Latent Variable Models for Multi-Agent Trajectories

Anonymous Author(s)
Affiliation
Address
email

Abstract

The supervised identification of spatiotemporal patterns usually requires a metric over time and space and a great deal of annotated situations. To overcome these limitations, we present a generative framework for modeling and annotating trajectories of multiple agents over time and space. Our approach leverages semi-supervised learning for variational autoencoders where equivariance and interaction of agents are appropriately represented by graph networks. The proposed approach is generally applicable to a wide range of problems, including generation, classification, and pattern detection. Empirically, our approach outperforms all competitors on various interactive spatiotemporal problems.

1 Introduction

Analyzing the spatiotemporal behavior of multiple agents bears a great deal of value in many domains such as autonomous driving [48, 13], public transportation [30, 57], migration [38, 22] and sports analytics [52]. Particularly, the detection of collective patterns across time and space is of interest to many communities but non-trivial inter-dependencies between agents render estimating the involved multi-modal distributions often difficult.

Existing approaches to spatiotemporal pattern detection thus capitalize on variants from graph networks [54, 12] and variational autoencoders [26, 7]; being purely unsupervised, their functionality is grounded in uncovering the implicit modular structure in agent-wise latent variables directly from the observed trajectories via approximate inference [28, 51, 58, 16, 14, 32]. Although a generative view on the problem is very appealing, approaches in this regime either lack interpretable and controllable latent factors (e.g., [58]), and are thus limited in practical applicability, or put a strong focus on relationship discovery (e.g., [28]), which limits their generative capacity.

An alternative is offered by supervised approaches that aim to counteract these issues by incorporating domain-knowledge in form of discrete labels into the learning process [62]. Ideally, these labels encode intents or goals of the agents. In practical applications, however, such annotations hardly exist. Instead of arguing in favor of expensive and tedious manual annotations of spatiotemporal data, recent approaches propose to incorporate heuristic surrogates as makeshift labels into the problem and report impressive predictive accuracies [63, 62, 39]. However, these weakly supervised approaches require the existence of label sequences for all training instances and cannot include inexpensive and possibly abundant unlabeled data due to their purely supervised nature.

To close this gap, we propose semi-supervised variational autoencoders for spatiotemporal multi-agent problems to process (weakly) labeled as well as unlabeled data. Our contribution generalizes the class of semi-supervised variational autoencoders [27] to spatiotemporal domains by using ideas from variational recurrent neural networks [7]. Additional adaptive graph network layers provide order in- and equivariance of agents and render the proposed approach appropriate for multi-agent scenarios. The resulting semi-supervised approach is applicable to a wide range of problems including the generation of collective movements as well as classifying situations of interest. Empirically,

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

our contribution significantly advances the state-of-the art in generation and classification tasks involving multiple agents.

2 Related Work

Semi-supervised generative models Semi-supervised variational autoencoders have been originally studied by Kingma et al. [27] who propose to incorporate additional labels in latent space. Related approaches introduce auxiliary variables that leave the original model unchanged but increase the flexibility of the variational posterior [33]. Joy et al. [24] propose to explicitly capture label characteristics in latent space instead of the label values themselves and Narayanaswamy et al. [40] study generalizations that allow for learning more complex dependency structures. However, all the above approaches are introduced only for static domains. Our contribution constitutes a generalization of the above approaches to spatiotemporal domains.

Sequential generative models Approaches designed for predicting agent trajectories frequently leverage ideas from sequential generative models to address the stochasticity inherent in future predictions. Extensions of variational autoencoders in this respect include STORN [3], VRNN [7], SRNN [11], or Z-Forcing [15]. Particularly, the VRNN [7] is related to our contribution as both deploy a VAE instance per time step conditioned on a recurrent state to capture temporal dependencies. Other approaches associate a single global latent variable with each sequence [4, 17, 9, 8].

Graph neural networks Deep graph-based (GNNs) approaches [49] are a natural choice for applications requiring permutation invariance of the inputs. These methods represent inputs as graph nodes where connecting edges capture interactions between the nodes. GNNs operate by learning a chain of hidden representations for each node through an iterative process that relies on aggregating messages along edges in each network layer. Prominent instantiations such as GCN [29], GraphSAGE [18], GAT [54], and others [12, 53, 20, 2, 61] mainly differ in their notion of message passing and neighborhood aggregation [59]. For example, GATs incorporate directoral biases into the aggregation via attention mechanisms. GNN layers can be stacked and allow for context-based and adaptive neighborhoods for each node by incorporating particularly designed skip connections [56]. See Chen et al. [5] for a discussion on the function approximation capabilities of this neural network family.

Generative models for sequential multi-agent data In the remainder, we empirically compare our approach with state-of-the-art methods in spatiotemporal pattern detection for multiple agents. Given the potential benefit across various domains, a great deal of recent publications focus on movements of pedestrians or self-driving cars [36, 48, 13, 37, 60, 47, 1, 21]. Makansi et al. [35], however, show that these standard benchmark datasets generally exhibit weak social interaction patterns, so in the remainder, we focus on methods using team sport data. Related approaches show that learning and computing realistic rollouts significantly benefit from the incorporation of makeshift annotations and heuristic surrogates into the problem [63, 62, 39]. Felsen et al. [10] and Li et al. [32] propose an architecture based on conditional VAEs [50] to model distributions of future movements of basketball players. Yeh et al. [58] and Sun et al. [51] combine ideas from graph networks [12] and VRNNs [7] into a unified framework aiming at modeling data from basketball and soccer. Other related approaches [31, 28, 16, 14] aim to explicitly infer discrete latent variables to represent interaction types while executing a trajectory prediction task.

Hence, our contribution constitutes the first semi-supervised generalization of variational autoencoders for spatiotemporal domains that lifts applicability to a wide range of prediction problems and gives rise to addressing tasks beyond representational or generative modeling.

3 Main Contribution

3.1 Variational Autoencoders

Variational autoencoders (VAEs) [26, 42] allow to capture intrinsic conditional dependencies and are often used in structured problems. Essentially, they consist of a θ-parameterized generative model $p_\theta(\boldsymbol{x}, \boldsymbol{z}) = p(\boldsymbol{z})p_\theta(\boldsymbol{x}|\boldsymbol{z})$ and an associated ϕ-parameterized variational distribution $q_\phi(\boldsymbol{z}|\boldsymbol{x})$.

Optimal parameters θ are computed by maximizing the variational lower bound on the marginal likelihood, given by

$$\log p_\theta(\boldsymbol{x}) \geq \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x})}[\log p_\theta(\boldsymbol{x}|\boldsymbol{z})] - \mathcal{KL}[q_\phi(\boldsymbol{z}|\boldsymbol{x}) \parallel p(\boldsymbol{z})].$$

It is often helpful to augment discrete labels y into the model to guide the generating process [27]. The resulting model $p(\boldsymbol{x}, \boldsymbol{z}, y) = p(\boldsymbol{x}|\boldsymbol{z}, y)p(y)p(\boldsymbol{z})$ then acts on labeled and unlabeled instances. Consequentially, the variational distribution $q_\phi(y, \boldsymbol{z}|\boldsymbol{x}) = q_\phi(y|\boldsymbol{x})q_\phi(\boldsymbol{z}|y, \boldsymbol{x})$ needs to be defined over both quantities, latents $\boldsymbol{z}$ and labels y. Semi-supervised VAEs are trained by maximizing

$$\sum_{(\boldsymbol{x},y)} \big(\mathcal{L}(\boldsymbol{x}, y) + \lambda \log q_\phi(y|\boldsymbol{x})\big) + \sum_{\boldsymbol{x}} \mathcal{U}(\boldsymbol{x}),$$

where $\mathcal{L}(\boldsymbol{x}, y) = \mathbb{E}_{q_\phi(\boldsymbol{z}|\boldsymbol{x},y)}[\log p_\theta(\boldsymbol{x}|y, \boldsymbol{z}) + \log p(y) + \log p(\boldsymbol{z}) - \log q_\phi(\boldsymbol{z}|\boldsymbol{x}, y)]$ and $\mathcal{U}(\boldsymbol{x}) = \sum_y q_\phi(y|\boldsymbol{x})(\mathcal{L}(\boldsymbol{x}, y)) + \mathcal{H}(q_\phi(y|\boldsymbol{x}))$ denote (variational) lower bounds on labeled and unlabeled instances, respectively, and λ is a balancing term. In fact, the original motivation for this framework was inspired by semi-supervised classification tasks. Meanwhile, their contribution is also used for learning meaningful representations and generating new data, thus rendering it a suitable methodological foundation for performing tasks with all sorts of data.

3.2 Problem Formulation

We are interested in modeling spatiotemporal multi-agent scenarios and focus on positions $\boldsymbol{x}_t^{(a)} \in \mathbb{R}^2$ of an agent $a \in \mathcal{A}$ over time $1 \leq t \leq T$, denoted by $\boldsymbol{x}_{\leq T}^{(a)}$. We assume that a non-empty subset of agents in $\mathcal{A}$ is present in any observed segment and collect their trajectories (in random order) to form a collective movement $\boldsymbol{x}_{\leq T} := \{\boldsymbol{x}_{\leq T}^{(a)} \mid a \in \mathcal{A}\}$. We will make use of velocities $d\boldsymbol{x}/dt$ and linearized motion $\Delta\boldsymbol{x}_t = \boldsymbol{x}_{t'} - \boldsymbol{x}_t$, for $t' > t$, of agents in the remainder.

In addition, we introduce discrete labels $y_t^{(a)} \in \mathcal{Y}$ associated with agent $a \in \mathcal{A}$ at time t. The collection of sequences $\boldsymbol{y}_{\leq T} := \{\boldsymbol{y}_{\leq T}^{(a)} \mid a \in \mathcal{A}\}$ may be best thought of as discriminative causes of variation in the corresponding trajectories $\boldsymbol{x}_{\leq T}$. In practice, $\boldsymbol{y}_{\leq T}$ may contain anything that aids in generating multi-agent rollouts, including behavioral indicators, long-term goals, or current tasks. Finally, observations $\boldsymbol{x}_{\leq T}$ accompanied by labels $\boldsymbol{y}_{\leq T}$ assemble the labeled share of the data, denoted by $\mathcal{D}_L = \{(\boldsymbol{x}_{\leq T}, \boldsymbol{y}_{\leq T})_n\}_{1 \leq n \leq N}$, while unlabeled observations are collected in $\mathcal{D}_U = \{(\boldsymbol{x}_{\leq T})_m\}_{1 \leq m \leq M}$. We refer to all data by $\mathcal{D} = \mathcal{D}_L \cup \mathcal{D}_U$.

We aim at estimating the underlying distribution that generated the observations in $\mathcal{D}$. A straight forward way is to simply maximize the log likelihood given by

$$\log p(\mathcal{D}) = \sum_{\mathcal{D}_L} \log p_\theta(\boldsymbol{x}_{\leq T}, \boldsymbol{y}_{\leq T}) + \sum_{\mathcal{D}_U} \log p_\theta(\boldsymbol{x}_{\leq T}) \tag{1}$$

However, human movement is inherently multi-modal [anonymized reference] which renders maximum likelihood clearly inappropriate. It is therefore common to model the stochasticity inherent in future agent behaviors via latent variables. To lower bound the objective, we need to derive lower bounds on $\log p_\theta(\boldsymbol{x}_{\leq T})$ and $\log p_\theta(\boldsymbol{x}_{\leq T}, \boldsymbol{y}_{\leq T})$ as follows.

3.3 VAEs for Multi-Agent Trajectories

We associate disentangled latent realizations $\{\boldsymbol{z}_t, y_t\}$ with each time step t of the segment while injecting the dependency structures introduced in Section 3.1 for the generative and inference parts. To model temporal dependencies, we propose to additionally condition the model components on the past observations $\boldsymbol{x}_{<t}$ and latent variables $\boldsymbol{z}_{<t}$ and $\boldsymbol{y}_{<t}$ (cmp. [7]). The joint distribution factorizes into

$$p_\theta(\boldsymbol{x}_{\leq T}, \boldsymbol{z}_{\leq T}, \boldsymbol{y}_{\leq T}) = \prod_t \Big[p_\theta(\boldsymbol{x}_t|\boldsymbol{x}_{<t}, \boldsymbol{z}_{\leq t}, \boldsymbol{y}_{\leq t})p_\theta(\boldsymbol{z}_t|\boldsymbol{x}_{<t}, \boldsymbol{z}_{<t}, \boldsymbol{y}_{<t})p_\theta(\boldsymbol{y}_t|\boldsymbol{x}_{<t}, \boldsymbol{z}_{<t}, \boldsymbol{y}_{<t})\Big],$$

with generating distribution $p_\theta(\boldsymbol{x}_t|\boldsymbol{x}_{<t}, \boldsymbol{z}_{\leq t}, \boldsymbol{y}_{\leq t})$ and prior distributions $p_\theta(\boldsymbol{z}_t|\boldsymbol{x}_{<t}, \boldsymbol{z}_{<t}, \boldsymbol{y}_{<t})$ and $p_\theta(\boldsymbol{y}_t|\boldsymbol{x}_{<t}, \boldsymbol{z}_{<t}, \boldsymbol{y}_{<t})$ for latents $\boldsymbol{z}_t$ and $\tilde{\boldsymbol{y}}_t$. Note that we differentiate between observed labels $\boldsymbol{y}$ and latent variables $\tilde{\boldsymbol{y}}$.

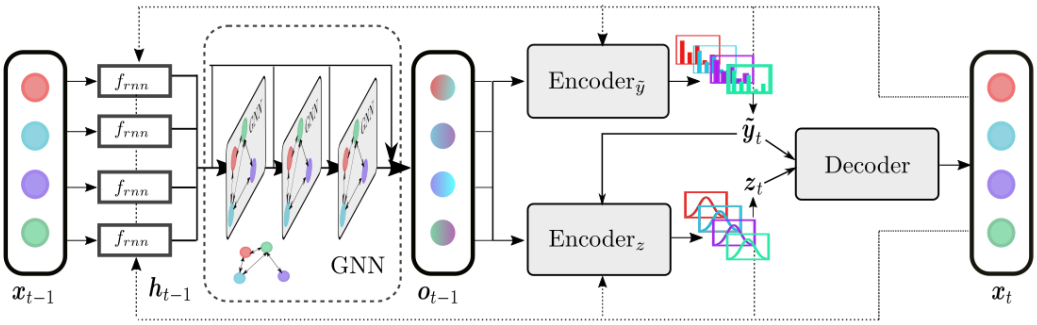

Figure 1: High-level overview of the network computations.

Also note that in our sequential setting, the priors must be included in the optimization rather than being represented by fixed distributions to capture temporal dependencies between subsequent latent realizations. This enables the prior samples to provide valuable information on the agents' future movements at inference time. Hence, the surrogate posteriors q_ϕ also need to condition on previous observations and random variables to allow for accurate approximations of the true posterior and provide a tractable lower bound on the log-likelihood in Eqn (1). Using the factorization introduced in Section 3.1 yields

$$q_\phi(\boldsymbol{z}_{\leq T}|\boldsymbol{x}_{\leq T}, \boldsymbol{y}_{\leq T}) = \prod_t q_\phi(\boldsymbol{z}_t|\boldsymbol{x}_{\leq t}, \boldsymbol{z}_{<t}, \boldsymbol{y}_{\leq t}) \text{ and } q_\phi(\tilde{\boldsymbol{y}}_{\leq T}|\boldsymbol{x}_{\leq T}) = \prod_t q_\phi(\tilde{y}_t|\boldsymbol{x}_{\leq t}, \boldsymbol{z}_{<t}, \boldsymbol{y}_{<t}). \tag{2}$$

Labeled instances render reasoning over latents $\tilde{\boldsymbol{y}}_{\leq T}$ unnecessary since the true $\boldsymbol{y}_{\leq T}$ is already known. Hence, the right-hand side in Eqn (2) can be ignored when sampling from $\mathcal{D}_L$. We arrive at the following results and refer to Appendix A for the proofs.

Theorem 1. *A lower bound on* $\log p_\theta(\boldsymbol{x}_{\leq T}, \boldsymbol{y}_{\leq T})$ *in Eqn* (1) *is given by*

$$\log p_\theta(\boldsymbol{x}_{\leq T}, \boldsymbol{y}_{\leq T}) \geq \sum_t \log p_\theta(y_t|\boldsymbol{x}_{<t}, \boldsymbol{z}_{<t}, \boldsymbol{y}_{<t}) + \mathbb{E}_{q_\phi(\boldsymbol{z}_t|\boldsymbol{x}_{\leq t}, \boldsymbol{z}_{<t}, \boldsymbol{y}_{\leq t})}\Big[\log p_\theta(\boldsymbol{x}_t|\boldsymbol{x}_{<t}, , \boldsymbol{z}_{\leq t}, \boldsymbol{y}_{\leq t})\Big]$$
$$- \mathcal{KL}[q_\phi(\boldsymbol{z}_t|\boldsymbol{x}_{\leq t}, \boldsymbol{z}_{<t}, \boldsymbol{y}_{\leq t}) \parallel p_\theta(\boldsymbol{z}_t|\boldsymbol{x}_{<t}, \boldsymbol{z}_{<t}, \boldsymbol{y}_{<t})] \equiv \sum_{t=1}^{T} -\mathcal{L}(\boldsymbol{x}_t, \boldsymbol{y}_t). \tag{3}$$

Correspondingly, for unlabeled instances drawn from $\mathcal{D}_U$, sequential label information need to be estimated as shown in the following theorem.

Theorem 2. *Let* $\mathcal{H}(\beta)$ *be the entropy of quantity* β*. A lower bound on* $\log p_\theta(\boldsymbol{x}_{\leq T})$ *in Eqn* (1) *is given by*

$$\log p_\theta(\boldsymbol{x}_{\leq T}) \geq \sum_t \Big(\mathcal{H}\big(q_\phi(\tilde{\boldsymbol{y}}_t|\boldsymbol{x}_{\leq t}, \boldsymbol{z}_{<t}, \boldsymbol{y}_{<t})\big) - \mathbb{E}_{q_\phi(\tilde{\boldsymbol{y}}_t|\boldsymbol{x}_{\leq t}, \boldsymbol{z}_{<t}, \tilde{\boldsymbol{y}}_{<t})}\big[\mathcal{L}(\boldsymbol{x}_t, \tilde{\boldsymbol{y}}_t)\big]\Big) \equiv \sum_t -\mathcal{U}(\boldsymbol{x}_t) \tag{4}$$

The full evidence lower bound (ELBO) is obtained by combining the lower bounds for all data $\mathcal{D}$,

$$p(\mathcal{D}) \geq \sum_{\mathcal{D}_L} \sum_t -\mathcal{L}(\boldsymbol{x}_t, \boldsymbol{y}_t) + \sum_{\mathcal{D}_U} \sum_{t=1}^{T} -\mathcal{U}(\boldsymbol{x}_t). \tag{5}$$

This formalization focuses on either fully labeled or unlabeled observations, however, a more general formulation can be obtained to also allowing for partially labeled observations.

3.4 Semi-supervised Sequential Learning

Similar to inferring label sequences for unlabeled examples, our contribution leverages to sequence annotation tasks where posterior $q_\phi(\boldsymbol{y}_{\leq T}|\boldsymbol{x}_{\leq T})$ acts as a classification network annotating new

observations. However, the ELBO in Eqn (5) is oblivious to classification and simply ignores relevant gradient updates for $\mathcal{L}(\boldsymbol{x}_t, \boldsymbol{y}_t)$ by discarding Eqn (2). This is clearly inappropriate for semi-supervised learning where the overall goal is to learn an unknown mapping $\boldsymbol{x} \mapsto \boldsymbol{y}$. Following Kingma et al. [27], we circumvent this by heuristically incorporating Eqn (2) into the label-dependent objective. Hence, the full objective to learn $\{\theta, \phi\}$ is given by

$$\mathcal{J}_{MAT}(\theta, \phi; \mathcal{D}) = \lambda_0 \sum_{\mathcal{D}_U} \sum_t \mathcal{U}(\boldsymbol{x}_t) + \sum_{\mathcal{D}_L} \sum_t \Big(\mathcal{L}(\boldsymbol{x}_t, \boldsymbol{y}_t) - \lambda_1 \log q_\phi(y_t | \boldsymbol{x}_{\leq t}, \boldsymbol{z}_{<t}, \boldsymbol{y}_{<t}) \Big), \quad (6)$$

where λ_1 balances generative and discriminative learning and λ_0 the contribution of labeled and unlabeled data, respectively. Thus, the semi-supervised variant contains fully (un)supervised modeling tasks as special cases via adjusting λ_1, λ_2 accordingly. Note that the supervised target differs from its unsupervised counterpart only in observing $\boldsymbol{y}_t$ instead of factorizing over $\mathcal{Y}$, which results in computing the log-likelihood (auxiliary loss) instead of the entropy $\mathcal{H}(q_\phi(\boldsymbol{y}_t | \boldsymbol{x}_{\leq t}, \boldsymbol{z}_{<t}, \boldsymbol{y}_{<t}))$.

Eqn (4) encodes the behavior of classifier $q_\phi(\boldsymbol{y}_{\leq T} | \boldsymbol{x}_{\leq T})$ when dealing with unsupervised data. Since $\log p_\theta(\boldsymbol{x}_t | \boldsymbol{x}_{<t}, , \boldsymbol{z}_{\leq t}, y_{\leq t})$ is a main contributing factor in $\mathcal{L}(\boldsymbol{x}_t, y_t)$, optimizing for ϕ encourages the model to choose a high probability mass for label $y \in \mathcal{Y}$ that incurs the smallest reconstruction loss compared to other candidates in label space. This provides an effective way to learn from unobserved data.

3.5 Design Choices

In Section 3.2, we argued that not all agents need to be present in every example and we simply added agents to training instances in arbitrary and possibly random order. However, order-agnostic methods rely on the independence of agent trajectories, which is clearly inappropriate for multi-agent scenarios where future movements need to be coordinated with other agents. We thus require the generative model to be in- or equivariant wrt. agent permutations and employ deep graph-neural networks (GNNs) [49] as a methodological remedy to account for both, equivariance and agent interactions, respectively.

We process every agent's trajectory $\boldsymbol{x}_{\leq T}^{(a)}$ individually and assume an underlying connecting graph $G = (\mathcal{A}, E)$ at each timestep, where edges in E encode direct agent dependencies. Since agent-level predictions mainly rely on (spatially) nearby agents, we neglect direct influences from distant peers; however, this choice may be problem dependent. Thus, while the set of agents $\mathcal{A}$ remains constant across the temporal dimension, the composition of edges E changes dynamically, depending on the k-nearest agents $\boldsymbol{x}_t^{(a)}$.

Note that eradicating remote agents in the input graph does not curtail their indirect influence on an agent of interest. By adding depth to the network through stacking up GNN layers, the receptive field is increased and thus also the number of nodes that affect the computations. Skip-like connections proposed by Xu et al. [56] aggregate the final feature representation over all agent embeddings of previous layers. This mechanism adds to the expressiveness of the network as it allows to adaptively accommodate structural information from different levels of granularity for each agent. In team sports, for example, agent nodes are able to capture immediate player influences as well as holistic team strategies/formations dependent on the interaction structure best suited at prediction time. To further narrow down the training process towards important pieces of data, every GNN layer uses GAT update equations [54], implicitly assigning different weights to different nodes in a neighborhood.

There are various possibilities to incorporate previously described GNNs in the overall scheme[1]. An efficient way to model the distributions defining Eqn (6) is to employ hidden states for agents $\boldsymbol{h}_t^{(a)}$ that encapsulate the past realizations of all random variables as node representations,

$$\boldsymbol{h}_t^{(a)} = f_{rnn}(\varphi^{\boldsymbol{x}}(\boldsymbol{x}_t^{(a)}), \varphi^{\boldsymbol{z}}(\boldsymbol{z}_t^{(a)}), \varphi^{\boldsymbol{y}}(\boldsymbol{y}_t^{(a)}), \boldsymbol{o}_{t-1}^{(a)}) \quad \text{with} \quad \boldsymbol{o}_t = \text{GNN}(\boldsymbol{h}_t),$$

where f_{rnn} denotes an RNN transition function, GNN is the described GAT-based GNN, the φ's are nonlinear function approximators, and $\boldsymbol{o}_t = \{\boldsymbol{o}_t^{(a)} \mid a \in \mathcal{A}\}$ refers to the set of updated agent representations. Since the elements of $\boldsymbol{o}_t$ additionally encode neighboring information, the resulting

[1] We empirically evaluate different configurations in Appendix D.

individual feature vectors can be used to describe the entire interactive past. While it is reasonable to enforce a dependency structure on the latent variables at time t via encoding and decoding GNN modules, we argue that capturing past agent interactions is sufficiently informative for high-frequency data [45]. Hence, assuming conditional independence within a single timestep, the joint movement distribution factorizes across agent dimensions and is given by

$$p_\theta(\boldsymbol{x}_t|\boldsymbol{x}_{<t}, \boldsymbol{z}_{\leq T}, \boldsymbol{y}_{\leq T}) = p_\theta(\boldsymbol{x}_t|\psi_t) = \prod_{a\in\mathcal{A}} p_\theta(\boldsymbol{x}_t^{(a)}|\psi_t^{(a)})$$

where $\psi_t = \{\psi_t^{(a)} \mid a \in \mathcal{A}\}$ are the distribution parameters where $\psi_t^{(a)} = \varphi(\boldsymbol{z}_t^{(a)}, \boldsymbol{y}_t^{(a)}, \boldsymbol{o}_{t-1}^{(a)})$. The overall logic is schematically depicted in Fig. 1.

4 Empirical Evaluation

For evaluation, we focus on team sports as the coordination of players renders these tasks more difficult than other domains [35].[2] Hence, we experiment on STATS SportVU NBA[3] for comparison, and tracking data from elite soccer[4]. As detailed in Section 3.2, we use agent velocities as input to all models and assume linear motion between consecutive observations. For NBA, we adopt the experimental setup and processing strategy from Monti et al. [39][5].

The STATS SportVU NBA data comprises tracking positions of offensive plays from the 2016 NBA regular season covering more than 1200 different games where game segment are given by sequences of length 50 and contain two-dimensional positions of all agents (10 players and ball) sampled at 5 frames per second. The data is split into 60% training, 20% validation, and 20% test sets. All data is translated so that the origin of the underlying coordinate system is mapped onto the top-left corner.

The soccer data consists of 12 matches where positions of players and ball are sampled at 25 frames per second. The tracking data is accompanied by manual event annotations that we will also make use of in the remainder. Models are trained on eight matches, the remaining six are distributed evenly into validation and test data (three matches each).

4.1 Baselines

We evaluate versus several baselines on two distinct tasks: *trajectory forecasting* (Section 4.2 and 4.4) and *spatiotemporal classification* (Section 4.3 and 4.4). Since our approach constitutes the first semi-supervised model for label predictions in multi-agent scenarios, we need to rely on self-constructed supervised baselines for scenarios where only a few labels are available, which we discuss in more detail in the relevant sections.

By contrast, trajectory forecasting is an active topic in different fields (see, e.g., [44] for an overview), which allows comparisons with a wide range of baseline models. Fully supervised generative models such as *DAG-Net* [39] and *Weak-Sup* [62] use weak labels in the form of agent objectives to improve trajectory prediction. In these settings, an agent's objective at each timestep t is to reach the next area where the observed movement speed falls below a predefined threshold. Adopting the experimental framework of Monti et al. [39] allows us to make further comparisons with *Social-Ways* [1] and *STGAT* [21] without additional experiments. STGAT [21] augments a standard GAT with an LSTM [19] to capture both spatial and temporal dependencies of social interactions while Social-Ways uses Info-GAN to learn multimodal predictive distributions.

It is difficult to determine the current state-of-the-art in generative baselines for sports tracking data because there is no consistent setting in terms of forecasting horizon or data (see e.g., [32, 39]). Hence, we additionally benchmark against the most recent and methodologically most relevant work described below. *GVRNN* [58] is a graph extension of the VRNN for explicitly modeling joint agent movements for basketball and soccer data. *dNRI* [16] dynamically extracts interaction types that determine the parametrization of the decoder. Finally, *GRIN* [32] recovers (static) interactions via attention using a disentangled latent space.

[2] We report on results on Stanford Drone Data (SDD) [43] in the Appendix.

[3] `https://github.com/alexmonti19/dagnet/tree/master/datasets`

[4] Details omitted for anonymity

[5] The source code will be made publicly available after acceptance of the paper.

Table 1: Error for NBA in meters for a prediction interval of 10 timesteps with an observation period of 40 timesteps. Bold is the lowest avg L_2 & final L_2 for the supervised and unsupervised generative models, respectively.

	GVRNN	dNRI	GRIN	DAG-Net.	U-MAT	S-MAT
avg L_2	2.60	2.77	3.00	2.10	***2.31***	**1.94**
final L_2	5.66	5.52	6.12	4.28	***4.80***	**3.99**

Table 2: Error for NBA in meters for a prediction interval of 40 timesteps with an observation period of 10 timesteps. Top rows show results for offense, bottom rows the defense. Bold is the lowest avg L_2 & final L_2 for the supervised and unsupervised generative models, respectively.

	STGAT	Social-Ways	GVRNN	Weak-Sup.	DAG-Net	U-MAT	S-MAT
avg L_2	9.94	9.91	9.73	9.47	8.98	***9.01***	**8.11**
final L_2	15.80	15.19	15.89	16.98	14.08	***13.28***	**12.52**
avg L_2	7.26	7.31	7.29	7.05	6.87	***6.88***	**6.21**
final L_2	11.28	10.21	10.62	10.56	9.76	***9.04***	**8.45**

4.2 Exploring Boundary Cases for Trajectory Prediction

Methods We study the purely (un)supervised generation of trajectories by maximizing the respective lower bounds on NBA. For the fully-supervised variant, the weak labels described in the previous section can be naturally integrated into the overall scheme via treating them as semantic concepts $y_t^{(a)}$. These target areas naturally vary over time so that the sequential label $\boldsymbol{y}_{<T}$ denotes always the next desired position for every involved agent at time t. Following related work, we obtain 90 different areas/class labels. The underlying data distribution is estimated by maximizing Eqn (3) via setting $\lambda_0 = \lambda_1 = 0$ in Eqn (6). The supervised variant of our contribution is denoted by *S-MAT*.

Although done in [39, 62], direct comparison of unsupervised and supervised generative models is inappropriate inasmuch as the latter use their (weakly) generated labels over the entire observation period (which include future information) at prediction time. Thus, we additionally benchmark against a fully unsupervised instantiation of our proposed framework lacking ground-truth information for structuring the discrete latent subspace. To encourage the model to learn semantically meaningful concepts describing distinct movement patterns, we parametrize the decoder dependent on the predicted agent class $\tilde{y}_t^{(a)}$. In this context, $\tilde{y}_t^{(a)}$ may be best thought of dynamically assigned agent roles that accurately explain the observed trajectories. We maximize Eqn (4) using 3 agent types and refer to the unsupervised variant as *U-MAT*.

Metrics We measure the quality of the generated trajectories by their *average* L_2 distance between predicted $\hat{\boldsymbol{x}}_t^{(a)}$ and observed positions $\boldsymbol{x}_t^{(a)}$ over the entire sequence given by

$$L_2 = \frac{1}{T \cdot |\mathcal{A}|} \sum_{t=1}^{T} \sum_{a \in \mathcal{A}} \|\hat{\boldsymbol{x}}_t^{(a)} - \boldsymbol{x}_t^{(a)}\|_2$$

and report the best result out of 20 generated samples for every instance. In addition, we give an account of the *final* L_2 distance which follows the same logic but evaluates only the difference of the final timestep T in the test sequences. We report all metrics on un-normalized scales.

Results Tables 1 and 2 show the results. The proposed approaches clearly outperform their peers and realize the lowest average L_2 and final L_2 distances. The *MAT* variants effectively accommodate not only mutual influences across agents, but also discrete generative factors, leading to better approximations of the underlying multi-modal distributions. Unsurprisingly, Table 1 also shows that defending players exhibit more structure and are easier to predict than offensive players. The results of an ablation study are contained in Appendix D. The right part of Fig. 2 displays exemplary rollouts.

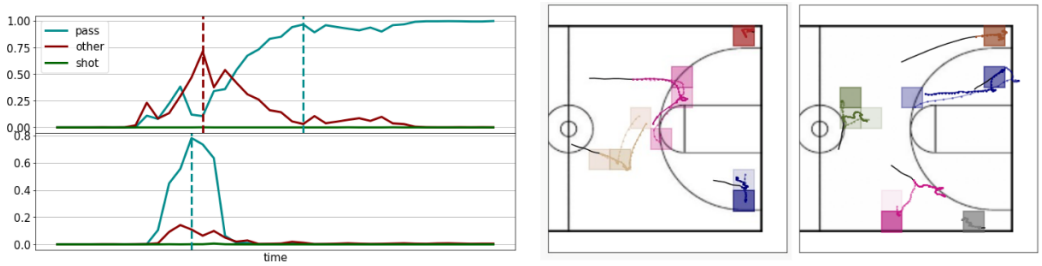

Figure 2: *Left:* Prediction probabilities for different events. The dashed lines correspond to the human annotations. *Right*: Exemplary rollouts for offensive players. The figure shows observed (light color) and generated player trajectories (intense color) as well as label information indicated by colored boxes, where the color intensity corresponds to the ground-truth frequency of the (weakly obtained) location-based labels.

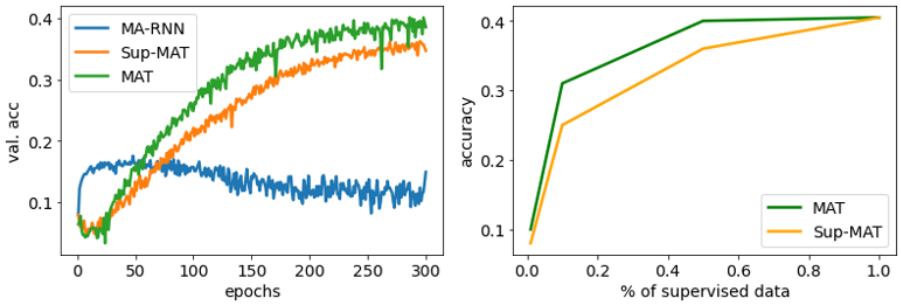

Figure 3: Semi-supervised results on NBA. *Right*: Progression of accuracy during a training run. *Left*: Test accuracies for varying amount of labeled data.

4.3 Semi-Supervised Classification

We now extend the experimental protocol by targeting accurate label discovery in semi-supervised scenarios. We refer to the full model simply as *MAT* where $\lambda_0, \lambda_1 > 0$ and incorporate the fully supervised *S-MAT* ($\lambda_0 = 0$) as an additional baseline. We also compare to an RNN-based classification network that addresses inter-agent dependencies by inferring hidden states of agents by the GNN architecture described in Section 3.5. A final softmax layer is used for classification and the model is optimized by minimizing the negative log-likelihood. We refer to this baseline as *MA-RNN*.

The semi-supervised *MAT* is trained on 50% labeled and 50% unlabeled examples while the supervised baselines have only access to the labeled part. Figure 3 shows the performance of the three models on NBA. The semi-supervised *MAT* clearly emerges as the strongest classifier and distinguishes by the lowest long-term generalization error. Comparing the numbers to its supervised peer *S-MAT* highlights the value of the unlabeled data that is effectively utilized by *MAT*. Comparing the fully supervised approaches *S-MAT* and *MA-RNN*, the importance of an effective regularization mechanism becomes obvious; *MA-RNN* quickly overfits while the lower-bound in S-MAT acts as a regularizer and guarantees gradually increasing predictive accuracies on validation sets.

4.4 Combining Generation and Classification in Semi-Supervised Settings

In our last set of experiments, we study the combination of trajectory generation and classification and resort to tracking data from elite soccer. The reason for favoring this data over NBA is that the soccer tracking data is manually annotated by experts from the data provider. Since the data requires human annotations for further processing, a successful semi-supervised application bears

Table 3: *Left*: Results for semi-supervised data generation. *Right*: Results for semi-supervised event detection.

NAME	L_2	FINAL L_2
VRNN	2.91	7.10
VRNN + GNN	2.87	6.98
S-MAT	2.87	6.91
MAT-DIAG	2.84	6.90
MAT	**2.81**	**6.88**

NAME	ACCURACY	F1
MA-RNN-DIAG	0.74	0.80
MA-RNN	0.85	0.88
S-MAT	0.82	0.85
MAT	**0.88**	**0.92**

a great deal of practical relevance by reducing the expensive labeling effort. We focus on labels $\mathcal{Y} = \{pass, other\ ball\ action, shot, none\}$, where class *none* denotes the absence of all other labels in a frame. Every label is propagated to the previous five and the subsequent 30 frames. We generate a balanced training set where half of the segments carry label *none* and study the combined task of trajectory generation and classification.

Generation For trajectory prediction, we compare *MAT* against *VRNN* [7] that implicitly assumes independence across agent dimensions, an interactive version of thereof with GNNs on the hidden states (*VRNN+GNN*), the supervised *S-MAT* with long-term goals, and a diagonal *MAT-Diag*. Note that the *VRNN* constitutes a strong competitor for the task at-hand, as it effectively corresponds to a diagonal GVRNN, which has been shown to produce competitive results on soccer data [58].

The left part of Table 3 shows the results. Although all competitors realize similar performances, the semi-supervised *MAT* generates trajectories that are closest in both metrics to the observed ones. *MAT* effectively exploits the additional expert annotations and performs significantly better than its (weakly) supervised peers.

Classification *MAT* learns a model over the label space $\mathcal{Y}$ simultaneously to the generative model. To evaluate classification performance, we turn the output of the corresponding softmax for a given frame into a class label whenever it exceeded a predefined threshold.We compare the prediction performance again to *MA-RNN*, a diagonal variant *MA-RNN-Diag*, and *S-MAT*.

The resulting accuracies and F1 scores are summarized in Table 3. Once more, *MAT* clearly beats the baselines and stands out by the highest accuracy and F1 scores. The result impressively demonstrates the benefit of including unlabeled data for the task. Surprisingly, *MA-RNN* performs better than *S-MAT*. This finding is in contrast to the results on NBA shown in Figure 3. However, soccer is played on a much larger pitch and movements are likely more linear compared to basketball, which leads to less multimodality of the distributions involved and a consequently decreased benefit of variational methods. Together with a significantly reduced label space and sufficient amounts of data this may lead to a simpler learning task where *MA-RNN* is less prone to overfitting.

The right part of Fig. 2 shows exemplary prediction probabilities for two segments and three possible events. In both segments, the algorithm is highly confident no shot-action will occur in the near future. The segment on top contains an event of the class *other ball action* followed by an event of class *pass*. Both are correctly identified by *MAT*. The former is clearly indicated by a peak in the corresponding probability chart which then decreases to give rise to the following *pass* action. The segment on the bottom shows solely a single *pass* event that is also clearly identifiable by a peak at the correct frame.

5 Conclusion

We presented semi-supervised variational autoencoders for spatiotemporal multi-agent scenarios. The proposed approach effectively combined ideas from semi-supervised variational autoencoders, variational recurrent neural networks, and graph neural networks. Empirically, our approach clearly outperformed the state-of-the-art in sequential generation and (semi-supervised) classification tasks in all experiments. The performance underlines the benefit of including unlabeled data in spatiotemporal problems where labeled sequences are either scarce or assembled from weak makeshift signals.

References

[1] Javad Amirian, Jean-Bernard Hayet, and Julien Pettré. Social ways: Learning multi-modal distributions of pedestrian trajectories with gans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[2] Peter W Battaglia, Razvan Pascanu, Matthew Lai, Danilo Rezende, and Koray Kavukcuoglu. Interaction networks for learning about objects, relations and physics. *arXiv preprint arXiv:1612.00222*, 2016.

[3] Justin Bayer and Christian Osendorfer. Learning stochastic recurrent networks. *arXiv preprint arXiv:1411.7610*, 2014.

[4] Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*, 2015.

[5] Zhengdao Chen, Soledad Villar, Lei Chen, and Joan Bruna. On the equivalence between graph isomorphism testing and function approximation with gnns. *Advances in neural information processing systems*, 32, 2019.

[6] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[7] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. *Advances in neural information processing systems*, 28:2980–2988, 2015.

[8] Andrew M Dai and Quoc V Le. Semi-supervised sequence learning. *Advances in neural information processing systems*, 28:3079–3087, 2015.

[9] Otto Fabius and Joost R Van Amersfoort. Variational recurrent auto-encoders. *arXiv preprint arXiv:1412.6581*, 2014.

[10] Panna Felsen, Patrick Lucey, and Sujoy Ganguly. 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)*, pages 732–747, 2018.

[11] Marco Fraccaro, Søren Kaae Sønderby, Ulrich Paquet, and Ole Winther. Sequential neural models with stochastic layers. *arXiv preprint arXiv:1605.07571*, 2016.

[12] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR, 2017.

[13] Roger Girgis, Florian Golemo, Felipe Codevilla, Jim Aldon D'Souza, Martin Weiss, Samira Ebrahimi Kahou, Felix Heide, and Christopher Pal. Autobots: Latent variable sequential set transformers. *arXiv preprint arXiv:2104.00563*, 2021.

[14] Dong Gong, Frederic Z Zhang, Javen Qinfeng Shi, and Anton van den Hengel. Memory-augmented dynamic neural relational inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11843–11852, 2021.

[15] Anirudh Goyal, Alessandro Sordoni, Marc-Alexandre Côté, Nan Rosemary Ke, and Yoshua Bengio. Z-forcing: Training stochastic recurrent networks. *arXiv preprint arXiv:1711.05411*, 2017.

[16] Colin Graber and Alexander Schwing. Dynamic neural relational inference for forecasting trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 1018–1019, 2020.

[17] David Ha and Douglas Eck. A neural representation of sketch drawings. *arXiv preprint arXiv:1704.03477*, 2017.

[18] William L Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 1025–1035, 2017.

[19] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9 (8):1735–1780, 1997.

[20] Yedid Hoshen. Vain: Attentional multi-agent predictive modeling. *arXiv preprint arXiv:1706.06122*, 2017.

[21] Yingfan Huang, HuiKun Bi, Zhaoxin Li, Tianlu Mao, and Zhaoqi Wang. Stgat: Modeling spatial-temporal interactions for human trajectory prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6272–6281, 2019.

[22] Franziska Hübl, Sreten Cvetojevic, Hartwig Hochmair, and Gernot Paulus. Analyzing refugee migration patterns using geo-tagged tweets. *ISPRS International Journal of Geo-Information*, 6(10):302, 2017.

[23] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[24] Tom Joy, Sebastian Schmon, Philip Torr, N Siddharth, and Tom Rainforth. Capturing label characteristics in vaes. In *International Conference on Learning Representations*, 2020.

[25] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[26] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[27] Diederik P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*, pages 3581–3589, 2014.

[28] Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. Neural relational inference for interacting systems. In *International Conference on Machine Learning*, pages 2688–2697. PMLR, 2018.

[29] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[30] John Krumm and Eric Horvitz. Predestination: Inferring destinations from partial trajectories. In *International Conference on Ubiquitous Computing*, pages 243–260. Springer, 2006.

[31] Jiachen Li, Fan Yang, Masayoshi Tomizuka, and Chiho Choi. Evolvegraph: Multi-agent trajectory prediction with dynamic relational reasoning. *arXiv preprint arXiv:2003.13924*, 2020.

[32] Longyuan Li, Jian Yao, Li Wenliang, Tong He, Tianjun Xiao, Junchi Yan, David Wipf, and Zheng Zhang. Grin: Generative relation and intention network for multi-agent trajectory prediction. *Advances in Neural Information Processing Systems*, 34, 2021.

[33] Lars Maaløe, Casper Kaae Sønderby, Søren Kaae Sønderby, and Ole Winther. Auxiliary deep generative models. In *International conference on machine learning*, pages 1445–1453. PMLR, 2016.

[34] Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

[35] Osama Makansi, Julius von Kügelgen, Francesco Locatello, Peter Gehler, Dominik Janzing, Thomas Brox, and Bernhard Schölkopf. You mostly walk alone: Analyzing feature attribution in trajectory prediction. *arXiv preprint arXiv:2110.05304*, 2021.

[36] Karttikeya Mangalam, Harshayu Girase, Shreyas Agarwal, Kuan-Hui Lee, Ehsan Adeli, Jitendra Malik, and Adrien Gaidon. It is not the journey but the destination: Endpoint conditioned trajectory prediction. In *European Conference on Computer Vision*, pages 759–776. Springer, 2020.

[37] Karttikeya Mangalam, Yang An, Harshayu Girase, and Jitendra Malik. From goals, waypoints & paths to long term human trajectory forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15233–15242, 2021.

[38] Patrick L McDermott, Christopher K Wikle, and Joshua Millspaugh. Hierarchical nonlinear spatio-temporal agent-based models for collective animal movement. *Journal of Agricultural, Biological and Environmental Statistics*, 22(3):294–312, 2017.

[39] Alessio Monti, Alessia Bertugli, Simone Calderara, and Rita Cucchiara. Dag-net: Double attentive graph neural network for trajectory forecasting. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 2551–2558. IEEE, 2021.

[40] Siddharth Narayanaswamy, Timothy Paige, Jan-Willem Van de Meent, Alban Desmaison, Noah Goodman, Pushmeet Kohli, Frank Wood, and Philip Torr. Learning disentangled representations with semi-supervised deep generative models. 2018.

[41] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[42] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pages 1278–1286. PMLR, 2014.

[43] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *European conference on computer vision*, pages 549–565. Springer, 2016.

[44] Andrey Rudenko, Luigi Palmieri, Michael Herman, Kris M Kitani, Dariu M Gavrila, and Kai O Arras. Human motion trajectory prediction: A survey. *The International Journal of Robotics Research*, 39(8):895–935, 2020.

[45] Yannick Rudolph and Ulf Brefeld. Modeling conditional dependencies in multiagent trajectories. In *International Conference on Artificial Intelligence and Statistics*, pages 10518–10533. PMLR, 2022.

[46] Amir Sadeghian, Vineet Kosaraju, Agrim Gupta, Silvio Savarese, and A Alahi. Trajnet: Towards a benchmark for human trajectory prediction. *arXiv preprint*, 2018.

[47] Amir Sadeghian, Vineet Kosaraju, Ali Sadeghian, Noriaki Hirose, Hamid Rezatofighi, and Silvio Savarese. Sophie: An attentive gan for predicting paths compliant to social and physical constraints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1349–1358, 2019.

[48] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16*, pages 683–700. Springer, 2020.

[49] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE transactions on neural networks*, 20(1):61–80, 2008.

[50] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28: 3483–3491, 2015.

[51] Chen Sun, Per Karlsson, Jiajun Wu, Joshua B Tenenbaum, and Kevin Murphy. Stochastic prediction of multi-agent interactions from partial observations. *arXiv preprint arXiv:1902.09641*, 2019.

[52] Karl Tuyls, Shayegan Omidshafiei, Paul Muller, Zhe Wang, Jerome Connor, Daniel Hennes, Ian Graham, William Spearman, Tim Waskett, Dafydd Steel, et al. Game plan: What ai can do for football, and what football can do for ai. *Journal of Artificial Intelligence Research*, 71: 41–88, 2021.

[53] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[54] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[55] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.

[56] Keyulu Xu, Chengtao Li, Yonglong Tian, Tomohiro Sonobe, Ken-ichi Kawarabayashi, and Stefanie Jegelka. Representation learning on graphs with jumping knowledge networks. In *International Conference on Machine Learning*, pages 5453–5462. PMLR, 2018.

[57] Andy Yuan Xue, Rui Zhang, Yu Zheng, Xing Xie, Jin Huang, and Zhenghua Xu. Destination prediction by sub-trajectory synthesis and privacy protection against such prediction.

In *Data Engineering (ICDE), 2013 IEEE 29th International Conference on*, pages 254–265. IEEE, 2013.

[58] Raymond A Yeh, Alexander G Schwing, Jonathan Huang, and Kevin Murphy. Diverse generation for multi-agent sports games. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4610–4619, 2019.

[59] Jiaxuan You, Zhitao Ying, and Jure Leskovec. Design space for graph neural networks. *Advances in Neural Information Processing Systems*, 33, 2020.

[60] Cunjun Yu, Xiao Ma, Jiawei Ren, Haiyu Zhao, and Shuai Yi. Spatio-temporal graph transformer networks for pedestrian trajectory prediction. In *European Conference on Computer Vision*, pages 507–523. Springer, 2020.

[61] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Ruslan Salakhutdinov, and Alexander Smola. Deep sets. *arXiv preprint arXiv:1703.06114*, 2017.

[62] Eric Zhan, Stephan Zheng, Yisong Yue, Long Sha, and Patrick Lucey. Generating multi-agent trajectories using programmatic weak supervision. *arXiv preprint arXiv:1803.07612*, 2018.

[63] Stephan Zheng, Yisong Yue, and Patrick Lucey. Generating long-term trajectories using deep hierarchical networks. *arXiv preprint arXiv:1706.07138*, 2017.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] Since this work constitutes the first semi-supervised approach for spatiotemporal multi-agent problems, it is difficult to discuss techical limitations at this stage. Application-wise, there are clear limitations e.g., by using weak makeshift labels. We discuss these issues across the paper (e.g., Section 4.)
   (c) Did you discuss any potential negative societal impacts of your work? [No] We have a clear focus on team sports. Though dual use of the proposed technique is certainly possible (e.g., military domains), it may be a bit far fetched as we have no expertise in these domains and would only be able to deliver uneducated guesses about whether this is realistic or not.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See Section 4.1. – 4.3.
   (b) Did you include complete proofs of all theoretical results? [Yes] All proofs are contained in the Appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The NBA data set is publicly available and we provide a URL in the paper. The soccer data is private and must not be shared with third parties. We will release the code and include a URL in the paper as well after the paper has been accepted.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4, more details are provided in the Appendixl.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Appendix
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]

(b) Did you mention the license of the assets? [Yes] See supplemental material

(c) Did you include any new assets either in the supplemental material or as a URL? [Yes]

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] The NBA data is publicly available and free, for soccer we have the permission of the data provider.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] The soccer and basketball players are probably identifiable in the data, but they/their contract/their clubs agreed in recording the data. The soccer data is private and cannot be shared.

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Proofs of the Theorems

We begin with Theorem 2 for simplicity and sketch the proof of Theorem 1 afterwards.

Theorem 2. *Let $\mathcal{H}(\beta)$ be the entropy of quantity β. A lower bound on $\log p_\theta(\boldsymbol{x}_{\leq T})$ in Eqn (1) is given by*

$$\log p_\theta(\boldsymbol{x}_{\leq T}) \geq \sum_t \Big(\mathcal{H}\big(q_\phi(\tilde{\boldsymbol{y}}_t|\boldsymbol{x}_{\leq t}, \boldsymbol{z}_{<t}, \boldsymbol{y}_{<t})\big) - \mathbb{E}_{q_\phi(\tilde{\boldsymbol{y}}_t|\boldsymbol{x}_{\leq t}, \boldsymbol{z}_{<t}, \tilde{\boldsymbol{y}}_{<t})}\big[\mathcal{L}(\boldsymbol{x}_t, \tilde{\boldsymbol{y}}_t)\big]\Big) \equiv \sum_t -\mathcal{U}(\boldsymbol{x}_t)$$

Proof. For unlabeled training instances, defining the marginal likelihood according to the generative structure introduced in Section 3.3 yields

$$\begin{aligned} p_\theta(\boldsymbol{x}_{\leq T}) &= \int_{\boldsymbol{z}_{\leq T}} \int_{y_{\leq T}} p_\theta(\boldsymbol{x}_{\leq T}, \boldsymbol{z}_{\leq T}, y_{\leq T}) dy_{\leq T} d\boldsymbol{z}_{\leq T} \\ &= \int_{\boldsymbol{z}_{\leq T}} \sum_{y_{\leq T}} \prod_{t=1}^{T} p_\theta(\boldsymbol{x}_t|\boldsymbol{x}_{<t}, \boldsymbol{z}_{\leq t}, y_{\leq t}) p_\theta(\boldsymbol{z}_t|\boldsymbol{x}_{<t}, \boldsymbol{z}_{<t}, y_{<t}) p_\theta(y_t|\boldsymbol{x}_{<t}, \boldsymbol{z}_{<t}, y_{<t}) d\boldsymbol{z}_{\leq T}. \end{aligned}$$

To derive the lower bound on the log likelihood, we incorporate the variational information into the above definition and apply Jensen's inequality:

$$\begin{aligned} &\int\int q_\phi(\boldsymbol{z}_{\leq T}, y_{\leq T}|\boldsymbol{x}_{\leq T}) \log \prod_{t=1}^{T} \frac{p_\theta(\boldsymbol{z}_t|\boldsymbol{x}_{<t}, \boldsymbol{z}_{<t}, y_{<t}) p_\theta(y_t|\boldsymbol{x}_{<t}, \boldsymbol{z}_{<t}, y_{<t})}{q_\phi(\boldsymbol{z}_t|\boldsymbol{x}_{\leq t}, \boldsymbol{z}_{<t}, y_{\leq T}) q_\phi(y_t|\boldsymbol{x}_{\leq t}, \boldsymbol{z}_{\leq t}, y_{<t})} d\boldsymbol{z}_{\leq T} dy_{\leq T} \\ &= \sum_{t=1}^{T} \int\int q_\phi(\boldsymbol{z}_{\leq T}, y_{\leq T}|\boldsymbol{x}_{\leq T}) \log \frac{p_\theta(\boldsymbol{z}_t|\boldsymbol{x}_{<t}, \boldsymbol{z}_{<t}, y_{<t}) p_\theta(y_t|\boldsymbol{x}_{<t}, \boldsymbol{z}_{<t}, y_{<t})}{q_\phi(\boldsymbol{z}_t|\boldsymbol{x}_{\leq t}, \boldsymbol{z}_{<t}, y_{\leq T}) q_\phi(y_t|\boldsymbol{x}_{\leq t}, \boldsymbol{z}_{\leq t}, y_{<t})} d\boldsymbol{z}_{\leq T} dy_{\leq T} \\ &= \sum_{t=1}^{T} \int\int q_\phi(\boldsymbol{z}_{\leq t}, y_{\leq t}|\boldsymbol{x}_{\leq t}) \log \frac{p_\theta(\boldsymbol{z}_t|\boldsymbol{x}_{<t}, \boldsymbol{z}_{<t}, y_{<t}) p_\theta(y_t|\boldsymbol{x}_{<t}, \boldsymbol{z}_{<t}, y_{<t})}{q_\phi(\boldsymbol{z}_t|\boldsymbol{x}_{\leq t}, \boldsymbol{z}_{<t}, y_{\leq t}) q_\phi(\mathcal{G}_t|\boldsymbol{x}_{\leq t}, \boldsymbol{z}_{\leq t}, y_{<t})} d\boldsymbol{z}_{\leq t} dy_{\leq t} \\ &= \sum_{t=1}^{T} \int\int q_\phi(\boldsymbol{z}_{<t}, y_{<t}|\boldsymbol{x}_{<t}) \Bigg(- \mathbb{E}_{q_\phi(y_t|\boldsymbol{x}_{\leq t}, \boldsymbol{z}_{<t}, y_{<t})}\big[\mathcal{KL}[q_\phi(\boldsymbol{z}_t|\boldsymbol{x}_{<t}, \boldsymbol{z}_{<t}, y_{\leq T} \parallel p_\theta(\boldsymbol{z}_t|\boldsymbol{x}_{<t}, \boldsymbol{z}_{<t}, \mathcal{G}_{<t})]\big] \\ &\qquad - \mathcal{KL}[q_\phi(y_t|\boldsymbol{x}_{\leq t}, \boldsymbol{z}_{<t}, y_{<t}) \parallel p_\theta(y_t|\boldsymbol{x}_{<t}, \boldsymbol{z}_{<t}, y_{<t})]\Bigg) d\boldsymbol{z}_{<t} dy_{<t} \\ &= \mathbb{E}_{q_\phi(\boldsymbol{z}_{\leq T}, y_{\leq T}|\boldsymbol{x}_{\leq T})}\Bigg[\sum_{t=1}^{T} -\mathcal{KL}[q_\phi(\boldsymbol{z}_t|\boldsymbol{x}_{\leq t}, \boldsymbol{z}_{<t}, y_{\leq T}) \parallel p_\theta(\boldsymbol{z}_t|\boldsymbol{x}_{<t}, \boldsymbol{z}_{<t}, y_{<t})] \\ &\qquad - \mathcal{KL}[q_\phi(y_t|\boldsymbol{x}_{\leq t}, \boldsymbol{z}_{<t}, y_{<t}) \parallel p_\theta(y_t|\boldsymbol{x}_{<t}, \boldsymbol{z}_{<t}, y_{<t})]\Bigg]. \end{aligned}$$

Thus, we can write

$$\begin{aligned} \log p_\theta(\boldsymbol{x}_{\leq T}) \geq \mathbb{E}_{q_\phi(\boldsymbol{z}_{\leq T}, y_{\leq T}|\boldsymbol{x}_{\leq T})}\Bigg[&\sum_{t=1}^{T} \log p_\theta(\boldsymbol{x}_t|\boldsymbol{x}_{<t}, \boldsymbol{z}_{\leq t}, y_{\leq t}) \\ &- \mathcal{KL}[q_\phi(\boldsymbol{z}_t|\boldsymbol{x}_{\leq t}, \boldsymbol{z}_{<t}, y_{\leq t}) \parallel p_\theta(\boldsymbol{z}_t|\boldsymbol{x}_{<t}, \boldsymbol{z}_{<t}, y_{<t})] \\ &- \mathcal{KL}[q_\phi(y_t|\boldsymbol{x}_{\leq t}, \boldsymbol{z}_{<t}, y_{<t}) \parallel p_\theta(y_t|\boldsymbol{x}_{<t}, \boldsymbol{z}_{<t}, y_{<t})]\Bigg], \end{aligned}$$

which is identical to Eqn 3. □

Theorem 1. *A lower bound on $\log p_\theta(\boldsymbol{x}_{\leq T}, \boldsymbol{y}_{\leq T})$ in Eqn (1) is given by*

$$\begin{aligned} \log p_\theta(\boldsymbol{x}_{\leq T}, \boldsymbol{y}_{\leq T}) \geq \sum_t &\log p_\theta(y_t|\boldsymbol{x}_{<t}, \boldsymbol{z}_{<t}, \boldsymbol{y}_{<t}) + \mathbb{E}_{q_\phi(\boldsymbol{z}_t|\boldsymbol{x}_{\leq t}, \boldsymbol{z}_{<t}, \boldsymbol{y}_{\leq t})}\Big[\log p_\theta(\boldsymbol{x}_t|\boldsymbol{x}_{<t}, , \boldsymbol{z}_{\leq t}, \boldsymbol{y}_{\leq t})\Big] \\ &- \mathcal{KL}[q_\phi(\boldsymbol{z}_t|\boldsymbol{x}_{\leq t}, \boldsymbol{z}_{<t}, \boldsymbol{y}_{\leq t}) \parallel p_\theta(\boldsymbol{z}_t|\boldsymbol{x}_{<t}, \boldsymbol{z}_{<t}, \boldsymbol{y}_{<t})] \equiv \sum_{t=1}^{T} -\mathcal{L}(\boldsymbol{x}_t, \boldsymbol{y}_t). \end{aligned}$$

Proof.

$$\log p_\theta(\boldsymbol{x}_{\leq T}, y_{\leq T}) = \log \int_{\boldsymbol{z}_{\leq T}} \prod_{t=1}^{T} \frac{q_\phi(\boldsymbol{z}_t|\boldsymbol{x}_{\leq t}, \boldsymbol{z}_{<t}, y_{\leq t})}{q_\phi(\boldsymbol{z}_t|\boldsymbol{x}_{\leq t}, \boldsymbol{z}_{<t}, y_{\leq t})} p_\theta(\boldsymbol{x}_t|\boldsymbol{x}_{<t}, \boldsymbol{z}_{\leq t}, y_{\leq t}) p_\theta(\boldsymbol{z}_t|\boldsymbol{x}_{<t}, \boldsymbol{z}_{<t}, y_{<t})$$

$$\geq \mathbb{E}_{q_\phi(\boldsymbol{z}_{\leq T}|\boldsymbol{x}_{\leq T}, y_{\leq T})} \Bigg[\sum_{t=1}^{T} \log p_\theta(\boldsymbol{x}_t|\boldsymbol{x}_{<t}, \boldsymbol{z}_{\leq t}, y_{\leq t})$$

$$- \mathcal{KL}[q_\phi(\boldsymbol{z}_t|\boldsymbol{x}_{\leq t}, \boldsymbol{z}_{<t}, y_{\leq t}) \parallel p_\theta(\boldsymbol{z}_t|\boldsymbol{x}_{<t}, \boldsymbol{z}_{<t}, y_{<t})] + \log p_\theta(y_t|\boldsymbol{x}_{<t}, \boldsymbol{z}_{<t}, y_{<t}) \Bigg].$$

□

B Implementation Details

All models were implemented using PyTorch [41]. For our experiments we define the model components as follows. The encoders are modeled as $q_\phi(\tilde{y}^{(a)}|\boldsymbol{x}_{\leq t}, \boldsymbol{z}_{<t}, y_{<t}) = \text{Cat}(y_t^{(a)}|f_{enc}^{y}([\boldsymbol{o}_{t-1}^{(a)}, \varphi^{\boldsymbol{x}}(\boldsymbol{x}_t^{(a)})])$ and $q_\phi(\boldsymbol{z}_t^{(a)}|\boldsymbol{x}_{\leq t}, \boldsymbol{z}_{<t}, y_{\leq t}) = \mathcal{N}(\boldsymbol{z}_t^{(a)}|f_{enc}^{\boldsymbol{z}}([\boldsymbol{o}_{t-1}^{(a)}, \varphi^{\boldsymbol{x}}(\boldsymbol{x}_t^{(a)}), \varphi^{y}(y_t^{(a)})])$ for uncovering discrete and continuous latent information, respectively. The prior distributions are computed analogously omitting input $\boldsymbol{x}_t$. Agent movements are represented as $p_\theta(\boldsymbol{x}_t^{(a)}|\boldsymbol{x}_{<t}, \boldsymbol{z}_{\leq t}, \boldsymbol{y}_{\leq t}) = \mathcal{N}(\boldsymbol{z}_t^{(a)}|f_{dec}([\boldsymbol{o}_{t-1}^{(a)}, \varphi^{\boldsymbol{x}}(\boldsymbol{x}_t^{(a)}), \varphi^{y}(y_t^{(a)})])$.

All functions f and feature extractors φ are two-layer MLPs with LeakyRelu [55] activations. We use 2-layer GRU networks [6] for recurrence, and GNNs with 3 GAT layers implemented according according to [54]. We implement skip-connections as described in [56] and use concatenation followed by a linear transformation to aggregate intermediate layer embeddings to the GNN model output. For generative tasks, we used $\lambda_1 = 0$, however, including the auxiliary loss yields negligible deterioration. For classification, we use $\lambda_0 = 1$ and $\alpha = 10 * (1/\%$ of labeled data$)$

The model operates solely on agent velocities (input and output). However, we inject position tuples to the all model components via teacher forcing. We use the gumbel-softmax trick [34] [23] when sampling from categorical distributions. Training was executed on an Nvidia V100 GPU and took about 16 hours to complete 300 epochs while comsuming $\sim$ 15Gb for basketball. All models were optimized using Adam [25] with a learning rate of $0,001$ and gradient clipping using a `max` norm of 10.

Soccer Application Since the labels denote ball-centric events, we use the output of the ball node for loss computation and evaluation. The $F1$ score is computed as follows. We annotate a multi-agent segment when the derived probability estimates exceed an externally defined threshold value. We obtain TP values (FP values) when the predicted event coincides (disagrees) with the ground truth annotation. FN values are defined by anotated segments that remain undetected. We compute F1-scores for 100 distinct threshold values in the range between 0.5 and 0.98 and only report the maximum F1-score. However, threshold optimization yields only negligible improvement over simply using 0.5.

Baselines As stated in Section 4.1, we report against the values reported in [39]. For [16, 32, 36], we adapt the source code from their official repositories to our experimental setting[6] For GVRNN [58], we re-implemented the model according to the descriptions in their paper and designed the overall architecture such that it is comparable in parameter number to our method.

C Experiments with Drone Data

To showcase applicability to scenarios that exhibt variable numbers of agents, we also report results on the Stanford Drone Data (SDD) [43]. SDD is a collection of videos recorded by drones at eight locations at Stanford. While pedestrians predominate as interacting agents, cyclists, skateboarders,

[6]For [36], we used `https://github.com/crowdbotp/OpenTraj/blob/master/datasets/SDD/estimated_scales.yaml` to map between real-world and pixel coordinate. s

Table 4: Results for SDD (observation phase of 8 timesteps and a prediction horizon with 12 timesteps) expressed in real-world coordinates.

NAME	$avgL_2$	FINAL L_2
STGAT	0.58	1.11
SOCIAL-WAYS	0.62	1.16
DAG-NET	0.53	1.04
PECNET	0.67	**1.03**
S-MAT	**0.51**	**1.03**

cars, buses, and golf carts are also present. We use the TrajNet benchmark [46] of the data, providing sequential two-dimensional real-world coordinates at a frame rate of 2.5 frames per second. We follow Monti et al. [39] regarding the data processing strategy[7]. We incorporate *PECNet* [36] for comparison as it is considered state-of-the-art for SDD [35]. PECNet models stochasticity in the final position of the pedestrians conditioned on the past motion history.

To generate weak labels, we move a time window through each trajectory, with the respective end cells acting as agent targets $y_t^{(a)}$. Static time windows allow us to use the fully-supervised variant for all comparisons (including fully unsupervised models). The scene at hand is thereby discretized into 960 areas and results are shown in Table 4. We observe the same general pattern as for the basketball experiments.

D Ablation Study

Table 5: Ablation study on NBA.

NAME	L_2	FINAL L_2
S-MAT-DIAG	8.93	13.92
S-MAT-FULL	8.87	13.87
S-MAT-GVRNN	9.78	14.27
S-MAT-GVRNN-HIDDEN	9.90	15.20
S-MAT	**8.11**	**12.52**

Table 5 validates the proposed architecture by showing results of an ablation study. We test a fully connected graph (*S-MAT-Full*), an independent version with diagonal adjacency matrices (*S-MAT-Diag*), a variant that employs GNNs for variational, generative, and prior distribution parameters, but ignores interactive updates for the hidden states (*S-MAT-GRVNN*) similar to Yeh et al. [58], and a variant that is identical to the latter but additionally includes a GNN for the hidden state (*S-MAT-GRVNN-Hidden*). The table provides supporting evidence for our design choices: though the GVRNNs are theoretically able to capture intra-timestep dependencies, we observe significant drops in performance compared to the other competitors. The S-MAT-Diag and MAT-Full experiments suggest that our model captures interaction patterns among agents very well; *S-MAT* denotes a valuable contribution to the large body of research that explicitly addresses modeling multi-agent data accurately.

[7]Data and preprocessing can be accessed at `https://github.com/alexmonti19/dagnet/tree/master/datasets`