Generating Multi-Agent Trajectories using Programmatic Weak Supervision

Eric Zhan[1], Stephan Zheng[2], Yisong Yue[1], Long Sha[3], Patrick Lucey[3]

1: Caltech, 2: Salesforce Research, 3: STATS

Problem

Learning to generate **multi-agent** trajectories over **long** time horizons that **generalize** well has various challenges:

- Individual behavior is **complex** and **multi-modal**.
- Team **coordination** is coherent over time.
- Behaviors and goals can evolve and change.
- Space of joint trajectories is **exponentially** large.

Our Solution

We present a hierarchical framework with intermediate variables that capture high-level behavioral semantics in an **interpretable** and **manipulable** way.

- We leverage **programmatic weak supervision** to produce useful weak labels for supervised learning of intermediate variables.
- Intermediate variables are interpretable and allows for conditional generation to manipulate behaviors.
- Our approach can generate high-quality trajectories and effectively encode long-term coordination between multiple agents.

Sequential Generative Models

Goal: learn generative distribution over collection of sequences of length T: $\mathcal{D} = \{\mathbf{x}_{\leq T}\}$, where $\mathbf{x}_{\leq T} = \{\mathbf{x}_1, \dots, \mathbf{x}_T\}$.

Learning objective: factorize joint distribution and maximize the log-likelihood (common for RNN models):

$$\theta^* = \mathrm{argmax}_\theta \sum_{\mathbf{x}_{\leq T} \in \mathcal{D}} \sum_{t=1}^{T} \log p_\theta(\mathbf{x}_t | \mathbf{x}_{<t}). \quad (1)$$

Stochastic latent variable models introduce latent variables $\mathbf{z}$ and optimize the ELBO using amortized variational inference. In particular, variational RNNs (VRNN, [1]) maximize:

$$\mathbb{E}_{q_\phi(\mathbf{z}_{\leq T}|\mathbf{x}_{\leq T})}\Bigg[\sum_{t=1}^{T} \log p_\theta(\mathbf{x}_t \mid \mathbf{z}_{\leq t}, \mathbf{x}_{<t}) - D_{KL}\Big(q_\phi(\mathbf{z}_t \mid \mathbf{x}_{\leq t}, \mathbf{z}_{<t}) || p_\theta(\mathbf{z}_t \mid \mathbf{x}_{<t}, \mathbf{z}_{<t})\Big)\Bigg]. \quad (2)$$

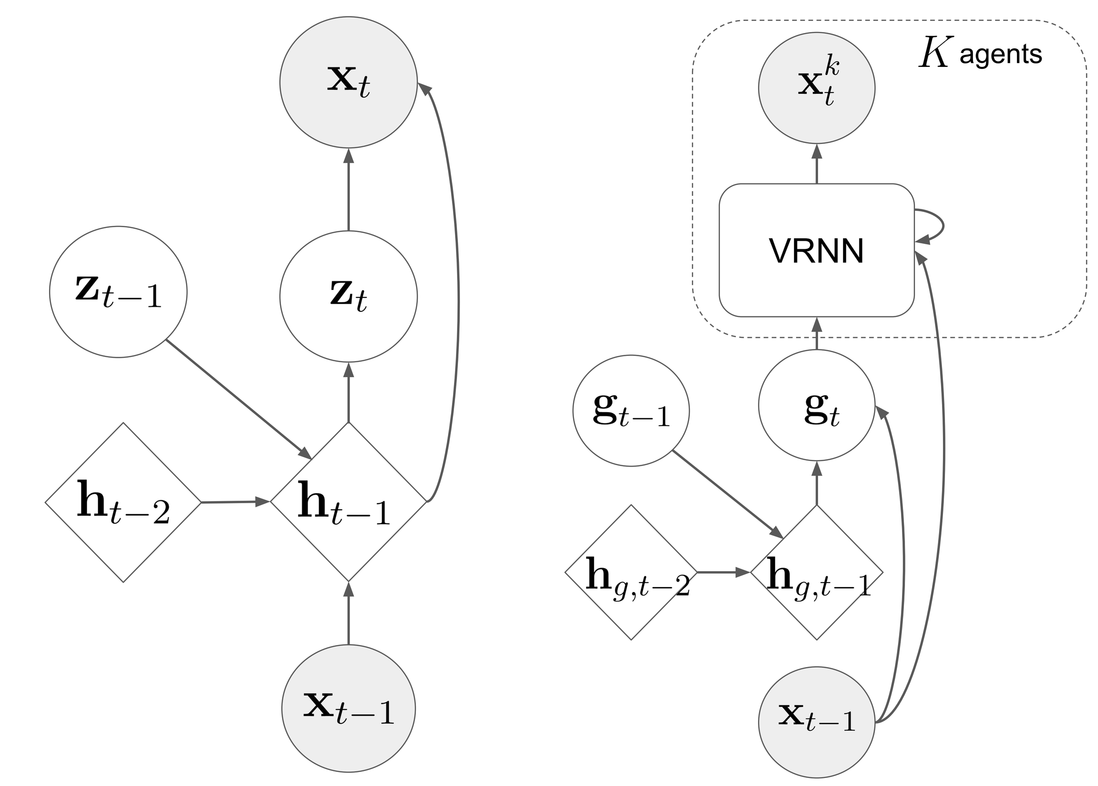
In **multi-agent settings**, sequences contain K trajectories:

$$\theta^* = \mathrm{argmax}_\theta \sum_{\mathbf{x}_{\leq T} \in \mathcal{D}} \sum_{t=1}^{T} \sum_{k=1}^{K} \log p_{\theta_k}(\mathbf{x}_t^k | \mathbf{x}_{<t}). \quad (3)$$

Two natural baselines are:

- Single model for all agents: $\theta = \theta_1 = \dots = \theta_K$.
- Independent models for each agent: $\theta = \{\theta_1, \dots, \theta_K\}$.

Our Framework



(a) VRNN (b) Ours

We model each agent independently with a sequential generative model (VRNN in our experiments), but condition all agents on a **shared macro-intent variable** $\mathbf{g}_t$ that:

- tractably captures coordination between agents,
- encodes long-term intents of agents,
- enables long-term planning at a higher-level timescale,
- and compactly represents some low-dimensional structure in an exponentially large multi-agent state space.

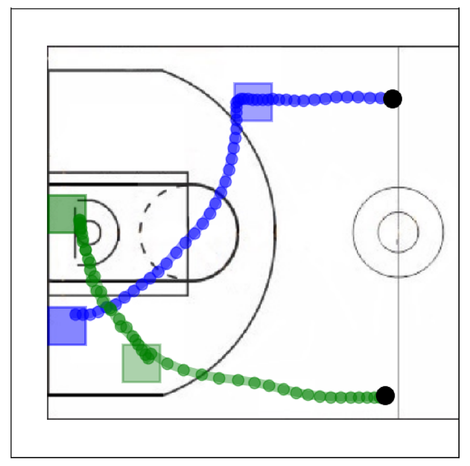
Learning Macro-Intents

Learning macro-intents via unsupervised learning can be ineffective, and obtaining expert macro-intent labels is expensive.

Inspired by [2], we use **labeling functions** to compute heuristics as weak labels to train our macro-intent model via supervised learning. These labels are cheap to obtain and allow users to incorporate domain knowledge into the model.

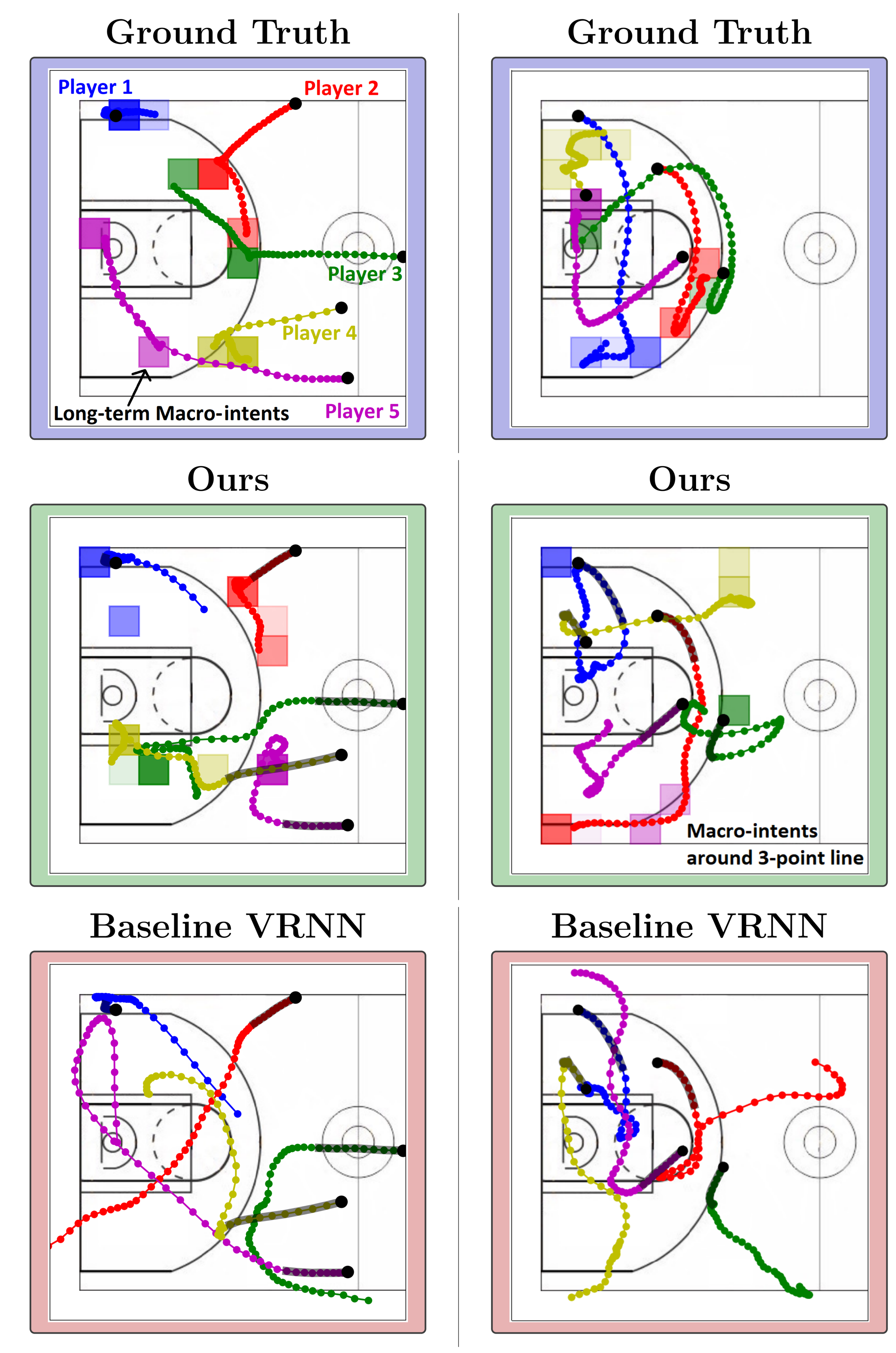
For example, labeling functions for basketball trajectories:

- **LF-50**: last region on court every 50-timestep window
- **LF-25**: last region on court every 25-timestep window
- **LF-st**: regions on court in which players are stationary



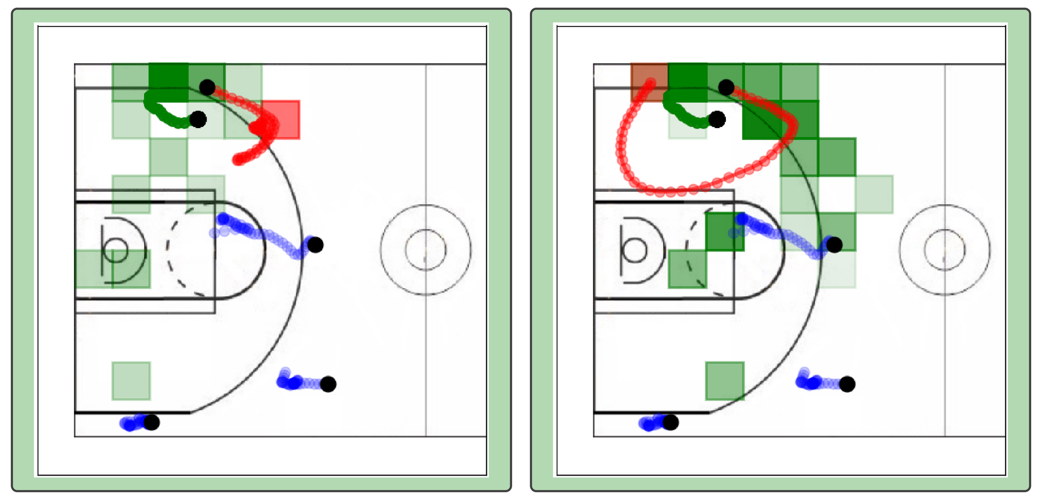
Above: macro-intents (boxes) computed using **LF-st** for two players [3]. Macro-intents are visible to both players in our model and provide a compact summary of their trajectories.

Generated Trajectories for Basketball



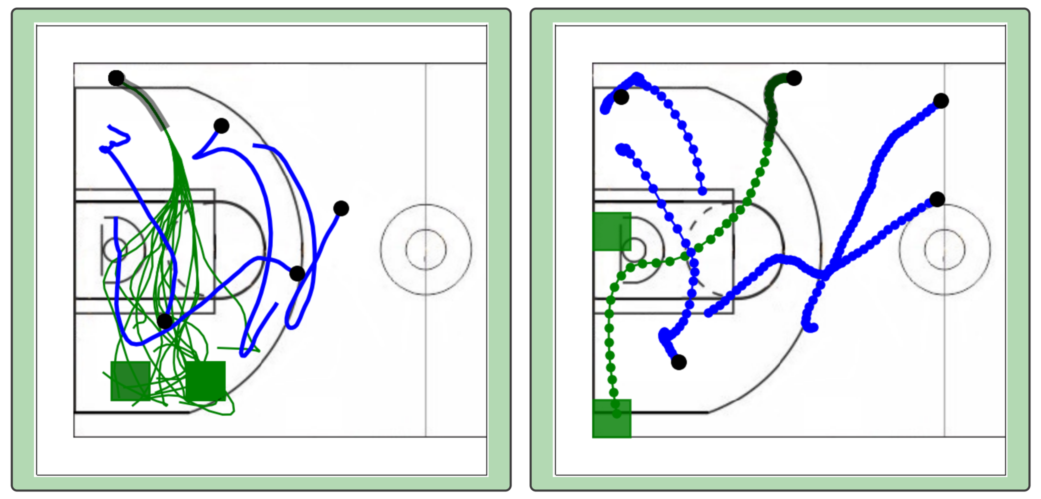
- **Baselines**: Players move in the wrong direction, out of bounds, and not cohesively.
- **Ours**: Generated macro-intents guide players to stay in bounds and reveal team formations for players to execute.

Multi-Agent Coordination



Above images show macro-intent distribution from 20 rollouts of the green player. As we change the macro-intent for the red player, the distribution of macro-intents for the green player changes such that they occupy different areas on the court.

Conditional Generation



- **Left**: The green player takes different paths towards the same macro-intents in 15 rollouts, which suggests our model captures a distribution over possible trajectories.
- **Right**: Macro-intents are manually fixed to guide the green player towards the basket and then to the bottom-left corner.

Quantitative Evaluation

vs. Model	Win/Tie/Loss	Avg Gain
vs. VRNN-single	25/0/0	0.57
vs. VRNN-indep	15/4/6	0.23

Human preference study. We recruited 14 professional sports analysts as judges to compare the quality of 25 generated trajectories from each of our model and baselines. All judges preferred our model over baselines with 98% statistical significance.

Model	Speed (ft)	Dist. (ft)	OOB (%)
RNN-gauss	3.05	149.57	46.93
VRNN-single	1.28	62.67	45.67
VRNN-indep	0.89	43.78	33.78
VRAE-info	0.98	48.25	20.09
Ours (LF-50)	0.99	48.53	28.84
Ours (LF-25)	0.87	42.99	**14.53**
Ours (LF-st)	**0.79**	**38.92**	15.52
Ground-truth	0.77	37.78	2.21

Domain statistics. We computed and compared several basketball statistics (average speed, average distance traveled, % of timesteps with players out-of-bounds). Generated trajectories from our models match the closest with the ground-truth, with better results as we use more informative labeling functions. **ELBOs** on test sequences are comparable: VRNN-single (2302), VRNN-indep (2360), VRNN-info (2349), and ours (2362). This is not necessarily a good indicator of generation quality.

References

[1] J. Chung, K. Kastner, L. Dinh, K. Goel, A. C. Courville, and Y. Bengio. A recurrent latent variable model for sequential data. In *NIPS*, 2015.

[2] A. Ratner, C. D. Sa, S. Wu, D. Selsam, and C. RĂI. Data programming: Creating large training sets, quickly. In *NIPS*, 2016.

[3] S. Zheng, Y. Yue, and P. Lucey. Generating long-term trajectories using deep hierarchical networks. In *NIPS*, 2016.