


## Article

# Low-Cost Data-Driven Robot Collision Localization Using a Sparse Modular Point Matrix

Haoyu Lin , Pengkun Quan , Zhuo Liang, Dongbo Wei and Shichun Di \*

School of Mechatronics Engineering, Harbin Institute of Technology, Harbin 150001, China; linhaoyu@hit.edu.cn (H.L.); quanpengkun@hit.edu.cn (P.Q.); liangzhuo@hit.edu.cn (Z.L.); weidb@hit.edu.cn (D.W.)

\* Correspondence: dishichun@hit.edu.cn; Tel.: +86-139-046-05946

**Abstract:** In the context of automatic charging for electric vehicles, collision localization for the end-effector of robots not only serves as a crucial visual complement but also provides essential foundations for subsequent response design. In this scenario, data-driven collision localization methods are considered an ideal choice. However, due to the typically high demands on the data scale associated with such methods, they may significantly increase the construction cost of models. To mitigate this issue to some extent, in this paper, we propose a novel approach for robot collision localization based on a sparse modular point matrix (SMPM) in the context of automatic charging for electric vehicles. This method, building upon the use of collision point matrix templates, strategically introduces sparsity to the sub-regions of the templates, aiming to reduce the scale of data collection. Additionally, we delve into the exploration of data-driven models adapted to SMPMs. We design a feature extractor that combines a convolutional neural network (CNN) with an echo state network (ESN) to perform adaptive feature extraction on collision vibration signals. Simultaneously, by incorporating a support vector machine (SVM) as a classifier, the model is capable of accurately estimating the specific region in which the collision occurs. The experimental results demonstrate that the proposed collision localization method maintains a collision localization accuracy of 91.27% and a collision localization RMSE of 1.46 mm, despite a 48.15% reduction in data scale.

**Keywords:** automatic charging; data-driven collision localization; sparse modular point matrix; convolutional neural network; echo state network; support vector machine



**Citation:** Lin, H.; Quan, P.; Liang, Z.; Wei, D.; Di, S. Low-Cost Data-Driven Robot Collision Localization Using a Sparse Modular Point Matrix. *Appl. Sci.* **2024**, *14*, 2131. <https://doi.org/10.3390/app14052131>

Academic Editor: Paolino Di Felice

Received: 5 February 2024

Revised: 25 February 2024

Accepted: 1 March 2024

Published: 4 March 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

In the domain of robot-assisted automatic electric vehicle charging, the connection between the charger and the charging port relies on precise visual positioning [1]. However, the visual positioning system may be subject to disturbances in unstructured environments, such as changes in lighting conditions, leading to instances of positioning failure. Visual positioning failure typically results in three scenarios: in cases of minimal localization deviation, the charger carried by the robot's end-effector is able to connect to the charging port, but may experience jamming. In such situations, impedance control implemented on the robot can effectively suppress jamming [2]. When the positioning deviation is substantial, the charger on the robot's end-effector may fail to make contact with the charging port during the connection process, potentially causing contact with other parts of the electric vehicle's body. In this case, implementing a collision classification protection system on the robotic arm ensures the safety of the vehicle and the robot. When the positioning deviation falls between the aforementioned scenarios, meaning that the charger can make contact with the charging port but cannot smoothly insert due to the presence of a visual positioning deviation, imparting a collision localization capability to the robotic arm can effectively correct the deviation caused by the visual positioning failure, serving as a supplementary localization strategy in the event of a visual failure [3,4].

In the exploration of model-based collision localization methods, J. Vorndamme et al. achieved collision localization for humanoid robots by constructing a generalized momentum observer to calculate joint torques and estimate joint accelerations [5]. This method enables point estimation in single-contact situations using only onboard sensors. Additional force/torque sensors are introduced only when estimating multi-contact positions. M. Iskandar et al. developed a momentum-based external force estimation framework for robot collision localization [6]. This approach includes joint-level residual estimation and uncoupled force–torque estimation in Cartesian space, eliminating the need for acceleration estimation and consequently mitigating the introduction of noise associated with acceleration estimation. D. Zurlo et al. addressed the problem of difficulty in achieving high-precision collision localization solely by relying on a generalized momentum observer to a certain extent by combining the generalized momentum observer method with a particle filtering strategy [7].

In the pursuit of achieving high-precision collision localization, artificial-skin-based methods are generally considered a more favorable option. P. Piacenza et al. utilized low-cost optical components installed along the edges of the perception region to achieve higher accuracy in contact localization by measuring the impact of touch on the passage of light through elastic material [8]. X. Fan et al. designed a set of ultrasound sensors deployable on the surface of a robotic arm to achieve high-precision contact localization and analyze contact pressure [9]. P. Mittendorf et al. achieved interactive touch in different parts of a humanoid robot by employing self-organizing, multimodal artificial skin [10]. X. Li et al. developed a tactile sensor composed of overlapping air chambers, leveraging the spatiotemporal continuity of learning contact positions to achieve high-precision and high-resolution collision localization [11].

With the rapid advancement of artificial intelligence technology, supervised learning strategies have become widely utilized to address collision localization problems in robotics. These methods are commonly referred to as data-driven collision localization approaches. D. Popov et al. employed onboard sensors to collect collision data from robots, utilizing neural network methods to learn from the relevant data, thereby achieving collision localization at the centimeter level [12]. X. Ha et al. utilized information from multi-core fiber Bragg grating sensors, combined with a k-nearest neighbor (KNN) model to fit a free-space curvature model, successfully estimating collision positions for continuum robots [13]. F. Min et al. mounted accelerometers on the joint near the base and end-effector of a robotic arm to capture vibration signals during collisions. They performed reasonable feature extraction on the collision vibration signals and, in conjunction with an artificial neural network (ANN), successfully achieved the localization of the contact points [14]. W. McMahon et al. mounted four accelerometers on a single robotic arm to form an accelerometer array, capturing collision vibration data. They employed a support vector machine (SVM) to learn from the vibration information of different collision positions, thereby achieving collision localization with an error in the centimeter range [15].

In the realm of robot-assisted automatic electric vehicle charging, the end-effector, which is incapable of establishing direct physical contact with the charging port, exclusively interfaces with it through the intermediary of the carried charger. This situation may introduce unknown disturbances in signal measurements within the model-based method, posing challenges for achieving high accuracy in collision localization with model-based methods. Additionally, during the plug-in process, the forces generated during collisions typically act along the robot. As discussed in [5], model-based methods face increased difficulty in handling collision issues when external forces act along the robot. In addition, due to the frequent contacts and substantial contact forces inherent in the plug-in process, this demanding operational environment will significantly diminish the lifespan of artificial skin. Simultaneously, the deployment of artificial skin encounters certain challenges, serving as a constraint that restricts its application in addressing this issue. Considering data-driven collision localization methods, these approaches heavily depend on formulating rules for gathering data tailored to specific scenarios.

In our previous research, we introduced a collision point matrix template specifically designed for millimeter-level collision localization in the scenario of automatic charging for electric vehicles [4]. The collision point matrix template consists of collision points spaced at 1 mm intervals on a plane. By pre-setting the collision point matrix template on the surface of the charging port and then colliding with each collision point using the charger carried by the automatic charging robot, we can obtain collision vibration information suitable for collision localization. Utilizing a collision point matrix template composed of densely distributed collision points, the collision localization problem can be transformed into a classification problem, with collision information corresponding to different points in the template. To enhance the generalization ability of the collision localization method, it is necessary to consider the adaptation of the collision localization method to variations in joint configurations during the data collection process. Therefore, it is generally required to collect collision information under as many different joint configurations as possible. As the accuracy of collision localization in this method depends on the dense distribution of collision points in the template, the cost of data collection is typically very high.

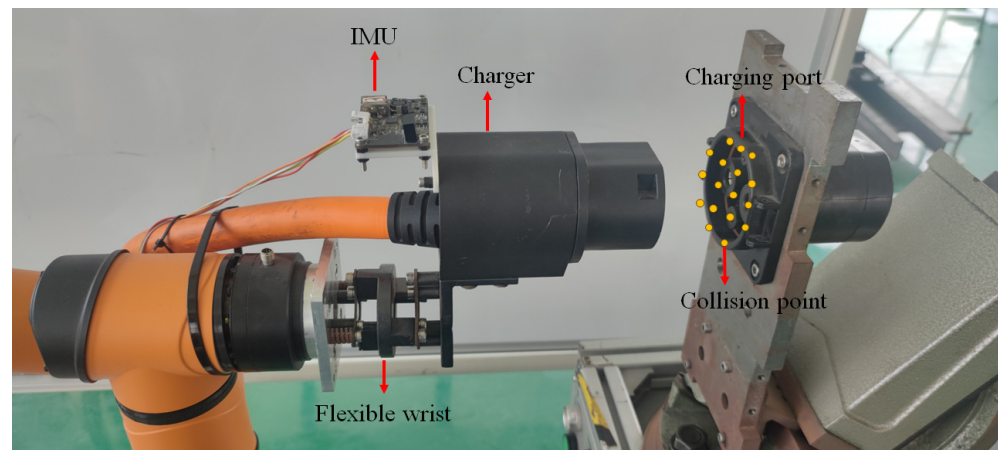
To alleviate the significant burden of data collection associated with such an approach, we propose a data-driven collision localization method based on a sparse modular point matrix (SMPM). Unlike the earlier collision point matrix template, the SMPM efficiently reduces the density of collision point distribution, thereby reducing the scale and associated costs of building the collision dataset to some extent. The main contributions of this paper are as follows:

1. Building upon the collision point matrix template, the SMPM is first introduced to achieve local sparsity of the template, thereby reducing the data scale required for the data-driven collision localization method;
2. Comparative experiments are conducted by constructing SMPMs of various forms and degrees of sparsity, exploring the optimal way to build SMPMs effectively while maintaining high collision localization performance with a reduced data scale;
3. A data-driven collision localization method combining a convolutional neural network (CNN), an echo state network (ESN), and a support vector machine (SVM) is proposed to enable the SMPM to achieve optimal performance in collision localization.

## 2. Materials and Methods

### 2.1. Dataset Description

The SMPM proposed in this study was constructed based on the collision point matrix template introduced in ref. [4]. To investigate the effectiveness of the proposed SMPM, the data used in this study were consistent with our previous work [4]. Specifically, the datasets comprised vibration signals encompassing 3-axis acceleration and 3-axis angular velocity, collected by the IMU mounted on the charger at a sampling frequency of 1500 Hz. An AUBO-i5 robot, a commercially available general-purpose 6-DOF robotic arm, was employed as the automatic charging equipment. It was connected to the charger via a flexible wrist, as depicted in Figure 1. During the data collection process, the end-effector robot moves in a linear motion at a speed of 15 mm/s to execute the collision. Each collision point in the collision point matrix template attached to the charging port undergoes five collisions in the same pattern to minimize the impact of robot positioning errors on the results. Additionally, we considered the impact of different joint configurations on the collision localization results, thereby constructing three independent collision vibration signal datasets named D1, D2, and D3. Each dataset corresponds to three sets of distinct joint configurations, with each dataset containing 4335 samples. For more details, please refer to the table entitled “Joint configuration of the datasets” in ref. [4].



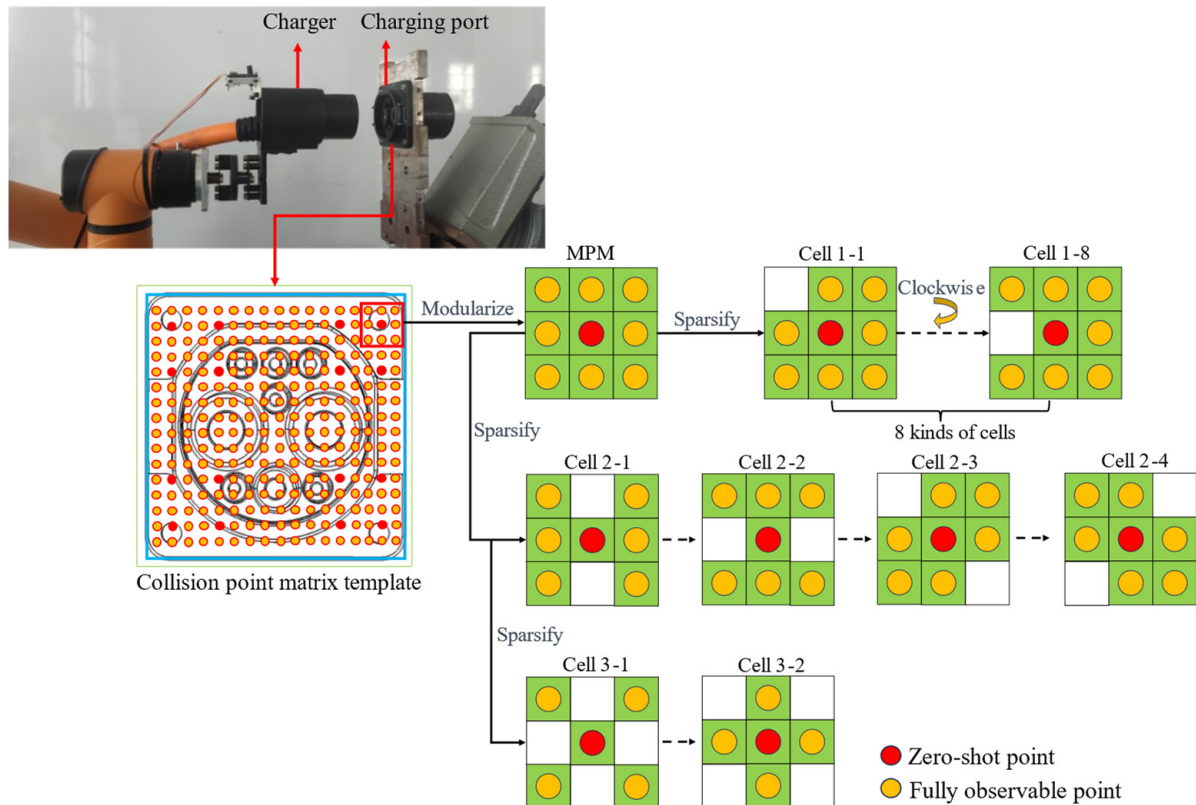
**Figure 1.** Illustration of the automatic charging equipment.

## 2.2. SMPM Method

In our previous work, we observed that when using the collision point matrix template, the estimated positions of collision points are prone to confusion with their neighboring collision points. This implies that it is possible to estimate the occurrence of collisions at a particular collision point by leveraging the collision vibration information from its neighboring points. Building upon this idea, we propose a local modularization and sparsification approach for the collision point matrix template, as illustrated in Figure 1. The collision point matrix template mentioned here is identical to the one in ref. [4]. Collision points are defined as intersection points between the central axis of the charger and the plane in which the charging port is located. The template comprises collision points with 1 mm spacing, arranged in 17 rows and 17 columns, with its center located at the intersection of the central axis of the charging port and the plane in which the charging port is situated. In practical applications, the template can be scaled without altering the spacing between collision points. Modularization is achieved by exploiting the similarity in collision vibration signals between the estimated collision points and their neighboring points, eliminating the need to collect data for the estimated collision points during the data collection process. We refer to the estimated collision points that do not appear in the dataset as “zero-shot points”, while the collision points used to estimate zero-shot points, requiring collection in the dataset, are defined as “fully observable points”. In the process of implementing local modularization, we consider the information from fully observable points nearest to the zero-shot points to estimate collisions occurring at the zero-shot points. This process leads to the formation of a modular point matrix (MPM), as illustrated in the figure, comprising a central zero-shot point and its surrounding eight fully observable points. The MPM utilized results in a 1/9 reduction in the quantity of data collected, compared to the original collision point matrix template.

Furthermore, it is crucial to consider whether utilizing all eight fully observable points is necessary for accurately estimating a collision occurrence at a zero-shot point. In theory, the collision vibration information obtained from these eight fully observable points may contain redundancy when estimating a zero-shot point. If this hypothesis holds true, eliminating the redundant fully observable points could further reduce the scale of the collision dataset. Taking this into consideration, we propose three sparsification methods for the MPM, as illustrated in Figure 2. The first sparsification method involves removing one of the fully observable points from the MPM. This approach results in eight sparse modular point matrices (SMPMs), denoted as Cell 1-1, Cell 1-2, ..., Cell 1-8, obtained by sequentially removing one fully observable point in clockwise direction starting from the top left corner. The second sparsification method involves removing two fully observable points from the MPM. During this removal process, we consider two extreme cases: one that maximally preserves the zero-shot point’s farthest adjacent points (resulting in Cell 2-1 and

Cell 2-2), and another that maximally preserves the zero-shot point's nearest adjacent points (resulting in Cell 2-3 and Cell 2-4). The third sparsification method involves removing four fully observable points from the MPM, specifically resulting in Cell 3-1, which excludes all of the nearest adjacent points, and Cell 3-2, which excludes all of the farthest adjacent points. The collision localization effects arising from the different sparsification methods are explained in detail in the experimental section.



**Figure 2.** Different sparsification forms of SMPMs.

### 2.3. Collision Localization Model

In our previous work, we explored collision localization models based on a CNN [3] and an ESN [4], respectively. Drawing inspiration from these two approaches, we proposed a collision localization model that integrates a CNN and an ESN as feature extractors. In this model, the CNN demonstrates a propensity for capturing salient features along the line of sight, making it a prevalent choice for feature engineering. Meanwhile, the ESN exhibits the capability to unfold in accordance with the temporal sequence, finding widespread applications in time series analysis. To enhance the ultimate localization performance, we also integrate an SVM model as the final region classifier. Capitalizing on distinctive attributes of the CNN and ESN, we formulate a collision localization model based on a CNN-ESN-SVM (CE-SVM) architecture.

#### 2.3.1. CNN

The CNN, a representative deep learning method, is known for its efficacy in processing time-series and image signals [16,17]. A typical CNN structure comprises two main components: the convolutional layer and the pooling layer. In the convolutional layer, the convolution operation is applied between the input features and convolution kernels, resulting in the generation of new features. Following convolution, the obtained results typically undergo non-linear processing, often facilitated by activation functions. Commonly employed activation functions include Sigmoid, tanh, and ReLU [18]. Based

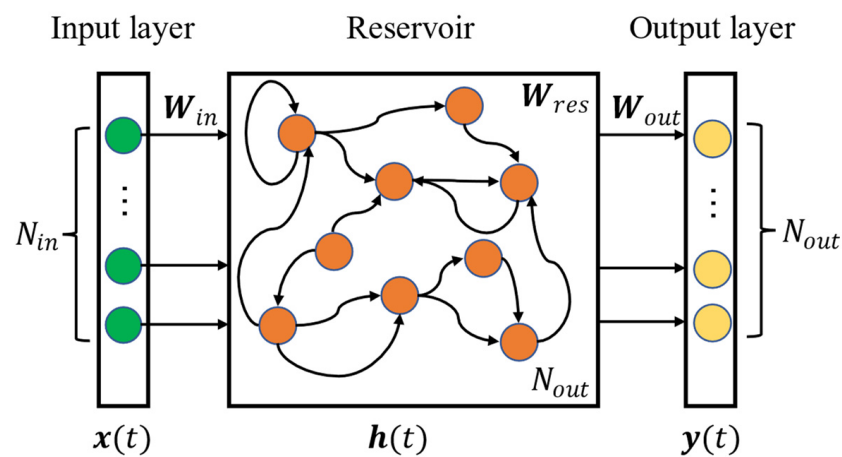


on previous research results, ReLU activation functions are a suitable choice for collision localization problems.

The pooling layer serves two primary functions: dimensionality reduction and mitigating overfitting. There are two main types of pooling methods: average pooling and maximum pooling. In average pooling, the operation involves taking the average of the convolution-derived features as the output, while in maximum pooling, the operation involves selecting the maximum value from the convolution-derived features as the output. In this research, we adopted the same pooling method as in our previous work, specifically utilizing the maximum pooling approach.

### 2.3.2. ESN

An echo state network (ESN) is a type of recurrent neural network proposed by Jaeger et al. [19], consisting of three main components: an input layer, a reservoir, and an output layer. The reservoir is essentially a randomly connected recurrent network of a certain size, where neurons form a dense structure through random connections. These connections are predetermined and remain unaltered during training. A basic ESN model is illustrated in Figure 3.



**Figure 3.** The structure of the ESN.

Let  $N_{in}$ ,  $N_{res}$ , and  $N_{out}$  represent the numbers of neurons in the input layer, reservoir, and output layer, respectively. The matrices  $W_{in}$ ,  $W_{res}$ , and  $W_{out}$  denote the weight matrices from the input layer to the reservoir, within the reservoir, and from the reservoir to the output layer, respectively.  $W_{in}$  and  $W_{res}$  are randomly initialized and remain fixed throughout the training process. Only  $W_{out}$  undergoes adjustments during the learning process. The specific ESN model can be expressed as follows:

$$\mathbf{h}(t) = \varepsilon \tanh(\mathbf{W}_{in}\mathbf{x}(t) + \mathbf{W}_{res}\mathbf{h}(t) + \mathbf{W}_{out}\mathbf{y}(t)) \quad (1)$$

where  $\tanh(\cdot)$  represents the non-linear activation function of the reservoir and  $\varepsilon \in (0, 1]$  is the leakage rate.  $\mathbf{x}(t)$ ,  $\mathbf{h}(t)$ , and  $\mathbf{y}(t)$  denote the input vector, the state vector of the reservoir, and the output vector, respectively. Compared to conventional RNNs, the training process of the ESN is simpler, only involving parameter adjustments in the output layer. The entire network does not require the complex process of backpropagation. Furthermore, due to the randomness and dense connections in the reservoir, this structure facilitates enhanced generalization capabilities, enabling the network to capture the non-linear dynamics of input signals effectively. This property contributes to the ESN's strong performance in handling time-series tasks.

In terms of hyperparameter settings, since  $W_{in}$  and  $W_{res}$  are generated through random initialization, it is essential to predefine the range for their random initialization before training. The appropriate values for these two weight matrices were adopted from Ref. [20].

Additionally, following our previous work [4], the leakage rate  $\varepsilon$ , spectral radius  $\rho$ , and  $N_{res}$  were set. Specifically, the hyperparameters of the ESN used in this paper are presented in Table 1.

**Table 1.** Key parameters of ESN.

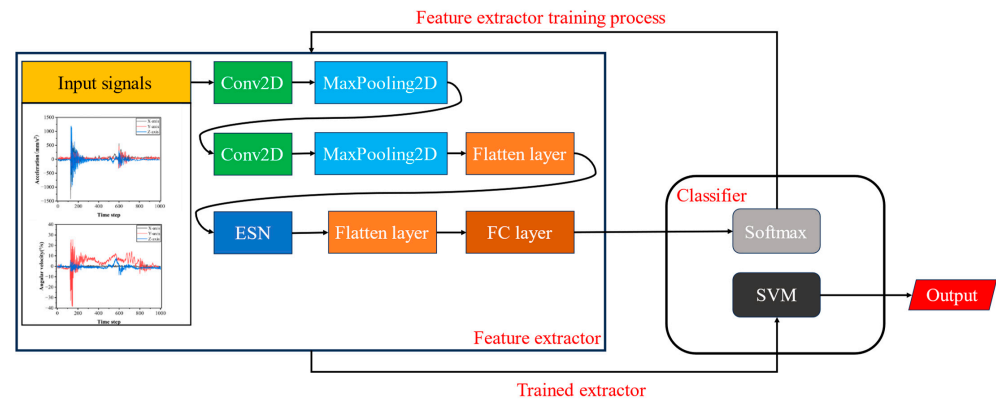
Parameters	Symbolic Representations	Values
Weight matrices from the input layer to the reservoir	$W_{in}$	$[-0.5, 0.5]$
Weight matrices within the reservoir	$W_{res}$	$[-0.5, 0.5]$
Leakage rate	$\varepsilon$	0.5
Spectral radius	$\rho$	1
Numbers of neurons in the reservoir	$N_{res}$	64

### 2.3.3. Framework of CE-SVM

The framework of the proposed CE-SVM is illustrated in Figure 4, consisting primarily of a feature extractor and a classifier. The collision vibration signals employed are 3-axis acceleration and 3-axis angular velocity signals collected by the IMU mounted on the charger. After normalization, these signals serve as inputs to the model. The definition of the input data length follows the concept of the “effective period” from our previous work, where a segment with rich information meeting collision localization requirements is extracted from the initial data length, as detailed in ref. [4]. As discussed in ref. [4], an effective period with 290 sampling points already contains sufficient information for collision localization. Therefore, this paper also sets the effective period to 290 sampling points. The input layer of the feature extractor is followed by two CNN layers, each composed of a convolutional layer and a maxpooling layer. In the diagram, Conv2D denotes a 2D convolution layer, and MaxPooling2D denotes a 2D pooling layer. Post Conv2D, batch normalization is applied to ensure the data’s generalization ability. Subsequently, a non-linear ReLU activation function is used to process the features, enhancing the model’s capacity for effective non-linear information processing. Notably, the Conv2D structure employed in this study differs from that of [3]. While the previous work involved symmetric  $3 \times 3$  convolutional kernels, in this study, we adopt asymmetric kernels to maximally preserve temporally reasonable features extracted by the CNN for subsequent processing by the ESN layer. The convolution kernel size in the temporal direction is significantly larger than that in the different axis dimensions. To effectively transmit temporal information to the ESN layer, the time-distributed technique [21] is employed for the flattening layer connecting the CNN layers and the ESN layer. To enable the SVM to effectively utilize the features extracted by the ESN, the features need to undergo flattening processing after the ESN process. Simultaneously, a fully connected layer is employed to reduce the dimensionality of the features to prevent the curse of dimensionality. In the feature extractor training process, Softmax is used as the final classifier. Based on feedback from the Softmax layer’s estimation results, the weights of different components in the feature extractor are adjusted. In the training of the classifier SVM for collision localization, the weights of the pre-trained feature extractor are fixed and used solely for feature extraction. The SVM is then constructed based on the features extracted by the pre-trained feature extractor.

Detailed hyperparameters for the feature extractor and classifier are provided in Table 2. The hyperparameters of the ESN and SVM are taken from [4], while the hyperparameters of the CNN remain consistent with those outlined in ref. [3], with the exception of the convolutional kernel aspect. Since the padding method is not utilized during the initial convolutional computation, it should ensure that the length of the input can be evenly divisible by the convolutional length along the temporal dimension. Furthermore, to maintain an appropriate convolutional size along the temporal dimension, we specify the kernel size in this direction as 10. Meanwhile, the kernel size remains consistent with ref. [3] in the other direction. Our experiments utilized a Windows-based system with

the following specifications: Processor: Intel (R) Core (TM) i7-10700K CPU @ 3.80 GHz, Memory: 31.9 GiB, GPU: NVIDIA GeForce RTX 3080.



**Figure 4.** Structure of the CE-SVM.

**Table 2.** Hyperparameters of the CE-SVM.

Type	Name of Parameter	Values
Conv2D	Number of filters	64
	Kernel size	(10, 3)
	Stride	1
Batch normalization	-	-
ReLU	-	-
Maxpooling	Pool size	(2, 2)
	Stride	1
	Padding	same
Conv2D	Number of filters	64
	Kernel size	(10, 3)
	Stride	2
	Padding	same
Batch normalization	-	-
ReLU	-	-
Maxpooling	Pool size	(2, 2)
	Stride	1
	Padding	same
Time-distributed flattening	-	-
ESN	$N_{res}$	64
	$\varepsilon$	0.5
	$\rho$	1
FC	Number of hidden units	512
SVM	Regularization parameter	100
	Kernel function	rbf

### 3. Results and Discussion

To explore the effectiveness of the SMPM in reducing the required data scale for collision localization model construction, the experimental design of this research mainly consists of two parts. The first part aims to analyze the SMPM structure under discrete distributions and select the optimal structure based on the structural analysis results. The second part aims to investigate the effectiveness of the proposed collision localization method when employing the optimal SMPM layout across the entire collision point template. In the first part of the experiment, various SMPM structures are predefined based



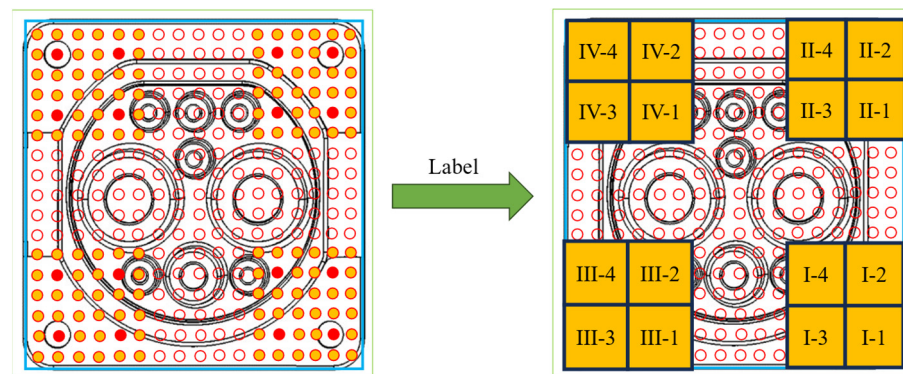
on the characteristics of vibration signals corresponding to collision points. Subsequently, utilizing multiple data-driven models, SMPMs with diverse structures across the discrete distribution of collision point templates are evaluated, leading to the selection of the optimal SMPM structure. In the second part of the experiment, the optimal SMPM is deployed throughout the entire collision point template with varying degrees of sparsity. This deployment allows for testing the performance of the optimal SMPM in reducing the necessary data scale for constructing the collision localization model while maintaining collision localization performance. By integrating the test outcomes, the collision localization model that best complements the SMPM is also identified.

### 3.1. Optimal SMPM Structure

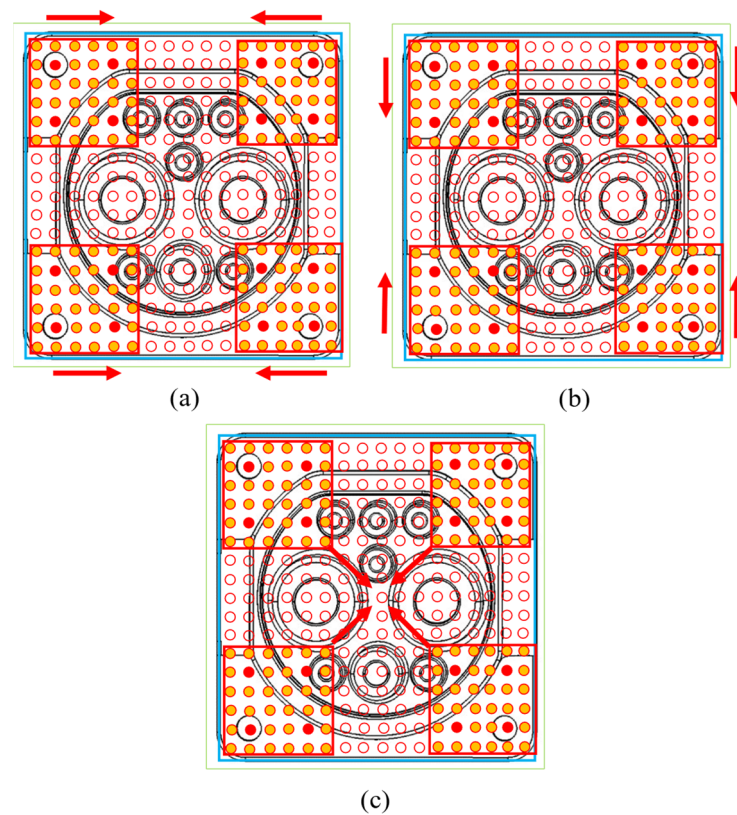
In conducting a comprehensive structural analysis and optimization of the SMPM across the entire collision point matrix template, significant computational costs are incurred. In this study, we mitigate these computational challenges by decomposing the SMPM optimization problem into distinct local regions. This subdivision results in a substantial reduction in the workload for the optimal SMPM structure selection method. The proposed optimal SMPM structure selection method consists of two main steps: firstly, various forms of sparsification are applied to the MPM distributed in the four-corner region. Collision localization is then performed on the SMPMs using multiple models. Based on the accuracy of the localization results, SMPMs with superior performance are initially identified. Subsequently, the position of the SMPM relative to the collision point matrix template is adjusted, and further collision localization using multiple models is conducted on the initially screened SMPMs to select those with the optimal structure. In terms of model selection, the proposed CE-SVM method from this study is employed, along with the DCNN-SVM method introduced in [3], as well as the ESN-SVM, LSTM-SVM, and GRU-SVM methods mentioned in [4]. Additionally, based on our previous findings, the effective handling of collision point localization in the testing set when a particular collision point in the collision point matrix template is present in both the training and testing sets has been validated. Therefore, in the SMPM selection process, greater attention can be directed towards evaluating the performance of collision localization for zero-shot points. Consequently, in the SMPM selection process, we employ all points in datasets D1, D2, and D3 that meet the definition of zero-shot points as the testing set. Simultaneously, all points in datasets D1, D2, and D3 that conform to the definition of fully observable points are utilized as the training set.

The distribution of MPMs in four corner regions of the collision point matrix template are illustrated in Figure 5. To introduce a certain level of similarity interference, four MPMs are set in each region, and are tightly connected to form a square area. These MPMs are labeled for collision localization. Specifically, we defined the regions in the four corners as I, II, III, and IV. Subsequently, we assigned numerical labels from 1 to 4 to the MPMs within each region. Then, different forms of sparsification were applied to the MPMs, as shown in Figure 2. Based on these various sparsification forms, collision localization tests were conducted to preliminarily identify preferable sparsification forms.

Building upon the aforementioned preferable configurations, we further displaced the SMPM to occupy different positions on the collision point matrix template. As illustrated in Figure 6, there are three types of movements: vertical (up and down), horizontal (left and right), and toward the center. A collision localization test was conducted each time the SMPM was shifted by a distance equivalent to one collision point. Based on the results of these tests, we refined the selection of the optimal SMPM structure. To facilitate the explanation, we defined the following situations: horizontal movement by one collision point as LR1 and by two collision points as LR2; vertical movement by one collision point as UD1 and by two collision points as UD2; and movement toward the center by one collision point as CT1 and by two collision points as CT2.



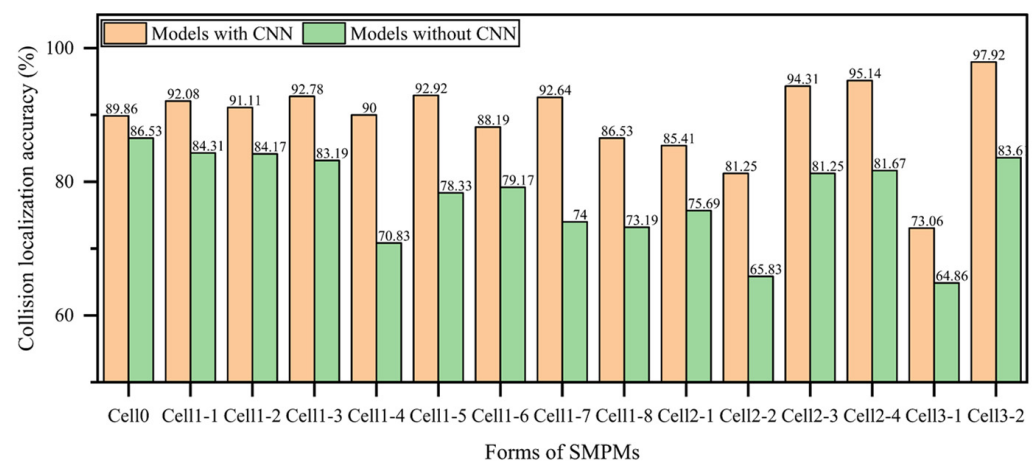
**Figure 5.** Illustration of SMPMs at four corners of the collision point matrix template.



**Figure 6.** Illustration of SMPMs with different distributions. (a) Case with horizontal movement; (b) case with vertical movement; (c) case with movement toward the center.

As shown in Figure 7, the average collision localization accuracy results for the SMPM positioned at the corners of the collision point matrix template are presented. The results labeled “models with CNN” represent the average collision localization accuracy of fusion models incorporating convolutional modules, specifically the DCNN-SVM and CE-SVM methods. Conversely, “models without CNN” correspond to the average collision localization accuracy of models excluding convolutional modules, including ESN-SVM, LSTM-SVM, and GRU-SVM. From the graph, it is evident that fusion models with convolutional modules significantly outperform those relying solely on recursive neural networks for handling collision localization when applied in conjunction with an SMPM. When employing collision localization methods with convolutional modules, the accuracy of SMPMs (Cell 1-1 to Cell 1-8) after removing single points is slightly higher overall than the accuracy achieved by the MPM. In contrast, for collision results obtained using collision localization models without convolutional layers, single-point removal SMPMs are comparatively dis-

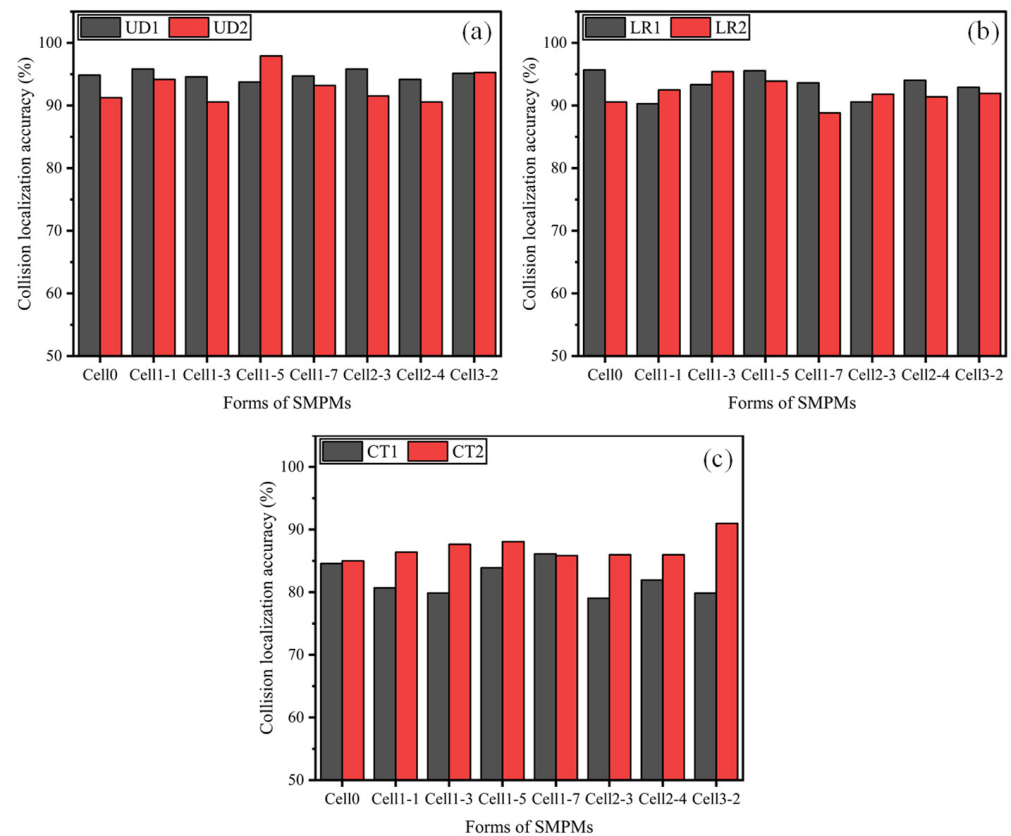
advantaged. This suggests that, when estimating the localization of collisions at zero-shot points, choosing an appropriate model enables the achievement of accuracy levels, even with sample size reduction, equivalent to or higher than those achieved without reducing the sample size. In the case of double-point and quadruple-point removal SMPMs, it is notably observed that when using models with convolutional modules, Cell 2-3, Cell 2-4, and Cell 3-2 achieve significantly higher average collision localization accuracy compared to single-point removal SMPMs. Especially, Cell 3-2 consistently achieves the highest collision accuracy across different sparsification forms. This implies that certain points in the MPM provide redundant or even disruptive information for collision localization. Furthermore, it can be observed that, in the single-point removal SMPM, using models with convolutional modules results in higher collision accuracy for Cell 1-1, Cell 1-3, Cell 1-5, and Cell 1-7 compared to their adjacent counterparts, Cell 1-2, Cell 1-4, Cell 1-6, and Cell 1-8. Similarly, in the double-point removal SMPM, Cell 2-3 and Cell 2-4 achieve significantly higher collision localization accuracy than Cell 2-1 and Cell 2-2, while in the quadruple-point removal SMPM, Cell 3-2 demonstrates markedly higher collision localization accuracy than Cell 3-1. This phenomenon indicates that the sparsification method removing the farthest adjacent points of zero-shot points in the MPM is more effective than removing the nearest adjacent points. Moreover, the standout performance of Cell 3-2 suggests that information from the farthest adjacent points in the MPM may lead to confusion in different regions, resulting in a decrease in collision localization accuracy.



**Figure 7.** Collision localization accuracy with different forms of SMPMs at four corners.

Considering the significant advantages of collision localization models with convolutional modules when combined with an SMPM, we focused solely on utilizing collision localization models with convolutional modules when analyzing the zero-shot point localization situation of the SMPM at different positions within the collision point matrix template. In the selection of SMPM structures, we experimentally chose SMPM structures that exhibited clear advantages at the corners of the collision point matrix template, specifically those removing the farthest adjacent points: Cell 1-1, Cell 1-3, Cell 1-5, Cell 1-7, Cell 2-3, Cell 2-4, and Cell 3-2. For the purposes of comparison with cases without any sparsification, Cell 0 was introduced as a control experiment. Figure 8 depicts the collision localization results based on SMPMs positioned at different locations. The best performance in UD1 and UD2 is observed with Cell 1-5 and Cell 2-3, in LR1 and LR2 with Cell 0 and Cell 1-3, and in CT1 and CT2 with Cell 1-7 and Cell 3-2. Comparing the results of SMPMs with those of the MPM, it is evident that the vibration signals acquired at the farthest neighboring points may indeed contain information that could degrade the localization model's performance. From the average accuracy results of different movement point numbers in UD, LR, and CT, the relative differences in average accuracy for SMPMs removing single points (Cell 1-1, Cell 1-3, Cell 1-5, and Cell 1-7) compared to the MPM are 1.29%, −0.2%, and −0.02%; for SMPMs removing double points (Cell 2-3 and Cell 2-4), they are −0.03%, −1.18%, and −1.56%;

and for SMPMs removing four points (Cell 3-2), they are 2.16%,  $-0.7\%$ , and  $0.62\%$ . This indicates that information contained in some of the farthest adjacent points is not always redundant. However, even with the removal of these points, collision localization accuracy does not significantly decrease compared to using the MPM, suggesting that removing the farthest neighboring points is effective in reducing the dataset size while maintaining high collision localization accuracy. Furthermore, it is noteworthy that the use of the Cell 3-2 sparsification form consistently demonstrates excellent collision localization performance for SMPMs positioned at different locations. This sparsification form, compared to others, minimizes the required data collection scale to the greatest extent. Hence, we consider the Cell 3-2 sparsification form of the SMPM the optimal choice.



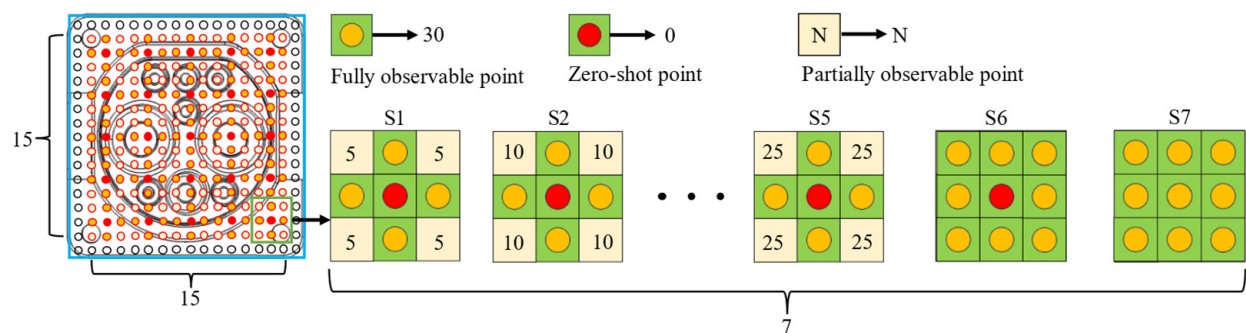
**Figure 8.** Collision localization results using SMPMs with different distributions. (a) UD; (b) LR; (c) CT.

### 3.2. Collision Localization Results across the Entire Template

When applying the SMPM with the Cell 3-2 form for collision localization at the charging port, the encountered challenge is not solely limited to accurately identifying collisions occurring at zero-shot points. Rather, it extends to efficiently locating collisions within the entire collision point matrix template. Therefore, it is imperative for the Cell 3-2 SMPM to be comprehensively deployed across the entire collision point matrix template. In order to investigate the feasibility of the proposed sparsification method across the entire domain, a thorough analysis of the complete collision point matrix template area is required under different robot joint angles. The training datasets used for this purpose are D1 and D2, while the testing dataset is D3, isolated from D1 and D2. Additionally, the collisions occurring at the locations of removed fully observable points also need to be addressed. Despite the theoretical capability of SMPMs to effectively locate collisions at both zero-shot and fully observable points, there is currently a lack of adequate information for handling collisions at removed fully observable points. Hence, we introduce the concept of partially observable points, i.e., collision data sampled with a demand lower than that of fully



observable points but greater than zero. In conjunction with training datasets D1 and D2, the specific forms of SMPMs containing locally observable points used in the experiments are illustrated in Figure 9. To ensure the comprehensive deployment of SMPMs across the entire collision point matrix template, a departure from the approach outlined in [7] is taken. Specifically, the outermost points of the collision point matrix template are disregarded, and analysis is conducted only on the inner 15 rows and 15 columns. Building upon the Cell 3-2 form, partially observable points are positioned at the farthest adjacent point of the MPM. The samples include 30 instances of fully observable points, 0 instances of zero-shot points, and  $N$  instances of partially observable points. To investigate the impact of different sparsity levels of partially observable points on the collision localization results, five different SMPMs are defined, with  $N$  values of 5, 10, 15, 20, and 25, denoted as S1 to S5, respectively. Additionally, for comparison with the case of ample data collection, an MPM (S6) is introduced, and a control experiment (S7) utilizes the collision point matrix template directly without downsampling.



**Figure 9.** Illustration of SMPMs in the Cell3-2 form with varying degrees of sparsification.

In evaluating the effectiveness of applying SMPMs with varying degrees of sparsity to collision localization, we employ assessment criteria that include collision localization accuracy and root mean square error (RMSE). In this study, we treat the SMPM as a unified entity, with its central position representing the estimated positions of individual points within the SMPM. The results of collision localization accuracy are presented in Table 3, where the data scale of S7 is defined as 100%. In S7, the collision localization accuracy of different models exceeded 96%, with the highest reaching 98.67%. As the data scale decreases, the accuracy of collision localization for each method also declines accordingly. At a data scale of 51.85%, the average accuracy of each method only drops by 8.72%. Notably, the CE-SVM method exhibits the smallest decrease, with a reduction of only 7.22%, maintaining an accuracy above 90%. Furthermore, in cases S1 to S6, the CE-SVM method outperforms other methods, especially in situations with higher sparsity, highlighting the pronounced advantage of CE-SVM. Regarding the deviation in collision localization, the RMSEs of various models for collision localization are presented in Table 4. Due to the utilization of the center position of the SMPM as the estimated location for individual points within the SMPM, additional localization biases are introduced in the RMSEs, even when the collision area is correctly predicted. Therefore, the RMSEs of various models are consistently greater than 1 mm. Without any reduction in data scale, the CNN-SVM model achieves the lowest RMSE in collision localization. However, with the introduction of varying degrees of sparsity, the CE-SVM consistently exhibits a notable advantage. When the data scale is reduced to 51.85%, the RMSE of the CE-SVM method increases by only 0.21 mm, compared to the increase in RMSE for CNN-SVM, which is only 55% of its value. In conclusion, it is evident that the SMPM is capable of maintaining a high level of collision localization performance even in scenarios involving significant data scale reduction. From both the collision localization accuracy and RMSEs perspectives, the SMPM maintains a high level of performance even in scenarios of substantial data scale



reduction. Particularly noteworthy is its outstanding collision localization performance when employed in conjunction with the CE-SVM method.

**Table 3.** Collision localization accuracy achieved using SMPMs with different levels of sparsification.

Case	CE-SVM	CNN-SVM	LSTM-SVM	ENS-SVM	GRU-SVM	Data Scale
S1	<b>91.27%</b>	89.29%	88.33%	89.53%	88.73%	51.85%
S2	<b>94.29%</b>	91.79%	90.62%	91.54%	90.8%	59.26%
S3	<b>95%</b>	94.17%	93.24%	93.27%	93.06%	66.67%
S4	<b>96.2%</b>	95%	93.92%	94.2%	93.88%	74.07%
S5	<b>97.07%</b>	95.12%	94.44%	95.22%	94.72%	81.48%
S6	<b>96.73%</b>	95.74%	94.78%	95.65%	95.28%	88.89%
S7	98.49%	<b>98.67%</b>	96.7%	98.64%	98.27%	100%

**Table 4.** Collision localization RMSEs achieved using SMPMs with different levels of sparsification (mm).

Case	CE-SVM	CNN-SVM	LSTM-SVM	ENS-SVM	GRU-SVM
S1	<b>1.46</b>	1.62	1.73	1.63	1.75
S2	<b>1.4</b>	1.52	1.59	1.54	1.61
S3	<b>1.37</b>	1.46	1.51	1.5	1.49
S4	<b>1.33</b>	1.45	1.48	1.47	1.48
S5	<b>1.34</b>	1.39	1.54	1.42	1.42
S6	<b>1.3</b>	1.39	1.45	1.42	1.44
S7	1.25	<b>1.24</b>	1.35	1.26	1.26

#### 4. Conclusions

To achieve higher precision in collision localization, the existing data-driven method for the plug-in process of electric vehicle (EV) automatic charging suffers from high data collection costs. In this study, we propose a novel data-driven approach for robot collision localization specifically tailored to automatic charging scenarios for EVs, effectively mitigating this issue. Our method is grounded in a collision point matrix template and integrates a sparse modular point matrix (SMPM) to reduce the necessary size of the collision dataset for data-driven techniques. By employing an optimized SMPM structure to sparsify the entire template, we achieve a reduction in data scale of 48.15% while maintaining an average localization accuracy of 89.43% and an average RMSE of 1.64 mm. Compared to scenarios without sparsification, the average localization accuracy decreases by only 8.72%, with a minimal increase of 0.37 mm in RMSE for collision localization. Additionally, we exploit the characteristics of convolutional neural network (CNNs) and echo state network (ESNs) to develop an integrated adaptive extractor for dynamic feature extraction from collision vibration signals. Utilizing a support vector machine (SVM) as the classifier, we demonstrate the exceptional performance of the model in addressing collision localization issues when combined with the SMPM. Specifically, even with a 48.15% reduction in data scale, our model achieves an outstanding collision localization accuracy of 91.27% and an RMSE of 1.46 mm for collision localization.

Although our proposed method effectively reduces dataset size while maintaining collision localization performance at a high level, there is still a noticeable decrease in collision localization accuracy when compared to the scenario without any sparsification. In subsequent research, we will focus on exploring whether data augmentation techniques can be employed to generate data for sparsified points, creating a virtual supplement to the dataset. We aim to enhance the performance of the collision localization method further while reducing the need for experimental data acquisition.

**Author Contributions:** H.L. developed the methodology; H.L. conceived and designed the experiment; Z.L. and P.Q. conducted the data curation and collection; H.L. wrote the original draft of the paper; S.D. and D.W. reviewed and edited the paper. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The raw data supporting the conclusions of this article will be made available by the authors on request.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Pan, M.; Sun, C.; Liu, J.; Wang, Y. Automatic Recognition and Location System for Electric Vehicle Charging Port in Complex Environment. *IET Image Process.* **2020**, *14*, 2263–2272. [\[CrossRef\]](#)
2. Zhang, H.; Zhu, W.; Huang, Y. A Research on the Control Strategy of Automatic Charging Robot for Electric Vehicles Based on Impedance Control. *J. Phys. Conf. Ser.* **2022**, *2303*, 012085. [\[CrossRef\]](#)
3. Lin, H.; Quan, P.; Liang, Z.; Lou, Y.; Wei, D.; Di, S. Collision Localization and Classification on the End-Effector of a Cable-Driven Manipulator Applied to EV Auto-Charging Based on DCNN-SVM. *Sensors* **2022**, *22*, 3439. [\[CrossRef\]](#)
4. Lin, H.; Quan, P.; Liang, Z.; Lou, Y.; Wei, D.; Di, S. Precision Data-Driven Collision Localization with a Dedicated Matrix Template for Electric Vehicle Automatic Charging. *Electronics* **2024**, *13*, 638. [\[CrossRef\]](#)
5. Vorndamme, J.; Schappler, M.; Haddadin, S. Collision Detection, Isolation and Identification for Humanoids. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 4754–4761.
6. Iskandar, M.; Eiberger, O.; Albu-Schaffer, A.; Luca, A.D.; Dietrich, A. Collision Detection, Identification, and Localization on the DLR SARA Robot with Sensing Redundancy. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021.
7. Zurlo, D.; Heitmann, T.; Morlock, M.; De Luca, A. Collision Detection and Contact Point Estimation Using Virtual Joint Torque Sensing Applied to a Cobot. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May 2023; pp. 7533–7539.
8. Piacenza, P.; Dang, W.; Hannigan, E.; Espinal, J.; Hussain, I.; Kymissis, I.; Ciocarlie, M. Accurate Contact Localization and Indentation Depth Prediction with an Optics-Based Tactile Sensor. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017.
9. Fan, X.; Lee, D.; Jackel, L.; Howard, R.; Lee, D.; Isler, V. Enabling Low-Cost Full Surface Tactile Skin for Human Robot Interaction. *IEEE Robot. Autom. Lett.* **2022**, *7*, 1800–1807. [\[CrossRef\]](#)
10. Mittendorfer, P.; Yoshida, E.; Cheng, G. Realizing Whole-Body Tactile Interactions with a Self-Organizing, Multi-Modal Artificial Skin on a Humanoid Robot. *Adv. Robot.* **2015**, *29*, 51–67. [\[CrossRef\]](#)
11. Li, X.; Zhang, Y.; Xie, X.; Li, J.; Shi, G. Improving Robotic Tactile Localization Super-Resolution via Spatiotemporal Continuity Learning and Overlapping Air Chambers. *AAAI* **2023**, *37*, 6192–6199. [\[CrossRef\]](#)
12. Popov, D.; Klimchik, A.; Mavridis, N. Collision Detection, Localization & Classification for Industrial Robots with Joint Torque Sensors. In Proceedings of the 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), Lisbon, Portugal, 28–31 August 2017; pp. 838–843.
13. Ha, X.T.; Wu, D.; Lai, C.-F.; Ourak, M.; Borghesan, G.; Menciassi, A.; Poorten, E.V. Contact Localization of Continuum and Flexible Robot Using Data-Driven Approach. *IEEE Robot. Autom. Lett.* **2022**, *7*, 6910–6917. [\[CrossRef\]](#)
14. Min, F.; Wang, G.; Liu, N. Collision Detection and Identification on Robot Manipulators Based on Vibration Analysis. *Sensors* **2019**, *19*, 1080. [\[CrossRef\]](#)
15. McMahan, W.; Romano, J.M.; Kuchenbecker, K.J. Using Accelerometers to Localize Tactile Contact Events on a Robot Arm. In Proceedings of the Workshop on Advances in Tactile Sensing and Touch-Based Human-Robot Interaction, ACM/IEEE International Conference on Human-Robot Interaction, Boston, MA, USA, 5–8 March 2012.
16. Livieris, I.E.; Pintelas, E.; Pintelas, P. A CNN-LSTM Model for Gold Price Time-Series Forecasting. *Neural Comput. Appl.* **2020**, *32*, 17351–17360. [\[CrossRef\]](#)
17. Sun, Y.; Xue, B.; Zhang, M.; Yen, G.G.; Lv, J. Automatically Designing CNN Architectures Using the Genetic Algorithm for Image Classification. *IEEE Trans. Cybern.* **2020**, *50*, 3840–3854. [\[CrossRef\]](#)
18. Zha, W.; Liu, Y.; Wan, Y.; Luo, R.; Li, D.; Yang, S.; Xu, Y. Forecasting Monthly Gas Field Production Based on the CNN-LSTM Model. *Energy* **2022**, *260*, 124889. [\[CrossRef\]](#)
19. Jaeger, H. Adaptive Nonlinear System Identification with Echo State Networks. In Proceedings of the Advances in Neural Information Processing Systems 15 (NIPS 2002), Vancouver, BC, Canada, 9–14 December 2002.

20. Hua, Z.; Zheng, Z.; Péra, M.-C.; Gao, F. Remaining Useful Life Prediction of PEMFC Systems Based on the Multi-Input Echo State Network. *Appl. Energy* **2020**, *265*, 114791. [[CrossRef](#)]
21. Montaha, S.; Azam, S.; Rafid, A.K.M.R.H.; Hasan, M.Z.; Karim, A.; Islam, A. TimeDistributed-CNN-LSTM: A Hybrid Approach Combining CNN and LSTM to Classify Brain Tumor on 3D MRI Scans Performing Ablation Study. *IEEE Access* **2022**, *10*, 60039–60059. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.