

# ShuttleNet: Position-aware Fusion of Rally Progress and Player Styles for Stroke Forecasting in Badminton

Wei-Yao Wang, Hong-Han Shuai, Kai-Shiang Chang, Wen-Chih Peng

National Yang Ming Chiao Tung University, Hsinchu, Taiwan

{sf1638.cs05, hhshuai, kevin5260523.cs05}@nctu.edu.tw, wcpeng@cs.nctu.edu.tw

## Abstract

The increasing demand for analyzing the insights in sports has stimulated a line of productive studies from a variety of perspectives, *e.g.*, health state monitoring, outcome prediction. In this paper, we focus on objectively judging what and where to return strokes, which is still unexplored in turn-based sports. By formulating stroke forecasting as a sequence prediction task, existing works can tackle the problem but fail to model information based on the characteristics of badminton. To address these limitations, we propose a novel Position-aware Fusion of Rally Progress and Player Styles framework (ShuttleNet) that incorporates rally progress and information of the players by two modified encoder-decoder extractors. Moreover, we design a fusion network to integrate rally contexts and contexts of the players by conditioning on information dependency and different positions. Extensive experiments on the badminton dataset demonstrate that ShuttleNet significantly outperforms the state-of-the-art methods and also empirically validates the feasibility of each component in ShuttleNet. On top of that, we provide an analysis scenario for the stroke forecasting problem.

## Introduction

In recent years, sports analytics has drawn significant attention due to the enormous market, which focuses on collecting sports data and implementing advanced techniques for mining useful information from the data. In Major League Baseball, for example, teams started to shift to defense by moving infielders to specific positions according to the hitting pattern of opposing batters, and these types of shifts dramatically rose from 4.62% in 2012 to 21.17% in 2019 (Bechtold 2019). Furthermore, there are about 100 sports-related organizations currently investigating new technologies for delivering interesting stream contents to fans in 2021 (Morrison 2021). Generally, the target audience of sports analytics is composed of both coaching-oriented groups and community-oriented groups. Coaching-oriented groups aim at improving player performance, *e.g.*, tactic investigation (Decroos, Haaren, and Davis 2018; Beal et al. 2020) and action valuing (Jayanth et al. 2018; Sicilia, Pelechrinis, and Goldsberry 2019), while community-oriented groups try to boost the spectator engagement, *e.g.*, highlight prediction

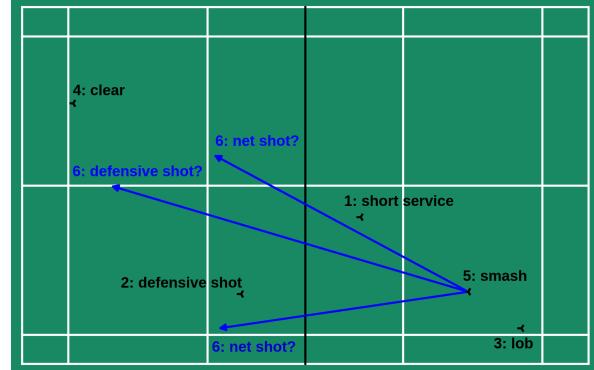


Figure 1: An example of stroke forecasting in a singles rally. The black line in the middle is the net. Each stroke consists of the order in the rally and its shot type. Blue lines with shot types represent possible choices in the next stroke.

(Decroos et al. 2017), play retrieval (Wang et al. 2019) and autonomous broadcast production (Giancola and Ghanem 2021). There are also sports analytics applications that are both for coaching-oriented groups and community-oriented groups (Decroos et al. 2019; Merhej et al. 2021), *e.g.*, player performance analysis and providing the relation between performance and market value of the player.

In this paper, we focus on turn-based sports and use badminton as the demonstration example. The related works on badminton mainly focus on quantifying stroke performance (Sharma et al. 2021; Wang et al. 2021) or detecting stroke-related information from videos (Chu and Situmeang 2017; Hsu et al. 2019; Wang et al. 2020; Yoshikawa et al. 2021). However, there is another application that has not yet been addressed by previous works: to forecast the future strokes including shot types and locations given the past stroke sequences. Predicting future strokes based on the past strokes is essential and beneficial for coaching the player and determining the strategies since it simulates the tactics of the players, which can be used to investigate what shot types are often returned and where the strokes are returned to by the player for decision-making. In addition, stroke forecasting can also benefit the community for storytelling by assessing returning probability distributions during the matches.

Figure 1 illustrates an example of stroke forecasting. Sup-

pose the past five strokes with corresponding shot types and destination locations are known, and the fifth stroke is a smash from the left side, the player on the right side has several choices to return such as defensively returning to the middle of the left side, or returning close to the net to mobilize the opponent from the back court to front court. To the best of our knowledge, there is no existing method that can predict the next strokes.

To tackle this challenging problem, stroke forecasting can be formulated as the sequence prediction task. One possible solution is to use statistical methods like n-gram models in natural language processing to calculate the probabilities of occurrence for predicting next future strokes. Nevertheless, the probabilities of occurrence in n-gram models become sparse when increasing window size. To solve the issue, the sequence-to-sequence model (Sutskever, Vinyals, and Le 2014) can be applied to encode the input sequence and then decode the output sequence with encoding vectors. However, there are three challenges for applying the sequence-to-sequence model directly for stroke forecasting. 1) *Mixed sequence*. One of the characteristics of badminton is that there are two players returning strokes alternatively to form a rally. Therefore, stroke forecasting is a turn-based sequence prediction task rather than a conventional sequence prediction task with the same target in the sequence. 2) *Multiple outputs*. Differing from general sequence tasks that only predict one output, the stroke forecasting task has multiple outputs (shot types and area coordinates) at each timestamp. 3) *Player dependence*. Returning strokes are based on the overall styles of the players and the current situation in the rally. Furthermore, the importance of the overall styles of the players and the current situation in the rally also varies when encountering different players and different positions. It is challenging to disentangle the player features directly from the rally sequences.

To address the aforementioned challenges, we propose a novel Position-aware Fusion of Rally Progress and Player Styles framework (ShuttleNet), which consists of two encoder-decoder extractors for modeling rally progress and retrieving player styles from turn-based sequence and a fusion network to take into account the dependencies between rally progress and player styles at each stroke. To predict multiple outputs at each step, two task-specific predictors are adopted in the end for predicting shot types and area coordinates. Specifically, the first encoder-decoder extractor, named Transformer-Based Rally Extractor (TRE), is designed to capture the progress of the rally. Moreover, the second encoder-decoder extractor, named Transformer-Based Player Extractor (TPE), separates the information of each player to generate the context of each player. Finally, a Position-aware Gated Fusion Network (PGFN) is adopted to fuse rally contexts and contexts of two players by incorporating information weights and position weights. In this manner, we can learn different contributions at each stroke to predict future shot types and area coordinates. In summary, our contributions are as follows:

- A novel framework named Position-aware Fusion of Rally Progress and Player Styles (ShuttleNet) is proposed to predict future strokes by giving past observed

strokes. To the best of our knowledge, this is the first work for stroke forecasting in sports, which can be applied to turn-based sports analytics.

- The proposed framework first generates rally contexts and contexts of players by leveraging two encoder-decoder extractors and then fuses these contexts based on information weights and position weights. Furthermore, we introduce an attention mechanism to better integrate the information of shot types and locations.
- Extensive experiments and ablation studies on a real-world badminton dataset are conducted to demonstrate the effectiveness of the proposed ShuttleNet framework.

## Related Works

### Sport Data Analytics

The area of artificial intelligence for sports contains five technical issues (Decroos et al. 2020): representation (Decroos, Haaren, and Davis 2018), interpretability (Silver and Huffman 2021), decision making (Sicilia, Pelechrinis, and Goldsberry 2019), understanding behavior (Weeratunga, Dharmaratne, and How 2017), and experimental evaluation (Liu and Schulte 2018). For instance, Decroos, Haaren, and Davis (2018) proposed SPADL to address the data science challenges as unique definitions of different vendors by unifying soccer event-based data, which reduces the burden on redesigning data formats when serving different objectives. Mimic learning is another approach used in sports that builds a model with both accurate predictions and interpretable insights (Sun et al. 2020). Action valuing designs objective metrics by valuing scoring and defensive performance of each action, which can be used as evaluation tools for understanding the behavior of players (Decroos et al. 2019; Merhej et al. 2021). Outcome prediction utilizes machine learning approaches and has been applied to cricket, soccer, and badminton, to help coaches to select the optimal players to win the game (Jayanth et al. 2018; Robberechts, Haaren, and Davis 2021; Sharma et al. 2021). Pappalardo et al. (2019) proposed PlayeRank by combining multi-dimensional and role-aware evaluations from a massive soccer database, in order to provide evaluations and rankings of soccer players. Our focus, in contrast, addresses stroke forecasting in badminton, which is also critically related to the above issues. Predicting future strokes can facilitate decision-making and provide behavioral understanding of returning strokes within probability distributions. Moreover, ground truth labels can also be objectively obtained when labeling instead of customizing evaluation methods.

### Sequence Prediction

The sequence-to-sequence model (Sutskever, Vinyals, and Le 2014) was proposed to deal with machine translation by encoding input with a Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber 1997) and then using another LSTM as a decoder to generate predictions. The encoder-decoder architecture has been widely adopted for various sequence prediction tasks such as pedestrian trajectory prediction (Mohajerin and Rohani 2019) and citation forecasting (Liu et al. 2017). Xu, Yang, and Du (2020) de-

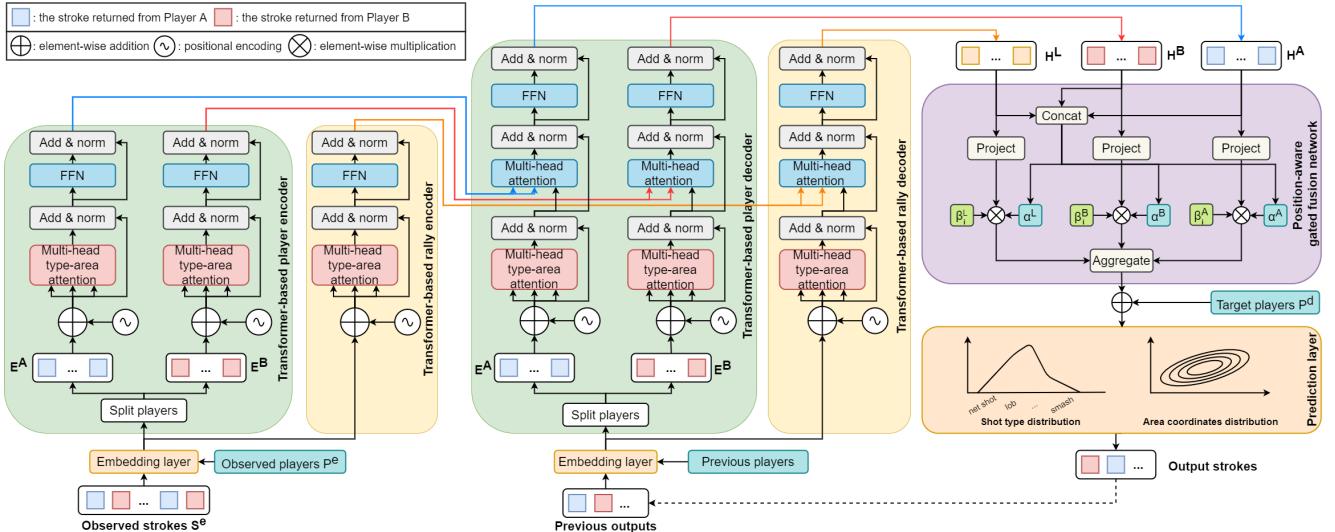


Figure 2: Illustration of the ShuttleNet framework. The Transformer-based rally extractor (TRE) in the yellow part generates rally contexts. The Transformer-based player extractor (TPE) in green shares the same set of parameters and generates contexts of both players. The contexts of the rally and both players are fed into the position-aware gated fusion network (PGFN) to weigh the contributions based on information and position for predicting future strokes.

signed CF-LSTM with a feature-cascaded module to robustly capture dynamic information of trajectories from previous steps without other pedestrians' information. More recently, Transformer has become popular for sequence prediction. Giulari et al. (2020), for example, adopted Transformer Networks to predict future trajectory. DMA-Nets introduced two temporal attention mechanisms to model local temporal information and global temporal information for citation forecasting (Ji et al. 2021), which is the most similar setting (multiple outputs) to stroke forecasting. However, these previous works focus on the same target in a sequence, and thus cannot be directly applied to turn-based sequences.

## Problem Formulation

Let  $R = \{S_r, P_r\}_{r=1}^{|R|}$  denote historical rallies of badminton matches, where the  $r$ -th rally is composed of a stroke sequence with type-area pairs  $S_r = (\langle s_1, a_1 \rangle, \dots, \langle s_{|S_r|}, a_{|S_r|} \rangle)$  and a player sequence  $P_r = (p_1, \dots, p_{|S_r|})$ . At the  $i$ -th stroke,  $s_i$  represents the shot type,  $a_i = \langle x_i, y_i \rangle \in \mathbb{R}^2$  are the coordinates of the shuttle destinations, and  $p_i$  is the player who hits the shuttle. We denote Player A as the served player and Player B as the other for each rally in this paper. For instance, given a singles rally between Player A and Player B,  $P_r$  may become  $(A, B, \dots, A, B)$ . We formulate the problem of stroke forecasting as follows. For each rally, given the observed  $\tau$  strokes  $(\langle s_i, a_i \rangle)_{i=1}^\tau$  with players  $(p_i)_{i=1}^\tau$ , the goal is to predict the future strokes including shot types and area coordinates for the next  $n$  steps, i.e.,  $(\langle s_i, a_i \rangle)_{i=\tau+1}^{\tau+n}$ .

## Methodology

Figure 2 illustrates the overview of the proposed framework. The input of the encoder side is the sequence of observed  $\tau$

strokes  $S^e = (\langle s_i, a_i \rangle)_{i=1}^\tau$  and players  $P^e = (p_i)_{i=1}^\tau$ , and the decoder auto-regressively predicts the sequence of the future  $n$  strokes  $S^d = (\langle s_i, a_i \rangle)_{i=\tau+1}^{\tau+n}$  by taking encoding contexts and target players  $P^d = (p_i)_{i=\tau+1}^{\tau+n}$ . Each stroke encompassing a shot type and area coordinates is embedded with player information from the embedding layer as a personal stroke. Each encoder-decoder extractor is based on the Transformer (Vaswani et al. 2017). We replace the first multi-head self-attention layer in both the encoder and decoder with the proposed type-area-attention layer to better integrate the information of shot types and area. Moreover, contexts of the rally are generated by the Transformer-based rally extractor, and contexts of the two players are obtained by the Transformer-based player extractor. Outputs of these contexts are fused by the position-aware gated fusion network using information weights and position weights for predicting future shot types and area coordinates.

## Embedding Layer

Each stroke contains a shot type and area coordinates with the player who hits the stroke. The output of embedding layer at  $i$ -th stroke  $e_i$  is calculated as follows:

$$e_i = \langle e_i^s, e_i^a \rangle = \langle s'_i + p'_i, a'_i + p'_i \rangle, \quad (1)$$

where  $s'_i$  is a type embedding projected from  $s_i$  using  $M^s \in \mathbb{R}^{N_s \times d}$ , where  $N_s$  is the number of shot types,  $p'_i$  is a player embedding projected from  $p_i$  using  $M^p \in \mathbb{R}^{N_p \times d}$ , where  $N_p$  is the number of players, and  $a'_i$  is an area embedding projected from  $a_i$  using  $M^a \in \mathbb{R}^{2 \times d}$  with the ReLU activation function. In order to make use of the player in shot types and area, player embeddings are added to both type embeddings and area embeddings. The parameters of embedding layers in the encoder side and the decoder side are shared similar to (Press and Wolf 2017) for size reduction.

## Transformer-based Rally Extractor (TRE)

TRE reflects the current situation in the rally, which is a critical condition for returning strokes. For example, if the player defensively returns a stroke like a lob to the back court, this indicates the player may become passive and the other player can seize the chance to return an aggressive stroke.

To capture the progress in the rally, we first add positional encodings (Vaswani et al. 2017) to the embeddings by

$$\begin{aligned} E^L &= (\langle \tilde{e}_1^s, \tilde{e}_1^a \rangle, \langle \tilde{e}_2^s, \tilde{e}_2^a \rangle, \dots) \\ &= (\langle e_1^s + pe_1, e_1^a + pe_1 \rangle, \langle e_2^s + pe_2, e_2^a + pe_2 \rangle, \dots), \end{aligned} \quad (2)$$

where  $pe_i$  is the position encoding for  $i$ -th stroke.

Afterward, we adopt a modified Transformer framework by replacing the first multi-head self-attention layer in the encoder and decoder with the proposed multi-head type-area-attention layer. Specifically, we take  $E^L$  as the inputs of the Transformer-based rally extractor and generate the contexts of the rally  $H^L = (h_{\tau+1}^L, h_{\tau+2}^L, \dots)$ , where the  $i$ -th stroke  $h_i^L \in \mathbb{R}^d$  is a  $d$  dimension vector.

**Type-area-attention layer** Since there are two components (shot type and area) of each stroke, the self-attention mechanism can only be applied in an early-fusion manner (*e.g.*, concatenation), which forces to attend shot types and area at the same position. However, it is expected that playing strategies should be considered from different aspects in badminton matches. For example, returning a current shot type may be considered the last shot type to decide the proper choice, while where to return may be considered the previous location returned from the same player because the opponent is weak on a specific side.

Therefore, inspired by disentangled attention (He et al. 2021), we propose an attention mechanism to separately characterize the importance of shot types and area and then aggregate corresponding scores as final attention scores. Here, we illustrate the computation of attention contexts on the encoder side, and the decoder follows a similar process. Given the input sequence with positional type embeddings  $E^s = (\tilde{e}_1^s, \dots, \tilde{e}_\tau^s)$  and positional area embeddings  $E^a = (\tilde{e}_1^a, \dots, \tilde{e}_\tau^a)$ , the formula of the multi-head type-area attention is derived as follows:

$$Q_s = E^s W^{Q_s}, K_s = E^s W^{K_s}, V_s = E^s W^{V_s}, \quad (3)$$

$$Q_a = E^a W^{Q_a}, K_a = E^a W^{K_a}, V_a = E^a W^{V_a}, \quad (4)$$

$$A = Q_a K_a^T + Q_s K_s^T + Q_s K_a^T + Q_a K_s^T, \quad (5)$$

$$TAA(E_s, E_a) = \text{softmax}\left(\frac{A}{\sqrt{4d}}\right)(V_a + V_s), \quad (6)$$

$$\text{MultiHead}(E_s, E_a) = \text{Concat}(TAA_1, \dots, TAA_h)W^o, \quad (7)$$

where  $TAA$  denotes the function of type-area attention with single head,  $Q_s$ ,  $K_s$ , and  $V_s$  are queries, keys and values of  $E^s$  projected using projection matrices  $W^{Q_s}$ ,  $W^{K_s}$ ,  $W^{V_s} \in \mathbb{R}^{d \times d}$ , respectively.  $Q_a$ ,  $K_a$ , and  $V_a$  are queries, keys and values of  $E^a$  projected using matrices  $W^{Q_a}$ ,  $W^{K_a}$ ,  $W^{V_a} \in \mathbb{R}^{d \times d}$ , respectively.  $h$  is the number of heads, and  $W^o \in \mathbb{R}^{hd \times d}$  is a learnable matrix.

## Transformer-based Player Extractor (TPE)

In addition to the information of the rally, returning strokes also needs to consider the overall style of each player. That is, the player should minimize their opponents' advantages and maximize their own. To this end, we designed an extractor to split the sequence into two subsequences based on the player and then to produce the contexts of each player.

First, the outputs of the embedding layer are alternatively split based on the players as follows:

$$E^A = (e_1, e_3, \dots), E^B = (e_2, e_4, \dots), \quad (8)$$

where  $E^A$  is the sequence of Player A,  $E^B$  is the sequence of Player B. TPE adopts two encoder-decoder architectures to capture the two sequences split by players, both of which are the same as the architecture in TRE. Specifically,  $E^A$  and  $E^B$  are fed into TPE to generate corresponding contexts.

It is worth noting that the positional encodings are added separately to the two subsequences to specify the order of each player rather than using the entire sequence in Equation 2. Further, the parameters of the two architectures are shared not only to reduce the number of parameters but also to prevent player information from falling on the same side, which would cause imbalance.

Since the lengths of two subsequences are shorter than the original sequence, sequence alignment is applied to align two subsequences with the same length of rally sequence after generating two contexts of the players. The alignment principle is to add a copy stroke to the next stroke, which becomes the opponent to return. The formula of the sequence alignment is derived as follows:

$$H^A = (h_{\tau+1}^A, h_{\tau+2}^A, h_{\tau+2}^A, \dots), \quad (9)$$

$$H^B = (0, h_{\tau+1}^B, h_{\tau+1}^B, h_{\tau+2}^B, h_{\tau+2}^B, \dots), \quad (10)$$

where the  $i$ -th stroke  $h_i^A \in \mathbb{R}^d$  denotes the output from the decoder for Player A, and the  $i$ -th stroke  $h_i^B \in \mathbb{R}^d$  is the output from the other decoder for Player B. Zero is padded at the first stroke of  $H^B$  since the first stroke is always served by Player A.

## Position-aware Gated Fusion Network (PGFN)

When returning strokes, players consider various important information about both players and current rally. Moreover, the importance of these types of information at each stroke will vary. To take the above consideration into the design, we propose a position-aware gated fusion network based on gated multi-modal units (Ovalle et al. 2020) to fuse rally contexts and contexts of two players.

Given the contexts of Player A and Player B ( $h_i^A$  and  $h_i^B$ ), and the rally  $h_i^L$  at  $i$ -th stroke, the PGFN first projects to hidden vectors of fusing contexts:

$$\tilde{h}_i^A = \delta_t(h_i^A W^A), \tilde{h}_i^B = \delta_t(h_i^B W^B), \tilde{h}_i^L = \delta_t(h_i^L W^L), \quad (11)$$

where  $\delta_t(\cdot)$  is the tanh activation function, and  $W^A, W^B, W^L \in \mathbb{R}^{d \times d}$  are learnable matrices. The information weights to represent the importance of the three contexts are calculated as follows:

$$\alpha^A = \delta_s([\tilde{h}_i^A, \tilde{h}_i^B, \tilde{h}_i^L] \tilde{W}^A), \quad (12)$$

$$\alpha^B = \delta_s([\tilde{h}_i^A, \tilde{h}_i^B, \tilde{h}_i^L] \tilde{W}^B), \quad (13)$$

$$\alpha^L = \delta_s([\tilde{h}_i^A, \tilde{h}_i^B, \tilde{h}_i^L] \tilde{W}^L), \quad (14)$$

where  $\delta_s(\cdot)$  is the sigmoid activation function,  $[\cdot, \cdot, \cdot]$  denotes the concatenation operator, and  $\tilde{W}^A, \tilde{W}^B, \tilde{W}^L \in \mathbb{R}^{3d \times d}$  are learnable matrices.

Finally, the  $i$ -th fusing output is calculated as:

$$z_i = \delta_s(\beta_i^A \otimes \alpha^A \otimes \tilde{h}_i^A + \beta_i^B \otimes \alpha^B \otimes \tilde{h}_i^B + \beta_i^L \otimes \alpha^L \otimes \tilde{h}_i^L), \quad (15)$$

where  $\otimes$  denotes the element-wise multiplication, and  $\beta_i^A, \beta_i^B, \beta_i^L \in \mathbb{R}^d$  are learnable position weights to learn how much to pass at each stroke.

## Prediction Layer

To predict the shot type and area coordinates at  $i$ -th stroke, we first assume area coordinates follow a bi-variate Gaussian distribution since there exists uncertainty and potential multi-modality when returning strokes. For instance, when the opponent returns the stroke to the back court, the player can return to the non-handedness side to force the opponent to return the shuttle with back hand, or to return to near the net to make the opponent return defensively. Moreover, the predictive distribution enables the ability to investigate the locations of frequent and less frequent stroke returns to better understand the players' behaviors. Specifically, area coordinates are sampled from a bi-variate Gaussian distribution with the mean  $\mu_i = \langle \mu_x, \mu_y \rangle_i$ , standard deviation  $\sigma_i = \langle \sigma_x, \sigma_y \rangle_i$ , and correlation coefficient  $\rho_i$ .

Hard parameter sharing is adopted to share the same fusion outputs to predict multiple outputs at each step. Two linear layers are used to predict the parameterized distribution  $\langle \mu_{i+1}, \sigma_{i+1}, \rho_{i+1} \rangle$  and the shot type  $\hat{s}_{i+1}$  at  $(i+1)$ -th stroke by combining the target player embedding  $p_{i+1}$  and the fusing output  $z_i$ , respectively:

$$\hat{s}_{i+1} = \text{softmax}((z_i + p_{i+1})W^s), \quad (16)$$

$$\langle \mu_{i+1}, \sigma_{i+1}, \rho_{i+1} \rangle = (z_i + p_{i+1})W^a, \quad (17)$$

where  $W^s \in \mathbb{R}^{d \times N_s}$  and  $W^a \in \mathbb{R}^{d \times 5}$  are two learnable matrices. The predicted area coordinates are sampled by  $\langle \hat{x}_{i+1}, \hat{y}_{i+1} \rangle \sim \mathcal{N}(\mu_{i+1}, \sigma_{i+1}, \rho_{i+1})$ . The reason for adding the target player embedding to the fused contexts is to specify the player who returns the stroke.

We minimize cross-entropy loss  $\mathcal{L}_{type}$  to learn the prediction of shot types:

$$\mathcal{L}_{type} = - \sum_{r=1}^{|R|} \sum_{i=\tau+1}^{|S_r|} s_i \log(\hat{s}_i). \quad (18)$$

We also minimize the negative log-likelihood loss  $\mathcal{L}_{area}$  to learn the prediction of area coordinates:

$$\mathcal{L}_{area} = - \sum_{r=1}^{|R|} \sum_{i=\tau+1}^{|S_r|} \log(\mathcal{P}(x_i, y_i | \mu_i, \sigma_i, \rho_i)). \quad (19)$$

The total loss  $\mathcal{L}$  of our model is jointly trained with:

$$\mathcal{L} = \mathcal{L}_{type} + \mathcal{L}_{area}. \quad (20)$$

## Results and Analysis

### Experimental Setup

**Dataset.** Since there is no public dataset of stroke event records, we collected real-world badminton singles matches from public sources<sup>1</sup> and asked domain experts to manually label them. The dataset contains 75 high-ranking matches from 2018 to 2021 played by 31 players from men's singles and women's singles. After filtering flaw data, e.g., replay highlights, the dataset contains 180 sets, 4,325 rallies, and 43,191 strokes. The average length of the rallies is 10. There are 10 shot types defined by domain experts for distinguishing the strokes: *net shot, clear, push/rush, smash, defensive shot, drive, lob, drop, short service, and long service*.

For the stroke forecasting task, each stroke contains the id of the rally, the order of the stroke in a rally, the player returning the stroke, the shot type, and the area coordinates where the shuttle was returned to. We split the first 80% of the rallies of each match as training data to ensure that the model is equipped with past information of all players, and the remaining rallies were used for testing. We conducted 5-fold cross-validation for tuning hyper-parameters.

**Implementation Details.** The dimension of embeddings and contexts ( $d$ ) was set to 32, the number of heads ( $h$ ) used in multi-head attention and multi-head type-area attention was set to 2, and the inner dimension of feed-forward layer was 64. The max sequence length of a rally was 35.  $n$  is the rally length and varies in different rallies. The layer normalization (Ba, Kiros, and Hinton 2016) and dropout technique with a dropout rate of 0.1 were used for each sub-layer, similar to (Vaswani et al. 2017). The batch size was 32 and the number of training epochs was 150 using Adam (Kingma and Ba 2015) as the optimizer. The learning rate was set to 0.0001. In the training phase, we adopted zero padding for sequences and used ground truth labels as the next step input in decoding. In the evaluation phase, we replaced the ground truth labels with sampled shot types and area coordinates. Following the procedures for evaluating stochastic models in previous work, e.g., (Amirian, Hayet, and Pettré 2019), we generated  $K = 10$  samples and took the closest one to ground truth for evaluation. The input of area coordinates was normalized with the mean as zero. All the training and evaluation phases were conducted on a machine with Intel i7-8700 3.2GHz CPU, Nvidia GTX 2070 8GB GPU, and 32GB RAM. The average results from 10 runs are reported. Our code is available at <https://github.com/wywyWang/ShuttleNet>.

**Baselines** Due to the lack of baselines for the proposed tasks, we compared the proposed model with the baselines of various sequence prediction tasks:

- Seq2Seq (Sutskever, Vinyals, and Le 2014) consists of one LSTM as encoder and another LSTM as decoder.
- CF-LSTM (Xu, Yang, and Du 2020) is a feature-cascaded LSTM integrating feature information from previous two steps as dynamic interactions.
- TF (Giuliari et al. 2020) utilizes the Transformer Network to learn contexts of pedestrians.

<sup>1</sup><http://bwf.tv>

Model	$\tau = 8$			$\tau = 4$			$\tau = 2$		
	CE	MSE	MAE	CE	MSE	MAE	CE	MSE	MAE
Seq2Seq	2.5219	1.7124	1.4181	2.5192	1.6674	1.4049	2.5325	1.6799	1.4022
CF-LSTM	<u>2.3138</u>	2.1805	1.6844	<u>2.2623</u>	2.2510	1.7055	<u>2.3860</u>	2.0392	1.5966
TF	2.3843	<u>1.6427</u>	<u>1.4017</u>	2.3881	1.6665	1.4033	2.4243	1.6317	1.3773
dNRI	2.4391	2.4056	1.7903	2.4475	2.3518	1.7822	2.4441	2.3025	1.7587
DMA-Nets	2.4949	1.8419	1.4791	2.6710	1.8463	1.4876	2.5975	1.8436	1.4813
ShuttleNet <sub>P2R</sub> (Ours)	2.3892	1.6665	1.4052	2.3112	1.6296	<b>1.3838</b>	2.3963	1.5900	<u>1.3693</u>
ShuttleNet <sub>R2P</sub> (Ours)	2.3528	1.6864	1.4233	2.3874	1.6278	<u>1.3882</u>	2.3923	<b>1.5627</b>	<b>1.3563</b>
ShuttleNet (Ours)	<b>1.9802</b>	<b>1.5856</b>	<b>1.3802</b>	<b>1.9916</b>	<b>1.5867</b>	1.3896	<b>2.0755</b>	<u>1.5761</u>	1.3747

Table 1: Quantitative results of our models and baselines on different given lengths. The best result in each column is in boldface while the second best result is underlined.

- dNRI (Graber and Schwing 2020) models dynamic entity relations for neural relational inference.
- DMA-Nets (Ji et al. 2021) constructs a hierarchical dynamic attention layer by considering local temporal information and global temporal information for citation forecasting, which is the closest setting to stroke forecasting with multiple outputs at each timestamp.

Due to the lack of official codes, we reproduced CF-LSTM and DMA-Nets by following the corresponding implementation details in their papers. For fair comparison, the same embedding layer, prediction layer, and hyper-parameters were used for all the baselines. Moreover, since all the baselines take single inputs<sup>2</sup>, we concatenated shot types and area and projected them to same dimension of these baselines instead. In addition, the loss function of shot types was added to the baselines to fit the stroke forecasting task.

To better explore the design of our proposed method, we extended two variants of our method in the experiments:

- ShuttleNet<sub>P2R</sub> feeds the outputs of the embedding layer first to TPE and then feeds the outputs of TPE into TRE.
- ShuttleNet<sub>R2P</sub> feeds the outputs of the embedding layer first to TRE and then feeds the outputs of TRE into TPE.

## Quantitative Results

**Comparison with Baselines.** To evaluate the results of shot type prediction, we used cross-entropy (CE), which has been widely used for uncertainty measurement (Schmidt, Mandt, and Hofmann 2019). Moreover, mean absolute error (MAE) and mean square error (MSE) were used for evaluating the predicted area coordinates similar to (Graber and Schwing 2020). We conducted three sets of experiments with the number of observed strokes  $\tau$  set to 8, 4, and 2 to investigate the performance with different numbers of observed strokes. Table 1 reports the best results of different models, which shows that our model consistently outperforms other baselines for both shot types and area prediction in terms of all metrics and different  $\tau$ . Specifically, our method surpasses all the baselines by at least 12.0% and 3.4% in terms of CE and MSE, respectively.

<sup>2</sup>DMA-Nets has multiple inputs but also using concatenation.

As CF-LSTM and dNRI are based on the characteristics of trajectories (such as velocity), they fail to perform well on forecasting area coordinates. In other words, the trajectories in the dataset are different from the human trajectories since they are event-based data and dramatically change positions. Moreover, Seq2Seq, TF, and DMA-Nets are biased in their ability to forecast area coordinates well, which indicates that integrating rally information is insufficient for returning strokes. Also, these methods lack the capacity for turn-based targets in a sequence since they assume each element in the sequence belongs to the same target. With the Transformer-based player extractor that considers player information, our model is capable of extracting the contexts of the players to correctly predict future strokes.

It is worth noting that the performance of both variants of ShuttleNet<sub>P2R</sub> and ShuttleNet<sub>R2P</sub> did not improve for shot types but improved for area coordinates. This indicates the effect on applying the fusing technique. Early integration of player contexts when extracting the rally information and vice versa, both hamper the model’s learning of the other information. Our model thus demonstrates the need of learning each type of information separately and then employing the fusing technique afterward to achieve the best results.

**Ablation Study** An extensive ablation study was conducted to verify the design of ShuttleNet. We developed six variants to investigate the relative contributions of different components introduced in ShuttleNet: 1) w/o L, which is ShuttleNet without the Transformer-based rally extractor, 2) w/o A, which is ShuttleNet without information of Player A in the Transformer-based player extractor, 3) w/o B, which is ShuttleNet without information of Player B in the Transformer-based player extractor, 4) w/o  $\alpha$ , which is ShuttleNet without information weights in the position-aware gated fusion network, 5) w/o  $\beta$ , which is ShuttleNet without position weights in the position-aware gated fusion network, and 6) w/o TAA, which is ShuttleNet with the type-area-attention mechanism replaced by the self-attention mechanism using concatenation of shot types and area.

It should be noted that the related terms of w/o L, w/o A, and w/o B were also removed in the fusion network and the fusion network was removed in w/o A + w/o B, w/o A +

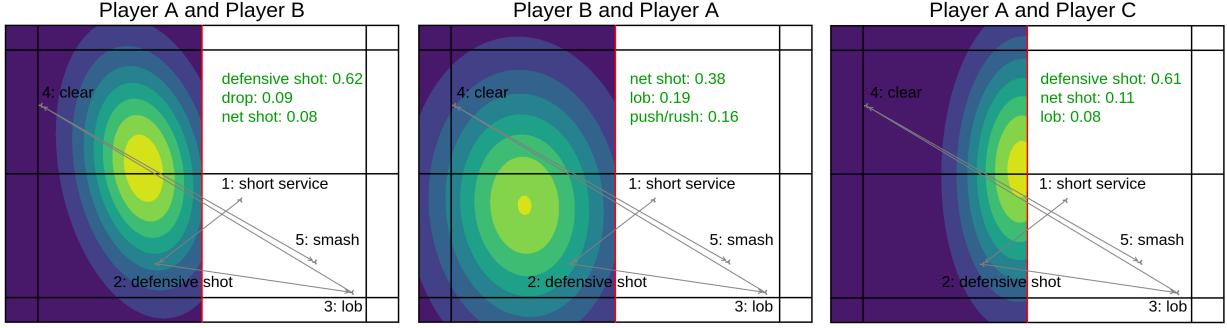


Figure 3: Illustration of three different matchups. The latter one of each case is the player returning the next stroke. That is, Player B is in the left figure, Player A is in the middle figure, and Player C is in the right figure. Black lines represent the court with the net in red. The top three shot types and corresponding probabilities are shown in green text.

Model	CE	MSE	MAE
w/o L	1.9917	1.6471	1.4085
w/o A	1.9900	1.6418	1.4111
w/o B	1.9848	1.6347	1.4071
w/o A + w/o B	2.4169	1.6525	1.4113
w/o A + w/o L	2.1693	2.7223	1.8647
w/o B + w/o L	2.0805	2.6625	1.8343
w/o $\beta$	2.1197	1.6344	1.4002
w/o $\alpha$	1.9822	1.6405	1.4088
w/o TAA	1.9880	1.6628	1.4290
ShuttleNet (Ours)	<b>1.9802</b>	<b>1.5856</b>	<b>1.3802</b>

Table 2: Ablation study of our model.

w/o L, and w/o B + w/o L since there is only one context information of each. Table 2 shows the results with  $\tau = 8$ . We summarize the observations as follows.

**The effect of each context.** It is obvious that removing any one context in ShuttleNet results in a significant performance drop. Also, as expected, using a single context leads to inferior performance in both shot types and area prediction. The results verify the reasonable and effective design of our model. Further, using only the rally context deteriorates the shot performance more substantially, while using only the context of either Player A or Player B negatively impacts more on area performance. We suggest that the locations of the returns rely more on current progress, while the types of the returns can be more likely affected by the styles of each player. Meanwhile, our model demonstrates the ability of incorporating the use of each context.

**The performance of PGFN.** When we discard either fusing weights or position weights, the performance degrades in comparison with ShuttleNet. The results suggest that fusing the information of contexts is effective for different significance. Also, position weights play an important role in the discrepancy of the importance of each stroke.

**Comparison with self-attention mechanism.** To testify the effectiveness of the proposed type-area-attention mechanism, we compared it to the original self-attention mechanism. It is

clear that applying the self-attention mechanism reduces the performance by 0.4% on CE, 4.6% on MSE, and 3.4% on MAE in contrast to ShuttleNet. These results signify that binding shot types and area on the same position is inadequate, whereas attending to different positions enhances the ability to capture the dependency from various aspects.

### Case Study: A Usage of Stroke Forecasting

Analyzing returning strategies in different matchups with the same situation can help understand the possible strategies that the player may use by considering past information to formulate tactics. In this case, the goal is to predict the shot type and area coordinates that the player returns at the sixth stroke of different matchups. Figure 3 shows three top-ranking players of men's singles. It shows that these players are likely to return with a passive shot type when encountering a smash from the opponent. However, area distributions are quite different with respect to the players. The distribution of Player A is more in the middle of court, while both Player B and Player C are closer to the net. Since the fourth stroke is at back court, if the return is nearer the net, the opponent will have a greater distance to move, which will consume more energy of the opponent. This case demonstrates a scenario analysis of stroke forecasting in badminton, and our model is capable of assisting not only coaches for tactic investigation but communities for storytelling.

### Conclusions and Future Works

In this paper, we present ShuttleNet for tackling the challenging stroke forecasting problem. Based on the encoder-decoder architecture, our model incorporates rally information and player information with two extractors. In addition, a position-aware gated fusion network is proposed leveraging information dependency and position weights to decide the importance of rally contexts and contexts of the players for returning each strokes. The quantitative evaluation conducted on the real-world dataset demonstrates the effectiveness of our proposed approach compared to state-of-the-art baselines. For future work, we plan to extend our model to cope with extra conditions, e.g., win and loss, which can be analyzed for advanced tactic investigation.

## Acknowledgments

This work was supported by the Ministry of Science and Technology of Taiwan under Grants MOST-110-2627-H-A49-001, MOST-110-2218-E-A49-018 and MOST-109-2221-E-009-114-MY3.

## References

- Amirian, J.; Hayet, J.; and Pettré, J. 2019. Social Ways: Learning Multi-Modal Distributions of Pedestrian Trajectories With GANs. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2964–2972.
- Ba, L. J.; Kiros, J. R.; and Hinton, G. E. 2016. Layer Normalization. *CoRR*, abs/1607.06450.
- Beal, R.; Chalkiadakis, G.; Norman, T. J.; and Ramchurn, S. D. 2020. Optimising Game Tactics for Football. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems*, 141–149.
- Bechtold, T. 2019. State of analytics: how the movement has forever changed baseball - for better or worse. <https://www.statsperform.com/resource/state-of-analytics-how-the-movement-has-forever-changed-baseball-for-better-or-worse/>.
- Chu, W.; and Situmeang, S. 2017. Badminton Video Analysis based on Spatiotemporal and Stroke Features. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, 448–451.
- Decroos, T.; Bransen, L.; Haaren, J. V.; and Davis, J. 2019. Actions Speak Louder than Goals: Valuing Player Actions in Soccer. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1851–1861.
- Decroos, T.; Bransen, L.; Haaren, J. V.; and Davis, J. 2020. VAEP: An Objective Approach to Valuing On-the-Ball Actions in Soccer (Extended Abstract). In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, 4696–4700.
- Decroos, T.; Dzyuba, V.; Haaren, J. V.; and Davis, J. 2017. Predicting Soccer Highlights from Spatio-Temporal Match Event Streams. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 1302–1308.
- Decroos, T.; Haaren, J. V.; and Davis, J. 2018. Automatic Discovery of Tactics in Spatio-Temporal Soccer Match Data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 223–232.
- Giancola, S.; and Ghanem, B. 2021. Temporally-Aware Feature Pooling for Action Spotting in Soccer Broadcasts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 4490–4499.
- Giuliari, F.; Hasan, I.; Cristani, M.; and Galasso, F. 2020. Transformer Networks for Trajectory Forecasting. In *25th International Conference on Pattern Recognition*, 10335–10342. IEEE.
- Graber, C.; and Schwing, A. G. 2020. Dynamic Neural Relational Inference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.
- He, P.; Liu, X.; Gao, J.; and Chen, W. 2021. Deberta: decoding-Enhanced Bert with Disentangled Attention. In *9th International Conference on Learning Representations*.
- Hochreiter, S.; and Schmidhuber, J. 1997. Long Short-Term Memory. *Neural Comput.*, 9(8): 1735–1780.
- Hsu, T.; Wang, C.; Lin, Y.; Chen, C.; Ju, N. P.; Ik, T.; Peng, W.; Wang, Y.; Tseng, Y.; Huang, J.; and Ching, Y. 2019. CoachAI: A Project for Microscopic Badminton Match Data Collection and Tactical Analysis. In *20th Asia-Pacific Network Operations and Management Symposium*, 1–4.
- Jayanth, S. B.; Anthony, A.; Abhilasha, G.; Shaik, N.; and Srinivasa, G. 2018. A team recommendation system and outcome prediction for the game of cricket. *Journal of Sports Analytics*, 4(4): 263–273.
- Ji, T.; Self, N.; Fu, K.; Chen, Z.; Ramakrishnan, N.; and Lu, C. 2021. Dynamic Multi-Context Attention Networks for Citation Forecasting of Scientific Publications. In *Thirty-Fifth AAAI Conference on Artificial Intelligence*, 7953–7960.
- Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations*.
- Liu, G.; and Schulte, O. 2018. Deep Reinforcement Learning in Ice Hockey for Context-Aware Player Evaluation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, 3442–3448.
- Liu, X.; Yan, J.; Xiao, S.; Wang, X.; Zha, H.; and Chu, S. M. 2017. On Predictive Patent Valuation: Forecasting Patent Citations and Their Types. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 1438–1444.
- Merhej, C.; Beal, R. J.; Matthews, T.; and Ramchurn, S. D. 2021. What Happened Next? Using Deep Learning to Value Defensive Actions in Football Event-Data. In *KDD ’21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 3394–3403.
- Mohajerin, N.; and Rohani, M. 2019. Multi-Step Prediction of Occupancy Grid Maps With Recurrent Neural Networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 10600–10608.
- Morrison, M. 2021. The sports tech trends shaping sports media & fan engagement in 2021. <https://www.statsperform.com/resource/the-sports-tech-trends-shaping-sports-media-fan-engagement-in-2021/>.
- Ovalle, J. E. A.; Solorio, T.; Montes-y-Gómez, M.; and González, F. A. 2020. Gated multimodal networks. *Neural Comput. Appl.*, 32(14): 10209–10228.
- Pappalardo, L.; Cintia, P.; Ferragina, P.; Massucco, E.; Pedreschi, D.; and Giannotti, F. 2019. PlayeRank: Data-driven Performance Evaluation and Player Ranking in Soccer via a Machine Learning Approach. *ACM Trans. Intell. Syst. Technol.*, 10(5): 59:1–59:27.
- Press, O.; and Wolf, L. 2017. Using the Output Embedding to Improve Language Models. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, 157–163. Association for Computational Linguistics.

- Robberechts, P.; Haaren, J. V.; and Davis, J. 2021. A Bayesian Approach to In-Game Win Probability in Soccer. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 3512–3521.
- Schmidt, F.; Mandt, S.; and Hofmann, T. 2019. Autoregressive Text Generation Beyond Feedback Loops. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 3398–3404.
- Sharma, M.; Monika; Kumar, N.; and Kumar, P. 2021. Badminton match outcome prediction model using Naïve Bayes and Feature Weighting technique. *J. Ambient Intell. Humaniz. Comput.*, 12(8): 8441–8455.
- Sicilia, A.; Pelechrinis, K.; and Goldsberry, K. 2019. Deep-Hoops: Evaluating Micro-Actions in Basketball Using Deep Feature Representations of Spatio-Temporal Data. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2096–2104.
- Silver, J.; and Huffman, T. 2021. Baseball Predictions and Strategies Using Explainable AI. In *The 15th Annual MIT Sloan Sports Analytics Conference*.
- Sun, X.; Davis, J.; Schulte, O.; and Liu, G. 2020. Cracking the Black Box: Distilling Deep Sports Analytics. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 3154–3162.
- Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems*, 3104–3112.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, 5998–6008.
- Wang, W.; Chan, T.; Yang, H.; Wang, C.; Fan, Y.; and Peng, W. 2021. Exploring the Long Short-Term Dependencies to Infer Shot Influence in Badminton Matches. *CoRR*, abs/2109.06431.
- Wang, W.-Y.; Chang, K.-S.; Chen, T.-F.; Wang, C.-C.; Peng, W.-C.; and Yi, C.-W. 2020. Badminton Coach AI: A Badminton Match Data Analysis Platform Based on Deep Learning. *Physical Education Journal*, 53(2): 201–213.
- Wang, Z.; Long, C.; Cong, G.; and Ju, C. 2019. Effective and Efficient Sports Play Retrieval with Deep Representation Learning. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 499–509.
- Weeratunga, K.; Dharmaratne, A. T.; and How, K. B. 2017. Application of Computer Vision and Vector Space Model for Tactical Movement Classification in Badminton. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 132–138.
- Xu, Y.; Yang, J.; and Du, S. 2020. CF-LSTM: Cascaded Feature-Based Long Short-Term Networks for Predicting Pedestrian Trajectory. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*, 12541–12548.
- Yoshikawa, Y.; Shishido, H.; Suita, M.; Kameda, Y.; and Kitahara, I. 2021. Shot detection using skeleton position in badminton videos. In *International Workshop on Advanced Imaging Technology 2021*, volume 11766, 312 – 317.