# Where Will They Go? Predicting Fine-Grained Adversarial Multi-Agent Motion using Conditional Variational Autoencoders

Panna Felsen[1,2], Patrick Lucey[2], and Sujoy Ganguly[2]

[1]BAIR, UC Berkeley [2]STATS
panna@eecs.berkeley.edu, {plucey, sganguly}@stats.com

**Abstract.** Simultaneously and accurately forecasting the behavior of many interacting agents is imperative for computer vision applications to be widely deployed (*e.g.*, autonomous vehicles, security, surveillance, sports). In this paper, we present a technique using conditional variational autoencoder which learns a model that "personalizes" prediction to individual agent behavior within a group representation. Given the volume of data available and its adversarial nature, we focus on the sport of basketball and show that our approach efficiently predicts context-specific agent motions. We find that our model generates results that are *three times* as accurate as previous state of the art approaches (5.74 ft vs. 17.95 ft).

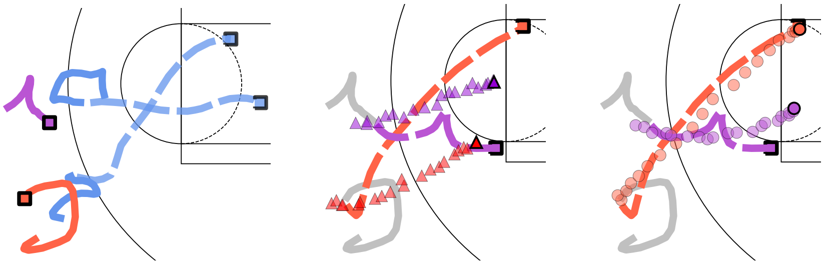
**Keywords:** forecasting, motion prediction, multi-agent tracking, context aware prediction, conditional variational autoencoders

## 1 Introduction

Humans continuously anticipate the future states of their surroundings. Someone extending a hand to another is likely initiating a handshake. A couple entering a restaurant is likely looking for a table for two. A basketball player on defense is likely trying to stay between their opponent and the basket. These predictions are critical for shaping our daily interactions, as they enable humans to navigate crowds, score in sports matches, and generally follow social mores. As such, computer vision systems that are successfully deployed to interact with humans must be capable of forecasting human behavior.

In practice, deploying a computer vision system to make a fine-grain prediction is difficult. Intuitively, people rely on context to make more accurate predictions. For example, a basketball player may be known to stay back in the lane to help protect the rim. The ability to leverage specific information, or *personalize*, should improve the prediction of fine-grained human behavior.

The primary challenge of personalizing prediction of multi-agent motion is to develop a representation that is simultaneously robust to the number of possible permutations arising in a situation and sufficiently fine-grained, so the output prediction is at the desired level of granularity. One typically employees one



a) Input b) VAE baseline c) Our method

Fig. 1: a) Given a 2D trajectory history of moving agents (solid lines), and the future motion of a subset of the agents (blue dashed lines); our **prediction task** b) is to generate the most likely motion of the other agents (orange, purple dashed lines). Standard approaches are unable to capture the influence of the group motion (triangles). c) Our method improves performance by incorporating context-specific information (circles).

of two approaches: i) *bottom-up* – where each trajectory has the same model applied to it individually, or ii) *top-down* – where a group representation of all trajectories has one model applied to it all at once. The data and target application mainly drive the choice of approach. Typically, in settings with a variable number of agents, *e.g.*, autonomous vehicles or surveillance, one uses a bottom-up approach [1–3]. When the number of agents is fixed, *e.g.*, sports, faces, and body pose one prefers a top-down approach [4–7].

While efficient for heavily structured problems, current top-down methods cannot incorporate the necessary context to enable *personalized* prediction, and often require pre-computing some heuristic group representation. Whereas, bottom-up approaches can personalize via a large refinement module [1]. In this paper, we show that by using a conditional variational autoencoder (CVAE), we can create a generative model that simultaneously learns the latent representation of multi-agent trajectories and can predict the agents' context-specific motion.

Due to the vast amount of data available and its adversarial, multi-agent nature, we focus on predicting the motion paths of basketball players. Specifically, we address the problem of forecasting the motion paths of players during a game (Fig. 1a). We demonstrate the effectiveness of our approach on a new basketball dataset consisting of sequences of play from over 1200 games, which contains position data of players and the ball.

To understand the function of initial data representation, context, personalization of agent trajectory prediction and generative modeling, we divide our problem into three parts. First, to understand the role of data representation on prediction, we predict the offense given the motion history of all players (Fig. 1b). By applying *alignment* to the multi-agent trajectories we minimize the problem of permutation allowing our group representation of player motion to outperform the current state of the art methods. Next, to understand the role of context, we compare the prediction of offensive agents given the motion of the defense, player and team identities. We use separate encoders for context and player/team identity which we connect to the variational layer, as opposed to being used in a ranking and refinement layer, and thus act directly as conditionals. By conditioning on context with alignment and identity, we can generate a very accurate, fine-grained, prediction of any group of agents without the need for an additional refinement module (Fig 1c). Finally, we tackle the challenge of forecasting the motion of subsets of players (a mixture of offense and defense), given the motion of the other remaining players. Again we find that our CVAE far outperforms the previous state of the art methods by a factor of two and that it can make reasonable predictions given only the motion history and the player and team identities when predicting the future motion of all ten players.

**Our primary contributions are:**

1. How to use context and identity as conditionals in CVAE thus removing the need for ranking and refinement modules.
2. Utilizing multi-agent alignment to personalize prediction
3. A dataset of fine-grained, personalized, adversarial multi-agent tracking data which will be made publicly available for research purposes.

## 2 Related Work

**Forecasting Multi-Agent Motion** Lee et al. [1] provide an excellent review of recent path prediction methods, in which they chronicle previous works that utilize classical methods, inverse reinforcement learning, interactions, sequential prediction and deep generative models. For predicting multi-agent motion paths, there are two primary bodies of work: *bottom-up* and *top-down* approaches.

Regarding bottom-up approaches, where the number of agents varies, Lee *et al.* [1] recently proposed their DESIRE framework, which consisted of two main modules. First, they utilized a CVAE-based RNN encoder-decoder which generated multiple plausible predictions. These predictions, along with context, were fed to a ranking and refinement module that assigns a reward function. The predictions are then iteratively refined to obtain a maximum accumulated future reward. They demonstrated the approach on data from autonomous vehicles and aerial drones and outperformed other RNN-based methods [3]; however, in the absence of the refinement module, the predictions were poor.

For predicting variable numbers of humans moving in crowded spaces, Alahi et al. [2] introduced the idea of "Social LSTMs" which connected neighboring LSTMs in a social pooling layer. The intuition behind this approach is that instead of utilizing all possible information in the scene, the model only focuses on people who are near each other. The model will then learn that behavior from data, which was shown to improve over traditional approaches which use hand-crafted functions such as social forces [8]. Many authors have applied similar methods for multi-agent tracking using trajectories [9–11].

Nearly all work that considers multiple agents via a top-down approach is concerned with modeling behaviors in sports. Kim et al. [12] used the global motion of all players to predict the future location of the ball in soccer. Chen et al. [13] used an occupancy map of noisy player detections to predict the camera-motion of a basketball broadcast. Zheng et al. [14] used an image-based representation of player positions over time to simulate the future location of a basketball. Lucey et al. [5] learned role representations from raw positional data, while Le et al., [7] utilized a similar representation with a deep neural network to imitate the motion paths of an entire soccer team. Felsen et al. [15] used hand-crafted features to predict future events in water polo and basketball. Lastly Su et al. [16] used ego-centric appearance and joint attention to model social dynamics and predict the motion of basketball players. In this paper, we utilize the representation which most closely resembles Le et al. [7], the CVAE approach utilized by [1], and a prediction task similar to [16].

**Personalization to Tracking Data** Recommendation systems, which provide *personalized* predictions for various tasks often use matrix factorization techniques [17]. However, such techniques operate under the assumption that one can decompose the data linearly, using hand-crafted features to capture the non-linearities. However, in conjunction with deep models and the vast amount of vision data, recommendation engines based on vision data are starting to emerge. Recently, Deng et al. [18] used a factorized variational autoencoder to model audience reaction to full-feature length movies. Charles et al. [19] proposed using a CNN to personalize pose estimation to a person's appearance over time. Insafutdinov et al., [6] used a graph partitioning to group similar body-parts to enable effective body-pose tracking. In all of these works, they use their deep networks to find the low-dimensional embedding at the encoder state which they use to personalize their predictions. In this work, we followed a similar strategy but included the embedding in a variational module.

**Conditional Variational Autoencoders** Variational Autoencoders [20] are similar to traditional autoencoders, but have an added regularization of the latent space, which allows for the generation of new examples in a variety of contexts [21, 22]. Since the task of fine-grained prediction is naturally one in which history and context determine the future motions, we utilize a conditional variational autoencoder (CVAE) [23, 24]. In computer vision, CVAEs have recently been used for inpainting [25, 26], and for predicting the future motion of agents in complex scenes [1, 27]. In this paper, we apply the idea of conditioning on the history and the surrounding context to predict the personalized adversarial motion of multiple agents without ranking or refinement.

## 3 Basketball Tracking Dataset

Team sports provide an ideal setting for evaluating personalized behavior models. Firstly, there is a vast amount of labeled data in sports, including potentially thousands of data points for each player. Furthermore, the behaviors in team sports are well-defined and complex, with multiple agents simultaneously interacting collaboratively and adversarially. Therefore, sports tracking data is a good compromise between completely unstructured tracking data (*e.g.*, pedestrian motion where the number of agents is unconstrained) and highly structured

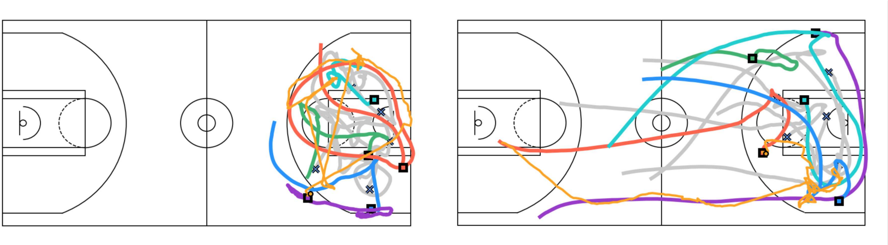


Fig. 2: **Dataset** Example plays from our basketball dataset, which contains 95,002 12-second sequences of offense (color), defense (gray), and ball (orange) 2D overhead-view trajectories. The identity, team, and canonical position of each player are known.

data (*e.g.*, body pose or facial tracking where the number of agents is both fixed and physically connected). To that end, we present basketball as a canonical example of a team goal sport, and we introduce a new basketball dataset.

Our proposed dataset is composed of 95,002 12-second sequences of the 2D basketball player and ball overhead-view point trajectories from 1247 games in the 2015/16 NBA season. The trajectories are obtained from the STATS in-venue system of six stationary, calibrated cameras, which projects the 3D locations of players and the ball onto a 2D overhead view of the court. Fig. 2 visualizes two example sequences. Each sequence, sampled at 25 Hz, has the same team on offense for the full duration, ends in either a shot, turnover or foul. By eliminating transition plays where teams switch from defense to offense mid-sequence, we constrain the sequences to contain persistent offense and defense. Each sequence is zero-centered to the court center and aligned, so the offense always shoots toward the court's right-side basket. In our experiments, we subsample the trajectory data at 5 Hz, thereby reducing the data dimensionality without compromising information about quick changes of direction.

**Personalization** We label each sequence with its player identity, team, canonical position (*i.e.*, point/shooting guard, small/power forward, center), and aligned position (Section 4.3). Only the 210 players with the most playing time across all sequences are assigned unique identities. The remaining players are labeled by their canonical position, thus limiting the set of player identities.

**Data splits** The data is randomly split into train, validation, and test sets with 60 708, 15 244, and 19 050 sequences in each respective split.

## 4 Methods

We frame the multi-agent trajectory prediction problem as follows: In a 2D environment, a set $\mathcal{A}$ of interacting agents are observed over the time history $[t_0, t_q]$ to have trajectories $X_{\mathcal{A}}^{[t_0,t_q]} = \{X_i^{[t_0,t_q]}\}|_{\forall i \in \mathcal{A}}$. The trajectory history of the $i^{th}$ agent is defined as $X_i^{[t_0,t_q]} = \{x_i^{t_0}, x_i^{t_0+1}, \cdots, x_i^{t_q}\}$, where $x_i^t$ represents the 2D coordinates of the trajectory at time $t$. We wish to predict the subsequent future

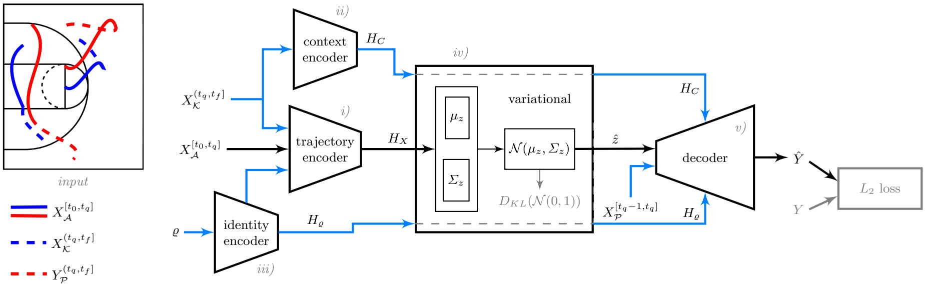


Fig. 3: **Model architecture.** The inputs to the *i)* trajectory encoder are the tracking history of all players $X_{\mathcal{A}}^{[t_0,t_q]}$, the identity $\varrho$, and the context $X_{\mathcal{K}}^{(t_q,t_f]}$. The trajectory context $X_{\mathcal{K}}^{(t_q,t_f]}$ is *ii)* encoded as $H_C$. The one-hot-encoded player or team identity $\varrho$ is *iii)* encoded as $H_\rho$. The *iv)* variational module predicts the mean $\mu_z$ and standard deviation $\Sigma_z$ of the latent variable distribution $\mathcal{N}(\mu_z, \Sigma_z)$. A random sample $\hat{z}$ from $\mathcal{N}(\mu_z, \Sigma_z)$ is input to the decoder, along with the conditionals $H_C$, $H_\rho$, and the last one second of of player motions $X_{\mathcal{A}}^{[t_q-fps,t_q]}$. The *v)* decoder then predicts the future paths $\hat{Y}$. At train time the KL divergence and $L_2$ loss are minimized.

motion, to time $t_f$, of a subset of agents $\mathcal{P} \subseteq \mathcal{A}$. In other words, our objective is to learn the posterior distribution $P(Y_{\mathcal{P}}^{(t_q,t_f]} | X_{\mathcal{A}}^{[t_0,t_q]}, \mathcal{O})$ of the future trajectory motion of the agents in subset $\mathcal{P}$, specifically $Y_{\mathcal{P}}^{(t_q,t_f]} = \{Y_j^{(t_q,t_f]}\}|_{\forall j \in \mathcal{P}}$.

In addition to the observed trajectory history, we also condition our learned future trajectory distribution on other available observations $\mathcal{O}$. In particular, $\mathcal{O}$ may consist of: 1) the identities $\varrho$ of the agents in $\mathcal{P}$, and 2) the future context $C$, represented by the future trajectories $X_{\mathcal{K}}^{(t_q,t_f]} = \{X_\ell^{(t_q,t_f]}\}|_{\forall \ell \in \mathcal{K}}$ of agents in the set $\mathcal{K} \subset \mathcal{A}$ s.t. $\mathcal{K} \cup \mathcal{P} = \mathcal{A}$, $\mathcal{K} \cap \mathcal{P} = \{\}$. One of the main contributions of this work is how to include various types of information into $\mathcal{O}$, and the influence of each information type on the prediction accuracy of $Y_{\mathcal{P}}^{(t_q,t_f]}$ (Section 5.1).

The conditionals and inputs to our model are each encoded in their encoders. To learn the posterior, we use a CVAE, which allows for the conditional generation of trajectories while modeling the uncertainty of future prediction. In our case, the CVAE learns to approximate the distribution $P(Y_{\mathcal{P}}^{(t_q,t_f]} \mid X_{\mathcal{A}}^{[t_0,t_q]}, \mathcal{O})$ by introducing a random $D_z$-dimensional latent variable $z$. The CVAE enables solving one-to-many problems, such as prediction, by learning a distribution $Q(z = \hat{z} \mid X_{\mathcal{A}}^{[t_0,t_q]}, \mathcal{O})$ that best reconstructs $Y_{\mathcal{P}}^{(t_q,t_f]}$.

Fig. 3 shows our overall model architecture, which is divided into the five modules: i) the trajectory encoder with $X_{\mathcal{A}}^{[t_0,t_q]}$ and $O$ as input, ii) the context encoder with $X_{\mathcal{K}}^{(t_q,t_f]}$ as input, iii) the identity encoder with $\varrho$ as input, iv) a variational module, and v) the trajectory decoder with sampled latent variable $\hat{z}$ and encoded conditionals as input. The input to the variational module is the joint encoding of the trajectory history $X_{\mathcal{A}}^{[t_0,t_q]}$ with the context and identity. The trajectory history, context, and identity serve as our conditionals in the

CVAE, where the context and identity are each separately encoded before being concatenated with $\hat{z}$ as input to the decoder. The trajectory history *conditional* $X_{\mathcal{P}}^{[t_q-1,t_q]}$ for $\hat{z}$ is the last one second of observed trajectory history of the agents in $\mathcal{P}$. This encourages the model predictions to be consistent with the observed history, as our decoder outputs $X_{\mathcal{P}}^{[t_q-1,t_q]}$ concatenated with $Y_{\mathcal{P}}^{(t_q,t_f]}$.

### 4.1 Training phase

We have modeled the latent variable distribution as a normal distribution

$$\begin{aligned} Q\left(z=\hat{z} \mid X_{\mathcal{A}}^{[t_0,t_q]}, X_{\mathcal{K}}^{(t_q,t_f]}, \varrho\right) &= Q\left(z=\hat{z} \mid H_x, H_C, H_\varrho\right) \\ &\sim \mathcal{N}\left(\mu_z, \Sigma_z\right). \end{aligned} \tag{1}$$

Therefore, at train time the variational module minimizes the Kullback-Leibler (KL) divergence ($D_{KL}$) and the trajectory decoder minimizes Euclidean distance $\left\|Y-\hat{Y}\right\|_2^2$. For simplicity, let $Y=(X_{\mathcal{P}}^{[t_q-1,t_q]}, Y_{\mathcal{P}}^{(t_q,t_f]})$. The total loss is

$$L=\left\|Y-\hat{Y}\right\|_2^2+\beta D_{KL}(P||Q), \tag{2}$$

where $P\left(z \mid X_{\mathcal{A}}^{[t_0,t_q]}, X_{\mathcal{K}}^{(t_q,t_f]}, \varrho\right) = \mathcal{N}(0,1)$ is a prior distribution and $\beta$ is a weighting factor to control the relative scale of the loss terms. We found that for $\beta=1$, our model without the conditionals (VAE) would roughly predict the mean trajectory, whereas when $\beta \ll 1$ we were able to predict input-dependent motion. In our proposed model, we observed that $\beta=1$ performed as well as $\beta \ll 1$, so in all our experiments except for the vanilla VAE, we use $\beta=1$.

### 4.2 Testing phase

At test time, the input into the trajectory encoder is the trajectory history of all agents $X_{\mathcal{A}}^{[t_0,t_q]}$, the future trajectories of the agents not predicted $X_{\mathcal{K}}^{(t_q,t_f]}$, and the encoded agent identities $\varrho$. The variational module takes the encoded trajectory $H_X$, which is also conditioned on the context $X_{\mathcal{K}}^{(t_q,t_f]}$ and the player identities $\varrho$, and returns a sample of the random latent variable $\hat{z}$. The trajectory decoder then infers the tracks of the agents to be predicted $Y_{\mathcal{P}}^{(t_q,t_f]}$ given a sampled $\hat{z}$, the encoded context $H_C$, the encoded identities $H_\varrho$, and the final one second of trajectory history for the agents to be predicted, $X_{\mathcal{P}}^{[t_q-1,t_q]}$.

### 4.3 Trajectory alignment

The network inputs are a concatenation of each 2D agent trajectories. For example, the input $X_{\mathcal{A}}^{[t_0,t_q]}$ forms an $|\mathcal{A}| \times (t_q \cdot 5) \times 2$ array, where $|\mathcal{A}|$ is the number of agents, $t_q \cdot 5$ is the total number of temporal samples over $t_q$ seconds sampled at 5 Hz. One of the significant challenges in encoding multi-agent trajectories is the presence of permutation disorder. In particular, when we concatenate the trajectories of all agents in $\mathcal{A}$ to form $X_{\mathcal{A}}^{[t_0,t_q]}$, we need to select a natural and consistent ordering of the agents. If we concatenate them in a random order, then two similar plays with similar trajectories will have considerably different representations. To minimize the permutation disorder, we need an agent ordering that is consistent from one play to another.

If we have a variable number of agents, it is natural to use an image-based representation of the agent tracks. In our case, where we have a fixed number of agents, we instead align tracks using a tree-based role alignment [28]. This alignment has recently been shown to minimize reconstruction error; therefore it provides an optimal representation of the multi-agent trajectories.

In brief, the tree-based role alignment uses two alternating steps, *i*) an Expectation-Maximization (EM) based alignment of agent positions to a template and *ii*) K-means clustering of the aligned agent positions, where cluster centers form the templates for the next EM step. Alternating between EM and clustering leads to a splitting of leaf nodes in a tree until either there are fewer than $M$ frames in a cluster or the depth of the tree exceeds $D$. For our experiments we used $D = 6$ and trained separate trees for offense ($M = 400$) and defense ($M = 4000$). To learn a per-frame alignment tree, we used 120K randomly sampled frames from 10 NBA games from the 2014/15 season.

### 4.4 Implementation details

**Architecture** All encoders consist of $N$ fully connected layers, where each layer has roughly half the number of units as its input layer. We experimented with different input histories, prediction horizons, and player representations, so we dynamically set the layer structure for each experiment, while maintaining 64 and 16 units in the final layer of the trajectory and context encoders, respectively. For the identity encoder, the final output size depended on the identity representation $\varrho$, which was either: 1) a (concatenated) one-hot encoding of the team(s) of the players in $\mathcal{P}$ (output dimension 5 for single team and 16 for mixed), and 2) a (concatenated) one-hot encoding of each player identity in $\mathcal{P}$. See the supplementary for the full architecture details.

**Learning** At train time we minimize the loss via backpropagation with the ADAM optimizer, batch size 256, initial learning rate 0.001, and 0.5 learning rate decay every 10 epochs of size 200K. We also randomly sample the training set so that the number of times a sequence appears in an epoch is proportional to the number of players it has with unique identity.

## 5 Experiments

We evaluate the effect on prediction performance of: 1) each information type input in our proposed model architecture (Section 5.1); 2) the number and types of agents in the input and output, *i.e.*, offense only, defense only, and both offense and defense (Section 5.2); 3) the predicted agents' during-play role

(Section 5.3); 4) the length of the history input (Section 5.4); and 5) the length of the prediction horizon (Section 5.5).

**Baselines** Our baselines are: velocity-based extrapolation, nearest neighbor retrieval, vanilla and Social LSTMs, and a VAE. Retrieval was performed using nearest neighbor search on the aligned (Section 4.3) trajectory history of the agents we wish to predict, matching the evaluation track histories to the training track histories based on minimum Euclidean distance. Then, we compare the error of the future trajectories of the top-k results to the ground truth. We found that these predictions are very poor, performing significantly worse than velocity-based extrapolation. Next, we compared our performance with the previous state of the art recurrent prediction methods, namely a vanilla LSTM and the Social LSTM. We found that the vanilla LSTM performed poorly with around 25 ft error for 4 s prediction horizon. The inclusion of social pooling improved the performance of the LSTM with 18 ft error for 4 s prediction horizon. However, the Social LSTM still performed significantly worse than simple velocity extrapolation at time horizons less than 6 s. The poor performances of the vanilla LSTM method and the Social LSTM method agrees with previous work on predicting basketball player trajectories conducted on a different data set [16]. As such, for most experiments, we use velocity-based extrapolation as our baseline, since it has the best performance.

**Performance metrics** We report three metrics. First, the $L_2$ distance (ft) between predicted trajectories and the ground truth, averaged over each time step for each agent. Second is the maximum distance between the prediction and ground truth for an agent trajectory, averaged over all agent trajectories. Last is the miss rate, calculated as the fraction of time the $L_2$ error exceeds 3 ft.

### 5.1 What information gives us the best prediction?

In our proposed problem, there are four sources of information with the potential to improve prediction: i) the trajectory history $X_{\mathcal{A}}^{[t_0,t_q]}$ of all agents, ii) the future motion $X_{\mathcal{K}}^{(t_q,t_f]}$ of the players not predicted, *i.e.*, context, iii) the player/team identities, *i.e.*, personalization and iv) the agent alignment. The observed trajectory history serves as the input to the model and is fixed to 4 s. The final 1 second of trajectory history of the players we predict, the context, and the identity are treated as conditionals (Fig. 3), whereas the agent alignment enables efficient trajectory encoding. For this section (Table 1), we only predict the offense, which avoids conflating the effect of agent type with the effect of the information sources. We also fix the prediction horizon at 4 s.

To understand the influence of alignment alone, we compare the result of the baseline VAE with random versus role aligned agents. In the absence of the alignment the VAE has moderate performance, outperforming baselines. For example, in the first row of Fig. 4 the VAE captures co-movement of players (red and purple) that velocity-based extrapolation does not. However, the VAE does not capture the two agents crossing.

To understand the influence of each conditional, we randomly order the input trajectories and perform a set of ablation studies using a variety of conditions.

| **Method** | **Alignment** | **Conditional** | | | **Error** (Offense, 4 s in future) | | |
|---|---|---|---|---|---|---|---|
| | | History | Context | Identity | Avg dist [ft] / (Top-5) | Max dist | Miss rate |
| Velocity | - | - | - | - | 7.77 | 14.45 | 82.18 |
| Retrieval | role | - | - | - | 11.41 / (8.80) | 28.57 | 86.77 |
| VAE | random | - | - | - | 7.10 | 19.24 | 74.90 |
| VAE | role | - | - | - | 6.85 | 18.84 | 72.78 |
| CVAE | random | 1 s | none | none | 6.90 | 18.98 | 73.83 |
| CVAE | random | none | encoded | none | 6.97 | 18.46 | 75.29 |
| CVAE | random | none | none | team | 7.05 | 19.25 | 74.15 |
| CVAE | random | none | none | player | 7.02 | 19.17 | 75.15 |
| CVAE | random | none | encoded | team | 6.98 | 18.46 | 75.65 |
| CVAE | random | 1 s | none | team | 6.91 | 18.95 | 74.18 |
| CVAE | random | 1 s | encoded | none | 6.73 | 18.11 | 74.64 |
| CVAE | random | 1 s | encoded | team | 6.76 | 18.15 | 74.97 |
| CVAE | random | 1 s | encoded | player | 6.64 | 18.00 | 74.29 |
| CVAE | position | 1 s | encoded | team | 6.09 | 16.87 | 70.37 |
| **CVAE** | **role** | **1 s** | **encoded** | **none** | 5.81 | **16.41** | 66.67 |
| **CVAE** | **role** | **1 s** | **encoded** | **team** | **5.80** | 16.45 | **66.39** |
| CVAE | role | 1 s | encoded | player | 5.96 | 17.03 | 67.07 |

Table 1: **Offense prediction error for 4 s history and prediction horizon**. We test three different trajectory alignments i) random, ii) canonical position, and iii) role. We also test 3 conditionals: a) the previous one second of player motions (history), b) the next 4 s of the defensive motions (context), and c) one-hot encoded player or team (identity). The miss rate is calculated with threshold 3 feet.

We apply each conditional separately to compare their individual effects on performance, including comparing the use of team versus player identity.

Interestingly, the VAE and the CVAE using a single conditional perform similarly. However, if we combine conditionals, we create an even stronger co-movement signal, *e.g.*, red and purple players in the first row in Fig. 4. Still, with all the conditionals and random agent ordering, we fail to get the crossing of the trajectories.

When we both align and condition, we are able to correctly predict tracks crossing (red and purple players first row in Fig. 4d). In particular, we see the greatest improvement in our prediction by including the context, history, and team identity (bold in Table 1). These results imply that alignment, context, and history contain complementary information. Though alignment and conditioning improve our predictions, we struggle to predict sudden changes in movement (red player in row 3 of Fig. 4d), and stationary players (green players in row 1 and blue player in row 3 of Fig. 4d).

The modest improvements found by including team identity vanish when we use multi-template tree-based role alignment; implying that the alignment contains the added information provided by conditioning on the team identity. In other words, the clusters in latent space that the variational module finds with canonical alignment are team sensitive. This sensitivity to the team implies that certain teams perform certain collective motions. However, after tree-alignment,

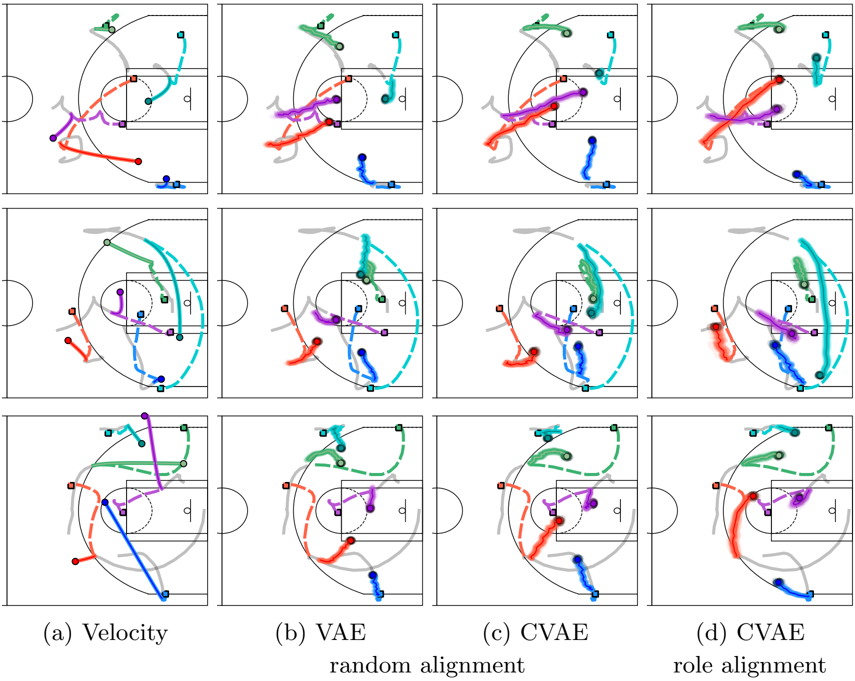


Fig. 4: **Offense player predictions.** Given a 4 s trajectory history (gray) for all players (defense not pictured), we predict (solid lines) the next 4 s of offense player motion. Dashed lines are ground truth. Each row represents the same play, and each trajectory color corresponds to a player. The color intensity is proportional to the likelihood. **Column a)** velocity-based extrapolation. **Column b)** VAE with random trajectory alignment. **Column c)** CVAE with random trajectory alignment and all conditionals (player ID). **Column d)** adding role alignment to the CVAE (team ID).

this vanishes, implying that the clusters found given optimal alignment exist below the level of player combinations.

### 5.2 How many and which agents can we predict?

To evaluate how many and which agents we can predict, we split our prediction tasks into i) exclusively predicting all 5 offense agents (Section 5.1), ii) exclusively predicting all 5 defense agents, and iii) predicting a mixture of offense and defense agents, from one of each (mix 1v1) to all 10 agents (mix 5v5).

**Defense only** Predicting defense is more straightforward than our other tasks because the defense reacts to the offense's play. Thus, the offense motion encodes much of the information about the defense motion. This is supported by the overall improvement in prediction for the defense as compared to the offense (Table 2a and b). The trends in the effect of conditionals and alignment are

| Method-Align-Pl. | Personnel | Error: Avg dist [ft] | | |
|---|---|---|---|---|
| | | 1 s | 4 s | 8 s |
| Velocity | offense | 7.74 | 7.72 | 7.74 |
| CVAE-rand-ID | offense | 7.06 | **6.64** | 6.86 |
| CVAE-role-none | offense | 6.04 | **5.81** | 6.21 |
| CVAE-role-team | offense | 6.05 | **5.80** | 6.16 |
| CVAE-role-team | defense | 4.23 | **4.10** | 4.31 |
| CVAE-role-team | mix 5v5 | 5.75 | 5.74 | 5.76 |

(a) observed history

| Method-Align-Pl. | Personnel | Error: Avg dist [ft] (4 s history) | | | | |
|---|---|---|---|---|---|---|
| | | 1 s | 2 s | 4 s | 6 s | 8 s |
| Velocity | offense | **1.93** | 4.10 | 7.72 | 11.50 | 24.02 |
| CVAE-rand-ID | offense | 2.66 | 4.23 | 6.64 | 8.14 | 9.41 |
| CVAE-role-none | offense | 2.38 | 4.00 | 5.81 | 7.07 | 8.28 |
| CVAE-role-team | offense | 2.35 | **3.95** | **5.80** | **7.08** | **8.07** |
| CVAE-role-team | defense | 2.08 | 3.01 | 4.10 | 4.98 | 5.85 |
| Vanilla LSTM | mix 5v5 | 10.44 | 18.29 | 25.36 | 28.07 | 29.56 |
| Social LSTM | mix 5v5 | 5.23 | 11.08 | 17.95 | 20.88 | 22.38 |
| CVAE-role-team | mix 5v5 | **2.44** | **3.92** | **5.74** | **7.21** | **8.33** |

(b) prediction horizon

| Method: CVAE-role-team | |
|---|---|
| Mixture | Error: Avg dist [ft] |
| 1v1 | 4.19 |
| 2v2 | 4.88 |
| 3v3 | 5.21 |
| 4v4 | 5.28 |
| 5v5 | 5.74 |

(c) num. players

Table 2: **Prediction error ablation.** a) We vary the observed history for a 4 s prediction, and observe that the optimal trajectory history is 4 s, though marginally so. b) We vary the prediction horizon given a 4 s observed history, and observe that the prediction error monotonically increases as a function of time horizon. c) We vary the number of players to predict for a 4 s horizon given a 4 s history, and observe an increase in average prediction error as we increase the number of agents per team from 1 to 5. For all experiments, we conditioned on the previous 1 s, the future motion of all agents not predicted, and the selected player or team identities. All errors are in feet.

similar to the offense-only prediction results, indicating the value of information is similar regardless of adversary predicted. Therefore, we use role alignment and conditionals history, context, and team identity in subsequent experiments.

**Mixed offense and defense** Our most challenging prediction task is to simultaneous predict the motion of offense and defense. This is akin to asking: can we predict the motion of unobserved agents given the motion of the remaining seen agents? In the most general case of trying to predict all players, we found that the prediction performance splits the difference between the prediction of the offense and defense alone (Table 2a).

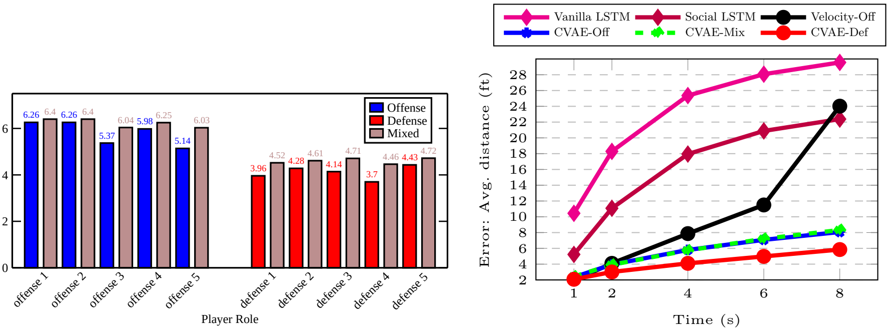
Next, we investigated how many agents per team we could predict over a 4 s time horizon, given a 4 s history (Table 2c). Surprisingly, we found relatively little performance degradation when predicting the motion of all ten players (5v5) versus one player each (1v1) on offense and defense (5.7 ft vs 4.2 ft). In the case of predicting all ten agents, the only conditionals are the player or team identities and the previous 1 s of history. The input is the 4 s trajectory history.

### 5.3 How does personnel influence prediction?

Since alignment improved our prediction results, we investigated the per-role prediction error (Fig. 5a) to uncover whether some roles are easier to predict than others. We found $\sim$ 16% difference in the per-role prediction error for predicting offense compared to defense only. However, the per role variation does not hold when predicting a mixture of agents, in which case the prediction error of all agents increases.

### 5.4 How much history do we need?

Next, we tested the effect of the observed trajectory duration on prediction performance, that is how the history length influences predictions. The conditionals



(a) **Per-role error (avg distance, ft).** (b) **Error vs prediction horizon.**

Fig. 5: **Prediction error ablation.** For all experiments, we provided 4 s of history and conditioned on the previous 1 second and the future of all agents not predicted. a) We evaluate the per-role prediction error for a 4 s prediction horizon, given a 4 s observed history. Defense is easier to predict than offense, and although mixed (2v2) appears to have better overall prediction than offense, per-role it is slightly worse, which makes sense because it's a harder task. b) We visualize the prediction errors as a function of horizon, given a 4 s observed trajectory history. The baselines are velocity (for offense only), and vanilla LSTM and Social LSTM (for all 10 agents), which we compare with our best method run on offense and defense only, as well as the mixture of all 10 agents. The precise values are reported in Table 2b.

are the previous 1 s of the agents we are predicting, the future motion of players we are not predicting, and the team or player identity. We varied the observed history from 1-8 s and predicted the subsequent 4 s. As before, the defense is the easiest to predict, and multi-template role alignment with team identity provides the best prediction performance (Table 2a). We find 4 s of history is barely optimal, either because the player motions decorrelate at this time scale, or our encoder architecture cannot recover correlations at longer timescales.

### 5.5 How far can we predict?

To evaluate how far in the future we can predict, we provided 4 s of history of all player motions and predicted out to at most 8 s. Additionally, we provided the last 1 s of player motions and the future of the un-predicted agents as a conditional. In Fig. 6 we can clearly see that as the we to underestimate the curvature of motions (cyan in example 1, $\mathcal{T} = 6\ s$), or underestimate the complexity of motion (purple in row 1, $\mathcal{T} = 6\ s$ and red in row 2, $\mathcal{T} = 6\ s$).

As expected, the prediction error increases monotonically with the prediction time horizon (Fig. 5b), and when we include team identity, the prediction error changes less with the time horizon. Also, we see that the prediction error for the defensive is smaller than mixed offense and defense or offense alone.

We also notice that we far outperform the current state of the art prediction methods (Fig. 5b). It is remarkable that even when predicting the motion of all

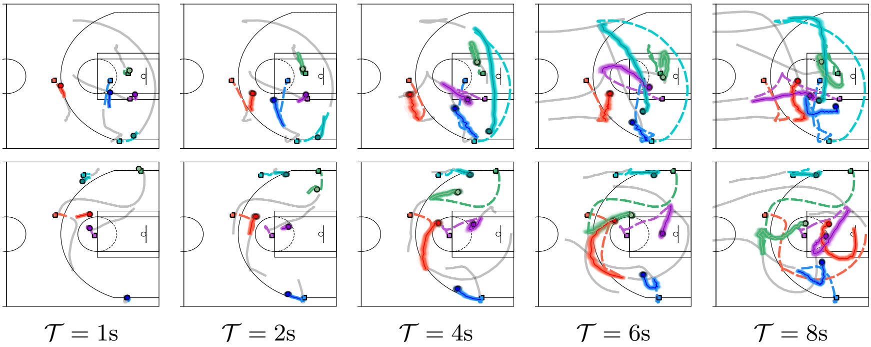


Fig. 6: **Prediction as a function of time horizon.** We input the previous 4 s of every agent's motion (grey), and predict the offense player trajectories over horizon $\mathcal{T}$ s. The conditionals are the future motion of the defense (not shown), the final one second of offense history, and team identity. Each row represents a different example, and each color represents the player tree-based role. Dashed lines are the ground truth.

agents that our performance is three times as good as the Social LSTM (for 4 s time horizon). Again, it is important to note that the performance of the LSTM baselines agrees with previous results on a similar dataset [16]. Lastly, we note that the prediction of player trajectories presented by Shan et al. [16] which uses far more information, specifically the egocentric appearance of all players produces a per player average error of 11.8 ft (3.6 m). Though not directly comparable, this shows the power of our proposed generative method: with less information, our method produces noticeably better results.

## 6 Conclusion

We have shown that a generative method based on conditional variational autoencoder (CVAE) is three times as accurate as the state of the art recurrent frameworks for the task of predicting player trajectories in an adversarial team game. Furthermore, these predictions improve by conditioning the predictions on the history and the context, *i.e.*, the motion of agents not predicted and their identity. Also, where available, further improvement in the quality of prediction can be found by providing multi-template aligned data. By aligning and conditioning of context and history, we can produce remarkably accurate, context-specific predictions without the need for ranking and refinement modules. We also found that our predictions were sensitive to the player role, as determined during alignment. However, we did not find any additional improvement in prediction when providing the player identity alone. The sensitivity to the player role, but not identity implies that role contains the information held in identity alone. Therefore, more fine-grained personalization may require additional player data, such as weight, height, age, minutes played.

## References

1. Lee, N., Choi, W., Vernaza, P., Choy, C., Torr, P., Chandraker, M.: DESIRE: Distance Future Prediction in Dynamic Scenes with Interacting Agents. (2017)
2. Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S.: Social LSTM: Human Trajectory Prediction in Crowded Spaces. (2016)
3. Jain, A., Singh, A., Koppula, H., Soh, S., Saxena, A.: Recurrent Neural Networks for Driver Activity Anticipation via Sensory-Fusion Architecture. (2016)
4. Akhter, I., Simon, T., Khan, S., Matthews, I., Sheikh, Y.: Bilinear spatiotemporal basis models. ACM Transactions on Graphics (TOG) (2012)
5. Lucey, P., Bialkowski, A., Carr, P., Morgan, S., Matthews, I., Sheikh, Y.: Representing and Discovering Adversarial Team Behaviors using Player Roles. (2013)
6. Insafutdinov, E., Andriluka, M., Pishchulin, L., Tang, S., Levinkov, E., Andres, B., Schiele, B.: ArtTrack: Articulated Multi-Person Tracking in the Wild. (2017)
7. Le, H., Yue, Y., Carr, P., Lucey, P.: Coordinated Multi-Agent Imitation Learning. (2017)
8. Yamaguchi, K., Berg, A., Ortiz, L., Berg, T.: Who are you with and where are you going? (2011)
9. Butt, A., Collins, R.: Multi-target Tracking by Lagrangian Relaxation to Min-Cost Network Flow. (2013)
10. Wang, S., Fowlkes, C.: Learning Optimal Parameters for Multi-Target Tracking. (2016)
11. Maksai, A., Wang, X., Fua, P.: What Players do with the Ball: A Physically Constrained Interaction Modeling. (2016)
12. Kim, K., Grundmann, M., Shamir, A., Matthews, I., Hodgins, J., Essa, I.: Motion Fields to PRedict Play Evolution in Dynamic Sports Scenes. (2010)
13. Chen, J., Le, H., Carr, P., Yue, Y., Little, J.: Learning Online Smooth Predictors for Realtime Camera Planning using Recurrent Decision Trees. (2016)
14. Zheng, S., Yue, Y., Lucey, P.: Generating Long-term Trajectories Using Deep Hierarchical Networks. (2016)
15. Felsen, P., Agrawal, P., Malik, J.: What will Happen Next?: Forecasting Player Moves in Sports Videos. (2017)
16. Su, S., Hong, J.P., Shi, J., Park, H.S.: Social behavior prediction from first person videos. CoRR **abs/1611.09464** (2016)
17. Koren, Y., Bell, R., Volinksy, C.: Matrix factorization techniques for recommender systems. Computer **42**(8) (2009)
18. Deng, Z., Navarathna, R., Carr, P., Mandt, S., Yue, Y., Matthews, I., Mori, G.: Factorized Variational Autoencoders for Modeling Audience Reactions to Movies. (2017)
19. Charles, J., Pfister, T., Magee, D., Hogg, D., Zisserman, A.: Personalizing Human Video Pose Estimation. (2016)
20. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)
21. Gregor, K., Danihelka, I., Graves, A., Wierstra, D.: DRAW: A recurrent neural network for image generation. CoRR **abs/1502.04623** (2015)
22. Bowman, S.R., Vilnis, L., Vinyals, O., Dai, A.M., Józefowicz, R., Bengio, S.: Generating sentences from a continuous space. CoRR **abs/1511.06349** (2015)
23. Kingma, D., Mohamed, S., Rezende, D., Welling, M.: Semi-Supervised Learning with Deep Generative Models. (2014)

24. Sohn, K., Lee, H., Yan, X.: Learning Structured Output Representation using Deep Conditional Generative Models. (2015)
25. van den Oord, A., Kalchbrenner, N., Kavukcuoglu, K.: Pixel recurrent neural networks. CoRR **abs/1601.06759** (2016)
26. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. (June 2016)
27. Walker, J., Doersch, C., Gupta, A., Hebert, M.: An uncertain future: Forecasting from static images using variational autoencoders. CoRR **abs/1606.07873** (2016)
28. Sha, L., Lucey, P., Zheng, S., Kim, T., Yue, Y., Sridharan, S.: Fine-Grained Retrieval of Sports Plays using Tree-Based Alignment of Trajectories. (2017)