



Toward Automatically Labeling Situations in Soccer

Dennis Fassmeyer¹, Gabriel Anzer^{2,3†}, Pascal Bauer^{2,4†} and Ulf Brefeld^{1*}

¹ Machine Learning Group, Leuphana University of Lüneburg, Lüneburg, Germany, ² Department of Sport Psychology and Research Methods, Institute of Sports Science, University of Tübingen, Tübingen, Germany, ³ Sportec Solutions AG, Subsidiary of the Deutsche Fußball Liga (DFL), Munich, Germany, ⁴ DFB-Akademie, Deutscher Fußball-Bund e.V. (DFB), Frankfurt, Germany

OPEN ACCESS

Edited by:

Rui Marcelino,
University Institute of Maia, Portugal

Reviewed by:

Paizis Christos,
Université de Bourgogne, France
Hendrik Meth,
Stuttgart Media University, Germany

*Correspondence:

Ulf Brefeld
brefeld@leuphana.de

†ORCID:

Gabriel Anzer
orcid.org/0000-0003-3129-8359
Pascal Bauer
orcid.org/0000-0001-8613-6635

Specialty section:

This article was submitted to
Elite Sports and Performance
Enhancement,
a section of the journal
Frontiers in Sports and Active Living

Received: 15 June 2021

Accepted: 06 October 2021

Published: 03 November 2021

Citation:

Fassmeyer D, Anzer G, Bauer P and
Brefeld U (2021) Toward Automatically
Labeling Situations in Soccer.
Front. Sports Act. Living 3:725431.
doi: 10.3389/fspor.2021.725431

We study the automatic annotation of situations in soccer games. At first sight, this translates nicely into a standard supervised learning problem. However, in a fully supervised setting, predictive accuracies are supposed to correlate positively with the amount of labeled situations: more labeled training data simply promise better performance. Unfortunately, non-trivially annotated situations in soccer games are scarce, expensive and almost always require human experts; a fully supervised approach appears infeasible. Hence, we split the problem into two parts and learn (i) a meaningful feature representation using variational autoencoders on unlabeled data at large scales and (ii) a large-margin classifier acting in this feature space but utilize only a few (manually) annotated examples of the situation of interest. We propose four different architectures of the variational autoencoder and empirically study the detection of corner kicks, crosses and counterattacks. We observe high predictive accuracies above 90% AUC irrespectively of the task.

Keywords: sports analytics, soccer, tracking data, variational autoencoders, labeling situations

INTRODUCTION

The acquisition of tracking/positional and event data has become ubiquitous in professional football. The benefits of the resulting digital reproduction of a match, widely available in professional leagues, are twofold: Firstly, coaches, analysts and other decision makers in clubs may use data as an objective and quantitative alternative to traditional analyzes of performance, and, secondly, the collected data enables media to tell automated stories, to provide data-driven insights in what is happening on the pitch.

For example, match-analysis departments have historically spend vast amounts of time analyzing their upcoming opponent before each match by manually evaluating video footage. This work intensive approach is nowadays being supported or even partially replaced by automatic insight generation based on available data. While some information is easily accessible from the collected data, e.g., extracting the preferred formation of a team (Shaw and Glickman, 2019), other (rather tactical) pieces of information cannot be automatically computed yet, either because they are too complex (e.g., how teams behave during counterattacks), depend on the actual game philosophy of a team, require large amounts of tactical knowledge, or are considered a niche with only few interested followers. Detecting such events and patterns automatically offers a huge potential for performance analysis and may revolutionize current pre- and post-match performance analyses in professional football.

When speaking about data in soccer, we differentiate between positional/tracking and event data. Positional data, describing player and ball positions at any point in time of a match, are collected automatically via computer vision algorithms and dedicated tracking cameras. Event data, on the other hand, provides basic annotations of game events (mainly on ball actions like passes, shots, tackles, etc.) and is still acquired manually by human operators. The manual collection of such events is unsurprisingly labor and cost intensive and involves up to five operators per game. The goal of this article is to bridge the gap from the status quo toward fully-automatic annotations of soccer games.

There are several recent studies aiming to detect basic events directly out of video footage (Ekin et al., 2003; Wickramaratna et al., 2005; Kolekar and Palaniappan, 2009) or positional data (Zheng and Kudenko, 2010; Motoi et al., 2012; Richly et al., 2016; Stein et al., 2019) and others focus on the identification of sophisticated tactical patterns (Hobbs et al., 2018; Andrienko et al., 2019; Shaw and Sudarshan, 2020; Anzer et al., 2021; Bauer and Anzer, 2021). The proposed approaches provide useful solutions for their respective tasks. However, they are also restricted to either a particular data source or type of events or pattern that is to be detected; none of the above approaches offer an all-encompassing framework to deal with general detection problems.

A challenge for designing a general detector of game situations is the available data structure. While vast amounts of positional data of players and ball exist, collecting the associated labels of interest is an expensive endeavor and requires manual annotation by human experts. For example, counterattack detection first involves defining strict criteria and definitions of counterattacks before engaging in extensive search processes to annotate the matching game snippets. Consequently, it is vital to reliably extract the game situations with little external supervision. In that sense, classical supervised learning methods fail to be a viable candidate since the algorithms typically require large amounts of annotated data to achieve a good generalization error (Erhan et al., 2010). However, a strategy to mitigate the necessity of a large number of labels is to incorporate abundantly available *unlabeled data* into the training process. While there are many conceivable ways to operate within such a semi-supervised framework, we focus particularly on the variational autoencoder (VAE) (Kingma and Welling, 2013; Rezende et al., 2014) family of methods.

Variational autoencoders learn implicit low-dimensional feature representations for input data by jointly training a probabilistic encoder and decoder network. The idea is that the original observations can be reconstructed (approximately) from this lower-dimensional feature space. In fact, our semi-supervised strategy relies on inferring these semantically salient representations for annotated situations, hence reducing the need to solve a large supervised learning problem in feature space. Our instance of semi-supervised learning achieves a substantial increase in generalization ability in cases where only a few observed labels are available (Kingma et al., 2014). An essential contribution of this paper is to lift the underlying principles to spatiotemporal structures to capture the temporal

and spatial dependencies of positional data. Existing body of research on extending VAEs to sequential data mainly focuses on the generative aspects of the models rather than on their potential benefits in the context of semi-supervised learning (Chung et al., 2015; Goyal et al., 2017).

In this paper, we propose novel VAE-based feature extraction methods. Starting from the vanilla VAE, we begin with proposing a rather straight forward generalization that can be applied to positional data. A second contribution incorporates existing auxiliary labels in the training process. The idea of the auxiliary labels is to foster discriminative causes of variation in the inferred latent feature representation. The main contribution however is the development of sequential counterparts of the two VAEs to match the spatiotemporal problem domain. After one of the VAEs has been trained using unlabeled or auxiliary labeled data, only a few of the feature representations, for which labels of interest exist, are fed into a support vector machine to train the final classifier. We empirically evaluate the effectiveness of our approach on three different detection tasks, involving the detection of cornerkicks, crosses (labels obtained from event data), and counterattacks (labels manually annotated by experts). We observe detection rates above 90% AUC for all tasks and discuss several findings on methodological issues derived from further experimentation.

The remainder is structured as follows. Section Problem Setting introduces the formal problem setting. The static and sequential models are presented in sections Static Models, Sequential Models, respectively. We report on our empirical findings in section Empirical Evaluation and provide a discussion in section Discussion. Section Related Work reviews related work and section Conclusion concludes.

PROBLEM SETTING

Positional data from professional soccer is introduced as follows. Let \mathcal{A} be the set of agents (i.e., players and ball) and \mathcal{T} be the set of timesteps. For each element of the cartesian product $\mathcal{A} \times \mathcal{T}$, whereabouts of all agents on the pitch in form of two-dimensional coordinates $(g, h) \in \mathbb{R}^2$ are observed. It will be convenient to further divide the set of agents into three disjoint subsets, \mathcal{A}_1 , \mathcal{A}_2 , and \mathcal{A}_3 , corresponding to the players on teams 1, team 2, and the ball¹, respectively.

Individual spatiotemporal movements of the agents allow to augment the positional data with additional pieces of information such as the (approximated) velocity of players $(\frac{dg}{dt}, \frac{dh}{dt})$. More precisely, linearized motion for agent $a \in \mathcal{A}$ is computed via

$$(\Delta g_t^{(a)}, \Delta h_t^{(a)}) = (g_{t'}^{(a)} - g_t^{(a)}, h_{t'}^{(a)} - h_t^{(a)})$$

with $t' > t$ and $(\Delta g_t^{(a)}, \Delta h_t^{(a)}) = (0, 0)$ for the case of $t' \notin \mathcal{T}$, i.e., using a small time window between two consecutive frames. Further defining \mathcal{Y} as an *auxiliary label space* that consists of inexpensive labels (e.g., provided by event data), we are given a subset of event annotations $\mathcal{T}_{\mathcal{Y}} \subset \mathcal{T}$ s.t. $|\mathcal{T}_{\mathcal{Y}}| \ll |\mathcal{T}|$, referred

¹We have $\mathcal{A}_i \subset \mathcal{A}$ s.t. $\mathcal{A}_1 \cup \mathcal{A}_2 \cup \mathcal{A}_3 = \mathcal{A}$ and $\mathcal{A}_i \cap \mathcal{A}_j = \emptyset$.

to as $y_S := \{y_t : t \in \mathcal{T}_Y\}$. We further denote \mathcal{Y}_b as the (binary) *target space* described by an action value of interest and a “no action” value with $\mathcal{T}_{\mathcal{Y}_b} \subset \mathcal{T}$ ($|\mathcal{T}_{\mathcal{Y}_b}| \ll |\mathcal{T}|$) defining the set $y_B := \{y_t : t \in \mathcal{T}_{\mathcal{Y}_b}\}^2$. We denote the composite of all pixel coordinates and velocity values of agents a at a certain timestep as $\mathbf{x}_t = \{(g_t^{(a)}, h_t^{(a)}, \Delta g_t^{(a)}, \Delta h_t^{(a)})\}_{a \in \mathcal{A}}$ and formulate our objective as quantifying the probability over \mathcal{Y}_b given the state representation \mathbf{x}_t for all $t \in \mathcal{T}$.

An emerging issue is to find a pertinent representation of the described data for model training. A plain random concatenation of the agents’ coordinates and velocities at time t is clearly inappropriate in the sense that divergent instantiations of agent orderings also translate into divergent representations for the exact same state. Accordingly, the function that transforms instances of $\{(g_t^{(a)}, h_t^{(a)}, \Delta g_t^{(a)}, \Delta h_t^{(a)})\}_{a \in \mathcal{A}}$ into an input representation of a neural network needs to be invariant under permutation of the agents. Since the locations of the agents are given as pixel coordinates, we choose to convert these coordinates into an image-based representation, resulting in a consistent representational structure across different game settings.

The mechanism for capturing position and motion information in a 3-dimensional image representation \mathbf{x}_t is based on the approach presented in Dick and Brefeld (2019). Here, the pitch size (105×68) defines the axes in the horizontal and vertical directions, with each channel of the tensor encoding a different subset of the available information. The first 3 channels capture positional information of \mathcal{A}_1 , \mathcal{A}_2 and \mathcal{A}_3 (in that very order) by assigning constant 1 s to the coordinates defined by $(g_t^{(a)}, h_t^{(a)}) \forall a \in \mathcal{A}$ and the corresponding channel. Since agent positions live in real-world coordinates, a transfer into image pixels requires a translation $(g_t^{(a)}, h_t^{(a)}) + t$ with $t = (\frac{105}{2}, \frac{68}{2})$, effectively shifting the origin from the center of the image to the top left corner. The remaining channels track motion information, with velocity values acting as value assignments for the indices instead of constant 1 s. The speed values in g direction (Δg_t) is covered for \mathcal{A}_1 , \mathcal{A}_2 , and \mathcal{A}_3 in channels 4, 6 and 8; the information in h direction (Δh_t) is handled by channels 5, 7 and 9. All other values in the resulting input representation $\mathbf{x}_t \in \mathbb{R}^{105 \times 68 \times 9}$ are 0.

In summary, the final dataset representing a soccer game is a collection of tensor representations for each timestep $\mathcal{D} = \{\mathbf{x}_1, \dots, \mathbf{x}_{|\mathcal{T}|}\}$ with additional label sets y_S (auxiliary labels) and y_B (target labels). The goal is to use the available evidence and auxiliary labels to construct detectors that work effectively to identify situations of interest defined in \mathcal{Y}_b . To this end, we adopt a two-stage optimization procedure, which relies on the derivation of semantically meaningful feature representations. This instance of semi-supervised learning is advantageous in the present context because a large part of the model training is already accomplished independently of the specific game situation of interest. Consequently, the general detection design can be described based on the following stages:

1. The training of a VAE-based feature extraction module to transform the high-dimensional tensor data \mathbf{x}_t into a low-dimensional embedding space.
2. The training of a classifier using the derived embeddings and the available label information.

Irrespective of the first step’s choice, we use a support-vector machine (SVM) (Cortes and Vapnik, 1995) for the second step. The technical contributions of this paper address the first stage and introduce novel feature extraction methods in sections Static Models and Sequential Models. See **Figure 1** for an illustration of the information flow.

STATIC MODELS

In this section, we present static models that operate only on a single timestamp to predict a labeling of the encoded situation. The term static stems from an equivalence class of model architectures whose resulting optimization targets are derived based on the assumption that each tensor frame is iid., i.e., the computation factors across the individual timesteps of a game. Note, however, that the data points themselves contain sequential information due to the inclusion of motion vectors for each agent. We discard the time subscripts for the tensor representations \mathbf{x} since we operate within a static domain.

Preliminaries

The idea of a variational autoencoder (VAE) (Kingma and Welling, 2013; Rezende et al., 2014) is to learn a deep generative model $p_\theta(\mathbf{x}, \mathbf{z}) = p(\mathbf{z})p_\theta(\mathbf{x}|\mathbf{z})$ by maximizing the marginal log-likelihood of the training data \mathcal{D} . Due to intractabilities that arise from the integration over the latent variables \mathbf{z} , the marginal likelihood is substituted by some variational lower bound to infer the model parameters. This requires introducing a variational approximation $q_\phi(\mathbf{z}|\mathbf{x})$, which is used to approximate the intractable true posterior. The resulting (negative) evidence lower bound (ELBO) denotes the VAE training criterion and enables concurrent optimization of θ and ϕ ,

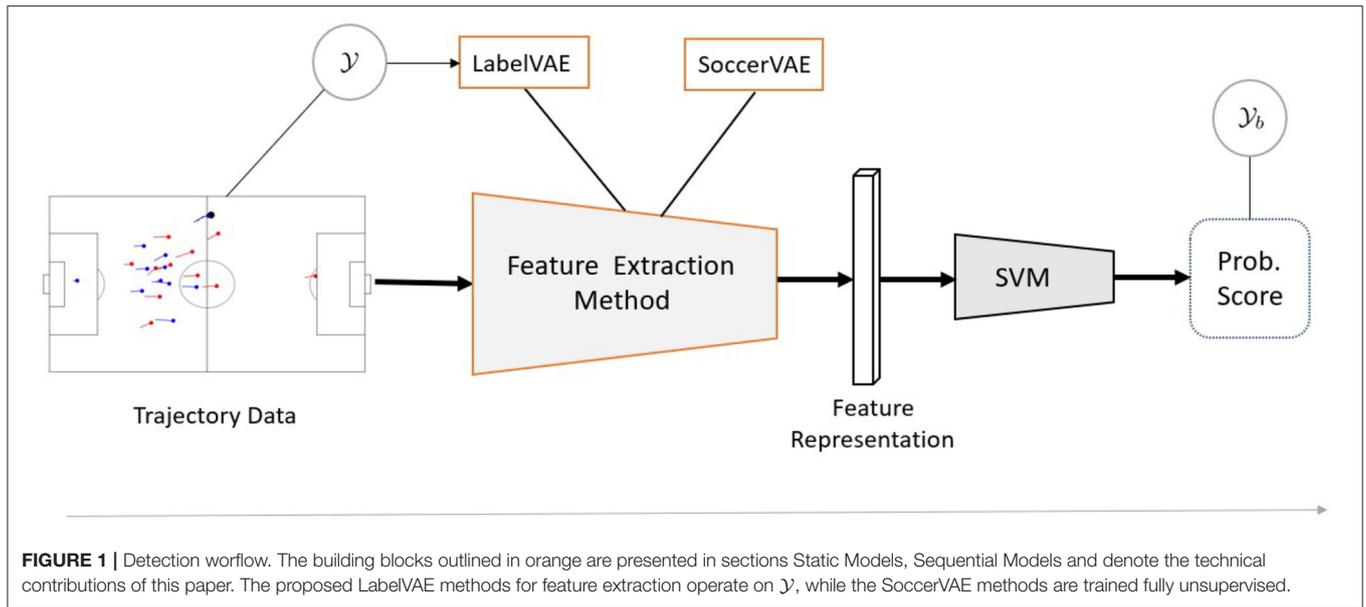
$$\begin{aligned} \log p_\theta(\mathbf{x}) &\geq \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] - \mathcal{KL}[q_\phi(\mathbf{z}|\mathbf{x}) \parallel p(\mathbf{z})] \\ &\equiv -\mathcal{L}_{\text{VAE}}(\theta, \phi; \mathbf{x}). \end{aligned} \quad (1)$$

The first term of (1) quantifies the reconstruction error and the second term measures the distance between variational approximation and the pre-defined prior in terms of the KL divergence. The learned variational distributions $q_\phi(\mathbf{z}|\mathbf{x})$ capture semantically meaningful low-dimensional feature representations of the higher-dimensional observations \mathbf{x} . This encoded information facilitates finding a generalizable discriminator, especially when labels are scarce. The merits of such a semi-supervised instance are e.g., explored in the M1 model in Kingma et al. (2014), where samples from the approximate posterior distribution over the latent variables $q_\phi(\mathbf{z}|\mathbf{x})$ are used as input data for a downstream classifier (e.g., an SVM) to learn a decision boundary in latent space.

SoccerVAE

We begin with a rather straight forward application of VAEs to the problem at-hand. The SoccerVAE uses the same optimization

²A description of the exact form and type of the label information used in this work is given in section Experimental Setup.



target as the vanilla VAE (cf. Equation 1) so that only the input and resulting choices on distribution type and architecture design need to be considered³. Regarding the former, the generating distribution of the generative model $p_\theta(\mathbf{x}, \mathbf{z})$ is modeled as a multivariate distribution of independent Bernoulli parametrized by a decoder neural net with parameters θ :

$$p_\theta(\mathbf{x}|\mathbf{z}) = \text{Bernoulli}(\mathbf{x}|\boldsymbol{\mu}(\mathbf{z}; \theta)) = \prod_{j=1}^D \text{Bernoulli}(x_j|\mu_j(\mathbf{z}; \theta)),$$

where D is the dimensionality of \mathbf{x} and $\boldsymbol{\mu}(\mu_1, \dots, \mu_D)^\top$ aggregates the individual $\mu_j \in [0, 1]$ parameters for each pixel. This constitutes a reasonable design choice as we constrain the observed values to lie in the interval $[0, 1]$.

Our generative and inference network definitions can be seen as instantiations of the class of CNN proposed by Radford et al. (2015). Specifically, the network $\boldsymbol{\mu}(\mathbf{z}; \theta)$, which incrementally converts a sampled vector \mathbf{z} to the observation space $\mathbf{x} \in \mathbb{R}^{105 \times 68 \times 9}$, is implemented using fractional-strided convolutions with ReLU activations (Nair and Hinton, 2010) and a sigmoid activation for the output layer, as well as batch normalization layers to reparametrize the intermediate layer activations (Ioffe and Szegedy, 2015; Bjorck et al., 2018). Each of the convolutional layers has kernels of the same size, with the number of kernels per layer decreasing proportionally to the depth of the network. All four proposed models deal with continuous priors given in form of standard multivariate Gaussians. The inference model $q_\phi(\mathbf{z}|\mathbf{x})$ is a diagonal Gaussian parametrized by an encoder neural net with parameters ϕ ,

$$q_\phi(\mathbf{z}|\mathbf{x}) = \mathcal{N}(\mathbf{z}|\boldsymbol{\mu}(\mathbf{x}; \phi), \text{diag}(\sigma^2(\mathbf{x}; \phi))).$$

³Unless explicitly stated, these choices are reused in the derivation of the other models.

The role of the encoder is to transform a static game situations into fixed-size vector representations. We use strided convolutions with the leaky rectified activation (Maas et al., 2013; Xu et al., 2015) and batch normalization to process the input tensors. A fully-connected layer is dedicated to mapping the final representation onto the parameter space of $q_\phi(\mathbf{z}|\mathbf{x})$, i.e., to the mean and standard deviation vector of a diagonal Gaussian, which are used in conjunction with $\mathcal{N}(\boldsymbol{\epsilon}|0, I)$ to generate the latent vector \mathbf{z} .

LabelVAE

The goal is to infer continuous latent embeddings that capture beneficial properties to detect a predefined (generally speaking: rarely occurring) game situation of interest. Hence, the quality of our approach is not primarily measured by reconstruction errors but in terms of the ability to discriminate between different types of situations in the subsequent supervised learning task. The second static model thus aims at directly optimizing a classification network. The model uses a VAE over the input variables that serves an effective regularizer. However, our envisaged optimization strategy is based on the extraction of general feature representations via pre-trained parameters to enable flexible adaption to the task at-hand.

The generative model reflects that causal factors of the observed \mathbf{x} can be broadly categorized into label-specific and label-unspecific factors,

$$p_\theta(\mathbf{x}, \mathbf{a}, \mathbf{z}) = p_\theta(\mathbf{x}|\mathbf{a}, \mathbf{z})p(\mathbf{z})p(\mathbf{a}), \tag{2}$$

where we assume that \mathbf{a} encapsulates all relevant label-specific information and \mathbf{z} the remaining label-unspecific characteristics. The dependency structure of the inference model embodies the consideration that the data-specific latent information \mathbf{z} may vary with respect to the class-specific information of \mathbf{a} , that is,

$$q_\phi(\mathbf{a}, \mathbf{z}|\mathbf{x}) = q_\phi(\mathbf{a}|\mathbf{x})q_\phi(\mathbf{z}|\mathbf{a}, \mathbf{x}). \tag{3}$$

The above approximate posterior is amenable to approximating the true posterior over the latent variables to provide a tractable lower bound on the log-likelihood $\log p_\theta(\mathbf{x})$. The resulting (negative) ELBO is the optimization target of an unsupervised data point

$$\log p_\theta(\mathbf{x}) = \log \int \int p_\theta(\mathbf{x}, \mathbf{a}, \mathbf{z}) d\mathbf{z} d\mathbf{a} \tag{4}$$

$$\begin{aligned} &= \mathbb{E}_{q_\phi(\mathbf{a}|\mathbf{x})} \left[\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{a})} [-\log p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{a})] \right. \\ &\quad \left. - \mathcal{KL}[q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{a}) \parallel p(\mathbf{z})] \right] - \mathcal{KL}[q_\phi(\mathbf{a}|\mathbf{x}) \parallel p(\mathbf{a})] \tag{5} \\ &\equiv -\mathcal{L}_u(\theta, \phi; \mathbf{x}) \end{aligned}$$

To encourage the model to capture the most relevant variational factors in the representations obtained via inference, we embed the available supervised learning signals concurrently with the unsupervised learning signals by means of an auxiliary classifier. Thus, the learning process is given by jointly maximizing the probability of each frame $\log p_\theta(\mathbf{x})$ and minimizing the auxiliary loss given the latent space realizations \mathbf{a} ,

$$\mathcal{L}_s(\theta, \phi, \xi; \mathbf{x}, y) = \mathcal{L}_u(\theta, \phi; \mathbf{x}) - \alpha \mathbb{E}_{q_\phi(\mathbf{a}|\mathbf{x})} [\log q_\xi(y|\mathbf{a})], \tag{6}$$

where ξ are the parameters of the classifier, α is a hyperparameter encoding the trade-off between generative and discriminative learning and $q_\xi(y|\mathbf{a}) = \text{Cat}(y|\boldsymbol{\pi}(\mathbf{a}; \xi))$. Equation (6) is essentially a regularized classification objective. More precisely, the second term quantifies the performance of a deep classification network with injected noise from the sampling operation $\mathbf{a} \sim q_\phi(\mathbf{a}|\mathbf{x})$ and the variational loss \mathcal{L}_u can be viewed as a form of regularization imposed on the learned representations of the supervised prediction model.

The full training criterion is then given by collecting \mathcal{L}_s and \mathcal{L}_u for the supervised and unsupervised data points of the evidence \mathcal{D} :

$$\begin{aligned} \mathcal{L}_{\text{LabelVAE}}(\theta, \phi, \xi; \mathcal{D}_u, \mathcal{D}_s) &= \sum_{(\mathbf{x}, y) \sim \mathcal{D}_s} \mathcal{L}_s(\theta, \phi, \xi; \mathbf{x}, y) \\ &\quad + \gamma \sum_{\mathbf{x} \sim \mathcal{D}_u} \mathcal{L}_u(\theta, \phi; \mathbf{x}), \tag{7} \end{aligned}$$

where $\mathcal{D}_s := \{(\mathbf{x}_t, y_t), \forall t \in \mathcal{T}_y\}$ and $\mathcal{D}_u := \mathcal{D} \setminus \mathcal{D}_s$, and trade-off γ balances the contribution of the unsupervised term to the overall objective. This can be advantageous in situations where the labeled data is very sparse ($N_l \ll N_u$) and therefore aim to externally impinge on the relative weight that is otherwise implicitly given by the data set (Siddharth et al., 2017). We define the feature vector for SVM training by concatenating the derived variables \mathbf{a} and \mathbf{z} into a single vector: $[\mathbf{a}, \mathbf{z}]$.

SEQUENTIAL MODELS

A clear limitation of the static models of the previous section is that their input is solely a single snapshot of the game.

Although direction of movement and velocities may add context to the otherwise isolated situation, the idea of processing short sequences around these situations may add important information. Hence, in this section, we present sequential variants of the previously introduced models.

We denote a slice of consecutive frames from the game \mathcal{D} as $\mathbf{x}_{\leq T}$, where T denotes the length of the game segment. Importantly, this implies that the time specifications of the frames \mathbf{x}_t refer more narrowly to the timestep in a segment within the soccer game $\mathbf{x}_{\leq T} = \mathbf{x}_1, \dots, \mathbf{x}_T$ and no longer to the timestep in the overall game (as we describe it in section Problem Setting).

SeqSoccerVAE

A viable avenue for inferring sequence-level features is to reconstruct the input sequence using a single global latent variable \mathbf{z} . While most approaches from the literature have been developed for modeling data distributions, we revisit this approach primarily to aggregate game sequences/multi-agent trajectories into informative vectors. Here we simply adapt the static VAE objective (1) to a sequential definition by assigning a temporal dimension to the data points:

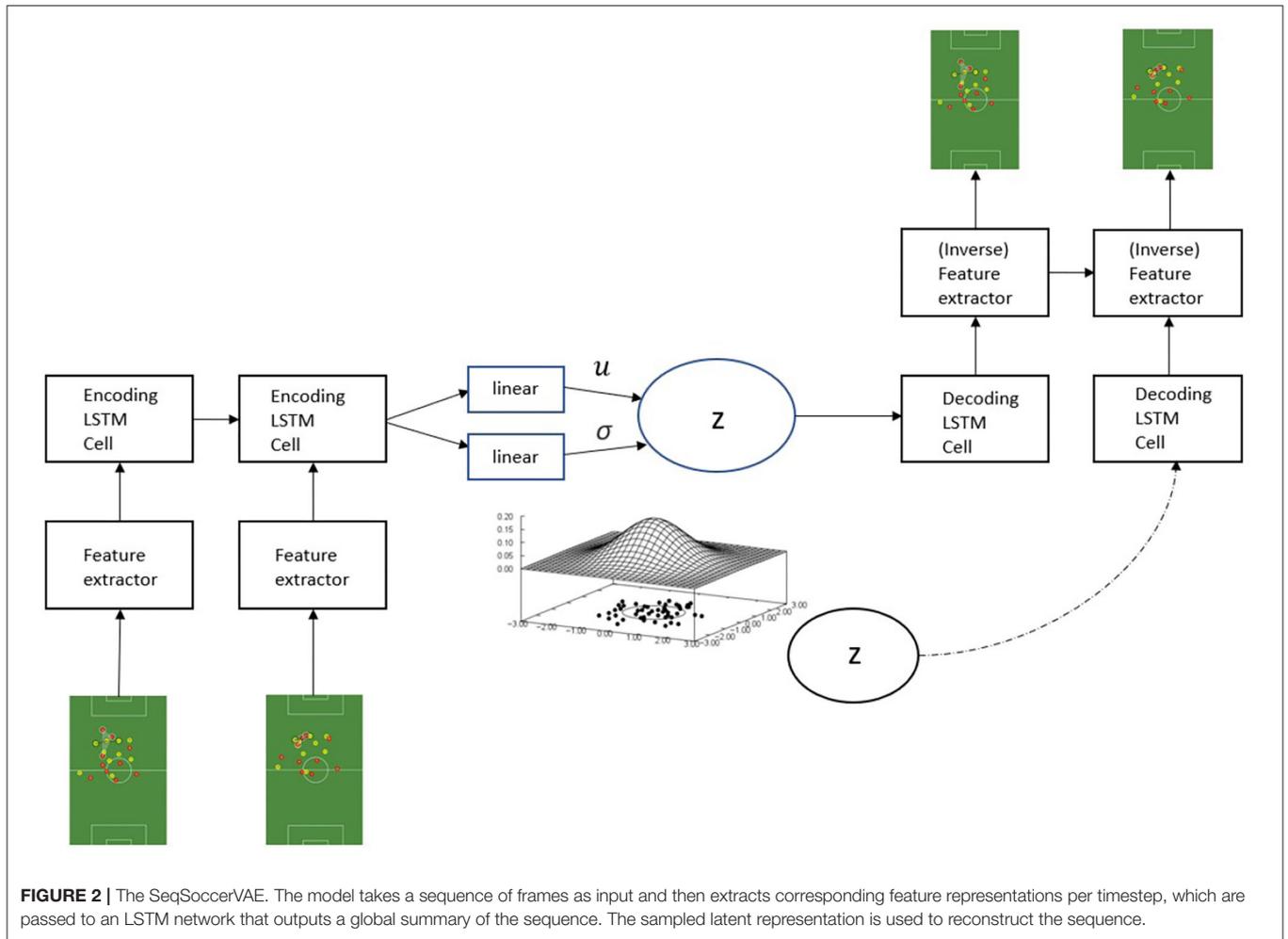
$$\begin{aligned} \mathcal{L}_{\text{SeqSoccerVAE}}(\theta, \phi; \mathbf{x}_{\leq T}) &= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x}_{\leq T})} [\log p_\theta(\mathbf{x}_{\leq T}|\mathbf{z}) \\ &\quad - \mathcal{KL}[q_\phi(\mathbf{z}|\mathbf{x}_{\leq T}) \parallel p(\mathbf{z})]. \tag{8} \end{aligned}$$

To model the components constituting Equation (8), we generalize the parameter functions for a given point to architectures suitable for sequential data. Accordingly, the parameters of the approximate posterior $q_\phi(\mathbf{z}|\mathbf{x}_{\leq T})$ are obtained from the last hidden state of an encoder RNN (parameterized by ϕ) working on the input sequence, and the generating distribution $p_\theta(\mathbf{x}_{\leq T}|\mathbf{z})$ is modeled by a decoder RNN (parameterized by θ) conditioned on the sampled hidden code alongside the previous data point, yielding the generating distribution $p_\theta(\mathbf{x}_{\leq T}|\mathbf{z}) = \prod_{t=1}^T p_\theta(\mathbf{x}_t|\mathbf{z}, \mathbf{x}_{<t})$. Thus, we force the model to encode all information about the data into the latent variable since it is the only source of information available for data reconstruction. The overall workflow of the SeqSoccerVAE is illustrated in **Figure 2**.

SeqLabelVAE

The static LabelVAE in section LabelVAE seeks to leverage discriminative information already existing in the data by injecting them into the latent space via a classification network to facilitate the detection of game situations. In this section, we propose a sequential generalization of the LabelVAE that builds upon the dependencies in inference and generative parts of its peer. Accordingly, the SeqLabelVAE utilizes a label-specific partition of the latent space into \mathbf{a}_t and \mathbf{z}_t , describing two distinct pieces of information about the data. We address the temporal dependency for successive observations by generating conditional independence for the random variables (the data and the latent variables) given the hidden states of two separate RNN networks,

$$\begin{aligned} \mathbf{h}_t^{\text{enc}} &= f_\phi(\mathbf{x}_t, \mathbf{h}_{t-1}^{\text{enc}}) \\ \mathbf{h}_t^{\text{dec}} &= g_\theta(\mathbf{a}_t, \mathbf{z}_t, \mathbf{h}_{t-1}^{\text{dec}}), \end{aligned}$$



where h_t^{enc} denotes the recurrent state for the inference model and h_t^{dec} denotes the recurrent state for the generative model.

The latent variables of the generative model at time t encode the observation x_t indirectly via the state representation h_t^{dec} , yielding the conditional distribution $p_\theta(x_t|z_{\leq t}, a_{\leq t})$. As in the previous models, we restrict ourselves to standard multivariate Gaussian priors for both latent variables per timestep. Using unconditional prior distributions may reduce the approximability of observation sequences, but our focus is on obtaining informative feature representations rather than on generating sequences. For the inference model, we condition the LabelVAE dependency structure of the posterior approximation on the RNN state h_t^{enc} , resulting in the factorization

$$q_\phi(z_{\leq T}, a_{\leq T}|x_{\leq T}) = \prod_{t=1}^T q_\phi(z_t|a_t, x_{\leq t})q_\phi(a_t|x_{\leq t}).$$

The derivations in the remainder of this section is analogous to the derivation of the static LabelVAE objective. Specifically, we optimize an unsupervised training instance by maximizing

the ELBO

$$\mathcal{J}_u(\theta, \phi; x_{\leq T}) = \mathbb{E}_{q_\phi(z_{\leq T}, a_{\leq T}|x_{\leq T})} \left[\sum_{t=1}^T -\log p_\theta(x_t|z_{\leq t}, a_{\leq t}) + \mathcal{KL}[q_\phi(z_t|x_{\leq t}, a_t) \parallel p(z_t)] + \mathcal{KL}[q_\phi(a_t|x_{\leq t}) \parallel p(a_t)] \right].$$

Also, we enforce the latent variables to encode discriminative information by introducing an auxiliary classifier for the supervised training loss

$$\mathcal{J}_s(\theta, \phi; x_{\leq T}, y) = \mathcal{J}_u(\theta, \phi; x_{\leq T}) - \alpha \mathbb{E}_{q_\phi(a_{\leq T}|x_{\leq T})} \left[\sum_{t=1}^T \log q_\xi(y_t|a_{\leq t}) \right],$$

where $\log q_\xi(y_t|a_{\leq t})$ is the per timestep classification loss and α is the hyperparameter that controls the trade-off between classification and generation. Note that the label $y \in \mathcal{Y}$ denotes the event annotation for the game situation $x_{\leq T}$, such that each frame is assigned an identical label: $y_1 = \dots = y_T = y$.

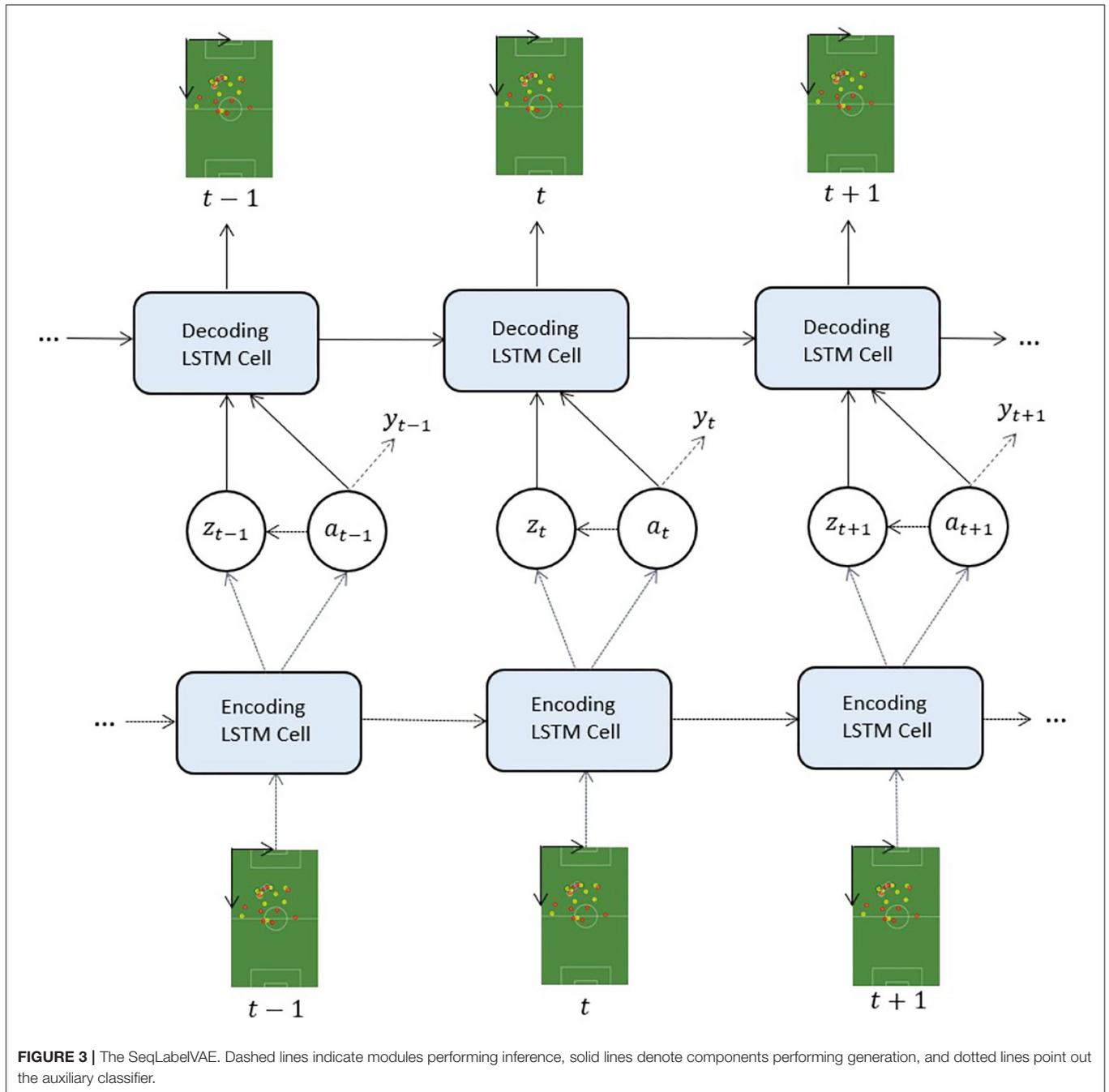


FIGURE 3 | The SeqLabelVAE. Dashed lines indicate modules performing inference, solid lines denote components performing generation, and dotted lines point out the auxiliary classifier.

We define the feature vector for classifier training on \mathcal{Y}_b by concatenating the derived variables $\mathbf{a}_{\leq T}$ and $\mathbf{z}_{\leq T}$ into a single vector: $[\mathbf{a}_1^T, \dots, \mathbf{a}_T^T, \mathbf{z}_1^T, \dots, \mathbf{z}_T^T]$. The SeqLabelVAE architecture is sketched in **Figure 3**.

EMPIRICAL EVALUATION

Data

We operate on two matches of the German national team. The tracking data consist of (g, h) positions of all players and ball, sampled at 25 frames per second. Following Dick et al. (2018),

the tensor representations of the games are computed as follows. Firstly, the origin centered representation of the player position is transformed into pixel values of the tensor representation. This is done by adding half of the size of the pitch along the horizontal and vertical direction to the position of the agents. To approximate the velocities of the players and the ball at each timestep, we compute differences in positions over the last five frames (corresponding to a time lag of 0.2 s), yielding movement vectors of the form $(\Delta g_t, \Delta h_t) = (g_{t+5} - g_t, h_{t+5} - h_t)$. Since we assume the outputs to be Bernoulli distributed, we map the resulting speed values onto the range $[0, 1]$. To obtain the final

input representation described in section Problem Setting, we incorporate the coordinates and velocity values into a 0-tensor of the size of the target shape (105, 68, 9). The updated tensor forms the input for a single timestamp. Every game consists of about 140,000 such frame representations.

Experimental Setup

As described in section Problem Setting, we define our setup with two different label spaces: the auxiliary label space \mathcal{Y} that includes all available (inexpensive) labels and the binary (expensive) label space \mathcal{Y}_b that indicates occurrences of the game situation of interest. The auxiliary label space \mathcal{Y} defines the label information y_S and originates in our study from the event data of the respective games (roughly 4000 observations per game). Note that only LabelVAE and SeqLabelVAE make use of these inexpensive labels in the training process to capture discriminative variations in the respective feature spaces. For simplicity, we focus on 5 auxiliary labels, $\mathcal{Y} = \{\text{shot}, \text{cross}^4, \text{ground}, \text{pass}, \text{other}\}$. If more than one auxiliary label is active in a snapshot, we select the minority label for the observations in question.

By contrast, the label space \mathcal{Y}_b defines the label information y_B used for SVM training and depends on the task at-hand. Our exemplary use cases target game actions of increasing difficulties by predicting variables encoded already in the available auxiliary labels or annotated manually by human experts. Accordingly, when employing fully unsupervised feature extraction methods (i.e., SoccerVAE and SeqSoccerVAE), targets y_B are the only label information required. We elaborate on the exact construction of the set y_B when discussing the predictive results in the following section.

We use one game for training and model selection and the other game for testing. In the training process, parameters of the static and sequential VAEs are optimized as well as parameters of the support vector machine which serves as the final classifier. After training, the best parameters are fixed and used for processing the test game. For every frame in the test game, probabilities of the quantities of interest are computed as follows: A static (section Static Models) or sequential (section Sequential Models) approach computes the embedding of the situation which is then used as input to the support vector machine which computes the prediction of interest and a softmax turns this prediction into a probability.

To assess the detection performance, we mainly use two different performance metrics: the area under the ROC curve (AUC) and the F1 score. We calculate the relevant components that constitute the F1 score (true positives (TPs), false positives (FPs), and false negatives (FNs)) as follows. To identify an action, we apply a threshold to the derived probability estimates for each frame of the test game. The independently detected frames are then converted into coherent game situations (or positive prediction instances), defined as a set of detected consecutive frames where the time gap between 2 successive frames is less than 10 s. The average length of the detected sequences depends on the concrete application, but it is in the range of a few

seconds in most cases. We obtain TP values (FP values) if any (no) element within the extracted sequences is assigned the label of interest. Further, we define FN values as true action frames that remain undetected, i.e., do not occur within the positively predicted regions. We compute F1-scores for 50 distinct threshold values in the range between 0.6 and 0.98 and only report the maximum F1-score in the subsequent section⁵.

We compare our approaches to a fully supervised deep convolutional network that directly processes the tensor frames. The architecture of the baseline is identical to the feature extraction modules of our inference models, i.e., it consists of convolutional and batch normalization layers with LeakyReLU activation functions. The output dimensionality equals 1, and we use the standard binary cross-entropy loss for training. That is, the baseline directly computes the prediction of the desired label without a need for an additional SVM but lacks the reconstruction part of the proposed networks. We train the model with RMSprop (Tieleman and Hinton, 2012) and a batch size of 4. All methods are implemented with Tensorflow 2.0 (Abadi et al., 2016)⁶.

To ensure clarity regarding the used baseline architecture, we replaced “feature extraction modules of our inference models” with “encoder network of the SoccerVAE.” We report the comparison with this supervised baseline in **Table 1**.

Predictive Accuracies

We showcase the expressivity of our approaches on three tasks with gradually increasing difficulty, the first one being the automatic **detection of cornerkicks**. The task should be the easiest one as the spatial distribution of agents is very indicative and event data provides ground-truth labels. The second task is the **detection of crosses**. Again, ground-truth is provided by event data, however, the spatial distribution of the agents is not as obvious as for cornerkicks. For both tasks, we train the models on one game and use another one for testing and evaluation.

The third task is the **detection of counterattacks** and clearly more involving than the former two. The task is more difficult than the previous two as many different temporal aspects need to be learned by the model, including gaining and maintaining ball possession, etc. Labels for this task are provided by human experts. Since the effort of labeling is tedious, we train the models only on the first half of a game and evaluate on the second.

We begin with the detection of cornerkicks. For this straight forward task, the variational autoencoders are trained on a single game. The subsequent SVM is trained on 16 labeled examples per class (cornerkick vs. no cornerkick), where the negative examples are randomly drawn from the training game. The test game contains 26 cornerkick situations. The baseline uses the same training and testing set as the downstream SVM. **Table 1** (top rows) summarizes the results for the different metrics on the test/validation game. All semi-supervised approaches outperform the fully-supervised baseline with SeqLabelVAE being the best predictor in this task. Comparing the static

⁵Therefore, unlike the reported AUC values, the F1 scores are validation values as we engage in threshold optimization.

⁶The source code is available at <https://github.com/fassmeyer/labeling-situations>.

TABLE 1 | Results for the detection of cornerkicks, crosses and counterattacks.

Task	Model	AUC	TP-Rate	Precision	F1	Length
Cornerkick	Baseline	0.909	0.904	0.478	0.620	13.624
	SoccerVAE	0.944	0.940	0.578	0.716	14.445
	LabelVAE	0.967	0.877	0.670	0.760	8.451
	SeqSoccerVAE	0.975	0.886	0.792	0.824	11.054
	SeqLabelVAE	0.986	0.920	0.785	0.850	14.560
Cross	Baseline	0.827	0.765	0.507	0.606	20.070
	SoccerVAE	0.920	0.933	0.575	0.707	24.229
	LabelVAE	0.924	0.927	0.577	0.711	24.812
	SeqSoccerVAE	0.931	0.983	0.578	0.728	19.138
Counterattack	SeqLabelVAE	0.940	0.812	0.683	0.739	16.750
	SeqSoccerVAE	0.835	0.855	0.533	0.651	7.586
	SeqLabelVAE	0.912	0.745	0.726	0.730	3.712

The highest values are indicated in bold face. The average length of the detected segments is given in seconds. All numbers are averages on the test game.

models shows decent improvements of the LabelVAE over the SoccerVAE. Furthermore, the average length of the detected sequences is significantly lower for the LabelVAE. Since the average length is a good indicator concerning the width of the predicted amplitudes, the value can be interpreted as a confidence measure of the predictions. Though LabelVAE performs worse than the sequential models, the static models provide solid results in this task, presumably because the agents' distribution on the playing field is easily distinguishable from other game situations. When comparing the sequential models, we find that the SeqLabelVAE performs better than SeqSoccerVAE. This improvement however comes at the cost of detection lengths.

Next, we study the detection of crosses using the same extracted features as for cornerkick detection. The classifier is trained on 33 examples per class (cross vs. no cross), and the test game consists of 38 cross situations. **Table 1** (center rows) summarizes the results for the different metrics for the test/validation game. The trends are largely consistent with those of the corner detection task but at a lower overall level. The drop in performance stems from the variance in spatial distributions of agents that render the detection of crosses naturally more difficult than cornerkicks.

For the detection of counterattacks, static methods cannot sensibly be applied as the sequential nature and complexity of the situation (change of ball possession, maintaining ball possession thereafter, etc.) cannot be captured by focusing on only a single point in time. Consequentially, we only evaluate the sequential models using the first half of a manually annotated game with 27 counterattack situations for training and use the second half containing 33 situations for testing the classifier. The inherent complexity of counterattacks render the task much more challenging compared to the detection of cornerkicks or crosses. **Table 1** (bottom rows) shows the results. As in the previous cases, the SeqLabelVAE emerges as the model of choice. Albeit detection performances are below previous ones, the findings show the potential of the models in challenging domains with manual labels. The detection rate of counterattacks is still above 91% AUC.

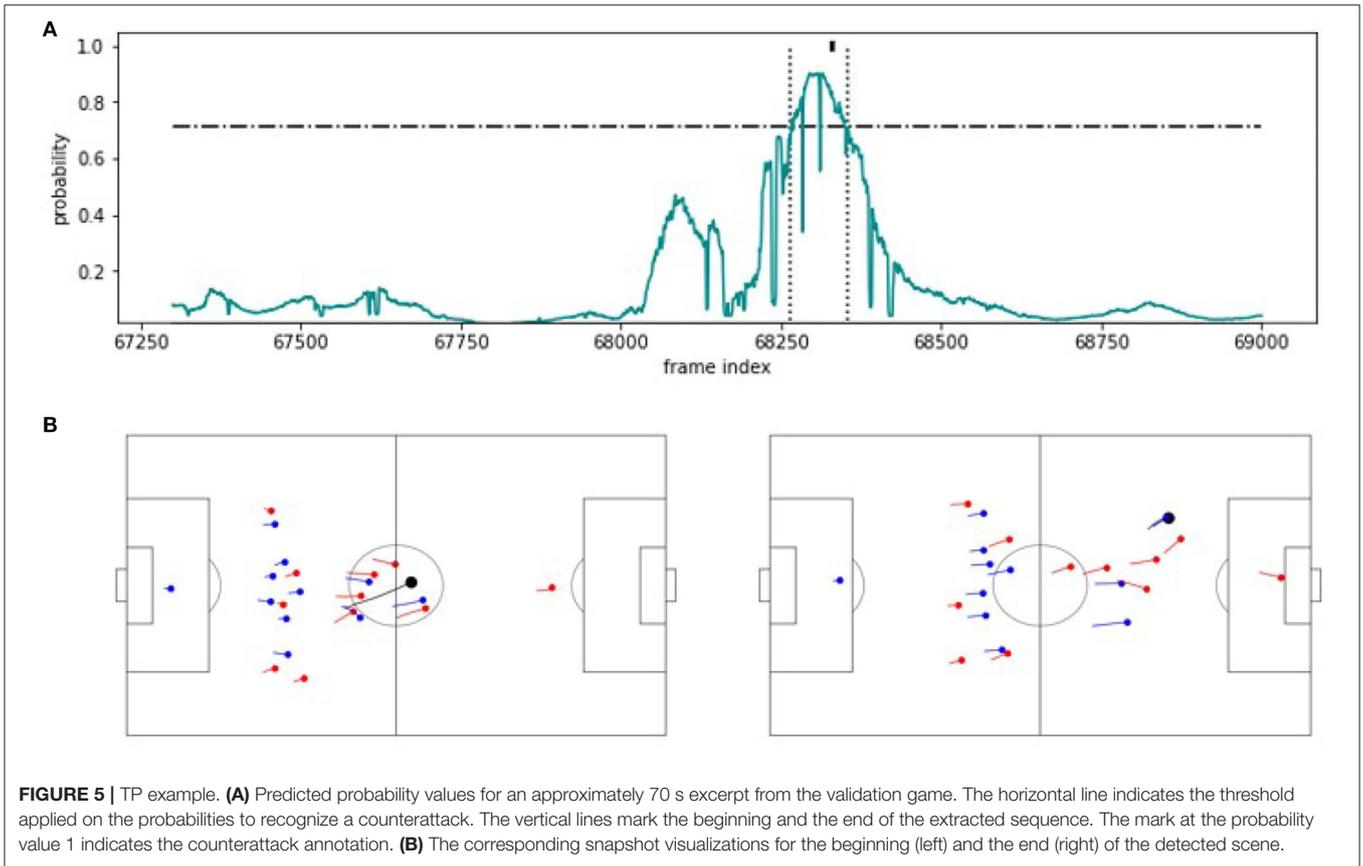
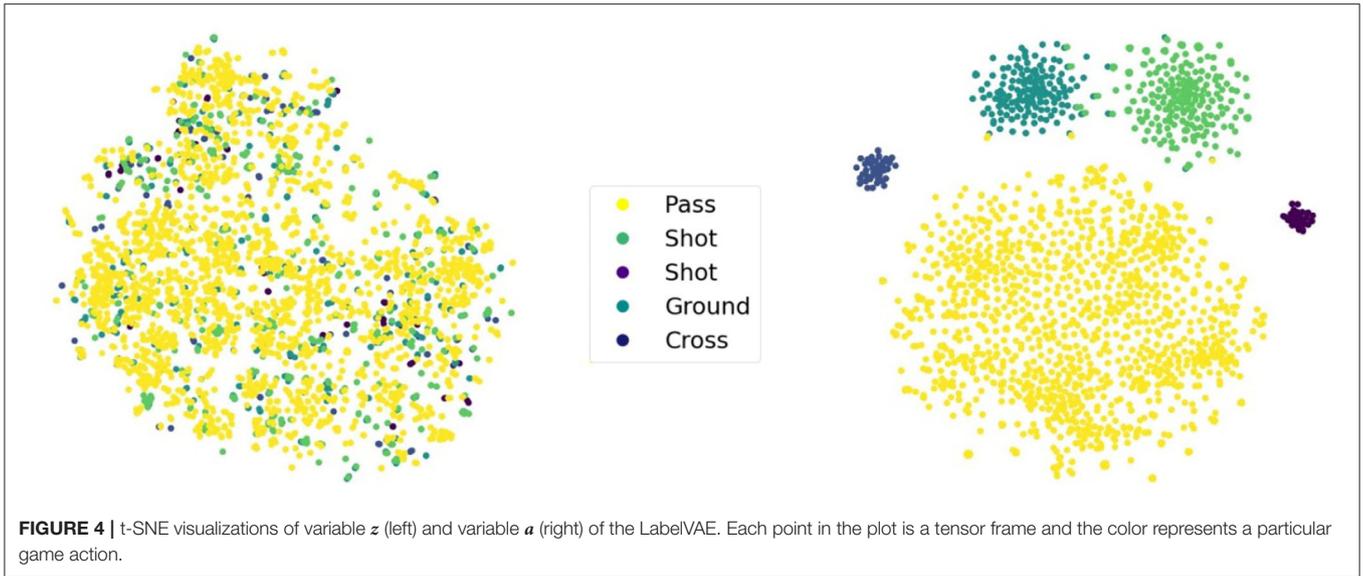
Analyzing LabelVAE

To shed light on the effect of the auxiliary labels used in LabelVAE and SeqLabelVAE, we visualize the latent space of the former using t-SNE (Van der Maaten and Hinton, 2008) in **Figure 4**. Recall that the generative model of LabelVAE makes use of two latent variables \mathbf{a} and \mathbf{z} . The former encodes label-specific information while the latter captures all label-unspecific traits. Thus, both latent variables are supposed to capture different properties which actually holds true for the trained models as can be seen in the figure. Every point in the figure corresponds to a game situation and its color indicates the attached auxiliary label. The difference of the two latent variables is clearly visible and accentuated by a clear separation into action clusters (right part of figure) for \mathbf{a} and the absence of any class structure (left part) for \mathbf{z} . Since both variables are used to reconstruct the tensor frames, but merely variable \mathbf{a} concurrently needs to accurately discriminate between the different actions, it stands to reason that \mathbf{z} captures position-specific information useful for frame reconstruction.

Recall, that the empirical results for the LabelVAE in **Table 1** are based on concatenating the two latent variables \mathbf{a} and \mathbf{z} into a single feature vector yielding an AUC of 96.7% for cornerkicks. Passing on only a single variable to the SVM decreases the performance to 94.0% for \mathbf{z} and 90.1% for \mathbf{a} , respectively. Hence, the two variables complement one another and focus on different aspects of the problem.

Qualitative Assessment

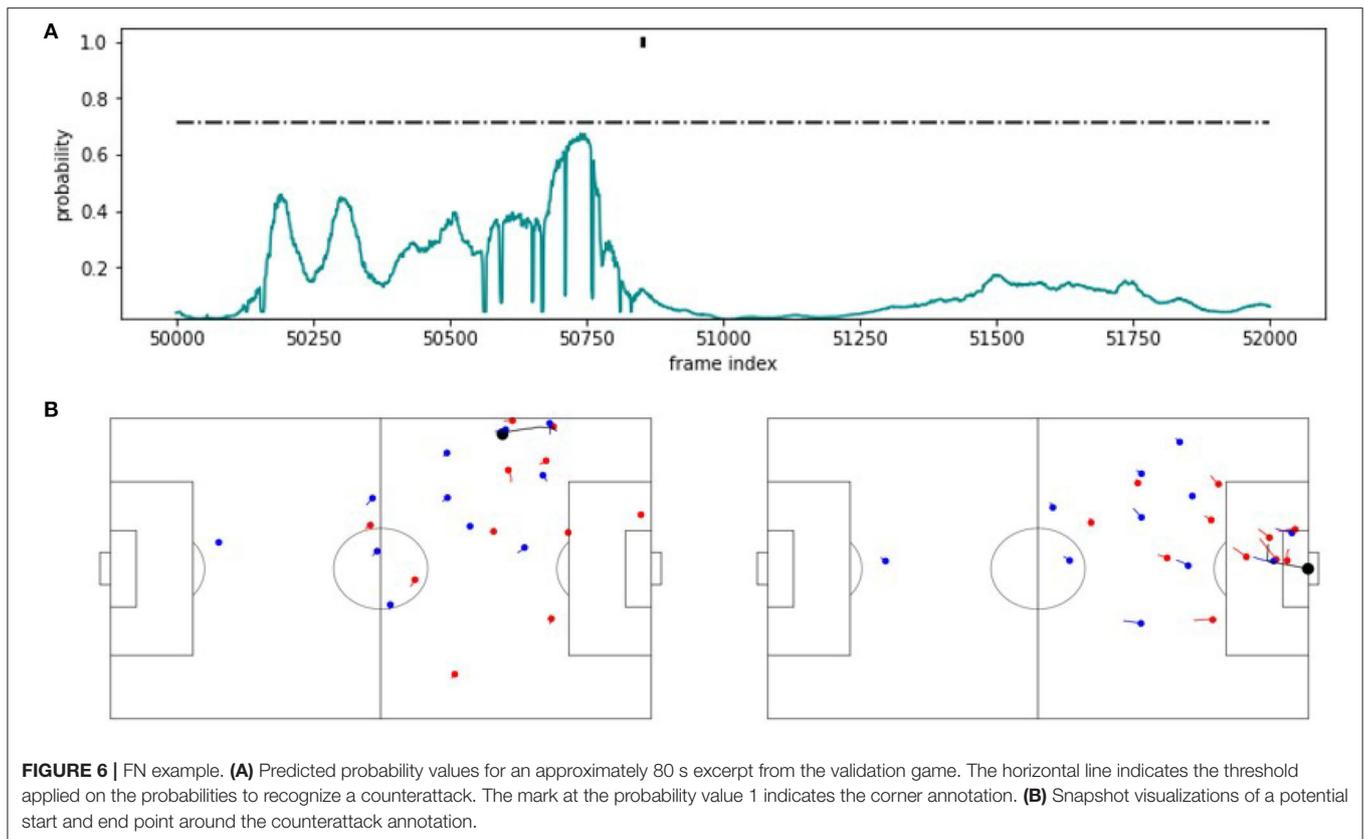
To shed light on the nature of the proposed methodology, we compare the structure of correctly and incorrectly predicted examples for the detection of counterattacks on the example of SeqSoccerVAE. We begin with a correctly identified counter attack in **Figure 5**. The upper part of the figure shows the detection probabilities computed from the output of the SVM. The black indicator on top of the figure at timestamp 68.330 indicates the true label by the experts. The SeqSoccerVAE classifies the indicated segment above the threshold (dashed line) between timestamps 68.265 and 68.355 as a successful



counterattack. The two figures below display the snapshots at the beginning and end of the detected scene and clearly show the successful counterattack that over both halves of halves of the pitch.

By contrast, **Figure 6** shows a false negative. The detection probabilities shown in the upper part of the figure stay

constantly below the threshold and consequentially, the turnover is missed by the classifier. Interestingly, the expert annotation is at a position, where the probability for a counterattack has decreased entirely and stays around zero. We credit this poor performance to the rather crowded origin of the situation and the many defending players behind the



ball. The situation is clearly different from the one shown in **Figure 5**.

Last but not least, **Figure 7** shows a false positive. As can be seen, the situation resembles the one agent in **Figure 5** but here, the turnover fails and correspondingly, there is no expert annotation. This result expresses both, the strength and the limitation of the SeqSoccerVAE, and possibly the use of VAEs in general for such tasks. By using an autoencoder, we implicitly assume that similar situations in feature space will have a similar outcome in the real world. On one hand, this assumption allows to use many unlabeled situations to extract meaningful features and render the entire classification approach with only a handful of (expert) labels feasible. On the other hand, once the feature representation is fixed, the subsequent SVM is unlikely to differentiate neighboring situations although their labels suggest separation. However, the overall performance impressively shows that the latter case does not occur very often, resulting in an excellent total detection rate.

Importance of Labeled Data

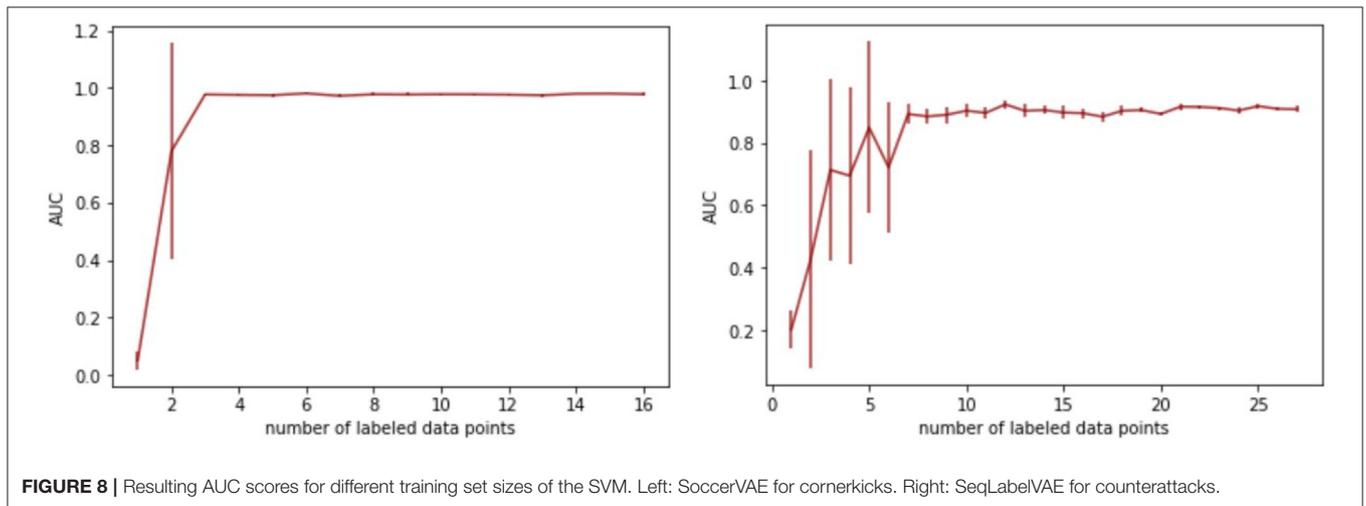
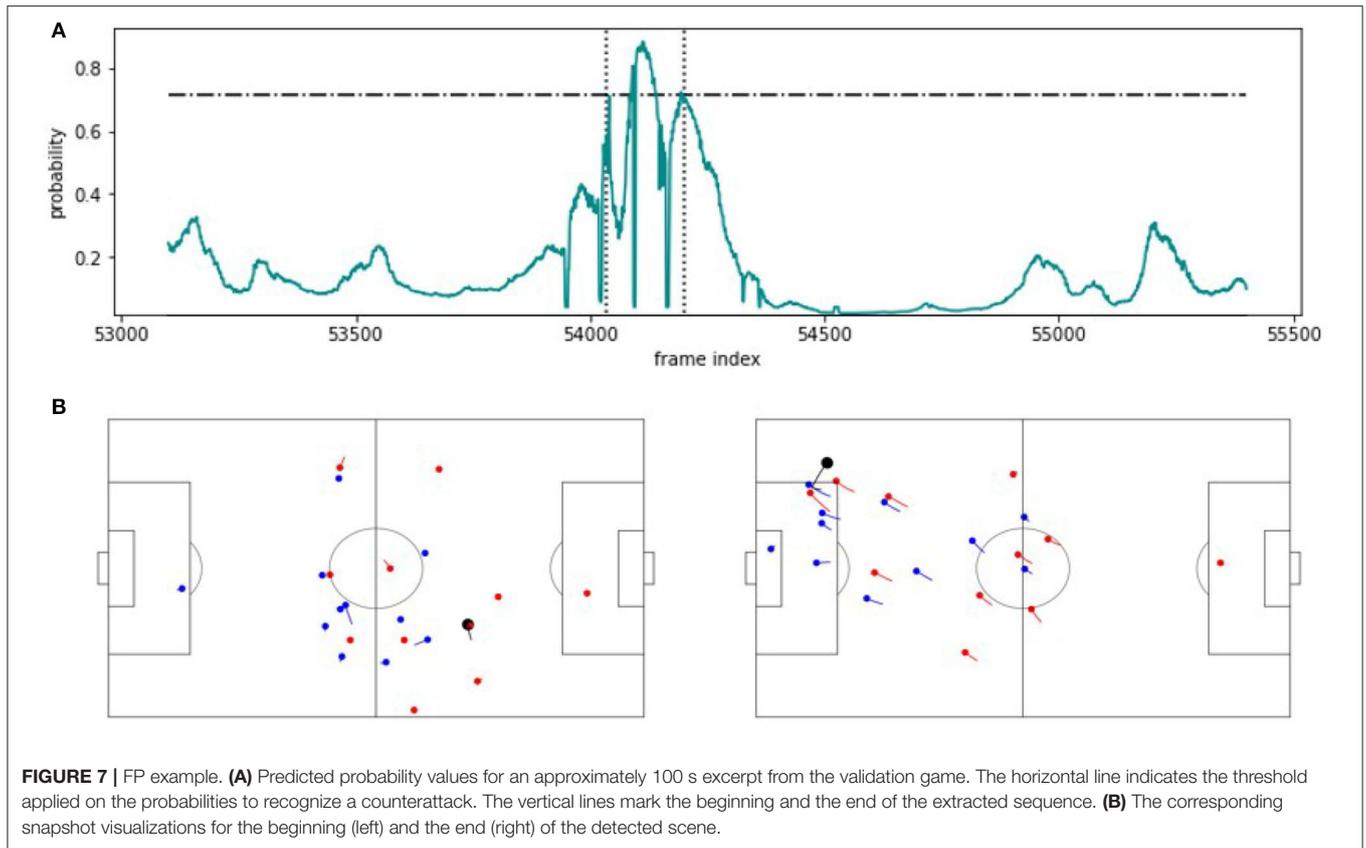
The idea of the paper grounds on splitting the original problem of labeling situations in soccer into two: an unlabeled⁷ grouping of similar situation by a variational autoencoder (VAE) and feeding the learned feature representation into a support vector machine (SVM) to compute the final prediction. This approach promises

⁷Recall that we use auxiliary labels in (Seq)LabelVAE to enforce sensible groupings.

a much better structured feature space that allows the SVM to learn an accurate hyperplane with only a few labeled instances. This, in turn, renders the approach useful for practitioners as they only need to provide manual labels for a handful of situations.

To investigate the models' applicability in a practical context, we quantify the (human) labeling effort to achieve accurate performance for the detection of cornerkicks and counterattacks, respectively. **Figure 8** shows the results. The *y*-axis shows AUCs and the *x*-axis depicts the number of positive training examples which are (manually) labeled. In addition, the same amount of negative examples are introduced, however, these are randomly drawn from the training games and do not need manual attention. To reduce the effect of the randomness in the training sets, we report on averages over five runs; error bars indicate standard error. The left part of the figure shows the results for the SoccerVAE and the detection of cornerkicks. A training set with only six instances, three (manually) labeled positive and three randomly drawn negative ones, is sufficient to obtain optimal performance. Adding more instances to the training set does not lead to further improvements.

For the detection of counterattacks with the SeqLabelVAE (right part of figure), the performance stabilizes for about seven manually labeled data points. Increasing the size of the training set further reduces the variance that is introduced by selecting only a few positive and negative examples and renders the classifier more robust. However, the key message is



that only seven manual annotations suffice to accurately detect counterattacks with a detection rate (AUC) of over 90%.

DISCUSSION

Our approach allows us to detect basic events (cornerkicks and passes) as well as more complicated patterns (i.e., counterattacks) without requiring massive sets of annotated data and without falling back to rule-based approaches. The detection of a

more complicated pattern, namely counterattacks, is addressed in Hobbs et al. (2018) using an unsupervised clustering. By making use of a few expert-labels, we combine a data-driven approach with expert guidance. The autoencoder-based approach introduced by of Karun Singh⁸ is improved in two ways: First, we use a variational autoencoder and second, we extend the

⁸Opta Analytics Pro Forum, 2019 London <https://www.youtube.com/watch?v=H1iho17lnoI>.

approach to use time series instead of static snippets of positional data. Bauer and Anzer (2021) compare a rule based model, to a machine learning based one to identify the tactical pattern of counterpressing automatically across 20,000 labels from 97 matches. For their trained model they extract 137 hand-crafted features. The advantage of our approach is that it not only requires far fewer labeled observations, but also works with very simple basic features. It is easily reproducible for any other pattern, and, can be adjusted quickly even if definitions of patterns slightly change when the game-philosophy shifts (e.g., because of a coaching change).

Besides the potential to reduce the costs of manual event data collection, our approach enables several team performance affecting applications: The automatic detection of relevant patterns saves coaches and match-analysis departments not only time, but furthermore increases consistency and offers scalability. This can consequently be used to perform long-term analysis across multiple seasons or even leagues. Furthermore, besides match-analysis this methodology could also be integrated in the player scouting process, by identifying certain beneficial individual action patterns and finding players that exhibit these frequently.

While our work describes the technical framework to achieve these results, for it to be usable in a club environment, one would need to integrate it in an application that fits in seamlessly into daily routines of match-analysis or scouting departments.

RELATED WORK

This paper explores issues related to VAE-based semi-supervised learning, with the main contribution in this field introduced by Kingma et al. (2014). Our SoccerVAE and LabelVAE are clearly inspired by their proposals M1 and M2. Specifically, the authors integrate label information into the assumption of the data generation process, thereby obviating the necessity for the otherwise required supervised learning task on extracted label-feature pairs. Recent work by Joy et al. (2020) argues that explicitly modeling the connection between labels and their corresponding latent variables improves the classification accuracy compared to the M2 approach and allows to learn meaningful representations of data effectively. Maaløe et al. (2016) also improve M2 classification performance by introducing an auxiliary variable that leaves the original model unchanged but increases the flexibility of the variational posterior. This can result in convergence to a parameter configuration that is closer to a local optimum of the actual data likelihood (due to potentially better fits to the complex posterior) while maintaining the computational efficiency of fully factorized models. Siddharth et al. (2017) choose a more generalized formulation of semi-supervised learning with VAE compared to the models in the work by Kingma et al. (2014). Their framework allows choosing complex models, such as when a random variable determines the number of latent variables itself.

In addition to static semi-supervised tasks, this work methodologically touches a branch of research that describes methods involving autoencoders to model sequential data. Bayer and Osendorfer (2014) incorporate stochasticity into vanilla RNNs by making the independently sampled latent variables an

additional input at each timestep. Chung et al. (2015) apply a similar model termed VRNN to speech data, sharing parameters between the RNNs for the generative model and the inference network. In Goyal et al. (2017), the latent variable participates to the prediction of the next timestep, and the variational posterior is informed about the whole future in the sequence modeled by an RNN processing the sequence backwards. While the previously mentioned methods sample a separate latent variable at each timestep, Bowman et al. (2015) propose an RNN-based VAE to derive global latent representations for sentences. The approach to modeling human-drawn images discussed in Ha and Eck (2017) shares many architectural similarities to Bowman et al. (2015), but uses an additional backward RNN encoder. Teng et al. (2020) introduce a semi-supervised training objective for modeling sequential data where the model specification draws inspiration from Kingma et al. (2014) and Chung et al. (2015).

CONCLUSIONS

We studied automatic annotation of non-trivial situations in soccer. We proposed to separate the problem into an unsupervised autoencoder to learn a meaningful feature representation and a supervised large-margin classification. The advantage of this separation lied in the use of abundant unlabeled data that allowed for learning a nicely structured feature space so that only a few labeled examples were needed in the classifier to learn the target concept of interest.

We proposed two variants of autoencoders, a straight forward application of existing results (SoccerVAE) and a more sophisticated variant that used auxiliary labels and allowed for even more discriminative feature spaces (LabelVAE). In addition to these two static variants, we devised their sequential peers to account for the spatiotemporal nature of soccer. Empirically, we studied the performance of the four approaches on three different detection tasks, involving cornerkicks, crosses, and counterattacks. The SeqLabelVAE turned out the best competitor and outperformed all others with detection rates of 91% AUC or higher in all problems for only a few labeled examples.

While our methods emerged as valuable tools for detection tasks in soccer, there are some shortcomings that could be addressed in future work. A possible starting point is to compare the implicit regularization of our semi-supervised approach against supervised sequential models with alternate regularization methods (Semeniuta et al., 2016). From the perspective of achieving the lowest possible generalization error, there are several avenues for potential variations. Future work might include alternate probabilistic assumptions (Goyal et al., 2017; Joy et al., 2020) such as conditioning the variational distribution on the full input sequence (Goyal et al., 2017), novel regularization techniques for VAE (Tolstikhin et al., 2017; Ma et al., 2019; Deasy et al., 2020), other approaches to semi-supervised learning (Kingma et al., 2014; Dai and Le, 2015) such as transfer learning (Fabius and Van Amersfoort, 2014; Srivastava et al., 2015), or to achieving consistent agent representations such as

graph-networks (Sun et al., 2019; Yeh et al., 2019) and tree-based role alignments (Lucey et al., 2013; Sha et al., 2017; Felsen et al., 2018).

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/supplementary materials, further inquiries can be directed to the corresponding author/s.

REFERENCES

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., et al. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv [Preprint]*. [arXiv:1603.04467](https://arxiv.org/abs/1603.04467).
- Andrienko, G., Andrienko, N., Anzer, G., Bauer, P., Budziak, G., Fuchs, G., et al. (2019). Constructing spaces and times for tactical analysis in football. *IEEE Trans. Vis. Comput. Graph.* 27, 2280–2297. doi: 10.1109/TVCG.2019.2952129
- Anzer, G., Bauer, P., and Brefeld, U. (2021). The origins of goals in the German Bundesliga. *J. Sports Sci.* doi: 10.1080/02640414.2021.1943981
- Bauer, P., and Anzer, G. (2021). Data-driven detection of counterpressing in professional football—A supervised machine learning task based on synchronized positional and event data with expert-based feature extraction. *Data Min. Knowl. Disc.* 35, 2009–2049. doi: 10.1007/s10618-021-00763-7
- Bayer, J., and Osendorfer, C. (2014). Learning stochastic recurrent networks. *arXiv [Preprint]*. [arXiv:1411.7610](https://arxiv.org/abs/1411.7610).
- Bjorck, J., Gomes, C., Selman, B., and Weinberger, K. Q. (2018). Understanding batch normalization. *arXiv [Preprint]*. [arXiv:1806.02375](https://arxiv.org/abs/1806.02375).
- Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R., and Bengio, S. (2015). Generating sentences from a continuous space. *arXiv [Preprint]*. [arXiv:1511.06349](https://arxiv.org/abs/1511.06349). doi: 10.18653/v1/K16-1002
- Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A., and Bengio, Y. (2015). A recurrent latent variable model for sequential data. *arXiv [Preprint]*. [arXiv:1506.02216](https://arxiv.org/abs/1506.02216).
- Cortes, C., and Vapnik, V. (1995). Support-vector networks. *Mach. Learn.* 20, 273–297. doi: 10.1007/BF00994018
- Dai, A. M., and Le, Q. V. (2015). Semi-supervised sequence learning. *arXiv [Preprint]*. [arXiv:1511.01432](https://arxiv.org/abs/1511.01432).
- Deasy, J., Simidjievski, N., and Liò, P. (2020). Constraining variational inference with geometric jensen-shannon divergence. *arXiv [Preprint]*. [arXiv:2006.10599](https://arxiv.org/abs/2006.10599).
- Dick, U., and Brefeld, U. (2019). Learning to rate player positioning in soccer. *Big Data* 7, 71–82. doi: 10.1089/big.2018.0054
- Dick, U., Tavakol, M., and Brefeld, U. (2018). Rating continuous actions in spatial multi-agent problems.
- Ekin, A. A., Tekalp, M., and Mehrotra, R. (2003). “Automatic soccer video analysis and summarization,” in *IEEE Transactions on Image Processing* 12.7, 796–807.
- Erhan, D., Courville, A., Bengio, Y., and Vincent, P. (2010). “Why does unsupervised pre-training help deep learning?” in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings* (Sardinia), 201–208.
- Fabius, O., and Van Amersfoort, J. R. (2014). Variational recurrent auto-encoders. *arXiv [Preprint]*. [arXiv:1412.6581](https://arxiv.org/abs/1412.6581).
- Felsen, P., Lucey, P., and Ganguly, S. (2018). “Where will they go? predicting fine-grained adversarial multi-agent motion using conditional variational autoencoders,” in *Proceedings of the European Conference on Computer Vision (ECCV)* (Munich), 732–747.
- Goyal, A., Sordani, A., Côté, M.-A., Ke, N. R., and Bengio, Y. (2017). Z-forcing: Training stochastic recurrent networks. *arXiv [Preprint]*. [arXiv:1711.05411](https://arxiv.org/abs/1711.05411).
- Ha, D., and Eck, D. (2017). A neural representation of sketch drawings. *arXiv [Preprint]*. [arXiv:1704.03477](https://arxiv.org/abs/1704.03477).
- Hobbs, J., Power, P., Sha, L., Ruiz, H., and Lucey, P. (2018). “Quantifying the value of transitions in soccer via spatiotemporal trajectory clustering,” in *MIT Sloan Sports Analytics Conference*, 1–11.

AUTHOR CONTRIBUTIONS

All authors listed have made a substantial, direct and intellectual contribution to the work, and approved it for publication.

ACKNOWLEDGMENTS

We would like to thank the German Football Association (DFB) for providing the data for this study.

- Ioffe, S., and Szegedy, C. (2015). “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *International Conference on Machine Learning* (Lille: PMLR), 448–456.
- Joy, T., Schmon, S. M., Torr, P. H., Siddharth, N., and Rainforth, T. (2020). Rethinking semi-supervised learning in vaes. *arXiv [Preprint]*. [arXiv:2006.10102](https://arxiv.org/abs/2006.10102).
- Kingma, D. P., Rezende, D. J., Mohamed, S., and Welling, M. (2014). Semi-supervised learning with deep generative models. *arXiv [Preprint]*. [arXiv:1406.5298](https://arxiv.org/abs/1406.5298).
- Kingma, D. P., and Welling, M. (2013). Auto-encoding variational bayes. *arXiv [Preprint]*. [arXiv:1312.6114](https://arxiv.org/abs/1312.6114).
- Kolekar, M. H., and Palaniappan, K. (2009). Semantic concept mining based on hierarchical event detection for soccer video indexing. *J. Multimedia* 4, 298–312. doi: 10.4304/jmm.4.5.298-312
- Lucey, P., Bialkowski, A., Carr, P., Morgan, S., Matthews, I., and Sheikh, Y. (2013). “Representing and discovering adversarial team behaviors using player roles,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (Portland, OR: IEEE), 2706–2713.
- Ma, X., Zhou, C., and Hovy, E. (2019). Mae: Mutual posterior-divergence regularization for variational autoencoders. *arXiv [Preprint]*. [arXiv:1901.01498](https://arxiv.org/abs/1901.01498).
- Maaløe, L., Sønderby, C. K., Sønderby, S. K., and Winther, O. (2016). “Auxiliary deep generative models,” in *International Conference on Machine Learning* (New York, NY: PMLR), 1445–1453.
- Maas, A. L., Hannun, A. Y., and Ng, A. Y. (2013). “Rectifier nonlinearities improve neural network acoustic models,” in *International Conference on Machine Learning (ICML)*. 30.
- Motoi, S., Misu, T., Nakada, Y., Yazaki, T., Kobayashi, G., Matsumoto, T., et al. (2012). Bayesian event detection for sport games with hidden Markov model. *Pattern Anal. Appl.* 15, 59–72. doi: 10.1007/s10044-011-
- Nair, V., and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *icml*.
- Radford, A., Metz, L., and Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv [Preprint]*. [arXiv:1511.06434](https://arxiv.org/abs/1511.06434).
- Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). “Stochastic backpropagation and approximate inference in deep generative models,” in *International Conference on Machine Learning* (Beijing: PMLR), 1278–1286.
- Richly, K., Bothe, M., Rohloff, T., and Schwarz, C. (2016). “Recognizing compound events in spatio-temporal football data,” in *IoTBD 2016-Proceedings of the International Conference on Internet of Things and Big Data March 2018* (Funchal), 27–35.
- Semeniuta, S., Severyn, A., and Barth, E. (2016). Recurrent dropout without memory loss. *arXiv [Preprint]*. [arXiv:1603.05118](https://arxiv.org/abs/1603.05118).
- Sha, L., Lucey, P., Zheng, S., Kim, T., Yue, Y., and Sridharan, S. (2017). Fine-grained retrieval of sports plays using tree-based alignment of trajectories. *arXiv [Preprint]*. [arXiv:1710.02255](https://arxiv.org/abs/1710.02255).
- Shaw, L., and Glickman, M. (2019). “Dynamic analysis of team strategy in professional football,” in *Barca Sports Analytics Summit*, 1–13.
- Shaw, L., and Sudarshan, G. (2020). “Routine inspection: A playbook for corner kicks,” in *International Workshop on Machine Learning and Data Mining for Sports Analytics* (Cham: Springer).

- Siddharth, N., Paige, B., Van de Meent, J.-W., Desmaison, A., Goodman, N. D., Kohli, P., et al. (2017). Learning disentangled representations with semi-supervised deep generative models. *arXiv [Preprint]*. *arXiv:1706.00400*.
- Srivastava, N., Mansimov, E., and Salakhudinov, R. (2015). “Unsupervised learning of video representations using lstms,” in *International Conference on Machine Learning* (Lille: PMLR), 843–852.
- Stein, M., Seebacher, D., Karge, T., Polk, T., Grossniklaus, M., and Keimet, D. A. (2019). From movement to events: improving soccer match annotations. *Lecture Notes Comput. Sci.* 11295, 130–142. doi: 10.1007/978-3-030-05710-5247_11
- Sun, C., Karlsson, P., Wu, J., Tenenbaum, J. B., and Murphy, K. (2019). Stochastic prediction of multi-agent interactions from partial observations. *arXiv [Preprint]*. *arXiv:1902.09641*.
- Teng, M., Le, T. A., Scibior, A., and Wood, F. (2020). Semi-supervised sequential generative models. *arXiv [Preprint]*. *arXiv:2007.00155*.
- Tieleman, T., and Hinton, G. (2012). “Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude,” in *COURSERA: Neural Networks for Machine Learning, Vol. 4*, 26–31.
- Tolstikhin, I., Bousquet, O., Gelly, S., and Schoelkopf, B. (2017). Wasserstein auto-encoders. *arXiv [Preprint]*. *arXiv:1711.01558*.
- Van der Maaten, L., and Hinton, G. (2008). Visualizing data using t-sne. *J. Mach. Learn. Res.* 9, 2579–2605.
- Wickramaratna, K., Chen, M., Chen, S.-C., and Shyu, M.-L. (2005). “Neural network based framework for goal event detection in soccer videos,” in *Proceedings-Seventh IEEE International Symposium on Multimedia* (Irvine, CA: IEEE), 21–28.
- Xu, B., Wang, N., Chen, T., and Li, M. (2015). Empirical evaluation of rectified activations in convolutional network. *arXiv [Preprint]*. *arXiv:1505.00853*.
- Yeh, R. A., Schwing, A. G., Huang, J., and Murphy, K. (2019). “Diverse generation for multi-agent sports games,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (Long Beach, CA), 4610–4619.
- Zheng, M., and Kudenko, D. (2010). Automated event recognition for football commentary generation. *Int. J. Gaming Comput. Mediated Simulat.* 2, 67–84. doi: 10.4018/jgcms.2010100105

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher’s Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright © 2021 Fassmeyer, Anzer, Bauer and Brefeld. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.