OneBEV: Using One Panoramic Image for Bird's-Eye-View Semantic Mapping

Jiale Wei[0009−0000−3064−3961], Junwei Zheng[0009−0005−4390−3044], Ruiping Liu[0000−0001−5245−2277], Jie Hu[0009−0002−8917−4703], Jiaming Zhang*[0000−00033471−328X], and Rainer Stiefelhagen[0000−0001−8046−4945]

Karlsruhe Institute of Technology, Germany
https://github.com/JialeWei/OneBEV

Abstract. In the field of autonomous driving, Bird's-Eye-View (BEV) perception has attracted increasing attention in the community since it provides more comprehensive information compared with pinhole front-view images and panoramas. Traditional BEV methods, which rely on multiple narrow-field cameras and complex pose estimations, often face calibration and synchronization issues. To break the wall of the aforementioned challenges, in this work, we introduce OneBEV, a novel BEV semantic mapping approach using merely a single panoramic image as input, simplifying the mapping process and reducing computational complexity. A distortion-aware module termed Mamba View Transformation (MVT) is specifically designed to handle the spatial distortions in panoramas, transforming front-view features into BEV features without leveraging traditional attention mechanisms. Apart from the efficient framework, we contribute two datasets, *i.e.*, nuScenes-360 and DeepAccident-360, tailored for the OneBEV task. Experimental results showcase that OneBEV achieves state-of-the-art performance with 51.1% and 36.1% mIoU on nuScenes-360 and DeepAccident-360, respectively. This work advances BEV semantic mapping in autonomous driving, paving the way for more advanced and reliable autonomous systems.

Keywords: BEV Mapping · Panoramic Images · Mamba

1 Introduction

Autonomous driving solutions that rely solely on cameras are becoming mainstream and showing promising results [24]. The bird's-eye view is a natural and straightforward candidate to serve as a unified representation [12]. Since bird's-eye view provides holistic information of locations and scale of objects, it is widely used in various autonomous driving scenarios such as perception and planning [12, 14]. Current BEV techniques [2, 5, 17, 22, 29] rely on multiple narrow-field cameras to capture different views, then transform these front-view

* Corresponding: jiaming.zhang@kit.edu

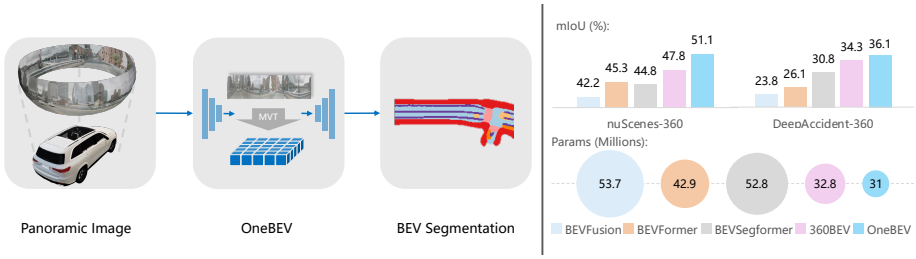


Fig. 1: OneBEV only requires one panoramic image for BEV semantic mapping, decoupling the specifications of the camera, and can be more flexibly applied to different autonomous vehicle platforms. Achieving higher performance (mIoU) with fewer parameters on both nuScenes-360 and DeepAccident-360 shows the efficacy of OneBEV.

perspectives into a top-view and combine them for a comprehensive BEV representation. Nevertheless, these methods are constrained by their reliance on intrinsic and extrinsic camera parameters for pose estimation and view mapping. Calibration imperfections, synchronization difficulties, environmental conditions can cause discontinuity and inconsistency in scene perception, resulting in poor BEV features and unsatisfactory performance. Besides, the previous methods [2, 5, 17, 22, 29] often suffer from higher computational complexity due to the requirement of processing data from multiple viewpoints in a scene.

To address the above issues, we propose **OneBEV**, a novel task to utilize **One** panorama for **BEV** semantic mapping in outdoor scenes. As shown in Fig. 1, this paradigm requires a single 360° image as input and gives a dense semantic map as output. The benefits of OneBEV include: (1) consistent omnidirectional perception, (2) simplified camera setup, (3) no need for camera synchronization, and (4) lower computational complexity.

However, the use of panoramas is always accompanied by spatial distortion and object deformation [34]. In order to solve this problem, we introduce a novel module Mamba View Transformation (MVT), a key component of OneBEV model. Inspired by the recent success of Mamba [6, 16, 36] in text sequence and image modeling, we incorporate the scanning mechanism from Mamba to tackle the transformation from front-view panoramic perspective to BEV. Additionally, MVT integrates deformability capabilities, enhancing its suitability for overcoming distortions compared to using pure Mamba. Building on these innovations, the OneBEV architecture consists of three main components: the feature encoder, the view transformation module, and the semantic decoder. We use the VMamba-T [16] model as the backbone. The MVT module then selects and transforms these features into BEV space. Finally, a lightweight semantic decoder processes BEV features to generate the final semantic segmentation map.

Furthermore, the datasets used are crucial for the success of OneBEV. But no dataset is available in the literature for OneBEV task. In order to overcome the lack of training data, we introduce two OneBEV datasets, nuScenes-360 and DeepAccident-360. These datasets are expansions of the well-known nuScenes [3] and DeepAccident [30] datasets, which are highly regarded in the field of au-

tonomous driving. The nuScenes dataset provides diverse urban driving data, including images from six views and the semantic maps. We carefully combine images from the six perspectives into a unified panoramic image, aligning them using intrinsic and extrinsic camera parameters (these parameters are only used for dataset construction, not for training). Similarly, the DeepAccident dataset, generated with the CARLA [4] simulator, offers synthetic yet realistic accident scenarios. We apply the consistent process to create panoramic images from its multi-view setup. In the end, for nuScenes-360 and DeepAccident-360, we contribute 34K and 48K panoramas with annotations, respectively.

Extensive experiments prove that OneBEV achieves state-of-the-art performance with 51.1% and 36.1% mIoU on nuScenes-360 and DeepAccident-360, respectively, and has 1.8 million fewer parameters than the baseline. OneBEV provides a practical solution for semantic scene understanding in dynamic outdoor environments by simplifying the process and minimizing computational complexity. This progress is noteworthy for the development of more efficient and dependable autonomous driving systems.

To summarize, our contributions can be outlined as follows:

- We introduce a new **OneBEV** paradigm, performing Bird's-Eye-View semantic mapping by using one 360° image.
- To address data scarcity, we introduce **nuScenes-360** and **DeepAccident-360** datasets for the first time. We benchmark OneBEV by comparing multi-view methods on both real and synthetic datasets.
- A new **Mamba View Transformation (MVT)** module is proposed to achieve efficient BEV transformation by using State Space Model (SSM) instead of cross-attention mechanism.
- Extensive experiments and ablation studies prove the effectiveness of OneBEV paradigm and architecture, paving the way for new BEV semantic mapping.

2 Related Work

2.1 State Space Models

State Space Models (SSMs) [7], like Transformers and RNNs, are used to process sequential data such as text, signals, etc. Initially introduced in the S4 [7] model, SSMs offer a unique architecture capable of efficiently modeling global information. Based on S4, Mamba [6] enhances SSMs by adding a selection mechanism for time-varying parameters and hardware optimization to achieve efficient training and inference. Subsequently, Vim [36] and VMamba [16] extend Mamba to visual tasks by proposing bidirectional scanning and cross-scanning mechanisms to achieve location-aware visual understanding. Pan-Mamba [9] and Sigma [28] solve more challenging multi-modal learning tasks. In our work, we explore the ability of Mamba to extract features as a visual backbone and how to apply Mamba to the task of view transformation, which poses new challenges to the Mamba model in solving complex visual tasks.

2.2 Panoramic Semantic Segmentation

In recent years, panoramic semantic segmentation has garnered increasing attention [11, 21, 32–34]. Unlike narrow-view pinhole camera images, panoramic images are typically obtained using equirectangular projection, which causes distortion and warping of objects in the image, posing challenges for semantic segmentation. In our work, we need to select appropriate datasets for our novel task. Datasets [15, 19] focused on panoramic image perception tasks only contain semantic labels for front-view images and lack semantic labels from a BEV perspective. Moreover, panoramic images often lack camera parameters, making it impossible to convert front-view semantic labels to BEV perspective using homography transformation. Datasets that include multi-view pinhole camera images provide BEV perspective semantic labels, but the multi-view front images require additional processing to be stitched into panoramic images. This is the case with the nuScenes [3] and DeepAccident [30] datasets used in our work. After our data processing, we contribute 34K and 48K panoramas with annotations, respectively.

2.3 BEV Semantic Mapping

For BEV semantic segmentation tasks, view transformation plays a core role [12]. It is mainly divided into two methods [12]: *2D-3D* and *3D-2D*. The *2D-3D* method was first introduced by LSS [23], which estimates implicit pixel depth information from 2D features and then uses camera geometry to establish the connection between BEV segmentation and feature maps in 3D space. Additionally, building on the work of LSS, [10,17,31] have explored better depth estimation methods. The *3D-2D* method was proposed many years ago by IPM [20], which uses homography transformation to convert camera images into BEV. Currently, mainstream works [5, 14, 26, 29, 35] employ multi-layer perceptron (MLP) or Transformer architectures to perform 3D-to-2D projection using camera parameters. BEVSegformer [22] completes the view projection without using camera parameters. In terms of panoramic views, 360BEV [27] was the first to apply the Transformer architecture to achieve view transformation using supervised ground-truth depth information. Based on 360BEV, we innovatively explore outdoor scenarios, and depth information doesn't included.

3 OneBEV

3.1 Task Definition

OneBEV is aim to create a BEV semantic map using only one panoramic image. This task streamlines the process of top-down perspective analysis across different domains, providing thorough spatial understanding using just one 360° image. To adapt to our new panoramic task, we combine together the multi-view images in the existing datasets [3, 30] to obtain the omni-directional panorama, which will be used as the **input** of the neural network. The high-definition semantic map in the dataset will be used as the **label** for network training.

3.2 Datasets

Data Processing. Based on the panoramic generation from [27], we project multi-view images onto a sphere to obtain the panoramic image. The Fig. 2(a) depicts the multi-camera setup on a vehicle in the nuScenes [3] dataset, where six cameras are mounted on the car roof, with each camera rotated at an interval of 55°. All cameras possess a 70° field of view (FoV) except for *Camera_Back*, which has a 110° FoV. Unlike indoor dataset setups in [27], where cameras are mounted on a tripod and rotate around a single center in an indoor environment, the cameras in the autonomous driving dataset are mounted at different positions around the car. Thus, as illustrated in Fig. 2(a)(b), the optical centers C_i ($i \in [1,6]$) of all cameras are distributed, indicating that each camera's imaging plane I_i is tangential to spheres S_i with different radii R_i, all centered around O, which is the average optical center of all cameras.

In order to accurately stitch panoramic images, we begin by calculating the fundamental parameters for each camera (see Fig. 2(b)). Initially, we determine the distance r_i from the optical center C_i of each camera to its imaging plane I_i by utilizing the image width W and FoV. Next, we calculate Δ_i, which represents the displacement of C_i from O. These displacements, Δ_{i_x} and Δ_{i_y}, represent the shifts along the x-axis and y-axis, respectively. These values are converted from millimeters into pixel coordinates using the intrinsic camera parameters $\mathbf{K}$, which include the focal length f_x and f_y. The following equations are used to convert these physical displacements into pixel units:

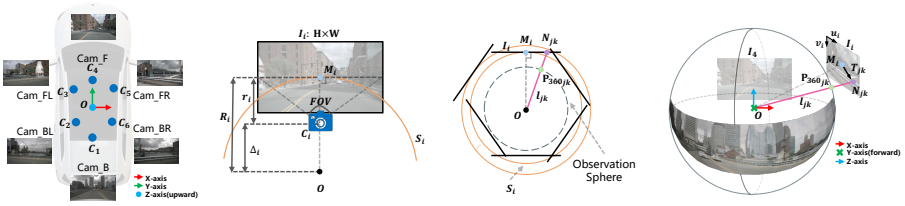
$$\begin{aligned}
\Delta_{i_u}[pixel] &= \Delta_{i_x}[mm] * pixel_size = \Delta_{i_x}[mm]\frac{f_x[pixel]}{f[mm]}, \\
\Delta_{i_v}[pixel] &= \Delta_{i_y}[mm] * pixel_size = \Delta_{i_y}[mm]\frac{f_y[pixel]}{f[mm]}, \\
\Delta_i[pixel] &= \sqrt{((C_{i_x} - O_x)\frac{f_x}{f})^2 + ((C_{i_y} - O_y)\frac{f_y}{f})^2}.
\end{aligned} \tag{1}$$

The overall displacement $\Delta_i$is computed as:

$$R_i = r_i + \Delta_i. \tag{2}$$

As illustrated in Fig. 2(d), the final panoramic image is projected onto a observation sphere, which, when unfolded, yields a 2D panoramic image $f_{360} \in \mathbb{R}^{H_{360} \times W_{360} \times 3}$. Each pixel $P_{360_{jk}}$ ($j \in [1,H_{360}], k \in [1,W_{360}]$) in the 2D panoramic image is mapped horizontally ±180° and vertically ±25° based on the FoV. This mapping allows us to determine the angular deviations Δu_{jk} and Δv_{jk} between each pixel and the center of the 2D image. By utilizing the radius R_i and angular deviations, we determine two crucial points M_i and N_{jk} during the stitching procedure (see Fig. 2(c)(d)). These points are used to establish the mapping between points in the panoramic image and points on the camera's image plane.

M_i: Each camera's image plane I_i is tangent to its corresponding sphere S_i at point M_i. Therefore, the image plane I_i can be mathematically represented



(a) Multi-camera setup in the nuScenes dataset.

(b) The optical center of the camera C_i is positioned at a distance Δ_i from the center O.

(c) The ray l_{jk} originating from center O passing through point P_{360jk} in the panoramic image and directed towards point N_{jk} on the camera's image plane I_i.

(d) Mapping relationship between points in the panoramic image and points on the camera's image plane in a 3D perspective.

Fig. 2: Data processing of nuScenes-360 and DeepAccident-360 dataset. (a) The original multi-view setup requires complex camera configuration and higher computational complexity. (b) Given the front view as an example, the camera parameters are visualized in the original image plane. (c) The panoramic image is calculated via the observation sphere according to the camera image plane. (d) The mapping relationship between narrow-FoV and omni-FoV images.

by Eq. (3). First, we compute the orientation offset $[\Delta yaw_i, \Delta pitch_i, \Delta roll_i]$ between the camera C_i and the reference camera C_4 using the camera extrinsic parameters. Then, we transform this offset from Euler angles to the Cartesian coordinates of point M_i on the surface of the observation sphere using Eq. (4):

$$\begin{gathered} M_{i_x}(x - M_{i_x}) + M_{i_y}(y - M_{i_y}) + M_{i_z}(z - M_{i_z}) = 0, \\ M_{i_x}^2 + M_{i_y}^2 + M_{i_z}^2 = R_i^2, \end{gathered} \tag{3}$$

$$M_i = \begin{bmatrix} M_{i_x} \\ M_{i_y} \\ M_{i_z} \end{bmatrix} = \begin{bmatrix} R_i * \cos(\Delta pitch_i) * \sin(\Delta yaw_i) \\ R_i * \cos(\Delta pitch_i) * \cos(\Delta yaw_i) \\ R_i * \sin(\Delta pitch_i) \end{bmatrix}. \tag{4}$$

N_{jk}: The view line l_{jk} is represented by a ray that originates from the center of the observation sphere O, passes through a point $P_{360_{jk}}$ in the panoramic image, and intersects the camera's image plane I_i at point N_{jk}. Eq. (5) describes the formula of the view line:

$$\begin{aligned} x &= \alpha_{jk} t, \\ y &= \beta_{jk} t, \\ z &= \gamma_{jk} t. \end{aligned} \tag{5}$$

where x, y, and z are proportional to t with proportionality constants α_{jk}, β_{jk}, and γ_{jk}. We solve the equations of the image plane (in Eq. (3)) and the view line (in Eq. (5)) by utilizing the angular deviations Δu_{jk} and Δv_{jk} in order to determine N_{jk} using Eq. (6) (7) (8):

$$\begin{bmatrix} \alpha_{jk} \\ \beta_{jk} \\ \gamma_{jk} \end{bmatrix} = \begin{bmatrix} \cos(\Delta v_{jk}) \sin(\Delta u_{jk}) \\ \cos(\Delta v_{jk}) \cos(\Delta u_{jk}) \\ \sin(\Delta v_{jk}) \end{bmatrix}. \tag{6}$$

$$t = \frac{R_i^2}{M_{i_x}\alpha_{jk} + M_{i_y}\beta_{jk} + M_{i_y}\gamma_{jk}}. \tag{7}$$

$$N_{jk} = \begin{bmatrix} N_{jk_x} \\ N_{jk_y} \\ N_{jk_z} \end{bmatrix} = \begin{bmatrix} \alpha_{jk}t \\ \beta_{jk}t \\ \gamma_{jk}t \end{bmatrix}. \tag{8}$$

The coordinates of N_{jk}, initially calculated in the observation sphere's coordinate system, must be transformed into the image plane's coordinate system. We know that M_i lies on the image plane. The vector from M_i to N_{jk} is denoted as T_{jk}. We first define the unit vector in the positive u and v directions:

$$\begin{aligned} u_i &= [\cos(\Delta yaw_i), -\sin(\Delta yaw_i), 0], \\ v_i &= M_i \times u_i. \end{aligned} \tag{9}$$

By projecting the vector T_{jk} onto the vectors u_i and v_i, we can determine the displacement of N_{jk} with respect to M_i on the image plane I_i. This displacement is represented by the values Δu_{jk} and Δv_{jk}.

These displacements allow us to locate N_{jk} on the image plane, thereby determining the corresponding pixel in the original camera image. If the displacement values fall outside the image boundaries, the point is considered invalid. By repeating this process for every point in the panoramic image, we construct the final panorama with contributions from all cameras, ensuring complete coverage. The example of multi-view images and the panoramic image are shown in Fig. 3.

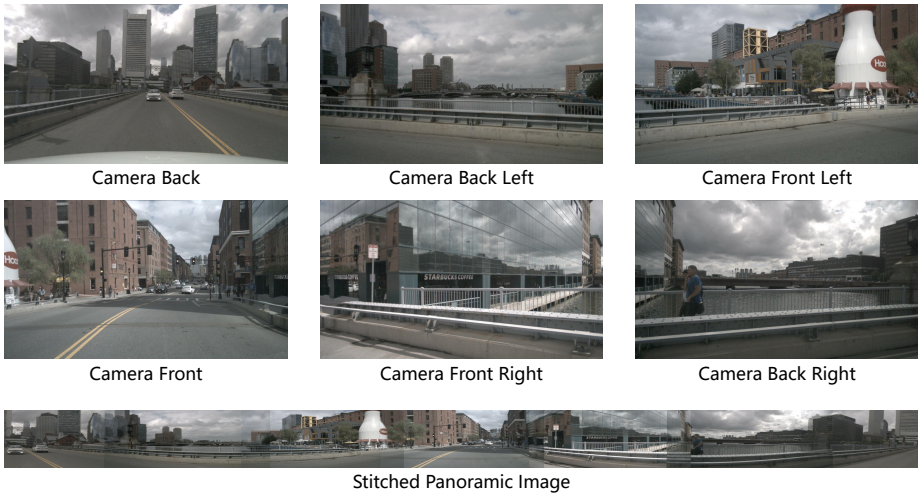


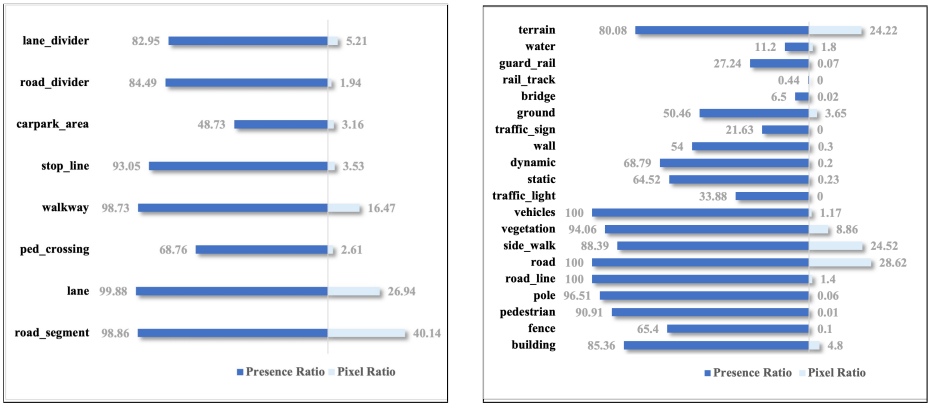
Fig. 3: Visualization of data samples. The first two rows are the multi-view images from the original nuScenes dataset, while the last row is the reconstructed panorama.

NuScenes-360 and DeepAccident-360 Dataset. The original nuScenes [3] dataset is a large-scale autonomous driving dataset. It includes 6 narrow-view images, semantic maps, and other features collected during driving. We combine the multi-view images using the camera parameters provided in the dataset and extracted the required classes from the semantic maps as labels, forming the new nuScenes-360 dataset. To explore different scenarios, we also introduce another synthetic dataset DeepAccident [30], which is generated from CARLA [4] simulator. DeepAccident has a similar structure to nuScenes, thus we employ the same generation to create new DeepAccident-360 dataset.

OneBEV Challenges. Since the overlapping regions between multiple cameras in datasets are limited, there are cases where the same object does not appear in adjacent images. Therefore, using feature points based methods [1, 18, 25] for matching can be slow or even fail. Our stitching method based on the physical model of the camera, which, although fast, does not achieve high stitching quality. Additionally, objects in the panoramic image may become distorted or warped, presenting challenges to the robustness of the neural network. Fig. 4 illustrates some of these challenges.

Fig. 4: Challenges in panoramic images. For example, a straight road appears curved and incomplete in the image (blue shade), distributed across different positions. Due to stitching quality issues, seams and ghosting can appear between the original adjacent images (orange boxes), causing misalignment between objects.

OneBEV Dataset Statistics. We accomplish an analysis of the semantic classes present in the dataset and carefully choose the appropriate ones to be used for training purposes. Two measures, namely "*pixel ratio*" and "*presence ratio*", are employed for evaluating these semantic classes. Specifically, the pixel ratio is the total number of pixels for each class across all frames divided by the total number of pixels for all classes across all frames. Similarly, the presence ratio is defined as the ratio of the total number of frames in which a particular class is present (considered present if at least two pixels belong to that class) to the total number of frames. The pixel ratio and presence ratio of each class in the datasets [3] and DeepAccident [30] are shown in Fig. 5. For the nuScenes-360 dataset, we retain or merge the classes (Fig. 5a) from the original nuScenes dataset based on the settings in [17], resulting in six classes: *drivable_area(road_segment, lane)*, *ped_crossing*, *walkway*, *stop_line*, *carpark_area*, and *divider(road_divider, lane_divider)*. For the DeepAccident-360 dataset, we analyze the 23 original classes. Initially, we remove the labels *unlabeled*, *other*, and *sky* as they are not relevant to our work. Based on the statistics in Fig. 5b, we further eliminate labels with a pixel ratio of 0: *traf-*



(a) Distribution of nuScenes dataset. **(b)** Distribution of DeepAccident dataset.

Fig. 5: Analysis of semantic class distributions. The per-class pixel ratio and presence ratio are presented in two datasets. Pixel ratio (%): Pixels of each class across all frames. Presence ratio (%): Frames where a class appears (at least 2 pixels).

fic_ sign, *traffic_ light*, and *rail_ track*, whose presence ratios are also less than 50%. This process ultimately results in 17 valid classes. The final panoramic datasets statistics are shown in Table 1.

Table 1: Data statistics of nuScenes-360 dataset and DeepAccident-360 dataset.

Dataset	#Scene	#Frame	#Class
nuScenes-360 `train`	700	28,130	6
nuScenes-360 `val`	150	6,019	6
nuScenes-360 `total`	850	34,149	6
DeepAccident-360 `train`	483	40,619	17
DeepAccident-360 `val`	104	8,193	17
DeepAccident-360 `total`	587	48,812	17

3.3 Model

OneBEV Architecture. OneBEV is an end-to-end model focusing on BEV semantic segmentation using panoramic image. Fig. 6 illustrates the overall structure. The front-view panoramic image serves as the input for the OneBEV model. The VMamba-T [16] backbone is employed to extract features from this input. Following this, the MVT module plays a crucial role by selecting specific feature points on the front-view image to be used as BEV queries. These selected feature points are then sent into the vanilla VSS block [16] for interaction. In the final stage, the semantic segmentation head takes charge, utilizing the processed features to perform the task of semantic segmentation.

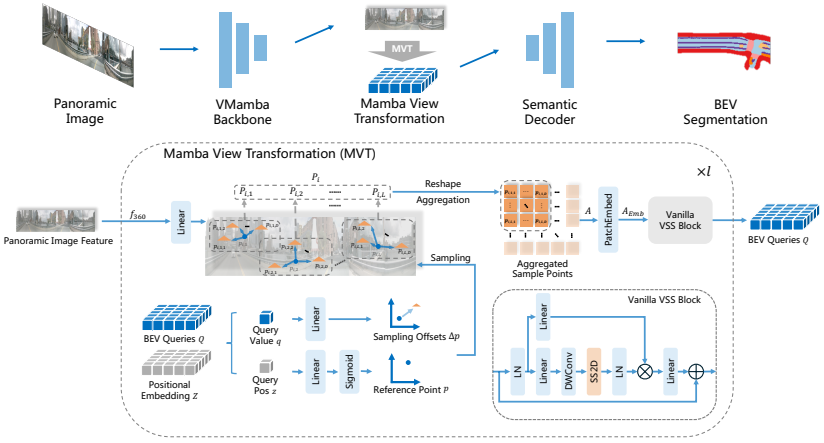


Fig. 6: Overall architecture of the proposed OneBEV. The VMamba-T [16] model as the backbone extracts features from the panorama. The Mamba-based view transformation (MVT) module transforms the front view features into bird's eye view features. A lightweight semantic decoder supports the semantic segmentation task.

Mamba View Transformation (MVT). The input to our OneBEV model is a panoramic image, and it lacks camera intrinsic and extrinsic parameters. The spatial cross-attention mechanism in [14] accomplishes view transformation based on these parameters, which is not applicable to our task. We design a novel module akin to the multi-camera deformable cross-attention [22] between the front-view feature and the BEV feature map, termed Mamba View Transformation (MVT). The detailed structure is depicted in the lower half of Fig. 6.

Following common practices, we first define BEV queries $Q \in \mathbb{R}^{H_q \times W_q \times C_{Emb}}$, along with their positional embeddings $Z \in \mathbb{R}^{H_q \times W_q \times C_{Emb}}$. For BEV queries Q, we perform a linear projection to obtain the sampling offsets $\Delta p \in \mathbb{R}^{H_q \times W_q \times N \times 2}$, while its positional embedding Z undergoes a linear projection followed by normalization through a sigmoid function to derive the reference points $p \in \mathbb{R}^{H_q \times W_q \times L \times 2}$, where H_q and W_q represent the spatial dimensions of BEV queries, C_{Emb} is the number of embedded dimensions, N is the total number of sampling points, and L is the number of sampling locations (in our task, N is always equal to L^2). The MVT module can be expressed as:

$$\begin{aligned} P_i &= W f_{360}(p_{i,j} + \Delta p_{i,j,k}), \\ A &= \text{Reshape}(P), \\ Q &= \text{VSS}(PatchEmbed(A)). \end{aligned} \tag{10}$$

Here, the panoramic features $f_{360} \in \mathbb{R}^{H_{360} \times W_{360} \times C_{Emb}}$ are linearly projected with the learnable parameter matrices $W \in \mathbb{R}^{C_{Emb} \times C_{Emb}}$. $p_{i,j}$ and $\Delta p_{i,j,k}$ denote the reference points and sampling offsets, respectively, where $i \in [1, ..., H_q W_q]$, $j \in [1, ..., L]$, $k \in [1, ..., D]$, and D is the number of sampling points at each

location (here $D = L$). Thus, we obtain a set of sampled points P_i from the panoramic features for each query q. Subsequently, each set $P_i \in \mathbb{R}^{LD \times C_{Emb}}$ is reshaped into a grid form $P_i \in \mathbb{R}^{L \times D \times C_{Emb}}$, and the set of all P_i, denoted as $P \in \mathbb{R}^{LH_qDW_q \times C_{Emb}}$, is reshaped into a larger grid form $A \in \mathbb{R}^{LH_q \times DW_q \times C_{Emb}}$. This process can be understood as each BEV query point q on the BEV feature map being "interpolate" by the sampling point grid $P_i \in \mathbb{R}^{L \times D \times C_{Emb}}$. The aggregated sample points A can be interpreted as the BEV feature map $Q \in \mathbb{R}^{H_q \times W_q \times C_{Emb}}$ originally input into the MVT being "interpolate" into a larger feature map $A \in \mathbb{R}^{LH_q \times DW_q \times C_{Emb}}$. To ensure that the reshaped feature map $A \in \mathbb{R}^{LH_q \times DW_q \times C_{Emb}}$ can be effectively processed, we apply a patch embedding to convert it back to the original BEV feature map size $A_{Emb} \in \mathbb{R}^{H_q \times W_q \times C_{Emb}}$. Finally, A_{Emb} is fed into vanilla VSS blocks from VMamba [16] to enable interaction between feature points. Q is the final output of the MVT module, which can then be used as the BEV queries of the next MVT module to perform sampling on the front view panoramic image.

Compared to the multi-camera deformable cross-attention module in BEVSegformer [22], our MVT module exhibits two main differences: **(1)** Exclusive sampling on panoramic feature map: MVT operates solely on the panoramic feature map, rather than on multi-view images. This approach provides the network with enhanced flexibility in selecting reference points, allowing for more precise feature extraction. **(2)** Absence of attention mechanism: Instead of the attention mechanism, our module introduces a novel Mamba-based method. In traditional attention mechanisms, attention weights are used to select sampled points relevant to each query. However, with the SS2D scanning operation in the vanilla VSS block, our approach not only facilitates interaction between each query's own sampled points but also enables interaction with sampled points from other queries, leading to a more integrated and comprehensive feature representation.

4 Experiment

4.1 Experimental Settings

Implementation Details. We set the embedding dimensions to 128. For the MVT module, we configure it with 4 layers, 5 sampling locations, and set the inner dimensions and depth of the vanilla VSS block to 128 and 2, respectively. The OneBEV model is trained using 4 A100 GPUs, with a batch size of 2 per GPU. The initial learning rate is 5×10^{-5} for the nuScenes-360 dataset and 8×10^{-5} for the DeepAccident-360 dataset. The training schedule consists of 50 epochs, utilizing both linear warm-up and cosine annealing strategy. We employ the AdamW optimizer with a weight decay of 0.01. The input for OneBEV training comprises panorama images, each sized at 180×3200. Following [17], the output BEV map is set to 200×200, covering a perception range of $[-50m, 50m] \times [-50m, 50m]$ around the ego vehicle. We also report the Intersection-over-Union (IoU) across 6 classes for nuScenes-360 and 17 classes for DeepAccident-360, along with the class-averaged mean IoU as our evaluation metrics.

Baselines. We conducted a comparative analysis of OneBEV against previous state-of-the-art models on the BEV semantic segmentation task. All models, except for [27], originally utilize images from multi-view cameras as input. In our work, we made a significant effort to reproduce these models and adapt their narrow-FoV inputs to 360° panoramic images. For the 360BEV model, which inherently accepts panoramic images but requires ground-truth depth for transformation, we adapt it to operate without depth information for a fair comparison. We standardized all models' view transformation modules to use the multi-camera deformable cross-attention from [22], adapting it for a single-camera setup.

4.2 Results

Results on Nuscenes-360 Dataset. Based on the results presented in Table 2 and the comparative analysis provided, it is evident that our proposed method, OneBEV, demonstrates significant improvements in performance compared to 360BEV on the nuScenes-360 dataset. Specifically, it shows a +3.7% improvement in *Walkway* IoU, +3.7% in *Stop Line* IoU, +4.0% in *Carpark* IoU and +3.3% in all classes mIoU. These results indicate the effectiveness of OneBEV in semantic segmentation tasks, showcasing its robustness and adaptability across various classes in the nuScenes-360 dataset. The combination of superior accuracy and a reduced parameter count (32.8M → 31M) positions OneBEV as a highly effective and efficient model.

Table 2: Per-class results on the `val` set of nuScenes-360 dataset.

Method	#Param.	Drivable_Area	Ped_Crossing	Walkway	Stop_Line	Carpark_Area	Divider	Mean
HDMapNet [13]	12.8M	72.8	37.8	44.2	33.8	35.8	31.4	42.6
BEVFusion [17]	53.7M	72.4	36.4	43.8	33.1	35.4	31.8	42.2
BEVSegformer [22]	52.8M	73.6	40.3	45.4	36.2	40.5	32.6	44.8
BEVFormer [14]	42.9M	74.7	40.5	46.1	38.1	37.3	35.0	45.3
360BEV [27]	32.8M	77.1	44.7	49.5	39.0	40.3	36.6	47.8
OneBEV (Ours)	**31M**	**79.2**	**47.6**	**53.2**	**43.4**	**44.3**	**39.0**	**51.1** (+3.3)

Results on DeepAccident-360 Dataset. The analysis of the results in Table 3 indicates that OneBEV exhibits moderate performance improvements against 360BEV on the DeepAccident-360 dataset, which is tailored to simulate real-world accident scenarios. Notably, it achieves a +4.6% improvement in *Building* IoU, +9.1% in *Water* IoU, +6.5% in *Ground* IoU and +1.8% in all classes mIoU. The DeepAccident-360 dataset, being a synthetic dataset, presents unique challenges, as it simulates complex and rare scenarios that are difficult to capture in real-world data. These results highlight OneBEV's capability to handle diverse and complex environments.

Table 3: Per-class results on the `val` set of DeepAccident-360 dataset.

Method	#Param.	Static	Dynamic	Building	Fence	Water	Terrain	Pedestrian	Pole	Road_Line	Road	Side_Walk	Vegetation	Vehicles	Wall	Ground	Bridge	Guard_Rail	Mean
HDMapNet [13]	12.8M	12.7	15.4	32.1	9.2	26.7	65.8	0.0	2.3	14.1	73.4	64.9	46.1	15.2	16.9	58.0	1.4	36.3	28.8
BEVFusion [17]	53.7M	10.0	11.1	24.5	9.1	17.9	57.4	0.0	0.8	12.0	66.4	55.7	40.7	13.2	16.9	43.5	0.0	25.8	23.8
BEVSegformer [22]	52.8M	13.6	17.6	34.7	11.0	42.6	63.7	0.0	2.7	13.9	71.6	62.7	49.2	22.8	20.0	56.4	0.5	**40.9**	30.8
BEVFormer [14]	42.9M	10.5	11.6	29.2	8.3	20.0	57.5	0.0	2.7	12.0	69.0	60.2	42.9	19.7	17.2	50.5	0.0	31.9	26.1
360BEV [27]	32.8M	15.6	16.3	42.1	10.1	50.4	68.0	0.0	**3.6**	16.9	76.7	70.4	50.6	29.5	**21.0**	67.4	**5.4**	39.6	34.3
OneBEV (Ours)	**31M**	**17.5**	**19.6**	**46.7**	**11.5**	**59.5**	**72.5**	0.0	2.8	**19.2**	**80.1**	**73.5**	**51.3**	**30.4**	19.7	**73.9**	4.7	31.7	**36.1** (+1.8)

4.3 Ablation Study

Effect of Backbone. Table 4 (rows 1-2) compares the MSCAN-B [8] backbone from 360BEV [27] and the VMamba-T [16] backbone within our model framework. Both use our MVT module for view transformation. The VMamba-T with the MVT module outperforms MSCAN-B in both model parameters and mIoU. This performance disparity is due to the architectural compatibility between VMamba-T and the MVT module, leading to more effective feature extraction and transformation. The VMamba-T backbone's design principles, such as better spatial resolution preservation and multi-scale feature aggregation, complement the MVT module's capabilities. Our findings suggest that the MVT module is more compatible with VMamba-T, highlighting the need for backbones designed to work effectively with advanced view transformation modules.

Table 4: Ablation study for OneBEV on the nuScenes-360 `val` set. 'VT': method of view transformation. '#Param.': number of parameters in the model, '#Dims': embedding dimensions of the model, '#Layers': repetitions of the view transformation module, '#Locs': number of sampling locations on the panorama, '#Points': total number of sampling points, '#Depth': repetitions of the vanilla VSS block.

Backbone	VT	#Param.	#Dims	#Layers	#Locs	#Points	#Depth	mIoU
MSCAN-B	MVT	35.1M	128	4	5	25	2	49.8
VMamba-T	MVT	31M	128	4	5	25	2	**51.1** (+1.5)
VMamba-T	CrossAtt	28.8M	128	2	6	24	-	35.8
VMamba-T	MVT	29.1M	128	1	5	25	1	**45.5** (+9.7)
VMamba-T	MVT	29.1M	128	1	5	25	1	45.5
VMamba-T	MVT	29.7M	128	2	5	25	1	48.5
VMamba-T	MVT	30.7M	128	4	5	25	1	**50.5** (+2.0)
VMamba-T	MVT	30.7M	128	4	5	25	1	50.5
VMamba-T	MVT	31M	128	4	5	25	2	**51.1** (+0.6)
VMamba-T	MVT	31M	128	4	5	25	2	**51.1**
VMamba-T	MVT	32.2M	128	6	5	25	2	50.5 (-0.6)

Effect of View Transformation Module. Table 4 (rows 3-4) shows an ablation experiment on the view transformation module. The model using multi-camera deformable cross attention [22] performs poorly compared to the the MVT module, despite the latter having a 0.3M higher parameter count and achieving a 9.7% higher mIoU. The cross-attention mechanism struggles to combine the features from the VMamba-T [16] backbone, while the MVT module, designed for these features, ensures high-quality feature fusion. The increased parameter count allows for more computations, leading to better performance. These findings emphasize the importance of using view transformation aligned with the backbone's feature characteristics for optimal performance.
Effect of Hyper-parameters. Table 4 (rows 5-7) examines the impact of the number of layers in the MVT module. The model performs best with four layers, balancing feature transformation and computational complexity. Additionally, Table 4 (rows 8-10) compares different depths of the vanilla VSS block [16], showing that greater depth leads to higher mIoU due to enhanced feature interaction. However, rows 11-12 indicate that increasing layers beyond six results in performance decline, suggesting over-fitting. These findings highlight the need for careful hyper-parameter selection to balance model complexity and performance, avoiding over-fitting while ensuring robust feature extraction and transformation. Future work should explore adaptive mechanisms to optimize layer and depth configurations dynamically.

5 Conclusion

In this work, we propose a novel approach for BEV semantic mapping, termed OneBEV, which employs a single panoramic image as input to tackle challenges occurring in traditional multi-camera setups, including calibration inaccuracies, synchronization issues, and high computational complexity. Instead of leveraging traditional attention mechanisms, we introduce MVT that facilitates interaction between the front-view feature map and the BEV feature map, to alleviate spatial distortions inherent in panoramas. Extensive experiments showcase that OneBEV achieves superior accuracy with fewer parameters, demonstrating its efficiency and effectiveness across multiple tasks. We also contribute two datasets, nuScenes-360 and DeepAccident-360, which will further facilitate panoramic-to-BEV semantic mapping research in the community. This paper aims to serve as a starting point, inspiring further discussions and developments on the panoramic-to-BEV semantic mapping in different domains, *e.g.*, indoor navigation, robotic perception and augmented reality.

Acknowledgments. This work was supported in part by the Ministry of Science, Research and the Arts of Baden-Württemberg (MWK) through the Cooperative Graduate School Accessibility through AI-based Assistive Technology (KATE) under Grant BW6-03, in part by BMBF through a fellowship within the IFI programme of DAAD, in part by the InnovationCampus Future Mobility funded by the Baden-Württem-berg Ministry of Science, Research and the Arts, and in part by the Helmholtz Association Initiative and Networking Fund on the HAICORE@KIT and HOREKA@KIT partition.

References

1. Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9. pp. 404–417. Springer (2006) 8
2. Borse, S., Klingner, M., Kumar, V.R., Cai, H., Almuzairee, A., Yogamani, S., Porikli, F.: X-align: Cross-modal cross-view alignment for bird's-eye-view segmentation. In: WACV (2023) 1, 2
3. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11621–11631 (2020) 2, 4, 5, 8
4. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: Carla: An open urban driving simulator. In: Conference on robot learning. pp. 1–16. PMLR (2017) 3, 8
5. Ge, C., Chen, J., Xie, E., Wang, Z., Hong, L., Lu, H., Li, Z., Luo, P.: Metabev: Solving sensor failures for 3d detection and map segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8721–8731 (2023) 1, 2, 4
6. Gu, A., Dao, T.: Mamba: Linear-time sequence modeling with selective state spaces. arXiv preprint arXiv:2312.00752 (2023) 2, 3
7. Gu, A., Goel, K., Ré, C.: Efficiently modeling long sequences with structured state spaces. arXiv preprint arXiv:2111.00396 (2021) 3
8. Guo, M.H., Lu, C.Z., Hou, Q., Liu, Z., Cheng, M.M., Hu, S.M.: Segnext: Rethinking convolutional attention design for semantic segmentation. Advances in Neural Information Processing Systems **35**, 1140–1156 (2022) 13
9. He, X., Cao, K., Yan, K., Li, R., Xie, C., Zhang, J., Zhou, M.: Pan-mamba: Effective pan-sharpening with state space model. arXiv preprint arXiv:2402.12192 (2024) 3
10. Huang, J., Huang, G., Zhu, Z., Ye, Y., Du, D.: Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. arXiv preprint arXiv:2112.11790 (2021) 4
11. Jaus, A., Yang, K., Stiefelhagen, R.: Panoramic panoptic segmentation: Towards complete surrounding understanding via unsupervised contrastive learning. In: 2021 IEEE Intelligent Vehicles Symposium (IV). pp. 1421–1427. IEEE (2021) 4
12. Li, H., Sima, C., Dai, J., Wang, W., Lu, L., Wang, H., Zeng, J., Li, Z., Yang, J., Deng, H., et al.: Delving into the devils of bird's-eye-view perception: A review, evaluation and recipe. IEEE Transactions on Pattern Analysis and Machine Intelligence (2023) 1, 4
13. Li, Q., Wang, Y., Wang, Y., Zhao, H.: Hdmapnet: An online hd map construction and evaluation framework. In: 2022 International Conference on Robotics and Automation (ICRA). pp. 4628–4634. IEEE (2022) 12, 13
14. Li, Z., Wang, W., Li, H., Xie, E., Sima, C., Lu, T., Qiao, Y., Dai, J.: BEVFormer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In: ECCV (2022) 1, 4, 10, 12, 13
15. Liao, Y., Xie, J., Geiger, A.: Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d. IEEE Transactions on Pattern Analysis and Machine Intelligence **45**(3), 3292–3310 (2022) 4
16. Liu, Y., Tian, Y., Zhao, Y., Yu, H., Xie, L., Wang, Y., Ye, Q., Liu, Y.: Vmamba: Visual state space model. arXiv preprint arXiv:2401.10166 (2024) 2, 3, 9, 10, 11, 13, 14

17. Liu, Z., Tang, H., Amini, A., Yang, X., Mao, H., Rus, D.L., Han, S.: Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In: 2023 IEEE international conference on robotics and automation (ICRA). pp. 2774–2781. IEEE (2023) 1, 2, 4, 8, 11, 12, 13
18. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60**, 91–110 (2004) 8
19. Ma, C., Zhang, J., Yang, K., Roitberg, A., Stiefelhagen, R.: Densepass: Dense panoramic semantic segmentation via unsupervised domain adaptation with attention-augmented context exchange. In: 2021 IEEE International Intelligent Transportation Systems Conference (ITSC). pp. 2766–2772. IEEE (2021) 4
20. Mallot, H.A., Bülthoff, H.H., Little, J., Bohrer, S.: Inverse perspective mapping simplifies optical flow computation and obstacle detection. Biological cybernetics **64**(3), 177–185 (1991) 4
21. Orhan, S., Bastanlar, Y.: Semantic segmentation of outdoor panoramic images. Signal, Image and Video Processing **16**(3), 643–650 (2022) 4
22. Peng, L., Chen, Z., Fu, Z., Liang, P., Cheng, E.: BEVSegFormer: Bird's eye view semantic segmentation from arbitrary camera rigs. In: WACV (2023) 1, 2, 4, 10, 11, 12, 13, 14
23. Philion, J., Fidler, S.: 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. pp. 194–210. Springer (2020) 4
24. Roddick, T., Cipolla, R.: Predicting semantic map representations from images using pyramid occupancy networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11138–11147 (2020) 1
25. Rublee, E., Rabaud, V., Konolige, K., Bradski, G.: Orb: An efficient alternative to sift or surf. In: 2011 International conference on computer vision. pp. 2564–2571. Ieee (2011) 8
26. Saha, A., Mendez, O., Russell, C., Bowden, R.: Translating images into maps. In: 2022 International conference on robotics and automation (ICRA). pp. 9200–9206. IEEE (2022) 4
27. Teng, Z., Zhang, J., Yang, K., Peng, K., Shi, H., Reiß, S., Cao, K., Stiefelhagen, R.: 360bev: Panoramic semantic mapping for indoor bird's-eye view. In: WACV (2024) 4, 5, 12, 13
28. Wan, Z., Wang, Y., Yong, S., Zhang, P., Stepputtis, S., Sycara, K., Xie, Y.: Sigma: Siamese mamba network for multi-modal semantic segmentation. arXiv preprint arXiv:2404.04256 (2024) 3
29. Wang, H., Tang, H., Shi, S., Li, A., Li, Z., Schiele, B., Wang, L.: Unitr: A unified and efficient multi-modal transformer for bird's-eye-view representation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6792–6802 (2023) 1, 2, 4
30. Wang, T., Kim, S., Wenxuan, J., Xie, E., Ge, C., Chen, J., Li, Z., Luo, P.: Deepaccident: A motion and accident prediction benchmark for v2x autonomous driving. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 5599–5606 (2024) 2, 4, 8
31. Xie, E., Yu, Z., Zhou, D., Philion, J., Anandkumar, A., Fidler, S., Luo, P., Alvarez, J.M.: M ^{2} bev: Multi-camera joint 3d detection and segmentation with unified birds-eye view representation. arXiv preprint arXiv:2204.05088 (2022) 4
32. Xu, Y., Wang, K., Yang, K., Sun, D., Fu, J.: Semantic segmentation of panoramic images using a synthetic dataset. In: Artificial Intelligence and Machine Learning in Defense Applications. vol. 11169, pp. 90–104. SPIE (2019) 4

33. Yang, K., Hu, X., Stiefelhagen, R.: Is context-aware cnn ready for the surroundings? panoramic semantic segmentation in the wild. IEEE Transactions on Image Processing **30**, 1866–1881 (2021) 4
34. Zhang, J., Yang, K., Ma, C., Reiß, S., Peng, K., Stiefelhagen, R.: Bending reality: Distortion-aware transformers for adapting to panoramic semantic segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16917–16927 (2022) 2, 4
35. Zhou, B., Krähenbühl, P.: Cross-view transformers for real-time map-view semantic segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13760–13769 (2022) 4
36. Zhu, L., Liao, B., Zhang, Q., Wang, X., Liu, W., Wang, X.: Vision mamba: Efficient visual representation learning with bidirectional state space model. arXiv preprint arXiv:2401.09417 (2024) 2, 3