Truncated Gaussian Policy for Debiased Exploration in Continuous Control*

Ganghun Lee[1], Minji Kim[2], Minsu Lee[3,†], and Byoung-Tak Zhang[4,†]

***Abstract*— In continuous domains, Proximal Policy Optimization (PPO) generally clips excessive actions to given boundaries. However, the unbounded support of a Gaussian policy can introduce a bias toward sampling boundary actions. This bias significantly limits the effective exploration range of action selections, risking suboptimal behaviors. In this paper, we introduce a truncated Gaussian as an alternative policy distribution to mitigate this bias, showing how the choice of distribution affects exploration in action selection. However, we find that a plain truncated Gaussian policy introduces the opposite bias, favoring interior actions. To balance this bias, we ultimately propose a scale-adjusted truncated Gaussian policy, where the distribution scale shrinks when the mean is near the boundaries. This property makes boundary actions more deterministic than in a plain truncated Gaussian, but still less so than in the original Gaussian. Empirical studies across various continuous control tasks demonstrate that truncated Gaussian policies significantly reduce boundary action usage, while scale-adjusted variants effectively balance the bias and counter-bias. These methods generally outperform Gaussian policies and achieve competitive performance compared to other bias-mitigation approaches.**

I. INTRODUCTION

In continuous domains, Gaussian policies are dominant due to their strong generality and mathematical advantages, such as differentiability and simplified parameterization [1], [2], [3]. In particular, the standard deviation σ is responsible for exploration. In practice, in many continuous control tasks where the action range is limited, the unbounded support of Gaussian distributions requires careful regulation to ensure that actions stay within these limits. A common solution is to clip out-of-bound actions to the nearest boundary, as in Proximal Policy Optimization (PPO) [4]. However, this approach can introduce a bias toward boundary actions [5], [6], risking collapse into a bang-bang controller, where σ diminishes, leading to performance degradation due to loss of exploration.

In this paper, we revisit this bias as the "boundary action bias" and analyze its nature in PPO with illustrative explanations. We propose introducing a truncated Gaussian policy, which applies minimal modifications to the Gaussian distribution to accommodate bounded support. However, considering the distribution shape, our observations also imply that a plain truncated Gaussian policy may over-constrain boundary actions. To balance the boundary action bias and its counter-bias in a truncated Gaussian policy, we finally propose a scale-adjusted truncated Gaussian policy, where a discounted scale near the boundaries facilitates the sampling of necessary boundary actions.

Our empirical studies on various continuous control tasks demonstrate how distribution choice affects boundary action usage and performance. Truncated Gaussian policies significantly reduced boundary action usage, and the scale-adjusted truncated Gaussian policy generally outperformed the original Gaussian policies. It also demonstrated competitive performance compared to previous methods designed to counteract boundary action bias, achieving the highest normalized scores in MuJoCo tasks. Our findings help advance the understanding of how distributional properties influence effective exploration in continuous RL domains.

II. RELATED WORK

Exploration in RL is often governed by the choice of policy distribution, especially in continuous control where stochasticity directly determines action diversity. Some previous works attempt to bypass continuous action spaces. Discretization [7], [8] simplifies the action space; however, it suffers from the curse of dimensionality and can lead to an oversimplification of the task. The Bernoulli policy [9], which formulates the problem as selecting between only two boundary actions, has shown improved performance in specific tasks; however, many other control tasks still require continuous actions.

Rather than bypassing continuous action spaces, other works address the issue of boundary action bias. Clipped Action Policy Gradient (CAPG) [6], [10], [11], [12] corrects the policy gradient for out-of-bound actions to reduce estimation variance, thereby informing the agent of action clipping in PPO. While CAPG improves policy performance by reducing the burden of policy gradient estimation, the use of boundary actions remains at a similar level to that of standard PPO, raising questions about whether the boundary action bias has truly been mitigated. The Beta policy [5], [13], [14], [15], [16], [17] employs a Beta distribution, which inherently constrains actions without requiring clipping. However, the Beta distribution has a more complex shape and mathematical form than the Gaussian distribution, making it less straightforward to use and sometimes leading

*This submission is based on our published work at AAAI 2025.

[1]Ganhun Lee is a Ph.D. candidate in the Interdisciplinary Program in Artificial Intelligence and AIIS, Seoul National University, Seoul, Korea zxc8594@snu.ac.kr

[2]Minji Kim is a Ph.D. candidate in the Department of Computer Science and Engineering, Seoul National University, Seoul, Korea cyqkcy01@snu.ac.kr

[3]Minsu Lee is with the School of AI Convergence, Sungshin Women's University, Seoul, Korea mslee@sungshin.ac.kr

[4]Byoung-Tak Zhang is with the Department of Computer Science and Engineering and AIIS, Seoul National University, Seoul, Korea btzhang@snu.ac.kr

[†]Corresponding authors

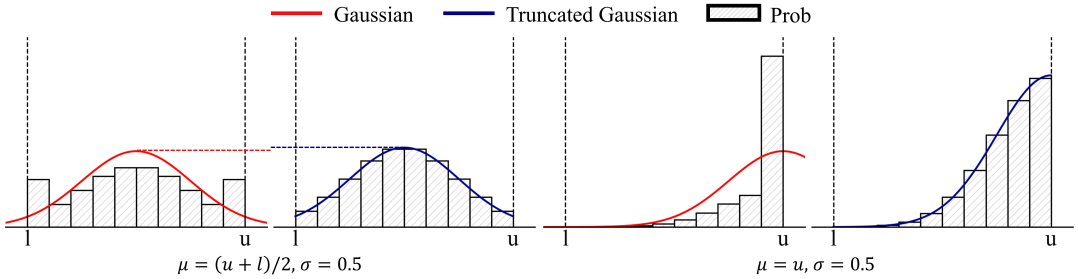

Fig. 1. Comparison of Gaussian and truncated Gaussian policies. (Left) The Gaussian PDF (red line) and the truncated Gaussian PDF (blue line) are centered within the two bounds l and u, sharing the parameters $\mu = (u+l)/2$ and $\sigma = 0.5$. Gaussian policy samples actions from the Gaussian PDF and clips out-of-bound actions to the nearest boundary, resulting in distorted action probabilities (hatched bars). Truncated Gaussian policy does not clip actions, preserving the consistency between PDF and action probabilities. (Right) The two PDFs moved to the right action bound, where $\mu = u$. The distortion has deepened with the Gaussian policy, but the truncated Gaussian policy maintains PDF-action consistency.

to unstable training. Applying a Jacobian transformation to a Gaussian policy, as in the Logit-normal distribution [18], [19], [20], [21], uses a squashing function to map unbounded actions into a bounded range with controllable boundary scaling. Although the base distribution remains Gaussian and shares the same parameters, Jacobian-transformed policies typically differ noticeably in overall shape, mode-location alignment, and unimodality. Compared to these alternatives, the truncated Gaussian [22] largely preserves the shape characteristics of the Gaussian distribution.

III. BOUNDARY ACTION BIAS IN GAUSSIAN POLICY

We define boundary action bias as an underlying tendency to sample near-boundary actions during policy learning. While various factors, such as task properties, contribute to this bias, we focus on the role of the policy distribution.

In PPO, the combination of a Gaussian policy and action clipping contributes to the occurrence of this bias. As a visual example, refer to the leftmost graph in Figure 1, where the red line represents the Gaussian PDF when the location μ is at the center of the action boundaries l and u (dashed lines). The hatched bars represent the actual action probabilities. The difference between the Gaussian PDF and the action probabilities arises because the unclipped samples $\bar{a} \sim \mathcal{N}((u+l)/2, 0.5)$ outside the boundaries are clipped to $a = \text{clip}(\bar{a}, l, u)$. As a result, the Gaussian overly imposes probability mass on boundary actions, which were supposed to have the lowest probability. This also violates the unimodality of the Gaussian and counteracts the strategy of using relatively large initial scale parameters to promote exploration. The third graph in Figure 1 shows the case where the Gaussian location parameter μ is moved to the boundary u. Since the probability of out-of-bound samples being clipped becomes much higher, the exaggeration of boundary actions is further intensified, strengthening the bias.

The parameter μ is even allowed to go outside the bounds, for example, $\mu = 3$ when $u = 1$ (see the left graph in Figure 2). This is not just a theoretical case, since in a basic scenario such as `HalfCheetah-v4` in MuJoCo, a standard PPO model shows that the maximum $|\mu|$ of a trained policy reaches 3–5, where most actions would be out of bounds and clipped to u or l. This bias significantly limits policy expressiveness and also diminishes the role of the scale parameter σ, which is responsible for exploration, since variation in out-of-bound samples does not actually affect the final actions.

If the policy tends to sample boundary actions more easily, the proportion of boundary actions in the experience is also likely to be relatively large. If boundary actions are not significantly unsuitable, the policy risks collapsing into a bang-bang controller, as once μ moves beyond the boundary for certain states, the chances of sampling interior actions become very scarce.

IV. METHOD

In this section, we introduce the truncated Gaussian and then propose a truncated Gaussian policy, followed by a scale-adjusted truncated Gaussian policy to compensate for its potential counter-bias.

A. Preliminaries

1) Markov Decision Process: The problem of RL is commonly formulated as a Markov Decision Process (MDP), defined by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, $\mathcal{P}$ is the transition function, $\mathcal{R}$ is the reward function, and $\gamma \in [0,1)$ is the discount factor. At each time step t, the agent observes state s_t, takes action $a_t \sim \pi(s_t)$, and transitions to $s_{t+1} \sim \mathcal{P}(s_t, a_t)$ while receiving reward $r_t = \mathcal{R}(s_t, a_t)$. The objective is to find a policy π that maximizes the expected return $J_\pi = \mathbb{E}[\sum_{t=0}^{\infty} \gamma^t r_t]$.

2) Truncated Gaussian Distribution: The truncated Gaussian distribution [22] is a variant of the Gaussian distribution that has bounded support. The probability density function (PDF) of the truncated Gaussian for $l \le x \le u$ is given by

$$f(x; \mu, \sigma, l, u) = \frac{1}{\sigma} \cdot \frac{\varphi((x-\mu)/\sigma)}{\Phi((u-\mu)/\sigma) - \Phi((l-\mu)/\sigma)}, \quad (1)$$

where μ and σ are the location and scale parameters, and l and u are the lower and upper bounds. The functions φ and Φ denote the PDF and cumulative distribution function (CDF) of the standard Gaussian distribution, respectively:

$$\varphi(x) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}x^2), \quad \Phi(x) = \frac{1}{2}(1 + \text{erf}(x/\sqrt{2})), \quad (2)$$

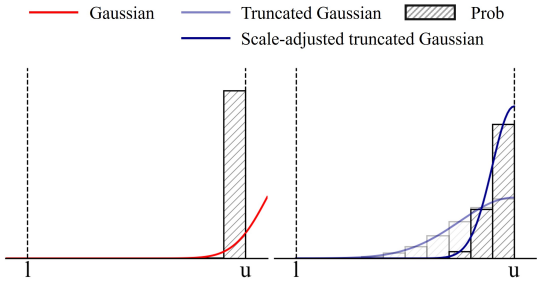

Fig. 2. Comparison between Gaussian and truncated Gaussian policies when the location is extreme. While the Gaussian policy (red line) can decisively sample boundary actions, the truncated Gaussian policy (transparent blue line) inevitably allows more within-boundary actions due to the location limit. If the scale is adjusted, the truncated Gaussian policy can become more deterministic on boundary actions (dark blue line).

where erf is the error function. The inverse CDF of the truncated Gaussian, given by

$$F^{-1}(x;\mu,\sigma,l,u) = \Phi^{-1}\left(\Phi\left((l-\mu)/\sigma\right)+ x\left(\Phi\left((u-\mu)/\sigma\right)-\Phi\left((l-\mu)/\sigma\right)\right)\right)\sigma+\mu, \quad (3)$$

can be used to generate samples from a uniform random variable $x \sim \mathscr{U}(0,1)$.

B. Truncated Gaussian Policy

We propose a truncated Gaussian policy to mitigate the boundary action bias. It maintains the key assumptions of the Gaussian while directly sampling actions from bounded support without action clipping. Specifically, the location parameter μ is bounded using a hyperbolic tangent to prevent gradients from becoming unstable. Let g be a policy network without an activation function. Then, the action of the truncated Gaussian policy is described as follows:

$$a \sim f(x;\mu = \frac{u-l}{2}\cdot(\tanh(g(s))+1)+l,\sigma,l,u). \quad (4)$$

A brief comparison between the Gaussian and truncated Gaussian policies is illustrated in Figure 1. The two graphs on the left compare them with the same parameters, where μ is located at the center of the action bounds. Unlike Gaussian policies (red line), which distort action probabilities from the original distribution, truncated Gaussian policies (blue line) show action probabilities consistent with their PDF. Since the area under a PDF should be 1, the truncated Gaussian PDF, which has finite support, is slightly taller than the Gaussian PDF, raising the probability of all interior actions. The two graphs on the right show the differences when μ is moved to the right bound.

Since the truncated Gaussian allows the policy to assign more probability mass to interior actions, it enhances the exploration of fine-grained actions. However, it may introduce a counter-bias toward overly favoring small actions. In Figure 2, the left graph and the dimmed graph on the right compare Gaussian and truncated Gaussian policies when their means μ are near the boundary. While boundary actions of a Gaussian policy are almost deterministic, a truncated Gaussian policy assigns limited probability due to the restriction of μ, which can act as a counter-bias when boundary actions are actually required.

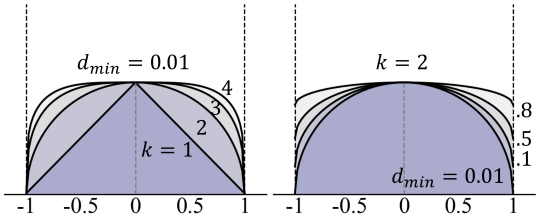

Fig. 3. Scale-adjustment functions with different hyperparameters. k adjusts the kurtosis, and d_{min} regulates the maximum discount rate. The x-axis represents the location μ, where $l=-1$ and $u=1$.

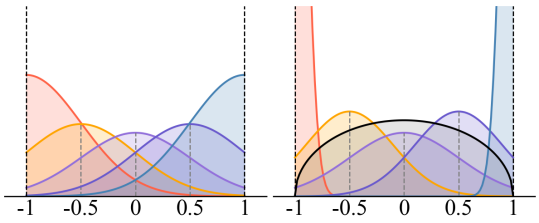

Fig. 4. Shape comparison of a plain truncated Gaussian policy (left) and a scale-adjusted policy (right), where $l=-1$ and $u=1$. The same colors indicate the same location and scale parameter. The scale-adjusted one is more decisive on boundary actions than the plain truncated Gaussian policy. The black line in the right graph is an example of scale-adjustment function.

C. Scale-Adjusted Truncated Gaussian Policy

To avoid potential overcompensation of a truncated Gaussian policy, we propose a scale-adjusted truncated Gaussian policy. In the right graph in Figure 2, the dimmed graph indicates a plain truncated Gaussian policy, while the solid graph indicates the scale-adjusted one. With the same location and scale parameters, the scale-adjusted policy exhibits more deterministic behavior for boundary actions. This is achieved by discounting the scale parameter as

$$\sigma' = \sigma \cdot d(\mu;l,u), \quad (5)$$

where σ' denotes the adjusted scale, and $d(\cdot)$ is a scale-adjustment function.

The design of $d(\cdot)$ should satisfy two conditions: (1) no discounting when μ is at the center of the action bounds, i.e., $\mu=(u+l)/2$, so $d((u+l)/2;l,u)=1$; (2) maximum discounting is applied when μ is at the boundaries, i.e., $\mu=l$ or $\mu=u$. We construct this function as a semi-ellipse centered at $(u+l)/2$ with a major axis of length $u-l$, as follows:

$$d(\mu;l,u) = \frac{\sqrt[k]{\left|\left(\frac{u-l}{2}\right)^k-\left(\mu-\frac{u+l}{2}\right)^k\right|}}{(u-l)/2}\cdot(1-d_{min})+d_{min}, \quad (6)$$

where d_{min} denotes the minimum value of d (since $\sigma'>0$), and k determines the kurtosis of the semi-ellipse. For $0<k<2$, the scale is more globally discounted, whereas for $k>2$, the discounting is more concentrated near the boundaries rather than around central locations. The function $d(\cdot)$ under different values of k and d_{min} is shown in Figure 3.

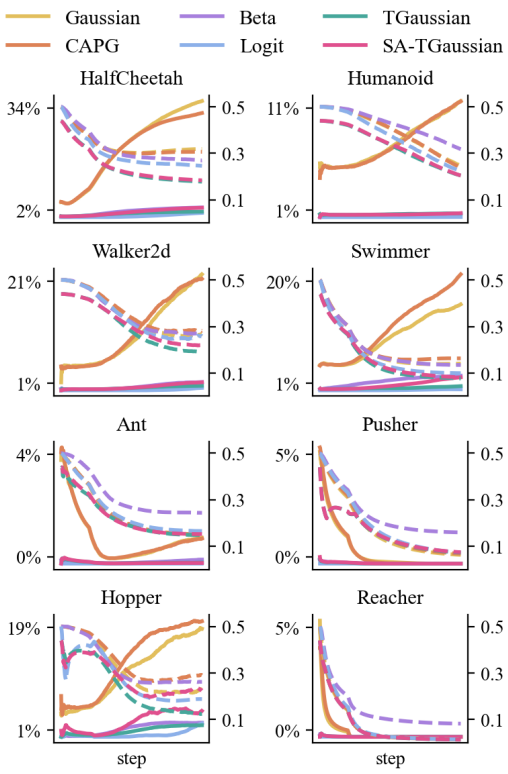

Fig. 5. Average Boundary Action Rates (aBAR) (solid lines, left axis) and average standard deviations (dashed lines, right axis) for each method across tasks.

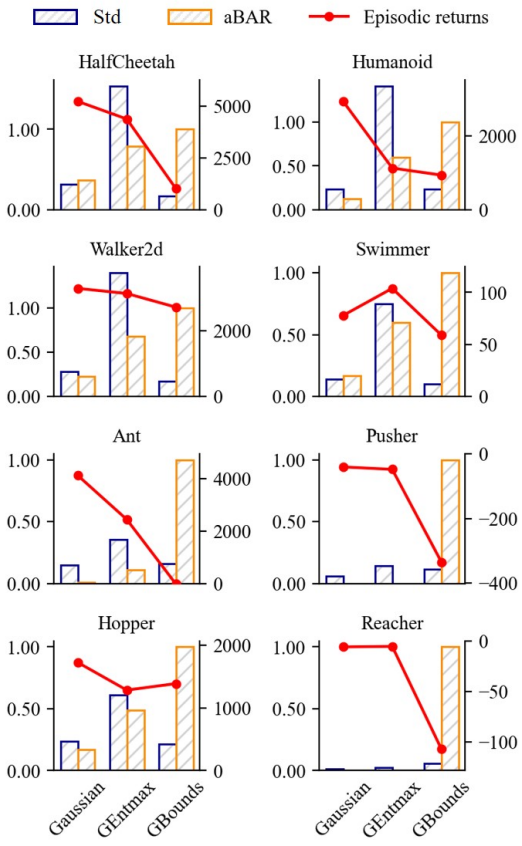

Fig. 6. Relationship between aBAR, the average standard deviation of policy distribution, and episodic return. aBAR (decimal) and standard deviation share the left axis, and episodic return uses the right axis.

Figure 4 compares the standard truncated Gaussian with the scale-discounted version, using five different locations and the same scale parameter, where $l = -1$ and $u = 1$. At the center, where $\mu = 0$, there is no difference in scale between the two distributions. However, the adjusted scale becomes narrower as the location approaches the boundaries.

V. EXPERIMENTS

In this section, we conduct experiments to answer the following questions: (1) How does each method influence boundary action usage? (2) How do boundary actions affect tasks differently? (3) Are the proposed methods competitive against other approaches? (4) Do the methods perform well in high-dimensional action spaces? (5) How do scale-adjustment hyperparameters impact performance?

A. Experimental Setup

1) Tasks: We conduct experiments on eight continuous control tasks from MuJoCo [23], including six locomotion tasks (`HalfCheetah`, `Walker2d`, `Ant`, `Hopper`, `Humanoid`, and `Swimmer`) and two manipulation tasks (`Pusher` and `Reacher`). For high-dimensional action space experiments, we use two locomotion tasks from HumanoidBench (`h1hand-walk`, `h1hand-reach`) (61 dimensions) [24] and two from the DeepMind Control Suite (DMC) (`dog-walk`, `dog-run`) (38 dimensions) [25]. Action ranges are rescaled to $(-1, 1)$, setting the boundaries to $l = -1$ and $u = 1$ for all tasks. Each MuJoCo result is averaged over 10 seeds, while HumanoidBench and DeepMind Control Suite results are averaged over 5 seeds.

2) Methods: We compare our truncated Gaussian **(TGaussian)** and scale-adjusted truncated Gaussian **(SA-TGaussian)** policy with (1) Gaussian policy **(Gaussian)**, (2) Gaussian policy with gradient correction **(CAPG)**, (3) Beta policy **(Beta)**, and (4) Hyperbolic tangent-based sample-squashing policy **(Logit)**. All policies are trained using PPO, following CleanRL [26] for implementations. We mainly use $k = 2$, $d_{min} = 0.01$ for our SA-TGaussian.

B. Empirical Study on Boundary Action Bias

1) Measurement of Boundary Action Usage: To investigate the policy trend in the selection of boundary actions, we define the average Boundary Action Rate (aBAR) as the average rate of actions falling within the top and bottom 1% of the action range for a single episode. Formally, the near-boundary action set is defined as $\mathscr{A}' = \{a \in \mathscr{A} \mid |a| > 0.99\}$, assuming $u = 1$ and $l = -1$.

Figure 5 shows the aBAR during training for the baselines and our methods. Gaussian and CAPG, which use action clipping, exhibit relatively higher aBARs (solid lines) than other methods. Since the interval of aBAR corresponds to only 2% of the entire action range, an aBAR of approximately 10–

	Gaussian	CAPG	Beta	Logit	TGaussian	SA-TGaussian
HalfCheetah	5165 ± 173.2	5399 ± 101.3	4749 ± 247.5	5175 ± 364.3	5216 ± 187.9	**5554 ± 115.3**
Walker2d	3153 ± 437.6	2871 ± 468.3	2622 ± 297.1	3559 ± 430.9	2746 ± 388.9	**3619 ± 376.2**
Ant	4001 ± 447.8	4124 ± 419.2	3393 ± 318.5	4099 ± 314.4	**4164 ± 299.1**	3972 ± 452.3
Hopper	1705 ± 267.3	1723 ± 276.4	1448 ± 181.0	1365 ± 158.2	**1825 ± 234.3**	1488 ± 239.6
Humanoid	3059 ± 553.5	3045 ± 622.6	**4102 ± 482.2**	2434 ± 405.2	2074 ± 277.1	3599 ± 577.0
Swimmer	77.23 ± 1.49	**81.32 ± 0.59**	70.99 ± 0.64	62.85 ± 0.86	59.74 ± 0.57	72.54 ± 0.55
Pusher	**-38.20 ± 2.50**	-42.34 ± 1.68	-48.16 ± 1.81	-42.31 ± 2.40	-40.91 ± 2.93	-41.06 ± 2.92
Reacher	-6.04 ± 0.92	-5.28 ± 0.82	-5.03 ± 0.54	-5.63 ± 0.61	**-4.75 ± 0.62**	-4.88 ± 0.61
Norm	0.609	0.679	0.311	0.451	0.554	**0.747**

TABLE I

PERFORMANCE OF ALL METHODS ON MUJOCO TASKS.

30% indicates the significance of the bias. Steadily increasing aBARs also imply a deepening bias as training continues. In contrast, Beta, Logit-normal, and truncated Gaussian policies record much lower aBARs of about 0–2%, demonstrating the debiasing effects of these distributions. Meanwhile, in `Pusher` and `Reacher`, aBAR drops to almost zero even for Gaussian and CAPG. Since these tasks inherently require fine-grained actions to reach the target position, noticeable disadvantages of boundary actions would have prevented the bias.

2) Impact of Boundary Action Bias: Another observation in Figure 5 is that aBAR has a positive correlation with the average standard deviations (dotted lines) of the policy distributions. For μ outside the boundaries in Gaussian and CAPG, the scale parameter σ has less incentive to decrease than when μ is inside the boundaries, because actions would largely be clipped to boundary actions regardless of changes in σ. Therefore, if boundary actions are frequent, the standard deviation tends to decrease less. Since a policy is generally expected to fine-tune actions by reducing σ throughout training [27], this observation suggests that policies with high aBAR may lead to premature convergence.

To investigate this more deeply, we conducted an additional experiment comparing three settings: (1) Gaussian policy (Gaussian), (2) a larger entropy maximization coefficient [28] (GEntMax), and (3) enforcing only boundary actions via action rounding (GBounds). Figure 6 illustrates the results of the three settings, displaying the final standard deviation (std), aBAR (bar chart using the left y-axis), and performance for each setup (line chart using the right y-axis). Since larger entropy implies larger std, GEntMax shows both larger std and aBAR than Gaussian, while exhibiting decreased overall performance. Moreover, GBounds, which has 100% aBAR, typically shows the poorest performance, implying that boundary action bias can be detrimental to policy performance. However, particularly in the cases of `Swimmer` and `Hopper`, the observed performance trend under high aBAR is either attenuated or reversed, suggesting that boundary actions may not be significantly detrimental to certain tasks. In contrast, in `Ant`, `Pusher`, and `Reacher`, where aBAR is extremely low in Gaussian, enforcing boundary actions severely impairs performance.

C. Comparison

1) Overall Effectiveness: Table I summarizes the performance of all methods across MuJoCo tasks, providing both raw and normalized scores. Overall, SA-TGaussian achieved the best performance, followed by CAPG. Gaussian ranked third, followed by TGaussian, while Logit and Beta performed the worst.

For CAPG, correcting the policy gradient enables agents to actively select boundary actions rather than passively, contributing to performance gains over Gaussian. Meanwhile, alternative distributions with bounded support, such as TGaussian, Logit, and Beta, underperform Gaussian despite significantly reducing aBAR (Figure 5). This result suggests that overcompensation inherent in their distributional design may have limited their ability to leverage the potential benefits of boundary actions. However, by adjusting the scale to make boundary actions more deterministic, SA-TGaussian demonstrates significant performance improvements over Gaussian and TGaussian, and even surpasses CAPG. This result highlights the importance of balancing the trade-off between boundary and interior actions for effective policy learning.

Although plain TGaussian performs worse than Gaussian, it achieves the best performance among the alternative distributions, Logit and Beta. This result implies that maintaining Gaussian properties, as in TGaussian, can positively contribute to overall performance. Moreover, Logit and Beta have long tails, which may exacerbate the negative impact of missing boundary actions, as they tend to avoid boundary actions when the location is near the boundary.

2) Task-Wise Anaylsis: Although SA-TGaussian achieved the highest overall normalized score, each method exhibits task-dependent strengths. For `HalfCheetah` and `Walker2d`, where Gaussian shows high aBAR and GEntmax/GBounds suffer moderate degradation, SA-TGaussian performs best, suggesting effective mitigation of boundary action bias. In `Ant`, where GEntmax and GBounds experience severe degradation and Gaussian exhibits moderate aBAR, TGaussian performs slightly better than other alternatives but remains comparable to Gaussian. For `Hopper`, which shows minor degradation in GEntmax and recovery in GBounds, TGaussian achieves

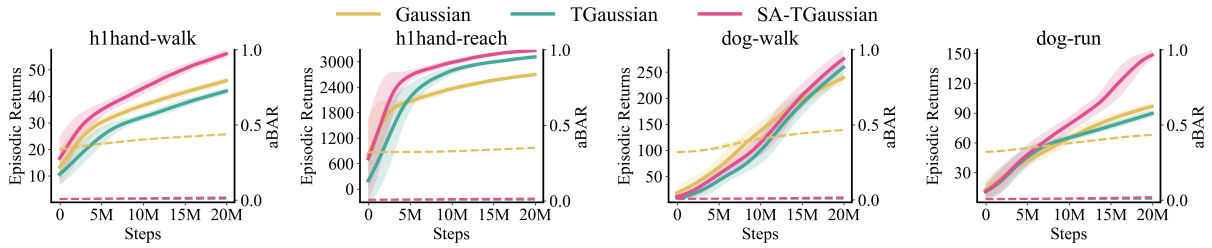

Fig. 7. Learning curves depicting episodic returns (solid lines, left axis) and aBAR (dashed lines, right axis) for each method across tasks from HumanoidBench (`h1hand-walk`, `h1hand-reach`) and DMC (`dog-walk`, `dog-run`).

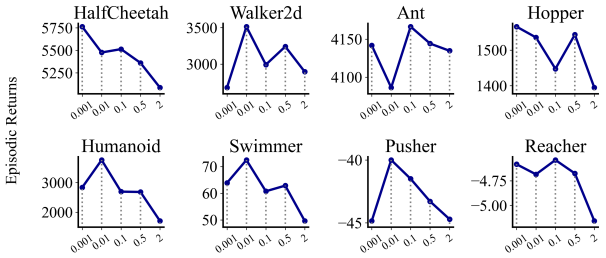

Fig. 8. Results of the ablation study on d_{min}.

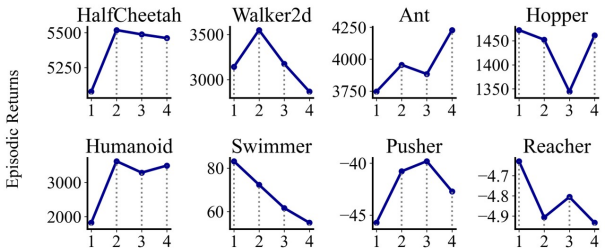

Fig. 9. Results of the ablation study on k.

the best performance. `Humanoid` follows a similar trend to `HalfCheetah` and `Walker2d`, with SA-TGaussian performing strongly, although Beta slightly outperforms all methods. In `Swimmer`, where GEntmax exhibits an inverse trend with less degradation in GBounds, Gaussian and CAPG perform best, likely reflecting the importance of boundary actions for large body movements [29]. In manipulation tasks such as `Pusher` and `Reacher`, Gaussian shows near-zero aBAR, indicating minimal boundary effects. Gaussian performs best in `Pusher`, while SA-TGaussian achieves the highest performance in `Reacher`, with TGaussian also performing competitively.

D. Ablation Study

1) High-Dimensional Action Space: To evaluate our methods in higher-dimensional action spaces, we conducted experiments on HumanoidBench and DMC. Figure 7 shows the learning curves, where solid lines denote episode returns and dashed lines indicate aBAR. Consistent with the MuJoCo results, the Gaussian policy exhibits high aBAR (near 50%), while TGaussian and SA-TGaussian maintain near-zero aBAR. Despite the need for precise actions in high-dimensional tasks, Gaussian struggles due to boundary actions. In `h1hand-walk` and `dog-run`, TGaussian underperforms Gaussian, likely due to overcompensation for boundary actions. In contrast, SA-TGaussian consistently achieves the best performance, highlighting the effectiveness of its balanced approach.

2) Scale-Adjustment Hyperparameters: Let $d_{\min}$ denote the minimum of d, where $\sigma' = \sigma \cdot d$. Smaller $d_{\min}$ makes the distribution more decisive toward boundary actions. Figure 8 shows the final episodic returns for $d_{\min} = 0.001, 0.01, 0.1, 0.5, 2.0$, revealing a decreasing trend as $d_{\min}$ increases. Overall, $d_{\min} = 0.01$ performed best, although task-specific trends varied.

Let k denote the kurtosis of the scale-discounting ellipse. Larger k reduces the strength of scale discounting away from the boundary. Figure 9 presents the final episodic returns for $k = 1, 2, 3, 4$. While $k = 2$ generally performed best, the optimal value varied across tasks, indicating greater task sensitivity than $d_{\min}$. For example, `Swimmer` showed a clear decrease in performance as k increased, likely due to reduced emphasis on beneficial boundary actions. In contrast, `Ant` exhibited the opposite trend, suggesting that less reliance on boundary actions was advantageous.

In summary, the ablation studies suggest that $d_{\min} = 0.01$ and $k = 2$ provide strong overall performance, although the optimal configuration remains task-dependent, particularly with respect to boundary action characteristics.

VI. DISCUSSION

In this study, we revisited the issue of boundary action bias in PPO, which significantly affects action exploration, and proposed a truncated Gaussian policy as a mitigation strategy. Our results demonstrate that the scale-adjusted truncated Gaussian effectively balances the two sides of the bias, showing competitive performance compared to the Gaussian policy and other approaches. Beyond proposing a method, we emphasize the importance of understanding the overall impact and roles of boundary and interior actions in RL tasks. Since continuous actions are essential for fine-grained control, establishing a more balanced exploratory condition for policy learning is important. This perspective, which has been largely overlooked due to the conventional reliance on Gaussian policies, offers valuable insights for advancing RL applications.

ACKNOWLEDGMENT

This work was supported in part by the IITP (RS-2022-I1220951-LBA/15%, RS-2022-I1220953-PICA/15%), NRF (RS-2024-00353991-SPARC/15%, RS-2023-00274280-HEI/15%, RS-2024-00358416-AutoRL/20%), KEIT (RS-2025-25453780/10%), and KIAT (RS-2025-25460896/10%) grants funded by the Korean government.

REFERENCES

[1] M. I. Ribeiro, "Gaussian probability density functions: Properties and error characterization," *Institute for Systems and Robotics, Lisboa, Portugal*, 2004.

[2] Y. Engel, S. Mannor, and R. Meir, "Reinforcement learning with gaussian processes," in *Proceedings of the 22nd international conference on Machine learning*, 2005, pp. 201–208.

[3] M. Kuss and C. Rasmussen, "Gaussian processes in reinforcement learning," *Advances in neural information processing systems*, vol. 16, 2003.

[4] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[5] P.-W. Chou, D. Maturana, and S. Scherer, "Improving stochastic policy gradients in continuous control with deep reinforcement learning using the beta distribution," in *International conference on machine learning*. PMLR, 2017, pp. 834–843.

[6] Y. Fujita and S.-i. Maeda, "Clipped action policy gradient," in *International Conference on Machine Learning*. PMLR, 2018, pp. 1597–1606.

[7] Y. Tang and S. Agrawal, "Discretizing continuous action space for on-policy optimization," in *Proceedings of the aaai conference on artificial intelligence*, vol. 34, no. 04, 2020, pp. 5981–5988.

[8] Y. Zhu, Z. Wang, Y. Zhu, C. Chen, and D. Zhao, "Discretizing continuous action space with unimodal probability distributions for on-policy reinforcement learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 36, no. 6, pp. 11 285–11 297, 2025.

[9] T. Seyde, I. Gilitschenski, W. Schwarting, B. Stellato, M. Riedmiller, M. Wulfmeier, and D. Rus, "Is bang-bang control all you need? solving continuous control with bernoulli policies," *Advances in Neural Information Processing Systems*, vol. 34, pp. 27 209–27 221, 2021.

[10] L. Xiao, S. Hong, S. Xu, H. Yang, and X. Ji, "Irs-aided energy-efficient secure wban transmission based on deep reinforcement learning," *IEEE Transactions on Communications*, vol. 70, no. 6, pp. 4162–4174, 2022.

[11] J. Markowitz, R. W. Gardner, A. Llorens, R. Arora, and I.-J. Wang, "A risk-sensitive approach to policy optimization," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 12, 2023, pp. 15 019–15 027.

[12] N. Mohamadi, S. T. A. Niaki, M. Taher, and A. Shavandi, "An application of deep reinforcement learning and vendor-managed inventory in perishable supply chain management," *Engineering Applications of Artificial Intelligence*, vol. 127, p. 107403, 2024.

[13] I. G. Petrazzini and E. A. Antonelo, "Proximal policy optimization with continuous bounded action space via the beta distribution," in *2021 IEEE symposium series on computational intelligence (SSCI)*. IEEE, 2021, pp. 1–8.

[14] J. Jerome, G. Palmer, and R. Savani, "Market making with scaled beta policies," in *Proceedings of the Third ACM International Conference on AI in Finance*, 2022, pp. 214–222.

[15] Q. Xiao, L. Jiang, M. Wang, and X. Zhang, "An improved distributed sampling ppo algorithm based on beta policy for continuous global path planning scheme," *Sensors*, vol. 23, no. 13, p. 6101, 2023.

[16] W. Chen, J. Peng, J. Chen, J. Zhou, Z. Wei, and C. Ma, "Health-considered energy management strategy for fuel cell hybrid electric vehicle based on improved soft actor critic algorithm adopted with beta policy," *Energy Conversion and Management*, vol. 292, p. 117362, 2023.

[17] G. Xu, Z. Lin, Q. Wu, J. Tan, and W. K. V. Chan, "Bi-level hierarchical model with deep reinforcement learning-based extended horizon scheduling for integrated electricity-heat systems," *Electric Power Systems Research*, vol. 229, p. 110195, 2024.

[18] J. Atchison and S. M. Shen, "Logistic-normal distributions: Some properties and uses," *Biometrika*, vol. 67, no. 2, pp. 261–272, 1980.

[19] 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.

[20] K. Ciosek and S. Whiteson, "Expected policy gradients for reinforcement learning," *Journal of Machine Learning Research*, vol. 21, no. 52, pp. 1–51, 2020.

[21] Jang, "Policy gradient in bounded continuous action space using logitnormal distribution," Ph.D. dissertation, Graduate School of Seoul National University, 2021.

[22] J. Burkardt, "The truncated normal distribution," *Department of Scientific Computing Website, Florida State University*, vol. 1, no. 35, p. 58, 2014.

[23] 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.

[24] C. Sferrazza, D.-M. Huang, X. Lin, Y. Lee, and P. Abbeel, "Humanoidbench: Simulated humanoid benchmark for whole-body locomotion and manipulation," *arXiv preprint arXiv:2403.10506*, 2024.

[25] S. Tunyasuvunakool, A. Muldal, Y. Doron, S. Liu, S. Bohez, J. Merel, T. Erez, T. Lillicrap, N. Heess, and Y. Tassa, "dm_control: Software and tasks for continuous control," *Software Impacts*, vol. 6, p. 100022, 2020.

[26] S. Huang, R. F. J. Dossa, C. Ye, J. Braga, D. Chakraborty, K. Mehta, and J. G. AraÃšjo, "Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms," *Journal of Machine Learning Research*, vol. 23, no. 274, pp. 1–18, 2022.

[27] H. Wang, T. Zariphopoulou, and X. Zhou, "Exploration versus exploitation in reinforcement learning: A stochastic control approach," *arXiv preprint arXiv:1812.01552*, 2018.

[28] Z. Ahmed, N. Le Roux, M. Norouzi, and D. Schuurmans, "Understanding the impact of entropy on policy optimization," in *International conference on machine learning*. PMLR, 2019, pp. 151–160.

[29] M. Franceschetti, C. Lacoux, R. Ohouens, A. Raffin, and O. Sigaud, "Making reinforcement learning work on swimmer," *arXiv preprint arXiv:2208.07587*, 2022.