BUTTERFLY EFFECTS OF SGD NOISE: ERROR AMPLIFICATION IN BEHAVIOR CLONING AND AUTOREGRESSION

Adam Block
Department of Mathematics
MIT
Cambridge, MA, 02139
ablock@mit.edu

Cyril Zhang, Dylan Foster & Akshay Krishnamurthy
Department of Computational Neuroscience
University of the Witwatersrand
Joburg, South Africa
{robot,net}@wits.ac.za

Max Simchowitz
Affiliation
Address
email

ABSTRACT

This work studies training instabilities of behavior cloning with deep neural networks. We observe that minibatch SGD updates to the policy network during training result in sharp oscillations in long-horizon rewards, despite negligibly affecting the behavior cloning loss. We empirically disentangle the statistical and computational causes of these oscillations, and find them to stem from the chaotic propagation of minibatch SGD noise through unstable closed-loop dynamics. While SGD noise is benign in the single-step action prediction objective, it results in catastrophic error accumulation over long horizons, an effect we term *gradient variance amplification* (GVA). We show that many standard mitigation techniques do not alleviate GVA, but find an exponential moving average (EMA) of iterates to be surprisingly effective at doing so. We illustrate the generality of this phenomenon by showing the existence of GVA and its amelioration by EMA in both continuous control and autoregressive language generation. Finally, we provide theoretical vignettes that highlight the benefits of EMA in alleviating GVA and shed light on the extent to which classical convex models can help in understanding the benefits of iterate averaging in deep learning.

1 INTRODUCTION

Deep neural networks are increasingly used in machine learning tasks that contain *feedback loops* as a defining characteristic: outputs of language models depend on previously predicted tokens (Vaswani et al., 2017), recommendation systems influence the users to whom they give suggestions (Krauth et al., 2020; Dean & Morgenstern, 2022), and robotic policies take actions in reactive control environments (Ross & Bagnell, 2010; Laskey et al., 2017). Because these tasks are so complex, it is standard practice to optimize surrogate objectives, such as next-token prediction, that typically ignore feedback loops altogether (Pomerleau, 1988; Vaswani et al., 2017; Florence et al., 2022).

When training deep models by gradient updates on the surrogate objective, surrogate performance often improves more or less monotonically as training progresses. At the same time, successive iterates can exhibit wild variations in their performance on the task of interest. Because it is often impractical to evaluate the desired performance metric at multiple checkpoints, these oscillations imply that we have high risk of selecting and deploying a poor policy. Thus, in order to determine best practices, we must first understand whether *better training* or *better data* will fix these instabilities. This leads us to ask:

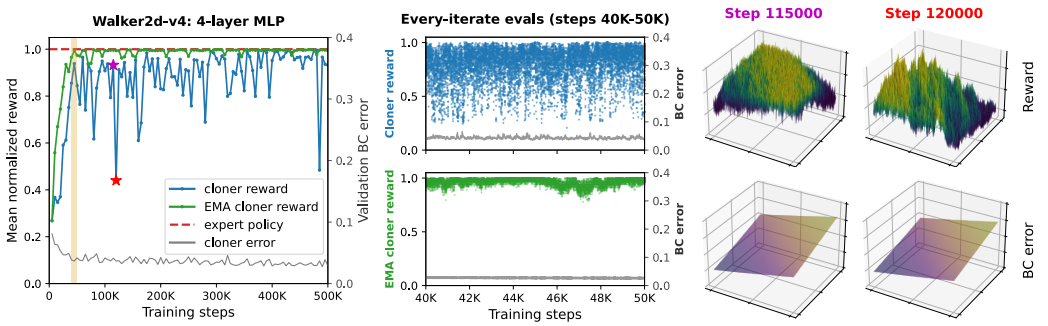


Figure 1: Typical reward instabilities over long-horizon ($H = 1000$) rollouts of neural behavior cloners for the `Walker2d-v4` MuJoCo locomotion task. *Left:* Rollout rewards (blue training curves) oscillate dramatically over the course of training (evaluated every 5000 iterations), while BC loss is stable. *Center:* Zoomed-in view of the highlighted region in (left). Large reward fluctuations are evident even between consecutive gradient iterates. *Right:* Exhaustive evaluation of small neighborhoods (in stochastic gradient directions) around iterates 115K and 120K, revealing a fractal reward landscape $\boldsymbol{\theta} \mapsto J_H(\pi_{\boldsymbol{\theta}})$; this jaggedness is invisible in the 1-step behavior cloning objective $\ell_{\mathrm{BC}}(\pi_{\boldsymbol{\theta}})$. Iterate averaging (EMA) drastically mitigates these effects (green training curves). Details are provided in Appendix B.1.1.

> *What causes instabilities in learning systems with feedback loops? To what extent can they be mitigated by algorithmic interventions alone, without resorting to collecting additional data?*

We explore this question in the context of behavior cloning (BC), a technique for training a policy to optimize a multi-step objective in a purely offline manner. This is achieved by introducing a surrogate loss function ℓ_{BC} (*behavior cloning loss*) that measures the distance between actions generated by some *expert policy* $\pi_{\boldsymbol{\theta}^\star}$ and those taken by the learner's policy, and then minimizing ℓ_{BC} over an offline dataset of expert trajectories. BC is sufficiently broad as to capture important tasks ranging from robotics and autonomous driving (Pomerleau, 1988; Codevilla et al., 2018; Chi et al., 2023) to autoregressive language generation (Chang et al., 2023a), and is popular in practice due to its simplicity and purely offline nature.

Our starting point is to observe that behavior cloning with deep neural networks exhibits **training instabilities** in which the multi-step objective (J_H), or *rollout reward*, of nearby checkpoints oscillates wildly during training, despite a well-behaved validation loss for ℓ_{BC}, even at the single iterate frequency; Figure 1 exhibits this phenomenon for a sample training curve of a behavior cloning policy in the `Walker2d-v4` MuJoCo locomotion task (Towers et al., 2023; Todorov et al., 2012). This oscillatory behavior is clearly undesirable; we cannot differentiate between low- and high-quality iterates based on (validation loss for) ℓ_{BC}, and thus cannot reliably select a high-quality policy.

With regard to the final performance of BC, it is well understood that scarce or low-quality data can lead to statistical challenges and consequent performance degradation of imitator policies; unsurprisingly, better data often improves the quality of a learned policy. Unfortunately, existing approaches to obtaining better data require either interactive access to the demonstrating expert (Ross & Bagnell, 2010; Laskey et al., 2017) or additional side information (Pfrommer et al., 2022; Block et al., 2023a); these interventions may be costly or impossible in many applications. Thus, in this work we treat the data generating process as fixed and aim to investigate whether we can mitigate oscillations and improve the performance of BC solely through the application of better algorithmic choices.

1.1 Contributions

In this paper, we aim to diagnose and ameliorate instabilities in behavior cloning that arise from training on the surrogate cost alone in the purely offline setting. Our findings are as follows.

Diagnosis of rollout oscillations: gradient variance amplification. In Section 3, we conduct an extensive empirical study (278 distinct interventions) of BC in continuous control tasks and inves-

tigate the effects that architecture, regularization, and optimization interventions have on training instability. We identify the presence of training oscillations and attribute them to *gradient variance amplification* (GVA): the propagation of minibatch SGD noise through closed-loop dynamics, leading to catastrophic error amplification resembling **butterfly effects** in chaotic systems. We ablate away much of the statistical difficulty, so that the presence of oscillations suggests that GVA is an *algorithmic* rather than *statistical* pathology.

Mitigating GVA: stabilizers for unstable optimizers. In Section 4, we investigate mitigations for GVA. Because GVA is caused by variance in the stochastic gradients, it can be ameliorated with variance reduction. Indeed, we observe (Section 3.2) that i) aggressively decaying the learning rate, and ii) greatly increasing the batch size through gradient accumulation, both have positive effects on the stability of training. Unfortunately, both of these interventions come at a great increase in compute cost. As such, our most significant finding (Section 4.1) is that *iterate averaging* by taking an Exponential Moving Average (EMA) of the optimization trajectory (Polyak & Juditsky, 1992; Ruppert, 1988), stabilizes training and mitigates GVA across a wide range of architectures and tasks, with essentially no downsides. While iterate averaging is popular in many deep learning research communities, this paper exposes iterate averaging as an *essential* design consideration when training any deep model in the presence of feedback loops.

A preliminary study of GVA in language generation. In Section 4.2, we broaden our focus by considering autoregressive sequence models. Our findings suggest that **unstable optimizers, when stabilized with iterate averaging to mitigate GVA, do not need full learning rate decay**, entailing potential computational and statistical benefits for training language models. For this reason, we suggest that EMA and related filters be designated as *stabilizers* in their own right and incorporated into deep learning pipelines in the same vein as modern optimizers and schedulers.

The applicability of convex theory. In Section 4.3, we complement our empirical results with theoretical vignettes. While the benefits of large learning rates cannot be explained in a convex setting, we demonstrate that—conditional on using theoretically suboptimal learning rates—stochastic convex optimization provides useful intuition for the causes and mitigations of GVA in deep learning. With our empirical results, these findings add to a line of work on surprising near-convex behavior in deep learning (Sandler et al., 2023; Frankle et al., 2020; Fang et al., 2022; Schaul et al., 2013).

1.2 Related work

Understanding and mitigating the effects of error amplification in behavior cloning has been the subject of much empirical work (Ross & Bagnell, 2010; Laskey et al., 2017), but most approaches use potentially impractical online query access to the expert policy; instead, we focus on a purely offline setting.

Complicated value function landscapes and their effect on training have been investigated in the context of planning in RL, with Dong et al. (2020) investigating natural examples of fractal reward functions, Wang et al. (2021) examining the instabilities arising from poor representations, and Emmons et al. (2021); Chang et al. (2023a) observing the fact that ℓ_{BC} is a poor proxy for J_H. To the best of our knowledge, there has not been a systematic study of training instability in the sense of rollout reward oscillation of nearby checkpoints.

In the context of stochastic optimization and optimization for deep learning, many previous works have attempted to reduce variance in theory (Polyak & Juditsky, 1992; Ruppert, 1988) and practice (Izmailov et al., 2018; Busbridge et al., 2023; Kaddour, 2022; Kaddour et al., 2023). Of particular note is Sandler et al. (2023), which demonstrates (empirically and in a toy theoretical setting) a form of equivalence between learning rate decay and iterate averaging. Our focus is not on variance reduction *per se*, but rather on the propagation of ,variance through unstable feedback loops. We expand on the relationship between our work and Sandler et al. (2023) and discuss other related work in Appendix A.

2 Preliminaries

MDP formalism. We let $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, H, \nu)$ denote a finite-horizon Markov decision process (MDP), where $\mathcal{S}$ is an abstract state space, $\mathcal{A}$ is an abstract action space, $P : \mathcal{S} \times \mathcal{A} \to \mathbf{\Delta}(\mathcal{S})$ is

a Markov transition operator. We denote by $r : \mathcal{S} \times \mathcal{A} \to [0, 1]$ a reward function and $H \in \mathbb{N}$ is the length of the horizon. Because we focus on continuous control tasks, we follow the notational conventions of control theory, denoting states by $\mathbf{x}$ and actions by $\mathbf{u}$. We let $\nu \in \mathbf{\Delta}(\mathcal{S})$ denote the initial distribution such that a trajectory from $\mathcal{M}$ consists of $\mathbf{x}_1 \sim \nu$ and $\mathbf{x}_{h+1} \sim P(\cdot \mid \mathbf{x}_h, \mathbf{u}_h)$ for all h.

The learner has access to a class of policies $\pi : \mathcal{S} \times \Theta \to \mathbf{\Delta}(\mathcal{A})$, where Θ is the *parameter* space and $\pi_{\boldsymbol{\theta}} : \mathcal{S} \to \Delta(\mathcal{A})$ is the policy induced by parameter $\boldsymbol{\theta} \in \Theta$. Given a policy $\pi_{\boldsymbol{\theta}}$, we denote its expected cumulative reward by $J_H(\pi_{\boldsymbol{\theta}}) = \mathbb{E}[\sum_{h=1}^{H} r(\mathbf{x}_h, \mathbf{u}_h)]$ where $\mathbf{u}_h \sim \pi_{\boldsymbol{\theta}}(\cdot|\mathbf{x}_h)$ and the expectation is with respect to both the transition dynamics of $\mathcal{M}$ and the possible stochasticity of the policy. Our experiments focus on MDPs whose transition operators P are deterministic, i.e., there exists a function $f : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ such that $\mathbf{x}_{h+1} = f(\mathbf{x}_h, \mathbf{u}_h)$ for all h. In this case the only stochasticity of the system comes from the sampling of the initial state $\mathbf{x}_1 \sim \nu$ (and possibly the policy).

Imitation learning and behavior cloning. In imitation learning, we are given an offline data set of N trajectories $\mathcal{D}_{\text{off}} = \{(\mathbf{x}_h^{(i)}, \mathbf{u}_h^{(i)})_{1 \le h \le H} \mid 1 \le i \le N\}$ generated by an expert policy $\pi_{\boldsymbol{\theta}^\star}$ interacting with the MDP $\mathcal{M}$. In this work, we always consider *deterministic policies*, i.e., where for all $\mathbf{x}$, $\pi_{\boldsymbol{\theta}}(\mathbf{x})$ has support on a single action; in particular this holds for the expert $\pi_{\boldsymbol{\theta}^\star}$. The goal of the learner is to produce a policy $\pi_{\widehat{\boldsymbol{\theta}}}$ that maximizes the expected cumulative reward $J_H(\pi_{\widehat{\boldsymbol{\theta}}})$ over an episode. We focus on the popular *behavior cloning* (BC) framework, where we fix a loss function $\ell_{\text{BC}} : \mathcal{A} \times \mathcal{A} \to \mathbb{R}$ that measures the distance from the actions produced by $\pi_{\boldsymbol{\theta}^\star}$, and learn $\pi_{\widehat{\boldsymbol{\theta}}}$ by attempting to minimize the empirical risk of ℓ_{BC} over $\mathcal{D}_{\text{off}}$; we abuse notation by denoting $\ell_{\text{BC}}(\pi_{\boldsymbol{\theta}}) := \mathbb{E}_{\mathcal{D}_{\text{off}}}[\ell_{\text{BC}}(\pi_{\boldsymbol{\theta}}(\mathbf{x}), \mathbf{u})]$ The basic premise behind behavior cloning is that ℓ_{BC} should be chosen such that if $\ell_{\text{BC}}(\pi_{\widehat{\boldsymbol{\theta}}}) \ll 1$ then $J_H(\pi_{\widehat{\boldsymbol{\theta}}}) \approx J_H(\pi_{\boldsymbol{\theta}^\star})$; that is, imitation of the expert is a surrogate for large cumulative reward. In line with common practice in BC (Janner et al., 2021; Shafiullah et al., 2022; Chi et al., 2023), the imitator policies in our experiments augment the state with the previous action, i.e., $\pi_{\boldsymbol{\theta}} : \mathcal{S} \times \mathcal{A} \to \mathcal{A}$, which can be integrated into the previous formalism by expanding the state space. For the special case of the first state $\mathbf{x}_1$, we always let $\mathbf{u}_0 = \mathbf{0}$.

Notation. Throughout the paper, we denote vectors by bold lower case letters and matrices by bold upper case letters.[1] We reserve $\boldsymbol{\theta}$ for a parameter of our policy and J_H for the cumulative reward over a trajectory, omitting H when it is clear from context. For conciseness, we often refer to J_H as the *reward*; the per-step reward function r makes no appearance in the rest of the paper. Given a set $\mathcal{U}$, we let $\mathbf{\Delta}(\mathcal{U})$ denote the class of probability distributions on $\mathcal{U}$.

3 Diagnosis of rollout oscillations: gradient variance amplification

3.1 Instabilities in behavior cloning of MuJoCo tasks

Experimental setup. We investigate instabilities in behavior cloning for the {`Walker2d`, `Hopper`, `HalfCheetah`, `Humanoid`, `Ant`}`-v4` environments from the OpenAI Gymnasium (Towers et al., 2023), all rendered in MuJoCo (Todorov et al., 2012). We focus on `Walker2d-v4` for the discussion that follows, and defer detailed discussion of further environments (which exhibit similar behavior) to Appendix B. Our expert is a multilayer perceptron (MLP) trained with Soft Actor Critic (SAC) (Haarnoja et al., 2018) for 3M steps with `stable-baselines3` (Raffin et al., 2021), with out-of-the-box hyperparameters.[2] The *default* imitator is a 4 layer MLP; details are in Appendix B. We examine several widths and depths, as well as Transformer (Vaswani et al., 2017) imitators.

Our first suite of experiments aims to isolate instability from *statistical difficulties*. We set up the experiments to make the behavior cloning problem as easy as possible. First, we focus on the "large-data" regime $N = H = 1000$, in which overfitting with respect to the BC loss $\ell_{\text{BC}}(\pi_{\widehat{\boldsymbol{\theta}}})$ is not a problem (see Figure 1), and thus poor rollout performance for $J_H(\pi_{\widehat{\boldsymbol{\theta}}})$ cannot be blamed on

[1] In particular, we denote states by $\mathbf{x}$ and actions by $\mathbf{u}$ in order to emphasize that, in our experiments, they are vectors in Euclidean space.

[2] By default, the Stable-Baselines3 SAC agent is stochastic, but we enforce determinism by selecting the mean action of the resulting policy. This results in negligible degradations to the rewards; see Figure 5.

insufficient data; this removes a typical source of statistical difficulty faced in applying behavior cloning to domains such as robotics (Chi et al., 2023; Pfrommer et al., 2022; Ross & Bagnell, 2010; Laskey et al., 2017). Beyond focusing on the large-data regime, (i) we consider only deterministic dynamics and deterministic experts, and (ii) we include within our default model the same class of MLPs that parameterize the expert policies, ensuring that expressivity is not an issue. As such, we have placed ourselves in perhaps the easiest possible setting for behavior cloning.

In Figure 1 (Left), we compare the evolution of the BC loss $\ell_{\mathrm{BC}}(\pi_{\widehat{\boldsymbol{\theta}}})$ (on a validation set) and reward $J_H(\pi_{\widehat{\boldsymbol{\theta}}})$ for imitator policies in the `Walker2d-v4` MuJoCo locomotion task. In this figure, we observe extreme oscillatory behavior in $J_H(\pi_{\widehat{\boldsymbol{\theta}}})$, juxtaposed with smoothly decaying $\ell_{\mathrm{BC}}(\pi_{\widehat{\boldsymbol{\theta}}})$. In Figure 1 (Middle), we zoom in on the training trajectory between iterates 40K and 50K and observe that the same instability persists even at the *every-iterate* level. Toward identifying what causes these instabilities, Figure 1 (Right) displays an experiment in which we independently sample two stochastic gradients of the training loss at a fixed checkpoint with good rollout reward. Policy weights are then perturbed by small steps in each of the two directions, and we evaluate the resulting reward $J_H(\pi_{\widehat{\boldsymbol{\theta}}})$ over 20 rollouts, along with the BC loss $\ell_{\mathrm{BC}}(\pi_{\widehat{\boldsymbol{\theta}}})$ on a held-out validation set. We see that nearby models vary erratically in terms of rollout performance, but vary smoothly in validation BC loss. These findings are reproduced consistently across the other environments and architectures in Appendix B; thus, we conclude:

(R1) **The reward landscape is highly sensitive to small changes in policy parameters**: small perturbations in model weights induce *butterfly effects* in the reward $J_H(\pi_{\widehat{\boldsymbol{\theta}}})$. In contrast, in the same regions, the *BC loss* landscape $\boldsymbol{\theta} \mapsto \ell_{\mathrm{BC}}(\pi_{\boldsymbol{\theta}})$ is well-behaved (nearly linear locally).

3.2 Instability is caused by gradient variance amplification

We now present compelling evidence that *variance in stochastic gradients* during training is responsible for training instability, because **gradient variance is amplified through the sensitivity of the rollout rewards to fluctuations in network parameters**. In Figure 2, we visualize evolution of both $\ell_{\mathrm{BC}}(\pi_{\widehat{\boldsymbol{\theta}}})$ and $J_H(\pi_{\widehat{\boldsymbol{\theta}}})$ over training for a variety of potential algorithmic interventions. We find that neither changing the model architecture and scale (1st row) nor standard regularization techniques (2nd row) ameliorate the training instabilities observed. We do see, however, that aggressively decaying the learning rate and increasing the batch size (3rd row) significantly reduces oscillations (at least when measuring mean rewards), at the expense of substantially slowing down training. Thus, we conclude that fluctuations from stochasticity in the gradients are to blame for oscillations in rollout rewards, and term this phenomenon *gradient variance amplification* (GVA). To summarize:

(R2) **GVA arises from algorithmic suboptimality rather than an information-theoretic limit.** Even with "infinite" training data (i.e., fresh trajectories with i.i.d. initial conditions at each training step), rollout oscillations persist.

(R3) **Training oscillations are *not* mitigated by many standard approaches to regularization**, including architectural interventions and increased regularization. On the other hand, **oscillations are ameliorated by variance reduction techniques**, such as large batch sizes, learning rate decay, and iterate averaging.

Appendix B shows that (R2) and (R3) remain true across environments and model architectures. In addition, we find that training instability is not the result of inadequate network architecture; we observe oscillations across model scales, and for both MLP and Transformer architectures.

While Figure 2 shows that it is possible to quell GVA using small learning rates or large batch sizes, this may not always be practical, as both interventions can incur steep computational costs.[3] Even worse, the success of continuous optimization in deep learning depends on non-convex feature learning mechanisms (Chizat et al., 2019), and too small a learning rate or too large a batch size can have deleterious effects on generalization.[4] Thus, it is vital to seek interventions that are holistically compatible with existing deep learning pipelines. Among these, Figure 2 highlights that

[3] As another unsatisfactory compromise, we also find that shallower models are less susceptible to GVA.

[4] We refer to some theoretical and empirical accounts in Appendix A.

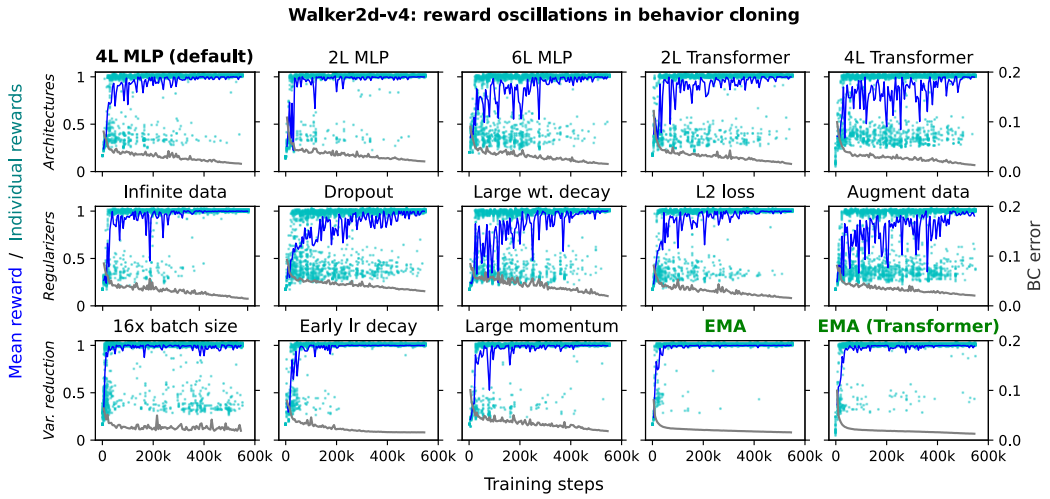


Figure 2: Highlights from a large suite of experiments, suggesting an algorithmic (rather than statistical) origin of reward oscillations. All plots use the 4-layer MLP architecture unless otherwise specified. Blue curves show mean rewards over 20 initial conditions, while teal dots show disaggregated per-episode rewards (such that each point represents the rollout reward of a fixed initial condition of the policy at the current iterate). These oscillations persist across dataset sizes, architectures, model scales, and choices of regularizers, and diminish toward the end of training as the learning rate decays to 0. They are most strongly mitigated by **variance reduction strategies**. Here, we opt for direct visualizations, providing a qualitative demonstration of GVA and its mitigations. We accompany these with quantitative comparisons in Appendix B.1.2.

a large momentum coefficient is mildly helpful, but taking an exponential moving average (EMA) of iterates (Polyak & Juditsky, 1992) is *extremely* effective. This motivates us to take a closer look at the latter in Section 4 through another suite of experiments.

3.3 Understanding GVA: mismatch between BC loss and rollout reward

The disparity between behavior cloning loss $\ell_{\mathrm{BC}}(\pi_{\widehat{\boldsymbol{\theta}}})$ and rollout reward $J_H(\pi_{\widehat{\boldsymbol{\theta}}})$ has long been appreciated in the imitation learning literature, and is understood to be caused by *error amplification*, the process by which mildly erroneous predictions, when fed repeatedly through feedback loops, result in highly suboptimal performance (Chen & Hazan, 2021; Wang et al., 2020a). More precisely, for given ℓ_{BC} and J_H as well as a policy $\pi_{\boldsymbol{\theta}^\star}$ and $\delta > 0$, we define the *error amplification constant* at scale δ to be the maximal value (with respect fo $\boldsymbol{\theta}$) for $J_H(\pi_{\boldsymbol{\theta}^\star}) - J_H(\pi_{\boldsymbol{\theta}})$ such that $\ell_{\mathrm{BC}}(\pi_{\boldsymbol{\theta}}) - \ell_{\mathrm{BC}}(\pi_{\boldsymbol{\theta}^\star}) < \delta$. The following proposition provides a simple theoretical illustration for how small fluctuations in BC loss can be drastically amplified by feedback between imperfectly-imitated policies and system dynamics.

Proposition 3.1 (Example of exponential error amplification)**.** *Let $\mathcal{B}_\delta$ denote the set of δ-Lipschitz functions $\Delta : \mathcal{S} \to \mathcal{A}$ with $\Delta(\mathbf{0}) = \mathbf{0}$. For any $\delta > 0$, there exists a deterministic MDP with horizon H and an expert policy $\pi_{\boldsymbol{\theta}^\star}$ such that the dynamics are Lipschitz in both state and action and $\pi_{\boldsymbol{\theta}^\star}$ is Lipschitz in the state, and such that*

$$\sup_{\Delta \in \mathcal{B}_\delta} \{J_H(\pi_{\boldsymbol{\theta}^\star}) - J_H(\pi_{\boldsymbol{\theta}^\star} + \Delta)\} \geq \Omega(H) \cdot \left(e^{\Omega(H\delta)} - 1\right),$$

yet $\sup_{\Delta \in \mathcal{B}_\delta} \ell_{\mathrm{BC}}(\pi_{\boldsymbol{\theta}^\star} + \Delta) \leq \mathcal{O}\left(H \cdot \delta^2\right)$, where ℓ_{BC} is the ℓ_2 loss. Thus, the error amplification constant is exponential *in the time horizon.*

Working model for GVA. Proposition 3.1 shows that even when ℓ_{BC} is uniformly small in a neighborhood around $\pi_{\boldsymbol{\theta}^\star}$, the rollout loss can be *exponentially large* in the same neighborhood. At the same time, there are good subsets of parameter space that do not experience this worst-case error amplification in our construction. We therefore hypothesize that, when stochastic optimization converges to a small neighborhood around zero-BC error models, it bounces between low-BC error

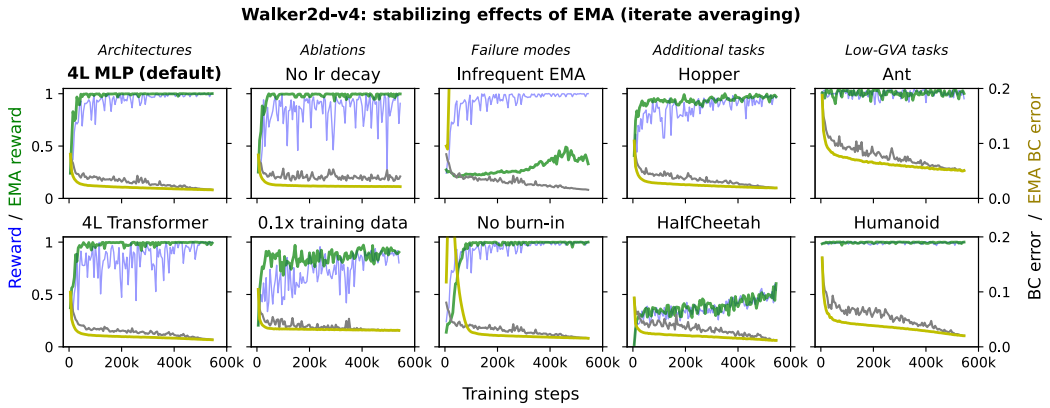


Figure 3: Iterate averaging significantly mitigates GVA-induced reward oscillations, **without needing to change the learning rate schedule or batch size**. These improvements hold across architectures, dataset sizes, and *some* tasks. *Column 2, bottom:* Algorithmic instabilities are more pronounced at smaller sample sizes; thus, **stabilization can lead to improved sample efficiency**. *Column 3:* We recommend updating the EMA at every iterate, with an initial burn-in phase, and with a tuned $\gamma^{(t)} = t^{-\alpha}$ decay, to avoid divergence or slower progress. *Columns 4-5*: We verify that the benefits of EMA are not exclusive to the `Walker2d-v4` task; for some other tasks (including the higher-dimensional `Humanoid-v4`), oscillations are more benign.

models that experience large error amplification, and those that do not. To recapitulate: *GVA is the phenomenon in which gradient stochasticity leads to optimization trajectories repeatedly visiting regions of parameter space with catastrophic error amplification.* Because our MuJoCo environments involve nonlinear contact dynamics (while the example in Proposition 3.1 is linear), oscillations in Figure 1 are even more chaotic than this example may suggest. We elaborate on this point further by studying the advantages of EMA on a discontinuous "cliff loss" problem in Section 4.3.

4 Mitigating GVA: Stabilizers for Unstable Optimizers

In Section 3.2, we isolated GVA as the primary cause of observed instabilities in BC (cf. Fig. 1) and identified iterate averaging with EMA (Polyak & Juditsky, 1992) as a promising remedy. In this section, we conduct an in-depth investigation of EMA as a mitigation. We start in continuous control (Section 4.1), and find EMA works almost unreasonably well at reducing GVA in the experimental testbed described in the prequel. Next, moving beyond continuous control (Section 4.2), we observe analogous effects in autoregressive language generation. In both settings, we find iterate averaging works so well as to **eliminate the need for full learning rate decay**; this leads us to recommend a conceptual reframing of EMA as a *stabilizer* for training neural networks, akin to (and interacting with) conventional optimizers and schedulers. We conclude (Section 4.3) by exploring the extent to which intuition on benefits of iterate averaging from the theory of stochastic convex optimization applies in our empirical settings.

4.1 The Outsized Benefit of Iterate Averaging

We recall the definition of the EMA method for iterate averaging (Polyak & Juditsky, 1992). Given an optimization trajectory $(\boldsymbol{\theta}^{(t)})_{0\le t} \subset \mathbb{R}^d$ and a sequence $(\gamma_t)_{1\le t} \subset [0,1]$, the EMA iterates $(\widetilde{\boldsymbol{\theta}}_\gamma^{(t)})$ are[5]

$$\widetilde{\boldsymbol{\theta}}_\gamma^{(0)} = \boldsymbol{\theta}^{(0)}, \quad \text{and } \widetilde{\boldsymbol{\theta}}_\gamma^{(t+1)} = (1-\gamma_t)\cdot \widetilde{\boldsymbol{\theta}}_\gamma^{(t)} + \gamma_t \cdot \boldsymbol{\theta}^{(t+1)}. \tag{4.1}$$

Many prior works have detailed the benefits of iterate averaging in stochastic convex optimization and beyond (see Appendix A). Here, we investigate its effect on GVA. We begin by considering the

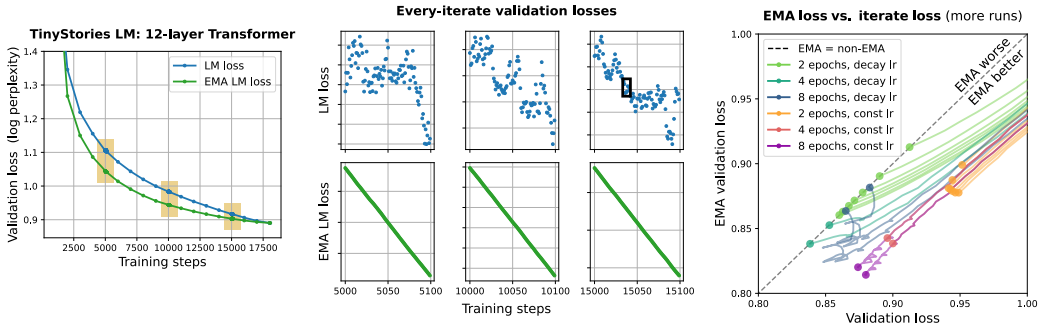
[5]Common heuristics include updating the EMA only after an initial "burn-in", and annealing γ with a polynomial decay: $\gamma^{(t)} = \max(t^{-\alpha}, \gamma_{\min})$. It is also customary to use $\beta^{(t)}$ to denote $1-\gamma^{(t)}$.

same MuJoCo framework as in Section 3. In Figure 3, we produce similar plots to those in Section 3, but this time juxtapose the vanilla trained models with an EMA of their iterates (further results and details are deferred to Appendix B). We observe the following:

(R4) **EMA iterate averaging strongly mitigates rollout oscillations**. *In every setup we consider,* across a variety of architectures and environments, EMA significantly reduces the oscillations in rollout reward; in no instance does it hurt performance.

We provide quantitative comparisons for a wide range of interventions in Figures 8 to 11.

4.2 AUTOREGRESSIVE SEQUENCE MODELS AND LANGUAGE GENERATION



[...] *One day, Tim and Spot were playing near the pit when they saw a big ball. They both wanted to play with...*

15027-15035: ...the ball, but they didn't want to share. Tim said, "I want the ball!" Spot barked, "No, I want the ball!" They started to fight over the ball. Tim's mom saw them fighting and said, "Stop fighting! You can both play with the ball if you share." Tim and Spot stopped fighting and [...]

15036-15037: ...it. Tim said, "Let's share the ball, Spot!" Spot barked happily, and they both played with the ball together. As they played, a little girl named Sue came by. She saw Tim and Spot sharing the ball and wanted to join them. Tim said, "Yes, you can play with us!" So, Tim, [...]

15038-15039: ...it. Tim said, "I want the ball!" Spot barked and wagged his tail. They both ran to the ball and started to play. As they played, the ball went into the pit. Tim and Spot looked at each other, not knowing what to do. Then, a friendly frog jumped out of the pit with the ball [...]

15040-15049: ...the ball, but they didn't want to share. Tim said, "I want the ball!" Spot barked, "No, I want the ball!" They started to fight over the ball. As they fought, the ball rolled into the pit. Tim and Spot stopped fighting and looked at the pit. They were sad because they couldn't [...]

Figure 4: **GVA in natural language generation**, with 270M-parameter Transformer models trained on TinyStories. *(Top row) Left:* Validation loss curves with and without EMA. *Center:* Zooming in on *(left)*, evaluations at every update demonstrate small per-iterate loss fluctuations, which are even smaller if EMA is applied; note that the green "lines" are also scatter plots. *Right:* Training paths in (model loss, EMA loss) space. EMA enables training without learning rate decay; this mitigates overfitting, resulting in the lowest-perplexity model. *(Bottom)* Examples of autoregressively generated text (with argmax decoding), where nearby training iterates can bifurcate. See Appendix B.2 for full results, including quantitative evaluations of these *"butterfly effects"* in generation.

We posit that GVA is a generic phenomenon that can manifest in disparate settings: whenever a model's predictions are applied within a (marginally stable or unstable) feedback loop, the closed-loop dynamics can amplify small fluctuations in a deleterious manner. A natural and timely setting with this structure—which complements continuous control–is autoregressive language modeling. Here, a network's parameters $\boldsymbol{\theta}$ are optimized on a 1-step prediction loss, which takes the role of $\ell_{\mathrm{BC}}(\pi_{\boldsymbol{\theta}})$. The network $\pi_{\boldsymbol{\theta}}$ is then used to generate a sequence of symbols $w_{1:H}$ by iteratively rolling out $\pi_{\boldsymbol{\theta}} : w_{1:h} \mapsto w_{h+1}$. Such models have been paradigm-shattering in NLP, code synthesis, and beyond. Motivated by the similarity of this pipeline to behavior cloning,[6] we perform a smaller set of analogous experiments on language generation. Our findings here parallel our findings for continuous control, and show (i) the presence of GVA, and (ii) substantial benefits of iterate averaging. In more detail, we train 270M-parameter 12-layer Transformer models on the TinyStories dataset (Eldan & Li, 2023), which serves as an inexpensive surrogate for a full-scale pretraining pipeline. Highlights are shown in Fig. 4, while Appendix B.2 provides full documentation, including larger-scale training runs with a non-synthetic corpus (Wikipedia). We summarize our findings below:

[6]Many works have investigated GPT-style pretraining through the lens of offline IL (Chang et al., 2023a). There are many degrees of freedom in evaluating performance; thus, we do not commit to a canonical notion of reward and measure GVA-induced oscillations via disagreements in long-horizon rollouts.

(R5) Autoregressive LMs exhibit significant rollout oscillations throughout training. **EMA stabilizes the trajectory, accelerates training, and improves generalization**, complementing (and potentially obviating) standard practices in learning rate annealing.

4.3 To what extent does convex theory explain the benefits of EMA?

We close by providing mathematical intuition as to why iterate averaging with EMA can reduce the oscillations caused by GVA. As discussed in Section 3.3, oscillations can occur when there is a disparity between the BC loss $\ell_{\mathrm{BC}}(\pi_{\widehat{\boldsymbol{\theta}}})$ on which we train and the rollout reward function $J(\pi_{\widehat{\boldsymbol{\theta}}})$ on which we evaluate. To study this phenomenon, we a consider simple, horizon-one behavior cloning problem with a single action determined by the model parameter $\boldsymbol{\theta}$. We take the *training loss* to be a quadratic $\ell_{\mathrm{BC}}(\boldsymbol{\theta}) = \frac{1}{2} \cdot \|\boldsymbol{\theta} - \boldsymbol{\mu}\|^2$, and the *rollout reward* $J(\cdot)$ to be

$$J(\theta) = \begin{cases} -\|\boldsymbol{\theta} - \boldsymbol{\mu}\|^2, & \|\boldsymbol{\theta} - \boldsymbol{\mu}\| \leq \epsilon \\ -C, & \text{otherwise} \end{cases}, \tag{4.2}$$

where $C \gg \epsilon^2 > 0$ are constants. Here the training loss is convex, but rollout reward is not; the latter exhibits a "cliff," dropping sharply from $-\epsilon^2$ to $-C$ once $\|\boldsymbol{\theta} - \boldsymbol{\mu}\| > \epsilon$. The pair (ℓ_{BC}, J) may be thought of as a discontinuous, horizon-one analogue of the example in Proposition 3.1, illustrating the contrast between extreme sensitivity of reward and insensitivity of the loss to the parameter of interest. The reward function encapsulates discontinuities arising in control tasks from, e.g., contract forces. In the MuJoCo walker, "cliff"-type behavior may come from an expert policy close to overbalancing the agent, with the learner's policy falling down if the parameter is "over the cliff."

We analyze SGD iterates $\boldsymbol{\theta}^{(t+1)} = \boldsymbol{\theta}^{(t+1)} - \eta(\boldsymbol{\theta}^{(t)} - \mu + \mathbf{w})$, where $\eta > 0$ is a constant step size and $\mathbf{w} \sim \mathcal{N}(0, \mathbf{I})$. This corresponds to SGD on a noisy version of the BC loss given by $\tilde{\ell}_{\mathrm{BC}}(\boldsymbol{\theta}) := \mathbb{E}[\|\boldsymbol{\theta}_t - \mathbf{u} + \mathbf{w}\|^2]$, which satisfies $\mathbb{E}_{\mathbf{w}}[\tilde{\ell}_{\mathrm{BC}}(\boldsymbol{\theta})] = \ell_{\mathrm{BC}}(\boldsymbol{\theta}) + \text{constant}$. We show that applying EMA to the resulting iterates achieves substantially higher rollout reward than vanilla SGD.

Proposition 4.1 (Informal version of Proposition C.6)**.** *Consider the setting in Eq. (4.2) for parameters $C \gg \epsilon^2 > 0$ in dimension one, and let $\boldsymbol{\theta}^{(T)}$ denote the SGD iterate with learning rate $\eta > 0$ as described above. Let $\tilde{\boldsymbol{\theta}}_\gamma^{(T)}$ denote the EMA iterate (4.1) with fixed parameter $\gamma_t \equiv \gamma \leq \eta$ satisfying $\gamma \gg 1/T$. Then, $\mathbb{E}[\ell_{\mathrm{BC}}(\boldsymbol{\theta}^{(T)})]$ scales as $\Theta(\eta)$, while $\mathbb{E}[\ell_{\mathrm{BC}}(\tilde{\boldsymbol{\theta}}_\gamma^{(T)})]$ scales as $\Theta(\gamma) \leq \eta$. In particular, when $\eta > c_1\epsilon$, and $\gamma \log(C/\gamma) \leq c_2\epsilon$, for absolute constants $c_1, c_2 > 0$, we find that*

$$\mathbb{E}[J(\boldsymbol{\theta}^\star) - J(\boldsymbol{\theta}^{(T)})] \geq \tfrac{C}{2}, \quad \textit{but} \quad \mathbb{E}[J(\boldsymbol{\theta}^\star) - J(\tilde{\boldsymbol{\theta}}_\gamma^{(T)})] \leq \mathcal{O}(\gamma).$$

This proposition holds, which shows that the rollout performance for EMA can be arbitrarily small relative to that of SGD, holds even in the regime where SGD is initialized at $\boldsymbol{\theta}^{(0)} = \boldsymbol{\mu}$ (so that both $\boldsymbol{\theta}^{(T)}$ and $\boldsymbol{\theta}_\gamma^{(T)}$ are unbiased estimates of $\boldsymbol{\mu}$), and thus highlights that EMA can reduce the variance that arises from accumulation of SGD noise.[7] Notice that Proposition 4.1 requires $\eta \geq \gamma \gg 1/T$, which is above the optimal step size of $\eta_t = 1/t$.[8] Indeed, in Appendix C we show that EMA, with the parameters we find empirically successful, only benefits optimization *above* these aggressively-decayed theoretically optimal learning rate schedules. Thus, we conclude that **convex theory reveals the variance-reducing benefit of *either* learning rate decay or EMA, but does not suggest which one is better.** We defer further theoretical results to Appendix C, and present an empirical study of a system motivated by the cliff loss in Appendix B.5; in particular, our analysis provides a simple example where GVA provably occurs, both theoretically and empirically.

The above example reveals the difference between the **statistical and algorithmic difficulties** of BC: with enough data, the empirical risk minimizer (sample mean) $\widehat{\boldsymbol{\theta}}$ of BC loss exhibits $\ell_{\mathrm{BC}}(\boldsymbol{\theta}) \sim 1/T \ll \epsilon$, which ensures J_H is small; on the other hand, with minibatch SGD and too large a learning rate, there is a noise floor on how close the non-EMA'd iterate $\widehat{\boldsymbol{\theta}}$ will be to $\boldsymbol{\theta}^\star$, ensuring that J_H is large.

[7] We compare to similar findings (Sandler et al., 2023) in Appendix A.

[8] Note that the $\eta_t = \frac{1}{t}$ step size schedule gives the sample mean, which is the maximum likelihood estimator for our objective.

REFERENCES

Emmanuel Abbe and Colin Sandon. Poly-time universality and limitations of deep learning. *arXiv preprint arXiv:2001.02992*, 2020.

Emmanuel Abbe, Pritish Kamath, Eran Malach, Colin Sandon, and Nathan Srebro. On the power of differentiable learning versus pac and sq learning. *Advances in Neural Information Processing Systems*, 34:24340–24351, 2021.

Naman Agarwal, Brian Bullins, Elad Hazan, Sham Kakade, and Karan Singh. Online control with adversarial disturbances. In *International Conference on Machine Learning*, pp. 111–119. PMLR, 2019.

Maksym Andriushchenko, Aditya Vardhan Varre, Loucas Pillaud-Vivien, and Nicolas Flammarion. SGD with large step sizes learns sparse features. In *International Conference on Machine Learning*, pp. 903–925. PMLR, 2023.

Marcin Andrychowicz, Anton Raichuk, Piotr Stańczyk, Manu Orsini, Sertan Girgin, Raphaël Marinier, Leonard Hussenot, Matthieu Geist, Olivier Pietquin, Marcin Michalski, et al. What matters for on-policy deep actor-critic methods? A large-scale study. In *International Conference on Learning Representations*, 2020.

David Angeli. A Lyapunov approach to incremental stability properties. *IEEE Transactions on Automatic Control*, 47(3):410–421, 2002.

Kushal Arora, Layla El Asri, Hareesh Bahuleyan, and Jackie Chi Kit Cheung. Why exposure bias matters: An imitation learning perspective of error accumulation in language generation. *arXiv preprint arXiv:2204.01171*, 2022.

Boaz Barak, Benjamin Edelman, Surbhi Goel, Sham Kakade, Eran Malach, and Cyril Zhang. Hidden progress in deep learning: Sgd learns parities near the computational limit. *Advances in Neural Information Processing Systems*, 35:21750–21764, 2022.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pp. 2397–2430. PMLR, 2023.

Adam Block and Max Simchowitz. Efficient and near-optimal smoothed online learning for generalized linear functions. *Advances in Neural Information Processing Systems*, 35:7477–7489, 2022.

Adam Block, Daniel Pfrommer, and Max Simchowitz. Imitating complex trajectories: Bridging low-level stability and high-level behavior. *arXiv preprint arXiv:2307.14619*, 2023a.

Adam Block, Max Simchowitz, and Alexander Rakhlin. Oracle-efficient smoothed online learning for piecewise continuous decision making. In *The Thirty Sixth Annual Conference on Learning Theory*, pp. 1618–1665. PMLR, 2023b.

Adam Block, Max Simchowitz, and Russ Tedrake. Smoothed online learning for prediction in piecewise affine systems. *arXiv preprint arXiv:2301.11187*, 2023c.

Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010: 19th International Conference on Computational Statistics*, pp. 177–186. Springer, 2010.

Olivier Bousquet and André Elisseeff. Stability and generalization. *The Journal of Machine Learning Research*, 2:499–526, 2002.

Mark Braverman, Xinyi Chen, Sham Kakade, Karthik Narasimhan, Cyril Zhang, and Yi Zhang. Calibration, entropy rates, and memory in language models. In *International Conference on Machine Learning*, pp. 1089–1099. PMLR, 2020.

Dan Busbridge, Jason Ramapuram, Pierre Ablin, Tatiana Likhomanenko, Eeshan Gunesh Dhekane, Xavier Suau, and Russ Webb. How to scale your EMA. *arXiv preprint arXiv:2307.13813*, 2023.

Anthony Carbery and James Wright. Distributional and Lq norm inequalities for polynomials over convex bodies in rn. *Mathematical Research Letters*, 8(3):233–248, 2001.

Jonathan D Chang, Kiante Brantley, Rajkumar Ramamurthy, Dipendra Misra, and Wen Sun. Learning to generate better than your LLM. *arXiv preprint arXiv:2306.11816*, 2023a.

Jonathan D Chang, Qinqing Zhen, Brandon Amos, Wen Sun, and Mikael Henaff. A large scale study of deep imitation learning on the arcade learning environment. *Personal Communication*, 2023b.

Xinyi Chen and Elad Hazan. Black-box control for linear dynamical systems. In *Conference on Learning Theory*, pp. 1114–1143. PMLR, 2021.

Xiang Cheng, Dong Yin, Peter Bartlett, and Michael Jordan. Stochastic gradient and Langevin processes. In *International Conference on Machine Learning*, pp. 1810–1819. PMLR, 2020.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.

Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. *Advances in Neural Information Processing Systems*, 32, 2019.

Felipe Codevilla, Matthias Müller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. In *2018 IEEE international Conference on Robotics and Automation (ICRA)*, pp. 4693–4700. IEEE, 2018.

Alexandru Damian, Jason Lee, and Mahdi Soltanolkotabi. Neural networks can learn representations with gradient descent. In *Conference on Learning Theory*, pp. 5413–5452. PMLR, 2022.

Sarah Dean and Jamie Morgenstern. Preference dynamics under personalized recommendations. In *Proceedings of the 23rd ACM Conference on Economics and Computation*, pp. 795–816, 2022.

Sarah Dean, Horia Mania, Nikolai Matni, Benjamin Recht, and Stephen Tu. Regret bounds for robust adaptive control of the linear quadratic regulator. *Advances in Neural Information Processing Systems*, 31, 2018.

Aaron Defazio and Léon Bottou. On the ineffectiveness of variance reduced optimization for deep learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Kefan Dong, Yuping Luo, Tianhe Yu, Chelsea Finn, and Tengyu Ma. On the expressivity of neural networks for deep reinforcement learning. In *International conference on machine learning*, pp. 2627–2637. PMLR, 2020.

Simon S Du, Sham M Kakade, Jason D Lee, Shachar Lovett, Gaurav Mahajan, Wen Sun, and Ruosong Wang. Bilinear classes: A structural framework for provable generalization in RL. *International Conference on Machine Learning*, 2021.

Benjamin L Edelman, Surbhi Goel, Sham Kakade, and Cyril Zhang. Inductive biases and variable creation in self-attention mechanisms. In *International Conference on Machine Learning*, pp. 5793–5831. PMLR, 2022.

Ronen Eldan and Yuanzhi Li. TinyStories: How small can language models be and still speak coherent english? *arXiv preprint arXiv:2305.07759*, 2023.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1, 2021.

Scott Emmons, Benjamin Eysenbach, Ilya Kostrikov, and Sergey Levine. Rvs: What is essential for offline rl via supervised learning? *arXiv preprint arXiv:2112.10751*, 2021.

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.

Cong Fang, Yihong Gu, Weizhong Zhang, and Tong Zhang. Convex formulation of overparameterized deep neural networks. *IEEE Transactions on Information Theory*, 68(8):5340–5352, 2022.

Maryam Fazel, Rong Ge, Sham Kakade, and Mehran Mesbahi. Global convergence of policy gradient methods for the linear quadratic regulator. In *International Conference on Machine Learning*, pp. 1467–1476. PMLR, 2018.

Pete Florence, Corey Lynch, Andy Zeng, Oscar A Ramirez, Ayzaan Wahid, Laura Downs, Adrian Wong, Johnny Lee, Igor Mordatch, and Jonathan Tompson. Implicit behavioral cloning. In *Conference on Robot Learning*, pp. 158–168. PMLR, 2022.

Dylan J Foster, Sham M Kakade, Jian Qian, and Alexander Rakhlin. The statistical complexity of interactive decision making. *arXiv preprint arXiv:2112.13487*, 2021a.

Dylan J Foster, Akshay Krishnamurthy, David Simchi-Levi, and Yunzong Xu. Offline reinforcement learning: Fundamental barriers for value function approximation. *arXiv preprint arXiv:2111.10919*, 2021b.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, pp. 3259–3269. PMLR, 2020.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, pp. 2052–2062. PMLR, 2019.

Udaya Ghai, Holden Lee, Karan Singh, Cyril Zhang, and Yi Zhang. No-regret prediction in marginally stable systems. In *Conference on Learning Theory*, pp. 1714–1757. PMLR, 2020.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. PMLR, 2018.

Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International conference on machine learning*, pp. 1225–1234. PMLR, 2016.

Nicholas JA Harvey, Christopher Liaw, Yaniv Plan, and Sikander Randhawa. Tight analyses for non-smooth stochastic gradient descent. In *Conference on Learning Theory*, pp. 1579–1613. PMLR, 2019.

Aaron Havens and Bin Hu. On imitation learning of linear control policies: Enforcing stability and robustness constraints via lmi conditions. In *2021 American Control Conference (ACC)*, pp. 882–887. IEEE, 2021.

Elad Hazan and Karan Singh. Introduction to online nonstochastic control. *arXiv preprint arXiv:2211.09619*, 2022.

Haowei He, Gao Huang, and Yang Yuan. Asymmetric valleys: Beyond sharp and flat local minima. *Advances in Neural Information Processing Systems*, 32, 2019.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Jorgen Hoffmann-Jorgensen, Lawrence A Shepp, and Richard M Dudley. On the lower tail of gaussian seminorms. *The Annals of Probability*, pp. 319–342, 1979.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. *arXiv preprint arXiv:1803.05407*, 2018.

Michael Janner, Qiyang Li, and Sergey Levine. Offline reinforcement learning as one big sequence modeling problem. *Advances in Neural Information Processing Systems*, 34:1273–1286, 2021.

Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, John Langford, and Robert E Schapire. Contextual decision processes with low Bellman rank are PAC-learnable. In *International Conference on Machine Learning*, pp. 1704–1713, 2017.

Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M Kakade, and Michael I Jordan. How to escape saddle points efficiently. In *International conference on machine learning*, pp. 1724–1732. PMLR, 2017.

Chi Jin, Qinghua Liu, and Sobhan Miryoosefi. Bellman eluder dimension: New rich classes of RL problems, and sample-efficient algorithms. *Neural Information Processing Systems*, 2021.

Jean Kaddour. Stop wasting my time! Saving days of ImageNet and BERT training with latest weight averaging. *arXiv preprint arXiv:2209.14981*, 2022.

Jean Kaddour, Oscar Key, Piotr Nawrot, Pasquale Minervini, and Matt J Kusner. No train no gain: Revisiting efficient training algorithms for Transformer-based language models. *arXiv preprint arXiv:2307.06440*, 2023.

Sham Kakade, Akshay Krishnamurthy, Kendall Lowrey, Motoya Ohnishi, and Wen Sun. Information theoretic regret bounds for online nonlinear control. *Advances in Neural Information Processing Systems*, 33:15312–15325, 2020.

Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960.

Liyiming Ke, Jingqiang Wang, Tapomayukh Bhattacharjee, Byron Boots, and Siddhartha Srinivasa. Grasping with chopsticks: Combating covariate shift in model-free imitation learning for fine manipulation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6185–6191. IEEE, 2021.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

Karl Krauth, Sarah Dean, Alex Zhao, Wenshuo Guo, Mihaela Curmei, Benjamin Recht, and Michael I Jordan. Do offline metrics predict online performance in recommender systems? *arXiv preprint arXiv:2011.07931*, 2020.

Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in Neural Information Processing Systems*, 32, 2019.

Daniel Kunin, Javier Sagastuy-Brena, Surya Ganguli, Daniel LK Yamins, and Hidenori Tanaka. Neural mechanics: Symmetry and broken conservation laws in deep learning dynamics. *arXiv preprint arXiv:2012.04728*, 2020.

Simon Lacoste-Julien, Mark Schmidt, and Francis Bach. A simpler approach to obtaining an o (1/t) convergence rate for the projected stochastic subgradient method. *HAL*, 2012, 2012.

Michael Laskey, Jonathan Lee, Roy Fox, Anca Dragan, and Ken Goldberg. Dart: Noise injection for robust imitation learning. In *Conference on Robot Learning*, pp. 143–156. PMLR, 2017.

Jean-François Le Gall. *Brownian motion, martingales, and stochastic calculus*. Springer, 2016.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31, 2018.

Qianxiao Li, Cheng Tai, and E Weinan. Stochastic modified equations and dynamics of stochastic gradient algorithms i: Mathematical foundations. *The Journal of Machine Learning Research*, 20 (1):1474–1520, 2019a.

Yuanzhi Li and Yang Yuan. Convergence analysis of two-layer neural networks with relu activation. *Advances in neural information processing systems*, 30, 2017.

Yuanzhi Li, Colin Wei, and Tengyu Ma. Towards explaining the regularization effect of initial large learning rate in training neural networks. *Advances in Neural Information Processing Systems*, 32, 2019b.

Zhiyuan Li, Kaifeng Lyu, and Sanjeev Arora. Reconciling modern deep learning with traditional optimization analyses: The intrinsic learning rate. *Advances in Neural Information Processing Systems*, 33:14544–14555, 2020.

Ernest Lindelöf. Sur l'application de la méthode des approximations successives aux équations différentielles ordinaires du premier ordre. *Comptes rendus hebdomadaires des séances de l'Académie des sciences*, 116(3):454–457, 1894.

Bingbin Liu, Jordan T Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. Exposing attention glitches with flip-flop language modeling. *arXiv preprint arXiv:2306.00946*, 2023a.

Bingbin Liu, Jordan T Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. Transformers learn shortcuts to automata. 2023b.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Marlos C Machado, Marc G Bellemare, Erik Talvitie, Joel Veness, Matthew Hausknecht, and Michael Bowling. Revisiting the arcade learning environment: Evaluation protocols and open problems for general agents. *Journal of Artificial Intelligence Research*, 61:523–562, 2018.

Sadhika Malladi, Kaifeng Lyu, Abhishek Panigrahi, and Sanjeev Arora. On the sdes and scaling rules for adaptive gradient algorithms. *Advances in Neural Information Processing Systems*, 35: 7697–7711, 2022.

Stephan Mandt, Matthew D Hoffman, and David M Blei. Stochastic gradient descent as approximate bayesian inference. *arXiv preprint arXiv:1704.04289*, 2017.

Yurii Nesterov et al. *Lectures on convex optimization*, volume 137. Springer, 2018.

Charles Packer, Katelyn Gao, Jernej Kos, Philipp Krähenbühl, Vladlen Koltun, and Dawn Song. Assessing generalization in deep reinforcement learning. *arXiv preprint arXiv:1810.12282*, 2018.

Paavo Parmas, Carl Edward Rasmussen, Jan Peters, and Kenji Doya. Pipps: Flexible model-based policy search robust to the curse of chaos. In *International Conference on Machine Learning*, pp. 4065–4074. PMLR, 2018.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 32*, pp. 8024–8035. Curran Associates, Inc., 2019.

Juan Perdomo, Jack Umenberger, and Max Simchowitz. Stabilizing dynamical systems via policy gradient methods. *Advances in Neural Information Processing Systems*, 34:29274–29286, 2021.

Daniel Pfrommer, Thomas Zhang, Stephen Tu, and Nikolai Matni. Tasil: Taylor series imitation learning. *Advances in Neural Information Processing Systems*, 35:20162–20174, 2022.

Daniel Pfrommer, Max Simchowitz, Tyler Westenbroek, Nikolai Matni, and Stephen Tu. The power of learned locally linear models for nonlinear policy optimization. *arXiv preprint arXiv:2305.09619*, 2023.

Boris T Polyak and Anatoli B Juditsky. Acceleration of stochastic approximation by averaging. *SIAM journal on control and optimization*, 30(4):838–855, 1992.

Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. *Advances in Neural Information Processing Systems*, 1, 1988.

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. URL http://jmlr.org/papers/v22/20-1364.html.

Alexander Rakhlin, Ohad Shamir, and Karthik Sridharan. Making gradient descent optimal for strongly convex stochastic optimization. *arXiv preprint arXiv:1109.5647*, 2011.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, pp. 400–407, 1951.

Stéphane Ross and Drew Bagnell. Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 661–668. JMLR Workshop and Conference Proceedings, 2010.

Mark Rudelson and Roman Vershynin. Hanson-wright inequality and sub-gaussian concentration. 2013.

David Ruppert. Efficient estimations from a slowly convergent robbins-monro process. Technical report, Cornell University Operations Research and Industrial Engineering, 1988.

Daniel Russo and Benjamin Van Roy. Eluder dimension and the sample complexity of optimistic exploration. In *Advances in Neural Information Processing Systems*, pp. 2256–2264, 2013.

Mark Sandler, Andrey Zhmoginov, Max Vladymyrov, and Nolan Miller. Training trajectories, mini-batch losses and the curious role of the learning rate. *arXiv preprint arXiv:2301.02312*, 2023.

Sunny Sanyal, Jean Kaddour, Abhishek Kumar, and Sujay Sanghavi. Understanding the effectiveness of early weight averaging for training large language models. *arXiv preprint arXiv:2306.03241*, 2023.

Tuhin Sarkar and Alexander Rakhlin. Near optimal finite time identification of arbitrary linear dynamical systems. In *International Conference on Machine Learning*, pp. 5610–5618. PMLR, 2019.

Tom Schaul, Sixin Zhang, and Yann LeCun. No more pesky learning rates. In *International conference on machine learning*, pp. 343–351. PMLR, 2013.

Thibault Sellam, Steve Yadlowsky, Jason Wei, Naomi Saphra, Alexander D'Amour, Tal Linzen, Jasmijn Bastings, Iulia Turc, Jacob Eisenstein, Dipanjan Das, et al. The multiberts: Bert reproductions for robustness analysis. *arXiv preprint arXiv:2106.16163*, 2021.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.

Nur Muhammad Shafiullah, Zichen Cui, Ariuntuya Arty Altanzaya, and Lerrel Pinto. Behavior transformers: Cloning k modes with one stone. *Advances in neural information processing systems*, 35:22955–22968, 2022.

Max Simchowitz. Making non-stochastic control (almost) as easy as stochastic. *Advances in Neural Information Processing Systems*, 33:18318–18329, 2020.

Max Simchowitz and Dylan Foster. Naive exploration is optimal for online lqr. In *International Conference on Machine Learning*, pp. 8937–8948. PMLR, 2020.

Max Simchowitz, Horia Mania, Stephen Tu, Michael I Jordan, and Benjamin Recht. Learning without mixing: Towards a sharp analysis of linear system identification. In *Conference On Learning Theory*, pp. 439–473. PMLR, 2018.

Max Simchowitz, Karan Singh, and Elad Hazan. Improper learning for non-stochastic control. In *Conference on Learning Theory*, pp. 3320–3436. PMLR, 2020.

Samuel L Smith, Pieter-Jan Kindermans, Chris Ying, and Quoc V Le. Don't decay the learning rate, increase the batch size. *arXiv preprint arXiv:1711.00489*, 2017.

Hyung Ju Suh, Max Simchowitz, Kaiqing Zhang, and Russ Tedrake. Do differentiable simulators give better policy gradients? In *International Conference on Machine Learning*, pp. 20668–20696. PMLR, 2022.

Wen Sun, Nan Jiang, Akshay Krishnamurthy, Alekh Agarwal, and John Langford. Model-based RL in contextual decision processes: PAC bounds and exponential improvements over model-free approaches. In *Conference on learning theory*, pp. 2898–2933. PMLR, 2019.

Matus Telgarsky. Feature selection with gradient descent on two-layer networks in low-rotation regimes. *arXiv preprint arXiv:2208.02789*, 2022.

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. doi: 10.1109/IROS.2012.6386109.

Mark Towers, Jordan K. Terry, Ariel Kwiatkowski, John U. Balis, Gianluca de Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Arjun KG, Markus Krimmel, Rodrigo Perez-Vicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Andrew Tan Jin Shen, and Omar G. Younis. Gymnasium, March 2023. URL https://zenodo.org/record/8127025.

Stephen Tu, Alexander Robey, Tingnan Zhang, and Nikolai Matni. On the sample complexity of stability constrained imitation learning. In *Learning for Dynamics and Control Conference*, pp. 180–191. PMLR, 2022.

Arjan J Van Der Schaft and Hans Schumacher. *An introduction to hybrid dynamical systems*, volume 251. springer, 2007.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Ruosong Wang, Dean P Foster, and Sham M Kakade. What are the statistical limits of offline rl with linear function approximation? *arXiv preprint arXiv:2010.11895*, 2020a.

Ruosong Wang, Russ R Salakhutdinov, and Lin Yang. Reinforcement learning with general value function approximation: Provably efficient approach via bounded eluder dimension. *Advances in Neural Information Processing Systems*, 33, 2020b.

Ruosong Wang, Yifan Wu, Ruslan Salakhutdinov, and Sham Kakade. Instabilities of offline rl with pre-trained neural representation. In *International Conference on Machine Learning*, pp. 10948–10960. PMLR, 2021.

Tyler Westenbroek, Max Simchowitz, Michael I Jordan, and S Shankar Sastry. On the stability of nonlinear receding horizon control: a geometric perspective. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 742–749. IEEE, 2021.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International Conference on Machine Learning*, pp. 23965–23998. PMLR, 2022.

Dante Youla, Hamid Jabr, and Jr Bongiorno. Modern wiener-hopf design of optimal controllers–part ii: The multivariable case. *IEEE Transactions on Automatic Control*, 21(3):319–338, 1976.

Andrea Zanette. Exponential lower bounds for batch reinforcement learning: Batch rl can be exponentially harder than online rl. In *International Conference on Machine Learning*, pp. 12287–12297. PMLR, 2021.

Chiyuan Zhang, Oriol Vinyals, Remi Munos, and Samy Bengio. A study on overfitting in deep reinforcement learning. *arXiv preprint arXiv:1804.06893*, 2018.