On Proximal Policy Optimization's Heavy-tailed Gradients

Saurabh Garg [1] **Joshua Zhanson** [2] **Emilio Parisotto** [1] **Adarsh Prasad** [1] **J. Zico Kolter** [2] **Zachary C. Lipton** [1]
Sivaraman Balakrishnan [3] **Ruslan Salakhutdinov** [1] **Pradeep Ravikumar** [1]

Abstract

Modern policy gradient algorithms such as Proximal Policy Optimization (PPO) rely on an arsenal of heuristics, including loss clipping and gradient clipping, to ensure successful learning. These heuristics are reminiscent of techniques from robust statistics, commonly used for estimation in outlier-rich ("heavy-tailed") regimes. In this paper, we present a detailed empirical study to characterize the heavy-tailed nature of the gradients of the PPO surrogate reward function. We demonstrate that the gradients, especially for the *actor* network, exhibit pronounced heavy-tailedness and that it increases as the agent's policy diverges from the behavioral policy (i.e., as the agent goes further off policy). Further examination implicates the likelihood ratios and advantages in the surrogate reward as the main sources of the observed heavy-tailedness. We then highlight issues arising due to the heavy-tailed nature of the gradients. In this light, we study the effects of the standard PPO *clipping heuristics*, demonstrating that these tricks primarily serve to offset heavy-tailedness in gradients. Thus motivated, we propose incorporating GMOM, a high-dimensional robust estimator, into PPO as a substitute for three clipping tricks. Despite requiring less hyperparameter tuning, our method matches the performance of PPO (with all heuristics enabled) on a battery of MuJoCo continuous control tasks.

1. Introduction

As Deep Reinforcement Learning (DRL) methods have made strides on such diverse tasks as game playing and continuous control (Berner et al., 2019; Silver et al., 2017; Mnih et al., 2015), policy gradient methods (Williams, 1992; Sutton et al., 2000; Mnih et al., 2016) have risen as a popular alternative to dynamic programming approaches. Since Mnih et al. (2016)'s breakthrough results demonstrated the applicability of policy gradients in DRL, a number of popular variants have emerged (Schulman et al., 2017; Espeholt et al., 2018). Proximal Policy Optimization (PPO) (Schulman et al., 2017)—one of the most popular policy gradient methods—introduced the clipped importance sampling update, an effective heuristic for off-policy learning. However, while their stated motivation for clipping draws upon trust-region enforcement, the updates in practice tend to deviate from such trust regions (Ilyas et al., 2018). and exhibit sensitivity to implementation details such as random seeds and hyperparameter choices (Engstrom et al., 2019). This brittleness characterizes not just PPO, but policy gradient methods more generally (Ilyas et al., 2018; Henderson et al., 2017; 2018; Islam et al., 2017), raising a broader concern about our understanding of these methods.

In this work, we take a step towards understanding the workings of PPO, the most prominent and widely used deep policy gradient method. Noting that the heuristics implemented in PPO are evocative of estimation techniques from robust statistics in *outlier-rich* and *heavy-tailed* settings, we conjecture that the heavy-tailed distribution of gradients is the main obstacle addressed by these heuristics. We perform a rigorous empirical study to confirm the existence of heavy-tailedness in PPO gradients and to investigate its causes and consequences.

Our first contribution is to analyze the role played by each component of the PPO objective in the heavy-tailedness of the gradients. We observe that as training proceeds, gradients of both the actor and the critic loss grow more heavy-tailed. Our findings show that during *on-policy* gradient steps the advantage estimates are the primary contributors to the heavy-tailed nature of the gradients. Moreover, as *off-policyness* increases during training (i.e. as the behavioral and actor policy diverge), the likelihood ratios that appear in the surrogate objective exacerbate the heavy-tailedness.

Second, we highlight the consequences of the heavy-tailedness of PPO's gradients. Empirically, we find that heavy-tailedness in likelihood ratios induced during off-

[1]Machine Learning Department, Carnegie Mellon University [2]Computer Science Department, Carnegie Mellon University [3]Department of Statistics and Data Science, Carnegie Mellon University. Correspondence to: Saurabh Garg <sgarg2@andrew.cmu.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

policy training can be a significant factor causing optimization instability leading to low average rewards. Moreover, we also show that removing heavy-tailedness in advantage estimates can enable agents to achieve superior performance. Subsequently, we demonstrate that the clipping heuristics present in standard PPO implementations (i.e., gradient clipping, actor objective clipping, and value loss clipping) significantly counteract the heavy-tailedness induced by off-policy training.

Finally, motivated by this analysis, we present an algorithm that uses Geometric Median-of-Means (GMOM), a high-dimensional robust aggregation method adapted from the statistics literature. Without using any of the objective clipping or gradient clipping heuristics implemented in PPO, the GMOM algorithm nearly matches PPO's performance on MuJoCo (Todorov et al., 2012) tasks, which strengthens our conjecture that heavy-tailedness is a critical concern facing policy gradient methods, and that the benefits of PPO's clipping heuristics come primarily from addressing this problem.

2. Preliminaries

We define a Markov Decision Process (MDP) as a tuple $(\mathcal{S}, \mathcal{A}, R, \gamma, P)$, where $\mathcal{S}$ represents the set of environments states, $\mathcal{A}$ represents the set of agent actions, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, γ is the discount factor, and $P : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the state transition probability distribution. The goal in reinforcement learning is to learn a policy $\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}_+$ such that the expected cumulative discounted reward (known as returns) is maximized. Formally, $\pi^* := \operatorname{argmax}_\pi \mathbb{E}_{a_t \sim \pi(\cdot|s_t), s_{t+1} \sim P(\cdot|s_t, a_t)} \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right]$.

Policy gradient methods directly parameterize the policy (also known as *actor* network), i.e., they define a policy π_θ, parameterized by θ. Since directly optimizing the cumulative rewards can be challenging, modern policy gradient algorithms typically optimize a surrogate reward function which includes a likelihood ratio in order to re-use stale (off-policy) trajectories via importance sampling. For example, Schulman et al. (2015a) iteratively optimize:

$$\max_{\theta_t} \mathbb{E}_{(s_t,a_t)\sim\pi_{\theta_{t-1}}} \left[\frac{\pi_{\theta_t}(a_t|s_t)}{\pi_{\theta_{t-1}}(a_t|s_t)} A_{\pi_{\theta_{t-1}}}(s_t, a_t) \right], \quad (1)$$

where $A_{\pi_{\theta_t}} = Q_{\theta_t}(s_t, a_t) - V_{\theta_t}(s_t)$. Here, the Q-function $Q_{\theta_t}(s, a)$ is the expected discounted reward after taking an action a at state s and following π_{θ_t} afterwards and $V_{\theta_t}(s)$ is the value estimate (implemented with a *critic* network).

However, the surrogate is indicative of the true reward function only when π_{θ_t} and $\pi_{\theta_{t-1}}$ are close in distribution. Different policy gradient methods (Schulman et al., 2015a; 2017; Kakade, 2002) attempt to enforce the closeness in different ways. In Natural Policy Gradients (Kakade, 2002) and Trust Region Policy Optimization (TRPO) (Schulman et al., 2015a), authors utilize a conservative policy iteration with an explicit divergence constraint which provides provable lower bounds guarantees on the improvements of the parameterized policy. On the other hand, PPO (Schulman et al., 2017) implements a clipping heuristic on the likelihood ratio to avoid excessively large policy updates. Specifically, PPO optimizes the following objective:

$$\max_{\theta_t} \mathbb{E}_{(s_t,a_t)\sim\pi_{\theta_{t-1}}} \Big[\min \Big(\rho_t \hat{A}_{\pi_{\theta_{t-1}}}(s_t, a_t) \ , \ \operatorname{clip}(\rho_t, 1-\epsilon, 1+\epsilon) \hat{A}_{\pi_{\theta_{t-1}}}(s_t, a_t) \Big) \Big], \quad (2)$$

where $\rho_t := \frac{\pi_{\theta_t}(a_t,s_t)}{\pi_{\theta_{t-1}}(a_t,s_t)}$ and $\operatorname{clip}(x, 1-\epsilon, 1+\epsilon)$ clips x to stay between $1+\epsilon$ and $1-\epsilon$. We refer to ρ_t as *likelihood-ratios*. Due to a minimum with the unclipped surrogate reward, the PPO objective acts as a pessimistic bound on the true surrogate reward. As in standard PPO implementation, we use Generalized Advantage Estimation (GAE) (Schulman et al., 2015b). Instead of fitting the value network via regression to target values (denoted by V_{trg}), via

$$\min_{\theta_t} \mathbb{E}_{s_t\sim\pi_{\theta_{t-1}}} \left[(V_{\theta_t}(s_t) - V_{trg}(s_t))^2 \right], \quad (3)$$

standard implementations fit the value network with a PPO-like objective:

$$\min_{\theta_t} \mathbb{E}_{s_t\sim\pi_{\theta_{t-1}}} \max \Big\{ (V_{\theta_t}(s_t) - V_{trg}(s_t))^2 , (\operatorname{clip}(V_{\theta_t}(s_t), V_{\theta_{t-1}}(s_t) - \varepsilon, V_{\theta_{t-1}}(s_t) + \varepsilon) - V_{trg}(s_t))^2 \Big\}, \quad (4)$$

where ϵ is the same value used to clip probability ratios in PPO's loss function (Eq. 2). PPO uses the following training procedure: At any iteration t, the agent creates a clone of the current policy π_{θ_t} which interacts with the environment to collect rollouts $\mathcal{B}$ (i.e., state-action pairs $\{(s_i, a_i)\}_{i=1}^N$). Then the algorithm optimizes the policy π_θ and value function V_θ for a fixed K gradient steps on the sampled data $\mathcal{B}$. Since at every iteration the first gradient step is taken on the same policy from which the data was sampled, we refer to these gradient updates as *on-policy* steps. And as for the remaining $K-1$ steps, the sampling policy differs from the current agent, we refer to these updates as *off-policy* steps.

Throughout the paper, we consider a stripped-down variant of PPO (denoted PPO-NOCLIP) that consists of policy gradient with importance weighting, but has been simplified as follows: (i) no likelihood-ratio clipping (Eq. 1), i.e., no *objective function clipping* ; (ii) value network optimized via regression to target values (Eq. 3) without *value function clipping*; and (iii) no *gradient clipping*. Overall PPO-NOCLIP uses the objective summarized in App. A. One may argue that since PPO-NOCLIP removes the clipping heuristic from PPO, the unconstrained maximization of Eq. 1 may

lead to excessively large policy updates. In App. E, we empirically justify the use of Eq. 1 by showing that with the small learning rate used in our experiments (tuned hyperparameters in Table 1), PPO-NoClip maintains a KL-based trust region like PPO throughout the training.

2.1. Framework for estimating Heavy-Tailedness

We now formalize our setup for studying the distribution of gradients. Throughout the paper, we use the following definition of the heavy-tailed property:

Definition 1 (Resnick (2007)). *A non-negative random variable w is called* heavy-tailed *if its tail probability $F_w(t) := P(w \geq t)$ is asymptotically equivalent to $t^{-\alpha^*}$ as $t \to \infty$ for some positive number α^*. Here α^* (known as the tail index of w) determines the heavy-tailedness.*

For a heavy-tailed distribution with index α^*, its α-th moment exists only if $\alpha < \alpha^*$, i.e., $\mathbb{E}[w^\alpha] < \infty$ iff $\alpha < \alpha^*$. A value of $\alpha^* = 1.0$ corresponds to a Cauchy distribution and $\alpha^* = \infty$ (i.e., all moments exist) corresponds to a Gaussian distribution. Intuitively, as α^* decreases, the central peak of the distribution gets higher, the valley before the central peak gets deeper, and the tails get heavier. In other words, the lower the tail-index, the more heavy-tailed the distribution. However, in the finite sample setting, estimating the tail index is notoriously challenging (Simsekli et al., 2019; Danielsson et al., 2016; Hill, 1975).

In this study, we explore three estimators as heuristic measures to understand heavy tails and non-Gaussianity of gradients (refer to App. B for details): *(i) Alpha-index estimator* which measures alpha-index for symmeteric α-stable distributions; *(ii) Anderson-Darling test* (Anderson & Darling, 1954) on random projections of stochastic Gradient Noise (GN) to perform Gaussianity testing (Panigrahi et al., 2019). To our knowledge, the deep learning literature has only explored these two estimators for analyzing the heavy-tailed nature of gradients. Finally, in our work, we propose using *(iii) Kurtosis*. To quantify the heavy-tailedness relative to a normal distribution, we measure kurtosis (fourth standardized moment) of the gradient norms. Given samples $\{X_i\}_{i=1}^N$, the kurtosis κ is given by: $\kappa = \frac{\sum_{i=1}^N (X_i - \bar{X})^4 / N}{\left(\sum_{i=1}^N (X_i - \bar{X})^2 / N\right)^2}$ where $\bar{X}$ is the empirical mean of the samples. With a slight breach of notation, we use kurtosis to denote $\kappa^{1/4}$. In App. B, we show behavior of kurtosis on finite samples from Gaussian and Pareto distributions. It is well known that for a Pareto distribution with shape $\alpha \geq 4$, the lower the tail-index (shape parameter α) the higher the kurtosis. For $\alpha < 4$, since the fourth moment is non-existent, kurtosis is infinity. While for Gaussian distribution, the kurtosis value is approximately 1.31. In App. B, we discuss limitations of α-index estimator and Anderson-Darling test when used as heuristics to understand heavy tails. Hence, in the main paper, we include results with Kurtosis and relegate results with the other estimators.

3. Heavy-Tailedness in Policy-Gradients: A Case Study on PPO

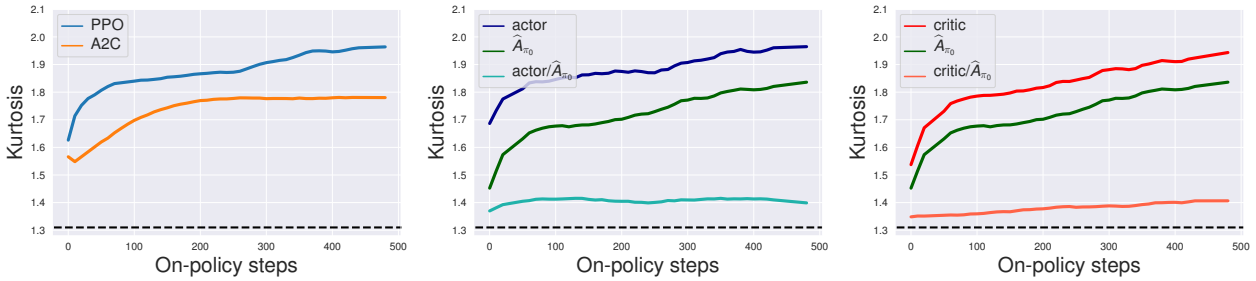
We now examine the distribution of gradients in PPO. To start, we examine the behavior of gradients at only on-policy steps. We fix the policy at the beginning of every training iteration and just consider the gradients for the first step (see App. D for details). As the training proceeds, the gradients clearly become more heavy-tailed (Fig. 1(a)). To thoroughly understand this behavior and the contributing factors, we separately analyze the contributions from different components in the loss function. We also separate out the contributions coming from actor and critic networks.

To decouple the behavior of naïve policy gradients from PPO optimizations, we consider a variant of PPO which we call PPO-NoClip as described in Section 2. Recall that in a nutshell PPO-NoClip implements policy gradient with just importance sampling. In what follows, we perform a fine-grained analysis of PPO at on-policy iterations.

3.1. Heavy-tailedness in on-policy training

Given the trend of increasing heavy-tailedness in on-policy gradients, we first separately analyze the contributions of the actor and critic networks. On both these component network gradients, we observe similar trends, with the heavy-tailedness in the actor gradients being marginally higher than the critic network (Fig. 1). Note that during on-policy steps, since the likelihood-ratios are just 1, the gradient of actor network is given by $\nabla_\theta \log\left(\pi_\theta(a_t, s_t)\right) \hat{A}_{\pi_0}(s_t, a_t)$ and the gradient of the critic network is given by $\nabla_\theta V_\theta \hat{A}_{\pi_0}(s_t, a_t)$ where π_0 is the behavioral policy. To explain the rising heavy-tailed behavior, we separately plot the advantages $\hat{A}_{\pi_0}$ and the advantage divided gradients (i.e, $\nabla \log(\pi_\theta(a_t|s_t))$ and $\nabla_\theta V_\theta$). Strikingly, we observe that while the advantage divided gradients are not heavy-tailed for both value and policy network, the heavy-tailedness in advantage estimates increases as training proceeds. This elucidates that during on-policy updates, outliers in advantage estimates are the only source of heavy-tailedness in actor and critic networks.

To understand the reasons behind the observed behaviour of advantages, we plot value estimates as computed by the critic network and the discounted returns used to calculate advantages (Fig. 9 in App. F) We don't observe any discernable heavy-tailedness trends in value estimates and a slight increase in returns. However, remarkably, we notice a very similar course of an increase in heavy-tailedness with negative advantages (whereas positive advantages remained light-tailed) as training proceeds. In App. F.3, we also pro-



(a) (b) (c)

Figure 1. **Heavy-tailedness in PPO during on-policy iterations**. All plots show mean kurtosis aggregated over 8 MuJoCo environments. For other estimators, see App. G. For individual environments with error bars, see App. I. Increases in Kurtosis implies an increase in heavy-tailedness. Dotted line represents the Kurtosis value for a Gaussian distribution. (a) Kurtosis vs on-policy iterations for A2C and PPO. Evidently, as training proceeds, the gradients become more heavy-tailed for both the methods. (b) Kurtosis vs on-policy iterations for actor networks in PPO. (c) Kurtosis vs on-policy iterations for critic networks in PPO. Both critic and actor gradients become more heavy-tailed as the agent is trained. Note that as the gradients become more heavy-tailed, we observe a corresponding increase of heavy-tailedness in the advantage estimates ($\hat{A}_{\pi_0}$). However, "actor/$\hat{A}_{\pi_0}$" and "critic/$\hat{A}_{\pi_0}$" (i.e., actor or critic gradient norm divided by advantage) remain light-tailed throughout the training. In App. F, we perform ablation tests to highlight the reason for heavy-tailed behavior of advantages.

vide evidence to this observation by showing the trends of increasing heavy-tailed behavior with the histograms of $\log(|A_{\pi_\theta}|)$ grouped by their sign as training proceeds for one MuJoCo environment (HalfCheetah-v2). This observation highlights that, at least in MuJoCo control environments, there is a positive bias of the learned value estimate for actions with negative advantages. In addition, our experiments also suggest that the outliers in advantages (primarily, in negative advantages) are the root cause of observed heavy-tailed behavior in the actor and critic gradients.

We also analyse the gradients of A2C (Mnih et al., 2016)—an on-policy RL algorithm—and observe similar trends (Fig. 1(a)), but at a relatively smaller degree of heavy-tailedness. Although they start at a similar magnitude, the heavy-tailed nature escalates at a higher rate in PPO[1]. This observation may lead us to ask: What is the cause of heightened heavy-tailedness in PPO (when compared with A2C)? Next, we demonstrate that off-policy training can exacerbate the heavy-tailed behavior.

3.2. Offpolicyness escalte heavytailness in gradients

To analyze the gradients at off-policy steps, we perform the following experiment: At various stages of training (i.e., at initialization, 50% of maximum reward, and maximum reward), we fix the actor and the critic network at each gradient step during off-policy training and analyze the collected gradients (see App. D for details). First, in the early stages of training, as the off-policyness increases, the heavy-tailedness in gradients (both actor and critic) increases. However, unlike with on-policy steps, actor gradients are the major contributing factor to the overall heavy-tailedness of the gradient distribution. In other words, the increase in heavy-tailedness of actor gradients due to off-policy training is substantially greater than for critic gradients (Fig. 2). Moreover, the increase lessens in later stages of training as the agent approaches its peak performance.

Now we turn our attention to explaining the possible causes for such a profound increase. The strong increase in heavy-tailedness of the actor gradients during off-policy training coincides with a increase of heavy-tailedness in the distribution of likelihood ratios ρ, given by $\pi_\theta(a_t, s_t)/\pi_0(a_t, s_t)$. The corresponding increase in heavy-tailedness in ratios can be explained theoretically. In continuous control RL tasks, the actor network often implements the policy with a Gaussian distribution, where the policy parameters estimate the mean and the (diagonal) covariance. With a simple example, we highlight the heavy-tailed behavior of such likelihood-ratios of Gaussian density function. This example highlights how even a minor increase in the standard deviation of the distribution of the current policy (as compared to behavior policy) can induce heavy-tails.

Example 1 (Wang et al., 2018). Assume $\pi_1(x) = \mathcal{N}\left(x; 0, \sigma_1^2\right)$ and $\pi_2(x) = \mathcal{N}\left(x; 0, \sigma_2^2\right)$. Let $\rho = \pi_1(x)/\pi_2(x)$ at a sample $x \sim \pi_2$. If $\sigma_1 \leq \sigma_2$, then likelihood ratio ρ is bounded and its distribution is not heavy-tailed. However, when $\sigma_1 > \sigma_2$, then w has a heavy-tailed distribution with the tail-index (Definition 1) $\alpha^* = \sigma_1^2/(\sigma_1^2 - \sigma_2^2)$.

During off-policy training, to understand the heavy-tailedness of actor gradients beyond the contributions from likelihood ratios, we inspect the actor gradients normalized

[1] In Appendix F.2, we show a corresponding trend in the heavy-tailedness of advantage estimates.

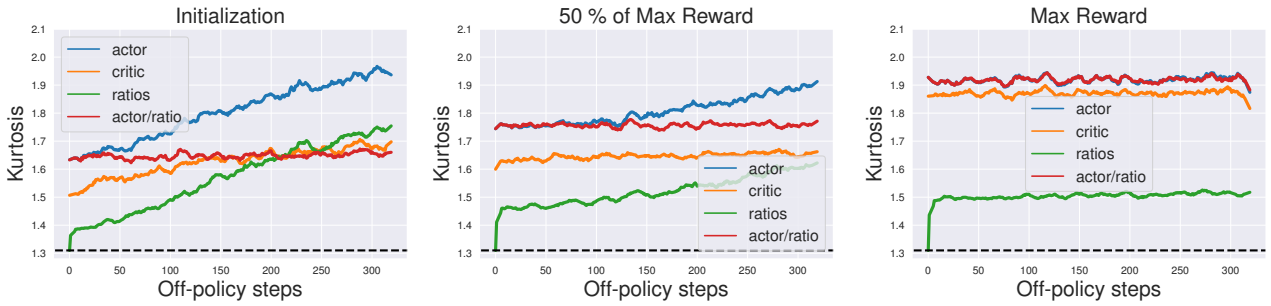


Figure 2. **Heavy-tailedness in PPO-NoClip during off-policy steps** at various stages of training iterations in MuJoCo environments. All plots show mean kurtosis aggregated over 8 Mujoco environments. Plots for other estimators can be found in App. G. We also show trends with these estimators (with error bars) on individual environments in App I. Increases in Kurtosis implies an increase in heavy-tailedness. Dotted line represents the Kurtosis value for a Gaussian distribution. Note that the analysis is done with gradients taken on a fixed batch of data within a single iteration. As off-policyness increases, the actor gradients get substantially heavy-tailed. This trend is corroborated by the increase of heavy-tailedness in ratios. Moreover, consistently we observe that the heavy-tailedness in "actor/ratios" stays constant. While initially during training, the heavy-tailedness in the ratio's increases substantially, during later stages the increase tapers off. The overall increase across training iterations is due to the induced heavy-tailedness in the advantage estimates (cf. Sec. 3.1).

by likelihood-ratios, i.e.,

$$\frac{\nabla_\theta \pi_\theta(a_t, s_t)/\pi_0(a_t, s_t)}{\pi_\theta(a_t, s_t)/\pi_0(a_t, s_t)} \hat{A}_{\pi_0}(s_t, a_t) = \nabla_\theta \log\left(\pi_\theta(a_t, s_t)\right) \hat{A}_{\pi_0}(s_t, a_t)\,.$$

Note that this gradient expression is similar to on-policy actor gradients. Since we observe an increasing trend in heavy-tailedness of the actor gradients even during on-policy training, one might ask: does these gradients' heavy-tailedness increase during off-policy gradient updates?

Recall that in PPO, we fix the value function at the beginning of off-policy training and pre-compute advantage estimates that will later be used throughout the training. Since the advantages were the primary factor dictating the increase during on-policy training, ideally, we should not observe any increase in the heavy-tailed behavior. Confirming this hypothesis, we show that the heavy-tailedness in this quantity indeed stays constant during the off-policy training (Fig. 2), i.e., $\nabla_\theta \log\left(\pi_\theta(a_t, s_t)\right) A_{\pi_0}(s_t, a_t)$ doesn't cause the increased heavy-tailed nature as long as π_0 is fixed.

Our findings from off-policy analysis strongly suggest that when the behavioral policy is held fixed, heavy-tailedness in the importance ratios is the fundamental cause. In addition, in Sec. 3.1, we showed that when importance-ratio's are 1 (i.e., the data on which the gradient step is taken is on-policy), advantages induce heavy-tailedness. With these two observations, we conclude that the scalars (either the likelihood-ratios or the advantage estimates) in the objective are the primary causes of the underlying heavy-tailedness in the gradients.

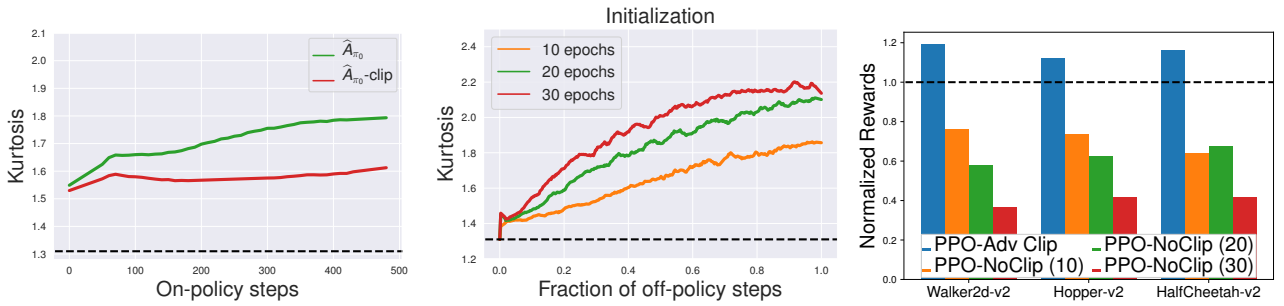
4. How do Heavy-Tailed Policy-Gradients affect Training?

In the previous section, we investigated into the root cause of the heavy-tailed behaviour. That apparent heavy-tailed nature of PPO's gradients may lead us to ask: *how do heavy-tailed gradients affect agents' performance?* In this section, we show that heavy-tailed gradients harm the performance of the underlying agent. Subsequently, we investigate into PPO heuristics and demonstrate how these heuristics alleviate for the heavy-tailed nature of the gradient distribution.

4.1. Effect of heavy-tailedness in advantages

Analysis in Sec. 3.1 shows that multiplicative advantage estimate in the PPO loss is a significant contributing factor to the observed heavy-tailedness. Motivated by this, we now study the impacts of *clipping advantages* on the underlying agent. In particular, we clip negative advantages which are the primary contributors to the induced heavy-tailedness.

Depending on the observed heavy-tailedness, we tune a per-environment clipping threshold for advantages to maximize the performance of the agent trained with PPO. Intuitively, we expect that clipping should improve optimization and hence should lead to an improved performance. Corroborating this intuition, we observe significant improvements (Fig. 3 (c)). We also plot the trend of heavy-tailedness in clipped advantage estimates during training. As we clip negative advantages below the obtained threshold, we observe that the induced heavy-tailedness stays constant throughout training (Fig. 3 (a)). Our experiment unearths an intriguing finding. Since the advantage estimates significantly contribute to the observed heavy-tailed behavior, we show that



(a) (b) (c)

Figure 3. (a) **Heavy-tailedness in PPO advantages with per-environment tuned advantage clipping threshold** and (b) **Heavy-tailedness in PPO-NoClip likelihood-ratios as the degree of off-policyness is varied** in MuJoCo environments. All plots show mean kurtosis aggregated over 8 Mujoco environments. With clipping advantages at appropriate thresholds (tuned per environment), we observe that the heavy-tailedness in advantages remains almost constant with training. For (b), we plot kurtosis vs the fraction of off-policy steps (i.e. number of steps taken normalized by the total number of gradients steps in one epoch). As the number of off-policy epochs increase, the heavy-tailedness in ratios increases substantially. (c) **Normalized rewards for PPO-AdvClip and for PPO-NoClip as the degree of off-policyness is varied** (number of off-policy steps in parenthesis). Normalized w.r.t. the max reward obtained with PPO (with all heuristics enabled) and performance of a random agent. Evidently, as off-policy training increases, the max reward achieved drops. With advantage clipping (tuned per environment), we observe improved performance of the agent. (See App J for reward curves on individual environments.)

clipping outlier advantages stabilizes the training and improves agents' performance on 5 out of 8 MuJoCo tasks (per environment rewards in App J). While tuning a clipping threshold per environment may not be practical, the primary purpose of this study is to illustrate that heavy-tailedness in advantages can actually hurt the optimization process, and clipping advantages leads to improvements in the performance of agent.

4.2. Effect of heavy-tailedness in likelihood-ratios

In Sec. 3.2, we demonstrated the heavy-tailed behavior of gradients during off-policy training which increases with off-policy gradient steps in PPO-NoClip. Moreover, we observe a corresponding increase in the heavy-tailedness of likelihood ratios. Motivated by this connection, we train agents with increased off-policy gradient steps to understand the effect of the off-policy induced heavy-tailedness on the performance of the agent. With PPO-NoClip, we train agents for 20 and 30 offline epochs (instead of 10 in Table 1)[2] and analyze its performance.

First, as expected, we observe an increase in heavy-tailedness in the likelihood ratios with escalated offline training (Fig. 3(b)). Moreover, the heavy-tailedness in advantages remains unaffected with an increase in the number of offline epochs (Fig. 20 in App. J) confirming that the observed behavior is primarily due to heightened heavy-tailedness in likelihood ratios. We conjecture that induced heavy-tailedness can make the optimization process harder. Corroborating this hypothesis, we observe that as the number of offline epochs increases, the performance of agent trained with PPO-NoClip deteriorates, and the training becomes unstable (Fig. 3 (c)). Findings from this experiment clearly highlight issues due to induced heavy-tailedness in likelihood ratios during off-policy training. While offline training enables sample efficient training, restricting the number of off-policy epochs allows effective tackling of optimization issues induced due to the heavy-tailed nature which are beyond just trust-region enforcement.

4.3. Explaining roles of various PPO objective optimizations

Motivated from our results from the previous sections, we now take a deeper look at how the core idea of likelihood-ratio clipping and auxiliary optimizations implemented in PPO and understand how they affect the heavy-tailedness during training. First, we make a key observation. Note that the PPO-clipping heuristics don't get triggered for the first gradient step taken (when a new batch of data is sampled). But rather these heuristics may alter the loss only when behavior policy is different from the policy that is being optimized. Hence, in order to understand the effects of clipping heuristics, we perform the following analysis on the off-policy gradients of the PPO-NoClip: At each update step on the agent trained with PPO-NoClip, we compute the gradients while progressively including optimizations from the standard PPO objective.

[2]Note that even with 20 and 30 offline epochs the agent maintains a KL based trust-region throughout training (Fig. 19 in App. J). Beyond 30 offline steps, successive policies often diverge—failing to maintain a KL based trust region.

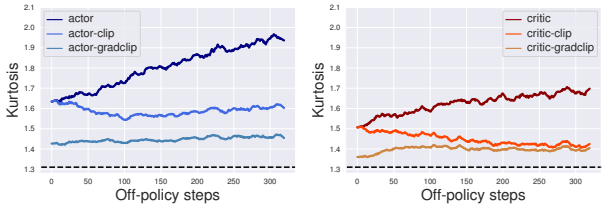


Figure 4. **Heavy-tailedness in PPO-NoClip with PPO-heuristics** applied progressively during off-policy steps, with kurtosis aggregated across 8 MuJoCo environments. For other estimators, see App. G. Dotted line represents the Kurtosis value for a Gaussian distribution. "-clip" denotes loss clipping on corresponding networks. "-gradclip" denotes both gradient clipping and loss clipping. Increases in Kurtosis implies an increase in heavy-tailedness. As training progresses during off-policy steps, the increased heavy-tailedness in actor and critic gradients is mitigated by PPO-heuristics.

Our results demonstrate that both the likelihood-ratio clipping and value-function clipping in loss during training offset the enormous heavy-tailedness induced due to off-policy training (Fig. 4). Recall that by clipping the likelihood ratios and the value function, the PPO objective is discarding samples (i.e., replacing them with zero when) used for gradient aggregation. Since heavy-tailedness in the distribution of likelihood ratios is the central contributing factor during off-policy training, by truncating likelihood-ratios ρ_t which lie outside $(1-\epsilon, 1+\epsilon)$ interval, PPO is primarily mitigating heavy-tailedness in actor gradients. Similarly, by rejecting samples from the value function loss which lie outside an ϵ boundary of a fixed *target* estimate, the heuristics alleviate the slight heavy-tailed nature induced with off-policy training in the critic network.

While PPO heuristics alleviate the heavy-tailedness induced with off-policy training, these heuristics don't alter heavy-tailed nature of advantage estimates. Since none of these heuristics directly target the outliers present in the advantage estimates, we believe that our findings can guide a development of fundamentally stable RL algorithms by targeting the outliers present in the advantage estimates (the primary cause of increasing heavy-tailedness throughout training).

5. Mitigating Heavy-Tailedness with Robust Gradient Estimation

Motivated by our analysis showing that the gradients in PPO-NoClip exhibit heavy-tailedness that increases during off-policy training, we propose an alternate method of gradient aggregation—using the gradient estimation framework from Prasad et al. (2018)—that is better suited to the heavy-tailed estimation paradigm than the sample mean. To support our hypothesis that addressing the primary benefit of PPO's various clipping heuristics lies in mitigating this heavy-tailedness, we aim to show that equipped with our robust estimator, PPO-NoClip can achieve comparable results to state-of-the-art PPO implementations, even with the clipping heuristics turned off.

Algorithm 1 Block-Gmom

input : Samples $S = \{x_1, \ldots, x_n\}$, number of blocks b, Model optimizer $\mathcal{O}_G$, b block optimizers $\mathcal{O}_B$, network f_θ, loss ℓ

Partition S into b blocks $B_1, \ldots B_b$ of equal size.
for i in $1 \ldots b$ **do**
  $\hat{\mu}_i = \mathcal{O}_B^{(i)}\left(\sum_{x_j \in B_i} \nabla_\theta \ell(f_\theta, x_j)/\left|B_i\right|\right)$
end for
$\hat{\mu}_{\text{Gmom}} = \mathcal{O}_G\left(\text{Weiszfeld}(\hat{\mu}_1, \ldots, \hat{\mu}_b)\right)$.

output : Gradient estimate $\hat{\mu}_{\text{Gmom}}$

We now consider robustifying PPO-NoClip (policy gradient with just importance sampling). Informally, for gradient distributions which do not enjoy Gaussian-like concentration, the empirical-expectation-based estimates of the gradient do not necessarily point in the right descent direction, leading to bad solutions. To this end, we leverage a robust mean aggregation technique called Geometric Median-Of-Means (Gmom) due to Minsker et al. (2015). We first split the samples into non-overlapping subsamples and estimate the sample mean of each. The Gmom estimator is then given by the geometric median-of-means of the subsamples. Formally, let $\{x_1, \ldots, x_n\} \in R$ be n i.i.d. random variables sampled from a distribution $\mathcal{D}$. Then the Gmom estimator for estimating the mean can be described as follows: Partition the n samples into b blocks $B_1, \ldots, B_b$, each of size $\lfloor n/b \rfloor$. Compute sample means in each block, i.e., $\{\hat{\mu}_1, \ldots, \mu_b\}$, where $\hat{\mu}_i = \sum_{x_j \in B_i} x_j / \left|B_i\right|$. Then the Gmom estimator $\hat{\mu}_{\text{Gmom}}$ is given by the *geometric median* of $\{\hat{\mu}_1, \ldots, \mu_b\}$ defined as follows: $\hat{\mu}_{\text{Gmom}} = \text{argmin}_\mu \sum_{i=1}^{b} \|\mu - \hat{\mu}_i\|_2$. We present Gmom algorithm along with the Weiszfeld's algorithm used for computing the approximate geometric median in App. C.

Gmom has been shown to have several favorable properties when used for statistical estimation in heavy-tailed settings. Intuitively, Gmom reduces the effect of outliers on a mean estimate by taking a intermediate mean of blocks of samples and then computing the geometric median of those block means. The robustness comes from the additional geometric median step where a small number of samples with large norms would not affect a Gmom estimate as much as they would a sample mean. Formally, given n samples from a heavy-tailed distribution, the Gmom estimate concentrates better around the true mean than the sample mean which satisfies the following:

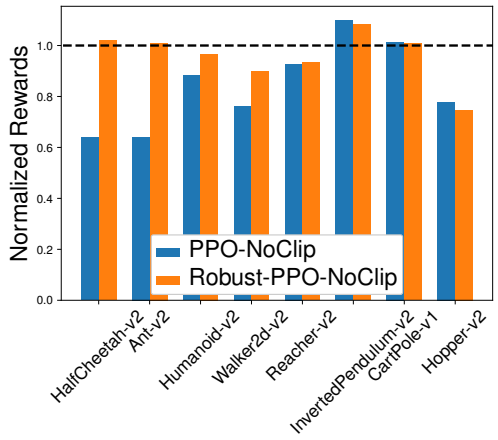


Figure 5. **Normalized rewards for ROBUST-PPO-NOCLIP and PPO-NOCLIP**. Normalized w.r.t. the max reward obtained with PPO (with all heuristics enabled) and performance of a random agent. (See App H for reward curves on individual environment.)

Theorem 1 (Minsker et al. (2015))**.** *Suppose we are given* n *samples* $\{x_i\}_{i=1}^n$ *from a distribution with mean* μ *and covariance* Σ*. Assume* $\delta > 0$*. Choose the number of blocks* $b = 1 + \lfloor 3.5 \log(1/\delta) \rfloor$*. Then, with probability at least* $1 - \delta$*,* $\|\mu_{\text{GMOM}} - \mu\|_2 \lesssim \sqrt{\frac{trace(\Sigma)\log(1/\delta)}{n}}$ *and* $\left\|\frac{1}{n}\sum_{i=1}^n x_i - \mu\right\|_2 \gtrsim \sqrt{\frac{trace(\Sigma)}{n\delta}}$.

When applying stochastic gradient descent or its variants in deep learning, one typically backpropagates the mean loss, avoiding computing per-sample gradients. However, computing GMOM requires per-sample gradients. Consequently, we propose a simple (but novel) variant of GMOM called BLOCK-GMOM which avoids the extra sample-size dependent computational penalty of calculating sample-wise gradients. Notice that by Theorem 1, the number of blocks required to compute GMOM is independent of the sample size to obtain the guarantee with high probability. To achieve this, instead of calculating sample-wise gradients, we compute block-wise gradients by backpropagating on sample-mean aggregated loss for each block. Moreover, such an implementation not only increases efficiency but also allows incorporating adaptive optimizers for individual blocks. Algorithm 1 presents the overall BLOCK-GMOM.

5.1. Results on MuJoCo environment

We perform experiments on 8 MuJoCo (Todorov et al., 2012) control tasks. To use BLOCK-GMOM aggregation with PPO-NOCLIP, we extract actor-network and critic-network gradients at each step and separately run the Algorithm 1 on both the networks. For our experiments, we use SGD as $\mathcal{O}_B$ and Adam as $\mathcal{O}_G$ and refer to this variant of PPO-NOCLIP as ROBUST-PPO-NOCLIP. We compare the performances of PPO, PPO-NOCLIP, and ROBUST-PPO-NOCLIP, using hyperparameters that are tuned individually for each method but held fixed across all tasks (Table 1).

For 7 tasks, we observe significant improvements with ROBUST-PPO-NOCLIP over PPO-NOCLIP and performance close to that achieved by PPO (with all clipping heuristics enabled) (Fig. 5). Although we do not observe improvements over PPO, we believe that this result corroborates our conjecture that PPO heuristics primarily aim to offset the heavy-tailedness induced with training.

6. Related Work

Studying the behavior of SGD, Simsekli et al. (2019) questioned the Gaussianity of SGD noise, highlighting its *heavy-tailed* nature. Subsequently, there has been a growing interest in understanding the nature of SGD noise in different deep learning tasks with a specific focus on its influence on generalization performance versus induced optimization difficulties (Şimşekli et al., 2020; Zhang et al., 2019b; Panigrahi et al., 2019). In particular, Zhang et al. (2019b) studied the nature of stochastic gradients in natural language processing (e.g., BERT-pretraining) and highlighted the effectiveness of adaptive methods (e.g. Adam and gradient clipping). Some recent work has also made progress towards understanding the effectiveness of gradient clipping in convergence (Zhang et al., 2019b;a; Şimşekli et al., 2020) in presence of heavy-tailed noise. On the other hand, Simsekli et al. (2019) highlighted *the benefits of heavy-tailed noise* in achieving wider minima with better generalization, by analyzing SGD as an SDE driven by Levy motion (whose increments are $\alpha-$stable heavy-tailed noise).

On the RL side, Bubeck et al. (2013) studied the stochastic multi-armed bandit problem when the reward distribution is heavy-tailed. The authors designed a robust version of the classical Upper Confidence Bound algorithm by replacing the empirical average of observed rewards with robust estimates obtained via the univariate median-of-means estimator (Nemirovski & Yudin, 1983) on the observed sequence of rewards. Medina & Yang (2016) extended this approach to the problem of linear bandits under heavy-tailed noise. There is also a long line of work in deep RL which focuses on reducing the variance of stochastic policy gradients (Gu et al., 2016; Wu et al., 2018; Cheng et al., 2020). On the flip side, Chung et al. (2020) highlighted the beneficial impacts of stochasticity of policy gradients on the optimization process. In simple MDPs, authors showed that larger higher moments with fixed variance leads to improved exploration. This aligns with the conjecture of Simsekli et al. (2019) in the context of supervised learning that heavy-tailedness in gradients can improve generalization. Chung et al. (2020) thus pointed out the importance of a careful analysis of stochasticity in gradients to better understand policy gradient algorithms.

We consider our work a stepping stone towards analyzing stochastic gradients beyond just their variance. We hypothesize that in deep RL where the optimization process is known to be brittle (Henderson et al., 2018; 2017; Engstrom et al., 2019; Ilyas et al., 2018), perhaps due to the flexibility of the neural representation, heavy-tailedness can cause heightened instability rather than help in efficient exploration. This perspective aligns with one line of work (Zhang et al., 2019b) where authors demonstrate that heavy-tailedness can cause instability in the learning process in deep models. Indeed with ablation experiments in Sec. 4, we show that increasing heavy-tailedness in likelihood ratios hurts the agent performance, and mitigating heavy-tailedness in advantage estimates improves the agent performance.

7. Conclusion

In this paper, we empirically characterized PPO's gradients, demonstrating that they become more heavy-tailed as training proceeds. Our detailed analysis showed that at on-policy steps, the heavy-tailed nature of the gradients is primarily attributable to the multiplicative advantage estimates. On the other hand, we observed that during off-policy training, the heavy-tailedness of the likelihood ratios of the surrogate reward function exacerbates the observed heavy-tailedness.

Subsequently, we examined issues due to heavy-tailed nature of gradients. We demonstrated that PPO's clipping heuristics primarily serve to offset the heavy-tailedness induced by off-policy training. With this motivation, we showed that a robust estimation technique could effectively replace all three of PPO's clipping heuristics: likelihood-ratio clipping, value loss clipping, and gradient clipping.

In future work, we plan to conduct similar analysis on gradients for other RL algorithms such as deep Q-learning. Moreover, we believe that our findings on heavy-tailed nature of advantage estimates can significantly impact algorithm development for policy gradient algorithms.

Acknowledgements

We acknowledge the support of Lockheed Martin, DARPA via HR00112020006, and NSF via IIS-1909816, OAC-1934584.

References

Anderson, T. W. and Darling, D. A. A test of goodness of fit. *Journal of the American statistical association*, 49 (268):765–769, 1954.

Berner, C., Brockman, G., Chan, B., Cheung, V., Dkebiak, P., Dennison, C., Farhi, D., Fischer, Q., Hashme, S., Hesse, C., et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

Bubeck, S., Cesa-Bianchi, N., and Lugosi, G. Bandits with heavy tail. *IEEE Transactions on Information Theory*, 59 (11):7711–7717, 2013.

Cheng, C.-A., Yan, X., and Boots, B. Trajectory-wise control variates for variance reduction in policy gradient methods. In *Conference on Robot Learning*, pp. 1379–1394. PMLR, 2020.

Chung, W., Thomas, V., Machado, M. C., and Roux, N. L. Beyond variance reduction: Understanding the true impact of baselines on policy optimization. *arXiv preprint arXiv:2008.13773*, 2020.

Danielsson, J., Ergun, L. M., de Haan, L., and de Vries, C. G. Tail index estimation: Quantile driven threshold selection. *Available at SSRN 2717478*, 2016.

Engstrom, L., Ilyas, A., Santurkar, S., Tsipras, D., Janoos, F., Rudolph, L., and Madry, A. Implementation matters in deep rl: A case study on ppo and trpo. In *International Conference on Learning Representations*, 2019.

Espeholt, L., Soyer, H., Munos, R., Simonyan, K., Mnih, V., Ward, T., Doron, Y., Firoiu, V., Harley, T., Dunning, I., et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. *arXiv preprint arXiv:1802.01561*, 2018.

Gu, S., Lillicrap, T., Ghahramani, Z., Turner, R. E., and Levine, S. Q-prop: Sample-efficient policy gradient with an off-policy critic. *arXiv preprint arXiv:1611.02247*, 2016.

Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., and Meger, D. Deep reinforcement learning that matters. *arXiv preprint arXiv:1709.06560*, 2017.

Henderson, P., Romoff, J., and Pineau, J. Where did my optimum go?: An empirical analysis of gradient descent optimization in policy gradient methods. *arXiv preprint arXiv:1810.02525*, 2018.

Hill, B. M. A simple general approach to inference about the tail of a distribution. *The annals of statistics*, pp. 1163–1174, 1975.

Ilyas, A., Engstrom, L., Santurkar, S., Tsipras, D., Janoos, F., Rudolph, L., and Madry, A. A closer look at deep policy gradients. *arXiv preprint arXiv:1811.02553*, 2018.

Islam, R., Henderson, P., Gomrokchi, M., and Precup, D. Reproducibility of benchmarked deep reinforcement learning tasks for continuous control. *arXiv preprint arXiv:1708.04133*, 2017.

Kakade, S. M. A natural policy gradient. In *Advances in neural information processing systems*, pp. 1531–1538, 2002.

Medina, A. M. and Yang, S. No-regret algorithms for heavy-tailed linear bandits. In *International Conference on Machine Learning*, pp. 1642–1650, 2016.

Minsker, S. et al. Geometric median and robust estimation in banach spaces. *Bernoulli*, 21(4):2308–2335, 2015.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *nature*, 518(7540): 529–533, 2015.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pp. 1928–1937, 2016.

Mohammadi, M., Mohammadpour, A., and Ogata, H. On estimating the tail index and the spectral measure of multivariate α-stable distributions. *Metrika*, 78(5):549–561, 2015.

Nemirovski, A. and Yudin, D. *Problem Complexity and Method Efficiency in Optimization*. A Wiley-Interscience publication. Wiley, 1983.

Panigrahi, A., Somani, R., Goyal, N., and Netrapalli, P. Non-gaussianity of stochastic gradient noise. *arXiv preprint arXiv:1910.09626*, 2019.

Prasad, A., Suggala, A. S., Balakrishnan, S., and Ravikumar, P. Robust estimation via robust gradient estimation. *arXiv preprint arXiv:1802.06485*, 2018.

Resnick, S. I. *Heavy-tail phenomena: probabilistic and statistical modeling*. Springer Science & Business Media, 2007.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897, 2015a.

Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015b.

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

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

Simsekli, U., Sagun, L., and Gurbuzbalaban, M. A tail-index analysis of stochastic gradient noise in deep neural networks. *arXiv preprint arXiv:1901.06053*, 2019.

Şimşekli, U., Zhu, L., Teh, Y. W., and Gürbüzbalaban, M. Fractional underdamped langevin dynamics: Retargeting sgd with momentum under heavy-tailed gradient noise. *arXiv preprint arXiv:2002.05685*, 2020.

Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pp. 1057–1063, 2000.

Todorov, E., Erez, T., and Tassa, Y. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033. IEEE, 2012.

Wang, D., Liu, H., and Liu, Q. Variational inference with tail-adaptive f-divergence. In *Advances in Neural Information Processing Systems*, pp. 5737–5747, 2018.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8(3–4):229–256, May 1992. ISSN 0885-6125. doi: 10.1007/BF00992696. URL https://doi.org/10.1007/BF00992696.

Wu, C., Rajeswaran, A., Duan, Y., Kumar, V., Bayen, A. M., Kakade, S., Mordatch, I., and Abbeel, P. Variance reduction for policy gradient with action-dependent factorized baselines. *arXiv preprint arXiv:1803.07246*, 2018.

Xie, Z., Sato, I., and Sugiyama, M. A diffusion theory for deep learning dynamics: Stochastic gradient descent exponentially favors flat minima. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=wXgk_iCiYGo.

Zhang, J., He, T., Sra, S., and Jadbabaie, A. Why gradient clipping accelerates training: A theoretical justification for adaptivity. *arXiv preprint arXiv:1905.11881*, 2019a.

Zhang, J., Karimireddy, S. P., Veit, A., Kim, S., Reddi, S. J., Kumar, S., and Sra, S. Why adam beats sgd for attention models. *arXiv preprint arXiv:1912.03194*, 2019b.