

Thinking the GOAT: Imitating Tennis Styles

Padmanaba Srinivasan^{α,ξ,*}, Raghavan Subramanian[‡], William Knottenbelt^α

^α Imperial College London, [‡] Infosys, ^ξ Correspondence to: ps3416@ic.ac.uk

* Work done while an InStep intern at Infosys

Abstract

A tactically aware coach is key to improving tennis players' games; a coach analyses past matches with two considerations in mind: 1) the **style** of the player and how that style translates to real-world shot-making, and 2) the **intent** of a shot, irrespective of the outcome. Modern Hawk-Eye technology deployed in top-tier tournaments has enabled deeper analysis of professional matches than ever before. The aim of this paper is to emulate and augment the qualities of great coaches using data collected by Hawk-Eye; we develop a deep learning approach to imitate tennis players' responses, to learn individual player styles efficiently, and we demonstrate this using performance metrics and illustrations.

1. Introduction

Tennis is a game of strategy where players outmanoeuvre their opponents to either force them into unfavourable situations where an error will be induced or set the rally up for themselves to hit a winner. Tennis points will (usually) end in one of the following ways: a winner, a forced error, or an unforced error. In the case of a winner, a player hits a shot that outright ends the rally in their favour, a forced error implies that the opponent makes an error due to the players strategy, and when making an unforced error, the opponent incorrectly executes an otherwise routine stroke. Unforced errors are the third and least studied of the end types. In this paper, we develop a method to imitate player styles combining advances in generative modelling, imitation learning, and language modelling with the following goals:

- Compare how two different players of interest (POIs) respond to the same incoming shot against the same opponent
- Explore how a POI's response to a shot will change when facing different opponents
- Analyse unforced errors with respect to player style and the current rally state to identify a POI's true intent in a data-driven manner.

In one essay, coach and analyst, Craig O'Shannessy argues for increased focus on *unforced* errors¹. Looking beyond highlight reel-darling winners (Wei, Lucey et al. 2013, Wei, Lucey et al. 2014), unforced errors should be inspected with a view to identify whether the **intent** behind the shot was correct. An unforced error typically bounces outside the court lines or lands in the net, which makes identifying alternative potential shots challenging. Our approach produces distributions over the parameters of shots to uncover the intention of the player. Furthermore, we respect and account for the individual styles of players and incorporate this into our algorithm.

¹ <https://www.nytimes.com/2018/06/08/sports/unforced-errors-tennis.html>

Predicting the probability of winning a point forms a classic analysis in tennis (Kemeny, Snell et al. 1960), and the notion of finding style descriptors (Wei, Lucey et al. 2016) to aid in this is an intriguing idea. Modern deep learning has been sparsely applied to analysis in tennis (Sipko and Knottenbelt 2015, Mora and Knottenbelt 2017, Cai and Tang 2018, Fernando, Denman et al. 2019, Candila and Palazzo 2020), but few works combine statistical analysis with player adaptation, and those that do (Wei, Lucey et al. 2013, Fernando, Denman et al. 2019), train N separate models for N unique players.

A popular way of classifying players is by their style: “serve-and-volleyer”, “aggressive-baselinier”, and “counterpuncher” are just a few of the terms used to describe players. These terms are often based on a *predominant* style that a player tends follow but it ignores how players adapt their own patterns when facing different opponents. Yet such descriptors are not without merit; they are useful for expressing style at a high-level and serve as priors for how a player is expected to react in-match. However, the nuance of *style* means we need a far more expressive method of describing players. In our paper, we represent player styles as latent embeddings that are learned jointly with the predictive model. Our method of learning styles is efficient, extensible and merits independent evaluation. We provide a visual overview of predictions generated by our model in Figure 1.

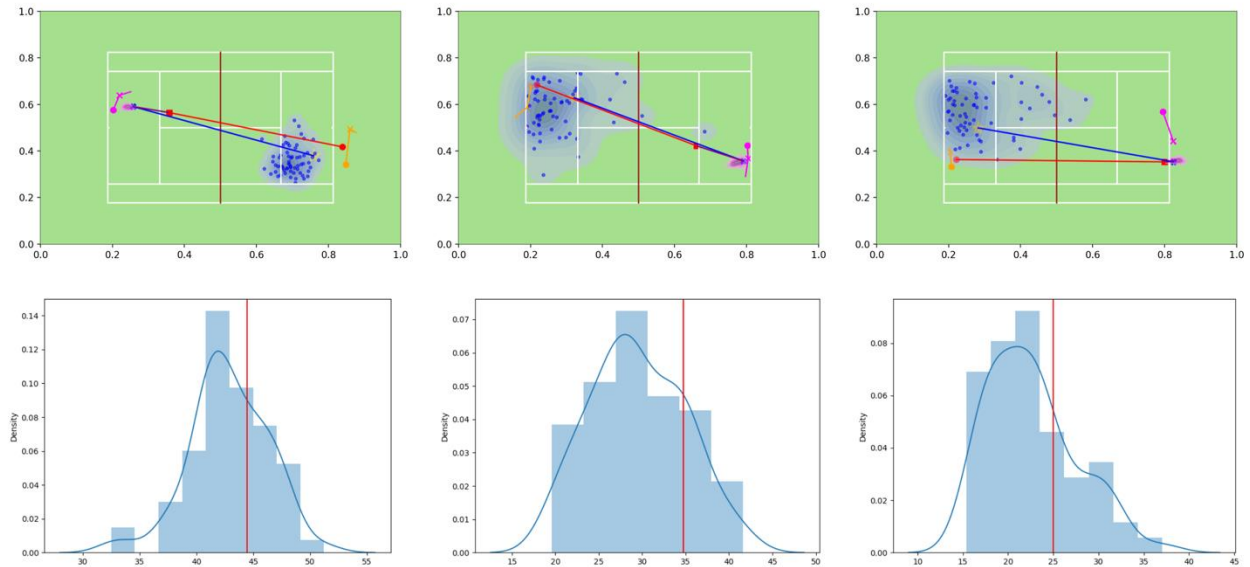


Figure 1: Samples of responses drawn from our predictive model conditioned on an incoming shot. Top: circles indicate the start of a motion path. The red line indicates the path of the incoming ball, the red square the bounce location and the maroon line the path of the ball after the bounce, which is included for clarity and is not provided to the model. The blue density indicates a KDE estimate over possible bounce locations of the POIs outgoing shot and the purple density indicates the distribution of the speed of predicted outgoing shots in blue, with the ground truth indicated by a red line.

2. Imitating Players and Learning Styles

Our approach hinges on effective data engineering to extract useful information from tracking data, as well as combining design elements from generative modelling and language modelling to yield a powerful framework for performing imitation learning in tennis. In the following subsections, we describe the components that comprise our model in turn.

2.1. Data

With improvements in data collection technology, namely Hawk-Eye (Owens, Harris et al. 2003), the improved quality and granularity of data has enabled increasingly nuanced knowledge discovery (Wei, Lucey et al. 2013, Wei, Lucey et al. 2015, Wei, Lucey et al. 2016, Kovalchik, Ingram et al. 2020). We use Hawk-Eye data from 2020 where each named player has hit at least 500 shots. We use data from the 2020 French Open tournament.

Player	# Shots	Winners	Errors
Nadal	1836	394	213
Djokovic	1453	341	144
Tsitsipas	1411	330	182
Schwartzman	1310	264	110
Thiem	1292	201	137

Table 1: Total shots hit by players used for evaluation.

Progression in professional tennis tournaments (with some exceptions) is determined by match outcome – only the player that wins a match proceeds to the next round – which results in the players that proceed to further rounds having more data samples. We select the five players with the most samples to focus our analysis on and summarize their details in Table 1. We hold 30% of the data of these five players for evaluation, ensuring evaluation over at least 300 data points per player. The 300 evaluation shots are formed from the final rallies of each match the player has competed in to ensure the temporal ordering is maintained.

The Hawk-Eye system captures the position of the tennis ball in 3D-space as a set of coordinates and the position of the players as a function of time, \mathbf{t} . Metrics derived from displacement, including velocities and accelerations can be computed and additional information such as the shot type, score, shot count and identify codes for players is also included.

The shots comprising every rally are sorted into incoming-outgoing shot pairs, $\mathbf{X} = \{\mathbf{x}^i, \mathbf{x}^o\}_n \forall n \in [1 \dots N]$, forming a dataset of N such pairs. This excludes ‘unprompted’ shots, such as the serves hit by a POI. Raw data is subsampled to extract information at the time of impact, the time at which the ball crosses the net and at the end of the shot (either the bounce, or the opponent’s hit time if the opponent hits a volley). Using ball and player positions, we can derive velocities and angles. For the players, we compute additional metrics including the distance travelled by each

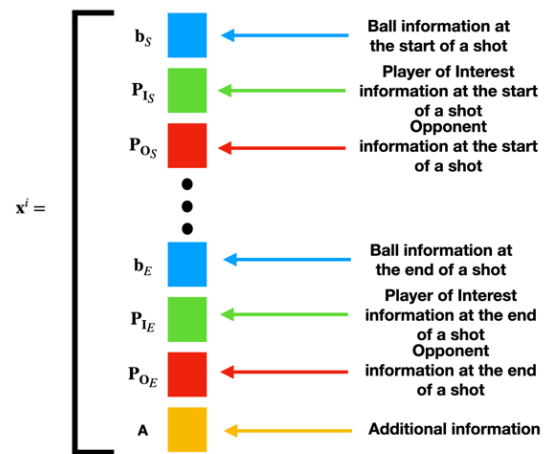


Figure 2: Illustration of an incoming shot feature vector.

player in the current rally in the lateral and medial directions, the score and the type of shot hit, which is a discrete variable that can take one of eight values:

1. Topspin/flat forehand
2. Forehand volley
3. Forehand slice
4. Topspin/flat backhand
5. Backhand volley
6. Backhand slice
7. Overhead (smash)
8. Serve.

This data is combined to form a state vector of the incoming shot \mathbf{x}^i and its pair \mathbf{x}^o consists of similar information that describes the key characteristics of the outgoing shot and excludes player information and other non-ball-related parameters. An illustration of the incoming feature vector is displayed in Figure 2.

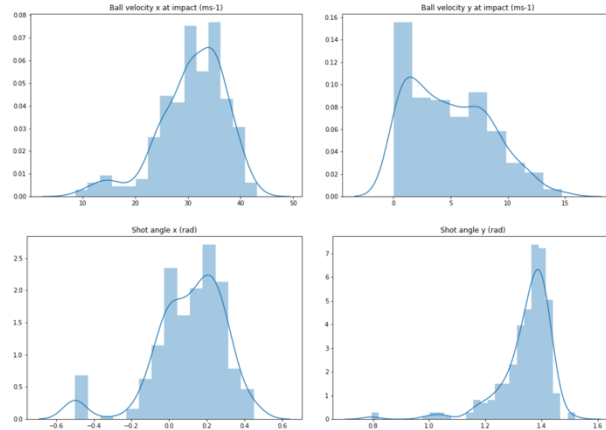


Figure 3: Marginal distributions for velocities and angles for Rafael Nadal. These densities are formed from responses to shots that bounce on the Ad-court side behind the service line, for which Nadal responds with a topspin backhand.

2.2. Introducing Imitation Learning

Imitation learning (IL) is widely used in robotics (Argall, 2009, Arulkumaran 2021) to learn control policies from demonstrations. A key underlying assumption of IL is the presence of (mostly) expert trajectories that can be used to train a deep learning model to behave like the expert.

The simplest technique employs a fully supervised approach to learning using Behavioural Cloning (BC) (Pomerleau 1988) which performs well for low-dimensional, simpler tasks. It has been shown that BC often fails to generalize well to higher-dimensional tasks over longer temporal spans as errors can accrue over time, leading to catastrophic failure.

More robust IL methods include distance-based approaches (Li et al. 2015) and adversarial methods (Ho et al. 2016, Fu et al. 2017, Ghasemipour et al. 2020, Kostrikov et al. 2018), the latter approach being adopted here.

In games such as (singles) tennis, where the state-space is high-dimensional and continuous, and where actions are subject to rules and limitations, manually describing detailed strategies can be challenging.

Describing positions, velocities and accelerations and using these to construct workable strategies is prohibitively time consuming as it requires the application of implicitly learned knowledge in combination with condensed knowledge. The scope of this challenge is illustrated in Figure 3 where we show the marginal distributions of velocities and angles for Rafael Nadal when hitting a forehand from the Ad-court side, when the ball bounces behind the service line. For a human to filter through and wield a large and diverse quantity of data is prohibitively time consuming and even then, they are likely to miss important contextual information. Assimilating a large quantity of

complex data and learning a hierarchical representation of it is a task well suited to machine learning algorithms. Teaching a computer to **imitate** from a dataset of observations can be a powerful tool for automatic pattern discovery, and training deep learning models on suitably descriptive data can save significant engineering effort while also generalizing well.

2.3. Learning Player Styles

Imitating player behaviour requires a deep understanding of player style, accounting for situational variations in style as well as how style changes against different opponents. (Wei, Lucey et al. 2016) present a similar argument, computing *style priors* for players using dictionary learning and proceed to compute the probability of a POI winning a point as a rally progresses. (Fernando, Denman et al. 2019) predict bounce locations deterministically for an outgoing shot, and adapt individual networks for each player, requiring N networks for N unique players. Our approach is novel, efficient, and expressive, adopting ideas from deep learning-based Language Modelling (LM).

Modern language modelling is dominated by transformers (Vaswani et al. 2017) which have seen extensive use across a range of language tasks. Learnable, latent word embeddings are closely associated with the performance demonstrated by transformers; with such embeddings, each word corresponds to an index in a list of embeddings. Each fixed-size embedding can be updated by a suitable learning algorithm and represent a high-dimensional quantitative representation of a word. We follow a similar approach: each player in our dataset is assigned an index that corresponds to a single embedding. The embedding can be updated when the model is trained on data samples containing the player. As a result, our approach is extensible and efficient as more players can be included easily and requires the training of one single neural network that models all players. We reinterpret the dataset to consist of the ‘policies’ of several different players, and by informing the model of which player and policy are currently in use, we perform player adaptation jointly for all players.

2.4. The Imitation Model

Markov chains are a classic method of modelling tennis (Kemeny, Snell et al. 1960, Klaassen and Magnus 2001). The Markov property assumes that points and rallies in tennis are independent and identically distributed and tennis can be modelled as a Markov Decision Process (MDP). Our dataset consists of trajectories of variable length $\xi = \{\tau_1, \tau_2, \dots, \tau_N\}$, $\tau_i = \{s_0, a_0, \dots, s_t, a_t\} \forall i \in N$ where each trajectory τ_i consists of sequences of state—action (used interchangeably with incoming-outgoing shot) pairs. Under the MDP model of tennis, we can train a model on individual state—action pairs under the Markov property and do not need to incorporate additional complexities borne by temporal considerations.

Behavioural Cloning (BC) (Pomerleau 1988) is a commonly used method of IL that uses a fully supervised objective – this makes BC a straightforward approach but can be brittle when modelling high-dimensional, complex distributions (see Figure 3) over long temporal spans. GAIL (Ho et al. 2016) proposes an adversarial approach that maximises entropy while matching occupancy measure and adopts a Generative Adversarial Model (GAN) (Goodfellow et al. 2020, Mirza and Osindero 2014) framework. Our generator, $\hat{\pi}$ learns a policy by mapping random noise \mathbf{z} to an action $\hat{\mathbf{a}}$. The discriminator, D , is trained on state—action pairs from the dataset and is tasked with differentiating between policies from the dataset, π^* and policies produced by $\hat{\pi}$. This is a minimax

game which reaches equilibrium when the Jensen—Shannon divergence between generated and real distributions is minimized.

We use neural networks with two hidden layers and 256 hidden units with the ReLU activation for the generator and discriminator along with the more stable LSGAN (Mao et al. 2017) objective in GAIL:

$$\min_D \frac{1}{2} \mathbb{E}_{\mathbf{x} \sim p_{data}(\mathbf{x})} [(D(\mathbf{x}) - 1)^2] + \frac{1}{2} \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})} [D(\hat{\pi}(\mathbf{z}))^2]$$

$$\min_{\hat{\pi}} \frac{1}{2} \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})} [(D(\hat{\pi}(\mathbf{z})) - 1)^2] - \lambda \mathcal{H}(\hat{\pi})$$

Our model offers the following properties:

- Matches the expert's occupancy measure
- Penalizes deviation from the expert policy
- Maximizes entropy
- Learns multi-modal distributions over actions.

The first three properties are offered by GAIL; the maximum entropy property of GAIL is especially desirable as it prevents overfitting and is a classic technique used in reinforcement learning (Williams and Peng 1991) to encourage exploration.

The fourth property is especially appealing for modelling tennis players as often, players in the same match state have made disparate choices. An illustrative example is when receiving a shot on the forehand side on the deuce court, a player often can make the choice between going down the line or cross court. A good imitation model must consider all possible outcomes.

2.5. Incorporating Historical Context

Recent approaches to language modelling have demonstrated the benefits to generation quality offered by the incorporation of additional external information from a (static) database (Borgeaud et al. 2021). We can extend this idea to our imitation model: it is trivial to consider a current incoming shot for a POI, search in the POI's past rallies for similar ones and include the past responses for additional historical context. Responses can be matched via a suitable measure, such as L2-distance and included in the input to $\hat{\pi}$. The minimum similarity for a sample to be included in the historical

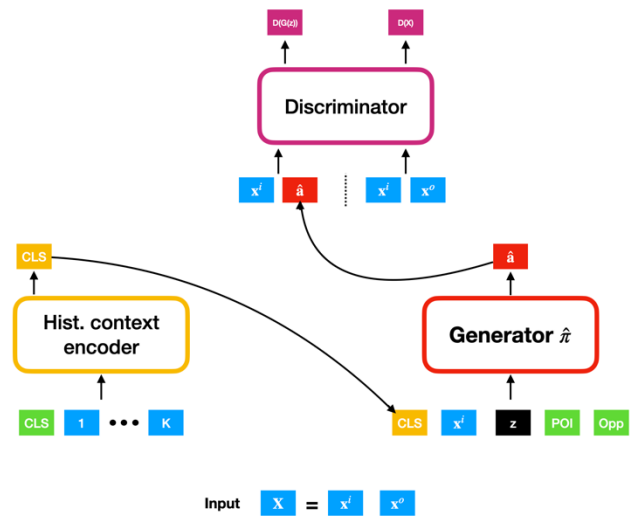


Figure 4: Architectural block diagram.

context is specified by a threshold and we select at most K such samples. This sequence of samples can be of variable length, and we encode this into a fixed-size historical context vector using a bidirectional transformer model as an encoder (Devlin et al. 2018).

2.6. Model Overview

We provide a visual overview of our modelling approach in Figure 4 which combines the techniques discussed in previous subsections. Identifying K past shots for historical context is performed during the data loading step and is passed as an additional input to the model.

3. Modelling Player Behaviour

With a description of our data and model complete, training can commence. The model training procedure is an extension of GAIL training procedure with changes to accommodate historical context and player embeddings. In the following subsections, we evaluate and exhibit the variety of insights that our model produces.

3.1. Variance from Ground Truth

Player	Mean Absolute Bounce Error(m)	Mean Absolute Speed Error (m/s)
Overall	0.873	1.985
Nadal	0.913	2.246
Djokovic	0.817	1.647
Thiem	0.798	1.660
Schwartzman	0.843	2.058
Tsitsipas	0.870	2.120

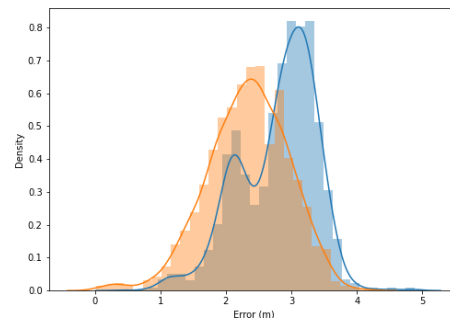


Figure 5: Bounce and speed deviation (left) and density plot of bounce deviation for non-error and error shots (right). Orange indicates deviation for non-error shots and blue indicates deviation for error shots.

We can quantify how much our model’s predictions for the next shot differ from that of each player for samples in the evaluation set. We compute the cartesian error for bounce location (Figure 5 left) and the RMSE for speed – both performance metrics are derived from combinations of true predictions and are more interpretable. Averaged over all players, we see a bounce location error of 0.85 metres. In context, we must consider the variation in shots hit from near-identical states as well as the maximum entropy objective; therefore, this is a high level of accuracy. For comparison (Wei, Lucey et al. 2013) report average bounce location errors of 1.89 metres and (Fernando, Denman et al. 2019) average errors of 0.93 metres.

The speed of the outgoing shot is subject to considerable variation – our model achieves an error of 2.1 metres per second or approximately 4.8 miles per hour. Note that our reported deviations include errors on **all** shots, including those shots that were errors. We can explore this in greater detail in the distributions of deviation for error and non-error ground truth shots in Figure 5.

3.2. Interpreting Player Style Embeddings

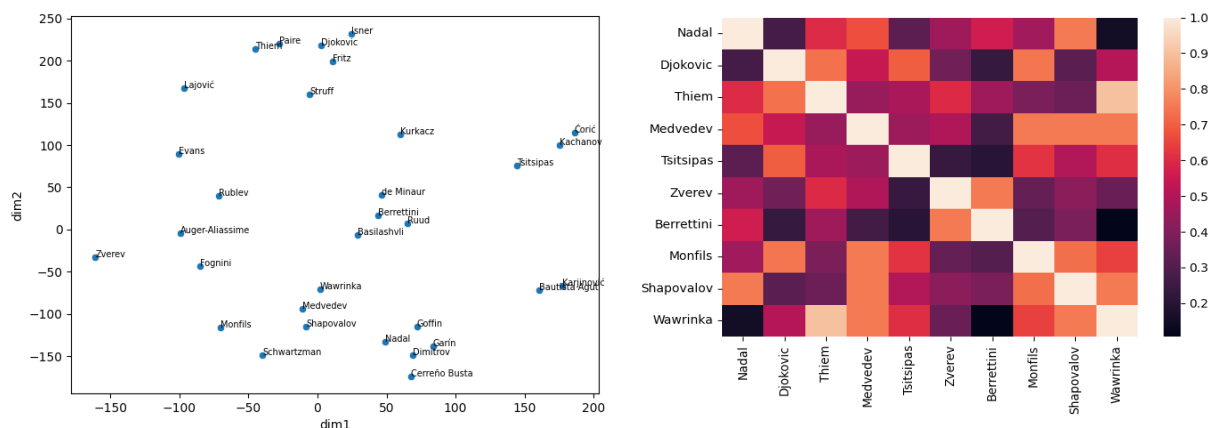


Figure 6: Left: t-SNE plot of player embeddings for all players in the dataset. Right: cosine similarity for the embeddings of 10 players with the most data.

Our model learns a latent representation of each player end-to-end with the predictive model during training. We compute the cosine similarity for the embeddings of 10 player (5 of whom form part of the evaluation set of players) and plot the heatmap in Figure 6. We see interesting patterns, such as, Thiem and Wawrinka having similar styles, and Thiem and Nadal playing with very dissimilar styles.

The resemblance in style between Wawrinka and Thiem has not gone unnoticed² with both players being credited with strong single-handed backhands and aggressive baseline play. In contrast, Wawrinka and Nadal have completely different styles, with Wawrinka preferring short baseline rallies whereas Nadal is more than content to outlast his opponent on court.

Currently, comparing similarity comes with a caveat: with the relatively small dataset size and number of players, we expect features in the player embeddings to be approximately orthogonal. Training with an increased number of samples and player count is likely to yield far more informative data-driven insights for reliable similarity analysis.

The t-SNE reduction of player embeddings for all players in Figure 6 looks to form distinct groupings of players. The t-SNE plot includes all players in the training dataset, including those players who are only ever present as opponents. The plot suggests that modelling a much larger number of players, perhaps including WTA players, could allow derivation of a much more nuanced and data-driven representation of style.

Continual learning can be used to update the model as more data becomes available. Inspecting player embeddings before and after updating can reveal *stylistic drift* – the change in player style over time – which can be analysed to show how a player’s style has evolved over their career relative to their peers.

² <https://lobandsmash.com/2018/09/17/the-uncanny-resemblance-between-dominic-thiem-and-stan-wawrinka/>

3.3. Comparing Responses for POIs

The generative model $\hat{\pi}$ maps random latent vectors from an isotropic Gaussian into a distribution conditioned on history, state, and player embeddings. Such a formulation can be manipulated to “swap” one POI’s embeddings for any other player’s embedding following which we can use MCMC to estimate distributions over posterior parameters while holding other factors constant. We plot distributions over bounce location only in Figure 7 to illustrate the effect of swapping POIs.

In the leftmost plots, we see key differences in shot-making between Dimitrov and Djokovic: Djokovic is far more likely to hit an aggressive backhand cross court, whereas Dimitrov’s style of play is far more conservative – this is reflected in dissimilarity of their embeddings.

In the central plots, we see this repeated with Djokovic likely his backhand to play a far more aggressive down-the-line shot than Dimitrov and in the rightmost, Tsitsipas and Schwartzman hit to similar locations as befits their offensive styles.

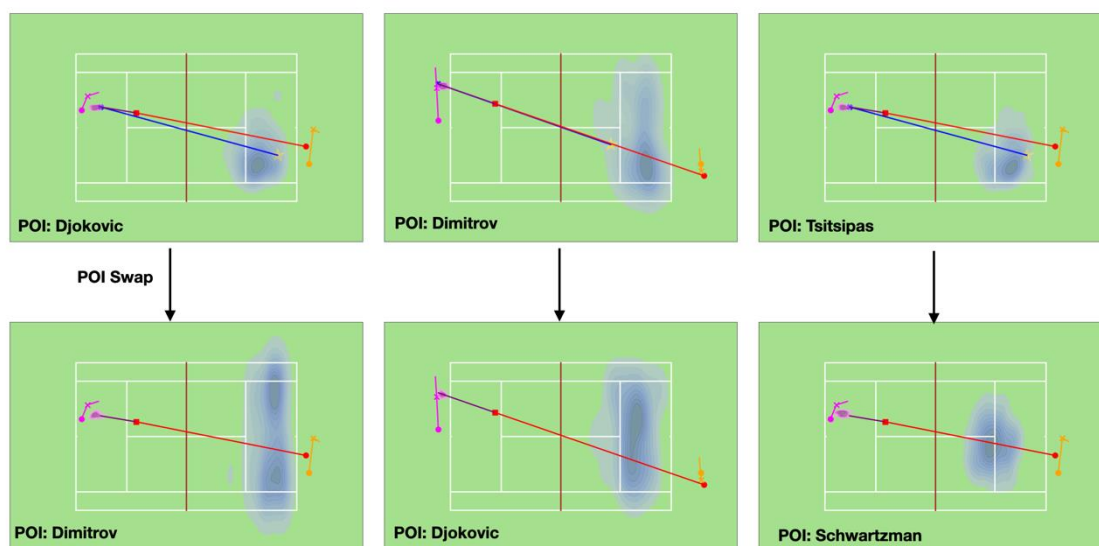


Figure 7: Changes in the outgoing shot when swapping POIs while holding opponent and state constant. Orange and magenta lines indicate opponent and POI movement, respectively. Circles indicate the start of a motion path. The red line indicates the path

3.4. Comparing Responses with Different Opponents

Following a similar procedure to POI swapping, we can also swap opponents to understand how a POI will respond differently when playing against different opponents. We show examples of opponent swapping in Figure 8. In the leftmost plot with Nadal as the POI, we see far more aggressive play against Djokovic compared to Dimitrov, where Nadal is far more likely to fall back to his well-known tactic of attacking the opponent’s backhand, especially the backhand of a right-handed player with a single-handed backhand. We see a similar pattern in the centre with the POI Dimitrov likely to hit to Schwartzman’s backhand than the taller Medvedev’s. In the rightmost plots, we see Tsitsipas is more likely to attack Djokovic’s backhand side, while actively seeking to avoid Nadal’s forehand.

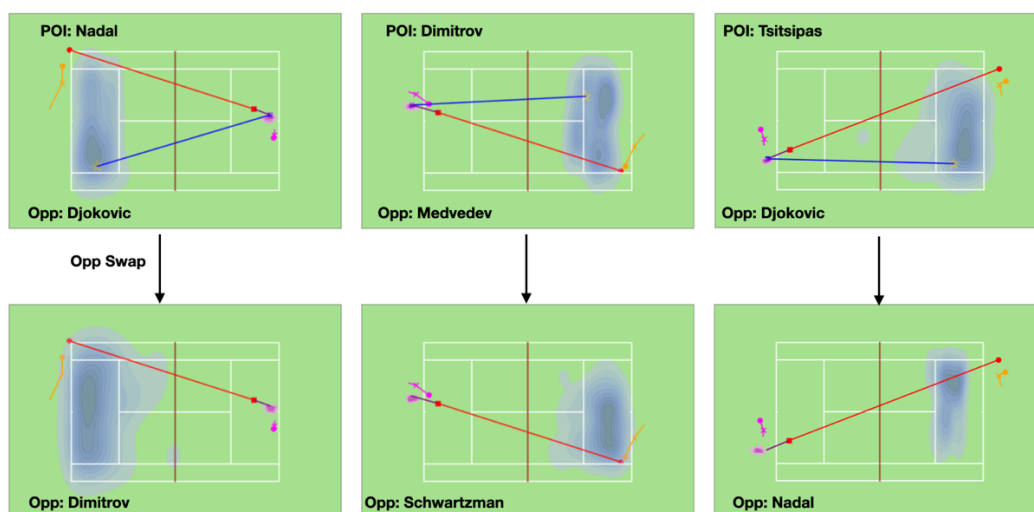


Figure 8: Examples of swapping opponent while holding the POI and other state constant.

3.5. Understanding Unforced Errors

For the ground truth shots that result in an unforced error, we can use MCMC sampling to produce distributions over outgoing shot parameters and compare the distributions against the observed ground truth. In Figure 9 we show top-view illustrations of ground truth bounce error location and the predicted bounce location density, as well as corresponding ground truth speed and distribution over speed for errors made by Djokovic.

In the presented plots, the predicted bounce location distribution is close to the ground truth and yields further insight into where the POI was aiming to hit to and by how much the error in execution changed the result. Inspecting the distributions of ball speed supplement this with information on the objective of the shot.

The leftmost plots suggest that the ball was hit faster than Djokovic usually does for this type of incoming shot, and likely was a contributing factor to the ball landing long. The positioning of the players implies that that this was a rally ball that Djokovic hits as a trademark down-the-line backhand, that in this instance landed long. The central plot sees Djokovic make the same mistake for a down-the-line forehand. In the rightmost plot, Djokovic hits the ball into the net, typically suggesting placement that is too low. We also see that Djokovic usually hits faster shots in similar scenarios, possibly contributing to the error.

The parameters of the outgoing shot can be displayed in more visually informative settings, such as in Infosys' Tennis Platform³. This can lead to a better understanding of the **intent** a POI had when making a shot in the context of the current state using their own style.

³ <https://www.infosys.com/atp/five-years-of-reinvention.html>

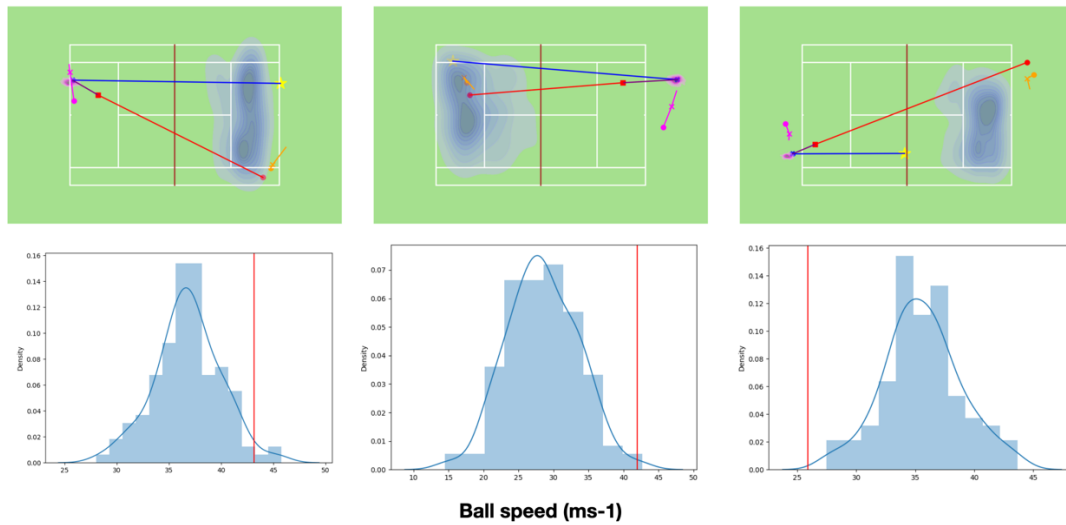


Figure 9: Understanding intent in unforced errors with ground truth path (blue) and bounce location (star) on top and below, the corresponding distribution of ball speed with the ground truth speed indicated by a red line. The POI is Djokovic for all samples.

3.6. Bucking the Trend: Deviation from Routine Behaviour

Our imitation model is a dictionary for the *typical* behaviour a POI exhibits. As a result, we can use our model to find when and how much a player deviates from their typical response. In Figure 10, we show examples of such situations when a player's true response lies far from their expected one. High density regions in the bounce location distributions suggest that the POI would typically place a shot there; however, their true response lies in an *unlikely pit*. Moreover, each of these shots is poorly placed with the bounce close to the opponent's location, or with the ball landing short. Suboptimal placement such as this can indicate turning points in a rally where the lack of aggressive player by a POI can allow an opponent to dominate the point and win. The converse is also true: an aggressive but unlikely shot hit by a POI in a typically defensive situation can equally turn the tide of the rally. The key insight here is that identification of suboptimal placement is crucial for post-match analysis. Our model can be used to find 'unlikely' shots and in combination with point outcome, can improve strategic analysis.

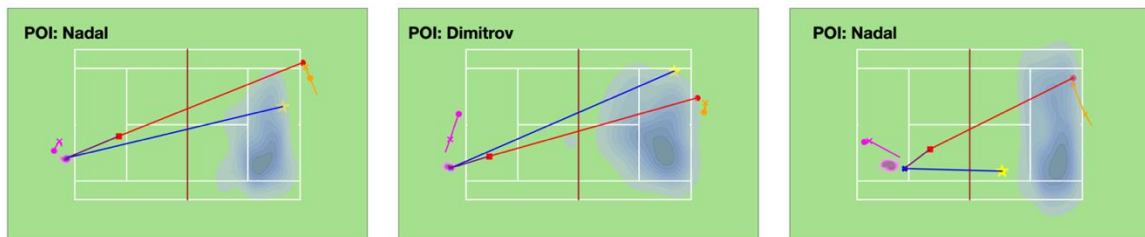


Figure 10: Examples of when a POI does not produce the expected response.

3.7. Varying Response with Score

The state vector provided to the policy includes the current score; the score can be modified to explore how players will vary play when **only** the score is changed, with all other factors held constant. We plot some response maps produced by such score changes in Figures 11-13. In Figure 11, we see Djokovic changing placement of the ball while holding speed mostly the same. In Figure 12 Djokovic's placement is roughly the same across all scores, but the speed of the outgoing ball increases at 40-0. Finally, in Figure 13, Djokovic's placement and speed are not very different at 40-40 and 0-40, but, at 0-0 speed seems faster overall and placement shifts,

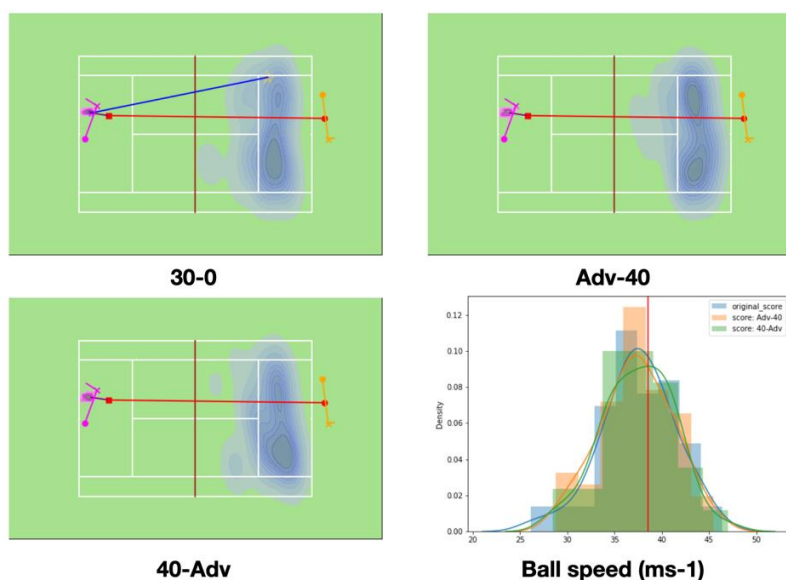


Figure 11: Response maps for Novak Djokovic. Top left: response map at original score. Bottom right: Distributions over outgoing ball speed with the ground truth speed indicated by a red line.

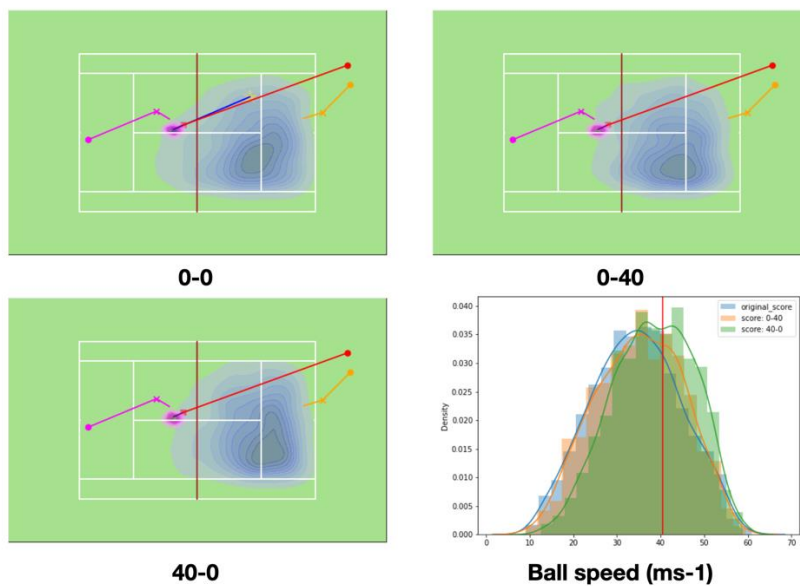


Figure 12: Response maps for Novak Djokovic. Top left: response map at original score. Bottom right: Distributions over outgoing ball speed with the ground truth speed indicated by a red line.

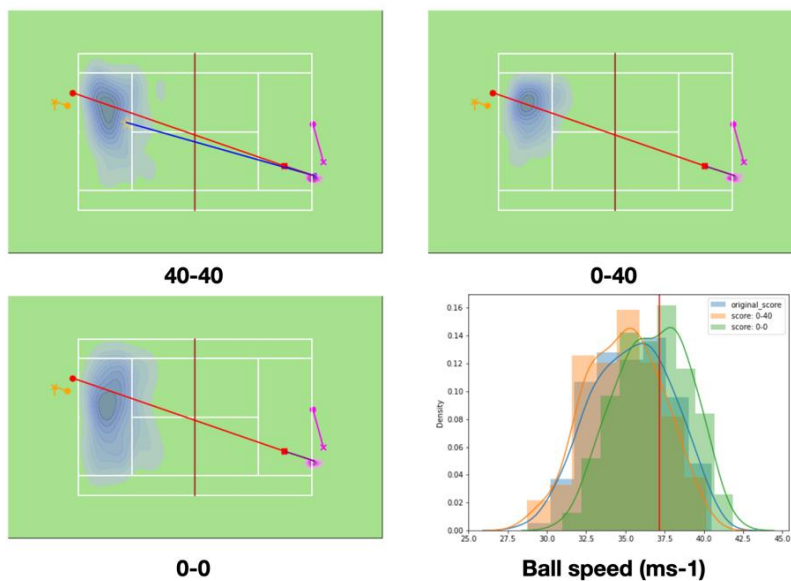


Figure 13: Response maps for Novak Djokovic. Top left: response map at original score. Bottom right: Distributions over outgoing ball speed with the ground truth speed indicated by a red line.

3.8. Ablation Experiments

We perform ablation experiments, removing components of our model and seeing their effects on performance. Our experiments remove the historical context (G-H) and the player (both POI and

opponent) embeddings (G-E). We report the cartesian bounce error in metres in Table 2. Removing the historical context has a minimal negative effect whereas the player encoding has a far more significant effect on performance. This demonstrates the impact and importance of capturing individual player styles in tennis and we have demonstrated the wealth of insights possible due to this.

Experiment	Overall	Nadal	Djokovic	Thiem	Schwartzman	Tsitsipas
G-H	+0.017	+0.007	+0.003	+0.012	-0.053	+0.048
G-E	+0.383	+0.136	+0.318	+0.087	+0.255	+0.487

Table 2: Results of ablation experiments showing relative mean absolute bounce error in metres compared to original results. Lower deviations are better, green indicates better performance than original model and red indicates inferior performance to the original model.

3.9. Just Some Fun: Hypothetical Scenarios

An indispensable aspect of being a fan of tennis is debating the *hypothetical* scenarios of one player against another in specific, controlled situations. Our imitation model can help with this: the insights demonstrated so far can be applied to appease the fans. How would Djokovic play against himself? How would Nadal play against himself in the same scenario? We answer these questions in Figure 14 with hypothetical scenarios of a player playing themselves, displaying responses to the same incoming shot. Figure 15 exhibits another scenario with Thiem and Tsitsipas.

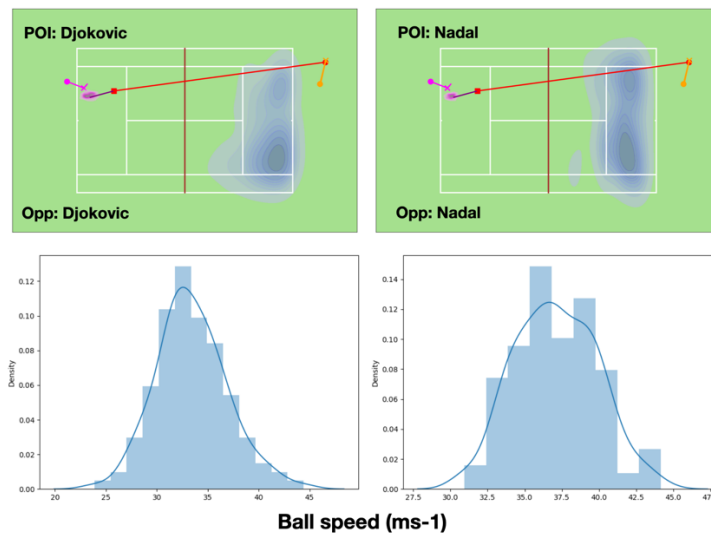


Figure 14: Modelling hypothetical scenarios with shot bounce and speed distributions. All scenarios are modelled under the same state with the only differences being the POI and opponent. Note the slightly differences in bounce position distribution on the upper illustrations and how Nadal is more likely to hit faster responses than Djokovic.

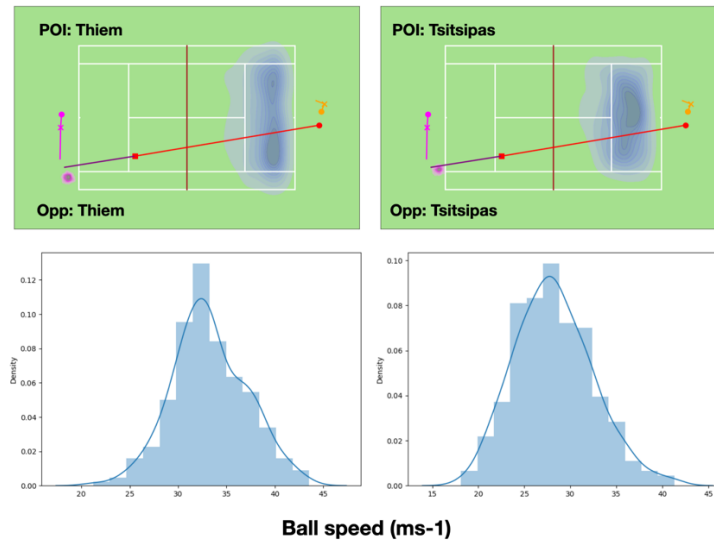


Figure 15: Modelling hypothetical scenarios with shot bounce and speed distributions. All scenarios are modelled under the same state with the only differences being the POI and opponent. Thiem is more likely to hit out wider and faster than Tsitsipas.

4. Conclusion

The wealth of information provided by high-granularity tracking data now justifies applying increasingly sophisticated modelling techniques for insights. We have applied modern deep learning techniques influenced by language modelling to the task of player imitation and have shown that this yields scalable, data-driven insights to learn player style efficiently and capture contextual behaviour in real-world rallies. In this paper, we have demonstrated the variety and quality of insights possible with such an approach; an approach that is likely extensible to many other sports with suitable data such as table tennis and badminton.

Future work could consider extending imitation to perform N -step rollouts, effectively simulating a rally exchange between two players by sampling an action for one player and then the other repeatedly alternating between the two. In practice, this would require a prompt – a valid service to begin the rally – in addition to a method of estimating information that the model does not predict (player locations, distances etc.) for sampled shots and a set of rules to ensure termination when an error or winner condition is met. Implementation of such a simulation environment could enhance unforced error analysis to explore what could happen if the execution of the shot had been more ideal.

Acknowledgements

The authors would like to thank *Fédération Française de Tennis* for their support and for providing the tracking data used in this research.

References

- Cai, J. and X. Tang (2018) "RGB Video Based Tennis Action Recognition Using a Deep Historical Long Short-Term Memory." [arXiv:1808.00845](#).
- Candila, V. and L. Palazzo (2020). "Neural Networks and Betting Strategies for Tennis." *Risks* **8**(3): 68.
- Fernando, T., S. Denman, S. Sridharan and C. Fookes (2019). "Memory augmented deep generative models for forecasting the next shot location in tennis." *IEEE Transactions on Knowledge and Data Engineering* **32**(9): 1785-1797.
- Kemeny, J. G., J. L. Snell and others (1960). *Finite markov chains*, van Nostrand Princeton, NJ.
- Kovalchik, S., M. Ingram, K. Weeratunga and C. Goncu (2020). "Space-Time VON CRAMM: Evaluating Decision-Making in Tennis with Variational generation of Complete Resolution Arcs via Mixture Modeling." [arXiv preprint arXiv:2005.12853](#).
- Mora, S. V. and W. J. Knottenbelt (2017) "Deep Learning for Domain-Specific Action Recognition in Tennis." 170-178 DOI: 10.1109/CVPRW.2017.27.
- Owens, N., C. Harris and C. Stennett (2003). *Hawk-eye tennis system*. 2003 International Conference on Visual Information Engineering VIE 2003.
- Sipko, M. and W. Knottenbelt (2015). "Machine learning for the prediction of professional tennis matches." *MEng computing-final year project, Imperial College London*.
- Wei, X., P. Lucey, S. Morgan, P. Carr, M. Reid and S. Sridharan (2015). Predicting Serves in Tennis using Style Priors. *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Sydney, NSW, Australia, Association for Computing Machinery: 2207–2215.
- Wei, X., P. Lucey, S. Morgan, M. Reid and S. Sridharan (2016). The thin edge of the wedge: Accurately predicting shot outcomes in tennis using style and context priors. *MIT Sloan Sports Analytics Conference (MITSSAC)*, (Boston, USA), 2016
- Wei, X., P. Lucey, S. Morgan and S. Sridharan (2013). Sweet-Spot: Using Spatiotemporal Data to Discover and Predict Shots in Tennis. *MIT Sloan Sports Analytics Conference (MITSSAC)*, (Boston, USA), 2013.
- Wei, X., P. Lucey, S. Vidas, S. Morgan and S. Sridharan (2014). Forecasting events using an augmented hidden conditional random field. *Asian Conference on Computer Vision*, Springer.
- Argall, B. D., Chernova, S., Veloso, M., & Browning, B. (2009). A survey of robot learning from demonstration. *Robotics and autonomous systems*, 57(5), 469-483.
- Arulkumaran, K., & Lillrank, D. O. (2021). A pragmatic look at deep imitation learning. [arXiv preprint arXiv:2108.01867](#).
- Pomerleau, D. A. (1988). Alvin: An autonomous land vehicle in a neural network. *Advances in neural information processing systems*, 1.
- Ho, J., & Ermon, S. (2016). Generative adversarial imitation learning. *Advances in neural information processing systems*, 29.
- Fu, J., Luo, K., & Levine, S. (2017). Learning robust rewards with adversarial inverse reinforcement learning. [arXiv preprint arXiv:1710.11248](#).
- Ghasemipour, S. K. S., Zemel, R., & Gu, S. (2020, May). A divergence minimization perspective on imitation learning methods. In *Conference on Robot Learning* (pp. 1259-1277). PMLR.
- Kostrikov, I., Agrawal, K. K., Dwibedi, D., Levine, S., & Tompson, J. (2018). Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning. [arXiv preprint arXiv:1809.02925](#).
- Li, Y., Swersky, K., & Zemel, R. (2015, June). Generative moment matching networks. In *International conference on machine learning* (pp. 1718-1727). PMLR.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.

Klaassen, F. J., & Magnus, J. R. (2001). Are points in tennis independent and identically distributed? Evidence from a dynamic binary panel data model. Journal of the American Statistical Association, 96(454), 500-509.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2020). Generative adversarial networks. Communications of the ACM, 63(11), 139-144.

Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784.

Mao, X., Li, Q., Xie, H., Lau, R. Y., Wang, Z., & Paul Smolley, S. (2017). Least squares generative adversarial networks. In Proceedings of the IEEE international conference on computer vision (pp. 2794-2802).

Borgeaud, S., Mensch, A., Hoffmann, J., Cai, T., Rutherford, E., Millican, K., ... & Sifre, L. (2021). Improving language models by retrieving from trillions of tokens. arXiv preprint arXiv:2112.04426.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805.

Williams, R. J., & Peng, J. (1991). Function optimization using connectionist reinforcement learning algorithms. Connection Science, 3(3), 241-268.

Wei, X., Lucey, P., Morgan, S., & Sridharan, S. (2016). Forecasting the next shot location in tennis using fine-grained spatiotemporal tracking data. IEEE Transactions on Knowledge and Data Engineering, 28(11), 2988-2997.