ON PROXIMAL POLICY OPTIMIZATION'S HEAVY-TAILED GRADIENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Modern policy gradient algorithms, notably 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 pronounced heavy-tailedness of the gradients, specifically for the *actor* network, which increases as the current policy diverges from the behavioral one (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. Subsequently, we study the effects of the standard PPO *clipping heuristics*, demonstrating how these tricks primarily serve to offset heavy-tailedness in gradients. Motivated by these connections, we propose incorporating GMOM, a high-dimensional robust estimator, into PPO as a substitute for three clipping tricks. Our method achieves comparable performance to that 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 emerged as a popular alternative to dynamic programming approaches. Since the breakthrough results of Mnih et al. (2016) 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 behavior of these methods tends to deviate from its key algorithmic principle (Ilyas et al., 2018), and exhibit sensitivity to implementation details (Engstrom et al., 2019). More generally, policy gradient methods are brittle, sensitive to both the random seed and hyperparameter choices, and poorly understood (Ilyas et al., 2018; Engstrom et al., 2019; Henderson et al., 2017; 2018; Islam et al., 2017). The ubiquity of these issues raises a broader concern about our understanding of policy gradient methods.

In this work, we take a step forward 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* 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 understand the causes of heavy-tailedness in PPO gradients. Furthermore, we provide a novel perspective on the clipping heuristics implemented in PPO by showing that these heuristics primarily serve to alleviate heavy-tailedness in gradients.

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 the training proceeds, gradients of both the actor and the critic loss get 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 (i.e. as the behavioral policy and actor policy diverge) dur-

ing training, the likelihood-ratios that appear in the surrogate objective exacerbates the heavy-tailed behavior. 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 and gradient clipping heuristics implemented in PPO, the GMOM algorithm nearly matches PPO's performance on MuJoCo (Todorov et al., 2012) continuous control tasks.

2 PRELIMINARIES

We define a Markov Decision Process (MDP) as a tuple $(\mathcal{S}, \mathcal{A}, R, \gamma, P)$, where $\mathcal{S}$ represent the set of environments states, $\mathcal{A}$ represent 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 $\pi_\theta : \mathcal{S} \times \mathcal{A} \to [0, 1]$, parameterized by θ, such that the expected cumulative discounted reward (known as returns) is maximized. Formally, $\pi^* := \text{argmax}_\pi \, \mathbb{E}_{a_t \sim \pi(\cdot|s_t), s_{t+1} \sim P(\cdot|s_t, a_t)} \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right]$.

Policy gradient methods directly parameterize the policy (also known as *actor* network). Since directly optimizing the cumulative rewards can be challenging, modern policy gradient algorithms typically optimize a surrogate reward function. Often the surrogate objective includes a likelihood ratio to allow importance sampling from a behavior policy π_0 while optimizing policy π_θ. For example, Schulman et al. (2015a) optimize:

$$\max_\theta \, \mathbb{E}_{(s_t, a_t) \sim \pi_0} \left[\frac{\pi_\theta(a_t, s_t)}{\pi_0(a_t, s_t)} A_{\pi_0}(s_t, a_t) \right] , \tag{1}$$

where $A_{\pi_\theta} = Q_{\pi_\theta}(s_t, a_t) - V_{\pi_\theta}(s_t)$. Here, Q-function , i.e. $Q_{\pi_\theta}(s, a)$, is the expected discounted reward after taking an action a at state s and following π_θ afterwards and $V_{\pi_\theta}(s)$ is the value estimate (implemented with a *critic* network). However, given finite samples, the surrogate is indicative of the true reward function only when π_θ and π_0 are close in distribution. Some methods (Schulman et al., 2015a; 2017; Kakade, 2002) attempt to enforce closeness via a KL divergence constraint, as in Trust Region Policy Optimization (TRPO) (Schulman et al., 2015a) or by clipping the likelihood ratio of the surrogate reward function, as in PPO (Schulman et al., 2017). Specifically, PPO optimizes the following objective:

$$\max_\theta \, \mathbb{E}_{(s_t, a_t) \sim \pi_0} \left[\min \left(\text{clip}(\rho_t, 1 - \epsilon, 1 + \epsilon) \hat{A}_{\pi_0}(s_t, a_t), \rho_t \hat{A}_{\pi_0}(s_t, a_t) \right) \right] , \tag{2}$$

where $\rho_t := \frac{\pi_\theta(a_t, s_t)}{\pi_0(a_t, s_t)}$. We refer to ρ_t as *likelihood-ratios*. As in standard PPO implementation, we use Generalized Advantage Estimation (GAE) (Schulman et al., 2015b).

Moreover, instead of fitting the value network via regression to target values:

$$L^V = (V_{\theta_t} - V_{targ})^2, \tag{3}$$

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

$$L^V = \max \left\{ (V_{\theta_t} - V_{targ})^2 , \left(\text{clip}\left(V_{\theta_t}, V_{\theta_{t-1}} - \varepsilon, V_{\theta_{t-1}} + \varepsilon\right) - V_{targ}\right)^2 \right\}, , \tag{4}$$

where ϵ is the same value used to clip probability raitos 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 (Eq. 1), but has been simplified as follows: i) no likelihood-ratio clipping, 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*.

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^* determines the heavy-tailedness and α^* is called tail index of w.*

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. This estimator is derived under the (strong) assumption that the stochastic Gradient Noise (GN) vectors are coordinate-wise independent and follow a symmetric alpha-stable distribution. *(ii) Anderson-Darling test* (Anderson & Darling, 1954) on random projections of 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. *(iii)* Finally, in our work, we propose using *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 $\kappa^{1/4}$. 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 show behavior of kurtosis on Gaussian and Pareto data with varying sample sizes and tail-indices for Pareto data.

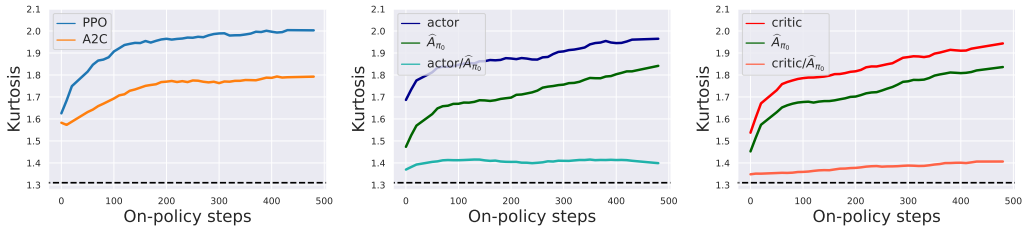
3 HEAVY-TAILEDNESS IN POLICY-GRADIENTS: A CASE STUDY ON PPO

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.

Alongside, we also perform ablations to understand how PPO heuristics affect the heavy-tailed nature of the gradient distribution. 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



(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. F. For individual environments, App. H. 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. E, we perform ablation tests to highlight the reason for heavy-tailed behavior of advantages.

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 characteristic of advantages, we plot value estimates as computed by the critic network and the discounted returns used to calculate advantages (Fig. 6 in App. E) We don't observe any discernable heavy-tailedness trends in both value estimates and 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. E.2, we also provide 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 continuous control environments, there is a positive bias of the learned value estimate for actions with negative returns. And in addition, our experiments also suggest that the outliers in advantanges (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. 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 for actor gradients due to off-policy

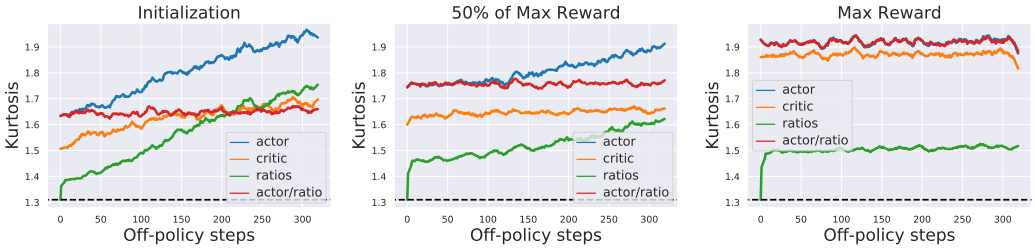


Figure 2: **Heavy-tailedness in PPO-NoClip during off-policy steps** at various stages of training iterations in MuJoCo environments. All plots show mean kurtosis aggregated over 8 Mujoco environments. Plots for other estimators can be found in App. F. We also show trends with these estimators on individual environments in App H. Increases in Kurtosis implies an increase in heavy-tailedness. Dotted line represents the Kurtosis value for a Gaussian distribution. 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 explained by the induced heavy-tailedness in the advantage estimates (cf. Sec. 3.1).

training is substantially greater than for critic gradients (Fig. 2). Furthermore, this increase lessens in later stages of training as the agent approaches its maximum 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 by likelihood-ratios, i.e.,

$$\frac{\nabla_\theta \pi_\theta(a_t, s_t)/\pi_0(a_t, s_t)}{\pi_\theta(a_t, s_t)/\pi_0(a_t, s_t)} \hat{A}_{\pi_0}(s_t, a_t) = \nabla_\theta \log\left(\pi_\theta(a_t, s_t)\right) \hat{A}_{\pi_0}(s_t, a_t)\,. \tag{5}$$

Note that the gradient expression in Eq. 5 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 ρ 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

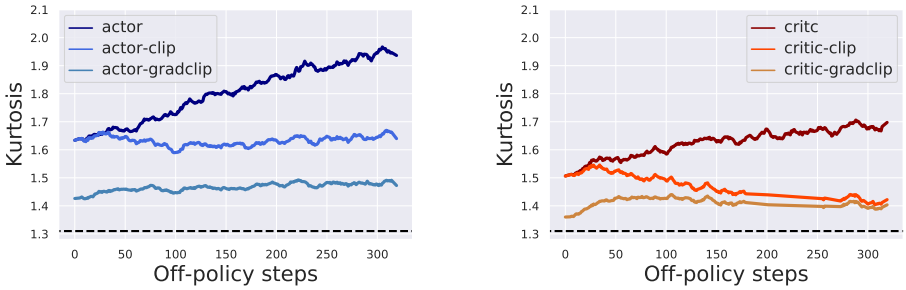


Figure 3: **Heavy-tailedness in PPO-NoClip with PPO-heuristics** applied progressively during off-policy steps, with kurtosis aggregated across 8 MuJoCo environments. For other estimators, see App. F. Dotted line represents the Kurtosis value for a Gaussian distribution. "-clip" denotes loss clipping 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.

scalars (either the likelihood-ratios or the advantage estimates) in the loss objective are the primary causes to the underlying heavy-tailedness in the gradients.

3.3 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 with progressively including optimizations from the standard PPO objective.

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. 3). 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 these PPO heuristics alleviate the heavy-tailedness induced with off-policy training, the effects of PPO clipping optimizations on mitigating heavy-tailedness induced during on-policy updates are far less clear. Since none of these heuristics directly target the outliers present in the advantage-estimates (the primary cause of increasing heavy-tailedness throughout training), we believe that our findings can guide a development of fundamentally stable RL algorithms.

4 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

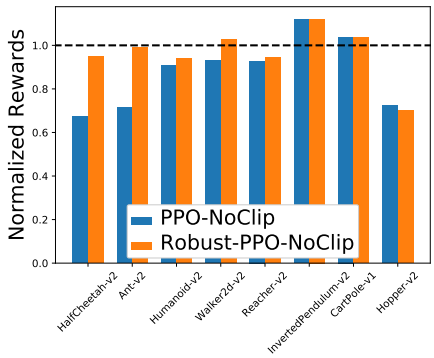


Figure 4: **Normalized rewards for ROBUST-PPO-NOCLIP and PPO-NOCLIP**. Normalized w.r.t. the max reward obtained with PPO (with all heuristics enabled) and performance of a random agent. (See App G for reward curves on individual enviornment.)

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

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

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

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.

We now consider robustifying PPO-NOCLIP (policy gradient with importance sampling but without any trust-region enforcing steps or PPO clipping tricks). 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). In short, 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: $\hat{\mu}_{\text{GMOM}} = \text{argmin}_\mu \sum_{i=1}^{b} \left\|\mu - \hat{\mu}_i\right\|_2$. Algorithm 2 in Appendix C presents the algorithm formally along with the Weiszfeld's algorithm used for computing the approximate geometric median.

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. More 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:

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 $1 - \delta$,*

$$\left\|\mu_{\text{GMOM}} - \mu\right\|_2 \lesssim \sqrt{\frac{\textit{trace}(\Sigma)\log(1/\delta)}{n}} \quad \textit{and} \quad \left\|\frac{1}{n}\sum_{i=1}^{n} x_i - \mu\right\|_2 \gtrsim \sqrt{\frac{\textit{trace}(\Sigma)}{n\delta}}\,.$$

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. To this end, 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 in Theorem 1, the number of blocks required to obtain the guarantee with high probability is independent of the sample size, i.e., we just need a constant (dependent

on δ) number of blocks to compute GMOM. 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.

4.1 RESULTS ON MUJOCO ENVIRONMENT

We perform experiments on 8 MuJoCo (Todorov et al., 2012) continuous 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. 4). 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.

5 RELATED WORK

On the RL side, a large-scale study of PPO was presented in (Ilyas et al., 2018), examining the extent that the PPO gradients align to the true underlying gradient. Likewise, Engstrom et al. (2019) provide a thorough study of the code-level heuristics used in trust region methods and indicate the necessity of such heuristics in obtaining strong model performance.

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. More recently, other works (Şimşekli et al., 2020; Zhang et al., 2019b) explore the heavy-tailed behavior of SGD noise, with emphasis on tasks that require adaptive methods. In particular, Zhang et al. (2019b) studied the nature of gradients in natural language processing (e.g., BERT-pretraining). Later work (Panigrahi et al., 2019) studied results through Gaussianity tests on random projections. They showed that at least early on in training, gradients remain near Gaussian for larger batch sizes. 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). However, its consequences modulo the bias- vs variance trade-off are not completely understood.

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.

6 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 PPO's clipping heuristics, showing that they serve primarily to offset the heavy-tailedness induced by off-policy training. Thus motivated, 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. We believe that progress in this direction can significantly impact algorithm development for reinforcement learning.

REFERENCES

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

Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemyslaw Dkebiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

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

Jon Danielsson, Lerby Murat Ergun, Laurens de Haan, and Casper G de Vries. Tail index estimation: Quantile driven threshold selection. *Available at SSRN 2717478*, 2016.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. Implementation matters in deep rl: A case study on ppo and trpo. In *International Conference on Learning Representations*, 2019.

Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Volodymir Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, et al. Impala: Scalable distributed deep-rl with importance weighted actor-learner architectures. *arXiv preprint arXiv:1802.01561*, 2018.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. *arXiv preprint arXiv:1709.06560*, 2017.

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

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

Andrew Ilyas, Logan Engstrom, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. A closer look at deep policy gradients. *arXiv preprint arXiv:1811.02553*, 2018.

Riashat Islam, Peter Henderson, Maziar Gomrokchi, and Doina Precup. Reproducibility of benchmarked deep reinforcement learning tasks for continuous control. *arXiv preprint arXiv:1708.04133*, 2017.

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

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

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

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pp. 1928–1937, 2016.

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

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

Abhishek Panigrahi, Raghav Somani, Navin Goyal, and Praneeth Netrapalli. Non-gaussianity of stochastic gradient noise. *arXiv preprint arXiv:1910.09626*, 2019.

Adarsh Prasad, Arun Sai Suggala, Sivaraman Balakrishnan, and Pradeep Ravikumar. Robust estimation via robust gradient estimation. *arXiv preprint arXiv:1802.06485*, 2018.

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

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897, 2015a.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015b.

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

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

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

Umut Şimşekli, Lingjiong Zhu, Yee Whye Teh, and Mert Gürbüzbalaban. Fractional underdamped langevin dynamics: Retargeting sgd with momentum under heavy-tailed gradient noise. *arXiv preprint arXiv:2002.05685*, 2020.

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

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

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

Ronald J. Williams. 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`.

Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. Why gradient clipping accelerates training: A theoretical justification for adaptivity. *arXiv preprint arXiv:1905.11881*, 2019a.

Jingzhao Zhang, Sai Praneeth Karimireddy, Andreas Veit, Seungyeon Kim, Sashank J Reddi, Sanjiv Kumar, and Suvrit Sra. Why adam beats sgd for attention models. *arXiv preprint arXiv:1912.03194*, 2019b.

A DETAILED BACKGROUD

We define a Markov Decision Process (MDP) as a tuple $(\mathcal{S}, \mathcal{A}, R, \gamma, P)$, where $\mathcal{S}$ represent the set of environments states, $\mathcal{A}$ represent 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 $\pi_\theta : \mathcal{S} \times \mathcal{A} \to [0, 1]$, parameterized by θ, such that the expected cumulative discounted reward (known as returns) is maximized. Formally,

$$\pi^* = \underset{\pi}{\operatorname{argmax}}\, \mathbb{E}_{a_t \sim \pi(\cdot|s_t), s_{t+1} \sim P(\cdot|s_t, a_t)} \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \right] . \tag{6}$$

Policy gradient methods directly optimize a paraterized policy function (also known as *actor network*). The central idea behind policy gradient methods is to perform stochastic gradient ascent on expected return (Eq. 6) to learn parameters θ. Under mild conditions (Sutton et al., 2000), the gradient of the Eq. 6 can be written as

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \nabla_\theta \log(\pi_\theta(a_t|s_t)) \right] ,$$

where $\tau \sim \pi_\theta$ are trajectories sampled according to $\pi_\theta(\tau)$ and $J(\theta)$ is the objective maximised in Eq. 6. With the observation that action a_t only affects the reward from time t onwards, we re-write the objective $J(\theta)$, replacing returns using the Q-function, i.e., the expected discounted reward after taking an action a at state s and following π_θ afterwards. Mathematically, $Q_{\pi_\theta}(s, a) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{k=0}^{\infty} \gamma^k R(s_{t+k}, a_{t+k}) | a_t = a, s_t = s \right]$. Using the Q-function, we can write the gradient of the objective function as

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{\infty} Q_{\pi_\theta}(s_t, a_t) \nabla_\theta \log(\pi_\theta(a_t|s_t)) \right] .$$

However, the variance in the above expectation can be large, which raises difficulties for estimating the expectation empirically. To reduce the variance of this estimate, a baseline is subtracted from the Q-function—often the value function or expected cumulative discounted reward starting at a certain state and following a given policy i.e., $V_{\pi_\theta}(s) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{k=0}^{\infty} \gamma^k R(s_{t+k}, a_{t+k}) | s_t = s \right]$. The network that estimates the value function is often referred to as *critic*. Define $A_{\pi_\theta}(s_t, a_t) = Q_{\pi_\theta}(s_t, a_t) - V_{\pi_\theta}(s_t)$ as the *advantage* of performing action a_t at state s_t. Incorporating an advantage function, the gradient of the objective function can be written:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{\infty} A_{\pi_\theta}(s_t, a_t) \nabla_\theta \log(\pi_\theta(a_t|s_t)) \right] . \tag{7}$$

Eq. 7 is the näive actor-critic objective and is used by A2C.

Trust region methods and PPO. Since directly optimizing the cumulative rewards can be challenging, modern policy gradient optimization algorithms often optimize a surrogate reward function in place of the true reward. Most commonly, the surrogate reward objective includes a likelihood ratio to allow importance sampling from a behavior policy π_0 while optimizing policy π_θ, such as the surrogate reward used by Schulman et al. (2015a):

$$\max_\theta \mathbb{E}_{(s_t, a_t) \sim \pi_0} \left[\frac{\pi_\theta(a_t, s_t)}{\pi_0(a_t, s_t)} \hat{A}_{\pi_0}(s_t, a_t) \right] , \tag{8}$$

where $\hat{A}_\pi = \frac{A_\pi - \mu(A_\pi)}{\sigma(A_\pi)}$ (we refer to this as the *normalized advantages*). However, in finite sample setting, the surrogate is indicative of the true reward function only when π_θ and π_0 are distributionally close. Resulting methods (Schulman et al., 2015a; 2017; Kakade, 2002) aim to enforce such a closeness by applying a KL divergence constraint, as in Trust Region Policy Optimization (TRPO) (Schulman et al., 2015a) or by clipping the likelihood ratio of the surrogate reward function, as in PPO Schulman et al. (2017). Specifically, PPO optimizes the following objective:

$$\max_\theta \mathbb{E}_{(s_t, a_t) \sim \pi_0} \left[\min \left(\text{clip}(\rho_t, 1 - \epsilon, 1 + \epsilon) A_{\pi_0}(s_t, a_t), \rho_t A_{\pi_0}(s_t, a_t) \right) \right] , \tag{9}$$

where $\rho_t := \frac{\pi_\theta(a_t, s_t)}{\pi_0(a_t, s_t)}$. We refer to ρ_t as *likelihood-ratios*. Moreover, instead of fitting the value network via regression to target values:

$$L^V = (V_{\theta_t} - V_{targ})^2, \tag{10}$$

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

$$L^V = \max\left[(V_{\theta_t} - V_{targ})^2, \left(\text{clip}\left(V_{\theta_t}, V_{\theta_{t-1}} - \varepsilon, V_{\theta_{t-1}} + \varepsilon\right) - V_{targ}\right)^2\right], \tag{11}$$

where ϵ is the same value used to clip probability raitos in PPO's loss function (Eq. 9). 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 collects rollouts $\mathcal{S}$ (i.e., state-action pair $\{(s_i, a_i)\}_{i=1}^N$). Then the algorithm optimizes the policy π_θ and value function for a fixed K gradient steps on the sampled data $\mathcal{S}$. 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 (Eq. 8), but has been simplified as follows: i) no likelihood-ratio clipping, i.e., no *objective function clipping*; ii) value network optimized via regression to target values (Eq. 10) without *value function clipping*; and iii) no *gradient clipping*.

B Details on estimators

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 2 (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^* determines the heavy-tailedness and α^* is called tail index of w.*

For a heavy-tailed distribution with index α^*, its α-th moment exist 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).

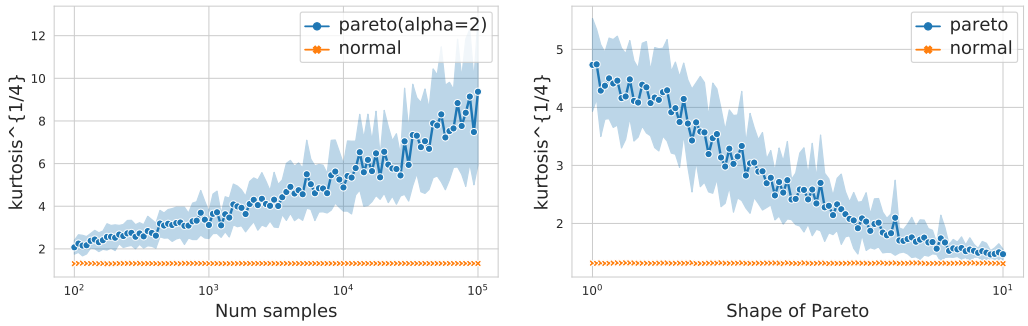
In this study, we explore three estimators as heuristic measures to understand heavy tails and non-Gaussianity of gradients.

- *Alpha-index estimator*. This estimator was proposed in Mohammadi et al. (2015) for symmeteric α-stable distributions and was used by Simsekli et al. (2019) to understand the noise behavior of SGD. This estimator is derived under the (strong) assumption that the stochastic Gradient Noise (GN) vectors are coordinate-wise independent and follow a symmetric alpha-stable distribution. Formally, let $\{X_i\}_{i=1}^N$ be a collection of $N = mn$ (centered) random variables. Define $Y_i = \sum_{j=1}^m X_{j+(i-1)m}$ for $i \in [n]$. Then, the estimator is given by

  $$\frac{1}{\alpha} := \frac{1}{\log m}\left(\frac{1}{n}\sum_{i=1}^n \log|Y_i| - \frac{1}{n}\sum_{i=1} N \log|X_i|\right).$$

  Instead of treating each co-ordinate of gradient noise as an independent scalar, we use these estimators on gradient norms. With alpha-index estimator, smaller alpha-index value signify higher degree of heavy-tailedness.

- *Anderson-Darling test* (Anderson & Darling, 1954) on random projections of GN to perform Gaussianity testing. Panigrahi et al. (2019) proposed the Gaussianity test on the projections of GN along 1000 random directions. Their estimate is then the fraction of directions accepted by the Anderson Darling test. While this estimator is informative about the



(a) (b)

Figure 5: Kurtosis plots. Analysis on norms of 100-dimensional vectors such that each coordinate is sampled iid from Pareto distribution or normal distribution. (a) Variation in kurtosis ($\kappa^{1/4}$) as the sample size is varied for samples from normal distribution and Pareto with tail index 2 (i.e, $\alpha = 2$). (b) Variation in kurtosis ($\kappa^{1/4}$) as the shape of Pareto is varied at fix sample size.

Gaussian behavior, it is not useful to quantify and understand the trends of heavy-tailedness if the predictor nature is non-Gaussian.

- To our knowledge, the deep learning literature has only exploredthese two estimators for analyzing the heavy-tailed nature of gradients. *(iii)* Finally, in our work, we propose using kurtosis *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.

B.1 Synthetic Study

In Figure 5, we show the trends with varying tail index and sample sizes. Clearly as the tail-index increases, i.e., the shape parameter increases, the kurtosis decreases (signifying its correlation to capture tail-index). Although for tail-index smaller than 4 the kurtosis is not defined, we plot empirical kurtosis and show its increasing trend sample size. We fix the tail index of Pareto at 2 and plot finite sample kurtosis and observe that it increases almost exponentially with the sample size. These two observations together hint that kurtosis is a neat surrogate measure for heavy-tailedness.

C GMOM ALGORITHM

Algorithm 2 GMOM

input : Samples $S = \{x_1, \ldots, x_n\}$, number of blocks b

1: $m = \lfloor n/b \rfloor$.
2: **for** i in $1 \ldots b$ **do**
3: $\hat{\mu}_i = \sum_{j=0}^{m} x_{j+i*m}$/ m.
4: **end for**
5: $\hat{\mu}_{\text{GMOM}} = \text{WEISZFELD}(\hat{\mu}_1, \ldots, \hat{\mu}_b)$.

output : Estimate $\hat{\mu}_{\text{GMOM}}$

Algorithm 3 WEISZFELD

input : Samples $S = \{\mu_1, \ldots, \mu_b\}$, number of blocks b

1: Initialize μ arbitrarily.
2: **for** iteration $\leftarrow 1, \ldots, n$ **do**
3: $d_j := \frac{1}{\|\mu - \mu_j\|_2}$ for j in $1, \ldots, b$.
4: $\mu := \left(\sum_{j=1}^{b} \mu_j d_j\right) / \left(\sum_{j=1}^{b} d_j\right)$
5: **end for**

output : Estimate μ

D EXPERIMENTAL SETUP FOR GRADIENT DISTRIBUTION STUDY

Recall that 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 collects rollouts $\mathcal{S}$ (i.e., state-action pair $\{(s_i, a_i)\}_{i=1}^{N}$). Then the algorithm optimizes the policy π_θ and value function for a fixed K gradient steps on the sampled data $\mathcal{S}$. 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. For all experiments, we aggregate our estimators across 10 seeds and 8 environments. We do this by first computing the estimators for individual experiments and then taking the sample mean across all runs. We now describe the exact experimental details.

In all of our experiments, for each gradient update, we have a batch size of 64. Hence for an individual estimate, we aggregate over 64 samples (batch size in experiments) to compute our estimators. For Anderson Darling test, we use 100 random directions to understand the behavior of stochastic gradient noise.

On-policy heavy-tailed estimation. At every on-policy gradient step, we freeze the policy and value network, and save the sample-wise gradients of the actor and critic objective. The estimators are calculated at every tenth on-policy update throughout the training.

Off-policy heavy-tailed estimation At every off-policy gradient step, we freeze the policy and value network, and save the sample-wise gradients of the actor and critic objective. Then at various stages of training (i.e. initialization, 50% max reward and max reward), we fix the collected trajectories and collect sample-wise gradients for the 320 steps taken.

Analysis of PPO-NOCLIP with progressively applying PPO heuristics. We compute the gradients for the off-policy steps taken with the PPO-NOCLIP objective as explained above. Then at each gradient step, we progressively add heuristics from PPO and re-compute the gradients for analysis. Note that we still always update the value and policy network with PPO-NOCLIP objective gradients.

E TRENDS WITH ADVANTAGES

E.1 KURTOSIS FOR RETURNS, VALUE ESTIMATE AND ADVANTAGES GROUPED WITH SIGN

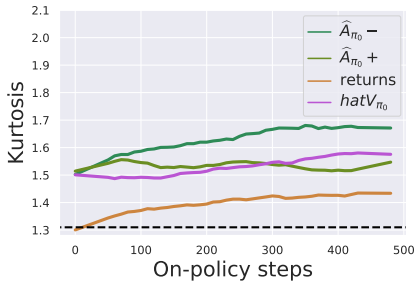


Figure 6: Heavy-tailedness in advantages grouped by their sign, rewards and value estimates. Clearly as the training progresses the negative advantages become heavy-tailed whereas heavytailedness in the returns, value estimates and positive advantages doesn't change.

E.2 HISTOGRAMS OF ADVANTAGES ON HALFCHEETAH OVER TRAINING ITERATIONS

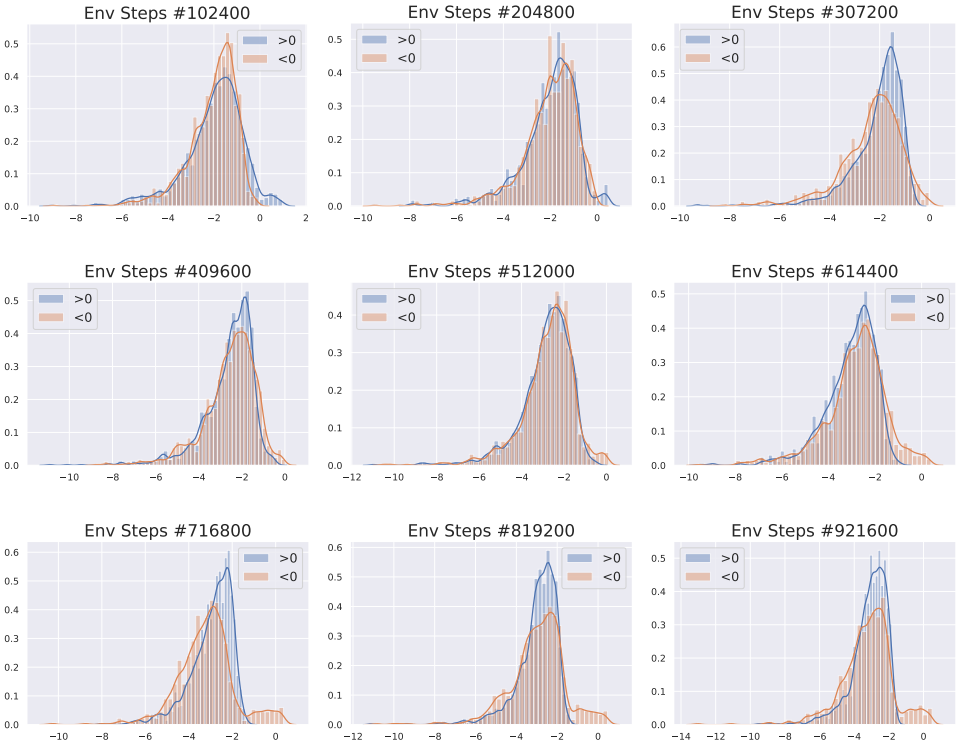
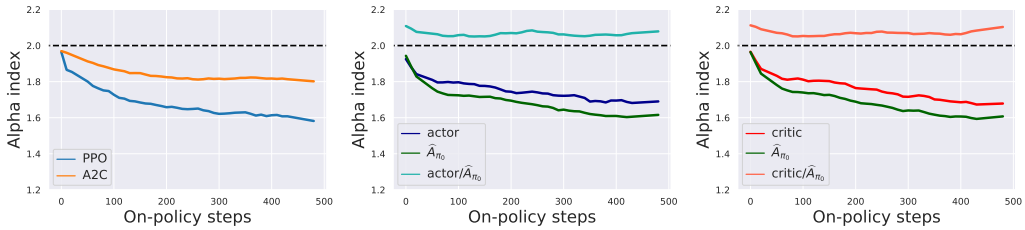


Figure 7: Distribution of $\log(|A_{\pi_\theta}|)$ over training grouped by sign of $\log(|A_{\pi_\theta}|)$ for HalfCheetah-v2 . To elaborate, we collect the advantages and separately plot the grouped advantages with their sign, i.e., we draw histograms separately for negative and positive advantages. As training proceeds, we clearly observe the increasing heavy-tailed behavior in negative advatanges as captured by the higher fraction of $\log(|A_{\pi_\theta}|)$ with large magnitude. Moreover, the histograms for positive advantages (which resembel Gaussain pdf) stay almost the same throughout training. This highlights the particular heavy-tailed (outlier-rich) nature of negative advantages corroborating our experiments with kurtosis and tail-index estimators.

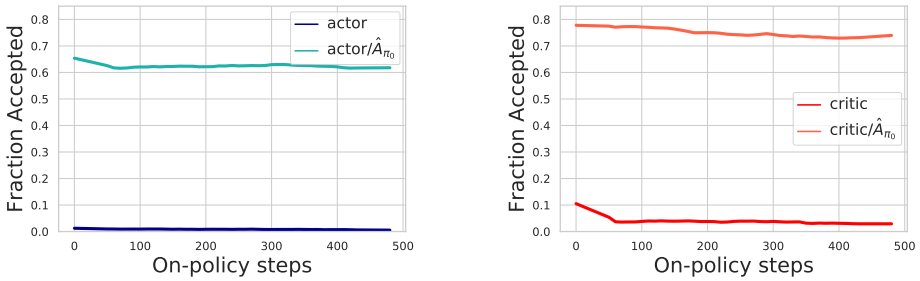
F ANALYSIS WITH OTHER ESTIMATORS

F.1 ON-POLICY GRADIENT ANALYSIS



(a) (b) (c)

Figure 8: **Heavy-tailedness in PPO during on-policy iterations**. All plots show mean alpha index aggregated over 8 MuJoCo environments. A decrease in alpha-index implies an increase in heavy-tailedness. (a) Alpha index vs on-policy iterations for A2C and PPO. Evidently, as training proceeds, the gradients become more heavy-tailed for both the methods. (b) Alpha index vs on-policy iterations for actor networks in PPO. (c) Alpha index vs on-policy iterations for critic networks in PPO. Both critic and actor gradients become more heavy-tailed on-policy steps 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 GAE estimates) remain light-tailed throughout the training.



(a) (b)

Figure 9: **Heavy-tailedness in PPO during on-policy iterations**. All plots show mean fraction of directions accepted by Anderon-Darling test over 8 MuJoCo environments. A higher accepted fraction indicates a Gaussian behavior. (b) Fraction accepted vs on-policy iterations for actor networks in PPO. (c) Fraction accepted vs on-policy iterations for critic networks in PPO. Both critic and actor gradients remain non-Gaussian as the agent is trained. However, "actor/$\hat{A}_{\pi_0}$" and "critic/$\hat{A}_{\pi_0}$" (i.e., actor or critic gradient norm divided by GAE estimates) have fairly high fraction of directions accepted, hinting their Gaussian nature.

F.2 OFF-POLICY GRADIENT ANALYSIS

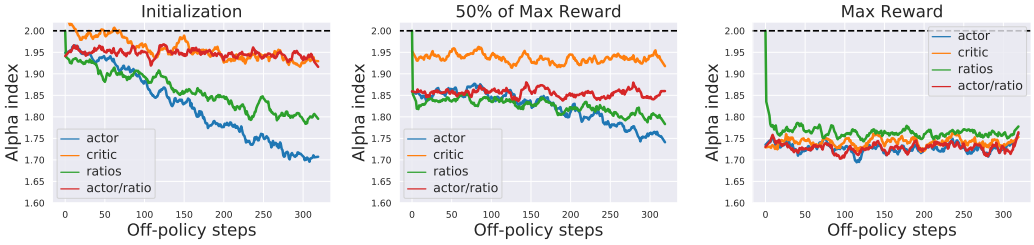


Figure 10: **Heavy-tailedness in PPO-NOCLIP during off-policy steps** at various stages of training iterations in MuJoCo environments. All plots show mean alpha index aggregated over 8 Mujoco environments. A decrease in alpha index implies an increase in heavy-tailedness. As off-policyness increases, 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 explained by the induced heavy-tailedness in the advantage estimates (cf. Sec. 3.1).

G HYPERPARAMETER SETTINGS AND REWARDS CURVES ON INDIVIDUAL ENVIORNMENTS

Hyperparameter	Values
Steps per PPO iteration	2048
Number of minibatches	32
PPO learning rate	0.0003
ROBUST-PPO-NOCLIP learning rate	0.00008
PPO-NOCLIP learning rate	0.00008
Discount factor γ	0.99
GAE parameter λ	0.95
Entropy loss coefficient	0.0
PPO value loss coefficient	2.0
ROBUST-PPO-NOCLIP value loss coefficient	2.0
PPO-NOCLIP value loss coefficient	2.0
Max global L2 gradient norm (only for PPO)	0.5
Clipping coefficient (only for PPO)	0.2
Policy epochs	10
Value epochs	10
GMOM number of blocks	8
GMOM Weiszfeld iterations	100

Table 1: Hyperparameter settings. Sweeps were run over learning rates { 0.000025, 0.00005, 0.000075, 0.00008, 0.00009 , 0.0001, 0.0003, 0.0004 } and value loss coefficient { 0.1, 0.5, 1.0, 2.0, 10.0} with 10 random seeds per learning rate.

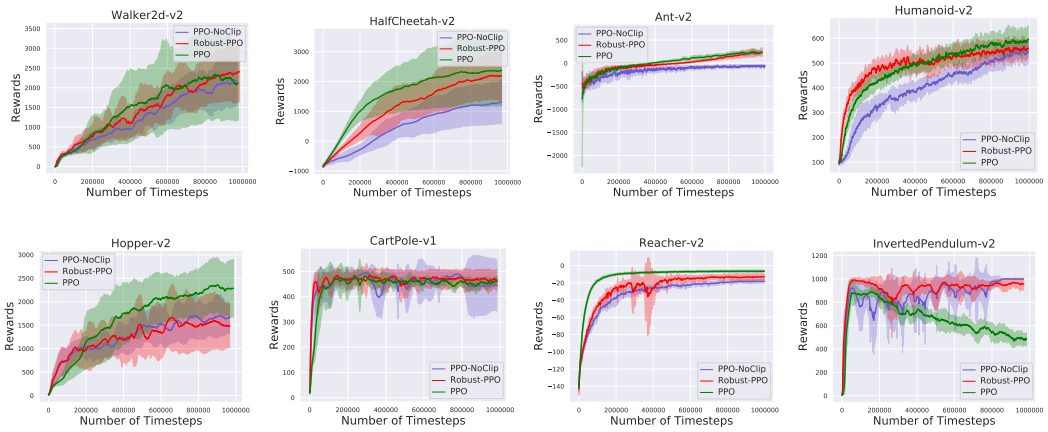


Figure 11: Reward curves as training progresses in 10 different Mujoco Environments aggregated across 10 random seeds and for hyperparameter setting tabulated in Table 1. The shaded region denotes the one standard deviation across seeds. Aggregated results shown in Fig. 4.

H ANALYSIS ON THREE OF THE INDIVIDUAL ENVIORNMENTS.

Overall, in figures below we show that the trends observed in aggregated plots in Section 3 hold true on individual enviorments.

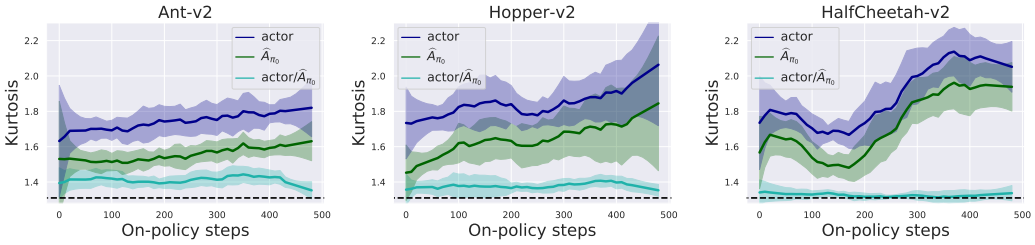


Figure 12: **Heavy-tailedness in actor gradients for PPO during on-policy steps** for three MuJoCo environments. All plots show mean and std of kurtosis aggregated over 10 random.

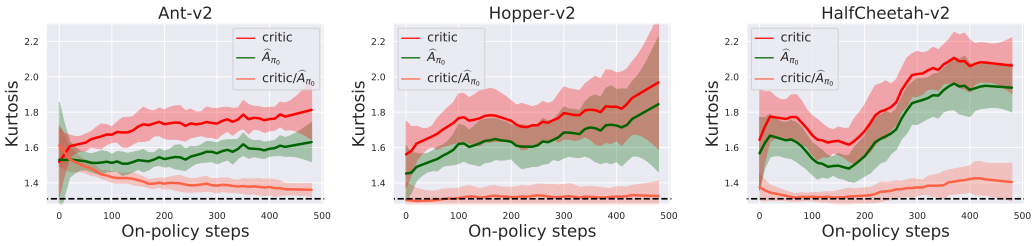


Figure 13: **Heavy-tailedness in critic gradients for PPO during on-policy steps** for three MuJoCo environments. All plots show mean and std of kurtosis aggregated over 10 random.

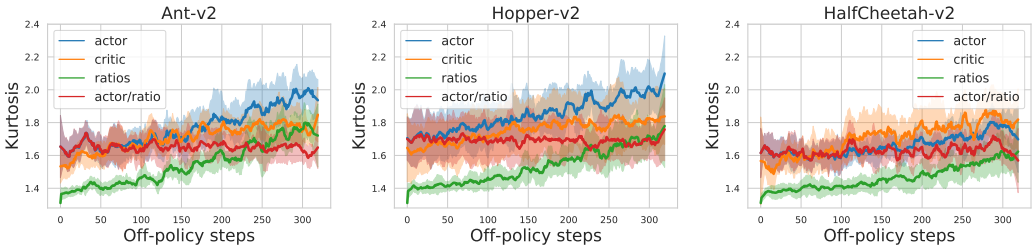


Figure 14: **Heavy-tailedness in PPO-NoClip during off-policy steps** for three MuJoCo environments. All plots show mean and std of kurtosis aggregated over 10 random.