Long Range Sequence Generation via Multiresolution Adversarial Training

Yukai Liu* Caltech

Rose Yu* Northeastern University

Stephan Zheng*† Salesforce Research

Yisong Yue* Caltech

Abstract

We present a multi-resolution approach for generative modeling of spatiotemporal sequences. Our approach focuses on a key challenge of modeling long-term temporal dependencies that is common in complex spatiotemporal phenomena. For instance, realistic modeling of basketball players behavior requires capturing long-term goals and how they influence short-term decisions. Our multi-resolution approach has several attractive properties. First, it is completely unsupervised, and requires no additional labeling of high-level semantics such as long-term goals. Second, the multi-resolution aspect allows us to model conditional distributions beyond forward sampling, such as conditioning on future behavior. Finally, our approach integrates generative adversarial training, which enables us to learn generative models that significantly outperform conventional generative sequence modeling. We validate the effectiveness of our model on synthetic sequences and spatiotemporal basketball player trajectory generation.

1 Introduction

Generative models for sequential data have broad applications in audio synthesis [7], machine translation [12], video generation [11], and sports analytics [14]. For example, given the movement trajectories from professional basketball team play, generative models can extract latent representations that imitate the decision making process of multiple players in spatiotemporal environments.

However, designing generative models for long-range spatiotemporal sequences is highly challenging, as generative modeling requires estimating the joint probability over the entire sequence, which is susceptible to covariant shift and compounding errors. Moreover, long-range sequences contain long-term dependencies which pose difficulties to many existing models such as RNNs [8].

In contrast to the traditional maximum likelihood objective, generative adversarial networks (GAN) [2] introduce a discriminator to the loss function, which has sparked a new paradigm of generative modeling. [3] proposed Generative Adversarial Imitation Learning (GAIL) to extend GAN to the sequential setting by using an equivalence between maximum entropy inverse reinforcement learning (IRL) and GANs. [12] further modified the reward structure and only gives a single reward when the sequence is completed. But these models only use simple generators, which limit their ability to model long-range sequences.

In order to generate long-range trajectories, [14] proposes to use manually defined macro goals from trajectories as weak labels to train a hierarchical RNN. [13] further extends this idea to the multi-agent setting with a hierarchical variational RNN. However, while using macro goals can significantly reduce the search space, obtaining the macro goals from trajectories using e.g. stationary points, can be ad-hoc and expensive. To avoid using weak labels, we propose a novel

*yukai@caltech.edu, roseyu@northeastern.edu, stephan.zheng@salesforce.com, yyue@caltech.edu

†This work was done while the author was at Caltech

Workshop on Modeling and Decision-Making in the Spatiotemporal Domain, 32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

hierarchical framework for generating long-range spatiotemporal sequences from raw data. The top level generator in this hierarchy operates at the coarsest resolution and the lower level generators inherit the previous predictions to refine the sequence. By training all levels of the generator using a GAN loss, our framework generates realistic long-range trajectories and outperforms baselines on several spatiotemporal metrics that we use to characterize real data (e.g., path length, step size change). Our method bears affinity to other multiresolution generative models such as Progressive Growing GAN [5] and multiscale autoregressive density estimation [9].

1.1 Background

GAIL Given expert trajectories $\tau_E = \{(s_0, a_0), ..., (s_T, a_T)\} \sim \pi_E$, where π_E corresponds to an unknown expert policy, we aim to approximate the expert policy by learning a generator π_θ. To do so, GAIL introduces another discriminator D_ω which outputs a score for every (s_t, a_t) pair indicating the probability of that pair coming from the true distribution. GAIL defines the optimization objective

$$\min_G \max_D V(D,G) = \mathbb{E}_{\tau\sim\pi_E}\left[\sum_t \gamma^t \log(D(x_t, a_t))\right] + \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_t \gamma^t \log(1 - D(x_t, a_t))\right], \quad (1)$$

where a generator G and a discriminator D play a two-player minimax game. We update discriminator parameters using backpropagation, and generator parameters using policy gradient as follows:

$$\nabla_\theta \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_t \gamma^t r(s_t, a_t))\right] = \mathbb{E}_{\tau\sim\pi_\theta}\left[\left(\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(a_t|s_t)\right)\left(\sum_t \gamma^t r(s_t, a_t))\right)\right].$$

2 Multiresolution Adversarial Training

2.1 Multiresolution Generator

To extract hierarchical structures from a sequence in an unsupervised way, multiresolution generative training (`MAT`) introduces a *multiresolution generator*. For a sequence of length T, the multiresolution generator consists of $N = \lfloor logT \rfloor$ levels of sequential models $M_1, M_2, ..., M_N$, each of which operate at increasing temporal resolutions.

- M_1 predicts every $n_1 = 2^{N-1}$ steps:

  $$p(x_{1+jn_1}|x_1, x_{1+n_1}...x_{1+(j-1)n_1}) = g_1(h_j^1), \quad h_j^1 = f_1(x_{1+(j-1)n_1}, h_{j-1}^1).$$

- $M_k, 1 < k \le N$ predicts every $n_k = 2^{N-k}$ steps, working recursively:
  - if $j\, mod\, 2 == 0$: copy the states from the top level and update hidden states:

    $$x_{1+jn_k} = x_{1+(j/2)n_{k-1}}, \quad h_j^k = f_k(x_{1+(j-1)n_k}, h_{j-1}^k).$$

  - Otherwise: interpolate the states from past and future, and update hidden states:

$$p(x_{1+jn_k}|x_1, x_{1+n_k}..., x_{1+(j+1)n_k}) = g_k(h_j^k, x_{1+(j+1)n_k}), \quad h_j^k = f_k(x_{1+(j-1)n_k}, h_{j-1}^k).$$

The dependency graph of this multiresolution generator is shown in the figure 1. At each level, the sequence model only need to capture the temporal dependency of a particular resolution.

2.2 Multiresolution Adversarial Training

One way to optimize the objective in Eqn 1 is to use policy gradients, but this procedure can be expensive and tends to suffer from high variance and sample complexity [4]. Instead, `MAT` takes a model-based approach and assumes the environment models are known and differentiable. Hence, we can use the "reparameterization trick" [6] to differentiate the generator objective with respect to the policy parameters. Similar ideas have been shown in [1, 10] to be more stable and sample-efficient.

Formally, `MAT` uses the multiresolution generator G with parameters θ, and a discriminator D parameterized by ω. We are given some trajectory data drawn from an expert policy $\tau_E \sim \pi_E$, where

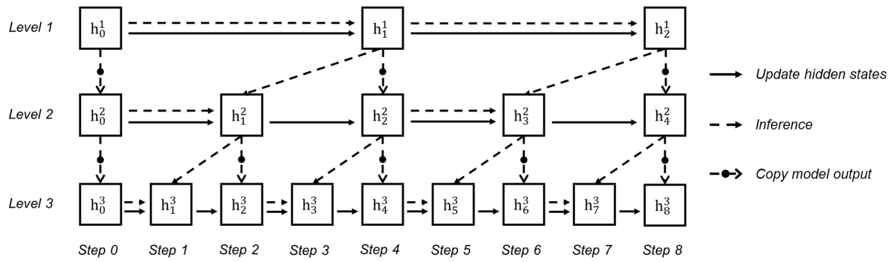

Figure 1: Dependency graph of Multiresolution Generator. Different RNNs examine and predict data at various temporal resolutions. Model outputs from the top level are used for lower level inference. Hidden states are updated within the current level.

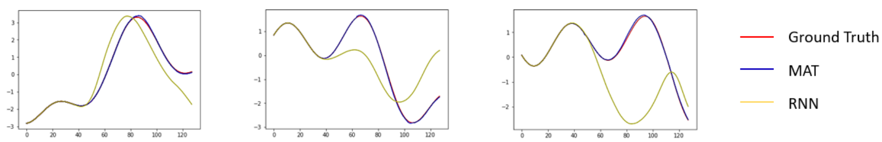

Figure 2: Comparisons between trigonometric functions. The model uses 32 time-steps as burn-in, and then predicts for 97 time-steps. With the same burn-in, `MAT` can perfectly recover the original sequence, while a single-layer RNN fails when the sequences become longer.

$\tau_E = \{x_1, x_2, ..., x_T\}$. Denote the subsequence $s_t = \{x_1, x_2, ..., x_t\}, 1 \leq t \leq T-1$. The generator produces a learner's policy π_θ. The discriminator D outputs the probability that the state-action pair (s_t, x_{t+1}) comes from the data rather than the generator. Thus our training objective function is:

$$\min_G \max_D O(D, G) = \mathbb{E}_{\tau \sim \pi_E}\left[\sum_t D(s_t, x_{t+1})\right] + \mathbb{E}_{\tau' \sim \pi_\theta}\left[\sum_t (1 - D(s_t, x_{t+1}))\right].$$

3 Experiments

3.1 Trigonometric functions

Trigonometric functions are common benchmarks to evaluate the performance of time-series models. When multiple trigonometric functions $f_i(x)$ of different scales and frequencies are stacked together, they can form complicated sequential patterns. To evaluate `MAT`, we model $f(x) = \sum_i f_i(x)$, where

$$f_1 = \sin\left(\frac{x\pi}{30}\right), \quad f_2 = \sin\left(\frac{(x+33)\pi}{77}\right), \quad f_3 = \sin\left(\frac{(x-42)\pi}{130}\right), \quad f_4 = \sin\left(\frac{(x+70)\pi}{60}\right).$$

We sampled 5000 sequences of length 129 with random initial points. `MAT` uses fewer weights (80,746) than the baseline, a single-layer RNN (182,658). As shown in Figure 2, as the sequence length increases, RNN fails to capture the pattern, while `MAT` can successfully recover the ground truth sequence.

3.2 Basketball Behavior Cloning

Data We applied `MAT` on the tracking data of professional basketball players (107,146 training and 13,845 test sequences). Each sequence has the (x, y)-coordinates of 5 offensive basketball player trajectories for 50 frames (sampled at 6.25Hz for 8 seconds) and takes place in the left half-court. Training data for sub-models of different levels are sampled at different rates. For example, we use frames (0, 16, 32, 48) for M_1, frames (0, 8, 16, 24, 32, 40, 48) for M_2, etc.

Baselines and Metrics We compare our model with the following 4 baselines:

- Plain RNN model with 400 2-layer GRU cells for the hidden state, 1,581,220 params.
- Plain RNN model *with adversarial training*, 1,581,220 params.

Table 1: Basketball Trajectory Metrics Comparison. Better models should have stats that are closer to the expert. Our MAT has competitive performance without using macro-goal data.

Models	RNN	RNN + GAN	HVRNN	HRNN + GAN	MAT	Expert
Path Length	1.36(+138%)	0.62(+9%)	0.67(+18%)	0.62(+9%)	**0.53(-8%)**	0.57
OOB Rate (10^{-3})	29.2(+1395%)	4.33(+122%)	7.16(+266%)	**3.70(+89%)**	4.02(+106%)	1.95
Step size Change (10^{-3})	10.0(+403%)	2.2(+11%)	2.7(+36%)	**2.03(+2%)**	2.13(+7%)	1.99
Path Difference	1.07(+91%)	0.41(-27%)	0.59(+5%)	0.51(-9%)	**0.54(-5%)**	0.56

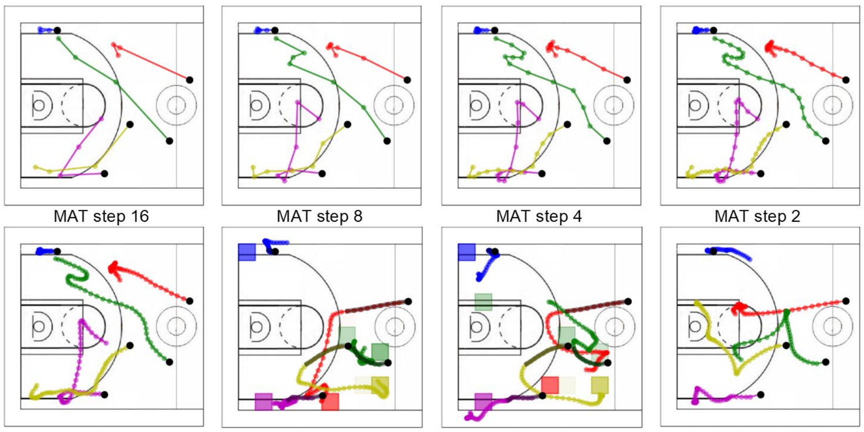

Figure 3: 1-4: Trajectories from different level MAT show that the model doesn't just linearly interpolate between two points. 5-8: Comparisons between different models show that MAT generates more realistic trajectories.

- Hierarchical Variational RNN model [13], **macro-goals required**, 4,372,190 params.
- H-RNN model [13], *with adversarial training*, **macro-goals required**, 3,285,280 params.

To evaluate the quality of generated samples, we use 4 metrics as proxies to compare the generated trajectories with real data: (1) *Average trajectory length* measures the macro strategies of players. (2) *Average out-of-bound rate* measures whether the model learns the "rule" of the game. (3) *Average step size change* quantifies the relationship between consecutive actions, and (4) *Max-Min path difference* to analyze the cooperation within a team.

Results Table 1 shows the quantitative comparison of different models using our metrics. RNN statistics significantly deviate from the ground truth, but greatly improve with adversarial training. HVRNN uses "macro-goals", and performs reasonably w.r.t macro metrics including average path length and max-min path difference. However, the big step size changes lead to unnatural trajectories. With adversarial training, issues were alleviated but the models require manually crated macro-goals. Figure 3 (1-4) shows a sample trajectory generated by MAT at different levels. The top level generates the macro goals. At lower levels, the model further refines the trajectories given the macro goals. Note that based on predictions of the previous level, our model can capture new spatiotemporal structures of that level without restricting itself to the previous level. Figure 3 (5-8) compares the trajectories generated by 3 different models and the ground truth. MAT has more natural step changes, and there are some interesting twists that can only be found in expert and MAT trajectories.

4 Conclusions

In this work, we propose a novel deep generative model for long-range sequences, namely, multiresolution adversarial training (MAT). By coupling a multiresolution generator with adversarial objective function, MAT can generate higher-quality long-range sequences without human-defined hierarchies. We demonstrate the effectiveness of MAT using both synthetic data as well as real-world basketball play trajectories. Experiment results suggest MAT can capture complicated temporal structure and generate more realistic long-range sequences.

References

[1] Nir Baram, Oron Anschel, Itai Caspi, and Shie Mannor. End-to-end differentiable adversarial imitation learning. In *International Conference on Machine Learning*, pages 390–399, 2017.

[2] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[3] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, pages 4565–4573, 2016.

[4] Sham Machandranath Kakade et al. *On the sample complexity of reinforcement learning*. PhD thesis, University of London London, England, 2003.

[5] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

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

[7] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[8] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pages 1310–1318, 2013.

[9] Scott Reed, Aäron van den Oord, Nal Kalchbrenner, Sergio Gómez Colmenarejo, Ziyu Wang, Dan Belov, and Nando de Freitas. Parallel multiscale autoregressive density estimation. *arXiv preprint arXiv:1703.03664*, 2017.

[10] Nicholas Rhinehart and M Kris. R2p2: A reparameterized pushforward policy for diverse, precise generative path forecasting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 772–788, 2018.

[11] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *Advances In Neural Information Processing Systems*, pages 613–621, 2016.

[12] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *AAAI*, pages 2852–2858, 2017.

[13] Eric Zhan, Stephan Zheng, Yisong Yue, and Patrick Lucey. Generative multi-agent behavioral cloning. *arXiv preprint arXiv:1803.07612*, 2018.

[14] Stephan Zheng, Yisong Yue, and Jennifer Hobbs. Generating long-term trajectories using deep hierarchical networks. In *Advances in Neural Information Processing Systems*, pages 1543–1551, 2016.