Normalizing Flow Model for Policy Representation in Continuous Action Multi-agent Systems

Extended Abstract

Xiaobai Ma
Stanford University
maxiaoba@stanford.edu

Jayesh K. Gupta
Stanford University
jkg@cs.stanford.edu

Mykel J. Kochenderfer
Stanford University
mykel@stanford.edu

ABSTRACT

Neural networks that output the parameters of a diagonal Gaussian distribution are widely used in reinforcement learning tasks with continuous action spaces. They have had considerable success in single-agent domains and even in some multi-agent tasks. However, general multi-agent tasks often require mixed strategies whose distributions cannot be well approximated by Gaussians or their mixtures. This paper proposes an alternative for policy representation based on normalizing flows. This approach allows for greater flexibility in action distribution representation beyond mixture models. We demonstrate their advantage over standard methods on a set of imitation learning tasks modeling human driving behaviors in the presence of other drivers.

ACM Reference Format:
Xiaobai Ma, Jayesh K. Gupta, and Mykel J. Kochenderfer. 2020. Normalizing Flow Model for Policy Representation in Continuous Action Multi-agent Systems. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020,* IFAAMAS, 3 pages.

1 INTRODUCTION

The multi-agent learning (MAL) literature is replete with examples of multiple self-interested agents with imperfect information that are strategically interacting with each other [2, 20]. Imperfect information as well as strategic interaction conditions require agents that can both model complex strategies of other interacting agents and formulate complex strategies in response. This often requires action distributions that are multi-modal to model the effects of hidden latent variables under imperfect information and to avoid being predictable to other agents.

However, past work has largely focused on discrete action domains or population-based methods [9]. Single agent continuous control tasks are often modeled as either deterministic policies [14] or unimodal multivariate Gaussian with diagonal covariances as stochastic policies [19]. Experimentally, we find that unimodal Gaussian distribution representations of stochastic policies can also restrict the model class leading to suboptimal performance in multi-agent domains. Consequently, devising methods to learn complex policy representations required for multi-agent systems is a significant challenge.

Density modeling is a rich field of study. Mixture models are often used to build multi-modal distributions from unimodal ones. This approach can be effective when the degree of multi-modality is known. However, complex multi-agent interactions often require more flexible distributions than those achieved by mixture models. Recent advances in generative modeling [3, 6, 7, 11, 12] have shown promise for modeling complex distributions. Various normalizing flow [3–5, 8, 10, 17] models allow learning complex distributions while maintaining ease of sampling and density evaluation.

In this work, we show two examples of multi-agent problems that require complex action distributions for agent policies. We show how normalizing flow models can be architected to act as useful policy representations and how they fare against Gaussian mixture model representations.

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 9–13, 2020, Auckland, New Zealand.

2 NORMALIZING FLOW POLICY REPRESENTATION

Flow models are invertible transformations that map observed data $\mathbf{x}$ to a latent variable $\mathbf{z}$ from a simpler distribution, such that both computing the probability density $p(\mathbf{x})$ and sampling $x \sim p(\mathbf{x})$ is efficient. Represented as a function f, the key idea is to stack individual simple invertible transformations [3, 4] as $f(\mathbf{x}) = f_1 \circ \cdots \circ f_L(\mathbf{x})$, with each f_i having a tractable inverse and a tractable Jacobian determinant. We focus on RealNVP [4] as our exemplary flow model. It uses an *affine coupling layer* as f_i. Given a D dimensional input $\mathbf{x}$ and $d < D$, the D dimensional output $\mathbf{y}$ from application of f_i is defined as:

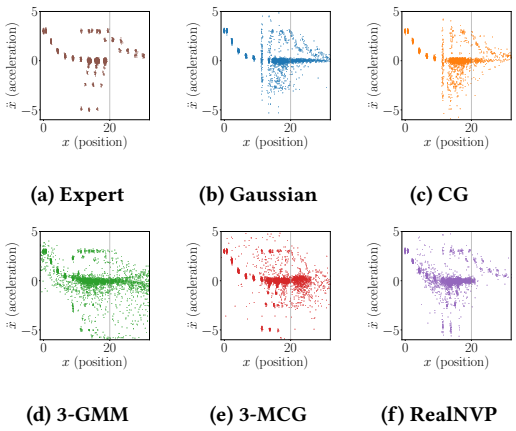
$$\mathbf{y}_{1:d} = \mathbf{x}_{1:d}; \quad \mathbf{y}_{d+1:D} = \mathbf{x}_{d+1:D} \odot e^{\alpha(\mathbf{x}_{1:d})} + t(\mathbf{x}_{1:d}) \tag{1}$$

where α and t are scale and translation functions from $\mathbb{R}^d \to \mathbb{R}^{D-d}$ and $\odot$ is the Hadamard product. These functions are represented by neural networks.

Conditioning the flow distribution $p(\mathbf{x})$ representing the action distribution, on some state s, allows us to use it as a representation for a policy network. Formally, we want to transform $\mathbf{z} \sim q$ to a policy, $a \sim \pi(a \mid s)$ by defining $a = f_\theta^{-1}(\mathbf{z}, s)$, whose log-likelihood is calculated as:

$$\log p(\mathbf{x} \mid s) = \log q(f(\mathbf{x}, s)) + \sum_{i=1}^{L} \log \left| \det \frac{\partial f_i(\mathbf{y}, s)}{\partial f_{i-1}(\mathbf{y}, s)} \right| \tag{2}$$

We propose to incorporate this change by replacing $f_\theta^{-1}(\mathbf{z}_m)$ with $f_\theta^{-1}(\mathbf{z}_m, s)$ in each flow. For RealNVP, this implies substituting $\alpha(\mathbf{x}_{1:d})$ and $t(\mathbf{x}_{1:d})$ with $\alpha(\mathbf{x}_{1:d}, s)$ and $t(\mathbf{x}_{1:d}, s)$. In practice, this means that the neural networks representing α and t take both $\mathbf{x}_{1:d}$ and s as input.



(a) Expert (b) Gaussian (c) CG

(d) 3-GMM (e) 3-MCG (f) RealNVP

Figure 1: Traffic Light Intersection: Distribution of agent acceleration along the road for different policy parameterizations. The gray line indicates the intersection location.

3 EXPERIMENTS

We evaluate flow policies for a synthetic and a real-world agent-modeling task, in multi-agent contexts. We use behavior cloning [16] to maximize the likelihood of actions in the training data.

We compare flow policies against the standard diagonal multivariate Gaussian policies, as well as the Gaussian policies with full covariance using Cholesky decomposition (denoted as CG). We also compare against the mixture of multivariate diagonal or full covariance Gaussian policies (denoted as GMM and MCG).

Synthetic. To verify that flow policies can learn to represent multi-modal behavior, we designed a simple environment to model human driving in response to a traffic light at an intersection scenario. As soon as the traffic light turns yellow, the driver either needs to *accelerate* or *decelerate* to avoid coming into conflict with the orthogonal traffic. Figure 1a shows the sample expert acceleration along the road ($\ddot{x}$) with noticeable multi-modal behavior. Some drivers decelerate and show negative $\ddot{x}$, while some others accelerate and display a positive $\ddot{x}$. We do a 10-fold cross validation and the test scores are reported in Table 1.

The performance of learned policies can be evaluated by sampling from them for the same batch of initial states. Figures 1b and 1c show that the single Gaussian policies fail to capture the expert action distribution. It tries to cruise along at the same speed even when closer to the intersection. Figures 1d and 1e as well as Table 1 suggest that mixture models help a little and have lower spread beyond the intersection along constant speeds. We see marked improvement in agent modeling from our conditional RealNVP, as can be seen in Figure 1f with very little spread of constant speeds beyond the intersection point.

Real World. Schmerling et al. [18] demonstrate the importance of modeling complex action distributions in human-robot interaction policies in the traffic-weaving scenario.[1] Two drivers intend to swap lanes without communication in a 135 meters straight road.

[1]Dataset from https://github.com/StanfordASL/TrafficWeavingCVAE

Policy	Traffic Light Intersection	Traffic Weaving
Gaussian	1.21 ± 0.02	−1.94 ± 0.10
Cholesky Gaussian	1.20 ± 0.02	−1.79 ± 0.21
k-Gaussian mixture	1.40 ± 0.07	0.19 ± 0.14
k-Cholesky Gaussian mixture	1.38 ± 0.07	0.17 ± 0.18
RealNVP	**1.46 ± 0.04**	**0.86 ± 0.25**

Table 1: Average best test log-likelihood scores. Best scores for Gaussian mixture models were achieved with $k = 3$ for Traffic Light Intersection, and $k = 20$ for Traffic Weaving. Higher is better.

The dataset contains 1105 trials recorded from 19 different pairs of human drivers.

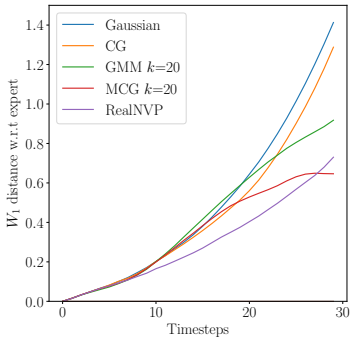


Figure 2: Traffic Weaving: First Wasserstein distance with respect to the test set expert trajectories.

The 10-fold cross validation scores are reported in Table 1. On this dataset, the RealNVP policy again has the highest test score.

Due to the complexity of true human behaviors, the quality of the generated trajectories from the trained policies could not be visually differentiated. We instead compute the per-timestep first Wasserstein distance [15] of vehicle positions between the expert trajectories in the test set and the rollout trajectories generated by the learned policies using the same initial conditions. The results are shown in Figure 2. The RealNVP policy has the lowest distance on almost every time step, indicating that the trajectories sampled from the RealNVP policy distribution are closest to the demonstration distribution compared to other approaches.

4 CONCLUSION

We focused on the task of choosing the best representation for agent policies in multi-agent continuous control contexts. We showed how even mixture models may not suffice for tractably modeling the required complex multi-modal action distributions for optimal behavior. Using conditional normalizing flows, we found significant performance improvements in learning complex, multi-modal agent behaviors. Incorporating our model with the recent developments in imitation learning [1] and reinforcement learning [13] for multi-agent systems is important future work.

Acknowledgements

This work was partially supported by Stanford Center for AI Safety.

REFERENCES

[1] Raunak Bhattacharyya, Derek J. Phillips, et al. 2019. Simulating emergent properties of human driving behavior using multi-agent reward augmented imitation learning. In *International Conference on Robotics and Automation (ICRA)*.

[2] L. Busoniu, R. Babuska, and B. De Schutter. 2008. A Comprehensive Survey of Multiagent Reinforcement Learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 38, 2 (March 2008), 156–172. https://doi.org/10.1109/TSMCC.2007.913919

[3] Laurent Dinh, David Krueger, and Yoshua Bengio. 2014. NICE: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516* (2014).

[4] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. 2016. Density estimation using Real NVP. *arXiv preprint arXiv:1605.08803* (2016).

[5] Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. 2015. Made: Masked autoencoder for distribution estimation. In *International Conference on Machine Learning (ICML)*. 881–889.

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS)*. 2672–2680.

[7] Karol Gregor, Ivo Danihelka, Andriy Mnih, Charles Blundell, and Daan Wierstra. 2014. Deep AutoRegressive Networks. In *International Conference on Machine Learning (ICML)*. 1242–1250.

[8] Jonathan Ho, Xi Chen, Aravind Srinivas, Yan Duan, and Pieter Abbeel. 2019. Flow++: Improving Flow-Based Generative Models with Variational Dequantization and Architecture Design. In *International Conference on Machine Learning (ICML)*, Vol. 97. PMLR, 2722–2730.

[9] Max Jaderberg, Wojciech M Czarnecki, Iain Dunning, Luke Marris, Guy Lever, Antonio Garcia Castañeda, Charles Beattie, Neil C Rabinowitz, Ari S Morcos, Avraham Ruderman, Nicolas Sonnerat, Tim Green, Louise Deason, Joel Z Leibo, David Silver, Demis Hassabis, Koray Kavukcuoglu, and Thore Graepel. 2019. Human-level performance in 3D multiplayer games with population-based reinforcement learning. *Science* 364, 6443 (May 2019), 859–865.

[10] Diedrik P. Kingma and Prafulla Dhariwal. 2018. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems (NeurIPS)*. 10215–10224.

[11] Diederik P. Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. 2016. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems (NeurIPS)*. 4743–4751.

[12] Diederik P. Kingma and Max Welling. 2013. Auto-Encoding Variational Bayes. In *International Conference on Learning Representations (ICLR)*.

[13] Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, et al. 2017. A unified game-theoretic approach to multiagent reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*. 4190–4203.

[14] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971* (2015).

[15] Gabriel Peyré, Marco Cuturi, et al. 2019. Computational optimal transport. *Foundations and Trends® in Machine Learning* 11, 5-6 (2019), 355–607.

[16] Dean Pomerleau. 1991. Efficient training of artificial neural networks for autonomous navigation. *Neural Computation* 3 (1991), 88–97.

[17] Danilo Rezende and Shakir Mohamed. 2015. Variational Inference with Normalizing Flows. In *International Conference on Machine Learning (ICML)*. 1530–1538.

[18] Edward Schmerling, Karen Leung, Wolf Vollprecht, and Marco Pavone. 2017. Multimodal Probabilistic Model-Based Planning for Human-Robot Interaction. In *IEEE International Conference on Robotics and Automation (ICRA)*. 1–9.

[19] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[20] Sriram Srinivasan, Marc Lanctot, Vinicius Zambaldi, Julien Pérolat, Karl Tuyls, Rémi Munos, and Michael Bowling. 2018. Actor-critic policy optimization in partially observable multiagent environments. In *Advances in Neural Information Processing Systems (NeurIPS)*. 3422–3435.