

## Article

# STEAM: Spatial Trajectory Enhanced Attention Mechanism for Abnormal UAV Trajectory Detection

Namkyung Yoon <sup>†</sup>, Dongjae Lee <sup>†</sup>, Kiseok Kim <sup>†</sup>, Taehoon Yoo , Hyeontae Joo  and Hwangnam Kim <sup>\*</sup>

School of Electrical Engineering, Korea University, Seoul 02841, Republic of Korea; nkyoon93@korea.ac.kr (N.Y.); djrandy@korea.ac.kr (D.L.); kisuk528@korea.ac.kr (K.K.); yth921@korea.ac.kr (T.Y.); motern800@korea.ac.kr (H.J.)

<sup>\*</sup> Correspondence: hnkim@korea.ac.kr; Tel.: +82-2-3290-4821

<sup>†</sup> These authors contributed equally to this work.

**Abstract:** Accurate unmanned aerial vehicle (UAV) trajectory tracking is crucial for the successful execution of UAV missions. Traditional global positioning system (GPS) methods face limitations in complex environments, and visual observation becomes challenging with distance and in low-light conditions. To address this challenge, we propose a comprehensive framework for UAV trajectory verification, integrating a range-based ultra-wideband (UWB) positioning system and advanced image processing technologies. Our key contribution is the development of the Spatial Trajectory Enhanced Attention Mechanism (STEAM), a novel attention module specifically designed for analyzing and classifying UAV trajectory patterns. This system enables real-time UAV trajectory tracking and classification, facilitating swift and accurate assessment of adherence to predefined optimal trajectories. Another major contribution of our work is the integration of a UWB system for precise UAV location tracking, complemented by our advanced image processing approach that includes a deep neural network (DNN) for interpolating missing data from images, thereby significantly enhancing the model's ability to detect abnormal maneuvers. Our experimental results demonstrate the effectiveness of the proposed framework in UAV trajectory tracking, showcasing its robust performance irrespective of raw data quality. Furthermore, we validate the framework's performance using a lightweight learning model, emphasizing both its computational efficiency and exceptional classification accuracy.

**Keywords:** UAV trajectory; ultra-wideband; Kalman filter; image interpolation; attention mechanism



**Citation:** Yoon, N.; Lee, D.; Kim, K.; Yoo, T.; Joo, H.; Kim, H. STEAM: Spatial Trajectory Enhanced Attention Mechanism for Abnormal UAV Trajectory Detection. *Appl. Sci.* **2024**, *14*, 248. <https://doi.org/10.3390/app14010248>

Academic Editor: Yosoon Choi

Received: 21 November 2023

Revised: 21 December 2023

Accepted: 26 December 2023

Published: 27 December 2023



**Copyright:** © 2023 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

Unmanned aerial vehicles (UAVs) can perform a variety of important tasks, replacing human labor in tasks such as escort, patrol, security, delivery, and search and rescue [1]. The optimal UAV flight trajectory may differ depending on the purpose of the mission performed by the UAV [2]. Accurately flying the predefined flight trajectory appropriate for the purpose of the mission are very important in terms of the purpose and efficiency of UAV operation [3,4]. However, it can be difficult to quickly and accurately check the flight trajectory [5,6].

The most basic conventional method to determine the location of a UAV is to use Global Positioning System (GPS), which utilizes satellites. However, GPS has the disadvantage that reception signals can be weak in complex urban terrain and cannot be used in indoor environments covered by architectural structures [7]. The way to check UAV flying with the naked eye is difficult to grasp the overall trajectory if the distance from the UAV increases, and it can be difficult to recognize UAV in low-light environments. Lastly, the method of attaching a camera to a UAV to determine the location of a UAV may also be incomplete due to unstable network connections in disaster situations, which are environments where UAVs are mainly used [8,9]. As UAVs often have to be operated in environments with incomplete network connectivity and have high mobility, frequent network communication losses may

occur during flight. In addition, the flight data recorder (FDR) of flight controllers such as ArduPilot Mega (APM) and Pixhawk can be used to record flight history by utilizing the UAV's own sensors, but accurate reproduction of 3D trajectories is limited by empirical problems, such as sampling frequency constraints or network delay and loss [10].

The ultra-wideband (UWB) positioning system is a range-based positioning method with relatively simple infrastructure configuration and high accuracy. The strength of UWB systems is that they use high frequency bandwidth and have low signal interference. Additionally, the UWB system has the advantage of enabling high-precision positioning using high nanosecond time resolution and can be effectively used both indoors and outdoors over long distances [11].

In addition, image processing technology is one of the core technologies widely used in the autonomous driving, medical, security, and entertainment industries by configuring and optimizing nonlinear modeling through a multi-layer neural network structure [12,13]. Recently, practical research is being conducted on unmanned aerial vehicles (UAVs), one of the many fields where such image processing can be utilized [14–17]. Among the many ongoing studies, there are increasing attempts to image the flight trajectory of a UAV and use it for performance verification [18] and failure diagnosis [19].

In general, UAV flight trajectory records are a key element in the UAV mission performance verification process to evaluate the accuracy and efficiency of UAV mission performance [20–23]. However, if frequent network communication loss occurs, a large portion of the flight trajectory records used to verify the UAV's mission performance may be lost, making accurate evaluation impossible. Therefore, in this study, we design a system that tracks the position of a UAV in real time by introducing location determination using this UWB positioning system into the UAV trajectory tracking process. Additionally, we propose a framework that can interpolate and evaluate UAV flight trajectories without loss by utilizing a deep interpolation model and spatial trajectory enhancement attention mechanism (STEAM), even in a limited network infrastructure environment based on UAV flight trajectory images.

The proposed system involves several key steps. First, we introduce a deep neural network (DNN) technique to interpolate missing data based on images. This approach is crucial as data quality can deteriorate due to various external factors, such as noise during data collection or omissions in data transmission and reception. Making predictions with low-quality data as input challenges the accuracy of even the most sophisticated DNN models. Traditional methodologies, based on statistics and probability, often interpolate data collected from sensors by focusing on numerical data. However, when the degree of missingness in data is significant, these traditional methods risk distorting the original data, rendering the interpolation process ineffective. To address this, our system leverages advanced DNN-based interpolation techniques that use images instead of numerical data, filling gaps in the data more effectively. This approach not only mitigates the degradation of data quality but also significantly improves the classification performance of the UAV abnormal maneuver detection model.

Secondly, we propose the STEAM, a new and efficient attention module optimized for analyzing trajectory pattern data of UAVs. STEAM adopts a maximum-pooling-based attention strategy to emphasize prominent trajectory features with high precision. This strategy effectively minimizes the number of parameters in the model, thereby reducing model complexity and enhancing interpretability. Our comprehensive experiments demonstrate superior performance of STEAM over traditional convolutional models, especially in UAV trajectory classification tasks, where STEAM shows improved accuracy with fewer computational resources.

Our system, using the above proposed methods, facilitates real-time UAV flight trajectory tracking and classification, enabling rapid and accurate evaluation of UAVs for predefined optimal trajectories. The results obtained from this process can pinpoint UAVs requiring flight trajectory modifications and facilitate efficient monitoring for precise task execution. Moreover, the ability to classify flight trajectories allows for the identification of

specific tasks being performed by the UAVs. This feature of the system extends its utility beyond mere monitoring, making it applicable to a wide range of unmanned vehicles requiring optimized routing for specific tasks.

Our contributions can be summarized as follows:

- We introduced an effective interpolation technique utilizing partial convolution with a suitable loss function.
- The proposed attention-based CNN classifier not only achieves outstanding classification performance but also features a lightweight learning model.
- We presented a framework for the validation and assessment of UAV trajectories, integrating a hybrid model selection approach.

The remainder of this paper is organized as follows. We introduce related work in Section 2. In Section 3, we provide a detailed explanation about our visualization system, deep interpolation model, and attention mechanism. Section 4 evaluates the performance of the proposed framework. Finally, Section 5 concludes this paper.

## 2. Related Work

Since we aim to complement and classify incomplete flight sensor data obtained from UAV sensors for a more accurate trajectory cognition, we present efforts to enhance the accuracy of flight pattern recognition or coverage path planning, such as attempts to apply DNN methods, classification approaches, and filtering algorithms for image interpolation used in our proposed method.

Ü. Çelik et al. applied two-dimensional manifold learning to obtain individual flight patterns from UAV sensor data to classify flight fingerprinting [24]. B. Wang et al. proposed a solution distinguishing four flight modes—ascend, descent, horizontal flight, and rotation—using a hybrid approach of UAV flight mode recognition and Gaussian Process-Unscented filter [25]. Several studies delved deep into flight trajectory patterns. M. Aksoy et al. proposed predicting flight categories using long short-term memory (LSTM) autoencoder for flight trajectory patterns and generating trajectory samples through a generative adversarial network (GAN) for flight trajectory generalization [26]. W. Dong et al. proposed an interpolation technique using optimal curvature smoothing and trajectory generation [27]. G. Ramos et al. applied extended Kalman filter to enhance trajectory tracking precision for UAVs in the maritime industry and attempted sensor fusion [28]. However, most of these studies are limited by their focus on traditional aircraft trajectories, lacking specificity for UAVs' dynamic and diverse operational patterns. For instance, W. Zeng et al. used deep autoencoder and Gaussian mixture model for aircraft trajectory clustering [29]. G. Vladimir et al. classified Aeroballistic Flying Vehicle paths using artificial neural networks [30]. However, to address the huge demand in air traffic in the future, more specific aircraft trajectory classification beyond standardized normal trajectory categories has recently emerged. A. Mcfadyen et al. attempted aircraft trajectory clustering using a circular statistic approach [31]. C. Gingrass et al. proposed classifying flight patterns into nine categories using a deep sequential neural network from flight path data based on automatic dependent surveillance broadcast (ADS-B) [32]. There have been similar attempts targeting ships rather than aircraft; Q. Dong et al. introduced the Mathematical–Data Integrated Prediction (MDIP) model for predicting ship maneuvers, blending mathematical forecasting using an extended Kalman filter with data-driven predictions via least squares support vector machines, showcasing improved generalization in various maneuvering tests [33]. N. Wang et al. presented SeaBil, a deep-learning-based system using a self-attention-weighted Bi-LSTM and Conv-1D network for ultrashort-term prediction of ship maneuvering, demonstrating higher accuracy in predicting various ship motion states compared to other methods [34]. These approaches, while effective for conventional aircraft, do not adequately address the unique challenges presented by UAVs, particularly in cluttered or unstructured environments.

On the other hand, the issue of limited sensor data sampling for distinguishing flight trajectory categories leads to problems with padding or inpainting [35–38]. G. Liu et al.

showed consistent effective performance of partial convolution binary masks in various tasks related to inpainting and padding [39]. D. Xing et al. effectively applied partial convolution for complementing missing geographic information of spatial data obscured by clouds [40]. However, these methods often fail to address the severe data loss scenarios typical in UAV operations, where data accuracy is paramount. Therefore, we attempt interpolation for trajectories with high data loss to fully utilize the advantages of partial convolution previously mentioned.

Attention mechanisms provide the model with the ability to selectively focus on the parts of the input that are most relevant during the learning process in deep learning architectures. Initially inspired by human cognitive attention, this mechanism has been applied to a broad array of applications, from natural language processing to computer vision [41]. In the realm of image analysis, attention mechanisms are typically explored to enhance expressive power by amplifying the importance of different spatial regions or feature channels [42].

The evolution of attention mechanisms in the field of deep learning aims to improve model interpretability and performance. Universal early forms of attention mechanisms are designed to enhance the performance of neural networks for tasks such as machine translation by focusing on relevant segments of the input sequence [43]. This concept expanded into the spatial domain of computer vision, evolving into sophisticated forms like the squeeze-and-excitation network (SE) and the convolutional block attention module (CBAM), which combine spatial and channel-specific attention [44,45]. Additionally, the advent of vision transformers (ViT) marked a significant shift in attention-based models. ViT applies the self-attention mechanism, originally successful in natural language processing, to computer vision. It segments images into patches, processes them as sequences, and learns by focusing on interactions between arbitrary image parts. Separated from traditional attention mechanisms, this approach replaces the CNN structure as a whole with transformers to handle the embedding sequence [46]. Models like CBAM and ViT have shown high performance in various fields through extensive experimentation and validation [47]. However, conventional models have not been studied for data representing UAV trajectory and movements and are not ideal for capturing subtle differences between static and dynamic elements within such images. We therefore emphasize the need for specialized attention mechanisms for UAV trajectories. To bridge this gap, our study introduces the Spatial Trajectory Enhanced Attention Mechanism (STEAM), specifically designed for UAV trajectory analysis. This approach addresses the shortcomings of existing models in handling the unique challenges of UAV flight pattern data.

The section that follows presents new solutions that provide an efficient and highly analytical UAV trajectory analysis model and compares performance with existing methods through experiments.

### 3. System Design

In this paper, we propose an integrated UAV maneuver detection system that visualizes and interpolates the UAV's trajectory calculated from UWB sensor data and detects the UAV's flight pattern using an attention mechanism. Our approach is structured to ensure the high accuracy and reliability of UAV flight pattern detection, which is paramount for the successful execution of UAV missions in complex environments. Our ultimate goal is to accurately detect the flight pattern of a UAV, and, for this, the quality of input data and the performance of the trained classification model are critically important. Figure 1 shows the process of the proposed overall system, which comprises three main stages: preprocessing, image-based interpolation, and UAV flight pattern detection using STEAM. First, a preprocessing stage generates a UAV trajectory based on data collected from UWB sensors and converts numerical data into images. This stage converts the numerical data into images, forming the foundation for subsequent analysis. The effectiveness of our system starts here, ensuring that the raw data are accurately transformed into a format suitable for deep learning applications.

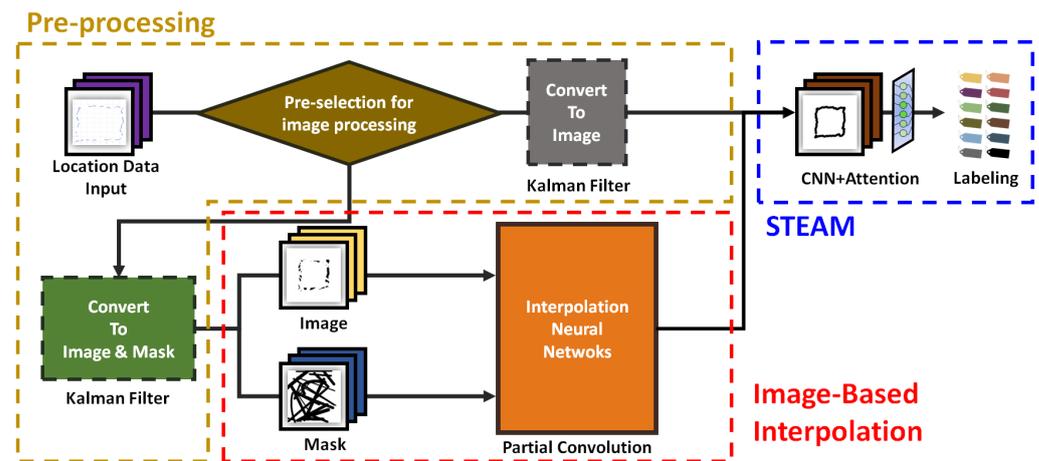


Figure 1. This is the proposed interpolation scheme using DNN.

Second, an image-based interpolation stage performs interpolation of severe missing trajectory images with DNN. This approach effectively mitigates issues related to data quality degradation from external factors, like noise and data omissions. By using image data for interpolation rather than numerical data, we enhance both the data quality and the performance of the UAV abnormal maneuver detection model.

Third is a STEAM stage that detects UAV flight patterns using a DNN classification model with an attention mechanism. STEAM uses a maximum-pooling-based attention strategy to precisely highlight key trajectory features, reducing model complexity and improving interpretability. Our experiments validate that STEAM surpasses traditional convolutional models in accuracy, particularly in UAV trajectory classification, and does so with less computational demand. In this section, we list the core technologies of the proposed system according to the process flow and provide specific explanations of how each technology is applied.

### 3.1. Data Visualization

In the proposed system, the UWB technology plays an important role in recognizing the trajectory by accurately determining the location of the UAV. The 2D localization system, which utilizes UWB sensors, comprises at least three anchors and one tag. Anchors are fixed in known locations, and tags are attached to UAVs. The localization process initiates with the TWR (Two-Way Ranging) protocol, marked by the exchange of signals between the anchor and the tag. The TWR protocol consists of multi-message exchange, starting with the sending of a poll message from the tag to the anchor. After that, a response message is transmitted from the anchor, and, finally, a final message is transmitted from the tag to the anchor. The time interval between the transmission and reception of these messages is accurately measured, and it is used with the speed of light to accurately measure the distance between the tag and the anchor. This process is depicted in Figure 2. The following is an equation for calculating the distance between one anchor and one tag using Time of Flight (ToF) and the speed of light.

$$T_{\text{tof}} = \frac{t_{\text{round1}} \times t_{\text{round2}} - t_{\text{reply1}} \times t_{\text{reply2}}}{t_{\text{round1}} + t_{\text{round2}} + t_{\text{reply1}} + t_{\text{reply2}}}, \quad (1)$$

$$d = c \times T_{\text{tof}}.$$

In the provided formula, the calculation of distance  $d$  between two devices uses Time of Flight (ToF), denoted as  $T_{\text{tof}}$ . ToF represents the time taken by a signal to travel from the sender to the receiver and back. This time is crucial for determining the distance between devices using the speed of light  $c$ . The formula involves four key time measurements:

- $t_{\text{round1}}$ : Time for the signal to travel from tag to anchor and back to tag.

- $t_{round2}$ : Time for the signal to travel from anchor to tag and back to anchor.
- $t_{reply1}$ : Time taken by anchor to process and reply to the signal from tag.
- $t_{reply2}$ : Time taken by tag to process and reply to the signal from device anchor.

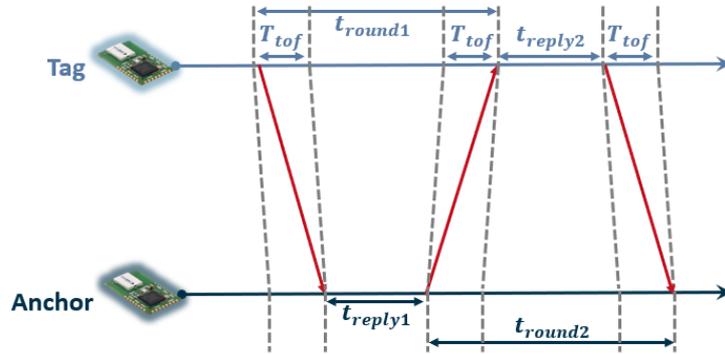


Figure 2. Two-way ranging with time-of-flight calculation diagram.

The product and sum of these times are used in the formula to calculate the ToF, which is then multiplied by the speed of light  $c$  to find the distance  $d$  between the devices. In the system described, each anchor transmits distance data to the server using the TWR protocol. The server then estimates the exact location of the tag by employing a trilateration formula. The formula is as follows:

$$\begin{aligned} (x - x_a)^2 + (y - y_a)^2 &= d_a^2, \\ (x - x_b)^2 + (y - y_b)^2 &= d_b^2, \\ (x - x_c)^2 + (y - y_c)^2 &= d_c^2, \end{aligned} \tag{2}$$

where  $x$  and  $y$  represent the target position coordinates. The coordinates of the three anchors are denoted as  $(x_a, y_a)$ ,  $(x_b, y_b)$ , and  $(x_c, y_c)$ , and  $d_a$ ,  $d_b$ , and  $d_c$  are the distances from each respective anchor to the target. By combining these expressions, we derive the following equations:

$$\begin{aligned} A &= -2x_a + 2x_b, & B &= -2y_a + 2y_b, & C &= d_a^2 - d_b^2, \\ D &= -2x_b + 2x_c, & E &= -2y_b + 2y_c, & F &= d_b^2 - d_c^2, \\ Ax + By &= C, & Dx + Ey &= F. \end{aligned} \tag{3}$$

Solving these equations for  $x$  and  $y$  yields

$$x = \frac{CE - BF}{AE - BD}, y = \frac{AF - CD}{AE - BD}. \tag{4}$$

Therefore, with known anchor coordinates and distances, the unknown coordinates  $x$  and  $y$  can be accurately determined.

The location data of the UAV obtained from the UWB sensors is essential in constructing the UAV trajectory. However, since the data are time series data, additional processing is required to effectively track the trajectory. In the proposed system, location data are smoothed through a Kalman filter to reduce noise and improve the quality of the data. After that, the data are converted into an image and processed so that the trajectory can be effectively checked.

Widely used to minimize uncertainty, a Kalman filter analyzes data by recursively estimating the current state of the system based on measurement data mixed with noise. Therefore, by applying the Kalman filter to the location data estimated from the previous process, the effect of smoothing the data is obtained by reducing the noise of the data. This reduces the volatility of the location data, thereby obtaining a more accurate trajectory. This

process of data smoothing is based on a mathematical model of the Kalman filter, which describes the dynamic behavior of the system and takes into account measurement noise and process noise. Kalman filters are defined as follows:

$$\begin{aligned}
 \hat{x}_{k|k-1} &= A\hat{x}_{k-1|k-1} + Bu_k, \\
 P_{k|k-1} &= AP_{k-1|k-1}A^T + Q, \\
 K_k &= P_{k|k-1}H^T(HP_{k|k-1}H^T + R)^{-1}, \\
 \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k(z_k - H\hat{x}_{k|k-1}), \\
 P_{k|k} &= (I - K_kH)P_{k|k-1}.
 \end{aligned} \tag{5}$$

In the first equation,  $\hat{x}_{k|k-1}$  represents the predicted state vector at time  $k$  based on information available up to time  $k - 1$ . Matrix  $A$  is the state transition model that is applied to the previous state  $\hat{x}_{k-1|k-1}$ , while  $B$  is the control input model that operates on the control vector  $u_k$ . The second equation predicts the covariance of the predicted state, where  $P_{k|k-1}$  is the estimate of the covariance of the state vector at time  $k$ . Matrix  $Q$  is the process noise covariance matrix, reflecting the uncertainty in the model. The third equation calculates Kalman gain  $K_k$ , which weighs the importance of the measurements versus the current state estimate. Here,  $H$  is the observation model that maps the true state space into the observed space.  $R$  is the measurement noise covariance matrix, quantifying the uncertainty in sensor measurements. Through the next equation, the state estimate is updated by incorporating the new measurement  $z_k$ . The term  $z_k - H\hat{x}_{k|k-1}$  represents the measurement residual, or the difference between the actual measurement and what the model predicted. Finally, the estimate of the covariance of the state vector is updated through the last equation. This step reflects the reduction in uncertainty as a result of the measurement update. In our application, matrices  $Q$  and  $R$  are empirically set to 0.0001 and 0.1, respectively. These values are chosen to balance the model's sensitivity to the process noise and measurement noise. Through the iterative application of these equations, the Kalman filter provides refined estimates of the system state  $x_k$ , continually updating the state and measurement covariance matrices  $P_k$  and  $P_{k-1}$ . This results in a more accurate and less volatile trajectory estimation for the UAVs, even in the presence of noisy and uncertain measurements.

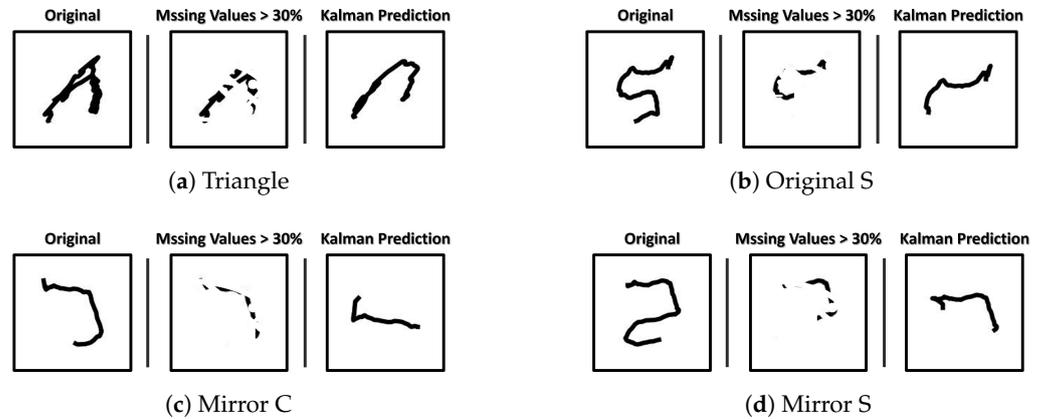
It converts the location data that have been applied to the Kalman filter into an image, making it possible to effectively check the trajectory of the UAV. This process corresponds to each location dataset as a pixel value between 0 and 255 to generate an image having a size of  $159 \times 158$  pixels. By going through this process, it is possible to effectively check the UAV's trajectory, which was difficult to confirm with numerical data, and make the CNN model's learning process more efficient and accurate, contributing to the improvement in UAV trajectory classification performance.

### 3.2. Deep Interpolation Model

This section delves into the process of enhancing the DNN model's accuracy by refining the quality of the input dataset. This work includes the process of gathering UWB sensor data collected from edge devices to a central server, calculating the location data for each UAV from the central server, and visualizing the movement flight path. However, this process makes it difficult to collect data reliably for environmental reasons, and, in some cases, it can cause a large amount of missing values. This can ultimately lead to a decrease in the performance of the classification model. To solve this, we propose an advanced technique to interpolate these missing values.

The Kalman filter stands out as a prominent interpolation technique, but its restoration capabilities are constrained by the extent of missing location data. Figure 3 shows the performance of the Kalman filter that varies depending on the missing rate of the location data that make up the trajectory. The trajectory depicted in Figure 3 is derived using the method outlined in Section 3.1. For this purpose, a Raspberry Pi equipped with a UWB sensor was utilized to simulate a UAV flying with the same sensor. When the distribution density of the original location data is high or the missing data are more than 50% of

the original datasets, the restored trajectory gradually loses the original meaning. These distorted datasets can result in completely different predicted values from the classification model, which can have a negative impact on our research to precisely monitor flight patterns and detect anomalies.



**Figure 3.** This is the Kalman filter’s prediction for more than 30% missing trajectory.

We propose a new interpolation methodology that can respond to complex and diverse interpolation problems with consistent performance and without distortion of the original data. Our proposal enables complex and sophisticated calculations through DNN, and operates so that the restored data are very similar to the original data even if serious data loss occurs. Figure 1 emphasizes the architecture of the proposed interpolation methodology with a red square. First, our suggestion performs interpolation for missing values based on images rather than numerical data. This image refers to data expressed in the form of an image by visualizing the trajectory composed of the location data of the UAVs calculated from the UWB sensor data covered in Section 3.1. Interpolation is not performed for all images, and the first criterion for judging interpolation for a single image is expressed as follows:

$$N = \frac{n_{total} - n}{n_{total}}, \tag{6}$$

where  $n_{total}$  denotes the total number of location data before data are lost, and  $n$  denotes the total number of location data after data are lost. The missing rate  $N$  is expressed as the number of missing coordinates divided by the total number of location data. For more precise control, we calculate the scaling constant  $M$  and it is expressed as

$$n_{total} = \mu t, \tag{7}$$

$$M = \log(\mu), \tag{8}$$

where  $\mu$  denotes the number of UWB sensor data transmitted per second from the anchor to the server and  $M$  refers to the value expressed in the logarithmic scale of lambda. Even if the  $N$  is the same, when the location coordinate generation rate is high, the degree of loss can be considered more serious because the amount of lost data is greater. For this reason, we define the interpolation value  $K$  as follows:

$$K = N * M. \tag{9}$$

Our proposal is designed to perform interpolation only if the value of  $K$  is greater than the specified interpolation threshold  $K_{th}$ . This is to reduce resource allocation and computing load resulting from unnecessary interpolation because deep neural networks for image processing require computationally intensive operations. In summary, we determine the necessity for interpolation through Equation (9) for each of the visualized trajectories and apply the DNN model described next only to cases where interpolation is necessary.

We use a DNN model that shows excellent performance in the problem of filling empty spaces called a hole in the image as an interpolation method. Our proposal begins with the idea that, when arranging the location data calculated based on the information collected from the UWB sensor in chronological order, missing values, which may occur for various reasons, are ultimately represented as holes in the images derived during the trajectory visualization process, and we can use an appropriate DNN to fill these holes through a rational process based on surrounding information. We can generally use convolution for image processing, but we consider a few more considerations. Since our data are not a continuous image, we do not have enough information to describe the context, and basically we need as much information as possible around the damaged pixels to fill the holes in the image, but our image sets do not have enough information to describe the UWB trajectory. It contains very little information and the information available is limited. For this reason, the use of standard convolutions can cause various artifacts such as blurriness and color discrepancy in the results. However, through research, we determined that partial convolutions, an advanced technique for image inpainting, would have excellent performance in our case, and the application results are shown in Figure 4.

The trajectory shown in Figure 4 is derived from simulation data, which include the coordinates of the UAV's location, following the methodology outlined in Section 3.1. Figure 4c shows that interpolation close to the original image is achieved even though the loss rate is almost 70–80%, and, when compared to the results using the Kalman filter in Figure 3, interpolation using partial convolutions shows better performance.

The base network of the interpolation neural network we consider has a U-net-based architecture, as shown in Figure 5, which can be seen as the most general network structure for image inpainting. The main difference in our network from a traditional U-net lies in the adoption of partial convolution across all layers instead of standard convolution. Partial convolution operates with a mechanism that gradually fills in holes within the original image by referencing a mask containing information about these holes [48]. As the input image traverses successive partial convolutional layers structured in a U-net architecture, the vacant areas progressively become filled starting from the periphery, and the value of each hole is computed based on surrounding pixel information. The first layer of our network receives as input a visualized UWB trajectory and a mask that covers the holes due to missing values and part of the surrounding area. After this, when performing partial convolution while sliding, convolution is applied only to the masked features of the image, and the renormalized image is passed as the input image of the next layer. At the same time, the effective region of the inpainted mask is updated and this is also passed on to the next layer. Afterwards, as it goes through stacked partial convolution layers, the previous operations are repeated identically, and, as a result, holes in the image corresponding to missing values are gradually filled. Our loss function is expressed as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{valid} + 10\mathcal{L}_{hole} + 0.05\mathcal{L}_{perceptual} + 60(\mathcal{L}_{style_{out}} + \mathcal{L}_{style_{comp}}) + 0.8\mathcal{L}_{tv}. \quad (10)$$

Due to the nature of the data, it was necessary to focus on hole restoration accuracy and hole boundary information, and the weights of related loss terms were set through empirical search.

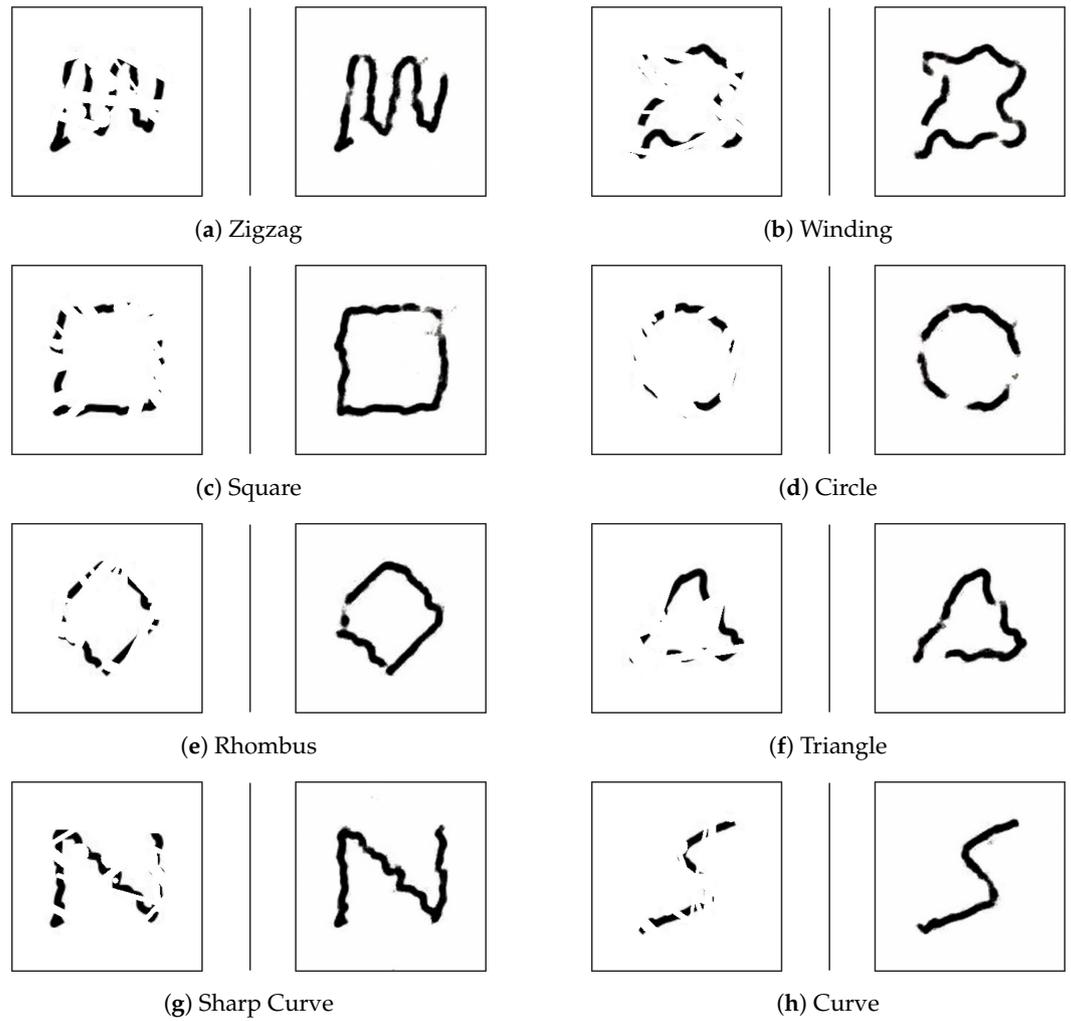


Figure 4. These are the results of image-based DNN interpolation.

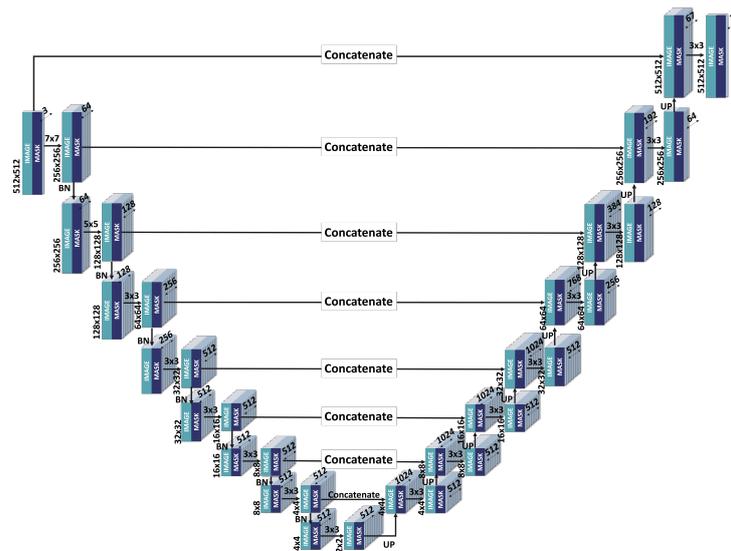


Figure 5. This is the network structure of the proposed interpolation methodology.

### 3.3. Attention Mechanism

Attention mechanisms enhance the performance of neural network models by focusing on the features of important data during the learning process. The STEAM for trajectory

patterns such as UAV surveillance integrates an approach that effectively captures both the features of trajectories and contextual backgrounds and then dynamically weights them to perform spatial attention mapping.

The STEAM process can be expressed as follows:

$$T = f_{\text{trajectory}}(X), \quad (11)$$

$$C = f_{\text{context}}(X), \quad (12)$$

$$F = \text{concatenate}(T, C), \quad (13)$$

$$D = f_{\text{dynamic}}(F), \quad (14)$$

$$A = f_{\text{spatial}}(D), \quad (15)$$

$$X' = X \odot A. \quad (16)$$

where  $X$  is the input feature map,  $T$  and  $C$  are, respectively, trajectory and context features extracted using convolutional layers,  $F$  is their combination,  $D$  represents dynamically weighted features,  $A$  is the attention map using convolutional layer, and  $X'$  is the final attended feature map.

As shown in Equations (11) and (12), two separate convolutional layers are used to extract detailed features of the trajectory and to capture the surrounding context. We then dynamically weigh them to produce a balanced representation of both trajectory and context, as shown in Equation (14). Finally, we generate a spatial attention map through the convolutional layer that highlights the areas of the images most relevant to understanding the trajectory of UAVs, as shown in Equation (15). As a result, the attended feature map generated by Equation (16) allows the model to focus on the most important aspects of the trajectory, improving the interpretability and performance of the neural network model for the trajectory pattern.

Methods such as SE and CBAM focus on channel relationships or provide sequential spatial and channel-specific attention. These are methods of focusing throughout the channel level or emphasizing important spatial areas in the feature map. In contrast, STEAM specifically targets dual aspects: it not only emphasizes the trajectory features of UAV movements but also pays particular attention to the contextual backgrounds. This dual focus is crucial, especially in cases where UAV trajectories are overlaid onto diverse or complex backgrounds. Unlike CBAM, which applies channel and spatial attention in a sequential manner and focuses on either trajectory or contextual information at a time, STEAM employs a more integrated approach. It simultaneously processes both trajectory and contextual information, ensuring a more comprehensive analysis of UAV trajectory images. This distinction allows STEAM to effectively handle the nuances and complexities inherent in UAV surveillance imagery, where both the path of the UAV and its interaction with the surrounding environment are critical for accurate interpretation.

Existing methods can be widely applied to many architectures for various image analysis tasks. On the other hand, visible trajectories or movement patterns may be insufficient to effectively capture subtle nuances. Although there are research results such as transformer-based ViT in terms of performance, it is unclear whether it will be appropriate for detecting UAV trajectories that require a real-time agile response in that prior learning is required. We demonstrate how STEAM operates, as shown in Algorithm 1, and how suitable STEAM is for UAV trajectories compared to conventional methods by performance evaluation in the following sections.

**Algorithm 1** Spatial Trajectory Enhanced Attention Mechanism (STEAM)**Input:** Input feature map  $X$ **Output:** Attended feature map  $X'$ **Initialize:** Trajectory Convolutional Layer:  $\text{Conv2D}_{\text{trajectory}}$ ; Context Convolutional Layer:  $\text{Conv2D}_{\text{context}}$ ; Dynamic Weights Layer:  $\text{Dense}_{\text{dynamic}}$ ; Spatial Attention Convolutional Layer:  $\text{Conv2D}_{\text{spatial}}$ ;**for each training iteration do** $T \leftarrow \text{Conv2D}_{\text{trajectory}}(X)$ ; /\* Extract Trajectory Features \*/ $C \leftarrow \text{Conv2D}_{\text{context}}(X)$ ; /\* Extract Contextual Background Features \*/ $F \leftarrow \text{concatenate}(T, C)$ ; /\* Concatenate Trajectory and Context Features \*/ $D \leftarrow \text{Dense}_{\text{dynamic}}(F)$ ; /\* Compute Dynamically Weighted Features \*/ $A \leftarrow \text{Conv2D}_{\text{spatial}}(D)$ ; /\* Generate Spatial Attention Map \*/ $X' \leftarrow X \odot A$  /\* Apply Spatial Attention to the Input Feature Map \*/**end****return**  $X'$ 

## 4. Evaluation

### 4.1. Experiment Setup

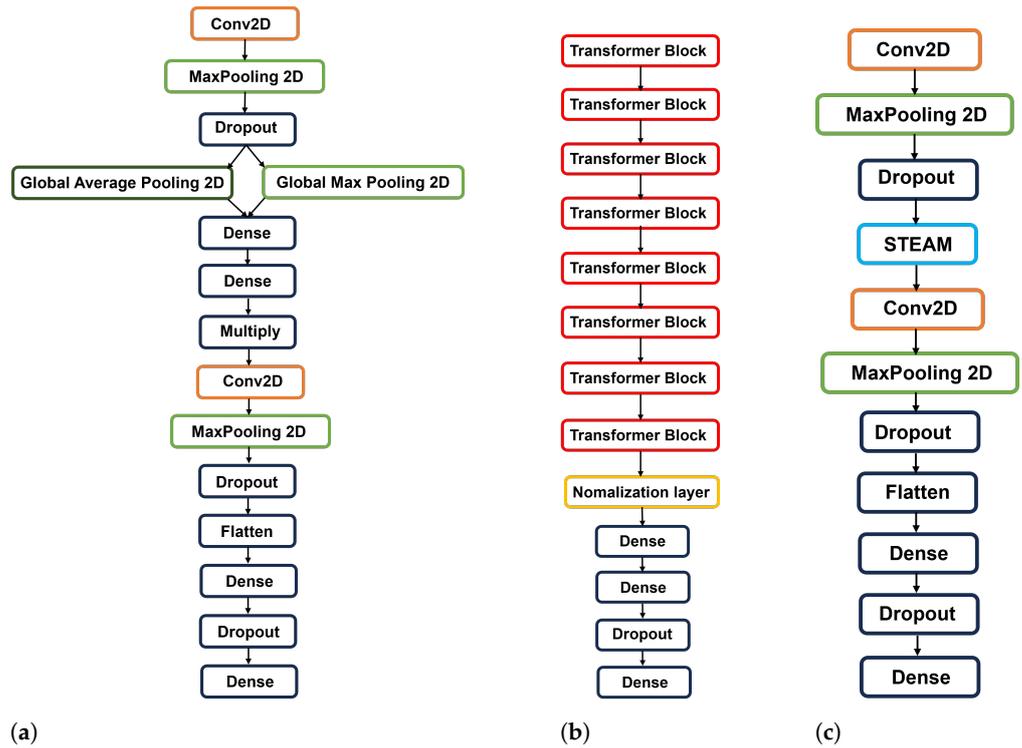
In this section, we evaluate our proposed method using several metrics to observe the benefits of each technique in detail and to measure the overall performance of the model. All evaluations are conducted using data from a simulation, which are based on measurements from Qorvo's DWM1000 UWB sensor rather than in real time. In this simulation, virtual UAVs were programmed to move along a trajectory, replicating the behavior of actual UAVs equipped with UWB sensors. This approach is chosen to accurately assess the performance of our proposals, eliminating performance degradation due to external factors and focusing specifically on flight trajectory prediction performance.

### 4.2. Performance Evaluation of STEAM

As described in Section 3, we demonstrate through experiments how suitable STEAM is for UAV trajectory data compared to conventional methods. To represent the performance of STEAM only, we implement a simple neural network model using STEAM as shown in Figure 6c. Then, we represent the utility of STEAM using CNN models that are the same as Figure 6c but without STEAM as a comparator. In addition, for the comparison group, the model with CBAM applied as shown in Figure 6a is implemented under similar conditions, and the ViT model is implemented as shown in Figure 6b. Since CBAM is a performance-like model with SE, comparison with SE is omitted [49].

We quantitatively evaluate the performance for each model shown as Table 1, and each metric is as follows [50]:

- **Loss:** This metric represents the error between the predicted and actual values of the data set. A smaller loss value means that the prediction value of the model is closer to the actual data.
- **Accuracy:** Accuracy measures the percentage of correct predictions made by the model among all predictions. High accuracy means that the model can correctly identify or classify data points.
- **Precision:** Precision evaluates the model's ability to accurately identify only the data points involved. High precision indicates that the model is likely to be accurate when predicting a class or outcome.
- **Recall:** Recall, also known as sensitivity, is the percentage at which the model predicted the correct answer among those that are actually correct. We propose that it can be an important indicator when it should not be mispredicted in practice such as UAV identification.



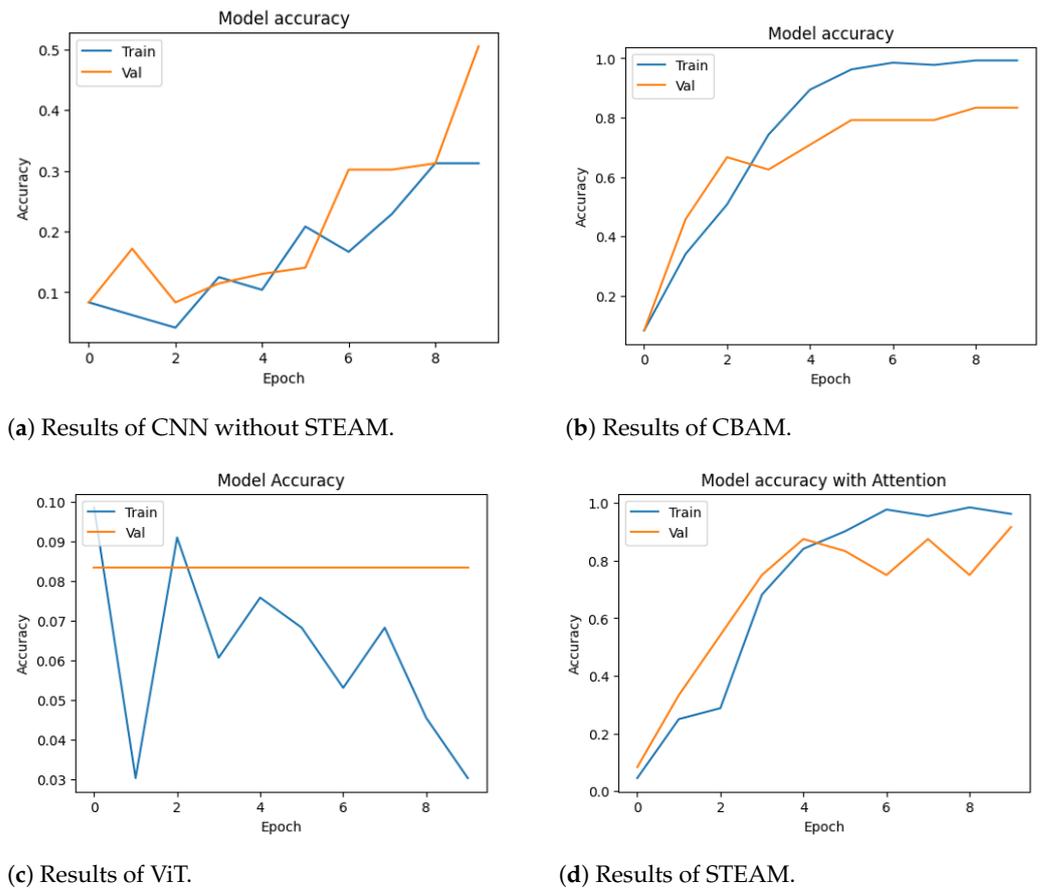
**Figure 6.** Performance evaluation of STEAM by comparing various attention mechanisms on UAV trajectory images. (a) Structure of CBAM model; (b) structure of ViT model; (c) structure of our STEAM model.

**Table 1.** Comparison of model performances on UAV trajectory data.

Model	Loss	Accuracy	Precision	Recall
Training Performance				
CNN	1.7753	0.3125	1.0000	0.0417
CBAM [44]	0.0568	0.9924	1.0000	1.0000
STEAM	0.1698	0.9792	1.0000	0.9375
ViT [46]	2.7292	0.0303	0.0000	0.0000
Validation Performance				
CNN	1.5938	0.5052	1.0000	0.0625
CBAM [44]	0.1941	0.8333	0.9886	0.9010
STEAM	0.1890	0.9688	0.9831	0.9115
ViT [46]	2.5221	0.0833	0.0000	0.0000

Figure 7 displays the overall results for both test and verification data. To evaluate the performance of our study on limited UAV identification data, we trained the model using only 50% of the entire dataset and devoted the remaining 50% to validation.

Experimental results highlight the effectiveness of STEAM in processing UAV trajectory data. STEAM achieved a remarkable accuracy of 97.92% and an impressive validation accuracy of 96.88% during training. Specifically, STEAM achieved a low loss of 0.1698, indicating its effectiveness in closely predicting the true values [51]. At the initial epoch, STEAM’s validation accuracy began at a modest rate, aligning closely with the training accuracy, which suggests consistency in model learning. As the epochs progressed, a steady increase in validation accuracy was observed, indicating the model’s improving ability to generalize. This demonstrates that STEAM has emphasized its ability to successfully capture most of the relevant functions in the data and identify them well in the invisible data.



**Figure 7.** Results of model learning and validation evaluation on UAV trajectories.

In comparison, CNN models without STEAM layers exhibited a fluctuating performance, with a training accuracy starting at around 31% and a validation accuracy at approximately 50%. This fluctuation was particularly noticeable between epochs 2 and 4, where a divergence between training and validation accuracy indicated potential overfitting to the training data. Another attention mechanism, CBAM, showed excellent training performance with 99.24% accuracy. However, the validation accuracy drops to 83.33% after Epoch 6, indicating that CBAM is effective for training but may not identify as effectively as STEAM in invisible data. Conversely, the ViT model performed much lower with an accuracy of 3.03% throughout the epochs. This may be due to the complexity of the ViT model and the possibility that the UAV trajectory data lacked sufficient context variability for effective learning in this model. In addition, the unique requirements of transformer-based models for pretraining on large datasets indicate that ViT is not suitable for real-time applications of UAV trajectory analysis.

In summary, these results shown as Table 1 indicate that STEAM is a very suitable and effective choice for UAV trajectory analysis, especially in scenarios that require real-time processing and robust generalization capabilities for new data. Furthermore, STEAM’s precision of 100% and recall of 93.75% during training, along with a precision of 98.31% and recall of 91.15% in validation, highlight its proficiency in correctly identifying relevant trajectory data while minimizing false positives and negatives. These metrics collectively underscore STEAM’s effectiveness in providing reliable and accurate analysis of UAV trajectories, making it an excellent tool for applications requiring precise and dependable trajectory analysis.

The excellence of STEAM in both training and verification phases highlights its potential as a reliable tool for UAV trajectory analysis in a variety of real-world applications. These model performance comparisons describe the underlying functions that contribute to the results, such as STEAM’s robust attention mechanisms and generalization capabilities,

which are pivotal to outstanding performance. Custom attention strategies for trajectory identification of STEAM, focusing on the most prominent trajectory functions, suggest the need for UAV pathways critical for real-time and reliable trajectory analysis in various operational contexts.

#### 4.3. Interpolation Performance Comparison

Our proposed methodology supports advanced technology that takes the domain of data collected from UWB from numerical values to images and fills in the holes using DNN. In Section 3.2, although the excellence of the Kalman filter in the numerical domain is recognized, we confirmed that data distortion occurs when the degree of missingness is severe. On the contrary, we also confirmed that our proposed methodology using DNN ensures stable interpolation performance in the image domain. This section lists specific quantitative evaluations of these facts discussed earlier and describes the analysis based on the results. Through this, we verify the excellence and feasibility of the proposed methodology.

First, we evaluate the similarity of images derived through the proposed methodology. The dataset used for evaluation is categorized into 12 labels and consists of a total of 240 images, 20 for each label. Each label represents a unique UWB trajectory, and approximately 50% missing rates are applied to all images for evaluation. Based on this dataset, we derive the results of applying our proposed methodology and Kalman filter to a single image and measure the similarity between them and the original image using various metrics. Table 2 shows the measurement results.

**Table 2.** Image similarity results for trajectory interpolation.

	1 Horizon	2 Left Down	3 Left Up	4 Mirror C	5 Mirror S	6 Original C	7 Original S	8 Right Down	9 Right Up	10 Square	11 Triangle	12 Vertical
RMSE (Pro- posed)	0.0022	0.0034	0.0029	0.0041	0.0042	0.0035	0.0047	0.0037	0.0028	0.0047	0.0035	0.0023
RMSE (Kalman filter)	0.0138	0.0152	0.0157	0.0163	0.0176	0.0167	0.0180	0.0155	0.0151	0.0177	0.0176	0.0121
PSNR (Pro- posed)	53.57	49.57	51.28	47.91	47.85	49.16	46.74	48.98	51.58	47.00	49.48	52.92
PSNR (Kalman filter)	37.46	36.40	36.09	35.79	35.08	35.54	34.92	36.22	36.47	35.05	35.12	38.63
SSIM (Pro- posed)	0.9963	0.9925	0.9947	0.9903	0.9893	0.9921	0.9872	0.9915	0.9950	0.9870	0.9918	0.9963
SSIM (Kalman filter)	0.9296	0.9191	0.9135	0.9077	0.8915	0.9025	0.8871	0.9159	0.9196	0.8893	0.8921	0.9419
FSIM (Pro- posed)	0.7592	0.7330	0.7357	0.7061	0.7124	0.7245	0.7002	0.7145	0.7573	0.7048	0.7260	0.7717
FSIM (Kalman filter)	0.3748	0.3230	0.2952	0.2907	0.2678	0.2749	0.2726	0.2971	0.3267	0.2999	0.2853	0.4483
SRE (Pro- posed)	62.63	60.52	61.37	59.61	59.48	60.22	58.90	60.22	61.52	58.97	60.27	62.30
SRE (Kalman filter)	52.49	51.86	51.69	51.47	51.02	51.33	50.92	51.76	51.89	50.92	51.02	53.07

In the table, a row represents the average value of a single label for various metrics and columns represent Root Mean Squared Error (RMSE), Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), Feature-based Similarity Index (FSIM), and

Signal to Reconstruction Error Ratio (SRE) between the predicted image and the original image for each interpolation technique. The description of each metric is listed as follows:

- RMSE is an indicator that measures the difference between the value predicted by the model and the actual value. This method squares the prediction error, calculates the average, and then takes its square root. The smaller the value, the closer the model's prediction is to the actual value. It is often used in regression analysis to evaluate the prediction accuracy of a model.
- PSNR is an indicator used to measure the quality of signals, such as video or voice. It is mainly used to evaluate the quality of image or video compression algorithms, and, the higher the PSNR value, the lower the distortion between the original and compressed signal. The higher the value, the higher the quality.
- SSIM measures the structural similarity between images and evaluates the similarity between the original image and the image being compared. It then quantifies image quality by considering brightness, contrast, and structural information. A high SSIM value indicates that the two images are structurally similar and of high quality, with values closer to 1 generally indicating higher quality.
- FSIM measures similarity by considering the features of the image and is based on features and structural information in the frequency domain. FSIM is used to measure the information loss in an image and evaluate its similarity to the original image, with higher values meaning higher quality and closer to the original image.
- SRE represents the ratio of a signal to the error that occurs in the process of reconstructing that signal. It is primarily used to evaluate how accurately a system reconstructs a signal through comparison between the original signal and the reconstructed signal. A higher SRE value means lower reconstruction error and higher accuracy.

According to the table results, our proposed methodology shows superior performance compared to Kalman filter for all metrics. The results from our proposed methodology have RMSE values close to 0 for all labels, which means that the images are numerically close to the original data. Moreover, they have an SSIM value close to 1, which means that they are structurally similar to the original data.

#### 4.4. Evaluation of System Performance

This section assesses the overall performance of our proposed system. It is hypothesized that defects may occur in the process of collecting UAV location data due to external environmental factors and network conditions. To address this, some images from each label were intentionally removed. These images were then recovered through interpolation and classified using the STEAM model. The dataset employed for this evaluation is divided into 12 labels, comprising a total of 240 images, with 20 images per label. For 30% of the images, 30% of their content was removed, 20% had 50% removed, 10% had 70% removed, and the remaining 40% were left intact.

Table 3 displays the classification results using the STEAM model, including precision, recall, and F1-score for each label. These metrics demonstrate the system's effectiveness in classifying UAV trajectory, even with partial missing data.

**Table 3.** Experimental results of classification utilizing proposed system.

Class	Precision	Recall	F1-Score
hor	1.00	1.00	1.00
ld	1.00	1.00	1.00
lu	1.00	1.00	1.00
mc	0.91	1.00	0.95
ms	0.95	1.00	0.98
oc	1.00	1.00	1.00
os	1.00	1.00	1.00
rd	1.00	0.90	0.95
ru	0.95	0.90	0.92
squ	1.00	0.90	0.95
tri	0.91	1.00	0.95
ver	1.00	1.00	1.00

The subsequent Table 4 details the accuracy of STEAM classifiers for each label relative to the extent of data omission.

**Table 4.** Accuracy of classification of each label separated as data missing rate.

Class	Data Missing Rate (0)	Data Missing Rate (30)	Data Missing Rate (50)	Data Missing Rate (70)
hor	100	100	100	100
ld	100	100	100	100
lu	100	100	100	100
mc	100	100	100	100
ms	100	100	100	100
oc	100	100	100	100
os	100	100	100	100
rd	87.5	100	75	100
ru	87.5	100	100	50
squ	100	100	75	50
tri	100	100	100	100
ver	100	100	100	100

While Table 4 indicates a decrease in accuracy for certain labels with increased missing data, the system demonstrated high accuracy for most labels, underscoring its resilience in effectively classifying UAV trajectory, even in scenarios with significant data gaps.

## 5. Conclusions

Our proposed framework, integrating a distance-based UWB positioning system and advanced image processing technologies, provides a comprehensive solution for accurate UAV trajectory tracking. The limitations faced by conventional GPS methods in complex environments and the difficulty of visual observation under various conditions require innovative approaches for UAV maneuver detection.

Therefore, we propose a new solution that utilizes UWB technology to accurately track and classify UAV trajectories in real time, even in complex environments. It also improves the reliability of abnormal maneuvering detection approaches by addressing data quality problems caused by external disturbances through image-based data interpolation. Moreover, the introduction of STEAM, a novel attention mechanism specifically designed for UAV trajectories, is a key innovation in our framework. It efficiently highlights important trajectory features with high precision and less complexity, outperforming existing models on classification tasks.

Our framework's experimental results underscore its effectiveness in UAV trajectory tracking, demonstrating robust performance independent of raw data quality. Moreover, the framework's validation using a lightweight learning model emphasizes its computational efficiency while maintaining exceptional classification accuracy.

We have several research plans for future work. First, we aim to explore the integration of supplementary data sources, such as infrared or LiDAR sensors, to enhance the comprehensive understanding of UAV trajectories, especially in challenging environments or adverse weather conditions. Second, evaluating the scalability and generalization capabilities of the proposed framework across diverse UAV platforms, mission types, and operational contexts will provide valuable insights into its broader applicability and performance under varied scenarios. Lastly, leveraging trajectory data from multiple UAVs, we intend to construct a framework capable of gaining extensive insights into complex flight trajectories formed by clusters of drones, offering a nuanced understanding of collaborative aerial maneuvers. By pursuing these future research directions, we aim to further advance the effectiveness and versatility of the proposed framework, addressing emerging challenges and extending its practical utility in diverse real-world scenarios.

**Author Contributions:** Conceptualization, K.K., D.L., T.Y. and N.Y.; methodology, K.K., D.L., N.Y. and H.K.; software, K.K., D.L. and N.Y.; validation, T.Y., H.J. and H.K.; formal analysis, K.K., D.L., N.Y. and H.K.; investigation, T.Y. and H.J.; resources, K.K., D.L., T.Y. and N.Y.; data curation, T.Y. and H.J.; writing—draft preparation, K.K., D.L., T.Y., N.Y., H.J. and H.K.; writing—review and editing, K.K., D.L., T.Y., N.Y., H.J. and H.K.; visualization, K.K., D.L. and H.J.; supervision, H.K.; project administration, H.K.; funding acquisition, H.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2021-0-01835) supervised by the IITP (Institute of Information & Communications Technology Planning & Evaluation). Moreover, this work was supported by the Human Resources Program in Energy Technology of the Korea Institute of Energy Technology Evaluation and Planning (KETEP) and the Ministry of Trade, Industry & Energy (MOTIE) of the Republic of Korea (No. 20204010600220).

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

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data is contained within the article.

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

## Abbreviations

The following abbreviations are used in this manuscript:

UAV	Unmanned Aerial Vehicle
GPS	Global Positioning System
UWB	Ultra-Wideband
DNN	Deep Neural Network
STEAM	Spatial Trajectory Enhanced Attention Mechanism
CBAM	Convolutional Block Attention Module
ViT	Vision Transformers
TWR	Two-Way Ranging
ToF	Time of Flight
RMSE	Root Mean Squared Error
PSNR	Peak Signal-to-Noise Ratio
SSIM	Structural Similarity Index Measure
FSIM	Feature-based Similarity Index Measure
SRE	Signal to Reconstruction Error Ratio

## References

1. Aggarwal, S.; Kumar, N. Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges. *Comput. Commun.* **2020**, *149*, 270–299. [\[CrossRef\]](#)
2. Cabreira, T.M.; Brisolará, L.B.; Paulo R.F.J. Survey on coverage path planning with unmanned aerial vehicles. *Drones* **2019**, *3*, 4. [\[CrossRef\]](#)
3. Shivgan, R.; Dong, Z. Energy-efficient drone coverage path planning using genetic algorithm. In Proceedings of the 2020 IEEE 21st International Conference on High Performance Switching and Routing (HPSR), Newark, NJ, USA, 11–14 May 2020; pp. 1–6.
4. Hayat, S.; Yanmaz, E.; Bettstetter, C.; Brown, T.X. Multi-objective drone path planning for search and rescue with quality-of-service requirements. *Auton. Robot.* **2020**, *44*, 1183–1198. [\[CrossRef\]](#)
5. Yoo, T.; Lee, S.; Yoo, K.; Kim, H. Reinforcement Learning Based Topology Control for UAV Networks. *Sensors* **2023**, *23*, 921. [\[CrossRef\]](#) [\[PubMed\]](#)
6. Park, C.; Lee, S.; Joo, H.; Kim, H. Empowering Adaptive Geolocation-Based Routing for UAV Networks with Reinforcement Learning. *Drones* **2023**, *7*, 387. [\[CrossRef\]](#)
7. Um, I.; Park, S.; Kim, H.T.; Kim, H. Configuring RTK-GPS architecture for system redundancy in multi-drone operations. *IEEE Access* **2020**, *8*, 76228–76242. [\[CrossRef\]](#)
8. Li, F.; Zlatanova, S.; Koopman, M.; Bai, X.; Diakité, A. Universal path planning for an indoor drone. *Autom. Constr.* **2018**, *95*, 275–283. [\[CrossRef\]](#)
9. Kim, H.T.; Kim, H. Precise localization of a UAV with single vision camera and deep learning. In Proceedings of the GLOBECOM 2020 IEEE Global Communications Conference, Taipei, Taiwan, 7–11 December 2020; pp. 1–6.
10. Ibrahim, N.A.; Zakaria, M.Y.; Kamal, A. Development of a Flight Test Data Analyzer for Pixhawk Autopilots. In Proceedings of the AIAA SCITECH 2023 Forum, National Harbor, MD, USA, 23–27 January 2023; p. 0482.
11. Lee, S.; Yoo, S.; Lee, J.Y.; Park, S.; Kim, H. Drone positioning system using UWB sensing and out-of-band control. *IEEE Sens. J.* **2021**, *22*, 5329–5343. [\[CrossRef\]](#)
12. Acharya, T.; Ray, A.K. *Image Processing: Principles and Applications*; John Wiley & Sons: Hoboken, NJ, USA, 2005.
13. Park, S.; Kim, H. Dagmap: Multi-drone slam via a dag-based distributed ledger. *Drones* **2022**, *6*, 34. [\[CrossRef\]](#)
14. Chmaj, G.; Selvaraj, H. Distributed processing applications for UAV/drones: A survey. In Proceedings of the Progress in Systems Engineering: Proceedings of the Twenty-Third International Conference on Systems Engineering, Indianapolis, IN, USA, 7–10 October 2015; pp. 449–454.
15. Tsouros, D.C.; Bibi, S.; Sarigiannidis, P.G. A review on UAV-based applications for precision agriculture. *Information* **2019**, *10*, 349. [\[CrossRef\]](#)
16. Zhao, K.; He, T.; Wu, S.; Wang, S.; Dai, B.; Yang, Q.; Lei, Y. Application research of image recognition technology based on CNN in image location of environmental monitoring UAV. *Eurasip J. Image Video Process.* **2018**, *2018*, 1–11. [\[CrossRef\]](#)
17. Daud, S.M.S.M.; Yusof, M.Y.P.M.; Heo, C.C.; Khoo, L.S.; Singh, M.K.C.; Mahmood, M.S.; Nawawi, H. Applications of drone in disaster management: A scoping review. *Sci. Justice* **2022**, *62*, 30–42. [\[CrossRef\]](#) [\[PubMed\]](#)
18. Liu, Y.; Liao, L.; Wu, H.; Qin, J.; He, L.; Yang, G.; Zhang, H.; Zhang, J. Trajectory and image-based detection and identification of UAV. *Vis. Comput.* **2021**, *37*, 1769–1780. [\[CrossRef\]](#)
19. Lin, C.E.; Shao, P.C. Failure analysis for an unmanned aerial vehicle using safe path planning. *J. Aerosp. Inf. Syst.* **2020**, *17*, 358–369. [\[CrossRef\]](#)
20. Koutsonikolas, D.; Das, S.M.; Hu, Y.C. Path planning of mobile landmarks for localization in wireless sensor networks. *Comput. Commun.* **2007**, *30*, 2577–2592. [\[CrossRef\]](#)
21. Huang, R.; Zaruba, G.V. Static path planning for mobile beacons to localize sensor networks. In Proceedings of the Fifth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PerComW'07), White Plains, NY, USA, 19–23 March 2007; pp. 323–330.
22. Rezaadeh, J.; Moradi, M.; Ismail, A.S.; Dutkiewicz, E. Superior path planning mechanism for mobile beacon-assisted localization in wireless sensor networks. *IEEE Sens. J.* **2014**, *14*, 3052–3064. [\[CrossRef\]](#)
23. Perazzo, P.; Sorbelli, F.B.; Conti, M.; Dini, G.; Pinotti, C.M. Drone path planning for secure positioning and secure position verification. *IEEE Trans. Mob. Comput.* **2016**, *16*, 2478–2493. [\[CrossRef\]](#)
24. Çelik, Ü.; Eren, H. Classification of manifold learning based flight fingerprints of UAVs in air traffic. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 5229–5238. [\[CrossRef\]](#)
25. Wang, B.; Liu, D.; Wang, W.; Peng, X. A hybrid approach for UAV flight data estimation and prediction based on flight mode recognition. *Microelectron. Reliab.* **2018**, *84*, 253–262. [\[CrossRef\]](#)
26. Aksoy, M.; Ozdemir, O.; Guner, G.; Baspinar, B.; Koyuncu, E. Flight trajectory pattern generalization and abnormal flight detection with generative adversarial network. In Proceedings of the AIAA Scitech 2021 Forum, National Harbor, MD, USA, 23–27 January 2021; p. 0775.
27. Dong, W.; Ding, Y.; Huang, J.; Yang, L.; Zhu, X. An optimal curvature smoothing method and the associated real-time interpolation for the trajectory generation of flying robots. *Robot. Auton. Syst.* **2019**, *115*, 73–82. [\[CrossRef\]](#)
28. Ramos, G.S.; Pinto, M.F.; Coelho, F.O.; Honório, L.M.; Haddad, D.B. Hybrid methodology based on computational vision and sensor fusion for assisting autonomous UAV on offshore messenger cable transfer operation. *Robotica* **2022**, *40*, 2786–2814. [\[CrossRef\]](#)

29. Zeng, W.; Xu, Z.; Cai, Z.; Chu, X.; Lu, X. Aircraft trajectory clustering in terminal airspace based on deep autoencoder and gaussian mixture model. *Aerospace* **2021**, *8*, 266. [[CrossRef](#)]
30. Vladimir, G.; Yury, M.; Natalya, K. Pattern recognition techniques for classifying aeroballistic flying vehicle paths. In *Neural Computing and Applications*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 1–13.
31. Mcfadyen, A.; O’Flynn, M.; Martin, T.; Campbell, D. Aircraft trajectory clustering techniques using circular statistics. In Proceedings of the 2016 IEEE Aerospace Conference, Big Sky, MT, USA, 5–12 March 2016; pp. 1–10.
32. Gingrass, C.J. Classifying ADS-B Trajectory Shapes Using a Dense Feed-Forward Neural Network. Ph.D. Thesis, Naval Postgraduate School, Monterey, CA, USA, 2020.
33. Dong, Q.; Wang, N.; Song, J.; Hao, L.; Liu, S.; Han, B.; Qu, K. Math-data integrated prediction model for ship maneuvering motion. *Ocean. Eng.* **2023**, *285*, 115255. [[CrossRef](#)]
34. Wang, N.; Kong, X.; Ren, B.; Hao, L.; Han, B. SeaBil: Self-attention-weighted ultrashort-term deep learning prediction of ship maneuvering motion. *Ocean. Eng.* **2023**, *287*, 115890. [[CrossRef](#)]
35. Jun, M.; D’Andrea, R. Path planning for unmanned aerial vehicles in uncertain and adversarial environments. In *Cooperative Control: Models, Applications and Algorithms*; Springer: Berlin/Heidelberg, Germany, 2003; pp. 95–110.
36. Jones, M.; Djahel, S.; Welsh, K. Path-planning for unmanned aerial vehicles with environment complexity considerations: A survey. *ACM Comput. Surv.* **2023**, *55*, 1–39. [[CrossRef](#)]
37. Liu, G.; Reda, F.A.; Shih, K.J.; Wang, T.C.; Tao, A.; Catanzaro, B. Image inpainting for irregular holes using partial convolutions. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 85–100.
38. Son, H.; Jang, Y. Partial convolutional LSTM for spatiotemporal prediction of incomplete data. *IEEE Access* **2020**, *8*, 164762–164774. [[CrossRef](#)]
39. Liu, G.; Dundar, A.; Shih, K.J.; Wang, T.C.; Reda, F.A.; Sapra, K.; Yu, Z.; Yang, X.; Tao, A.; Catanzaro, B. Partial convolution for padding, inpainting, and image synthesis. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 6096–6110. [[CrossRef](#)]
40. Xing, D.; Hou, J.; Huang, C.; Zhang, W. Spatiotemporal Reconstruction of MODIS Normalized Difference Snow Index Products Using U-Net with Partial Convolutions. *Remote Sens.* **2022**, *14*, 1795. [[CrossRef](#)]
41. Niu, Z.; Zhong, G.; Yu, H. A review on the attention mechanism of deep learning. *Neurocomputing* **2021**, *452*, 48–62. [[CrossRef](#)]
42. Guo, M.H.; Xu, T.X.; Liu, J.J.; Liu, Z.N.; Jiang, P.T.; Mu, T.J.; Zhang, S.H.; Martin, R.R.; Cheng, M.M.; Hu, S.M. Attention mechanisms in computer vision: A survey. *Comput. Vis. Media* **2022**, *8*, 331–368. [[CrossRef](#)]
43. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
44. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 10 September 2018; pp. 3–19.
45. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7132–7141.
46. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16 × 16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
47. Hafiz, A.M.; Parah, S.A.; Bhat, R.U.A. Attention mechanisms and deep learning for machine vision: A survey of the state of the art. *arXiv* **2021**, arXiv:2106.07550.
48. Liu, G.; Shih, K.J.; Wang, T.C.; Reda, F.A.; Sapra, K.; Yu, Z.; Tao, A.; Catanzaro, B. Partial convolution based padding. *arXiv* **2018**, arXiv:1811.11718.
49. Zhang, Z.; Wu, Y.; Zhang, J.; Kwok, J. Efficient channel attention for deep convolutional neural networks. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023.
50. Al-Sa’d, M.F.; Al-Ali, A.; Mohamed, A.; Khatib, T.; Erbad, A. RF-based drone detection and identification using deep learning approaches: An initiative towards a large open source drone database. *Future Gener. Comput. Syst.* **2019**, *100*, 86–97. [[CrossRef](#)]
51. Garg, I.; Roy, K. Samples With Low Loss Curvature Improve Data Efficiency. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–23 June 2023; pp. 20290–20300.

**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.