Increasing Critic Robustness with Degenerate Distribution Mixtures in Discrete Action Reinforcement Learning

Samuel Blad[1,2], Martin Längkvist[1,2], and Amy Loutfi[1,2]

***Abstract*— Reinforcement Learning (RL) commonly employs critics to assist in training the actor, often using deterministic or categorical action-value approximations. While continuous action spaces have their merits, they tend to suffer from expressiveness constraints and introduce limitations when calculating entropy. We propose a novel method, Degenerate Distribution Mixture (DDM), designed to combine the strengths of deterministic policy gradient (DPG) style gradients and discrete action distributions. Unlike existing techniques, DDM trains the critic on a mixture of the policy distribution and the degenerate distribution of sampled actions. This allows the critic to guide the actor efficiently towards actions that maximize the expected return while meeting entropy regularization constraints. Initial experiments suggest that our DDM approach has the potential for faster training times and may offer improved stability compared to existing techniques like Gumbel noise-based methods. While these findings are preliminary, they hint at the promise of more computationally efficient and robust RL methods that could be developed using our approach across diverse action spaces.**

I. Introduction

In Reinforcement Learning, training an actor frequently involves a critic to backpropagate reward signals through the Bellman equation, associating individual state-action transitions with future rewards. Pioneering works, like REINFORCE [1], employed critics that produce a single value for each state-action pair to instruct the actor.

However, performing an exhaustive sum over all actions is expensive. As demonstrated in [2], the variance of the value scales linearly with the sample vector's dimensions. In continuous action spaces, DPG-style algorithms [3] allow the critic to provide a gradient across all action parameters, leading to enhanced performance in settings like DDPG [4], or SAC [5]. Additionally, a compact parametrization can be defined, e.g., a Gaussian action distribution simply by its mean and variance vectors. This retains the semantic continuity in the action space, unlike the segmented nature of categorically subdividing the action space. However, one can utilize convolutional layers over the bins or employ a spatial autoencoder to enhance the continuity in categorical distributions as well, as illustrated in [6].

Nevertheless, continuous action spaces present challenges. Basic parametrizations, such as a uni-modal Gaussian distribution, may lack expressiveness. Actions, typically having bounded values, don't always fit into infinite-domain distributions. While transformations like the tanh function can squash distributions, they introduce complications. For instance, an augmented variance might result in reduced entropy, as depicted in Fig. 1. Additionally, the entropy term of a continuous distribution can cause numerical instability due to its potentially infinite range, unlike the bounded nature of the entropy in a categorical distribution.

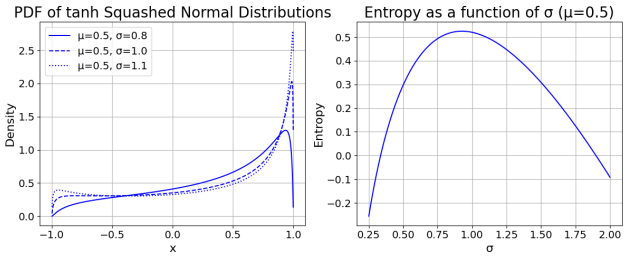

Fig. 1. When a tanh function is applied to samples from a normal distribution, for high values, there's an observed decrease in entropy as the standard deviation increases.

Prior solutions that attempt to integrate the advantages of DPG-style gradients with a categorical action distribution face challenges. For instance, adapting the SAC framework to use distribution parameters as actions allows for entropy regularization but complicates the critic's role. The critic, which should primarily assess the impact of individual actions on returns, finds it difficult to do so when the actor's policy converges to high entropy distributions, like a uniform policy. This issue compromises the critic's ability to guide the actor effectively.

In this paper, we introduce the Degenerate Distribution Mixture (DDM) method to address this challenge. We leverage a hybrid approach where the critic is trained on a mixture of the actor's policy and the degenerate distribution of sampled actions. This enables the critic to maintain a nuanced representation that effectively steers the actor toward an optimal policy.

Our work offers a potential middle ground by combining DPG's optimization efficiency in continuous action spaces with the expressiveness of discrete actions. This synergy may help circumvent computational bottlenecks and instability, commonly seen in discrete settings and the lack of expressiveness in continuous ones. While our findings are still preliminary, they suggest that our approach could pave the way for more balanced and versatile RL methods.

II. Degenerate Distribution Mixture (DDM)

In discrete action settings, a policy offers a categorical distribution, denoted as π, contingent on a specific state s.

[1] Center for Applied Autonomous Sensor Systems, Örebro University, Örebro, Sweden
[2] firstname.lastname@oru.se

An action a is sampled from π to transition the environment to its subsequent state.

Training a critic solely based on the pair (s, π) deprives it of the ability to leverage insights about how the action a influences the environment's transition, leading to a noisy feedback loop for the critic. On the other hand, one could contemplate training the critic on the degenerate distribution π_a, which consistently results in action a, echoing the strategy employed by REINFORCE. Nonetheless, this approach creates a hurdle: the critic never receives training on the diverse, non-degenerate vectors provided by the actor. Consequently, when such vectors are input into the critic following the DPG paradigm, the output values become unreliable.

For the critic to efficiently guide the actor, it must be adept at approximating expected values for vectors spanning the continuum between π and π_a. Such a configuration ensures that the gradients emanating from the critic naturally shepherd the actor toward the degenerate distribution perceived to generate the maximal return, all the while maintaining entropy regularization constraints.

To address this, the DDM approach trains the critic on a vector π_c situated linearly between π and π_a. The specific position of π_c on this linear path is governed by a probabilistic distribution.

In this work, we evaluate three distinct variants of the Beta distribution. The Beta distribution parameterized by $\beta(0.1, 0.1)$ exhibits pronounced density proximate to π (representing the current position) and π_a (representing the intended destination), while maintaining a nearly uniform, albeit low, density for intermediary values. We also consider the uniform distribution $\beta(1, 1)$, which offers a constant density across all policy values. Lastly, we study a one-sided variant, $\beta(0.1, 1)$, which provides a high density in proximity to the current policy π and gradually decays to its lowest density at π_a.

Algorithm 1 delineates the procedural flow of the sampling process as per the DDM methodology.

Algorithm 1 DDM sampling

Input: Policy distribution π, mixing distribution D
$a \leftarrow$ Sample action from π
Create π_a such that $\pi_a(a') = \begin{cases} 1 & \text{if } a' = a \\ 0 & \text{otherwise} \end{cases}$
$\delta \leftarrow$ Sample weight from D
$\pi_c \leftarrow (1 - \delta) \cdot \pi + \delta \cdot \pi_a$
Output: Environment action a, Critic action π_c

Given a sample from the replay buffer (s, a, r, s', d) where r is the reward from this transition, s' is the next state, and d is the termination flag, the loss L_C for the critic C and loss L_A for the actor A are:

$$q(r, s', d) = r + \gamma(1 - d)(C(s', A(s')) + \alpha H(A(s')))$$
$$L_C(s, a, r, s', d) = (C(s, \pi_c) - q(r, s', d))^2$$
$$L_A(s) = -(C(s, A(s)) + \alpha H(A(s)))$$

Here, π_c is derived from a as described Algorithm 1, γ denotes the discount factor, $H(\cdot)$ denotes the entropy of a distribution, and α is the weight of the entropy regularization. We also incorporate double q-learning [7] and target networks with Polyak averaging [8] to stabilize the training process of the critic. It's worth noting that, during the training phase of the actor, its output π is directed to the critic without undergoing any blend with the degenerate distribution.

The critic remains on-policy as actions a are always sampled from π and the target value for the critic $q(r, s', d)$ is computed on $A(s')$ directly. However, the critic is now additionally tasked with considering the implications of a hypothetical lower-entropy policy that would also yield the same current action a. In this way, the critic evaluates not only the action under the actor's current policy but also under conditions where the action could have been chosen by a more deterministic policy.

III. Related Work

A common approach when dealing with discrete action spaces is to train the actor with the REINFORCE style gradients. However, other approaches are also used which enable DPG style gradients. In [9], Gumbel noise is added to the logits from the actor. The action is then the argmax of the result, however, the critic learns the softmax instead of max. This procedure of action sampling has the same distribution as directly sampling from softmax, as long as the temperature of the softmax is 1. An outline of the Gumbel Noise sampling process for comparison can be observed in Algorithm 2.

Algorithm 2 Gumbel Noise sampling

Input: Policy distribution π
$g \leftarrow$ Sample Gumbel noise
Compute new logits: $l = \log(\pi) + g$
$a \leftarrow \arg\max_{a'} l(a')$
$\pi_c \leftarrow \text{softmax}(l)$
Output: Environment action a, Critic action π_c

This approach shares various similarities with DDM. Both methods produce discrete actions a sampled from the policy distribution π. Neither train the critic on π but shift it through some differentiable process with external noise to π_c, either with Gumbel noise or distribution D in combination with the degenerate distribution π_a. Depending on the entropy of π, Gumbel noise will yield vectors π_c closer to the degenerate distribution π_a or more evenly spread out across the distribution domain. DDM will similarly yield π_c close π_a or π depending on the mixing distribution D.

The key difference between Gumbel noise and DDM is that Gumbel noise generates π_c vectors in the sub-set of the distribution domain where the closest degenerate distribution vector is π_a, whereas DDM generate π_c on the line between π and π_a, which can happen anywhere in the domain. The implication of this difference can be seen in Fig. 2, where we notice that the critic will learn different functions. For

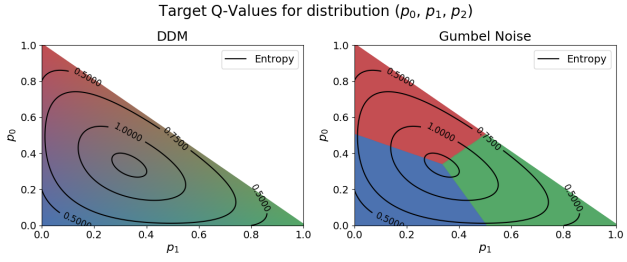

Fig. 2. The target return (i.e. the sampled action), denoted as Red/Green/Blue, for various inputs to the critic of a 3 action categorical distribution. Lower entropy distributions will have values towards the edges. When using Gumbel noise, the critic is always trained on the result of the argmax operation, when using DDM, the critic is trained on the expected result of the distribution.

Gumbel noise, the critic is trained to be a separator function between discrete actions, but using DDM the critic is trained to learn the expected return of any distribution.

The critic guides the actor to the best action through gradients. When using entropy regularization with a target entropy value, the actor has to balance the desired entropy with the value given by the critic, so it will attempt to settle somewhere along the black circles as shown in Fig. 2.

When using Gumbel noise the actor has to balance by averaging over multiple samples from the critic. When using DDM, the critic will need to learn the balanced values of the distribution through multiple samples. This transfers some of the modelling complexity from the actor to the critic.

IV. Preliminary Experiments

To assess the efficacy of DDM, we focus our investigation on three core research questions:

1) Will the inclusion of DDM outperform a critic that is directly trained on the policy π?
2) How does the performance of DDM stand in relation to existing algorithms like Gumbel noise and REINFORCE within discrete action spaces?
3) In continuous action environments, how does DDM measure up against SAC?

To validate our approach, we conducted experiments on 3 discrete action space environments [10]: Bipedal Walker (Discrete), Lunar Lander, and Cartpole, and 3 continuous action space environments [10] [11]: Bipedal Walker (Continuous), Reacher, and Swimmer. For comparison, we evaluated each environment using the same hyperparameters on two different seeds. We display the average performance per epoch for 2000 epochs. It's worth noting that some environments required more episodes to converge, such as Swimmer, while others, like Cartpole, needed fewer. To facilitate the use of DDM in continuous action environments, we discretize each action space into predefined buckets. This modification allows the policy to be modeled as a categorical distribution for each action.

Fig. 3 illustrates improvements in policy quality afforded by the integration of DDM. Without DDM, the algorithm risks failing to converge to an optimal policy. Choosing the

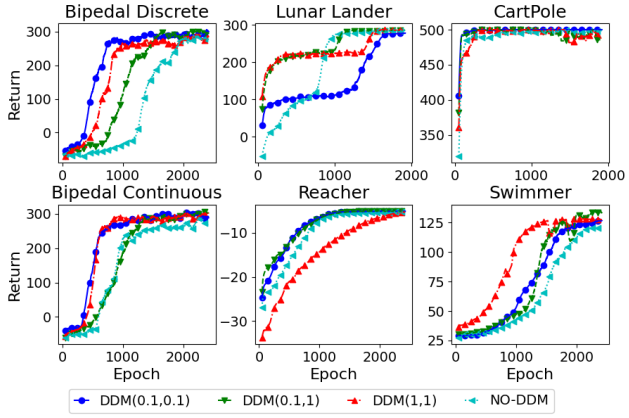

Fig. 3. Overall return when training of a critic on distribution parameters, with or without the inclusion of DDM, for various environments. In most environments, all methods ultimately converge to the same result, however for Bipedal Continuous not using DDM consistently achieves lower return. DDM seems to also speed up the learning process except for some environments where the inclusion of DDM slows down the learning process for certain choices of parameters.

right distribution D appears to have a large impact on the training time.

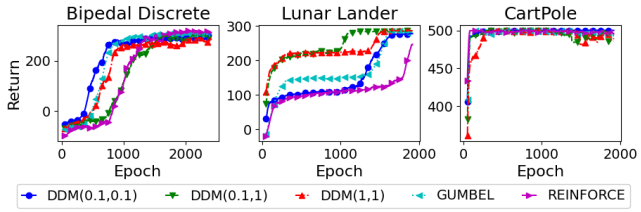

Fig. 4. Overall return of DDM, REINFORCE and Gumbel noise on environments with discrete action spaces. All methods seem to converge into the same results, but DDM is the fastest with certain parameters, and REINFORCE the slowest.

Fig. 4 provides insights into DDM's performance in discrete action spaces. DDM demonstrates comparable performance to Gumbel noise, with both algorithms significantly outperforming REINFORCE. Notably, DDM exhibits faster learning rates with certain parameters when compared to Gumbel noise. Interestingly, DDM with the uniform distribution has very similar convergence speed to Gumbel noise.

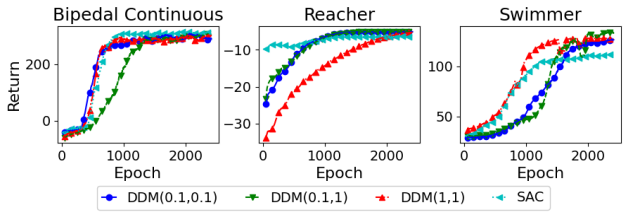

Fig. 5. Overall return of DDM and SAC on environments with continuous action spaces. DDM achieves higher end return in Swimmer, and similar for the other two. Different parametrizations of DDM achieve significantly different training times.

Lastly, Fig. 5 analyzes DDM's behavior in continuous action spaces. For two methods DDM performs on par with

SAC, however for Swimmer it surpasses the performance of SAC by a visible margin.

V. Concluding Remarks and Future Work

In this study, we explored the potential advantages of training a critic to map action distributions to their predicted outcomes, rather than individual actions. Our results, although preliminary due to a limited number of experiments and testing environments, suggest the potential for enhancing the robustness of the critic by broadening its training scope beyond the action distributions generated by the actor.

A potential drawback of DDM is the introduction of a new hyperparameter: the distribution D. This addition complicates the hyperparameter tuning process. Based on our experiments, we believe the $\beta(0.1, 1)$ distribution is a prudent choice for D. Although alternative choices for D might achieve faster convergence, the $\beta(0.1, 1)$ distribution concentrates its density around the current policy. This allows the critic to effectively gauge values in the proximate vicinity of the actor's outputs, steering the actor appropriately.

Our experiments comparing REINFORCE, the use of Gumbel noise, or SAC, with DDM, indicated faster convergence for the latter in most cases, and increased return in Swimmer, although these findings are not yet conclusive since different hyperparameters for either method might have great impact on training. Nevertheless, this opens up an intriguing area for further investigation: the possible benefits of training the critic on action distributions as opposed to individual actions. We hypothesize that a smoother target map for the critic could potentially increase the stability of actor-critic methodologies by allowing a more expressive critic to guide the actor with lower noise levels. The plateau-like behavior observed with the critic in the Gumbel noise scenario suggests that the gradients leading to optimal distributions might be challenging for the actor to navigate effectively.

An avenue for further research involves the selection of the mixing distribution D. In our study, we utilized three variations of the Beta distribution. While D might be perceived as a tunable hyperparameter, the chosen distributions notably influence the rate of convergence. This is evident in environments like Lunar Lander, Reacher, and Swimmer. We hypothesize that distributions with a high density around the policy action are generally effective. In environments where the optimal policy is more deterministic, a distribution that also has a high density near the degenerate distribution, like the uniform or $\beta(0.1, 0.1)$, might be advantageous.

Given the simplicity of DDM, we encourage further experimentation with this approach in more complex environments, characterized by intricate value functions. Such studies would allow for a more robust performance evaluation against existing methodologies.

References

[1] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine learning*, vol. 8, pp. 229–256, 1992.

[2] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proceedings of the 31st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, E. P. Xing and T. Jebara, Eds., vol. 32, no. 2. Bejing, China: PMLR, 22–24 Jun 2014, pp. 1278–1286. [Online]. Available: https://proceedings.mlr.press/v32/rezende14.html

[3] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *International conference on machine learning*. Pmlr, 2014, pp. 387–395.

[4] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[5] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, *et al.*, "Soft actor-critic algorithms and applications," *arXiv preprint arXiv:1812.05905*, 2018.

[6] C. Finn, X. Y. Tan, Y. Duan, T. Darrell, S. Levine, and P. Abbeel, "Deep spatial autoencoders for visuomotor learning," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 512–519.

[7] H. van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, Mar. 2016. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/10295

[8] B. T. Polyak and A. B. Juditsky, "Acceleration of stochastic approximation by averaging," *SIAM journal on control and optimization*, vol. 30, no. 4, pp. 838–855, 1992.

[9] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," *arXiv preprint arXiv:1611.01144*, 2016.

[10] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.

[11] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 5026–5033.