Learning Contextualized Player Representations with A Variational Hierarchical Encoder

Guiliang Liu and **Oliver Schulte**
Simon Fraser University, Burnaby, Canada
gla68@sfu.ca, oschulte@cs.sfu.ca

Abstract

Many recent papers have proposed advanced machine learning methods to compute the expected impact of player actions. These values, however, are rarely specific to individual players, and thus fail to model a player's influence on action outcomes. To overcome this limitation, we generate a contextualized representation for each player by a model proposed in this paper: Variational Hierarchical Encoder with Recurrence (VHER). VHER generates a latent player representation to predict which player is currently acting given the match context (current observation and game history). The encoder constructs a context-specific shared prior over player representations, which induces a shrinkage effect for the posterior representations. A player embedding is generated by sampling from the player's posterior distribution. To validate our VHER, we use the learned player embedding for downstream prediction tasks. Experimental results show the leading performance of VHER in the task of (1) identifying the acting player and (2) predicting the player's expected goals.

Introduction

With the advancement of high-frequency optical tracking and object detection systems, more and larger event stream datasets for sports matches have become available. There is an increasing opportunity for applying advanced machine learning to model the complex sports dynamics. Many recent works (Liu and Schulte 2018; Decroos et al. 2019; Fernández et al. 2019) have proposed to estimate the expected team success following a player's actions. These expected values support many downstream applications, such as predicting game outcomes or evaluating player performance. However, when estimating the expected values, previous works often overlook the player-specific features (e.g. scoring ability) and assign the same values to actions performed by different players. Neglecting differences among individual players compromises the model performance.

Some previous works have explored the approach to incorporating player information into modeling. Probably the most straightforward approach is to apply a one-hot vector recording the player identity (pid) and train the neural model to dynamically learn the correlations between pids and game context (Le et al. 2017). Despite its simplicity, the one-hot representation is not informative enough for a neural network to adequately model the correlation between individuals and success. As evidence, our experiment shows very limited improvement when we directly complement the input space with pids. A recent work (Ganguly and Frank 2018) proposed to learn player embeddings by training a neural network encoder to perform a secondary prediction task: given the current game context, predict the pids of all on-court players. They extracted the middle layer from the trained encoder and used it as a player embedding to facilitate the training of other primary tasks. However, the predictive accuracy on the secondary task was low. A problem with training a neural net as a deterministic regression model is that the player presence has a multi-modal distribution with several almost equally likely outcomes. A strength of variational auto-encoders is that they produce a distribution over outcomes that accommodates multiple modes.

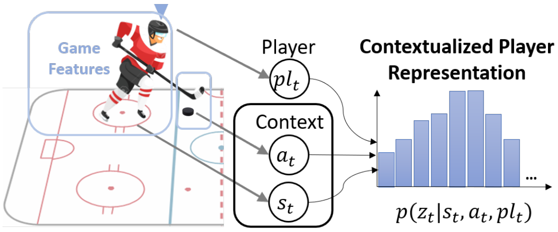


Figure 1: An example of our contextualized player representation, from which we sample the player embeddings.

To overcome the limitation, in this work, we build a Variational Hierarchical Encoder with Recurrence (VHER). VHER combines a Bayesian hierarchical model (Kruschke 2014) and variational inference to embed the player information in latent variables under different game contexts. The hierarchical model is trained to identify the acting player, where we sample some latent variables from a context-specific prior and compute N (the number of players) Bernoulli distributions to model the presence of each player. The Bernoulli parameters are then normalized for a categorical distribution to predict the current on-the-puck player. We then apply variational inference to learn the model parameters. Compared to Monte Carlo Markov Chain (MCMC) and grid approximation, variational inference can generalize well to complex parameter spaces and significantly reduce the learning time under neural network implementation (Blei, Kucukelbir, and McAuliffe 2016). During the inference, the posteriors for each player are encouraged to shrink toward the mode of the context-specific prior, which substantially reduces the training variance. This shrinkage effect in our hierarchical model naturally formalizes the idea that "similar players appear in a similar context". We also apply the posterior distribution for each player to predict the current on-the-puck player, which encourages the diversity of the posterior player representations. To demonstrate the effectiveness of our player embedding, we apply them to the secondary (embedding) task of identifying the acting player and the external validation task of predicting the expected goal. Experimental results show the improvement of model performance with our player embedding.

Related Works

In this section, we introduce the previous works that are most related to our model.

Variational Auto-Encoder

Variational Auto-Encoder (VAE) has achieved promising performance in recovering multimodal distributions and in generating many kinds of complicated data, including handwriting, faces (Kingma and Welling 2013), images (Gregor et al. 2015) and player actions (Mehrasa et al. 2019). VAE applies a set of latent variables $\mathbf{z}$ to capture the variations of observed variables $\boldsymbol{o}$. During the generative process, the prior of $\mathbf{z}$ is generally chosen to be a simple Gaussian distribution. VAE models the likelihood function $p(\boldsymbol{o}|\mathbf{z})$ with a decoder (usually implemented as a Gaussian or Bernoulli Multi-Layer Perceptron (MLP) (Kingma and Welling 2013), which applies a highly non-linear mapping from $\mathbf{z}$ to $\boldsymbol{o}$.

The non-linearity in complicated likelihood function $p(\boldsymbol{o}|\mathbf{z})$ leads to the intractable inference of the posterior $p(\mathbf{z}|\boldsymbol{o})$. Instead, VAE approximates the true posterior with a recognition model (decoder) $q(\mathbf{z}|\boldsymbol{o})$, which is usually defined as a Gaussian as $\mathbf{z} \sim \mathcal{N}[\boldsymbol{\mu}, diag(\boldsymbol{\sigma}^2)]$ ($\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are computed with observed variables $\boldsymbol{o}$). Parameters of both decoder and encoder are optimized by maximizing a lower bound of the marginal likelihood of observation $p(x)$:

$$\mathcal{L}(p(\boldsymbol{o})) = -KL(q(\mathbf{z}|\boldsymbol{o})||p(\mathbf{z})) + \mathbb{E}_{q(\mathbf{z}|\boldsymbol{o})}\Big[\log p(\boldsymbol{o}|\mathbf{z})\Big] \tag{1}$$

(Kingma and Welling 2013) introduced an alternative method for generating samples from $q(\mathbf{z}|pl)$ and described a reparameterizing trick for VAE. By rewriting:

$$\mathbb{E}\Big[\log(p(\boldsymbol{o}|\mathbf{z}))\Big] = \mathbb{E}\Big[\log p(\boldsymbol{o}|\mathbf{z} = \boldsymbol{\mu} + \boldsymbol{\sigma} \odot \boldsymbol{\epsilon})\Big] \tag{2}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(0, 1)$, reparameterizing makes the estimations of the expectation with respect to $q(\mathbf{z}|\boldsymbol{o})$ differentiable.

To handle sequential data, Chung et al. combined the latent variables with a recurrent model. The proposed Variational Recurrent Neural Network (VRNN) includes a VAE at every time step t. The object function is a timestep-wise variational lower bound:

$$\sum_{t=1}^{T}\Big[-KL(q(\mathbf{z}_t|\boldsymbol{o}_{\leq t}, \mathbf{z}_{<t})||p(\mathbf{z}_t|\boldsymbol{o}_{<t}, \mathbf{z}_{<t})) + \log p(\boldsymbol{o}_t|\mathbf{z}_{\leq t}, \boldsymbol{o}_{<t})\Big] \tag{3}$$

Hierarchical Models in Sports Analytics

Many previous works (Gelman 2006; Davis, Perera, and Swartz 2015) have built a multi-level hierarchical model and estimated the parameters of the posterior with Bayesian inference. The Bayesian inference naturally incorporates the shrinkage effect into estimating model parameters. The shrinkage effect pulls the estimates of low-level parameters closer together than they would be if there were not a higher-level distribution, and generally, shrinkage in hierarchical models encourages lower-level parameters to shift toward the modes of the higher-level distribution, which can significantly reduce the variance of estimation. Similar hierarchical models have many applications in sports analytics, for example, Kruschke built a hierarchical model to estimate the batting abilities for individual baseball players. They sampled the parameters of a likelihood function (modeling players' batting ability by the probabilities of hitting the ball) from a prior conditioning on player position. Accordingly, for players in the same position, despite the difference in performance, shrinkage toward the position-specific mode leaves the posterior distribution of their difference being nearly zero. Such a shrinkage effect can facilitate our player embedding model. Given the observation that similar players are likely to appear in similar contexts, our model is trained to dynamically encourage the embedding to shift toward the mode of a context-specific prior.

Contextualized Embedding

As a promising technique of incorporating background knowledge into the object modeling, contextualized embedding have been extensively studied under the topics related to Natural Language Processing, for example, a recent work (Akbik, Blythe, and Vollgraf 2018) proposed a contextual string embedding. The embedding model contextualizes words by their surrounding text. Correspondingly, the same word will have different embeddings depending on its contextual use. A more recent work (Peters et al. 2018) computed the contextualized embedding to model complex characteristics of word use (e.g., syntax and semantics) and extended the application of embeddings across linguistic contexts. They showed the embeddings can be easily added to existing models and significantly improved the state-of-the-art across six challenging NLP problems. To utilize the advantage of contextualized embedding, in this work, we compute the embeddings for NHL players conditioning on different game contexts (including the current observation and play history). Our results also demonstrate the benefit of applying contextualized embeddings in identifying pids and predicting expected goals.

Type	Name	Range
Spatial Features	X Coordinate of Puck	[-100, 100]
	Y Coordinate of Puck	[-42.5, 42.5]
	Velocity of Puck	$(-\infty, +\infty)$
	Angle between the puck and the goal	$[-3.14, 3.14]$
Temporal Features	Game Time Remain	$(-\infty, 3{,}600]$
	Event Duration	$(0, +\infty)$
In-Game Features	Score Differential	$(-\infty, +\infty)$
	Manpower Situation	{EV, SH, PP}
	Home or Away Team	{Home, Away}
	Action Outcome	{successful, failure}
Pre-game Statistics	Box Score	$(-\infty, +\infty)$

Table 1: Complete Feature List. We have experimented with the option of incorporating players' box scores into our embedding. The box score includes players' pre-games cumulative statistics: The total number of goals, assists, points, penalty minutes, and played games from the beginning of the 2017-18 NHL season to the beginning of the current game.

Modeling Play Dynamics

Dataset

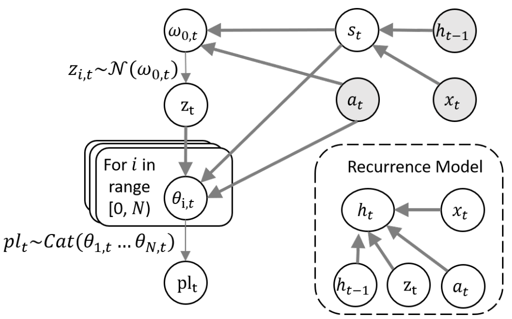
We utilize a dataset constructed by SPORTLOGiQ with computer vision techniques. The data provide information about **game events** and **player actions** for the entire 2018-2019 NHL (largest professional ice hockey league) season, which contains over 4 million events, covering 31 teams, 1,196 games and 1,003 players. The data track events around the puck, and record the identity and actions of the player in possession, with space and time stamps, as well as features of the game context. The table utilizes adjusted spatial coordinates where negative numbers refer to the defensive zone of the acting player, positive numbers to his offensive zone. Adjusted X-coordinates run from -100 to +100, Y-coordinates from 42.5 to -42.5, where the origin is at the ice center as in Figure 1. We augment the data with derived features and list the complete feature set in Table 1.

Contextual Variables for NHL Players

In the SPORTLOGiQ dataset, the play dynamics is captured by contextual variables as follows:

- The **action** $\mathbf{a}_t$ records the movements of players who control the puck. Our model applies a discrete action vector with the one-hot representation.
- The **environment variables** $\mathbf{x}_t$ describes the game environment where the action is performed. We represent it as a feature vector specifying a value of the features listed in Table 1 at a discrete-time step t.

In each game, we consider event data of the form $\mathbf{x}_0, pl_0, \mathbf{a}_0, \mathbf{x}_1, pl_1, \mathbf{a}_1, \ldots, \mathbf{x}_t, pl_t, \mathbf{a}_t, \ldots$: at time t, after observing environment $\mathbf{x}_t$, player pl_t takes a turn (possesses the puck) and chooses an action $\mathbf{a}_t$. The observations for a given player i form a set of triples $(pl_t = i, \mathbf{s}_t, \mathbf{a}_t)$, where to alleviate the partial observability in the dataset, the game state $\mathbf{s}_t$ includes the game history $\mathbf{s}_t \equiv (\mathbf{x}_t, \mathbf{a}_{t-1}, \mathbf{x}_{t-1}, \ldots, \mathbf{x}_0)$ (Liu and Schulte 2018; Hausknecht and Stone 2015). Each triple summarizes the observed player actions and the game environment, with a joint distribution $p(pl_t, \mathbf{s}_t, \mathbf{a}_t)$. This distribution can be factored into two components:



$\omega_{0,t}$
s_t
h_{t-1}
$z_{i,t}{\sim}\mathcal{N}(\omega_{0,t})$
z_t
a_t
x_t
For i in range [0, N)
$\theta_{i,t}$
$pl_t{\sim}Cat(\theta_{1,t} \ldots \theta_{N,t})$
pl_t
Recurrence Model
h_t
x_t
h_{t-1}
z_t
a_t

Figure 2: Graphical illustration of our hierarchical model. Thick (bold) line indicates logical function while thin line denotes stochastic dependence. The shaded nodes are given during generation. Our model applies hidden states $\mathbf{h}_{t-1}$ of a LSTM cell to capture the temporal dependence of a series of previously observed environment features and actions, so we represent the state as $\mathbf{s}_t \equiv (\mathbf{x}_t, \mathbf{h}_{t-1})$ and update the hidden states by $\mathbf{h}_t = f(\mathbf{x}_t, \mathbf{a}_t, \mathbf{z}_t, \mathbf{h}_{t-1})$. N is the number of embedded players.

$$p(pl_t, \mathbf{s}_t, \mathbf{a}_t) = p(pl_t|\mathbf{s}_t, \mathbf{a}_t)p(\mathbf{s}_t, \mathbf{a}_t). \qquad (4)$$

where the player-independent component $p(\mathbf{s}_t, \mathbf{a}_t)$ represents the game context (observed action and state at t) and $p(pl_t|\mathbf{s}_t, \mathbf{a}_t)$ models the dependency between the observed game context and the acting player pl_t. The second component describes a player tendency to act under different game states, which makes it an appropriate target for learning contextualized embedding for each player.

Contextualized Player Representation

We introduce our novel Variational Hierarchical Encoder with Recurrence (VHER) which combines a generative Bayesian hierarchical model with variational inference for obtaining a contextualized player Representation.

Bayesian Hierarchical Model

To model the player-related component $p(pl_t|\mathbf{s}_t, \mathbf{a}_t)$, we build a *Bayesian hierarchical model*, shown in Figure 2, where parameters are assigned distributions like random variables (McCallum et al. 1998; Kruschke 2014). Our hierarchical model splits $p(pl_t|\mathbf{s}_t, \mathbf{a}_t)$ into different components:

$$\sum_{\boldsymbol{\theta}_{i,t}} \big[p(pl_t|\boldsymbol{\theta}_{i,t})p(\boldsymbol{\theta}_{i,t}|\mathbf{z}_t, \mathbf{s}_t, \mathbf{a}_t)\big]p(\mathbf{z}_t|\boldsymbol{\omega}_{t,0})p(\boldsymbol{\omega}_{t,0}|\mathbf{s}_t, \mathbf{a}_t)$$

Figure 2 presents a graphical illustration of our hierarchical model. Conditioning on game context (state-action pair $\mathbf{s}_t, \mathbf{a}_t$), the prior on the player embeddings (the latent random variables) is represented by a Gaussian distribution:

$$\boldsymbol{\omega}_{0,t} := \psi^{prior}[\psi^c(\mathbf{s}_t, \mathbf{a}_t)] \qquad (5)$$

$$\mathbf{z}_t \sim p(\mathbf{z}_t|\mathbf{s}_t, \mathbf{a}_t) \equiv \mathcal{N}(\boldsymbol{\omega}_{0,t}) \qquad (6)$$

where $\boldsymbol{\omega}_{0,t} \equiv [\boldsymbol{\mu}_{t,0}, diag(\boldsymbol{\sigma}_{t,0})]$ denotes the parameters of the context-specific Gaussian prior. A neural network is trained to compute the parameter estimates by implementing a context function ψ^{c} that extracts context features and a prior function ψ^{prior} to compute $\boldsymbol{\omega}_{0,t}$ from the extracted features.

Given the sample latent variables $\mathbf{z}_t$ and game context $(\mathbf{s}_t.\mathbf{a}_t)$, our model generates the label of the on-the-puck (possessing the puck) player as follows:

$$\boldsymbol{\theta}_{i,t} := \boldsymbol{\sigma}\{\psi^{dec}[\psi^{z}(\mathbf{z}_t), \psi^{c}(\mathbf{s}_t, \mathbf{a}_t)]\} \tag{7}$$

$$pl_t|\mathbf{z}_t, \mathbf{s}_t, \mathbf{a}_t \sim Categorical[\boldsymbol{\phi}(\boldsymbol{\theta}_{1,t}, \ldots, \boldsymbol{\theta}_{N,t})] \tag{8}$$

where $\boldsymbol{\theta}_{i,t}$ denotes the parameters of Bernoulli distributions to model the presence of player $pl_{i,t}$. These parameters are computed as follows: (1) A neural network implements the player embedding function ψ^{z} to extract features from the player representations $\mathbf{z}_t$ (2) another neural network implements the context embedding function $\psi^{c}(\mathbf{s}_t, \mathbf{a}_t)$ to extract features from the game context. (3) The extracted features are input to a decoder function ψ^{dec}, whose outputs are mapped to [0,1] by a sigmoid function $\boldsymbol{\sigma}$ to compute Bernoulli parameters for each player i. (4) The softmax function $\boldsymbol{\phi}$ normalizes the Bernoulli parameters to obtain a categorical distribution over players acting at time t.

Variational Inference

We apply variational inference to derive an objective function for estimating the parameters of our hierarchical model. The inference is similar to that of Variational Auto Encoder (VAE) (Kingma and Welling 2013), because both models utilize a prior and approximate posterior on the latent variables to define an approximate log-likelihood function for the observed data. The main difference is that our hierarchical model conditions on the game context. In particular, the latent variable prior is learned to be a function of the game context, rather than a context-independent standard distribution.

Figure 3 illustrates the inference process of our model. After observing the pl_t, the approximate posterior on a player embedding follows the equation:

$$\boldsymbol{\omega}_{i,t} := \psi^{enc}[\psi^{pl}(pl_t), \psi^{c}(\mathbf{s}_t, \mathbf{a}_t)] \tag{9}$$

$$\mathbf{z}_t \sim q(\mathbf{z}_t|pl_t = i, \mathbf{s}_t, \mathbf{a}_t) \equiv \mathcal{N}(\boldsymbol{\omega}_{i,t}) \tag{10}$$

where we apply neural networks to implement (1) a observation function ψ^{pl} that extracts features from pl_t (represented as an one-hot vector of N dimensions) (2) a context function ψ^{c} that extract features from game context $(\mathbf{s}_t, \mathbf{a}_t)$, and (3) an encoding function ψ^{enc} generates the parameters $\boldsymbol{\omega}_{i,t} \equiv [\boldsymbol{\mu}_{i,t}, diag(\boldsymbol{\sigma}_{i,t})]$ of an approximate Gaussian posterior, with which we sample embeddings of individual players. The posterior $q(\mathbf{z}_t|pl_t = i, \mathbf{s}_t, \mathbf{a}_t)$ is used as a representation for player i, with which we can construct a context-dependent embedding vector. This real-valued vector can replace the one-hot player representation and facilitate downstream application such as expected goal or game outcome prediction.

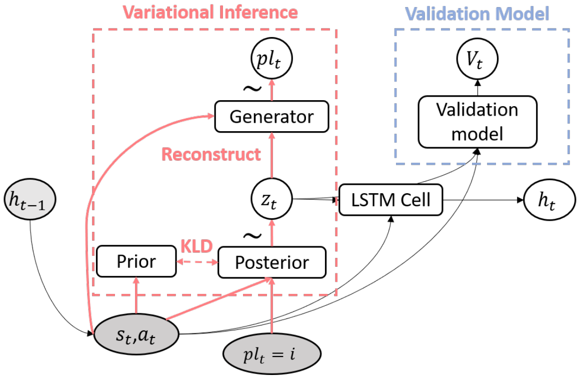


Figure 3: Learning the player representations and applying them to the validation model. The prior and the posterior respectively represent the Gaussian Prior and the approximate posterior of latent variables, with which the generator reconstructs pl_t. Red arrows indicate the process of variational inference, and the shaded nodes are given during training.

Based on the time-wise variational lower bound (Chung et al. 2015), the loss function for player embedding model is

$$\sum_{t=1}^{T} \Big\{ KL\Big[q(\mathbf{z}_t|pl_t = i, \mathbf{s}_t, \mathbf{a}_t)||p(\mathbf{z}_t|\mathbf{s}_t, \mathbf{a}_t)\Big] - \tag{11}$$
$$\mathbb{E}_{\mathbf{z}_t|pl_t,\mathbf{s}_t,\mathbf{a}_t}\Big[\log p(pl_t|\mathbf{z}_t, \mathbf{s}_t, \mathbf{a}_t) - \lambda^{V}\mathcal{L}^{V}(\mathbf{z}_t, \mathbf{s}_t, \mathbf{a}_t)\Big]\Big\}$$

where we add a validation loss $\mathcal{L}^{V}$ with a parameter λ^{V} to control its scale. This loss combines the gradient of the validation model into the embedding inference and dynamically incorporates player embeddings into different applications.

Interpretation and Motivation

We provide two interpretations of the hierarchical VAE model that in our view show why this is a good model for representing the available statistical information about a player.

Predictive Model Viewed as a predictive model, our VHER solves the *re-identification task* (Lavi, Serj, and Ullah 2018): identifying which player is currently acting given a history of events. For example, a computer vision system may try to identify a player's jersey number from video footage. As Equation (4) shows, this task captures the correlations between the identity of a player, and what they do in which match contexts. The prior distribution $p(\mathbf{z}_t|\mathbf{s}_t, \mathbf{a}_t)$ can be seen as representing the probability that a randomly chosen player acts in a given game context. Our experiment also studies the predictive of performance of VHER.

Shrinkage Effect A hierarchical model is commonly used in statistics to capture similarities among a group of individuals (McCallum et al. 1998; Kruschke 2014). The intuition motivating hierarchical models is that *statistically* *similar agents are assigned similar representations*. Previous hierarchical models have been constructed for parametric models, which estimate a separate parameter vector for each player (Murphy 2012). Parametric hierarchical models achieve a *shrinkage effect* where the differences between different parameter vectors for each individual are shrunk towards a common value. Shrinkage estimators have strong statistical properties because they allow information to be transferred between the observations of different individuals. In a Bayesian hierarchical model, shrinkage is achieved by estimating the individual agent parameters using the posterior distribution over parameters drawn from a common prior for the group.

As Equation (11) shows, the VAE loss function induces a shrinkage effect by regularizing the approximate posterior for each individual player towards a common prior $p(\mathbf{z}_t|\mathbf{s}_t, \mathbf{a}_t)$. The Conditional VAE, in fact, achieves a *joint* shrinkage effect where *statistically similar agents are assigned similar representations in similar contexts.* This is because we can interpret the prior distribution $p(\mathbf{z}_t|\mathbf{s}_t, \mathbf{a}_t)$ as a joint representation of the player's action and game context: After training, state-action pairs that tend to feature the same players will be associated with similar prior distributions. In sum, we have described a non-parametric hierarchical CVAE model that generates dynamic context-aware player representations with a joint shrinkage effect for players, actions, and game states.

Empirical Evaluation

Experiment Setting

Training settings: We divide the dataset containing 1,196 games into a training set (80%), a validation set (10%) and a testing set (10%). Our model is implemented in Tensorflow. The total number of player (N) is 1,003. The dimension of the player embedding as well as the dimension of parameters in Gaussian Prior and Posterior are set to 256.

Baseline models: Our first baseline model is a Deterministic Encoder (DE) model (Ganguly and Frank 2018). It is trained as a regressor to identify the acting player and implements a deterministic projection from the game context to player embedding (a middle layer of the neural network) without modeling the prediction uncertainty (or variance). The second baseline model is a Conditional Variational Auto-Encoder (CVAE) (Kingma and Welling 2013). Compared to our VHER, CVAE conditions the player representation on current game observation, which does not incorporate the play history into embedding computation. To study the influence of player embedding, we also include an LSTM as our third baseline model. LSTM directly finishes the experimented tasks without including any player information.

Identify the Acting Player

Similar to (Ganguly and Frank 2018), to learn the player representation, our VHER is trained for a secondary task of predicting the acting (on-the-ball) player given the game context $(\mathbf{s}_t, \mathbf{a}_t)$, so this experiment studies the performance of VHER as a predictive model and compares it with the other three baseline models. To assess alternative ways of including player information, we also experiment with the options of including players' pre-game cumulative box score (see table 1) into game context.

	No Box Score		With Box Score	
Method	ACC	LL	ACC	LL
DE	10.91 %	-19.482	14.85 %	-18.590
CVAE	7.42 %	-4.294	17.21 %	-4.850
LSTM	12.41%	-3.131	64.47%	-1.718
VHER	48.00 %	-2.228	82.13%	-1.402

Table 2: Results for player identification. Acc=Accuracy and LL=Log-Likelihood.

Table 2 shows the experimental results. Predictions from DE have a significantly lower log-likelihood than the other three methods. It is because trained as a standard regression model, the DE objective minimizes the distance between a single prediction and the ground truth. This method, however, will fail if the output space is multi-modal. To handle it, variational models compute multiple isotropic Gaussian priors (Equation 6) on the latent variables which creating a disentangled representations for each player, and thus facilitates the modeling of multiple modes. It explains why CVAE manages to improve the log-likelihood. The performance of CVAE is still limited by the lack of game information. To study the influence of incorporating play history, we directly apply a LSTM to identify the acting player and achieve a better performance. Compared to the above baseline models, our VHER utilizes the advantage of both CVAE and LSTM: shrinkage toward a context-specific prior and incorporating play history. Therefore, VHER, achieves a significant increase of prediction accuracy and log-likelihood over other baseline methods. We also find including the box score will further improve the performance. This is because the box-scores provide a strong prior for identifying the acting player.

Predict the Expected Goal

In this section, we validate the player embeddings in a practical task of predicting the Expected Goal (EG). Expected Goal (EG) weights each shot by the chance of it leading to a goal. To see if the embeddings will improve the prediction accuracy of EG, we generate the player embedding $\mathbf{z}_t$ for the on-the-ball player pl_t. As Figure 3 shows, at time t, we input $\mathbf{s}_t, shot_t, \mathbf{z}_t$ to a validation model, which, similar to a classifier, is trained to generate 1 if a goal is scored after the player pl_t makes the shot and 0 otherwise.

We refer to a neural net for the validation task as the validation model. Our validation model is an LSTM that is given the play history, combined with three comparison methods of including current-player information: 1) our dense VHER embeddings, 2) directly inputting one-hot player ids (Pids), 3) no player information. To train the validation model, we utilized the game context for action shot recorded in our NHL dataset and supervise the training by whether the shot will lead to a goal in real games. However, considering that most of the shots will not score any goals, the training data is highly imbalanced. We handle the imbalance with a resampling method (Good 2006) so that equal numbers of success and failed shots are included in the training dataset.

Model	Metric			
Player Info	P	R	F1	LL
N/A	0.144	0.808	0.245	-0.641
Pids	0.103	0.691	0.179	-0.573
DE	0.206	0.903	0.335	-2.756
CVAE	0.252	0.939	0.397	-2.589
VHER	0.624	0.846	0.718	-0.281

Table 3: Results for predicting the Expected Goal. The evaluation metrics include Precision (P), Recall (R), F1-score and Log-Likelihood (LL).

Table 3 shows the results on the testing set. Without including any player information, predictions from a LSTM model have large recall but very limited precision. Thus the model prefers labeling many shots as goals, but most of the predictions are incorrect. This problem has not been alleviated after adding the pids to the input space, which shows it is hard to utilize the player information with only a sparse one-hot label. Providing more useful player information, DE deterministically maps the player information into a dense player embedding vector and CVAE further improves the embedding with the latent variables. A common problem for the above embedding methods is the absence of play history during training. To overcome this limitation, our VHER applies a recurrent model to fit the play history and substantially improves the precision of predictions.

Discussion

In this section, we discuss the potential applications of variational player embeddings and the possibility of generalizing them to other sports.

Applications of Variational Player Embeddings

Variational player embeddings can potentially be applied for many tasks: the embedding prior for predicting a player ID, and the posterior for utilizing available player IDs to predict other quantities. Such prediction tasks include not only expected goal prediction, as examined in this paper, but also fundamental challenges such as player evaluation or game outcome prediction (Ganguly and Frank 2018). The task validation loss (Equation (11)) allows embeddings to be optimized for a specific task. During testing, the model generates a contextualized player embedding for the task model at each step, which maximize the statistical power of knowing which player is acting.

Generalize to Other Sports

Although we mainly focus on the ice hockey games, the contextualized player embedding model can be generalized to many other sports with a complex game context and a continuous flow of players' movements under a possession (e.g. basketball, soccer). These sports satisfy our assumption that the game context and a continuous play history have significant impact on the player performance. This means that the data can be represented as sequences of context features; our training method can be used to learn player embeddings for any data in this format. Specifically, VHER extracts from the game context a context-specific prior, and fits the play history with an LSTM, which allow learning a more predictive player embedding compared to previous methods.

Conclusion

Capturing what players have in common and how they differ is one of the main concerns of sports analytics. We proposed a deep representation learning approach, where each player is assigned a contextualized continuous-valued embedding vector, such that statistically similar players are mapped to similar embeddings in similar match contexts. To learn the player embeddings, we introduce a novel variational hierarchical auto-encoder, with recurrence. Recurrence allows us to model the dependence of player actions on the recent match context. The VHER learns a context-specific prior over player representations. The embedding for each player is derived from his posterior representation, given the player ID. Since the posterior representations share a common prior, the VHER induces a double shrinkage effect: similar players are mapped to similar representations in similar match contexts. The VHER is trained on the player-identification task of predicting which player is acting in a given match context. Empirical evaluation shows that the hierarchical player representations are effective for player identification, and also for the validation task of predicting whether a given player's shot will lead to a goal.

References

[Akbik, Blythe, and Vollgraf 2018] Akbik, A.; Blythe, D.; and Vollgraf, R. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, 1638–1649.

[Blei, Kucukelbir, and McAuliffe 2016] Blei, D. M.; Kucukelbir, A.; and McAuliffe, J. D. 2016. Variational inference: A review for statisticians. *CoRR* abs/1601.00670.

[Chung et al. 2015] Chung, J.; Kastner, K.; Dinh, L.; Goel, K.; Courville, A. C.; and Bengio, Y. 2015. A recurrent latent variable model for sequential data. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, 2980–2988.

[Davis, Perera, and Swartz 2015] Davis, J.; Perera, H.; and Swartz, T. B. 2015. A simulator for twenty20 cricket. *Australian & New Zealand Journal of Statistics* 57(1):55–71.

[Decroos et al. 2019] Decroos, T.; Bransen, L.; Haaren, J. V.; and Davis, J. 2019. Actions speak louder than goals: Valuing player actions in soccer. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019.*, 1851–1861.

[Fernández et al. 2019] Fernández, J.; Barcelona, F.; Bornn, L.; and Cervone, D. 2019. Decomposing the immeasurable sport: A deep learning expected possession value framework for soccer.

[Ganguly and Frank 2018] Ganguly, S., and Frank, N. 2018. The problem with win probability. In *Proceedings of the 12th MIT Sloan Sports Analytics Conference. Boston*.

[Gelman 2006] Gelman, A. 2006. Multilevel (hierarchical) modeling: What it can and cannot do. *Technometrics* 48(3):432–435.

[Good 2006] Good, P. I. 2006. *Resampling methods*. Springer.

[Gregor et al. 2015] Gregor, K.; Danihelka, I.; Graves, A.; Rezende, D.; and Wierstra, D. 2015. Draw: A recurrent neural network for image generation. In *International Conference on Machine Learning*, 1462–1471.

[Hausknecht and Stone 2015] Hausknecht, M. J., and Stone, P. 2015. Deep recurrent q-learning for partially observable mdps. In *2015 AAAI Fall Symposia, Arlington, Virginia, USA, November 12-14, 2015*, 29–37.

[Kingma and Welling 2013] Kingma, D. P., and Welling, M. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

[Kruschke 2014] Kruschke, J. 2014. *Doing Bayesian data analysis: A tutorial with R, JAGS, and Stan*. Academic Press.

[Lavi, Serj, and Ullah 2018] Lavi, B.; Serj, M. F.; and Ullah, I. 2018. Survey on deep learning techniques for person re-identification task. *CoRR* abs/1807.05284.

[Le et al. 2017] Le, H. M.; Carr, P.; Yue, Y.; and Lucey, P. 2017. Data-driven ghosting using deep imitation learning. In *MIT Sloan Sports Analytics Conference*.

[Liu and Schulte 2018] Liu, G., and Schulte, O. 2018. Deep reinforcement learning in ice hockey for context-aware player evaluation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, 3442–3448. International Joint Conferences on Artificial Intelligence Organization.

[McCallum et al. 1998] McCallum, A.; Rosenfeld, R.; Mitchell, T. M.; and Ng, A. Y. 1998. Improving text classification by shrinkage in a hierarchy of classes. In *ICML*, volume 98, 359–367.

[Mehrasa et al. 2019] Mehrasa, N.; Jyothi, A. A.; Durand, T.; He, J.; Sigal, L.; and Mori, G. 2019. A variational auto-encoder model for stochastic point processes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3165–3174.

[Murphy 2012] Murphy, K. P. 2012. *Machine learning - a probabilistic perspective*. Adaptive computation and machine learning series. MIT Press.

[Peters et al. 2018] Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, 2227–2237.