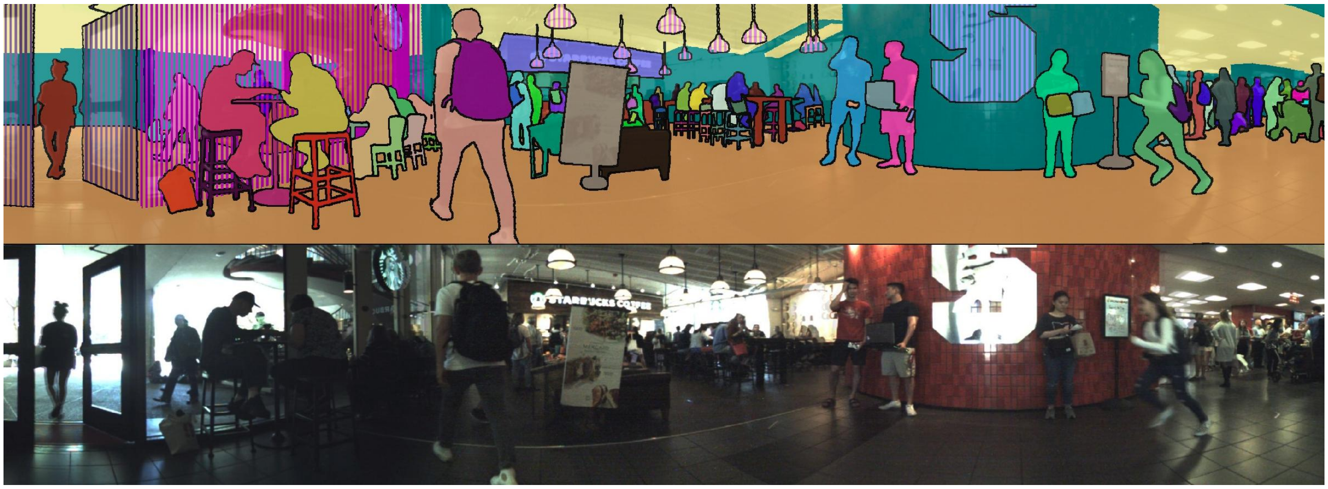

Figure 1. A panoramic frame (bottom) and panoptic annotation (top) from our *JRDB-PanoTrack* dataset. Our dataset features multi-label panoptic annotations, highlighted by the striped areas where multiple objects coexist. JRDB-PanoTrack also provides consistent tracking IDs for all *thing* classes across long periods of occlusion.

Abstract

Autonomous robot systems have attracted increasing research attention in recent years, where environment understanding is a crucial step for robot navigation, human-robot interaction, and decision. Real-world robot systems usually collect visual data from multiple sensors and are required to recognize numerous objects and their movements in complex human-crowded settings. Traditional benchmarks, with their reliance on single sensors and limited object classes and scenarios, fail to provide the comprehensive environmental understanding robots need for accurate navigation, interaction, and decision-making. As an extension of JRDB dataset, we unveil JRDB-PanoTrack, a novel open-world panoptic segmentation and tracking benchmark, towards more comprehensive environmental perception. JRDB-PanoTrack includes ***(1)*** *various data involving indoor and outdoor crowded scenes, as well as comprehensive 2D and 3D synchronized data modalities;* ***(2)*** *high-quality 2D spatial panoptic segmentation and temporal tracking annotations, with additional 3D label projections for further spatial understanding;* ***(3)*** *diverse object classes for closed- and open-world recognition benchmarks, with OSPA-based metrics for evaluation. Extensive evaluation of leading methods shows significant challenges posed by our dataset.*

1. Introduction

With the increasing demands for autonomous robots in human-crowded environments, environment understanding becomes paramount, which serves as a vital step in many robotic systems, such as navigation and human-robot interaction. Specifically, human-centric environment understanding can be mainly divided into two aspects: spatial and temporal understanding. Spatial understanding aims to

	Data			Domain			No. Class						
Dataset	**Data**	**Temp**	**Pano Cov.**	**In door**	**Out door**	**Platform**	**Thing**	**Stuff**	**Open world**	**Trk Len**	**No. Seq**	**No. Smp**	**No. M**
PanopticCOCO [20]	I			✓	✓	Int	80	91		-		164k	
Cityscapes [9]	I	✓		✓		Car	8	11			500	3k	10k
VIPSeg [25]	I	✓		✓	✓	Int	58	66		≤10s	3536	85k	926k
MOT-STEP [32]	I	✓			✓	Int	1	6		≤19s	4	2k	17k
KITTI-STEP [32]	I	✓			✓	Car	2	17		≤65s	50	19k	126k
Waymo [24]	I	✓	220°		✓	Car	8	20		≤1.2s	2060	100k	
SemanticKITTI [3]	P	✓			✓	Car	14	11			21	43k	
Nuscenes [12]	P	✓	360°		✓	Car	23	6			1000	40k	1.2M
JRDB-PanoTrack	I/P	✓	360°	✓	✓	Rob	60	11	✓	≤117s	54	20k	428k

Table 1. Typical datasets for 2D-3D panoptic segmentation and tracking. Abbreviations: I (Image), P (Point Cloud), Car (Autonomous Car), Rob (Mobile Robot), Int (Internet images/videos), Temp (Temporal data), Pano Cov. (Panoramic Coverage), No. Class (The number of classes), Trk Len (Track Length), No. Seq (The number of sequences), No. Smp (The number of samples) and No. M (the number of masks).

distinguish objects in human-crowded environments, while temporal understanding expects to recognize temporal relations of such objects.

The existing datasets for environment understanding, sourced primarily from self-driving vehicles [3, 9, 12, 24], or internet images/videos [20, 25], exhibit clear domain gaps when applied to robotic environments. These sources typically offer different perspectives from robots, and fail to encapsulate the challenges and interactions specific to robotic systems. As shown in Tab. 1, most of the existing datasets only contain a single data modality (RGB images or point clouds) [9, 25] and a small number of classes [9, 32]. They also lack temporal information [20] or 360-degree panoramic spatial perspectives [9, 24, 32]. In contrast, real-world applications where the robotic agents are deployed, usually involve multi-modal data, diverse classes, and both spatial and temporal understanding.

Built on top of JRDB dataset [11, 23, 30], inheriting its comprehensive annotation suite for human bodies, we introduce *JRDB-PanoTrack*, a novel comprehensive dataset for human-crowded environment understanding. Firstly, JRDB-PanoTrack offers a comprehensive dataset from various indoor and outdoor crowded scenes with 2D and 3D synchronized data modalities, supporting visual and robotic applications. Secondly, high-quality 2D panoptic segmentation and tracking annotations are provided for both spatial and temporal environment understanding, including 428K panoptic masks, 27K tracking labels and 7.3B annotated pixels. Additional 3D label projections are also presented for further spatial understanding. Thirdly, we introduce diverse objects and open-world benchmarks for generalization research. Finally, JRDB-PanoTrack annotates multiple classes for some areas, such as objects behind *glass* or hang on *wall* in Fig. 1. We propose metrics based on optimal sub-pattern matching (OSPA) to deal with such evaluation.

Based on the JRDB-PanoTrack dataset, we present several benchmarks, including *Closed-world (CW)* and *Open-world (OW)* panoptic segmentation and tracking. We extensively evaluate state-of-the-art (SOTA) methods on these benchmarks. Moreover, SOTA methods are also estimated on our 3D label projections. The results underline the imperative need for advanced methodologies that can adeptly handle the complexities presented by complex human-crowded environments. Our main contributions are:

- We present JRDB-PanoTrack, an extensive new dataset for spatial and temporal robotic environment understanding. In JRDB-PanoTrack, high-quality panoptic segmentation and tracking annotations are provided. We employ comprehensive data collected by a mobile robot, including 2D&3D modalities as well as indoor&outdoor human-crowded scenes.
- Closed- and open-world benchmarks are proposed for generalizable environment understanding. Our dataset also contains multi-class annotations and OSPA-based metrics for evaluation.
- We conduct extensive evaluations of SOTA closed- and open-world segmentation/tracking methods on JRDB-PanoTrack, and discuss their strengths and weaknesses.

2. Related Work

Panoptic Segmentation and Tracking Datasets. ***Panoptic segmentation***, as introduced in [17], is a task to generate instance-level masks for *thing* objects (countable, distinct entities) and class-level masks for *stuff* objects (amorphous and uncountable regions) to achieve a more complete visual understanding. Datasets like PanopticCOCO [20], ADE20K [41] and Cityscapes [9] are widely popular in this space, primarily focusing on 2D images. However, these datasets only support spatial understanding.

Panoptic tracking further integrates multi-object tracking

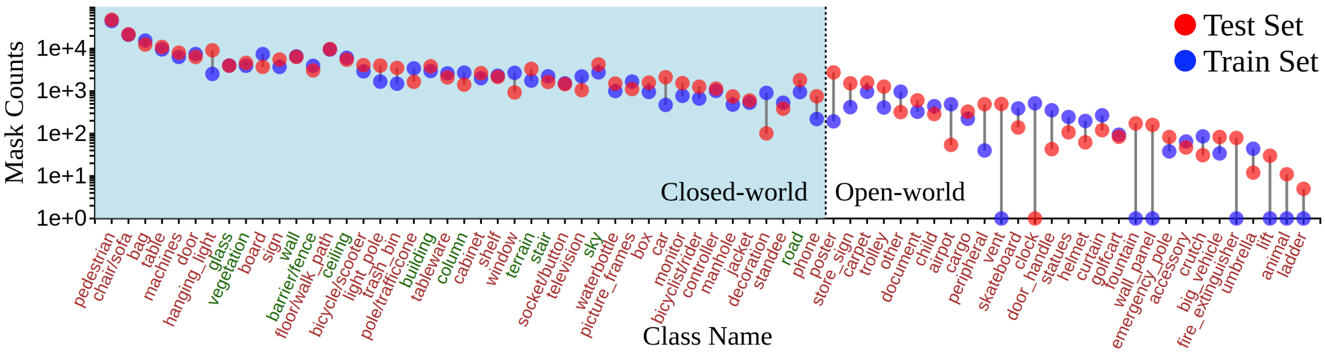

Figure 2. Distribution of object masks of *thing* (brown) and *stuff* (green) classes in JRDB-PanoTrack train and test sets, where x and y-axis indicate the class names and mask counts, respectively. Best viewed in color and zoomed in.

into panoptic segmentation, as seen in datasets like MOTSTEP [32], VIPSeg [25], and Waymo [24] for 2D tracking, and SemanticKITTI [3], Panoptic Nuscenes [12] for 3D tracking. However, these datasets, often sourced from self-driving cars [3, 12, 15, 24, 32], single-view surveillance cameras [32] or miscellaneous internet videos [25]. These datasets, although useful and large-scale, fall short in representing complex, human-centric environments for autonomous robotics due to the lack of synchronized multi-modal multi-view data, diverse object classes, complex human-crowded scenes, and domain consistency. Our JRDB-PanoTrack dataset addresses this gap by providing synchronized 2D and 3D data from a social mobile manipulator, capturing the complexity of crowded human spaces, offering diversity in objects and unique challenges in both closed-world and open-world settings.

OW Benchmarks. The development of OW benchmarks is crucial for assessing the generalization capabilities of models in diverse and unpredictable environments. Large-scale segmentation datasets such as COCO [20] and ADE20K [41] and OW segmentation datasets [27] are usually used for OW spatial understanding, while several datasets like TAO-OW [21] and OVTrack [19] are introduced for OW bounding box tracking. Moreover, these datasets are all from internet images/videos. Different from them, JRDB-PanoTrack introduces a unique and challenging OW benchmark for panoptic segmentation and tracking in robotic environments, with both 2D and 3D data modalities.

Previous JRDB Datasets. JRDB [23] is a large-scale and comprehensive dataset for autonomous robot research in human-centric environments. It collects 2D and 3D point cloud videos, audio as well as GPS positions by a social manipulator robot. In previous JRDB [23], JRDB-Act [11] and JRDB-Pose [30], 2D-3D human detection, tracking and forecasting, body skeleton pose estimation, human social grouping and activity recognition annotations have been introduced. In JRDB-PanoTrack, we complement this JRDB by providing new open-world panoptic segmentation and tracking annotations for more comprehensive human-centered scene understanding.

SOTA Frameworks. For ***panoptic segmentation***, initial approaches [16, 17, 33] handle semantic and instance segmentation as separate tasks by using dual sub-networks. Max-deeplab [31] introduces transformer-based architectures, moving away from bounding box-dependent models. Recent developments, including K-net [39], MaskFormer [6], Mask2Former [8] and Mask DINO [18], unifies semantic, instance and panoptic segmentation into a singular mask proposal prediction framework. In the OW domain, methods like [10, 28, 34, 35] generate mask proposals for all panoptic objects, and then align them with object names via large vision-language pre-training.

For ***multi-object tracking***, traditional motion-model-based algorithms often outperform modern integrated systems. SORT [4] exemplifies this with its linear-motion-based track association. ByteTrack [40] introduces low-confidence detection associations to improve tracking. OC-SORT [5] enhances filters and recovery strategies to solve the non-linear motion problem. More recently, BoT-SORT [1] advances the field by optimizing the Kalman filter state and incorporating camera-motion compensation. Notably, to the best of our knowledge, there is no available OW panoptic tracking method. We use those strong and popular trackers as baselines in our experiments.

3. The JRDB-PanoTrack Dataset

3.1. Dataset and Statistics

Data. JRDB-PanoTrack encompasses 20,000 images, sampled at 1Hz from 54 videos in the original JRDB dataset [23]. 4,000 360-degree panoramic images can be generated by merging 5 original images from 5 different camera views. 4,000 point clouds are also provided for 3D understanding.

Annotation. JRDB-PanoTrack retains all annotations from JRDB[23] and further enhances the dataset by introducing

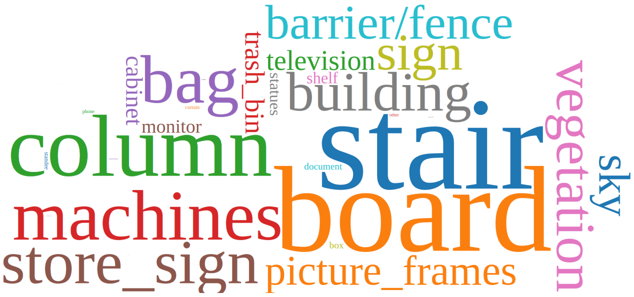

Figure 3. Word cloud of the most frequent classes seen through glass in JRDB-PanoTrack, with the size of the word proportional to the frequency of the class.

428K 2D panoptic segmentation and 27K tracking annotations to enable environment understanding.

Annotation process. The annotation process starts with an unlimited list of classes which can be extended by all annotators, any clearly visible and semantically meaningful objects would be annotated, objects that are behind the glass or being hang on wall will be annotated with multiple labels. Then annotators produce labels and senior annotators control the quality by multiple inspection rounds.

Object Class. There are 72 objects in JRDB-PanoTrack, which are classified into 61 *thing* (such as pedestrians, cars and laptops) and 11 *stuff* (like sky and walls) classes. Fig. 2 depicts the distributions of classes.

Special Class Labeling. Our dataset aims to analyze common environments for autonomous robots. There are some differences from traditional environment understanding datasets. **(1)** ***Floor differentiation***: human-robot interaction and navigation require robots to distinguish different floors. To address this, we provide instance segmentation labels for floors and regard them as *thing* objects. **(2)** ***Multi-class segmentation***: In modern environments, objects are often seen behind *windows* and *glass*, and sometimes being hang on walls (most frequently seen objects are shown in Fig. 3). Traditional datasets usually simply ignore these objects or interaction, while they might be crucial for environment understanding. In JRDB-PanoTrack, there are 9% of objects belonging to such cases. Therefore, we label multiple classes for pixels belonging to these objects, *i.e.*, including the front *windows* or *glass*, and the behind objects. We hope this will encourage the community to develop more robust models for better scene understanding.

Tasks. Our dataset supports panoptic segmentation and tracking tasks. ***Panoptic segmentation*** [17] expects to spatially understand environments, which generates masks for all the *thing* and *stuff* objects. ***Panoptic tracking*** [14] understands environments on both spatial and temporal aspects. It segments both *thing* and *stuff* objects, and tracks *thing* objects throughout a video. As shown in Fig. 4, there are up to 81 masks in an image, and the average number of masks is 22. In panoramic views, the maximum and average mask counts per image are 245 and 80, respectively. Fig. 4 also highlights the track length distribution in JRDB-PanoTrack. The maximum and average track lengths are 117s and 16s, respectively. The most populated scene in our dataset comprises a staggering 564 tracklets (1010 in panoramic views) in a single sequence, compared to the average 101 tracklets (198 in panoramic views) per sequence. According to Fig. 4, the testing set is more crowded than the training set with more masks per image and more tracklets per sequence.

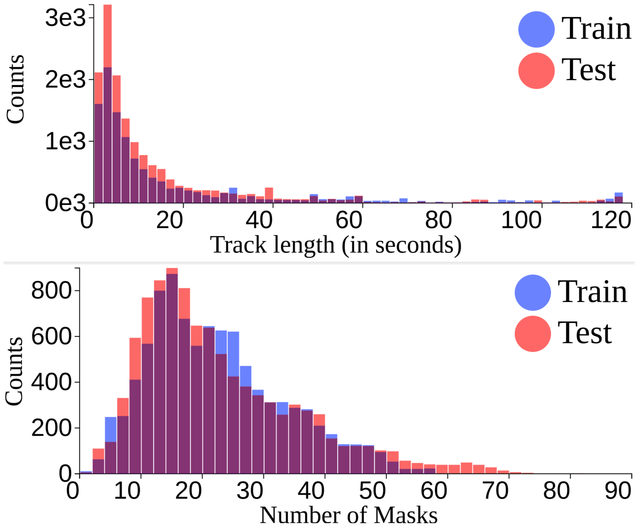

Figure 4. Analysis of Track length distribution (top) and Number of masks per frame (bottom) in the JRDB-PanoTrack dataset. Best viewed in color.

	# Inst. / img		**# Trk. / seq.**		**Track length**	
	Mean	**Max**	**Mean**	**Max**	**Mean**	**Max**
Train	22 / 78*	57 / 144*	87 / 178*	249/ 420*	17 / 27*	116/ 116*
Test	22 / 80*	81 / 245*	105/ 217*	564/ 1010*	14 / 24*	117/ 117*

[*] statistics for panoramic images

Table 2. The number of masks per image, the number of tracklets per sequence, as well as track lengths (in seconds).

Thing and Stuff classes. Following [17], we divide the object classes into *thing* and *stuff* classes. *Thing* classes are objects that can be segmented and tracked, such as *person, car, bicycle, chair, table, laptop, bottle, etc. Stuff* classes are background classes that can be segmented but not tracked, such as *sky, ground, wall, etc.* Fig. 2 shows the distribution of object instances in JRDB-PanoTrack, where *pedestrian* is the most frequent class with more than 40k instances, followed by the commonly seen objects in human-centric environments such as *chair, bag, table, door, board, machines, etc.* with 10k to 50k instances.

CW and OW. Based on class distributions shown in Fig. 2,

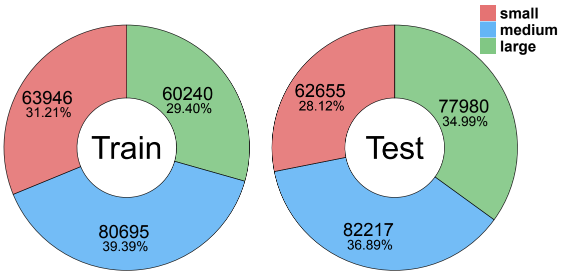

Figure 5. The count (top) and percentage (bottom) of Small, Medium and Large masks in JRDB-PanoTrack training and testing sets. Small and Large masks are the masks $\leq 32^2$ and $\leq 96^2$ pixels, and the sizes of Medium masks are in between. Image size is 752x480 (W x H).

we divide our 72 panoptic classes into two sets: 43 common and 29 long-tail classes as *known* and *unknown* classes, respectively. The 43 *known* classes can be used for training and evaluation at the closed-world (***CW***) scenario. At the open-world (***OW***) scenario, the 43 *known* classes can be employed for training, while 28 *unknown* classes are for testing (there is one class that occur in training set only).

Tracklet statistics. Tab. 2 offers detailed statistics on the number of masks per image, tracklets per sequence, and track lengths in the JRDB-PanoTrack dataset. It underscores the dataset's depth and diversity, with some tracks extending up to 117 seconds across multiple camera views. According to Tab. 2, the testing set is more crowded than the training set with more masks per image and more tracklets per sequence.

Mask Size. The distribution of mask sizes in JRDB-PanoTrack are as presented in Fig. 5. We have balanced mask sizes in both training and testing sets, which bring challenges to panoptic segmentation and tracking methods to carefully deal with objects of various sizes.

3.2. Benchmark and Metrics

Benchmark. Based on our JRDB-PanoTrack dataset, we propose several benchmarks for environment understanding, the four categories are:

- ***CW panoptic segmentation.***
- ***OW panoptic segmentation.***
- ***CW panoptic tracking.***
- ***OW panoptic tracking.***

In all benchmarks, we use half of our dataset for training, i.e., 9365 images in 27 sequences. For testing, we employ 9280 images in the other 27 sequences. Panoramic images and point clouds corresponding to the 9365/9280 images can be used for panorama and 3D understanding. In CW benchmarks, we release annotations of the 43 *known* classes for both training and testing. In OW benchmarks, methods can use *known* classes for training, while being tested on all of the classes.

Method	PQ↑	$\mathcal{O}_{\text{PS}}$↓	PQ$^{\text{Th}}$↑	$\mathcal{O}_{\text{PS}}^{\text{Th}}$↓	PQ$^{\text{St}}$↑	$\mathcal{O}_{\text{PS}}^{\text{St}}$↓
kMaX[37]	32.52	0.67	27.96	0.72	45.81	0.53
2Former[7]	33.25	0.66	28.74	0.71	46.38	0.52
DINO[18]	**36.57**	**0.64**	**33.07**	**0.68**	**46.74**	**0.51**

Table 3. **Results of CW panoptic segmentation methods on JRDB-PanoTrack.** All methods use ResNet-50 backbone and COCO pre-training. There are 43 classes, 32 *Thing* and 11 *Stuff*. (Kmax for Kmax-Deeplab, 2Former for Mask2Former and DINO for mask DINO.)

Metric. Current evaluation methods for panoptic segmentation, despite their utility, exhibit limitations that can skew method rankings, due to:

- Threshold-based, where the choice of the threshold can change the ranking of the methods, making it unreliable [26]: VPQ [15] and PTQ [14].
- Inadvertently penalizing the rectification of errors (ID recovery): VPQ [15] and PTQ [14].
- Inability to handle multi-label scenarios: VPQ [15], PTQ [14], and STQ [32].

Given the introduction of multi-label panoptic segmentation and tracking by JRDB-PanoTrack, these existing metrics become insufficient. To address the gaps, we introduce OSPA$_{PS}$ and OSPA$^2_{PT}$, specifically designed for panoptic segmentation and tracking.

OSPA for Panoptic Segmentation. The Optimal Sub-Pattern Matching (OSPA) metric, known for incorporating miss-distance in multi-object performance evaluation [29], has recently been adapted for bounding box/pose detection and tracking tasks [26, 30]. Building on this, we introduce OSPA$_{PS}$ ($\mathcal{O}_{PS}$), a variant of OSPA, specifically designed for multi-label panoptic segmentation.

Let $X = \{x_1, x_2, \ldots, x_m\}$ and $Y = \{y_1, y_2, \ldots, y_n\}$ be two sets of arbitrary mask regions ($x, y \subset \mathbb{R}^2$) on an image for all ground-truths and predictions, with cardinalities $|X|$ and $|Y|$, where $|Y| \geq |X|$ (otherwise flip X, Y). For a given class $c \in \mathbb{C}$, we calculate the normalised base distance between masks $d_K(x_i, y_i) = 1 - IOU(x_i, y_i) \in [0, 1]$. $\mathcal{O}_{PS}(X_c, Y_c)$ is then acquired by using OSPA equation [29]. The overall OSPA error is calculated by averaging the OSPA error over all classes:

$$\mathcal{O}_{PS}(X, Y) = \frac{1}{|\mathbb{C}|} \sum_{c \in \mathbb{C}} \mathcal{O}_{PS}(X_c, Y_c). \qquad (1)$$

OSPA for Panoptic Tracking. The OSPA metric is expanded to assess panoptic tracking with the introduction of $OSPA_{PT}$ ($\mathcal{O}^2_{PT}$). For each class $c \in \mathbb{C}$, consider $\mathbf{X}_c = \{X^{\mathcal{D}1}_{1c}, X^{\mathcal{D}2}_{2c}, \ldots, X^{\mathcal{D}m}_{mc}\}$ and $\mathbf{Y}_c = \{Y^{\mathcal{D}1}_{1c}, Y^{\mathcal{D}2}_{2c}, \ldots, Y^{\mathcal{D}n}_{nc}\}$ as sets of mask trajectories for ground-truth and predicted masks, respectively, where $\mathcal{D}i$

Method	PQ↑	$\mathcal{O}_{\text{PS}}$↓	PQ$^{\text{Th}}$↑	$\mathcal{O}_{\text{PS}}^{\text{Th}}$↓	PQ$^{\text{St}}$↑	$\mathcal{O}_{\text{PS}}^{\text{St}}$↓
ODISE-L[34]	10.57	**0.85**	7.03	0.90	**29.87**	**0.72**
ODISE-C[34]	**11.07**	**0.85**	**8.41**	**0.88**	25.55	0.78
FC-CLIP[38]	10.06	0.87	7.07	0.90	26.36	0.78

Table 4. **Results of SOTA OW panoptic segmentation models on JRDB-Panotrack testing set.** All models were trained solely on the COCO panoptic dataset and underwent zero-shot evaluation on JRDB. ODISE-L and ODISE-C represent the model with class label and caption label supervisions, respectively.

Train strategy		All		Thing		Stuff	
COCO	JRDB	PQ↑	$\mathcal{O}_{\text{PS}}$↓	PQ$^{\text{Th}}$↑	$\mathcal{O}_{\text{PS}}^{\text{Th}}$↓	PQ$^{\text{St}}$↑	$\mathcal{O}_{\text{PS}}^{\text{St}}$↓
	✓	31.41	0.67	27.12	0.72	43.88	0.53
✓	✓	**36.57**	**0.64**	**33.07**	**0.68**	**46.74**	**0.51**

Table 5. **CW panoptic segmentation results of MaskDino with different training strategies on JRDB-PanoTrack.** Top: we solely train the model on JRDB-PanoTrack. Bottom: we use COCO pertaining followed by finetuning on JRDB-PanoTrack.

Domain	Method	Known		Unknown	
		PQ↑	$\mathcal{O}_{\text{PS}}$↓	PQ↑	$\mathcal{O}_{\text{PS}}$↓
Cross	**ODISE-L**[34]	14.94	0.84	3.86	**0.92**
	ODISE-C[34]	13.58	0.84	**7.18**	**0.92**
	FC-CLIP[38]	14.29	0.86	3.56	0.93
In	**FC-CLIP**[38]	**24.95**	**0.75**	3.19	0.98

Table 6. **Performance of SOTA OW panoptic segmentation models on JRDB-PanoTrack.** Cross-domain methods are trained on COCO and tested on our dataset, while in-domain methods are trained with JRDB-PanoTrack *known* classes. ODISE-L and ODISE-C represent the model with class and caption supervisions, respectively.

contains the time indices where track i exists. Then, we calculate the time average distance of every pair of tracks $X_{ic}^{\mathcal{D}_i}$ and $Y_{jc}^{\mathcal{D}_j}$ similar to [29] using OSPA set distance $d_O(\{X_{ic}^t\}, \{Y_{jc}^t\} = 1 - IOU(x_{ic}^t, y_{jc}^t)$. If only $\{X_i^t\}$ or $\{Y_j^t\}$ exists, then $d_O(\{X_{ic}^t\}, \{Y_{jc}^t\}) = 1$, otherwise $d_O(\{X_{ic}^t\}, \{Y_{jc}^t\}) = 0$. The remaining step remained the same as the original OSPA^2, $\mathcal{O}_{PT}^2$ is the average of all classes similar to Eq. (1).

In JRDB-Panotrack, OSPA is preferred as it is an actual metric in mathematical terms, fulfilling the triangle inequality, not threshold-based, and treats masks equally regardless of their size, without penalising error rectification.

4. Experiments

To explore the distinct challenges of JRDB-PanoTrack, we first evaluate advanced panoptic segmentation in both 2D closed-world (CW) and open-world (OW) settings. Then we investigate panoptic tracking methods. We also briefly evaluate 3D CW segmentation and tracking using pseudo labels generated from 2D annotations. Note that all of the experiments presented are done using individual views, not stitched views. The results show that the JRDB-PanoTrack dataset provides a uniquely challenging environment for panoptic segmentation and tracking.

Evaluation protocol Due to the absence of pretrained models for close/open-world panoptic segmentation/tracking limits our evaluation in multi-label settings. We preprocess these areas into single-label ones, selecting *thing* objects and omitting *stuff* behind them, to utilize standard evaluation metrics alongside OSPA$_{PS}$ and OSPA$_{PT}^2$. We hope future research can exploit the full potential of our dataset's multi-class segmentation annotations.

4.1. Panoptic Segmentation

Implementation. We adopt the ResNet-50 backbone and COCO pertaining for all models, training with a batch size of 6 and a learning rate of 1×10^{-4}over 110K iterations on 2 RTX4090 GPUs. For other settings, we adhere to the default configuration in [7, 18, 37]. In OW experiments, we follow the official implementations of ODISE [34] and FC-CLIP [38]. For all cross-domain experiments, we use the weights pretrained on COCO and infer on JRDB-PanoTrack. For in-domain experiments, we train FC-CLIP on our OW training set and infer on our OW test set. The model is trained with two RTX4090 GPUs with batch size 8, learning rate 5×10^{-4}, other training setting use same as [38]. We do not train ODISE due to its very high computational costs.

CW Panoptic Segmentation. We evaluate SOTA methods on JRDB-PanoTrack (Tab. 3) and obtain the following findings: **(1)** Lower performance across all methods compared to COCO results, particularly for *Thing* classes. This highlights challenges like complex *Thing* instances and varied object scales in diverse environments. **(2)** Mask DINO stands out, which achieves PQ of 36.57% with COCO pertaining and 31.41% without it (Tab. 5). One reason is that our dataset contains crowded objects, and Mask DINO contains more object queries to capture a mass of object candidates. Meanwhile, MaskDino pretrained on JRDB-PanoTrack achieves higher performance on the COCO dataset (see the supplemental material), suggesting JRDB-PanoTrack's ability to generalize to other domains. These insights emphasize JRDB-PanoTrack's unique challenges in CW panoptic segmentation, leading us to further explore its role in OW panoptic segmentation setting.

OW Panoptic Segmentation. SOTA OW panoptic segmentation methods like FC-CLIP [38] and ODISE [34] show notably lower performance on JRDB-PanoTrack

	Trkr	STQ↑	Frag↓	IDF1↑	$\mathcal{O}^{2}_{\text{PT}}$↓	$\mathcal{O}^{2\text{T}}_{\text{PT}}$↑	$\mathcal{O}^{2\text{S}}_{\text{PT}}$↑
Kmax	**OS**	7.40	**4239**	28.70	**0.805**	**0.867**	**0.546**
	BT	8.77	6962	28.78	0.816	0.885	0.546
	BS	**14.20**	8932	**29.76**	0.831	0.909	0.546
2Former	**OS**	7.09	**4162**	27.49	**0.799**	**0.863**	**0.530**
	BT	8.58	7251	27.67	0.808	0.880	0.530
	BS	**13.89**	9392	**29.33**	0.817	0.902	0.530
DINO	**OS**	7.70	**4515**	30.40	**0.793**	**0.851**	**0.548**
	BT	9.00	7550	30.39	0.804	0.870	0.548
	BS	**14.50**	9609	**31.61**	0.822	0.901	0.548

Table 7. **2D CW panoptic tracking results.** Kmax for Kmax-Deeplab, 2Former for Mask2Former, DINO for mask DINO, OS for OC-SORT, BT for ByteTrack, and BS for BoT-SORT. $\mathcal{O}^{2\text{T}}_{\text{PT}}$ and $\mathcal{O}^{2\text{S}}_{\text{PT}}$ are the OSPA^2 metric for *Thing* and *Stuff* classes.

JRDB	COCO	PQ↑	PQTh ↑	PQSt ↑
	✓	42.16	47.43	34.20
✓	✓	**44.48**	**50.48**	**35.43**

Table 8. CW panoptic segmentation results on COCO val of Mask DINO[18] using different training data. Top: we solely train the model on COCO. Bottom: we use JRDB-PanoTrack pertaining followed by finetuning on COCO. JRDB refers to the JRDB-PanoTrack.

(Tab. 4) compared to other datasets. For instance, the PQ of FC-CLIP on ADE20K is 26.8 while 10.06 in our dataset, highlighting JRDB-PanoTrack's distinct and challenging nature, especially in recognizing and segmenting *Unknown* classes. Tab. 6 further shows cross- and in-domain evaluations for *Known* and *Unknown* classes on our dataset. Cross-domain results indicate that while prior knowledge from other datasets like COCO aids in understanding *Known* classes, it falls short with *Unknown* classes, underlining JRDB-PanoTrack's OW segmentation challenge. In contrast, in-domain training improves the segmentation performance for *Known* classes, but slightly impacts *Unknown* classes, suggesting that new approaches are needed to address this core challenge. Additionally, we assess the transferability of JRDB-PanoTrack knowledge to other domains. Training exclusively on JRDB-PanoTrack yields a 13.7 PQ on COCO (Tab. 11), demonstrating effective knowledge transfer to different domains. This finding indicates the potential usage of JRDB-PanoTrack to improve segmentation performance on other domains.

Generalizability of JRDB-Panotrack Tab. 8 presents comparative results of the Mask DINO model on the COCO validation set, highlighting the generalizability of JRDB-Panotrack. Specifically, Tab. 8 compares the performance when trained solely with the COCO dataset against a combined training scheme that includes both JRDB-PanoTrack and COCO datasets. Notably, the model pretrained on JRDB-PanoTrack followed by COCO tuning shows superior performance across all metrics compared to the model trained on COCO alone, which supporting JRDB-Panotrack is also beneficial for other domains.

Method	$\mathcal{O}_{\text{PS}}$↓	$\mathcal{O}^{\text{Small}}_{\text{PS}}$ ↓	$\mathcal{O}^{\text{Medium}}_{\text{PS}}$ ↓	$\mathcal{O}^{\text{Large}}_{\text{PS}}$ ↓
Kmax[37]	0.670	0.823	0.596	0.370
Mask2Former[7]	0.655	0.805	0.589	0.371
MaskDINO[18]	**0.636**	**0.785**	**0.552**	**0.364**

Table 9. CW panoptic segmentation results on JRDB-PanoTrack testing for objects of different scales.

Method	$\mathcal{O}_{\text{PS}}$↓	$\mathcal{O}^{\text{Small}}_{\text{PS}}$ ↓	$\mathcal{O}^{\text{Medium}}_{\text{PS}}$ ↓	$\mathcal{O}^{\text{Large}}_{\text{PS}}$ ↓
ODISE-L[34]	0.851	0.950	0.790	0.550
ODISE-C[34]	0.849	0.906	0.696	0.431
FC_CLIP[38]	0.868	0.968	0.863	0.619
FC_CLIP+[38]	**0.776**	**0.877**	**0.597**	**0.390**

Table 10. OW panoptic segmentation results on JRDB-PanoTrack testing for objects of different scales.

Knowledge transfer Tab. 8 demonstrates knowledge transferability between datasets in open-world setting. In the closed-world setting, we also show that the knowledge from JRDB-PanoTrack can help improve performance when fine-tuning on the COCO dataset (Tab. 11) and vice versa (Tab. 5). The results in both open-world and closed-world settings show that using JRDB-PanoTrack improves segmentation performance in other domains. This suggests that the size of JRDB-PanoTrack does not significantly hinder the performance.

Train data	PQ
COCO	10.06
JRDB	13.70

Table 11. **Cross-dataset validation results of FC-CLIP.** The first row indicates training on COCO and testing on JRDB-PanoTrack, while the second row is the opposite.

4.2. Panoptic Tracking

Implementation. We utilize default settings and implementations of recent popular tracking algorithms: ByteTrack[40], OC-SORT[5] and BoT-SORT[1]. Masks predicted from CW and OW segmentation models are converted into bounding boxes and then fed into these trackers.

CW Panoptic Tracking. In Tab. 7, our evaluation highlights diverse capabilities of SOTA tracking methods. BoT-SORT excels in STQ and IDF1 metrics, showcasing its proficiency in object tracking and identity maintenance.

	Trkr	STQ↑	Frag↓	IDF1↑	$\mathcal{O}_{\mathbf{PT}}^{\mathbf{2}}$↓	$\mathcal{O}_{\mathbf{PT}}^{\mathbf{2K}}$↓	$\mathcal{O}_{\mathbf{PT}}^{\mathbf{2U}}$↓
FC_CLIP	**OS**	2.50	**833**	8.43	**0.910**	**0.861**	**0.962**
	BT	2.78	1126	8.82	0.921	0.871	0.971
	BS	**4.71**	1956	**9.52**	0.921	0.869	0.979
ODISE-L	**OS**	3.11	**1373**	8.89	**0.924**	**0.854**	**0.977**
	BT	3.71	2013	9.31	0.928	0.867	0.979
	BS	**6.49**	3167	**9.97**	0.927	0.873	0.980
ODISE-C	**OS**	4.19	**1112**	9.22	**0.917**	**0.862**	0.979
	BT	5.07	1457	9.34	0.925	0.866	**0.978**
	BS	**8.32**	2139	**10.80**	0.924	0.863	0.980
FC_CLIP+	**OS**	4.90	**2833**	13.20	**0.897**	**0.826**	**0.990**
	BT	5.53	4614	13.44	0.905	0.836	0.993
	BS	**9.01**	6868	**14.51**	0.908	0.850	0.993

Table 12. **2D OW panoptic tracking results.** (OS for OC-SORT, BT for ByteTrack and BS for BoT-SORT). $\mathcal{O}_{\mathbf{PT}}^{\mathbf{2K}}$ and $\mathcal{O}_{\mathbf{PT}}^{\mathbf{2U}}$ are the OSPA^2 metric for *Known* and *Unknown* classes. ODISE-L and ODISE-C represent ODISE using class names and captions as supervison.

OC-SORT, favored by the $\mathcal{O}_{PT}^{2}$ metric, excels in consistently identifying objects across frames while minimizing noisy tracklets. BoT-SORT's performance, though strong in tracking objects, shows signs of instability, often losing track and struggling with consistent ID maintenance. Breaking down into *Thing* and *Stuff* classes, $\mathcal{O}_{PT}^{2S}$ error for *Stuff* remains constant because cardinality error is not penalised. The lower performance on our JRDB-Panotrack, compared to other datasets, can be attributed to our dense annotations and numerous tracklets, posing a significant challenge for segmentation and tracking.

OW Panoptic Tracking. OW panoptic tracking results, as shown in Tab. 12, indicate a different set of challenges. While BoT-SORT is good at maintaining object identities and delivering high-quality segmentation, it exhibits higher fragmentation, indicating inconsistency in track identity over time. In contrast, OC-SORT, though it may not always top the STQ or IDF1 scores, shows greater consistency with fewer fragmentations and lower OSPA errors. The overall lower performance on the JRDB-Panotrack dataset reflects the complexities of OW tracking, especially when handling *unknown* objects. This underscores the need for advanced tracking algorithms to adapt to unfamiliar objects and maintain consistent track identities.

3D panoptic segmentation

Method	IoU↑	PQ↑	RQ↑	$\mathcal{O}_{\mathbf{PT}}^{\mathbf{2}}$↓	$\mathcal{O}_{Card}^{\mathbf{2}}$↓	$\mathcal{O}_{Loc}^{\mathbf{2}}$↓
DSNet[13]	12.62	3.41	4.25	0.843	0.657	0.186
Mask4D[36]	13.51	3.57	5.39	0.826	0.643	0.183
MaskPLS[22]	**15.13**	**7.02**	**10.74**	**0.795**	**0.629**	**0.166**

3D Panoptic Tracking

Method	LSTQ↑	S$_{assoc}$↑	S$_{cls}$↑	$\mathcal{O}_{\mathbf{PT}}^{\mathbf{2}}$↓	$\mathcal{O}_{Card}^{\mathbf{2}}$↓	$\mathcal{O}_{Loc}^{\mathbf{2}}$↓
DSNet[13]	25.35	55.18	11.64	0.882	0.726	0.156
Mask4D[36]	**27.87**	**66.32**	**11.71**	**0.860**	**0.711**	**0.149**

Table 13. Results for 3D panoptic segmentation and tracking on JRDB-PanoTrack testing. S_{assoc} and S_{cls} are association and classification scores (components of LSTQ), respectively. $\mathcal{O}_{Card}^{\mathbf{2}}$and $\mathcal{O}_{Loc}^{\mathbf{2}}$are OSPA cardinality and localisation errors (components of $\mathcal{O}_{\mathbf{PT}}^{\mathbf{2}}$), as explained in [29].

4.3. 3D Panoptic Segmentation & Tracking

In this work, we briefly touch on 3D CW panoptic segmentation and tracking, though it is not the main focus of this paper. Specifically, we projected 2D panoptic labels onto 3D point clouds, using these projections as pseudo-labels for model training. For evaluation, we use our proposed OSPA, OSPA^2 and adopt popular metrics for 3D panoptic segmentation (PQ, IOU) and tracking (LSTQ[2]). It's important to note that these 3D pseudo-labels may contain noise, potentially affecting result accuracy. In terms of 3D panoptic segmentation, as shown in Tab. 13, MaskPLS emerges as the superior method, excelling in all metrics. This indicates MaskPLS's enhanced ability to identify and segment objects precisely in 3D space. In 3D panoptic tracking, Mask4D takes the lead in LSTQ[2] and achieves the best OSPA score of 0.860, denoting its strength in maintaining object identities and tracking consistency over time. Also, the higher S_{assoc} scores compared to S_{cls}, suggesting that these methods are better at object association and tracking than at precise classification in a 3D environment.

5. Conclusion

In this paper, we have introduced the *JRDB-PanoTrack* dataset, a novel dataset designed for open-world panoptic segmentation and tracking, particularly for robotics and vision applications. The uniqueness and complexity of JRDB-PanoTrack set it apart from the existing ones. Our extensive evaluations underscore the dataset's challenges, emphasizing the necessity for more robust methodologies in both closed-world and open-world scenarios. The dataset offers new ground for future research, especially in developing algorithms that can effectively handle densely populated environments and diverse object interactions that are typical in real-world settings.

Acknowledgments. This work has been partially funded by The Australian Research Council Discovery Project (ARC DP2020102427). We also acknowledge the partial sponsorship of our research by the DARPA Assured Neuro Symbolic Learning and Reasoning (ANSR) program, under award number FA8750-23-2-1016.

References

[1] Nir Aharon, Roy Orfaig, and Ben-Zion Bobrovsky. Bot-sort: Robust associations multi-pedestrian tracking. *arXiv preprint arXiv:2206.14651*, 2022. 3, 7

[2] Mehmet Aygun, Aljosa Osep, Mark Weber, Maxim Maximov, Cyrill Stachniss, Jens Behley, and Laura Leal-Taixé. 4d panoptic lidar segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5527–5537, 2021. 8

[3] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, and Jurgen Gall. Semantickitti: A dataset for semantic scene understanding of lidar sequences. In *ICCV*, pages 9297–9307, 2019. 2, 3

[4] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 3464–3468, 2016. 3

[5] Jinkun Cao, Jiangmiao Pang, Xinshuo Weng, Rawal Khirodkar, and Kris Kitani. Observation-centric sort: Rethinking sort for robust multi-object tracking. In *CVPR*, pages 9686–9696, 2023. 3, 7

[6] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *NeurIPS*, 34:17864–17875, 2021. 3

[7] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention Mask Transformer for Universal Image Segmentation. 2022. 5, 6, 7

[8] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, pages 1290–1299, 2022. 3

[9] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, pages 3213–3223, 2016. 2

[10] Zheng Ding, Jieke Wang, and Zhuowen Tu. Open-vocabulary panoptic segmentation with maskclip. *arXiv preprint arXiv:2208.08984*, 2022. 3

[11] Mahsa Ehsanpour, Fatemeh Saleh, Silvio Savarese, Ian Reid, and Hamid Rezatofighi. Jrdb-act: A large-scale dataset for spatio-temporal action, social group and activity detection. In *CVPR*, 2022. 2, 3

[12] Whye Kit Fong, Rohit Mohan, Juana Valeria Hurtado, Lubing Zhou, Holger Caesar, Oscar Beijbom, and Abhinav Valada. Panoptic nuscenes: A large-scale benchmark for lidar panoptic segmentation and tracking. *RA-L*, 7(2):3795–3802, 2022. 2, 3

[13] Fangzhou Hong, Hui Zhou, Xinge Zhu, Hongsheng Li, and Ziwei Liu. Lidar-based panoptic segmentation via dynamic shifting network. In *CVPR*, pages 13090–13099, 2021. 8

[14] Juana Valeria Hurtado, Rohit Mohan, Wolfram Burgard, and Abhinav Valada. Mopt: Multi-object panoptic tracking. *arXiv preprint arXiv:2004.08189*, 2020. 4, 5

[15] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Video panoptic segmentation. In *CVPR*, pages 9859–9868, 2020. 3, 5

[16] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *CVPR*, pages 6399–6408, 2019. 3

[17] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *CVPR*, pages 9404–9413, 2019. 2, 3, 4

[18] Feng Li, Hao Zhang, Huaizhe Xu, Shilong Liu, Lei Zhang, Lionel M Ni, and Heung-Yeung Shum. Mask dino: Towards a unified transformer-based framework for object detection and segmentation. In *CVPR*, pages 3041–3050, 2023. 3, 5, 6, 7

[19] Siyuan Li, Tobias Fischer, Lei Ke, Henghui Ding, Martin Danelljan, and Fisher Yu. Ovtrack: Open-vocabulary multiple object tracking. In *CVPR*, pages 5567–5577, 2023. 3

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the ECCV*, pages 740–755. Springer, 2014. 2, 3

[21] Yang Liu, Idil Esen Zulfikar, Jonathon Luiten, Achal Dave, Deva Ramanan, Bastian Leibe, Aljoša Ošep, and Laura Leal-Taixé. Opening up open world tracking. In *CVPR*, pages 19045–19055, 2022. 3

[22] Rodrigo Marcuzzi, Lucas Nunes, Louis Wiesmann, Jens Behley, and Cyrill Stachniss. Mask-based panoptic lidar segmentation for autonomous driving. *RA-L*, 8(2):1141–1148, 2023. 8

[23] Roberto Martin-Martin, Mihir Patel, Hamid Rezatofighi, Abhijeet Shenoi, JunYoung Gwak, Eric Frankel, Amir Sadeghian, and Silvio Savarese. Jrdb: A dataset and benchmark of egocentric robot visual perception of humans in built environments. *TPAMI*, 2021. 2, 3

[24] Jieru Mei, Alex Zihao Zhu, Xinchen Yan, Hang Yan, Siyuan Qiao, Liang-Chieh Chen, and Henrik Kretzschmar. Waymo open dataset: Panoramic video panoptic segmentation. In *ECCV*, pages 53–72. Springer, 2022. 2, 3

[25] Jiaxu Miao, Xiaohan Wang, Yu Wu, Wei Li, Xu Zhang, Yunchao Wei, and Yi Yang. Large-scale video panoptic segmentation in the wild: A benchmark. In *CVPR*, pages 21033–21043, 2022. 2, 3

[26] Tran Thien Dat Nguyen, Hamid Rezatofighi, Ba-Ngu Vo, Ba-Tuong Vo, Silvio Savarese, and Ian Reid. How trustworthy are performance evaluations for basic vision tasks? *TPAMI*, 2022. 5

[27] Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, Thomas Funkhouser, et al. Openscene: 3d scene understanding with open vocabularies. In *CVPR*, pages 815–824, 2023. 3

[28] Jie Qin, Jie Wu, Pengxiang Yan, Ming Li, Ren Yuxi, Xuefeng Xiao, Yitong Wang, Rui Wang, Shilei Wen, Xin Pan, et al. Freeseg: Unified, universal and open-vocabulary image segmentation. In *CVPR*, 2023. 3

[29] Dominic Schuhmacher, Ba-Tuong Vo, and Ba-Ngu Vo. A consistent metric for performance evaluation of multi-object filters. *IEEE transactions on signal processing*, 56(8):3447–3457, 2008. 5, 6, 8

[30] Edward Vendrow, Duy Tho Le, Jianfei Cai, and Hamid Rezatofighi. Jrdb-pose: A large-scale dataset for multi-person pose estimation and tracking. In *CVPR*, pages 4811–4820, 2023. 2, 3, 5

[31] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Max-deeplab: End-to-end panoptic segmentation with mask transformers. In *CVPR*, pages 5463–5474, 2021. 3

[32] Mark Weber, Jun Xie, Maxwell D Collins, Yukun Zhu, Paul Voigtlaender, Hartwig Adam, Bradley Green, Andreas Geiger, Bastian Leibe, Daniel Cremers, et al. Step: Segmenting and tracking every pixel. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021. 2, 3, 5

[33] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun. Upsnet: A unified panoptic segmentation network. In *CVPR*, pages 8818–8826, 2019. 3

[34] Jiarui Xu, Sifei Liu, Arash Vahdat, Wonmin Byeon, Xiaolong Wang, and Shalini De Mello. Open-vocabulary panoptic segmentation with text-to-image diffusion models. In *CVPR*, 2023. 3, 6, 7

[35] Xin Xu, Tianyi Xiong, Zheng Ding, and Zhuowen Tu. Masqclip for open-vocabulary universal image segmentation. In *ICCV*, pages 887–898, 2023. 3

[36] Kadir Yilmaz, Jonas Schult, Alexey Nekrasov, and Bastian Leibe. Mask4d: Mask transformer for 4d panoptic segmentation. *arXiv preprint arXiv:2309.16133*, 2023. 8

[37] Qihang Yu, Huiyu Wang, Siyuan Qiao, Maxwell Collins, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. k-means mask transformer. In *ECCV*, pages 288–307. Springer, 2022. 5, 6, 7

[38] Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Convolutions die hard: Open-vocabulary segmentation with single frozen convolutional clip. *arXiv preprint arXiv:2308.02487*, 2023. 6, 7

[39] Wenwei Zhang, Jiangmiao Pang, Kai Chen, and Chen Change Loy. K-net: Towards unified image segmentation. *NeurIPS*, 34:10326–10338, 2021. 3

[40] Yifu Zhang, Peize Sun, Yi Jiang, Dongdong Yu, Fucheng Weng, Zehuan Yuan, Ping Luo, Wenyu Liu, and Xinggang Wang. Bytetrack: Multi-object tracking by associating every detection box. In *ECCV*, pages 1–21. Springer, 2022. 3, 7

[41] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017. 2, 3