Enhanced Deep Q-Learning with Gaussian Mixtures

Extended Abstract

Chainesh Gautam
IIIT Bangalore
Bangalore, India
chainesh.gautam@iiitb.ac.in

Chandramouli Kamanchi
IBM Research
Bangalore, India
chandramouli.kamanchi@ibm.com

Raghuram Bharadwaj Diddigi*
IIIT Bangalore
Bangalore, India
raghuram.bharadwaj@iiitb.ac.in

ABSTRACT

Value-based reinforcement learning methods, like Deep Q-Networks (DQNs), typically estimate returns by minimizing the mean squared error between predicted and target values. From a Bayesian standpoint, this procedure implicitly assumes that returns follow a unimodal Gaussian distribution, with parameters learned via maximum likelihood estimation. However, this assumption can be limiting in environments characterized by high stochasticity or complex reward dynamics, where capturing uncertainty and multi-modality in the return distribution is critical for robust decision-making. We propose Gaussian Mixture Q-Networks (GQN), a novel extension of Q-learning that models return distribution as a mixture of Gaussians. Architecturally, GQN can be interpreted as a mixture-of-experts Q-learning algorithm, where each Gaussian component acts as an expert head and mixture weights are adaptively updated via temporal-difference responsibilities inspired by Expectation–Maximization. We evaluate GQN on the Atari benchmark suite and observe improvements in both learning stability and final performance compared to standard DQN baselines.

KEYWORDS

Deep Q-Learning; Gaussian Mixture Model

ACM Reference Format:
Chainesh Gautam, Chandramouli Kamanchi, and Raghuram Bharadwaj Diddigi. 2026. Enhanced Deep Q-Learning with Gaussian Mixtures: Extended Abstract. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 3 pages. https://doi.org/10.65109/KVOL6969

1 INTRODUCTION

Deep Q-Networks (DQNs) [4] have emerged as a powerful approach for approximating optimal Q-values in high-dimensional and complex environments by using deep neural networks. By combining Q-learning with function approximation and experience replay, DQN has achieved remarkable success on challenging benchmarks [1, 3, 8]. Standard DQN, however, assumes that the return distribution is unimodal and represents the Q-value as a single scalar expectation. This assumption can limit performance in environments with inherently multi-modal return structures, where the variance or shape of the return distribution varies significantly across different regions of the state-action space. Capturing richer information about the distribution of returns is key to improving the quality of the learned Q-function, especially in highly stochastic or partially observable domains.

To address this limitation, we propose Gaussian Mixture Q-Networks (GQN), a novel extension of Deep Q-Learning. GQN represents the action-value function as a mixture of experts, where each expert corresponds to a Gaussian component predicting a mean return, and the final Q-value is obtained as a weighted sum over these components. Unlike standard Mixture-of-Experts (MoE) approaches [6, 10] that rely on an auxiliary gating network, our method derives the routing probabilities (mixture weights) directly from a probabilistic formulation: we model returns as a Gaussian mixture and update both the component means and the mixing coefficients through a temporal-difference responsibility mechanism inspired by Expectation–Maximization [5]. This formulation enhances the expressiveness of the Q-function, provides a principled way to route learning signals among experts, and leads to more stable and robust value learning, especially in complex environments.

2 BACKGROUND

We consider a Markov Decision Process with states $s \in \mathcal{S}$, actions $a \in \mathcal{A}$, reward $r(s,a)$, transition dynamics $\mathcal{P}(s' \mid s,a)$, and discount factor $\gamma \in (0,1)$ [7]. A policy ν maps states to actions. The action value function of a policy ν is

$$Q^{\nu}(s,a) = \mathbb{E}_{\nu}\Big[\sum_{t=0}^{\infty} \gamma^t r_t \;\Big|\; s_0 = s,\ a_0 = a\Big].$$

The optimal action value function Q^* satisfies the Bellman optimality equation [2]

$$Q^*(s,a) = \mathbb{E}_{s'}\Big[r(s,a) + \gamma \max_{a'} Q^*(s',a') \;\Big|\; s,a\Big],$$

and the greedy optimal policy is $\nu^*(s) = \arg\max_a Q^*(s,a)$.

Q-learning estimates Q^* from sampled transitions (s,a,r,s') [9]. In deep Q-learning, a neural network $Q_\theta(s,a)$ is trained using a one step temporal difference target

$$y = r + \gamma \max_{a'} Q_{\bar{\theta}}(s',a'),$$

where $Q_{\bar{\theta}}$ is a lagged target network as in DQN [4]. The standard objective minimizes the squared temporal difference error

$$\mathcal{L}_{\mathrm{DQN}}(\theta) = \mathbb{E}\Big[\big(y - Q_\theta(s,a)\big)^2\Big].$$

*Dr. Raghuram Bharadwaj is supported by the Anusandhan National Research Foundation (ANRF) under the Prime Minister Early Career Research Grant ANRF/ECRG/2024/005235/ENS

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/KVOL6969

This squared error objective is equivalent to maximum likelihood estimation [5] under a fixed variance Gaussian noise model, which motivates our mixture based reparameterization in the next section.

3 GAUSSIAN MIXTURE Q-NETWORK

In our proposed model, the Q-value for a given state s and action a is modeled as a weighted sum of multiple Gaussian components. Each component, indexed by $i \in \{1, 2, \ldots, n_{\text{gaussians}}\}$, is parameterized by a mean $\mu_i(s, a)$, a shared variance σ^2, and a mixing coefficient $\pi_i(a)$. The means $\mu_i(s, a)$ are approximated by a neural network $\mathbf{f}(s; \theta)$, which takes the state s as input and outputs $\mu_i(s, a)$ for all actions a and all components i. That is,

$$\mu_i(s, a) = \mathbf{f}(s; \theta)_{i,a},$$

where $\mathbf{f}(s; \theta)_{i,a}$ denotes the output corresponding to action a from the i-th Gaussian component head of the network.

The mixing coefficients $\pi_i(a)$ govern the contribution of each component to the Q-value for action a, and are constrained such that $\sum_{i=1}^{n_{\text{gaussians}}} \pi_i(a) = 1$ for every a. The Q-values are computed as:

$$Q(s, a) = \sum_{i=1}^{n_{\text{gaussians}}} \pi_i(a)\, \mu_i(s, a), \tag{1}$$

which allows the model to blend the predicted means across components to approximate complex return distributions. The complete pseudocode of the proposed algorithm is provided in Algorithm 1.

4 EXPERIMENTS

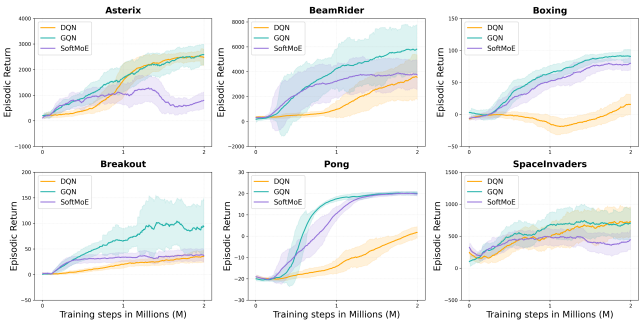

Figure 1: Training performance of DQN, GQN ($N_{\text{gaussians}}$=5), and Soft-MoE (5 experts) across six Atari games over 2M frames. Curves show the running average episodic return, and shaded regions indicate variability across five seeds.

We compare DQN [4], GQN, and Soft MoE [6] on six Atari games [1]. All agents use the same encoder and the same training pipeline, and we report the running average episodic return averaged over five seeds. Figure 1 shows learning curves over the fixed training budget. We use five components for GQN and five experts for Soft MoE to keep model capacity comparable.

Across all six games, GQN improves learning stability and final return compared to DQN and Soft MoE. The gains are strongest in games where temporal difference targets are more heterogeneous due to environment stochasticity and bootstrap error[3]. GQN helps because different components can specialize on different target regimes, so updates interfere less through shared parameters. Responsibilities assign soft credit to the components that best explain the current target, which reduces noisy gradient mixing and stabilizes value fitting. In simpler settings, the mixture often concentrates on a single dominant component, which matches the intuition that a single value head is sufficient.

Algorithm 1 Gaussian Mixture Deep Q-Learning (GQN)

Initialize replay buffer $\mathcal{D}$ to capacity N.
Initialize mixing coefficients $\pi_i(a) = \frac{1}{n_{\text{gaussians}}}$ uniformly for all actions $a \in A$.
Initialize fixed variance $\sigma^2_{i,a} = 1.0$ for all i gaussian components and actions a.

for episode = 1 to M **do**
  Initialize state s
  for t = 1 to T **do**
    With probability ϵ, select a random action a_t
    Otherwise, select $a_t = \arg\max_a Q(s, a)$, where

$$Q(s, a) = \sum_{i=1}^{n_{\text{gaussians}}} \pi_i(a) \cdot \mathbf{f}(s; \theta)_{i,a}$$

    Execute action a_t, observe reward r_t, and next state s'
    Store transition (s, a, r, s') in $\mathcal{D}$
    Sample minibatch of transitions $(s_j, a_j, r_j, s'_j)_{j=1}^{m}$ from $\mathcal{D}$
    Compute TD target:

$$y_j = r_j + \gamma \max_b \sum_{i=1}^{n_{\text{gaussians}}} \pi_i(b) \cdot \mathbf{f}(s'_j; \theta)_{i,b}$$

    Compute responsibilities $\eta_{i,a}$ for each Gaussian:

$$\eta_i^{(j)} = \frac{\pi_i(a_j)\, \mathcal{N}\left(y_j \mid \mu_i(s_j, a_j),\, \sigma^2\right)}{\sum_{k=1}^{n_{\text{gaussians}}} \pi_k(a_j)\, \mathcal{N}\left(y_j \mid \mu_k(s_j, a_j),\, \sigma^2\right)},\ \forall i$$

    Perform gradient descent step on:

$$\mathcal{L}(\theta) = \frac{1}{m} \sum_{j=1}^{m} \sum_{i=1}^{n_{\text{gaussians}}} \eta_i^{(j)} \left(y_j - \mathbf{f}(s_j; \theta)_{i,a_j}\right)^2.$$

    Update weights of neural network for entire minibatch:

$$\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}(\theta),$$

    Define $\eta_{i,a} = \frac{1}{|\{k:\, a_k = a\}|} \sum_{j \in \{k:\, a_k = a\}} \eta_i^{(j)}$
    Update mixture weights:

$$\pi_{i,a} \leftarrow (1 - \beta)\pi_{i,a} + \beta \eta_{i,a},\ \forall i, a$$

    Normalize $\pi_{i,a}$ to sum to 1

5 CONCLUSION

In this work, we propose a novel Deep Q-learning algorithm inspired by Gaussian Mixture Models (GMMs). The core idea is to model the return distribution as a mixture of Gaussians, offering a richer and more flexible representation. Building on this perspective, we develop a modified Q-learning approach. We demonstrate that our algorithm achieves notable performance improvements on complex Atari environments.

REFERENCES

[1] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling. 2013. The Arcade Learning Environment: An Evaluation Platform for General Agents. *Journal of Artificial Intelligence Research* 47 (jun 2013), 253–279.

[2] Richard Bellman. 1966. Dynamic programming. *Science* 153, 3731 (1966), 34–37.

[3] Marlos C. Machado, Marc G. Bellemare, Erik Talvitie, Joel Veness, Matthew J. Hausknecht, and Michael Bowling. 2018. Revisiting the Arcade Learning Environment: Evaluation Protocols and Open Problems for General Agents. *Journal of Artificial Intelligence Research* 61 (2018), 523–562.

[4] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nature* 518, 7540 (Feb. 2015), 529–533. http://dx.doi.org/10.1038/nature14236

[5] Todd K Moon. 1996. The expectation-maximization algorithm. *IEEE Signal processing magazine* 13, 6 (1996), 47–60.

[6] Johan Obando-Ceron, Ghada Sokar, Timon Willi, Clare Lyle, Jesse Farebrother, Jakob Foerster, Gintare Karolina Dziugaite, Doina Precup, and Pablo Samuel Castro. 2024. Mixtures of experts unlock parameter scaling for deep rl. *arXiv preprint arXiv:2402.08609* (2024).

[7] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction* (second ed.). The MIT Press. http://incompleteideas.net/book/the-book-2nd.html

[8] Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, et al. 2024. Gymnasium: A Standard Interface for Reinforcement Learning Environments. *arXiv preprint arXiv:2407.17032* (2024).

[9] Christopher JCH Watkins and Peter Dayan. 1992. Q-learning. *Machine learning* 8 (1992), 279–292.

[10] Timon Willi, Johan Obando-Ceron, Jakob Foerster, Karolina Dziugaite, and Pablo Samuel Castro. 2024. Mixture of Experts in a Mixture of RL settings. arXiv:2406.18420 [cs.LG] https://arxiv.org/abs/2406.18420