13TH ANNUAL
MIT SLOAN
SPORTS ANALYTICS CONFERENCE
MARCH 1 - 2, 2019 | BOSTON, MA
presented by ESPN

Using Deep Learning to Understand Patterns of Player Movement in the NBA

Akhil Nistala, John Guttag

Track: Basketball

Abstract

In 2011, SportVU fundamentally changed the way that basketball can be analyzed. STATS SportVU utilized a six-camera system installed in basketball arenas to track the real-time positions of players, at 25 frames per second. In this paper, we demonstrate how we can apply deep learning techniques to this data to produce a queryable database of basketball possessions.

We trained an unsupervised machine learning pipeline that generates a representation, called a *trajectory embedding*, of how individual players move on offense. The representation is a 32-dimensional vector of floating-point numbers that captures the semantics of a single player's movement, such as locations of the endpoints, screen actions, court coverage, and other spatial features. We generated nearly 3 million trajectory-embeddings from three seasons of data (2013-2014, 2014-2015, 2015-2016).

We found that the Euclidean distance between trajectory-embeddings is an excellent indicator of the visual similarity of the movements they encode. For example, two different movements of a post-up in the right block will have nearby embeddings; a post-up in the right block and a screen action above the left wing will have distant embeddings. This result led to the Similar Possessions Finder, a queryable database of basketball possessions.

The Similar Possessions Finder can be used to quickly answer queries such as "How much more frequently did Andre Drummond establish position on the right block than on the left block during the 2015-2016 regular season?" and "Find all possessions from the 2014 playoffs in which Chris Paul ran a screen action in the high post that ended with DeAndre Jordan scoring."

1. Introduction

Our goal was to develop an automated framework to quantitatively examine and compare patterns of individual player movements on offense. This process currently entails a film analyst watching hundreds of hours of game footage, carefully examining each possession and taking notes on certain plays. When a coach wants to look at a new play or movement, the entire process must be repeated. This approach is neither efficient (it takes hundreds of hours) nor comprehensive (film analysts don't retrieve all examples of a certain movement).

Some examples of the types of questions we want to be able to answer with our framework are:

- What fraction of Russell Westbrook's total movements from the 2013-2014 season were screen actions above the left wing?
- Which player has the most similar patterns of movement to LeBron James?
- Which possessions from the 2015 playoffs involved a dribble penetration and a kick out in which a 3-pointer was made?

2. Data

We used data collected by the STATS SportVU player tracking system [1]. This dataset contains player positions recorded at 25 frames per second over the course of every game. For this work, we used data from all games from the 2013-2014, 2014-2015, and 2015-2016 NBA seasons.

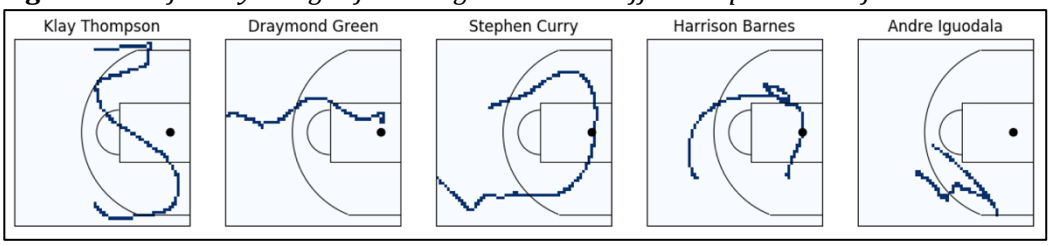
For each possession in the dataset, there is a time series $p(t) = < x_t, y_t >$ for each player, where t ranges from 0.00 seconds to the length of the possession. The possessions vary in length from only a few seconds to the length of the shot clock, with an average possession length of about 10 seconds. Table 1 shows the time series data for part of a Pacers possession.

Table 1: *SportVU time series data for part of a Pacers' possession, 2013-2014*

Gameclock (seconds)	Player	X	Y
715.34	David West	58.64	16.14
715.34	Roy Hibbert	75.86	37.16
715.34	George Hill	47.75	23.55
715.34	Paul George	71.53	6.07
715.34	Lance Stephenson	69.02	43.11
715.30	David West	59.09	16.13
715.30	Roy Hibbert	75.34	37.07
715.30	...	...	...

We represent the raw time series of each player's trajectory as an image, which we call a *trajectory-image*. This image-based representation is similar to previous work using basketball data [3][4]. Figure 1 has an example of five player trajectories in this format. We consider each player's trajectory on offense as a separate example.

Figure 1: *Trajectory-images for a single Warriors offensive possession from the 2016 NBA Finals*



We construct each image by first segmenting the offensive half of the basketball court into a 64 x 64 grid. This allows us to map each pixel in the image to a range of corresponding coordinates in the original time series format. We only consider the player's trajectory from the time the ball crosses half court to the end of the possession. For each coordinate (x, y) in the player's trajectory during the possession, we assign the corresponding pixel an intensity of 1. The resulting image shows each location the player reached during that possession, but not the order in which they were visited.

Building an image for each player's movement on offense gives a starting point for comparing player movements across many different possessions [5]. Trajectory-images allow us to represent any possession in a common, fixed-size input space. As we discuss in Section 3, representing possessions as images also allows us to take advantage of convolutional neural networks and other vision-based techniques.

3. Methodology

We want an encoding for each image that captures the *semantics* of what occurred during the possession. The Euclidean distance between a pair of image-representations should correlate with the semantic similarity in the images' movements. We want a dense representation for each possession that facilitates operations such as nearest neighbors and clustering. We therefore turn to unsupervised learning as a way to gain new insights into trajectory-images.

In other domains, unsupervised learning has been used to compute low-dimensional, neighborhood preserving embeddings of high dimensional data. Autoencoders are a type of unsupervised learning that have emerged in recent years as an effective way of learning low-dimensional encodings for high-dimensional input vectors [6]. An autoencoder architecture consists of a series of encoder layers that reduce the input image to a low-dimensional encoding, followed by a series of decoder layers that attempt to reconstruct the original image from the encoding.

Figure 2: *Convolutional autoencoder architecture*

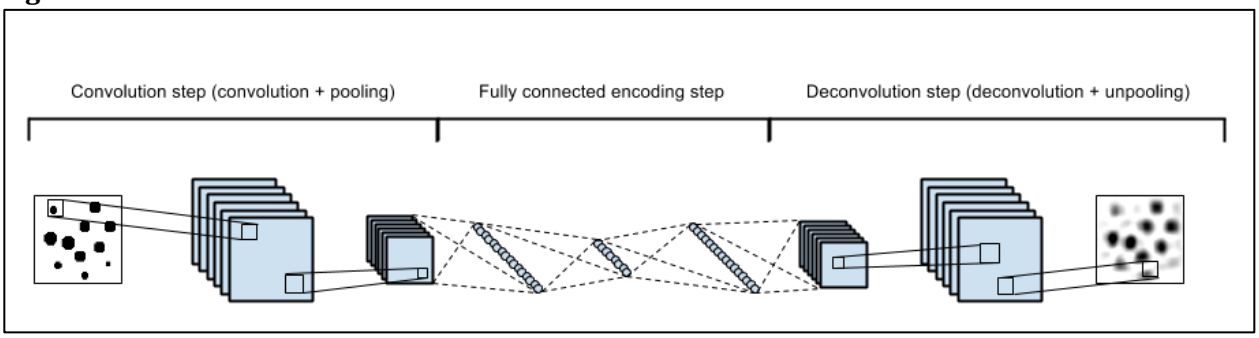
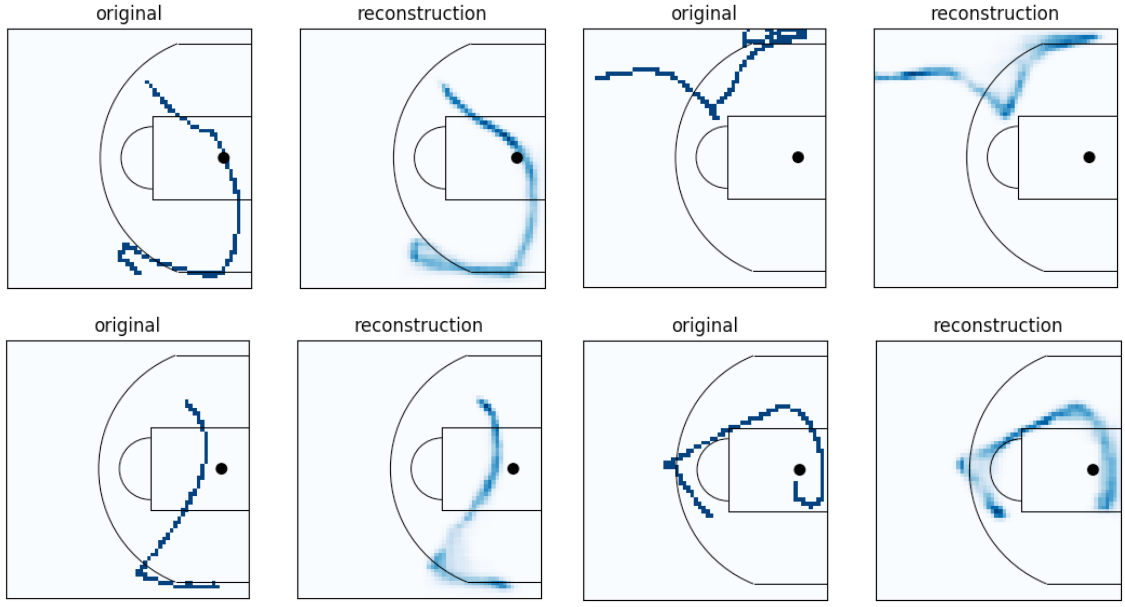


Figure 2 displays our autoencoder architecture [7]. It consists of a series of convolution and pooling operations that reduce the input 64x64 trajectory-image to a 4x4x8 volume. This volume is flattened to a 128-dimensional array and connected (with a fully-connected layer) to the 32-dimensional encoding. This encoding is connected to a 128-dimensional array, reshaped to a 4x4x8 volume, and put through a series of deconvolution and upsampling operations to build the reconstructed trajectory-image. Having fully-connected layers in the middle permits the network to learn patterns from different parts of each image, in addition to the spatial patterns it learns from the convolution and pooling operations.

We trained the network to optimize the binary cross entropy loss between the original and reconstructed trajectory-images. We trained the network for 500 epochs with a 70/30 split between training and validation data, and recovered the 32-dimensional encoding, the *trajectory-embedding*, for each of the 3 million trajectory-images. Figure 3 displays four trajectory-images and their reconstructions using our model.

Learning a low-dimensional encoding for each trajectory-image lets the network abstract away some details about each movement while preserving the more important ones. Although the reconstructions in Figure 3 are blurred, they still capture the overall movement and endpoints of their input trajectory-images.

Figure 3: *Four trajectory-images and their reconstructions from the autoencoder*



4. Experimental Results

We designed an experiment to evaluate the visual similarity of nearby trajectory-embeddings (by Euclidean distance). For 100 randomly chosen trajectory-embeddings, we found the 10 nearest neighbors in embedding-space and retrieved their trajectory-images. We gave each trajectory-embedding a score between 0 and 10, indicating how many of its ten nearby embeddings have a visually similar trajectory-image.

Figure 4

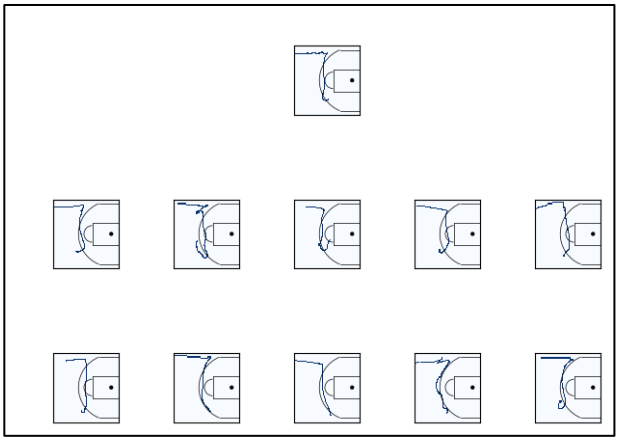


Figure 5

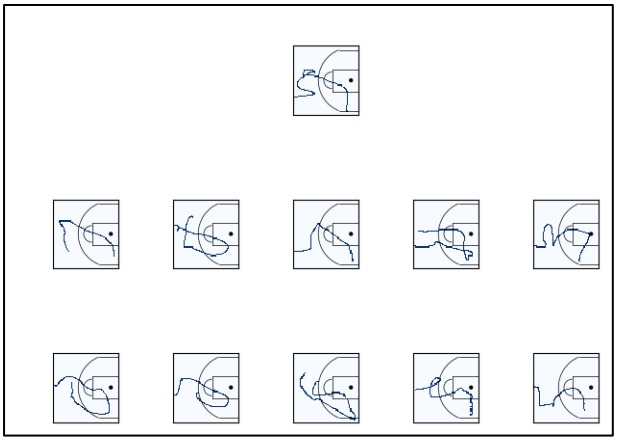


Figure 4 has the ten nearest-neighbors of a trajectory-image that scored a 10/10. Figure 5 has the ten nearest-neighbors of a trajectory-image that did not fare as well, scoring a 5/10. The average score for the 100 randomly chosen trajectory-embeddings was 9.1 (out of a possible 10), demonstrating that *trajectory-images that are nearby in embedding space are visually similar*.

5. Similar Possessions Finder

The Similar Possessions Finder is a queryable relational database of trajectory-embeddings. This database is possible as a direct result of the experimental results of section 4 – trajectory-images that are nearby in embedding space are visually similar.

Some sample queries that this tool supports are:

- Given any *N* input trajectory_images, find the twenty most similar possessions.
- Find all possessions from the 2015 playoffs in which Russell Westbrook drives to the basket.
- Find all possessions from the 2015-2016 season in which the Lakers run a screen action above the left wing with less than two minutes left in the quarter.

The output of a standard query is a list of tuples – (game_id, quarter, gameclock) – of the N nearest trajectory-embeddings by Euclidean distance. Table 2 displays a few rows of the Similar Possessions Finder. The five rows shown are all from a game between the Boston Celtics (Team_ID = 2) and the Milwaukee Bucks (Team_ID = 15).

- A Possession ID is the string concatenation of Game ID, Quarter, and Game Clock. This ensures that a Possession ID is shared only amongst the five trajectory embeddings of the five players on offense for a given possession.
- Each row's primary key is a concatenation of the Possession_ID and the Player_ID.

Table 2: *A few rows from the Similar Possession Finder database.*

Primary Key	Possession ID	Player ID	Game ID	Team ID	Quarter	Game Clock	Trajectory Embedding
…	…	…	…	…	…	…	…
"13723641152"	"13723641"	152	1372	2	3	641	*<–1.12, –0.4, …>*
"13723641236"	"13723641"	236	1372	2	3	641	*<–0.5, –0.37, …>*
"1372362011"	"13723641"	11	1372	2	3	641	*<0.37, 1.12, …>*
"13723620156"	"13723620"	156	1372	15	3	620	*<1.73, –0.73, …>*
"1372362097"	"13723620"	97	1372	15	3	620	*<–0.21, –0.97, …>*

In addition to the player-tracking time series data, we also received event data from STATS SportVU. This consists of the spatial coordinates and timestamps for events such as field goals, rebounds, and turnovers. Although we did not use this data to train our network, we did augment our instance of the Similar Possessions Finder (Table 3) in order to support metadata such as attempted_shot, made_shot, field_goal_coordinates, gave_assist, received_assist, had_turnover, ball_handling_time, primary_defender, current_score, possession_length, num_passes, was_fouled, and so forth.

Table 3: *A few rows from the Similar Possessions Finder database with supplemental metadata.*

Primary Key	Attempted Shot	Made Shot	Ball Handling Time	Primary Defender	Current Score	Possession Length	Num Passes
…	…	…	…	…	…	…	
"13723641152"	True	True	0.21	97	74-55	21 seconds	7
"13723641236"	False	False	0.35	156	74-55	21 seconds	7
"1372362011"	False	False	0.05	121	74-55	21 seconds	7
"13723620156"	True	False	0.75	11	74-58	9 seconds	3

“1372362097”	False	False	0.0	236	74-58	9 seconds	3

A database augmented with this metadata can support richer queries such as:

- Find all possessions in which Russell Westbrook drives to the basket and kicks it out to Andre Roberson for a 3-pointer.
- Find all possessions in which James Harden handles the ball for at least 80% of the possession but doesn't attempt a shot.
- Find all possessions in which LeBron James drives to the basket from the right and scores when defended by Draymond Green.
- Find all Spurs' possessions from the 2015 playoffs in which the Spurs passed the ball at least 5 times and a player converted a 3-pointer from the left corner.

We believe the Similar Possessions Finder can be a useful tool for NBA film analysts to quickly search for relevant possessions from many seasons of data. The format of the output – a tuple of (game_id, quarter, gameclock) for each relevant possession – is designed to let film analysts easily retrieve game footage.

6. Clustering

We are interested in learning a representative set of movements for how individual players move on offense. For example, point guards usually bring the ball up the court, good shooters may run to either 3-point corner to space the floor, and big men usually position themselves in the interior.

We first ran a clustering algorithm over the full set of dense representations to discover typical patterns of individual player movement. We then assigned a descriptive label to each cluster, and conducted an experiment to ensure that our clusters capture a diverse and comprehensive set of player movements. Sections 6.4 and 6.5 introduce the cluster profile, an instrument that summarizes player movements over the course of many games.

6.1 K-Means clustering

Running a clustering algorithm on all 3 million trajectory-embeddings gives us a way to group together similar trajectory-images. We can then look at the original trajectory of each cluster's central element (mediod) to get a sense of what kind of movement each cluster represents.

We used K-Means clustering because it is suitable for large datasets and is not as sensitive to outliers as other clustering techniques. We chose K=20 as the number of clusters after empirically finding it to yield a more interpretable clustering than smaller or higher values of K.

6.2 Cluster Mediods and Descriptions

We want to assign a label to each cluster that effectively describes what kind of movements its embeddings encapsulate. To do so, we computed each cluster's cluster-mean, and found the ten trajectory-embeddings nearest to this cluster mean. We looked at the trajectory-images of these ten embeddings, and assigned a description to each cluster based on the semantics of the movements captured by the trajectory-images.

Seven pairs of clusters contain movements that are symmetrical along an imaginary line that connects both baskets. For example, clusters 1 and 2 are movements to the right/left block from the top of the key. Clusters 7 and 8 are screen actions above the right and left wing. Five of the remaining clusters have movements that are inherently stand-alone, in that they equally span both

the left and right sides of the halfcourt. For example, cluster 15 is a run along the baseline and cluster 18 is a lateral movement in the high post. Only cluster 20 is neither stand-alone nor symmetric to another cluster.

Figure 7 displays the trajectory-image for each cluster-mediod. Table 4 has a full list of the cluster-mediod descriptions, with symmetric clusters paired together.

Figure 7: *Cluster mediods: trajectory-image of the nearest embedding to each cluster-mean*

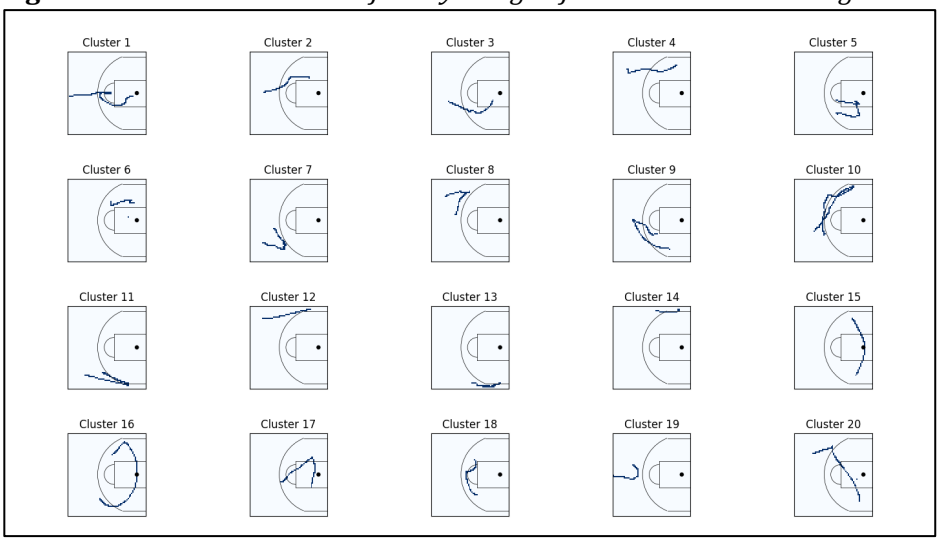


Table 4: *Cluster-mediod descriptions with near-reflection clusters paired together*

Cluster(s)	Descriptions
1, 2	Movement from top of key to right/left block
3, 4	Movement from top of key/above wings to right/left short corner
5, 6	Movement or screen action, on either side of the paint
7, 8	Movement or screen action above the wings
9, 10	Movement above wings
11, 12	Movement above wings, nearer to the sideline
13, 14	Movement along each sideline to corner
15	Run along baseline
16	Arc-like movement from sideline to sideline
17	Movement or screen action in paint
18	Lateral movement in high post
19	Run past halfcourt, screen action above the key
20	Movement from above left wing to right block

6.3 K-Means clusters capture a diverse and comprehensive set of player movements

We found that the trajectory-images of the ten embeddings nearest to each cluster-mean closely resemble reach other. However, this does not imply that all movements within a cluster look similar to each other. Some trajectories-images of embeddings that lie further away from the cluster-mean may look quite different than the cluster-mediod's trajectory-image, and contain patterns of movement similar to those in other clusters as well.

We want to evaluate the visual similarity of trajectory-images that are far from their cluster-mediod. We sorted each cluster's trajectory-embeddings by distance to the cluster-mediod, and then retrieved the trajectory-image of embeddings at the 50th-percentile and 75th-percentile positions.

Figure 7 shows the trajectory-image of each cluster's 50th-percentile embedding, and Figure 8 shows the trajectory-image of each cluster's 75th-percentile embedding. Despite being far from the cluster-mean, these trajectories look remarkably similar to the mediod-trajectories, and are consistent with the cluster-mediod descriptions, indicating that the twenty clusters do an excellent job of capturing a *diverse* and *comprehensive* set of player movements.

Figure 7: *Trajectory-image of each cluster's 50th-percentile trajectory-embedding*

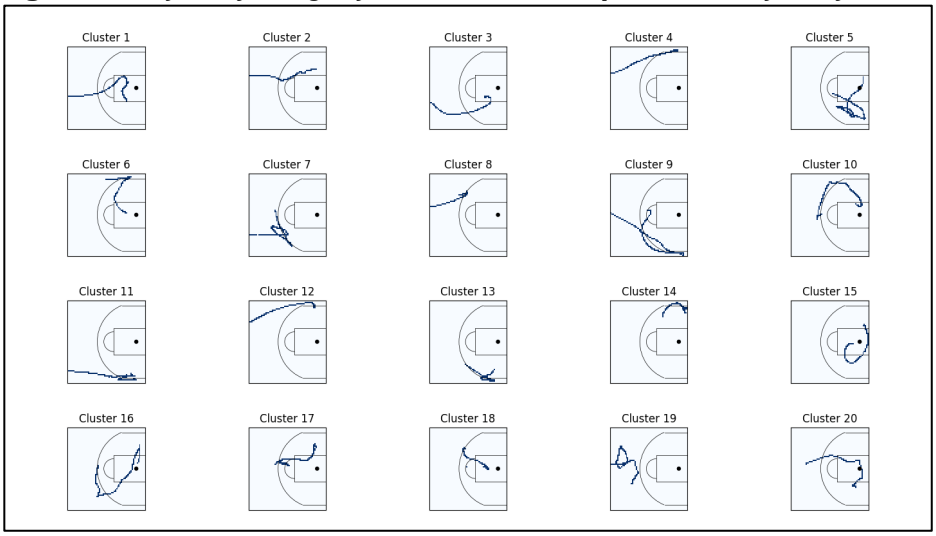
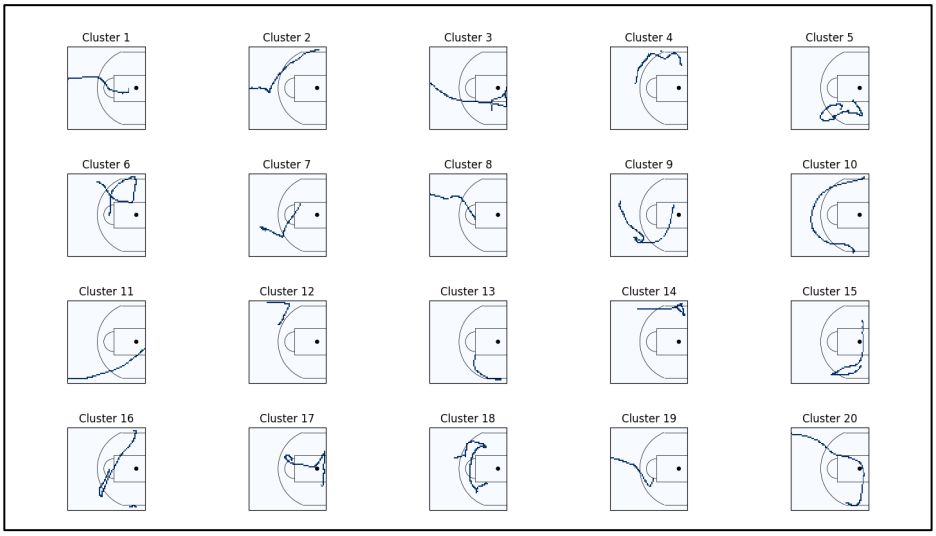


Figure 8: *Trajectory-image of each cluster's 75th-percentile trajectory-embedding*



6.4 Summarizing Player Movements with Cluster-Profiles

A player's patterns of movement over the course of many games can be summarized with a cluster-profile. A cluster-profile is a 20-dimensional vector, that represents how frequently a player's trajectory-images over a span of games fall into each of the 20 clusters. Each element of a cluster-profile for a player represents the fraction of his trajectories that fall into that corresponding cluster.

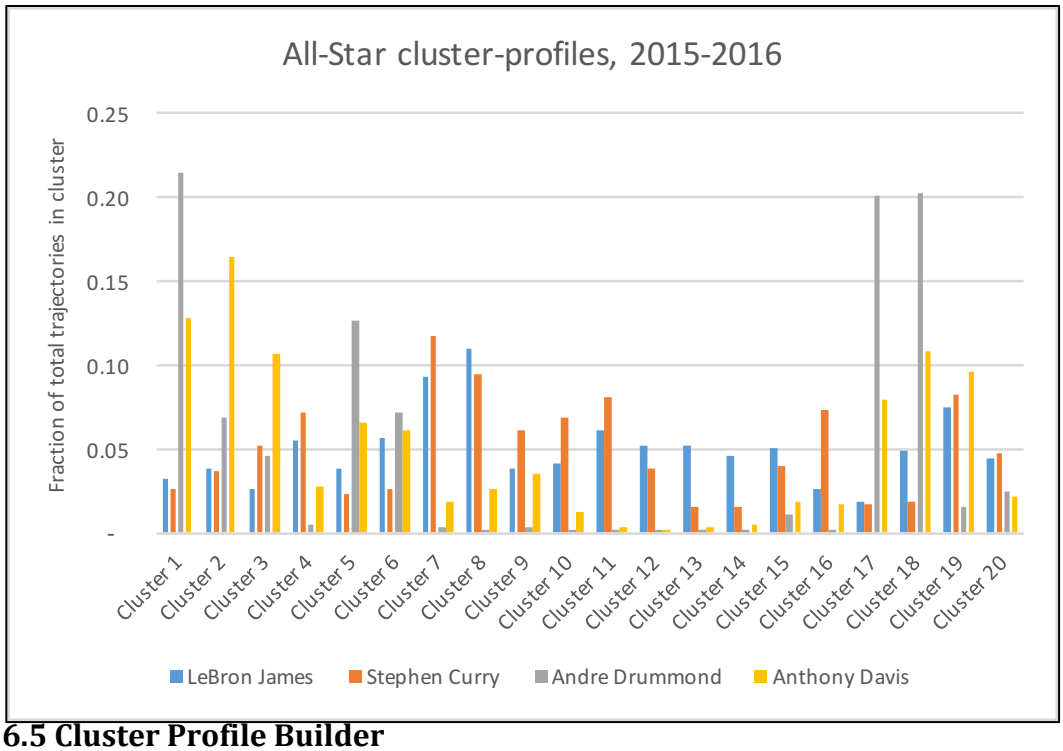
$$cluster\_profile_P, i = \#trajectory\_embeddings \text{ for player } P \text{ in cluster } i \,/\, \#total\ trajectory\_embeddings \text{ for player } P$$

In Figure 9, we show the cluster-profiles for four players, computed across all regular season games played in the 2015-2016 season. LeBron James (blue) and Stephen Curry (orange) have similar cluster-profiles, speaking to their versatility on offense. They are both strong passers, and capable of scoring on their own, from both inside and outside. This is in stark contrast to the cluster-profile of Andre Drummond (gray), who, as a pure center, is much more limited in the types of movements he makes on offense. The cluster-profile of Anthony Davis (yellow), a versatile center, is also given.

LeBron James shoots with his right hand but is naturally left-handed. His cluster-profile reveals that more of his movements fall into the left half of the court (clusters 4, 6, 8) than in the right half (clusters 3, 5, 7). The opposite is true for Andre Drummond, who is naturally right-handed and heavily favors the right side of the court. Drummond has more than twice as many movements in clusters 1, 3, and 5 (right-half) than in clusters 2, 4, and 6 (left-half).

James and Curry also have significantly more movements in clusters 9, 10, 11, and 12 than Drummond and Davis. These clusters all contain screen actions and movements above the wings, with clusters 11 and 12 nearer the sidelines. James and Curry are both strong passers and capable shooters, and often engage in screen actions above the wings to create mismatches. Drummond and Davis do more of their damage in the paint, with a large fraction of their movements in clusters 17 and 18.

Figure 9: *Cluster-profiles for James, Curry, Drummond, Davis from the 2015-2016 season*

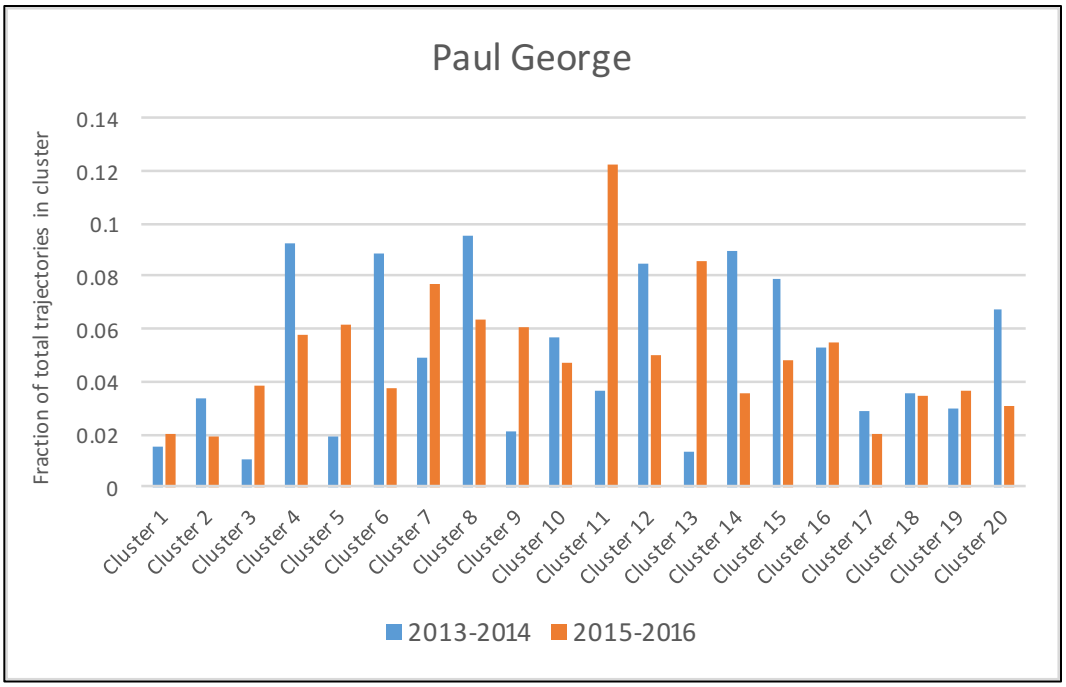


6.5 Cluster Profile Builder

The Cluster Profile Builder is a tool that generates cluster-profiles from an input set of players and a set of possessions. The set of players can be any group of players – a single player, a team, a position, and so forth. Similarly, the set of possessions can be a single game or many games, a range of dates, a season, an opponent, and even against a specific defender. In this section we use the Cluster Profile Builder to analyze a few prominent events from 2014 – Paul George's leg injury and the Kevin Love trade.

Figure 10 displays the cluster-profiles for Paul George before and after he broke his right leg during the 2014 offseason. We observe that Paul George made significantly more plays in the right half of the court during the 2015-2016 season than the 2013-2014 season. Aggregate movement in the right half of the court – clusters 1, 3, 5, 7, 9, 11, and 13 – increased from 16.5% to 46.6%, whereas movements in the left half of the court – clusters 2, 4, 6, 8, 10, 12, and 14 – decreased from 54.1% to 31.0%.

Figure 10: *Paul George's cluster-profiles for full seasons before and after his leg injury*



Kevin Love was the focal point of the Timberwolves' offense from 2010 to 2014. His role with the Cavaliers changed drastically as he became the third option on offense after LeBron James and Kyrie Irving. This manifests itself in his box-score statistics (Table 5) as well as his cluster-profile (Figure 11).

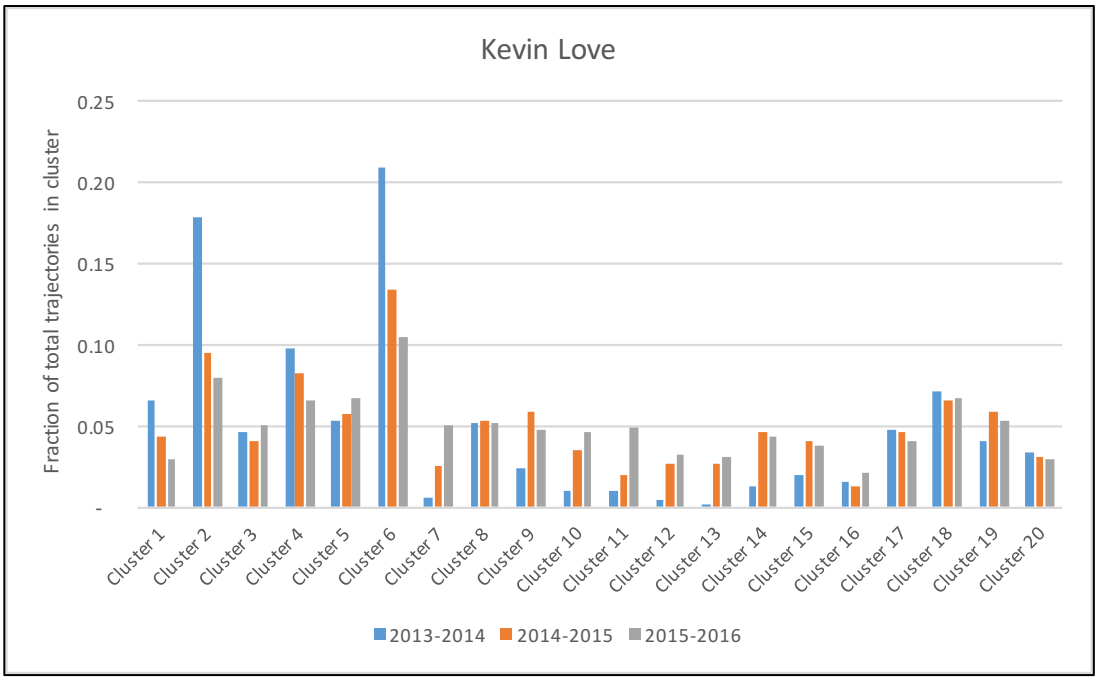
Love liked to post up on the left block in Minnesota, a place he did not occupy as much in Cleveland. Cluster 2 (run to the left block from the top of the key) decreased from 17.9% to 9.4% of Love's movements during his first season in Cleveland. Cluster 6 (movement left of the paint) also decreased from 20.9% to 13.3% of Love's movements during this time span.

Clusters 9 through 12 are perhaps the biggest indicator of the vastly different role Love had for the Cavaliers than he did for the Timberwolves. Clusters 9 and 10 contain movements along either wing; clusters 11 and 12 also contain movements along the wings, closer to the sidelines. These four clusters accounted for just 4.8% of Love's trajectory-images from 2013-2014, a number that increased to 14.0% in 2014-2015 and 17.5% in 2015-2016. Clusters 13 and 14 (movements along each sideline to the corner) also increased from 1.5% to 7.3% during this same time span. Love accepted a new role in the Cavaliers offense and became more of a perimeter player, veering away from his favorite spot near the left block.

Table 5: *Love per 36 minutes, 2013 to 2016*

	PTS	REB	AST	FG%
2013-2014	25.9	12.4	4.4	.457
2014-2015	17.5	10.4	2.4	.434
2015-2016	18.3	11.3	2.8	.419

Figure 11: *Kevin Love's cluster-profiles for 2013-2014, 2014-2015, and 2015-2016*


Kevin Love
Fraction of total trajectories in cluster
0.25
0.20
0.15
0.10
0.05
-
Cluster 1
Cluster 2
Cluster 3
Cluster 4
Cluster 5
Cluster 6
Cluster 7
Cluster 8
Cluster 9
Cluster 10
Cluster 11
Cluster 12
Cluster 13
Cluster 14
Cluster 15
Cluster 16
Cluster 17
Cluster 18
Cluster 19
Cluster 20
2013-2014
2014-2015
2015-2016

7. Summary and Conclusion

We learned a low-dimensional representation of a single player's movement over the course of one possession on offense. For each possession, we use each of the five offensive players' raw time series (X,Y) coordinates to build a trajectory-image depicting how that player moved. These images let us represent all movements on offense in a common, fixed-sized input space. We constructed 3 million images from three seasons of NBA player tracking data.

Next, we put all 3 million trajectory-images through a convolutional neural network to learn a low-dimensional encoding for each image. Each trajectory-embedding captures the spatial patterns in a trajectory-image, such as the start and end point, screen actions, and court coverage. Training a single model for all images lets us learn a low-dimensional encoding for each movement, in the context of all movements.

Our numerical abstraction for player movements over individual basketball possessions can be used to compare movements across players and teams, analyze changes in player movement over time, efficiently search a database of possessions for a certain type of movement, and so forth. This research has demonstrated that deep learning can be used to learn patterns of basketball movement on offense. We are very interested in applying the same methodology to defensive movements.

References

[1] Amick, Sam. NBA Caucus: Will Analytics Overtake the Old School? 5 Mar. 2014, www.usatoday.com/story/sports/nba/2014/03/05/mit-sloan-sports-analytics-conference-takeaways-phil-jackson/6087007/.
[2] https://www.stats.com/sportvu-basketball/
[3] Wang, Kuan-Chieh, and Richard Zemel. "Classifying NBA offensive plays using neural networks." Proc. MIT Sloan Sports Analytics Conference. 2016.
[4] Harmon, Mark, Patrick Lucey, and Diego Klabjan. "Predicting Shot Making in Basketball using Convolutional Neural Networks Learnt from Adversarial Multiagent Trajectories." arXiv preprint arXiv:1609.04849 (2016)
[5] Brooks, J. (2018) Using Machine Learning to Derive Insights from Sports Location Data
[6] G.E. Hinton, R.R. Salakhutdinov. Reducing the Dimensionality of Data with Neural Networks. http://science.sciencemag.org/content/313/5786/504
[7] https://swarbrickjones.files.wordpress.com/2015/04/conv_autoencoder.png