Appendix: Semi-Supervised Generative Models for Multi-Agent Trajectories

Dennis Fassmeyer
Leuphana University of Lüneburg
dennis.fassmeyer@leuphana.de

Pascal Fassmeyer
Leuphana University of Lüneburg
pascal.fassmeyer@leuphana.de

Ulf Brefeld
Luephana University of Lüneburg
brefeld@leuphana.de

1 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 $\log p_\theta(\boldsymbol{x}_{\leq T})$ 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 yields

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

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

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

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

$$= \sum_{t=1}^{T} \int\int q_\phi(\boldsymbol{z}_{\leq T}, y_{\leq t}|\boldsymbol{x}_{\leq T}) \log \frac{p_\theta(\boldsymbol{z}_t|\boldsymbol{x}_{<t}, \boldsymbol{z}_{<t}, y_{<t}) p_\theta(y_t|\boldsymbol{x}_{<t}, \boldsymbol{z}_{<t}, y_{<t})}{q_\phi(\boldsymbol{z}_t|\boldsymbol{x}_{\leq t}, \boldsymbol{z}_{<t}, y_{\leq t}) q_\phi(y_t|\boldsymbol{x}_{\leq t}, \boldsymbol{z}_{\leq t}, y_{<t})} d\boldsymbol{z}_{\leq T} dy_{\leq T}$$

$$= \sum_{t=1}^{T} \int\int q_\phi(\boldsymbol{z}_{\leq t}, y_{\leq t}|\boldsymbol{x}_{\leq t}) \log \frac{p_\theta(\boldsymbol{z}_t|\boldsymbol{x}_{<t}, \boldsymbol{z}_{<t}, y_{<t}) p_\theta(y_t|\boldsymbol{x}_{<t}, \boldsymbol{z}_{<t}, y_{<t})}{q_\phi(\boldsymbol{z}_t|\boldsymbol{x}_{\leq t}, \boldsymbol{z}_{<t}, y_{\leq t}) q_\phi(\boldsymbol{y}_t|\boldsymbol{x}_{\leq t}, \boldsymbol{z}_{\leq t}, y_{<t})} d\boldsymbol{z}_{\leq t} dy_{\leq t}$$

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

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

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

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

Thus, we can write

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

which is identical to our objective function. □

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

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

Proof.

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

□

2 Implementation Details

For our experiments, we define the model components as follows. We individually capture the spatiotemporal dependencies of the label-specific patterns $\boldsymbol{y}_{\leq T}$ and the data-specific patterns $\boldsymbol{x}_{\leq T}$, $\boldsymbol{z}_{\leq T}$, resulting in two separate variables $\boldsymbol{o}_t^{(a),\boldsymbol{l}}$ and $\boldsymbol{o}_t^{(a),\boldsymbol{d}}$. For ease of exposition, we refer to the concatenation of the two parts as $\boldsymbol{o}_t^{(a)}$ in the remainder of this paper. 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 omitti input $\boldsymbol{x}_t$. Agent movements are represented as $p_\theta(\boldsymbol{x}_t^{(a)}|\boldsymbol{x}_{<t}, \boldsymbol{z}_{\leq t}, \boldsymbol{y}_{\leq t}) = \mathcal{N}(\boldsymbol{x}_t^{(a)}|f_{dec}([\boldsymbol{o}_{t-1}^{(a)}, \varphi^{\boldsymbol{x}}(\boldsymbol{z}_t^{(a)}), \varphi^y(y_t^{(a)})])$.

All functions f and feature extractors φ are two-layer MLPs with LeakyRelu [14] activations. We use 2-layer GRU networks [1] for recurrence. Moreover, we define the graph network using 3 GAT layers implemented according to [13] and the graph structure wrt.the $k = 2$ and $k = 5$ spatially closest agents for basketball and soccer, respectively. We implement skip-connections as described in [15] and use concatenation followed by a linear transformation to aggregate intermediate layer embeddings to the GNN model output. For generative tasks, we use $\lambda_1 = 0$, 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 [6, 3] 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 are optimized using Adam [4] with a learning rate of $0,001$ and gradient clipping using a `max` norm of 10. All models are implemented using PyTorch [10].

Baselines As stated in the main text, we report against the values reported in [9]. For [2, 5, 8], we adapt the source code from their official repositories to our experimental setting[1] For GVRNN [16], 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.

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.

3 More Details on the MAT Models

Discrete variables $\boldsymbol{y}_{\leq T}$ are defined arbitrary (possibly latent) behavioral indicators. To validate this general formulation (and our proposed architecture), we vary the specific definition of the variable across experiments. Accordingly, we propose different framework instantiations that are described in more detail in this section.

S-MAT In the main text, we first observe that existing SOTA trajectory prediction approaches use heuristically generated labels for trajectory prediction that encode agents' intents or goals over a discretized position space. We then note that our formalization allows to naturally integrate such long-term goals into the overall scheme via treating them as discrete semantic concepts. Since these labels are generated heuristically based on the trajectory input prior to model training, this model instantiation is fully-supervised and is referred to as *S-MAT*. We refer to [17] for more details on how to produce the weak labels used for training S-MAT.

[1] For [8], we used `https://github.com/crowdbotp/OpenTraj/blob/master/datasets/SDD/estimated_scales.yaml` to map between real-world and pixel coordinates.

U-MAT However, we found (although done in [**?** 9]) that benchmarking against unsupervised baselines is inappropriate for the reasons described in the main text. Since most SOTA approaches are unsupervised generative models, we additionally propose a fully-unsupervised instantiation of our framework (*U-MAT*) employing no external guidance for discrete latent structuring. From the predicted (latent) categorical distribution q_ϕ, we sample a label value and exploit separate motion predictors (different decoder parameterizations) p_θ dependent on the realized value. Intuitively, this encourages the model to learn categories describing fundamentally different movement patterns, which can be interpreted as dynamic "agent roles". Thus, the concept of agent roles here is merely an intuitive explanation and realized via an inductive bias that increases the "scope" of latent information encoded without utilizing any supervision. To the best of our knowledge, parameterizing the generation module based on inferred agent categories is novel and could provide valuable insights for practitioners.

4 Experiments with Drone Data

Table 1: 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**

To showcase applicability to scenarios that exhibt variable numbers of agents, we also report results on the Stanford Drone Data (SDD) [11]. SDD is a collection of videos recorded by drones at eight locations at Stanford. While pedestrians predominate as interacting agents, cyclists, skateboarders, cars, buses, and golf carts are also present. We use the TrajNet benchmark [12] of the data, providing sequential two-dimensional real-world coordinates at a frame rate of 2.5 frames per second. We follow [9] regarding the data processing strategy[2]. We incorporate *PECNet* [8] for comparison as it is considered state-of-the-art for SDD [7]. 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 1. We observe the same general pattern as for the basketball experiments.

5 Ablation Study

Table 2: 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 2 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-*

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

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 [16], 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.

References

[1] 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.

[2] 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.

[3] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparametrization with gumble-softmax. In *International Conference on Learning Representations (ICLR 2017)*. OpenReview. net, 2017.

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

[5] 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.

[6] C Maddison, A Mnih, and Y Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *Proceedings of the international conference on learning Representations*. International Conference on Learning Representations, 2017.

[7] Osama Makansi, J von Kügelgen, F Locatello, P Gehler, D Janzing, Thomas Brox, and B Schölkopf. You mostly walk alone: Analyzing feature attribution in trajectory prediction. In *International Conference on Learning Representations (ICLR)*, 2022.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

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

[13] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

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

[15] 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.

[16] 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.

[17] Eric Zhan, Stephan Zheng, Yisong Yue, Long Sha, and Patrick Lucey. Generating multi-agent trajectories using programmatic weak supervision. In *International Conference on Learning Representations*, 2019.