Memory Augmented Deep Generative models for Forecasting the Next Shot Location in Tennis

Tharindu Fernando, *Student Member, IEEE,* Simon Denman, *Member, IEEE,* Sridha Sridharan, *Life Senior Member, IEEE,* and Clinton Fookes, *Senior Member, IEEE*

Abstract—This paper presents a novel framework for predicting shot location and type in tennis. Inspired by recent neuroscience discoveries we incorporate neural memory modules to model the episodic and semantic memory components of a tennis player. We propose a Semi Supervised Generative Adversarial Network architecture that couples these memory models with the automatic feature learning power of deep neural networks, and demonstrate methodologies for learning player level behavioural patterns with the proposed framework. We evaluate the effectiveness of the proposed model on tennis tracking data from the 2012 Australian Tennis open and exhibit applications of the proposed method in discovering how players adapt their style depending on the match context.

Index Terms—Neural Memory Networks, Generative Adversarial Networks, Tennis Shot Prediction, Player Behaviour Analysis

1 Introduction

The ability of professional athletes to accurately anticipate opponents' actions during a fast-ball sports such as tennis is considered a remarkable feat [1]. Considering the fact that present day ball speeds exceed 130mph, the time required by the receiver to make a decision regarding the opponents' intention, and initiate a response could exceed the flight time for the ball [1], [2], [3], [4].

Several studies have shown that this reactive ability is the product of pattern recognition skills that are obtained through a "biological probabilistic engine", that derives theories regarding opponents intentions with the partial information available [1], [5], [6]. For instance, it has been shown that expert tennis players are better at detecting events in advance [1], [7] and posses better knowledge/ expertise of situational probabilities [3]. Further investigation of human neurological structures have revealed that those capabilities occur due to a bottom-up computational process [1] within the human brain, from sensory memory to the experiences stored in episodic memory [8], [9] and knowledge derived in semantic memory [9], [10].

Despite the growing interest among researchers in the machine learning domain in better understanding factors influencing decision making in fast-ball sports, there have been very few studies transferring the observations of the underlying neural mechanisms to neural modelling in machine learning. Current state-of-the-art methodologies try to capture the underlying semantics through a handful of handcrafted features, without paying attention to essential mechanisms in the human brain, where the expertise and observations are stored and knowledge is derived. It is a broadly established fact that handcrafted features only capture abstract level semantics in a given environment [11], [12], [13], [14] and it is proven that these ill represent the context in several data mining and knowledge discovery tasks [15].

The goal of this paper is to derive a deep learning model to anticipate the next shot location in tennis, given current match context and a short history of player and ball behaviour from Hawk-eye ball tracking data [16]. Our predictions comprise the next shot location as well as the type of the shot to anticipate.

Inspired by the automatic loss function learning power of Generative Adversarial Networks (GAN) [12], [14], [17], [18], [19], [20], [21] and the capability of neural memory models [10], [17], [22], [23] to store and retrieve semantic level abstractions from historical agent behaviour, we propose a Memory augmented Semi Supervised Generative Adversarial Network (MSS-GAN). We demonstrate that the proposed framework can be utilised not only for high performance coaching [24], [25], and designing intelligent camera systems for automatic broadcasting [26], [27] where the system anticipate the next shot and shot type to better capture the player behaviour; but also for better understanding of player strategies, strengths and weaknesses.

Due to the conditional nature of the proposed framework, the derived knowledge from the proposed memory modules can be utilised for demonstrating player behaviour changes when encountering different contexts. The main contributions of the proposed work are summarised as follows:

1) We introduce a novel end-to-end deep learning method that learns to anticipate player behaviour.
2) We propose a Semi Supervised Generative Adversarial Network architecture that is coupled with neural memory modules to jointly learn to generate the return shot trajectory and to classify the shot type.
3) We demonstrate how the proposed framework could be utilised to infer player styles and opponent adaptation strategies.
4) We perform an extensive evaluation of the proposed

• *T. Fernando, S.Denman, S. Sridharan, C. Fookes are with Image and Video Research Laboratory, SAIVT, Queensland University of Technology, Australia.*
E-mail: t.warnakulasuriya@qut.edu.au

Manuscript received

method on tennis player tracking data from the 2012 Australian Tennis open.

5) We provide comprehensive analysis on the contribution of each component in the proposed framework by evaluating the proposed method against a series of counterparts.

2 Related Work

2.1 Sports Prediction

With the recent advancements in ball and player tracking systems in sports, there has been increasing interest among researchers to utilise this data in numerous data mining and knowledge discovery tasks.

In [28] the authors utilise a possession value model for predicting points and behaviour of the ball handler in basket ball, assuming that the player behaviour depends only on the current spatial arrangement of the team. In [27] a future player location prediction strategy is applied to move a robotic camera, with applications to automatic broadcasting. Wei et al. [29] proposed a graphical model for predicting the future shot location in tennis. They utilise handcrafted, dominance features together with the ball bounce location, ball speed and player feet locations when determining the future player behaviour. This model is further augmented in [5] where the authors utilise a Dynamic Bayesian Network to model the same set of features.

However recent studies in the sports prediction field [15] have demonstrated the importance of learning the underlying feature distribution in an automatic fashion. For instance in [15] the authors learn a dictionary of player formations in soccer for classifying the outcome of a shot. Even though they achieve comprehensive advancement towards automatic feature learning with player trajectories, those systems cannot be directly applied to model player strategies in tennis. When anticipating future player behaviour, based on the neurological observations presented in [1], [7] it is vital to incorporate a player's past experiences and derived knowledge into the context modelling process.

2.2 Neural Memory Networks

A memory module is required to store important facts from historical information. Neural memory modules have been extensively applied in numerous domains [17], [23], [30], [31], where the model learns to automatically store and retrieve important information that is vital for the prediction task.

In the reinforcement learning domain, Horzyk et al. [9] proposed an episodic memory architecture composed of a tree-structured memory. In a similar line of work, the authors in [10], [32], [33], [34] investigate possible interaction structures for semantic memory. However all these frameworks are proposed in the reinforcement learning domain and a substantial amount of re-engineering is required to adapt those strategies to the supervised learning domain. Furthermore, adaptation of the structure is necessary for modelling player specific knowledge.

We build upon the tree-memory structure proposed in [35]. The authors in [35] propose this model to map longterm temporal dependencies. We expand this memory structure and propose a novel neural memory structure for episodic memory (EM) and suggest a framework for multiple memory interactions and propagating the knowledge from the EM to the semantic memory (SM).

2.3 Generative Adversarial Networks

Generative adversarial networks (GAN) belong to the family of generative models, and have achieved encouraging results for image-to-image synthesis [13], [17], [18]. These models partake in a two player adversarial game where the Generator (G) ties to fool the Discriminator (D) with synthesised outputs while the D tries to identify them.

There exist numerous architectural augmentations for GANs. For instance, in [36] the authors utilise a recurrent network approach for handling sequential data. Most recently authors in [14] have utilised the GAN architecture for visual saliency prediction and further augmented it with memory architectures in [17] for capturing both low and high level semantics in modelling human gaze patterns.

We are inspired by the Semi-Supervised conditional GAN (SS-GAN) proposed in [11], where the authors couple the unsupervised loss of the GAN together with a supervised classification objective. The authors have shown this to enhance the generator's performance by incorporating class specific semantics into the synthesis process. We enhance the SS-GAN model by coupling it with neural memory networks by drawing parallels with recent neurological observations, and propose avenues to achieve player level adaptations of the model.

3 Architecture

We are motivated by the neuroscience observations provided in [1]. They present strong evidence towards activations of brain areas known to be involved with perception: Episodic Memory (EM) where personal experiences are used to determine the similarities between current sensory observation and the stored experiences [8], [37]; and Semantic Memory (SM) where foundations of knowledge and concepts are stored [9], [10].

Figure 1 shows the overall structure of the proposed approach. The Perception Network (PN) (see Sec. 3.1) processes incoming images to obtain an embedding that represents the shot. This is combined with embeddings from the Episodic Memory (EM) (Sec. 3.2) and Semantic Memory (SM) (Sec. 3.3) to predict the next shot via the Response Generation Network (RGN) (Sec. 3.4). Note that the output of the PN is also fed to the memories to learn historic behaviour. Finally, we use a GAN framework to learn the network, with the predicted shot from the RGN being passed to the Discriminator to determine if it is a realistic shot or not.

For the rest of the paper we use the following notation. All weights are denoted W, f denotes forget gates and $F^{NE}(X)$ denotes a function that passes input X through one of more layers of the network NE. $LSTM$ denotes a layer consisting of LSTM cells. c_t represents the current state (context) of the game. The current representation of the memory at time instant t is denoted by M_t, while the query vector that is used to read the memory is denoted

by q_t and m_t represents the output of the memory read operation. Attention values, η, denote the attention given to the content of M_t to answer the query vector q_t, while the normalised attention values are denoted by α.

3.1 Perception Network (PN)

Our observations are from Hawk-eye player tracking data [16], which stores the ball trajectory and player feet movements along with the ball speed and angle. For each shot event in the database we extract out the ball and player trajectory from the shot start time to the present, and generate an image depicting the perception of the shot receiver.

Fig. 2 (a) illustrates the input to our perception network. The opponent trajectory is denoted in blue where the circle denotes the ending location, and the ball trajectory is shown in red. Shot starting and ending locations are denoted with a yellow star and a circle, respectively. In order to account for the current position of the shot receiver we encode his trajectory in magenta where the circle denotes the ending location. A sample output generated by the proposed framework is given in Fig. 2 (b), where the ball trajectory for the predicted next shot is denoted with a white line. We utilise images to represent our observations as they preserve the relative spatial relationships [1].

Fig. 1 shows the architecture of the proposed Perception Network (PN). Deep Convolution Neural Networks (DCNN) have shown encouraging results when encoding information via automatic feature learning. Hence we utilise a C64-C128-C256-C512-C512-C512-C512-C512 DCNN structure in our PN where Ck denote a Convolution-BatchNorm-ReLU layer with k filters. We then pass this embedding through a fully connected layer with 512 neurons, (FC512), which concatenates the image representation with the current incident speed, s_t, angle, a_t, opponent id, op_t, and the current player points, p_t, in order to generate the current state embedding c_t.

3.2 Episodic Memory (EM)

When considering the human cognitive structure, EM is vital for storing spatio-temporal event information, for helping to form concepts in Semantic Memory (SM), and for guiding the response generation [38]. EM is composed of one's accumulated past experiences, and contains the encoded event information such as what, where and when. This allows us to mentally re-visit past experiences and generate observations comparing them with the current state and respond accordingly [9].

Hence, drawing parallels to a tennis tournament, we pass the observations of each player separately through the PN and store the embedded observations in a memory queue.

When deriving long term relationships among the stored sequential data, Fernando et. al [35] have shown tree-LSTM cells to preserve adjacent temporal relationships and propagate salient information effectively, aiding decision making in the present state. Hence we utilise tree-LSTM cells to summarise the content of our EM queue, arranging the content in a tree structure.

The functionality of the EM module can be summarised by the operations Memory Read (see Section 3.2.1) and Memory Write (3.2.2).

3.2.1 *Memory read*

Let $\ddot{x}_t = [x_t, s_t, a_t, op_t, p_t]$, where x_t is the current observation image, s_t is the incident speed, a_t is the incident angle, op_t is the opponent id and p_t is the points for shot receiver and the opponent. Then the encoded vector for the current state representation from the PN network at time step t can be denoted as,

$$c_t = F^{PN}(\ddot{x}_t), \tag{1}$$

where F^{PN} is the Perception Network (see Sec. 3.1) and $c_t \in \mathbb{R}^{1\times k}$. Consider the EM to have N embeddings stored, with each having the size $1 \times K$. Similar to [35], when computing the EM output at time instance t, we extract the memory tree configuration at time instance $t-1$. Let $M^{EM}_{(t-1)} \in \mathbb{R}^{k\times 2^l}$ be the memory matrix resultant from concatenating nodes from the top of the tree to depth $l = [1, \ldots]$.

The tree memory architecture hierarchically maps the memory with a bottom up tree structure. The bottom layer of the tree stores all historic states and the hierarchy is created such that the most significant features are concatenated, propagating two temporally adjacent neighbours to the upper layer. This can be seen as each layer performing information compression. Hence the top most layer contains the most compressed version of the information present in the memory. In the immediately preceding layer this compression is slightly relaxed. Using the depth (l) hyper-parameter we extract out information from multiple levels from the tree top, allowing us to extract different levels of abstraction. l ranges from $l = 1$ to the total number of layers in the tree hierarchy, where 1 denotes that the information is extracted only from the tree top. The optimal value for l is evaluated experimentally and this evaluation is presented in Fig. 8 (b).

The read head on the EM passes c_t through a read LSTM function, $LSTM^{EM,r}$, to generate a vector to query the memory such that,

$$q^{EM}_t = LSTM^{EM,r}(c_t, M^{EM}_{(t-1)}), \tag{2}$$

Then an attention vector η^{EM}_t determines the similarity between the current context vector c_t and the memory representation $M^{EM}_{(t-1)}$ by attending over each element such that,

$$\eta^{EM}_{(t,j)} = q^{EM}_t M^{EM}_{(t-1,j)}, \tag{3}$$

where $M^{EM}_{(t-1,j)}$ denotes the j^{th} item in the matrix $M^{EM}_{(t-1)}$, $j = [1, \ldots, 2^l - 1]$. Then the score values are normalised using soft attention [31], [39], [40], generating a probability distribution over each memory element as follows,

$$\alpha^{EM}_{(t,j)} = \frac{\mathbb{E}(\eta^{EM}_{(t,j)})}{\sum_{j=1}^{2^l-1} \mathbb{E}(\eta^{EM}_{(t,j)})}. \tag{4}$$

1. Please note that in Fig. 2 (b) we have used black and white masks without court outlines as it reduces the number of parameters in the prediction module that requires training.

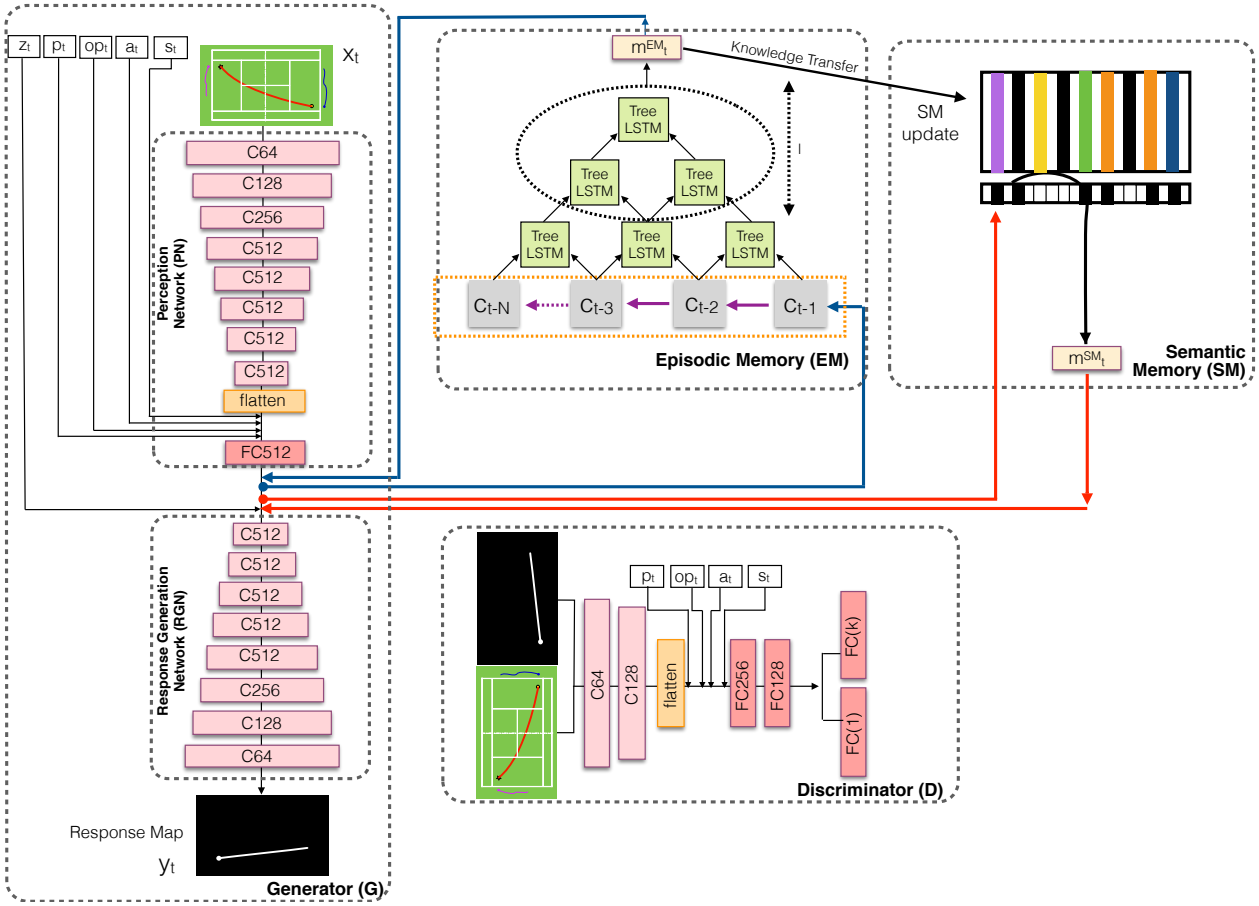
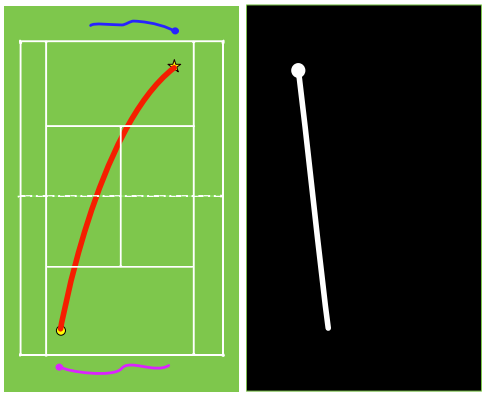


Fig. 1. Proposed MSS-GAN model: The model is composed of the **Perception Network (PN)** which encodes the visual input and concatenates it together with sparse speed (s_t), angle (a_t), opponent id (op_t) and point (p_t) representations; **Episodic Memory (EM)** which stores the temporally adjacent player experience embeddings; **Semantic Memory (SM)** which extracts out knowledge from the EM and a **Response Generation Network (RGN)** which generates a future shot location based on these observations with the aid of a latent noise distribution z_t. The **Discriminator (D)** receives the input perception together with the generated response from the RGN and determines whether the response is a true player behaviour or a synthasised one from the RGN, and classifies the shot type.



(a) (b)

Fig. 2. A sample input and output for the proposed MSS-GAN model. The observed incoming ball trajectory is denoted in red where the starting and ending locations are presented with yellow star and a circle. The opponent and player feet movements are denoted in blue and magenta colours where the circle denotes the ending location of the respective player trajectory. The predicted next shot is shown in (b) and is denoted with a white line where the ball landing location is given by a white circle.

Now we generate the output of the memory read by,

$$m_t^{EM} = \sum_{j=1}^{2^l-1} \alpha_{(t,j)}^{EM} M_{(t-1,j)}^{EM}. \quad (5)$$

3.2.2 Memory Write

The new encoded information, c_t, is appended to the end of the EM queue. This invokes the memory update operation which updates the content of the tree-LSTM cells. Each memory cell contains one input gate, i_t, one output gate, o_t, and two forget gates f_t^L and f_t^R for left and right child nodes. At time instance t each node in the memory network is updated in the following manner,

$$i_t = \sigma(W_{hi}^L h_{t-1}^L + W_{hi}^R h_{t-1}^R + W_{ui}^L u_{t-1}^L + W_{ui}^R u_{t-1}^R), \quad (6)$$

$$f_t^L = \sigma(W_{hf_l}^L h_{t-1}^L + W_{hf_l}^R h_{t-1}^R + W_{uf_l}^L u_{t-1}^L + W_{uf_l}^R u_{t-1}^R), \quad (7)$$

$$f_t^R = \sigma(W_{hf_r}^L h_{t-1}^L + W_{hf_r}^R h_{t-1}^R + W_{uf_r}^L u_{t-1}^L + W_{uf_r}^R u_{t-1}^R), \quad (8)$$

$$\beta = W_{hu}^L h_{t-1}^L + W_{hu}^R h_{t-1}^R, \quad (9)$$

$$u_t^P = f_t^L \times u_{t-1}^L + f_t^R \times u_{t-1}^R + i_t \times tanh(\beta), \quad (10)$$

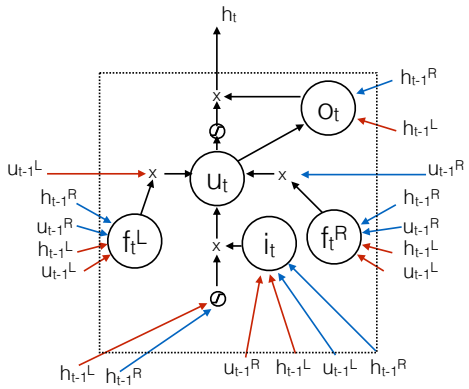


Fig. 3. Tree memory cell architecture. f_t^L, f_t^R, o_t, i_t represents the left forget gate, right forget gate, output gate and input gate respectively. $\times$ represents multiplication

$$o_t = \sigma(W_{ho}^L h_{t-1}^L + W_{ho}^R h_{t-1}^R + W_{uo}^P u_t^P), \tag{11}$$

$$h_t^P = o_t \ \times \ tanh(u_t^P), \tag{12}$$

where h_{t-1}^L, h_{t-1}^R, u_{t-1}^L and u_{t-1}^R are the hidden vector representations and cell states of the left and right children respectively; and h_t^P and u_t^P are the hidden state and cell state representations of the parent node. The relevant weight vectors, W, are represented with appropriate super and subscripts where the superscript represents the relevant child node, and the subscript represents the relevant gate the weight is attached to. The process is illustrated in Fig 3.

3.3 Semantic Memory

According to Wang et. al [10] semantic memory can be considered as the derived knowledge from the specific experiences stored in episodic memory. It doesn't contain any situational information such as what, where and when. However it contains derived salient information from the sequential information.

Hence, in contrast to EM, which is constructed as a queue representing the temporal structure, we design SM as matrix of b elements $M_t^{SM} \in \mathbb{R}^{k \times b}$ where k is the embedding dimension of c_t.

3.3.1 Memory Read

The read operation of the SM is identical to the read operation of the EM. Formally, let the content of the SM at time instance $t-1$ be M_{t-1}^{SM}. Then the memory read operation can be summarised as,

$$q_t^{SM} = LSTM^{SM,r}(c_t, M_{(t-1)}^{SM}), \tag{13}$$

$$\eta_{(t,j)}^{SM} = q_t^{SM} M_{(t-1,j)}^{SM}, \tag{14}$$

$$\alpha_{(t,j)}^{SM} = \frac{\mathbb{E}(\eta_{(t,j)}^{SM})}{\sum_{j=1}^{b} \mathbb{E}(\eta_{(t,j)}^{SM})}. \tag{15}$$

$$m_t^{SM} = \sum_{j=1}^{b} \alpha_{(t,j)}^{SM} M_{(t-1,j)}^{SM}, \tag{16}$$

where $j = [1, \ldots, b]$.

3.3.2 Memory Write

The memory update procedure of the EM triggers the update procedure of the SM. Let the output of the EM tree at time instance t be denoted by m_t^{EM}. We generate a vector $\grave{m}_t$ for the SM update by passing the output of the memory through a write LSTM function $LSTM^{SM,w}$,

$$\grave{m}_t = LSTM^{SM,w}(m_t^{EM}), \tag{17}$$

Then we generate the attention score values,

$$\grave{\eta}_{(t,j)}^{SM} = \grave{m}_t^{SM} M_{(t-1,j)}^{SM}, \tag{18}$$

and normalise them as,

$$\grave{\alpha}_{(t,j)}^{SM} = \frac{\mathbb{E}(\grave{\alpha}_{(t,j)}^{SM})}{\sum_{j=1}^{b} \mathbb{E}(\grave{\alpha}_{(t,j)}^{SM})}. \tag{19}$$

Then we update the SM as,

$$M_t^{SM} = M_{t-1}^{SM}(I - \grave{\alpha}_t \otimes e_k)^T + (\grave{m}_t \otimes e_b)(\grave{\alpha}_t \otimes e_k)^T, \tag{20}$$

where I is a matrix of ones, $e_b \in \mathbb{R}^b$ and $e_k \in \mathbb{R}^k$ are vector of ones and $\otimes$ denotes the outer product which duplicates its left vector b or k times to form a matrix.

3.4 Response Generation Network (RGN)

The proposed RGN takes a latent noise distribution z_t together with the embedded input vector c_t and the memory output vectors m_t^{EM} and m_t^{SM}, and generates the response map, y_t, denoting the ball trajectory of the next shot. Our RGN has the structure CD512-CD512-CD512-C512-C256-C128-C64 where CDk denotes a a Convolution-BatchNormDropout-ReLU layer with a dropout rate of 50%. The RGN generates a response map as illustrated by Fig. 2 (b).

4 Model Learning

Most recently, Generative Adversarial Networks (GAN) have shown exemplary results in image to image synthesis problems [13]. GANs are comprised of two components: a Generator,G, and a Discriminator, D, competing in a two player game. We draw our inspiration from the Semi-Supervised conditional GAN (SS-GAN) [11] architecture where G receives the observed state representation $\ddot{x}_t$ and a random noise vector z_t and tries to synthesise the response map y_t: $G(\ddot{x}_t, z_t) \rightarrow y_t$.

The discriminator receives the current state representation $\ddot{x}_t$ and the generated response map y_t from G and tries to discriminate the actual player responses (real) from generated response maps (fake). We additionally incorporate a classification head D_η in the discriminator, which learns to output the probabilities for shot types (η_t). This attaches a supervised objective to the unsupervised objective in the GAN, enabling it to learn from both labelled and unlabelled data. The (Real/ Fake) validation process contains the unsupervised objective of the SS-GAN where it learns the structure and dynamics of the player responses. The classification objective captures the hierarchical relationships among different shot types and determines when the players utilise them, enforcing the generator to learn

those player adaptation techniques. Formally the objective of the SS-GAN can be defined as,

$$\begin{aligned} \min_G \max_D \; & \mathbb{E}_{\ddot{x}_t, y_t \sim p_{data}(\ddot{x}, y)}[\log D(\ddot{x}_t, y_t)] + \\ & \mathbb{E}_{\ddot{x}_t \sim p_{data}(\ddot{x}), z_t \sim p_z(Z)}[\log(1 - D(\ddot{x}_t, G(\ddot{x}_t, z_t))] + \\ & \lambda_\eta \mathbb{E}_{\ddot{x}_t, \eta_t \sim p_{data}(\ddot{x}_t, \eta_t)}[\log D_\eta(\eta_t | \ddot{x}_t)], \end{aligned} \tag{21}$$

where D_η is the classification head of D and λ_η is a hyper parameter which controls the contributions of the supervised and unsupervised objectives.

4.1 Player Specific Adaptation

Neurological research on human EM has revealed that the EM stores one's own personal experiences rather than the acquired knowledge from other peoples' experiences [37]. This theory is further established by neuroscience research related to tennis shot prediction, where the researches have observed a positive correlation between active practice and performance, and not passive practice [1].

Hence, as the experience of different players depends on the tennis games that they have played, it is not ideal to model EM as a global memory module. Therefore we maintain a separate EM for each player. However it has been shown that SM incorporates extracted knowledge of a cohort of people [10]. When drawing parallels to a tennis scenario SM would represent the rules of the game, the game dynamics, etc. Hence we maintain a global SM for all players. Furthermore, players pay varied levels of attention towards different input features. For instance one player may pay more attention to opponent location where as another may pay more attention to ball incident speed and angle. Hence it is vital to maintain player specific PNs and RGNs. Fig. 4 illustrates the local and global memory architecture that we propose, enabling player specific adaptations. EM = unique for each Player SM = global (same for everyone)

5 Evaluations and Discussion

5.1 Dataset

We used tennis tracking data from the Hawk-eye system for an entire tournament of the 2012 Australian Open Men's singles. The system records (x, y, z) positions of the ball as a function of time, along with the player feet positions at millisecond granularity, and other meta data including current points, time duration, sever and receiver. The dataset consists of around 10,000 shots, however as the tournament progresses in a knock-out format, similar to [5] we focus our analysis on the top 3 players.

When training the respective player models, we maintain the chronological order of the inputs from the start of the tournament; every shot that has occurred in a game the player has played is fed to the model in this order. Therefore we retain the order of experiences that have occurred, allowing the episodic and semantic memories to replicate the player's brain activities.

5.2 Shot Type Prediction

In this experiment we evaluate the performance of the proposed method when predicting the outcome of the next shot, where the model predicts whether the next shot is either a winner, an error or a return shot. The details of the shots played by each player are given in Tab. 1. For each player, we utilise 70%, 25% and 5% of shots, chronologically, for training, testing and validation.

TABLE 1
Counts for different shot types played by the top 3 players in our dataset

Player	Total Shots	Winner	Error	Return
Djokovic	3,410	378	554	2,478
Nadal	3,488	215	426	2,847
Federar	1,882	187	579	1,116
Total	**8,780**	**780**	**1,559**	**6,441**

5.2.1 Baselines

As the first baseline model we utilise the Dynamic Bayesian Network (DBN) model proposed in [5]. This model utilises speed, ball bounce location, player feet locations and a set of hand crafted dominance features to classify the return shot type. In the next baseline we adapt the classifier model proposed in [41] for classifying the shot success in table tennis matches. Essentially the model utilises hits (number of shots that the player of interest has played thus far from the beginning of the rally), a shot quality variable computed by speed and bounce position of the incoming shot, shot direction, player ids, and current player points. Then we classify these features using a SVM classifier.

To provide a fair comparison against deep learning models, we pass ball and player trajectories through an LSTM model [42] and generate the respective classification by passing the LSTM embeddings through softmax classification function [43].

5.2.2 Validation

Similar to [5] for all the models we measure the area under the ROC curve (AUC) to assess the performance. Tab. 2 presents our evaluation results.

We observe the poorest performance from [41] as it doesn't posses any capacity to oversee player or scene specific context. The model neither incorporates historical player behaviour nor the ball trajectory information when predicting the shot outcome. The baseline LSTM model incorporates this information, and gains a significant performance boost compared to [41]. We would like to compare it against the model of Wei et. al [5], where the former model has the attained the classification process through an automatic feature learning method. Comparatively similar accuracies emphasises the importance of the hierarchical feature learning process of deep learning models, which automatically learn semantic correspondences through back propagation. The model proposed by Wei et. al captures the current context through hand crafted player specific and game specific context. In contrast, the proposed model learns these attributes automatically via modelling the player knowledge and experiences through neural memory networks and outperforms the state-of-the-art baselines.

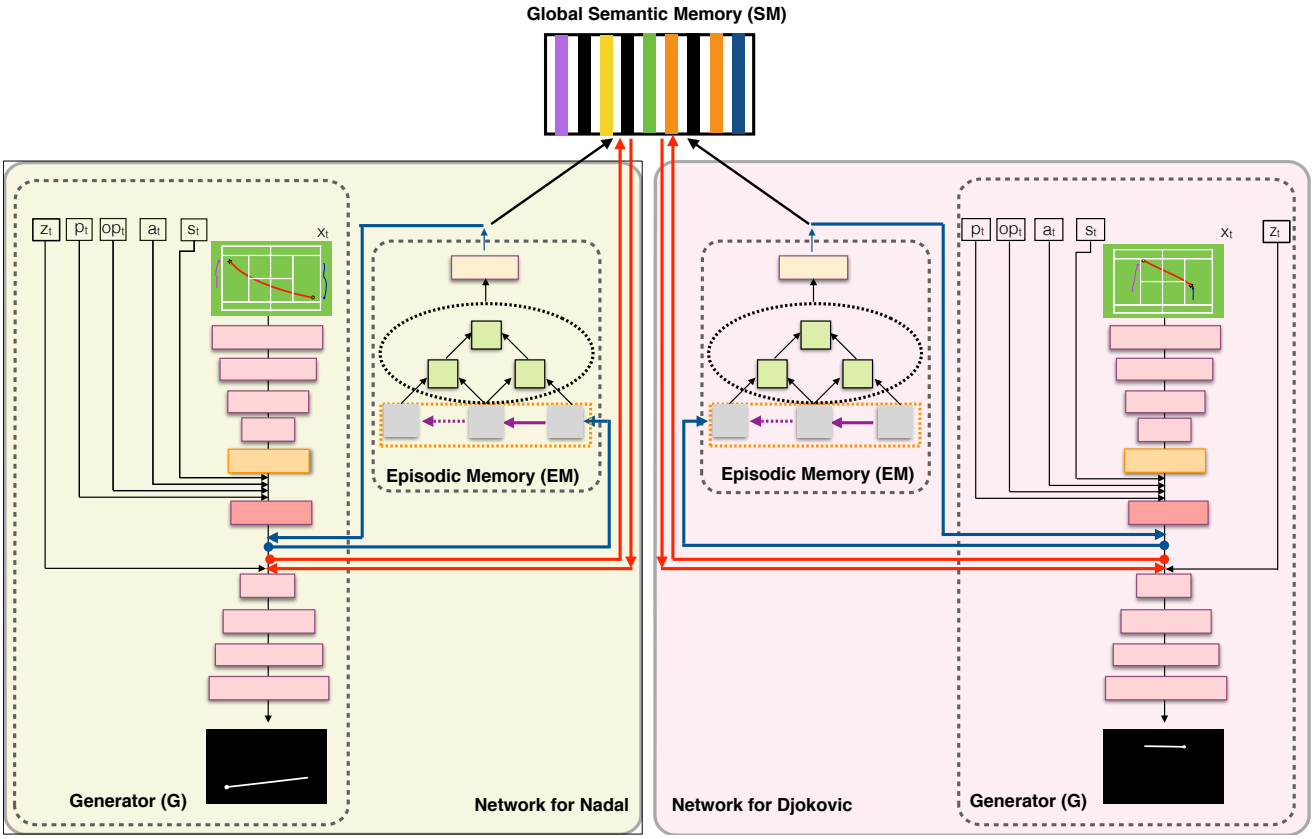


Fig. 4. Each player model is composed of a player specific PN, EM and RGN however SM is shared between the players. For the clarity of the illustration we demonstrate the model for only a 2 player scenario, however it could be directly extended to any number of players.

TABLE 2
Shot Type Classification Results: We measure the area under the ROC curve (AUC) to assess the performance. NA stands for Not Available as the metric is not evaluated in that baseline method

Method	AUC- Winner Shots	AUC- Error Shots	AUC- Return Shots
Draschkowitz et. al [41]	52.45	61.33	61.89
LSTM model	64.61	66.60	72.69
Wei et. al [5]	71.60	77.03	NA
MSS-GAN	**82.65**	**88.33**	**89.01**

5.3 Shot Location Prediction

In this experiment we test the performance of the proposed next shot location prediction method against the state-of-the-art baselines. Similar to the previous experiment, for each player, we utilise 70%, 25% and 5% of shots, chronologically, for training, testing and validation.

5.3.1 Baselines

As the first baseline model we incorporate the continuous shot location prediction model of [5]. As the next baseline we utilised the trajectory prediction method of [44]. We adapted the system of [44] where they try to predict the future ball trajectory from the observed trajectory. Inspired by the encouraging results obtained in [45] for golf shot prediction, we model the same features of [44] using the method of [45]. The model proposed in [45] is a deep LSTM model which maps sequential hierarchical relationships among the input features.

5.3.2 Validation

For all the methods we measure the Euclidian distance between the predicted and ground truth locations in meters as the prediction error. Tab. 4 presents the performance of the proposed method against the 3 state-of-the-art baselines. As [44] utilises a simple physical model to predict the future ball trajectory, assuming that the ball follows a parabolic arc under gravitational force, we observe poor performance from it. The model in [45] improves upon this via hierarchical feature learning from the ball trajectory characteristics through a recurrent model.

The method of [5] builds upon the trajectory features, via incorporating the game and player context into the prediction framework, however, fails to capture salient information from longterm dependencies among player behavioural patterns. In contrast, we capture those hierarchically, allowing us to effectively propagate this information into the future action generation pipeline.

TABLE 3
Shot Location Prediction Results: We measured the distance between the predicted and ground truth shot locations in meters and report the Average (μ) and the Standard Deviation of this distances (σ)

Method	Nadal		Djokovic		Federer		Overall	
	μ	σ	μ	σ	μ	σ	μ	σ
Kumar et. al [44]	4.23	1.0043	3.87	0.9145	5.83	2.1534	4.64	1.357
Jansson et al. [45]	2.11	0.8114	1.95	0.7671	3.41	0.9833	2.49	0.8539
Wei et. al [5]	1.72	0.5430	1.64	0.3034	2.32	0.7016	1.89	0.5160
MSS-GAN	**0.87**	**0.0302**	**0.79**	**0.0210**	**1.14**	**0.0330**	**0.93**	**0.0280**

5.4 Ablation Experiments

To further demonstrate our proposed approach, we conducted a series of ablation experiments, identifying the crucial components of the proposed methodology. In the same setting as Sec. 5.3, we evaluate the proposed MSS-GAN model against a series of counterparts constructed by removing strategic components of the MSS-GAN as follows:

1) $G^G/(D, M^{EM}, M^{SM})$: removes the discriminator, M^{EM} and M^{SM} models and is trained with the supervised learning objective.
2) $(G, D)^G/(M^{EM}, M^{SM})$: removes the M^{EM} and M^{SM} models and is trained with the GAN objective of [13].
3) $(G, D)^{G,*}/(M^{EM}, M^{SM})$ Similar to the method 2, however it is trained with the semi supervised objective defined in Eq. 21.
4) $(G, D, M^{EM})^G/(M^{SM})$: removes the M^{SM} and is trained globally without the player adaptation techniques of Sec. 4.1.
5) $(G, D, M^{EM})^L/(M^{SM})$: Similar to the previous model however uses the player adaptation techniques of Sec. 4.1.
6) $(MSS-GAN)^G$: Same as the proposed MSS-GAN model, however we train one global model without the player adaptation technique.

We observe the lowest accuracy from the $G^G/(D, M^{EM}, M^{SM})$ model which fails to capture the context and the dynamics of the tennis game through an off the shelf supervised learning loss function. We observe a significant performance boost with the unsupervised GAN objective. However it lacks the capacity to capture salient information from the historical embeddings, and as such the performance increases from $(G, D)^G/(M^{EM}, M^{SM})$ to $(G, D, M^{EM})^G/(M^{SM})$ where the latter has further capacity to understand the player level behavioural differences. We would like to compare the performance of the $MSS-GAN^G$ model against the proposed $MSS-GAN$ model, where the former model learns one single network for all the players. The different in accuracies are significant, emphasising the importance of capturing player level semantics separately. We further compare models $(G, D)^G/(M^{EM}, M^{SM})$ against $(G, D)^{G,*}/(M^{EM}, M^{SM})$. Similar to observations of [11], [46], our results demonstrate the importance of a semi supervised learning objective, which compliments the generator in learning the hierarchical attributes of the scene.

5.5 Qualitative Evaluations

Qualitative evaluations from the proposed MSS-GAN model are presented in Fig. 5. We denote the incoming shot in red where the yellow star and the circle denotes the shot starting and ending locations. The feet movements of the player of interest and the opponent are denoted in magenta and blue. Ground truth and predicted trajectories are denoted with cyan and yellow lines, respectively.

In the first two rows we have presented accurate predictions while in the last row we observe some deviations from the ground truth. However they are all possible next shot locations for the incoming shots, and for scenarios such as Fig. 5 (h) and (i) we observe that the predicted shot maximises the wining probability for the player of interest compared to the actual ground truth trajectory.

Fig. 6 shows the distribution of activations from the first layer of the proposed EM tree for the Djokovic model, for the observed shot trajectory given in Fig. 5 (a). As N = 1,100, there exist 1,100 memory slots in this layer, hence the historical embeddings range from t to $t-1100$. For different peaks and valleys in the memory activations, we show what the model has seen at that particular time step.

We observe higher responses for recent events as well as for similar shot patterns in the long term history. It can be clearly seen that the activation pattern takes the current context into consideration and attends all the previous experiences of the player stored in the memory, in order to determine the optimal way to behave. Hence we observe higher activations to similar shot patterns that reside within the entire history captured by the EM module (see the activation peeks between $t-300$ to $t-400$ and $t-800$ to $t-1100$).

5.6 Impact of the Training Data

In this section we investigate the impact of training set size, input image dimensions, and the selection of player identities for extraction of training data, for the performance of the proposed MSS-GAN model.

5.6.1 Impact of Training Set Size

In order to analyse the robustness of the proposed model to different training set sizes we analyse the distribution of the average shot location prediction error and training time for an epoch on a single core of an Intel Xeon E5-2680 2.50GHz CPU, for different training set sizes for the training data defined in Sec. 5.3. For this evaluation we used the same testing and validation sets used in Sec. 5.3 which are not used for training. When creating the reduced training set, we take data from the first shot Nadal played in the tournament up to the specified number of samples, i.e. when training with 100 samples we take the first 100 shots, and when training with 1000 samples we take the first 1000 shots.

TABLE 4
Ablation Experiment Results: We measured the distance between the predicted and ground truth shot locations in meters.

Method	Nadal	Djokovic	Federer	Average
$G^G/(D, M^{EM}, M^{SM})$	2.03	1.98	3.88	2.63
$(G, D)^G/(M^{EM}, M^{SM})$	1.63	1.59	2.18	1.80
$(G, D)^{G,*}/(M^{EM}, M^{SM})$	1.44	1.32	1.95	1.57
$(G, D, M^{EM})^G/(M^{SM})$	1.21	1.15	1.44	1.26
$(G, D, M^{EM})^L/(M^{SM})$	1.13	1.10	1.36	1.19
$(MSS - GAN)^G$	1.02	1.03	1.23	1.09
MSS-GAN	**0.87**	**0.79**	**1.14**	**0.93**

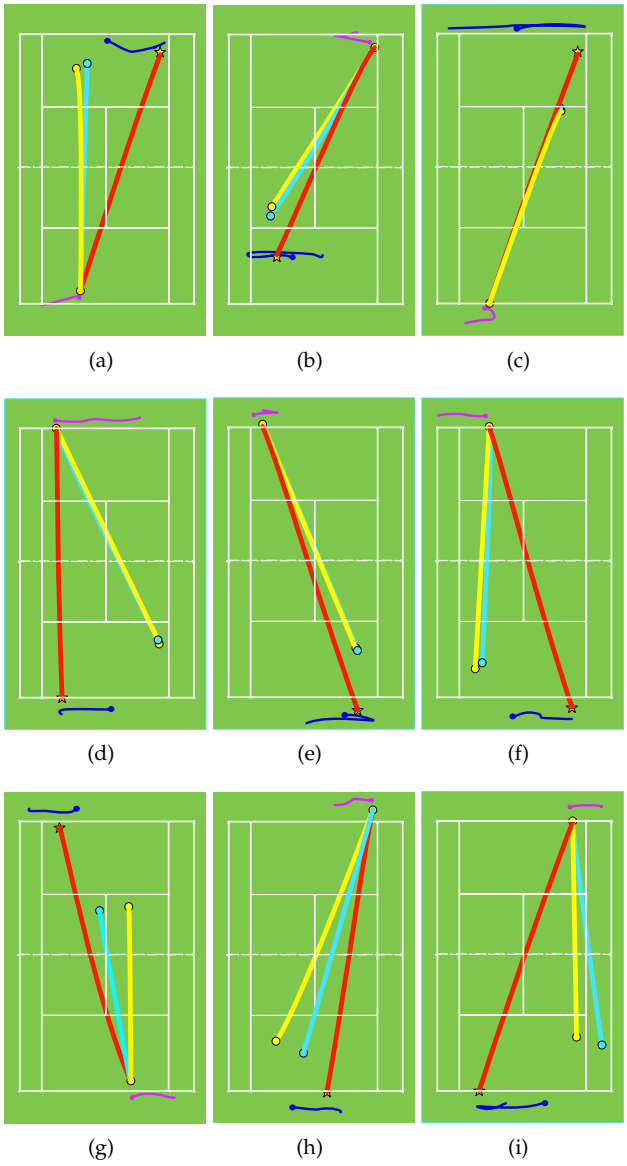


Fig. 5. Qualitative results from the proposed MSS-GAN model. Incoming shot is in red where the yellow star and the circle denotes the shot starting and ending locations. Feet movements of player of interest and the opponent are in magenta and blue colours. Ground truth and predicted trajectories are denoted in cyan and yellow lines, respectively.First two rows show accurate predictions while the 3rd row shows scenarios where the predicted trajectory deviates from the ground truths. However in (g) and (h) we observe that the predicted trajectory maximises the opportunity of the winning probability of the player of interest. Please note that we have overlaid the predictions from RGN on top of court outlines for the clarity of illustration

In Fig. 7 we visualise the average shot location prediction error against different training set sizes (in red) and the elapsed time per epoch (in blue). As could be expected, the training time increases gradually as more samples are added to the corpus. However the model accuracy converges around 1600 training examples and we do not observe substantial improvement, irrespective of the introduction of additional examples. 1600 shots are roughly equivalent to the total number of shots that he has played in first 3 matches.

5.6.2 *Impact of Input Image Size*

In our implementation the input/output image size is set to be 512 x 512 pixels. In order to demonstrate the robustness of the proposed MSS-GAN model to different input image sizes we evaluated different input/outputs sizes of 2048 x 2048, 1024 x 1024, 256 x 256, 128 x128, and 64 x 64 pixels and the evaluated shot location prediction error for the same conditions as in Sec. 5.3. The receptive field sizes of the layers and the number of kernels are kept constant in this experiment. Tab. 5 shows the average error for the predictions, and we observe that for sizes 1024 x 1024 and 256 x 256 there isn't significant deviation in performance compared to the results of Tab. 4. However the performance starts degrading when the input/output sizes are smaller than 256 x 256 pixels. This is due to the granularity of the spatial representation. When the input size is smaller the ground truth trajectory is substantially downscaled to represent it in the image representation. Hence the input and ground truth trajectory representations are less informative to the model. Hence reduction in the granularity of the image representation leads to poor performance. Conversely when the input size it too large it leads to a substantial increase in the number of parameters that require training. Hence the model could not be effectively trained using the limited data available, again leading to poor performance. It should be noted that the same input and output sizes are used for the convenience in the implementation. However this is not a constraint of the proposed MSS-GAN method.

TABLE 5
Shot Location Prediction Results for Different Input/ Output Image Sizes: We measured the distance between the predicted and ground truth shot locations in meters.

Input/ Output size	Average
2048 x 2048	1.05
1024 x 1024	0.94
512 x 512	**0.93**
256 x 256	**0.93**
128 x 128	1.13
64 x 64	1.24

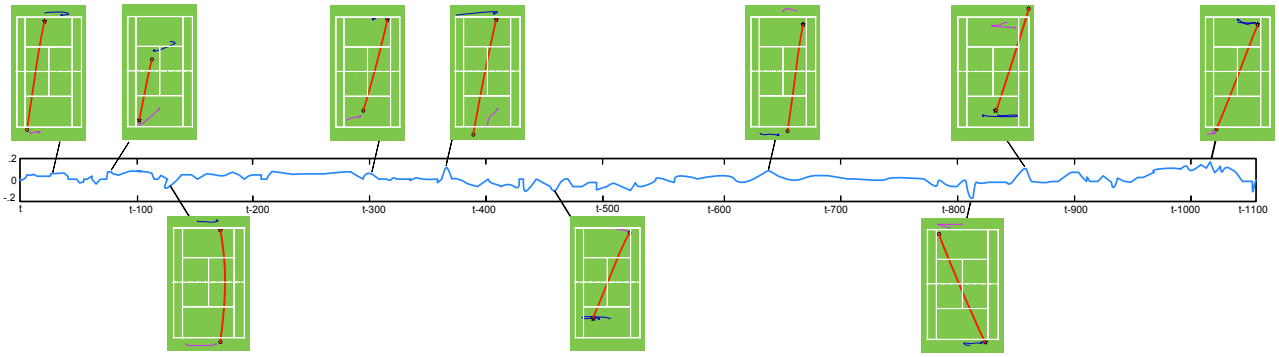


Fig. 6. Distribution of memory activations from the first layer of the EM module for the observed shot in Fig. 5 (a). This layer contains 1,000 memory slots which are denoted t to t -1100 indicating the history that has been observed. For different peaks and valleys of the memory activations we also show what the memory has seen at those particular time steps. The model generates higher activations for similar shot patterns, and activations closer to zero for cases where the player has observed different experiences. We effectively propagate this information from the first layer of the memory to the top most layer via combining the salient information in a hierarchical manner.

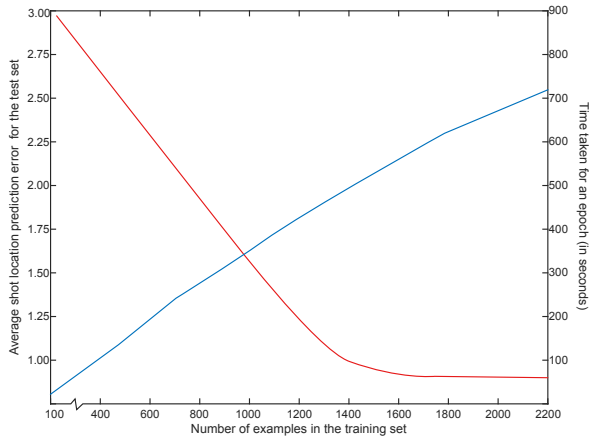


Fig. 7. Change in average shot location prediction error (in meters) and training time for an epoch (in seconds), against training set size

5.6.3 Impact of Selected Players

Similar to [5] we present the main evaluation in Sec. Sec. 5 only for the top-3 players as it allows us to present direct comparisons with [5]. However in order to better evaluate the predictive power of the proposed MSS-GAN model we randomly selected 4 players from the 2012 Australian Open Men's singles tournament who had progressed up to the fourth round. We trained the model with the data from first 3 rounds and tested on the 4th round. For these players we evaluate the shot location prediction accuracy as it is a more challenging task than predicting the shot type.

In order to better appreciate the predictive performance of the proposed MSS-GAN model we also trained Wei et. al's [5] model on the selected 4 players. When comparing the prediction results presented in Tab. 4 with Tab. 6 it is clear that the reduced training data has a significant impact on the performance of Wei et. al's [5] approach. In contrast we do not observe significant deviation in the proposed model's performance, indicating its ability to infer different player styles even without very large volumes of data.

5.7 Implementation Details

The implementation of the MSS-GAN module presented in this paper is completed using Keras [47] with the Theano [48] backend. We choose batch size to be 32 and trained the model using the Adam optimiser [49] with a learning rate of 0.002 for 10 epochs and set the learning rate to be 0.0002 for another 20 epochs. Hyper parameters l, N and b are evaluated experimentally. Using the validation set of Sec. 5.3 we fine tuned each parameter individually, holding the rest of the parameters constant. The experimental evaluations are illustrated in Fig. 8. As $N = 1100$, $l = 3$ and $b = 80$ gives us the minimum error values, we set the respective parameters accordingly. Fig. 9 shows learning curves of the proposed MSS-GAN model showing the model convergence. We note that the model doesn't overfit.

5.8 Time Efficiency

The proposed MSS-GAN model doesn't require any special hardware such as GPUs to run and has 33.7M trainable parameters. We ran the test set of Sec. 5.3 on a single core of an Intel Xeon E5-2680 2.50GHz CPU and the algorithm was able to generate 1000 shot location predictions 28.5 seconds.

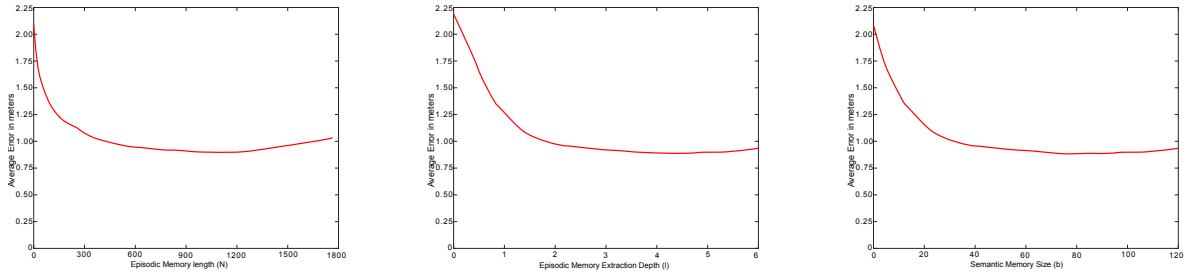
5.9 Application: Tactics analysis

The conditional nature of the proposed model allows us to directly infer the opponent adaptation strategies against different players. In Fig 10 we visualise the predicted return shots for the same incoming shot, and the same context for Djokovic, Nadal and Federer against different opponents. We held the shot location, player and opponent location, incident speed (s_t) and angle (a_t) and the points (p_t) constant and changed only the opponent id (op_t) in the system. This allows us to infer different opponent adaptation strategies employed, and we can explore the way that a given will player will subtlety vary their play style depending on the opponent. From Fig. 10 we can see variations in the depth and angle of return shots, as players vary the tactics in response to their opponent.

To further demonstrate the importance and value of the proposed context modelling scheme, we investigate how players change their strategies depending on the current

TABLE 6
Shot Location Prediction Results for Randomly Selected 4 Players from fourth round: We measured the distance between the predicted and ground truth shot locations in meters.

Method	J-W Tsonga	Richard Gasquet	K Nishikori	Andy Murray	Average
Wei et. al [5]	2.13	2.60	3.31	2.90	2.74
MSS-GAN	**0.97**	**0.91**	**1.07**	1.03	**1.00**



(a) (b) (c)

Fig. 8. Hyper parameters evaluation process. We evaluate Episodic Memory length, N, Episodic Memory extraction depth, l and Semantic Memory size, b, experimentally holding the rest of hyper parameters constant. As $N = 1100$, $l = 3$ and $b = 80$ gives us the optimal results we set the respective sizes accordingly.

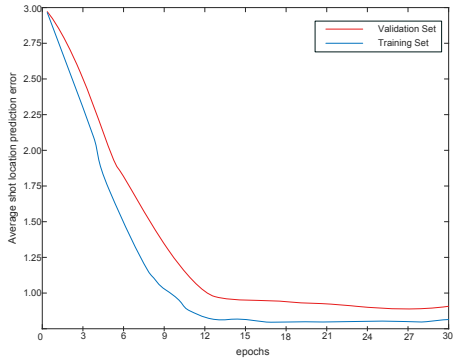
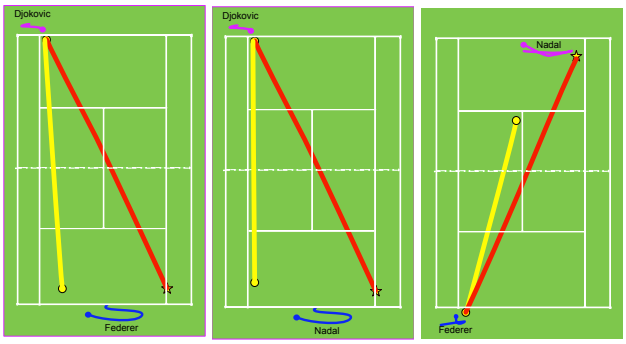
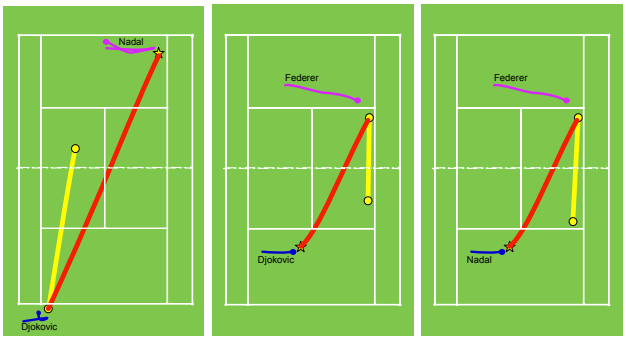


Fig. 9. Learning curves for training and validation sets for shot location prediction Error



(a) Djokovic to Federer (b) Djokovic to Nadal (c) Nadal to Federer



(d) Nadal to Djokovic (e) Federer to Djokovic (f) Federer to Nadal

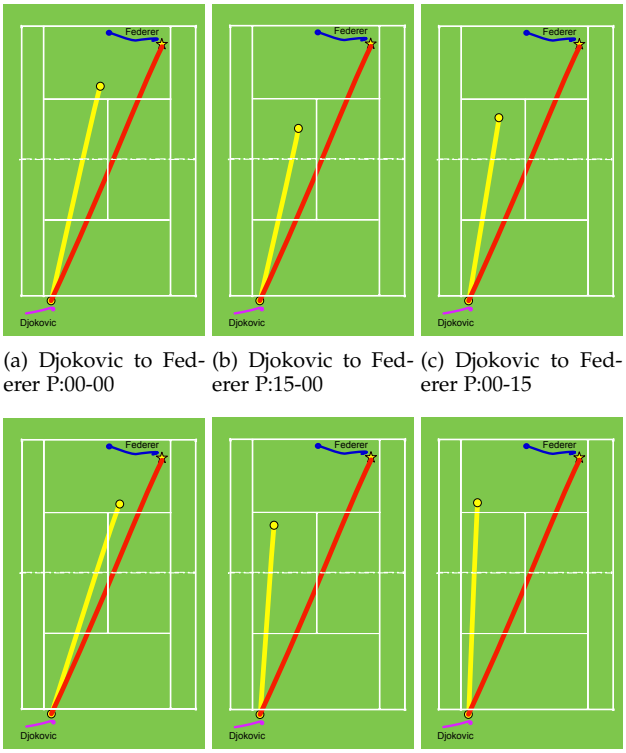
Fig. 10. Given the same incoming shot, opponent and player locations, speed (s_t), angle (a_t) and points (p_t), we can change the opponent id (op_t) and see how the player of interest changes his strategy to adapt to the opponent. Incoming shot trajectory is denoted in red where the yellow start and circle defines the starting and ending locations. The predicted return shot trajectory is denoted in yellow line where the ending location is represented in a yellow circle. Observed feet movements for the player of interest and opponent are denoted in magenta and blue colours.

score. In Fig. 11 we visualise the shot location predictions where we held the shot location, player and opponent location, incident speed (s_t) and angle (a_t) and the opponent id (op_t) constant and changed only the points (p_t) in the system. We see clear differences in the return shot as the score changes, suggesting the has learned when to be aggressive within not only a point, but in the wider context of the match.

These examples demonstrate that the proposed MSS-GAN model is capable of capturing match context and the player tactical elements which are essential when anticipating player behaviour.

6 Conclusion

In this paper we propose a method to anticipate the next shot type and location in tennis, by analysing the structure of the player behaviour's and it's temporal accordance. We contribute a novel data driven method to capture salient information from the observed game context and propose methodologies to capture longterm historical experiences of different players, emulating the episodic memory behaviour



(a) Djokovic to Federer P:00-00 (b) Djokovic to Federer P:15-00 (c) Djokovic to Federer P:00-15

(d) Djokovic to Federer P:30-00 (e) Djokovic to Federer P:30-15 (f) Djokovic to Federer P:15-30

Fig. 11. Given the same incoming shot, opponent and player locations, speed (s_t), angle (a_t) and opponent id (op_t), we can change the points (p_t) and see how the player of interest changes his strategy to adapt to the current context. Incoming shot trajectory is denoted in red where the yellow start and circle defines the starting and ending locations. The predicted return shot trajectory is denoted in yellow line where the ending location is represented in a yellow circle. Observed feet movements for the player of interest and opponent are denoted in magenta and blue colours.

of the human brain. Additionally, we introduce a novel methodology for learning abstract level concepts through a tree structured episodic memory and propose methods for transferring this acquired knowledge to a neural semantic memory component. Our quantitative and qualitative evaluations on a tennis player tracking dataset from the 2012 Australian Men's Open demonstrate the capacity of the proposed method to anticipate complex real world player strategies, and its potential to be applied to other data mining/ knowledge discovery applications.

Acknowledgments

The authors would like to thank Tennis Australia for providing access to the Hawk-eye player tracking database for this analysis. The authors also thank QUT High Performance Computing (HPC) for providing the computational resources for this research.

References

[1] S. Cacioppo, F. Fontang, N. Patel, J. Decety, G. Monteleone, and J. T. Cacioppo, "Intention understanding over t: a neuroimaging study on shared representations and tennis return predictions," *Frontiers in human neuroscience*, vol. 8, p. 781, 2014.

[2] L. Crognier and Y.-A. Féry, "Effect of tactical initiative on predicting passing shots in tennis," *Applied Cognitive Psychology*, vol. 19, no. 5, pp. 637–649, 2005.

[3] A. M. Williams, P. Ward, N. J. Smeeton, and D. Allen, "Developing anticipation skills in tennis using on-court instruction: Perception versus perception and action," *Journal of Applied Sport Psychology*, vol. 16, no. 4, pp. 350–360, 2004.

[4] M. J. Wright and R. C. Jackson, "Brain regions concerned with perceptual skills in tennis: An fmri study," *International Journal of Psychophysiology*, vol. 63, no. 2, pp. 214–220, 2007.

[5] X. Wei, P. Lucey, S. Morgan, and S. Sridharan, "Forecasting the next shot location in tennis using fine-grained spatiotemporal tracking data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 11, pp. 2988–2997, 2016.

[6] D. I. Shvorin, "Understanding the relationship between performance characteristics, shot selection and decision-making in the game of tennis," Ph.D. dissertation, Clemson University, 2017.

[7] A. M. Williams, P. Ward, J. M. Knowles, and N. J. Smeeton, "Anticipation skill in a real-world task: measurement, training, and transfer in tennis." *Journal of Experimental Psychology: Applied*, vol. 8, no. 4, p. 259, 2002.

[8] J. A. Etzel, N. Valchev, V. Gazzola, and C. Keysers, "Is brain activity during action observation modulated by the perceived fairness of the actor?" *PLoS One*, vol. 11, no. 1, p. e0145350, 2016.

[9] A. Horzyk, J. A. Starzyk, and J. Graham, "Integration of semantic and episodic memories," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 12, pp. 3084–3095, 2017.

[10] W. Wang, A.-H. Tan, and L.-N. Teow, "Semantic memory modeling and memory interaction in learning agents," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 11, pp. 2882–2895, 2017.

[11] E. Denton, S. Gross, and R. Fergus, "Semi-supervised learning with context-conditional generative adversarial networks," *Data, Learning and Inference Workshop*, 2017.

[12] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *Advances in Neural Information Processing Systems*, 2016, pp. 4565–4573.

[13] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *arXiv preprint*, 2017.

[14] J. Pan, C. Canton, K. McGuinness, N. E. O'Connor, J. Torres, E. Sayrol, and X. Giro-i Nieto, "Salgan: Visual saliency prediction with generative adversarial networks," *arXiv preprint arXiv:1701.01081*, 2017.

[15] T. Fernando, S. Sridharan, C. Fookes, and S. Denman, "Deep decision trees for discriminative dictionary learning with adversarial multi-agent trajectories," *CVPR Workshop on Computer Vision in Sports*, 2018.

[16] "https://www.hawkeyeinnovations.com/sports/tennis."

[17] T. Fernando, S. Denman, S. Sridharan, and C. Fookes, "Task specific visual saliency prediction with memory augmented conditional generative adversarial networks," *Applications of Computer Vision (WACV), 2018 IEEE Winter Conference on*, 2018.

[18] T. Arici and A. Celikyilmaz, "Associative adversarial networks," *arXiv preprint arXiv:1611.06953*, 2016.

[19] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[20] D. Yoo, N. Kim, S. Park, A. S. Paek, and I. S. Kweon, "Pixel-level domain transfer," in *European Conference on Computer Vision*. Springer, 2016, pp. 517–532.

[21] J. Zhao, M. Mathieu, and Y. LeCun, "Energy-based generative adversarial network," *arXiv preprint arXiv:1609.03126*, 2016.

[22] A. Kumar, O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, V. Zhong, R. Paulus, and R. Socher, "Ask me anything: Dynamic memory networks for natural language processing," in *International Conference on Machine Learning*, 2016, pp. 1378–1387.

[23] E. Parisotto and R. Salakhutdinov, "Neural map: Structured memory for deep reinforcement learning," *arXiv preprint arXiv:1702.08360*, 2017.

[24] J. Whitmore, *Coaching for performance: Growing human potential and purpose-the principles and practice of coaching and leadership*. Nicholas Brealey, 2010.

[25] M. Reid, R. Duffield, B. Dawson, J. Baker, and M. Crespo, "Quantification of the physiological and performance characteristics of

on-court tennis drills," *British Journal of Sports Medicine*, vol. 42, no. 2, pp. 146–151, 2008.
[26] J. Wang, C. Xu, E. Chng, K. Wah, and Q. Tian, "Automatic replay generation for soccer video broadcasting," in *Proceedings of the 12th annual ACM international conference on Multimedia*. ACM, 2004, pp. 32–39.
[27] P. Carr, M. Mistry, and I. Matthews, "Hybrid robotic/virtual pan-tilt-zom cameras for autonomous event recording," in *Proceedings of the 21st ACM international conference on Multimedia*. ACM, 2013, pp. 193–202.
[28] D. Cervone, A. DAmour, L. Bornn, and K. Goldsberry, "Pointwise: Predicting points and valuing decisions in real time with nba optical tracking data," in *Proceedings of the 8th MIT Sloan Sports Analytics Conference, Boston, MA, USA*, vol. 28, 2014, p. 3.
[29] X. Wei, P. Lucey, S. Morgan, and S. Sridharan, "Predicting shot locations in tennis using spatiotemporal data," in *Digital Image Computing: Techniques and Applications (DICTA), 2013 International Conference on*. IEEE, 2013, pp. 1–8.
[30] M. Malinowski and M. Fritz, "A multi-world approach to question answering about real-world scenes based on uncertain input," in *Advances in neural information processing systems*, 2014, pp. 1682–1690.
[31] T. Fernando, S. Denman, S. Sridharan, and C. Fookes, "Learning temporal strategic relationships using generative adversarial imitation learning." *International Foundation for Autonomous Agents and Multiagent Systems*, 2018.
[32] J. E. Laird and S. Mohan, "A case study of knowledge integration across multiple memories in soar," *Biologically Inspired Cognitive Architectures*, vol. 8, pp. 93–99, 2014.
[33] S. T. Mueller and R. M. Shiffrin, "Rem ii: A model of the developmental co-evolution of episodic memory and semantic knowledge," in *International conference on learning and development (ICDL), Bloomington, IN*. Citeseer, 2006.
[34] G. J. Rinkus, "A neural model of episodic and semantic spatiotemporal memory," in *Proceedings of the Annual Meeting of the Cognitive Science Society*, vol. 26, no. 26, 2004.
[35] T. Fernando, S. Denman, A. McFadyen, S. Sridharan, and C. Fookes, "Tree memory networks for modelling long-term temporal dependencies," *arXiv preprint arXiv:1703.04706*, 2017.
[36] L. Yu, W. Zhang, J. Wang, and Y. Yu, "Seqgan: Sequence generative adversarial nets with policy gradient." in *AAAI*, 2017, pp. 2852–2858.
[37] E. Tulving and D. Murray, "Elements of episodic memory," *Canadian Psychology*, vol. 26, no. 3, pp. 235–238, 1985.
[38] R. Rinkus and J. Leveille, "Superposed episodic and semantic memory via sparse distributed representation," *arXiv preprint arXiv:1710.07829*, 2017.
[39] Y. Duan, M. Andrychowicz, B. Stadie, J. Ho, J. Schneider, I. Sutskever, P. Abbeel, and W. Zaremba, "One-shot imitation learning," *arXiv preprint arXiv:1703.07326*, 2017.
[40] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International Conference on Machine Learning*, 2015, pp. 2048–2057.
[41] L. Draschkowitz, C. Draschkowitz, and H. Hlavacs, "Predicting shot success for table tennis using video analysis and machine learning," in *International Conference on Intelligent Technologies for Interactive Entertainment*. Springer, 2014, pp. 12–21.
[42] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[43] K. Duan, S. S. Keerthi, W. Chu, S. K. Shevade, and A. N. Poo, "Multi-category classification by soft-max combination of binary classifiers," in *International Workshop on Multiple Classifier Systems*. Springer, 2003, pp. 125–134.
[44] A. Kumar, P. S. Chavan, V. Sharatchandra, S. David, P. Kelly, and N. E. O'Connor, "3d estimation and visualization of motion in a multicamera network for sports," in *Machine Vision and Image Processing Conference (IMVIP), 2011 Irish*. IEEE, 2011, pp. 15–19.
[45] A. Jansson, "Predicting trajectories of golf balls using recurrent neural networks," Master thesis in Computer Science, 2017.
[46] A. Odena, "Semi-supervised learning with generative adversarial networks," *arXiv preprint arXiv:1606.01583*, 2016.
[47] F. Chollet, "Keras," *URL http://keras. io, 2017*, 2017.
[48] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, "Theano: A cpu and gpu math compiler in python," in *Proceedings of 9th Python in Science Conference*, 2010, pp. 1–7.
[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

Tharindu Fernando is a PhD student at Queensland University of Technology, Australia. He received his Bachelor of Computer Science with first class honours from University of Peradeniya, Sri Lanka, in 2015. Prior to the beginning of his PhD program, he has conducted variety of research projects, resulting in automated systems to evaluate player biomechanics and perform strategic analysis in sports. His research interests focus mainly onto human behaviour analysis and prediction.

Dr. Simon Denman received a BEng (Electrical), BIT, and PhD in the area of object tracking from the Queensland University of Technology (QUT) in Brisbane, Australia. He is currently a Senior Research Fellow with the Speech, Audio, Image and Video Technology Laboratory at QUT. His active areas of research include intelligent surveillance, video analytics, and video-based recognition.

Professor Sridha Sridharan has a BSc (Electrical Engineering) degree and obtained a MSc (Communication Engineering) degree from the University of Manchester, UK and a PhD degree from University of New South Wales, Australia. He is currently with the Queensland University of Technology (QUT) where he is a Professor in the School Electrical Engineering and Computer Science. Professor Sridharan is the Leader of the Research Program in Speech, Audio, Image and Video Technologies (SAIVT) at QUT, with strong focus in the areas of computer vision, pattern recognition and machine learning. He has published over 500 papers consisting of publications in journals and in refereed international conferences in the areas of Image and Speech technologies during the period 1990-2016. During this period he has also graduated 60 PhD students in the areas of Image and Speech technologies. Prof Sridharan has also received a number of research grants from various funding bodies including Commonwealth competitive funding schemes such as the Australian Research Council (ARC) and the National Security Science and Technology (NSST) unit. Several of his research outcomes have been commercialised.

Clinton Fookes is a Professor in Vision Signal Processing and the Speech, Audio, Image and Video Technologies group within the Science and Engineering Faculty at QUT. He holds a BEng (Aerospace/Avionics), an MBA with a focus on technology innovation/management, and a PhD in the field of computer vision. Clinton actively researches in the fields of computer vision and pattern recognition including video surveillance, biometrics, human- computer interaction, airport security and operations, command and control, and complex systems. Clinton has attracted over $\$15M$ of cash funding for fundamental and applied research from external competitive sources and has published over 140 internationally peer- reviewed articles. He has been the Director of Research for the School of Electrical Engineering and Computer Science. He is currently the Head of Discipline for Vision Signal Processing. He is the Technical Director for the Airports of the Future collaborative research initiatives. He is a Senior Member of the IEEE, and a member of other professional organisations including the APRS.