Fast Convergence for Unstable Reinforcement Learning Problems by Logarithmic Mapping

Wang Zhang [1] **Lam M. Nguyen** [2 3] **Subhro Das** [3] **Alexandre Megretski** [1] **Luca Daniel** [1] **Tsui-Wei Weng** [4]

Abstract

For many of the reinforcement learning applications, the system is assumed to be inherently stable and with bounded reward, state and action space. These are key requirements for the optimization convergence of classical reinforcement learning reward function with discount factors. Unfortunately, these assumptions do not hold true for many real world problems such as an unstable linear–quadratic regulator (LQR)[1]. In this work, we propose new methods to stabilize and speed up the convergence of unstable reinforcement learning problems with the policy gradient methods. We provide theoretical insights on the efficiency of our methods. In practice, we achieve good experimental results over multiple examples where the vanilla methods mostly fail to converge due to system instability.

1. Introduction

One of the mainstream methods to solve an RL problem is policy optimization via gradient descent. However, the convergence of the policy optimization algorithm heavily relies on an unapparent yet critical assumption of the system dynamics itself: stability[2]. In addition, to ensure the convergence of policy optimization, we also require the Lipschitz property of the cost function and its gradient. In fact, in many of the existing RL benchmark examples such as OpenAI's classical control environments, the state space/actions/costs are clipped to ensure that the policy would never move to extreme conditions and costs/states are bounded, in order to reduce the error derivatives (Mnih et al., 2015). Unfortunately, similar formulations are not directly applicable to unstable systems, such as a LQR where the spectral radius of the state matrix is outside the unit circle, as the standard policy gradient based methods are likely to fail.

To address this issue and with the aim to enable and speed up the convergence of policy gradient methods for unstable RL problems, in this work we propose a logarithmic mapping method on loss functions supported by rigorous theoretical proof and experimental results.

2. Background and Related Work

2.1. Model-free Reinforcement Learning

Distinguished by directly modelling the system dynamics/environments or not, reinforcement learning methods could be categorized into "model-based" and "model-free" approach (Arulkumaran et al., 2017). In this work, we mainly focus on the latter where the agent learns the policy by directly interacting with the system output such as rewards/costs, bypassing the inference of the underlying dynamics. By using a model-free approach, we unify the stability sources and only consider small policy variance that may show dramatically different rewards under the unstable system.

2.2. Value Function Mapping for Reinforcement Learning

In reinforcement learning, there are a few previous work mapping the value function to another space for various purposes. Van Seijen et al. (2019) used Logarithmic Q-learning to revise the action gaps due to low discount factors. (Fatemi & Tavakoli, 2022) decomposes the value function to a linear combination of a class of mapping functions to facilitate the learning process.

[1]Massachusetts Institute of Technology [2]IBM Research, Thomas J. Watson Research Center [3]MIT-IBM Watson AI Lab [4]University of California San Diego. Correspondence to: Wang Zhang <wzhang16@mit.edu>, Lam M. Nguyen <LamNguyen.MLTD@ibm.com>.

Decision Awareness in Reinforcement Learning Workshop at the 39th International Conference on Machine Learning (ICML), Baltimore, Maryland, USA, 2022. Copyright 2022 by the author(s).

[1]By unstable LQR we mean the matrix A in LQR transition Equation (1) has a spectral norm outside the unit circle.

[2]In this paper, "stability" denotes "input-to-output" stability, where a small perturbation of the input signal (control action/state perturbation) will not lead to a large deviation in the system output cost. We use model-free methods in this paper, therefore the system output target is the cost function and input is the policy. The intrinsic instability brings challenges for optimization and this paper aims to address it. For formal unstable RL definition, please refer to Appendix A

2.3. LQR Problem

For a discrete-time linear system, its state equation is represented by:

$$x_{t+1} = Ax_t + Bu_t \tag{1}$$

where $x_t \in \mathbb{R}^n$ and $u_t \in \mathbb{R}^m$ denote the system state and control action at time step t, $A \in \mathbb{R}^{n\times n}$ and $B \in \mathbb{R}^{n\times m}$ are the system transition matrices. The feedback gain is parameterized by matrix $K \in \mathbb{R}^{m\times n}$ s.t. $u_t = -Kx_t$. The intermediate cost function is in the quadratic form of state x_t and control u_t, where $Q \in \mathbb{R}^{n\times n}$ and $R \in \mathbb{R}^{m\times m}$ are given positive definite matrices to parameterize the quadratic cost. The optimal control problem can be formulated as $\underset{K}{\text{minimize}}\ \mathbb{E}_{x_0\sim P_0}\left[\sum_{t=0}^{T} x_t^\top Q x_t + u_t^\top R u_t\right]$, where, $x_t = (A-BK)^t x_0$, $u_t = -Kx_t$ and P_0 is the distribution of initial condition x_0.

Note that if $T \to \infty$, then the problem is called infinite horizon LQR, otherwise it is called finite horizon LQR. According to (Abbeel, 2012), a T-time-step system is controllable if we can reach any target state x^* from any initial state x_0. In this paper, we assume the LQR is controllable.

2.4. Convergence of Policy Gradient Methods for LQR Problems

Fazel et al. (2018) was the first to achieve the global convergence of policy gradient methods for infinite-horizon LQR problems. Bhandari & Russo (2019) extends the LQR setup to a more general class of control policies. Perdomo et al. (2021) proposes to stabilize a dynamical system with a discounted annealing algorithm by gradually increasing discount factor γ dependent on the system and current policy. In the scenario of finite horizon LQR and stochastic noise, Hambly et al. (2021) provides a global linear convergence guarantee. Tu & Recht (2018) studied Least-Squares Temporal Difference (LSTD) method on LQR and number of samples needed for LSTD estimator of value function. Nevertheless, all the above work require the assumption of the system to be *stable* under the policy throughout optimization, or equivalently, $A - BK$ has a spectral radius less than 1. Unfortunately, with an unstable A and random initialization of policy K, this assumption is mostly invalid. In contrast, our proposed algorithm target at a finite horizon setup and could still perform well without this assumption at all.

3. Proposed Methods

3.1. Finite Horizon LQR

Let $C_{K,T} = \mathbb{E}_{x_0\sim D}\left[\sum_{t=0}^{T} x_t^\top Q x_t + u_t^\top R u_t\right]$ be the expected cost of trajectory for T time steps. For unstable LQR in the infinite horizon, the cost function is not traceable since $C_{K,T} \to \infty$ when $T \to \infty$. To bypass this barrier, we focus on a finite horizon case considering the cost for first T steps. The problem formulation is: $\min_K C_{K,T}$ s.t. $x_{t+1} = Ax_t + Bu_t, u_t = -Kx_t$. In practice, infinite horizon cases are also approximated by finite step trajectories (Fazel et al., 2018) so that the implementations are identical.

3.2. Finite Horizon Unstable RL

Road-map for convergence rate analysis:
In this section, we provide a theoretical view of convergence rate bound for unstable RL problems under the vanilla setting and our proposed method. First, we formulate a Markov decision process (MDP) problem with cost formulation in Assumption 3.1, where the cost function is allowed to exponentially grow against time t with some base number ϕ. We also assume the Lipschitz property (Assumption 3.4) and local strong convexity (Assumption 3.5) of such ϕ against model parameter θ. For the optimization process, we view it as a dynamical system, translating the necessary condition for monotonic decreasing as bounding the updating step by the inverse of the spectral norm of the optimization Hessian matrix. In order to satisfy such condition for convergence, the learning rate should be bounded and therefore the convergence rate is limited. The final results are summarized in Theorem 3.8 for vanilla policy gradient and in Theorem 3.10 for our proposed logarithmic mapping. We defer the proofs and other few supportive theorems and lemmas to Appendix G.

3.2.1. Problem Formulation and Assumptions

To formulate the problem, consider a discrete-time continuous MDP $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{C}\rangle$, where $\mathcal{S}$ is the continuous state space, $\mathcal{A}$ is the continuous action space, $\mathcal{P}(s_{t+1}|s_t, a)$ is the transition probability, $c_t(s, a)$ is the immediate cost at time step t and s_0 is the initial condition. Assume that the cost is upper bounded by a polynomial of time step, s.t., $|c_t(s, a)| \le DC^t$, with positive constants $D > 0$ and $C > 0$. The target is to find an optimal policy to decrease the accumulated cost. When $C \le 1$, the cost is bounded by D independent of t and the system is I/O stable with bounded output being the common setup for RL problems. In this work, we consider a more general setting with possibility that $C > 1$.

We formulate the problem into finite time horizon of step $T \in \mathbb{Z}^+$, with accumulated cost $v_T(s, \theta)$ as a function of initial state $s \in \mathbb{R}^n$ and policy parameter $\theta \in \mathbb{R}^d$.

$$v_T(s,\theta) = \mathbb{E}_{s_{t+1}\sim p(s_t,a_t), a_t\sim\pi(s_t,\theta)}\left[\sum_{t=0}^{T}(c_t|s_0 = s)\right],$$
$$V_T(\theta) = \mathbb{E}_{s\sim\mathcal{D}}\left[v_T(s,\theta)\right].$$

where $V_T(\theta)$ is the expected cost over the initial state distribution. For a vanilla policy gradient method, we have the following update step:

$$\theta \leftarrow \theta - \eta \nabla_\theta V_T(\theta), \tag{2}$$

with $\eta > 0$. The Hessian matrix is denoted as: $J_T(\theta) = \nabla_\theta^2 V_T(\theta)$. Denote $\rho_{max}(A) = \max\{|\lambda| : \lambda \text{ is an eigenvalue of } A\}$ as the spectral radius of the state matrix A.

Assumption 3.1. Assume the step cost can be parameterized by basis function $\mathbb{E}[c_t] \sim \sum_{k=1}^m d_k \phi_k(\theta)^t$, $V_T(\theta) \sim \sum_{k=1}^m d_k \sum_{t=0}^T \phi_k(\theta)^t$, where $d_k > 0$ and $0 \leq \underline{C} < \phi_k(\theta) \leq \overline{C}$.

Remark 3.2. The motivation of such a basis function is inspired by the departing output trajectories of unstable systems. Taking the finite horizon LQR example in Section 3.1, the step cost $x_t^\top Q x_t + u_t^\top R u_t$ can be rewritten as $x_0^\top (A - BK)^{t\top} (Q + K^\top R K)(A - BK)^t x_0$, which is bounded by $\|x_0\|^2 \|Q + K^\top R K\| \|A - BK\|^{2t}$. The $\|A - BK\|^2$ term corresponds to ϕ in Assumption 3.1 with exponential growth with time step t and other terms remain positive constant. Therefore, the finite horizon LQR cost can be formulated into Assumption 3.1 regardless of stability. The formulation is also valid for optimal control problems with polynomial cost functions or other unstable RL problems with exponentially growing cost as a function of time.

Remark 3.3. Despite the basis function $\phi_k(\theta)$ plays an important role in optimization and determines the maximum step size (will be shown in Lemma G.1), their actual value and gradient properties might not be calculated or even learnable from practice. The introduction of such decomposition is used for theoretical analysis. In experiments, we select learning rates as hyperparameter for adaptation to different $\phi_k(\theta)$.

Assumption 3.4. Assume $\phi_k(\theta)$ is twice differentiable with Lipschitz constant L_2 and the gradient of $\phi_k(\theta)$ is also Lipschitz continuous with L_1, s.t.,

$$\|\nabla \phi_k(\theta_1) - \nabla \phi_k(\theta_2)\| \leq L_1 \|\theta_1 - \theta_2\|,$$
$$\|\phi_k(\theta_1) - \phi_k(\theta_2)\| \leq L_2 \|\theta_1 - \theta_2\|.$$

for $L_1, L_2 \in \mathbb{R}^+, \theta_1, \theta_2 \in \mathbb{R}^d$.

Assumption 3.5. Assume local α-strong convexity of $\phi_k(\theta)$: $\phi_k(\theta_1 + \theta_2) \geq \phi_k(\theta_1) + \theta_2^\top \nabla_\theta \phi_k(\theta_1) + \frac{\alpha}{2}\|\theta_2\|^2$. for all the $\theta_1, \theta_2 \in \mathcal{A}$ and $\theta_* = \arg\min_{\theta \in \mathcal{A}} \phi_k(\theta)$

Remark 3.6. The strong convexity assumption is indeed a "strong" one for many of the unstable RL examples and policy basis functions. The purpose of such assumption is to pave the path for convergence rate analysis. Like many other popular optimization methods in ML field such as gradient descent, we do not guarantee the algorithm's convergence to the global optimal. In practice, the stochastic gradient method could find a near-optimal result. Besides, we also assume that all the $\phi_k(\theta)$ reach local optimal with the same θ_*, since the actual cost function can be decomposed into more basis functions without the loss of generality.

3.2.2. Convergence Rate Derivation

Theorem 3.7. *Suppose $V_T(\theta)$ satisfies Assumption 3.1 and Assumption 3.4, using the vanilla gradient descent algorithm from Equation (2), if $\eta < 1/\sum_{k=1}^m d_k[L_1(\sum_{t=0}^T t\phi_k(\theta)^{t-1}) + {L_2}^2(\sum_{t=0}^T t(t-1)\phi_k(\theta)^{t-2}]$, then $\eta < 2/\rho_{max}(J_T(\theta))$ is satisfied for monotonic decrease of value function.*

By Theorem 3.7, we claim that to stabilize the convergence for the finite horizon unstable problem, the learning rate η needs to be smaller than the inverse of polynomial term of $\max(\phi_k(\theta))$, otherwise the optimization is likely to diverge.

Theorem 3.8. *Assume $\phi_k(\theta)$ is local strong convex as stated in Assumption 3.5 and the fixed learning rate η satisfies the conditions in Theorem 3.7, then if we run gradient descent for $V_T(\theta)$, it yields a solution:*

$$\|\theta_l - \theta_*\|^2 \leq q^l \|\theta_0 - \theta_*\|^2, \tag{3}$$

where $\sqrt{q}$ denotes the convergence rate and its square q is lower bounded, s.t., $q \geq (1 - \frac{2\omega^* \alpha}{\rho_{max}(J_T(\theta_0))})$, where $\omega^* = \min_\theta \sum_{k=0}^m d_k[(\sum_{t=0}^T t\phi_k(\theta)^{t-1})]$.

Our Proposed Logarithmic Mapping of Finite Horizon Value Function:

In the vanilla setup, the value function of gradient descent is $V_T(s, \theta)$. We propose a logarithmic mapping,

$$\widetilde{V}_T(s, \theta) := log(V_T(s, \theta)), \tag{4}$$

to regularize the spectral radius of gradient Hessian and gradient variance. The sampled gradient approximation has the form of $\widehat{\widetilde{V}}_T(s, \theta) := \frac{1}{b}\sum_{j=1}^b log(v_T(s_j, \theta))$.

Theorem 3.9. *Consider the parameterization of $V_T(\theta)$ and Lipschitz condition in Assumption 3.1 and Assumption 3.4, if we run gradient descent for logarithm mapped $V_T(\theta)$, then $\eta < V_T(\theta)/\sum_{k=1}^m d_k[L_1(\sum_{t=0}^T t\phi_k(\theta)^{t-1}) + {L_2}^2(\sum_{t=0}^T t(t-1)\phi_k(\theta)^{t-2}]$ is satisfied for monotonic decrease of the value function.*

Theorem 3.10. *Assume $\phi_k(\theta)$ is local strong convex as stated in Assumption 3.5 and the fixed learning rate $\eta_l < \frac{\underline{C}}{L_1 T + {L_2}^2 T(T-1)}$, then running the gradient descent for logarithm mapped $\widetilde{V}_T(\theta)$ with Equation (4) yields a solution*

$$\|\theta_{l+1} - \theta_*\|^2 \leq q_l \|\theta_l - \theta_*\|^2,$$

where the square of the step convergence rate q_l has a varying lower bound s.t. $q_l \geq (1 - \frac{2\omega^* \alpha}{\rho_{max}(J_T(\theta_l))})$.

Remark 3.11. Compared with Equation (8), the $\frac{2\omega^* \alpha}{\rho_{max}(J_T(\theta))}$ term is proportional to the inverse of current spectral radius. Considering an unstable system initialized with random policy, the initial $\rho_{max}(J_T(\theta_0))$ could be much larger than $\rho_{max}(J_T(\theta_l))$ for θ_l in the later part of the optimization. Practically, using the logarithmic mapping achieves a much faster convergence rate.

3.3. Regulating Spectral Radius as Fast Pre-processing

When an unstable system is controlled by a random policy and initialized by an arbitrary condition, the states/costs will most likely grow rapidly due to the diverging nature of the system. It is computationally costly to optimize the value function from such initial policy. To speed up the optimization, we propose a fast pre-processing method in Algorithm 1 by finding a policy close to the stable zone. Algorithm 1 is deferred to Appendix. In the pre-process, we neglect the value function and only regulate the spectral radius of the system dynamics estimated by power iteration.

4. Experiments

In this section, we use LQR as illustrative examples to demonstrate the efficacy of our methods. We also apply these methods to general unstable RL applications in continuous control and defer the results to Appendix F. For each experiment, we run 3 random seeds for reproducibility.

Figure 1 shows the vulnerability of optimization for unstable systems with large learning rates. For larger ρ_{max}, the optimization is more likely to fail with larger learning rates. However, a small learning rate could result in slow convergence. Therefore, the major challenge for vanilla policy gradient method on unstable RL problem is how to find an optimal learning rate without crashing the optimization. In our experiments, we test different learning rates by log intervals such as $\{1e\text{-}1, 1e\text{-}2, \ldots\}$ and select the largest one without breaking the optimization.

In Figure 2 we compare the vanilla gradient method and its variants with our proposed logarithmic mapping. The subplots are normalized cost difference towards optimal, normalized policy difference to the optimal and estimated spectral norm by power iteration, respectively. Note that the y-axis of the cost figure is log scale. The optimal policy $K^\star$ is the analytical optimal solution for infinite horizon LQR. When optimizing the model with vanilla sum loss function from random initialization without any pre-processing, the system starts from an extremely unstable condition with large cost. From Lemma G.1, a small learning rate is needed when the system is unstable with control policy parameters. In practice, $1e$-14 is the maximum learning rate to accommodate the instability. If we run an efficient policy pre-process with a total of 300 episodes, we could use $1e$-8 as learning rate. However, the optimization is impractical because the controller gradually stabilizes and the fixed learning rate is then too small for further parameter update. We also tested other discount factor such as $\gamma = 0.5$ to shrink the cost and allow a larger learning rate $1e$-7, but this method also fails because the small γ neglect the long term effect of the dynamics with spectral radius reaching 2. Combining our proposed pre-process and log-mapped loss function, we are able to use $1e$-2 as learning rate and both cost and policy parameters approach optimal quickly.

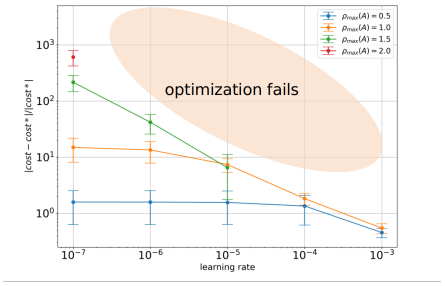

Figure 1. Vanilla PG LQR loss difference to optimal after 100 epochs under different learning rates and spectral radius (missing point for $\rho_{max} = 1.5, 2$ means the cost goes to NaN when optimization crashes)

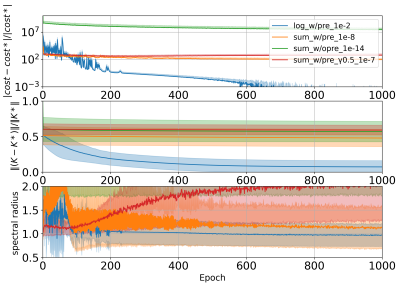

Figure 2. LQR loss difference to optimal: sum loss vs log mapping, $\rho_{max}(A) = 5$

5. Conclusion

In this paper, we focus on the gradient-based optimization for a special branch of RL problems. Due to the unstable nature of the system, small deviation leads to exponentially growing effects on the state evolving trajectory and the reward/cost function, which raised issues for gradient-based optimizations. We proposed two methods to alleviate the effect of instability and their effectiveness is validated from both theoretical and experimental points of view.

References

Abbeel, P. *Optimal Control for Linear Dynamical Systems and Quadratic Cost*. 2012.

Arulkumaran, K., Deisenroth, M. P., Brundage, M., and Bharath, A. A. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, 34(6):26–38, 2017. doi: 10.1109/MSP.2017.2743240.

Bhandari, J. and Russo, D. Global optimality guarantees for policy gradient methods. *arXiv preprint arXiv:1906.01786*, 2019.

Fatemi, M. and Tavakoli, A. Orchestrated value mapping for reinforcement learning. *arXiv preprint arXiv:2203.07171*, 2022.

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

Hambly, B., Xu, R., and Yang, H. Policy gradient methods for the noisy linear quadratic regulator over a finite horizon. *SIAM Journal on Control and Optimization*, 59(5): 3359–3391, 2021.

Lin, F., Zhang, W., and Brandt, R. Robust hovering control of a pvtol aircraft. *IEEE Transactions on Control Systems Technology*, 7(3):343–351, 1999. doi: 10.1109/87.761054.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529–533, 2015.

Perdomo, J. C., Umenberger, J., and Simchowitz, M. Stabilizing dynamical systems via policy gradient methods. *NeurIPS*, 2021.

Sontag, E. D. *Input to State Stability: Basic Concepts and Results*. 2008.

Sontag, E. D. and Wang, Y. On characterizations of the input-to-state stability property. *Systems & Control Letters*, 24(5):351–359, 1995.

Sontag, E. D. and Wang, Y. Notions of input to output stability. *Systems & Control Letters*, 38 (4):235–248, 1999. ISSN 0167-6911. doi: https://doi.org/10.1016/S0167-6911(99)00070-5. URL https://www.sciencedirect.com/science/article/pii/S0167691199000705.

Tu, S. and Recht, B. Least-Squares Temporal Difference Learning for the Linear Quadratic Regulator. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 5005–5014. PMLR, 2018.

Van Seijen, H., Fatemi, M., and Tavakoli, A. Using a logarithmic mapping to enable lower discount factors in reinforcement learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Fast Convergence for Unstable Reinforcement Learning Problems by Logarithmic Mapping
Supplementary Material, ICML 2022 Workshop

A. Formulating Instability

In the dynamical system literature, stability usually denotes the input-to-state (ISS) (Sontag & Wang, 1995) stability in system dynamics, where a small deviation of system state or control action perturbation will not lead to dramatic change of future states. Formally, consider a general continuous dynamical system $\dot{x} = f(x, u)$ with continuously differentiable $f(\cdot)$ and $x(t, x_0, u)$ denote the trajectory of x given initial condition x_0 and control feedback u. Then the system is ISS stable if there exist $\mathcal{K}$ function $\gamma : \mathbb{R}^+ \to \mathbb{R}^+$ and $\mathcal{KL}$ function $\beta : \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+$, s.t.

ISS stable:
$$\|x(t, x_0, u)\| \leq \gamma(\|u\|_\infty) + \beta(\|x_0\|, t). \tag{5}$$

where $\|u\|_\infty = sup\{\|u(t)\|\} < \infty$ for $t \geq 0$. Function $\gamma(\cdot)$ is called $\mathcal{K}$ function if $\gamma(\cdot)$ is continuously increasing and $\gamma(0) = 0$, $\beta(\cdot, \cdot)$ is called $\mathcal{KL}$ function if $\beta(\cdot, t)$ is $\mathcal{K}$ function for all the $t \geq 0$.

In this paper, we refer to stability as the input-to-output (I/O) stability (Sontag & Wang, 1999). Let the output $y = h(x(t))$ be a function of x, for instance, y can be the quadratic function to regulate the error or other system properties we wish to stabilize. The system is input-to-output (I/O) stable if there exist $\mathcal{K}$ function $\gamma(\cdot) : \mathbb{R}^+ \to \mathbb{R}^+$ and $\mathcal{KL}$ function $\beta(\cdot, \cdot) : \mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+$, s.t.

I/O stable:
$$\|y(x(t, x_0, u))\| \leq \gamma(\|u\|_\infty) + \beta(\|x_0\|, t), \tag{6}$$

Both ISS and I/O stability indicate bounded inputs leading to bounded system behavior, while there does not exist any causal relationship between the two with arbitrary choice of output function $y(\cdot)$. For further clarification, we provide a linear system example in Appendix B. In reinforcement learning setting, the agent always returns an observable "cost-to-go" but not necessarily the whole trajectory. This is because the returned cost serves as the evaluation metric for RL algorithm. Thus, I/O stability is a more suitable stability concept for analyzing RL problems and that's why it is selected for this work.

B. Example on stability definition

We use the illustrative example from (Sontag, 2008). Consider a n dimension linear system $\dot{x} = Ax + Bu$ where $A \in \mathbb{R}^{n\times n}$ being full rank matrix, $x \in \mathbb{R}^n$ with initial condition $x(0) = x_0$, $B \in \mathbb{R}^{n\times m}$ and $u = u(t) \in \mathbb{R}^m$. By solving inhomogeneous ODE, the solution is

$$x(t) = e^{At}x_0 + \int_0^t e^{A(t-\tau)}Bu(\tau)\,\mathrm{d}\tau.$$

Lemma B.1. *The system is ISS stable if all the eigenvalues of A are strictly negative.*

Proof. let $\beta(x_0, t)$ be $\|e^{At}\|\|x_0\|$ and $\gamma(x_0)$ be $\|B\| \int_0^\infty \|e^{A\tau}\|\,\mathrm{d}\tau$. With all the eigenvalues of A being strictly negative, both $\|e^{At}\|$ and $\|B\| \int_0^\infty \|e^{A\tau}\|\,\mathrm{d}\tau$ are bounded. $\|x(t, x_0, u)\| \leq \|e^{At}\|\|x_0\| + \|B\| \int_0^\infty \|e^{A\tau}\|\,\mathrm{d}\tau\|u\|_\infty$, therefore satisfies 5. □

Consider $y(x) := x$ itself, the system is I/O stable. While if we take $y(x) := \frac{1}{\|x\|}$, then the system is ISS but not I/O stable with $y \to \infty$ with $x_0, u \to 0$.

Now suppose A has non-negative eigenvalues and AB is not empty matrix, then $\int_0^t e^{A(t-\tau)}Bu(\tau)\,\mathrm{d}\tau$ is not bounded by γ function since $\|B\| \int_0^\infty \|e^{A\tau}\|\,\mathrm{d}\tau$ when $t \to \infty$, which also means the effect of previous action u will grow or at least not

vanish along the time trajectory. Then the system is not ISS. But the system could be I/O stable if we take trivial output like $y(\cdot) := 0$.

In this paper, we consider I/O stability, regardless the problem being ISS or not ISS. While in many of the real-world RL applications such as target tracking, the output function $y(\cdot)$ is correlated to the norm of x such as using distance to target as cost function. In this case, I/O stability is dependent on ISS. Specifically, for LQR problems, the eigenvalues of system matrix A determines the system ISS and also I/O stability(discrete LQR requires the eigenvalue within the unit circle and continuous LQR requires eigenvalues to left half-plane). Therefore, in the discrete LQR experiments, we use matrix A to manipulate I/O stability. Since we are dealing with I/O stability, RL scenarios with ISS but not I/O stable system is beyond the scope of this paper, for instance, a unstable invert pendulum problem with cost clipped to $[0, 1]$.

C. Algorithm for Pre-processing

Algorithm 1 Regulating system spectral norm by power iteration

Input: system state transition function f and policy θ, finite time step T, batch size b,
Initialize θ
for $l = 1$ **to** N **do**
  Sample $\{x_0^1 ... x_0^i ... x_0^b\}$
  $Loss \leftarrow 0$
  for $t = 0$ **to** T **do**
    for $i = 1$ **to** b **parallel do**
      $x_t^{i+1} \leftarrow f(x_t^i, \theta)$
      $Loss \leftarrow Loss + \text{ReLU}(\frac{\|x_T^i\|}{\|x_{T-1}^i\|} - 1)$
    end for
  end for
  $Loss \leftarrow Loss/b$
  Update parameters: $\theta \leftarrow \theta - \eta \nabla_\theta Loss$
end for

D. Comparing logarithmic Mapping with Gradient Normalization

Normalizing gradient is a classical approach to speed up the convergence, where we have the follow update step:

$$\theta \leftarrow \theta - \eta \frac{\nabla_\theta V_T(\theta)}{\|\nabla_\theta V_T(\theta)\|},$$

Compared with logarithmic mapping, the gradient normalization has similar theoretical performance in deterministic case by controlling the spectral radius of the optimization step. In the stochastic case, the updating step consists of a summation of gradient over the mini-batch followed by a normalization process. In logarithmic mapping, the log function is applied on individual examples ahead of summation. Therefore, the outliers with relatively large noise can be "normalized" to prevent them from dominating sampling summation. Besides, the portion of unstable examples with large loss are expected to drop during optimization, it is necessary to map the exponentially growing effect of these unstable cases into linear forms. In practice, our logarithmic mapping outperforms the gradient normalization in the convergence speed, as shown in the experiment section (Section 4).

Figure 3 are the comparisons between logarithmic mapping and normalizing gradient, where learning rate $1e$-1 and $1e0$ will crash the optimization respectively. The plots of $\eta = 1e$-2 for logarithmic mapping and $\eta = 1e$-1 for normalizing gradient effectively show similar convergence rate with minor fluctuation at the beginning. The logarithmic mapping eventually reaches a slightly better performance due to normalizing gradient's trapping in the local minimum. Noticeably, if both methods are coupled, the initial fluttering disappears and plots are smoother.

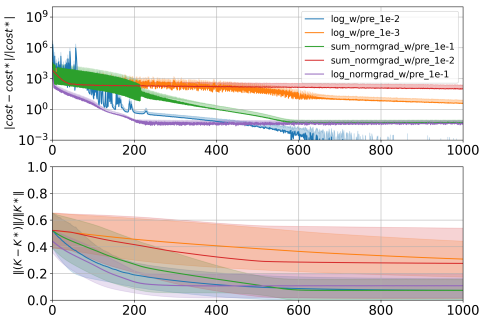

Figure 3. LQR loss difference to optimal: normalizing gradient vs log mapping, $\rho_{max}(A) = 5$

E. More experiments of unstable LQR with different spectral radius

We include additional results for unstable LQR in Figure 4 both with pre-process enabled. $\rho_{max}(A) = 2$ is a relatively moderate case, the vanilla method could use a learning rate of $\eta = 1e-4$ and slowly converge to optimal. In $\rho_{max}(A) = 10$ case, the vanilla method crashes for $\eta > 1\text{e}-11$ and the optimization stagnates for $\eta = 1\text{e}-12$. The logarithmic mapping has similar performance in $\rho_{max}(A) = 2$ case and converges faster than the latter in $\rho_{max}(A) = 10$ case.

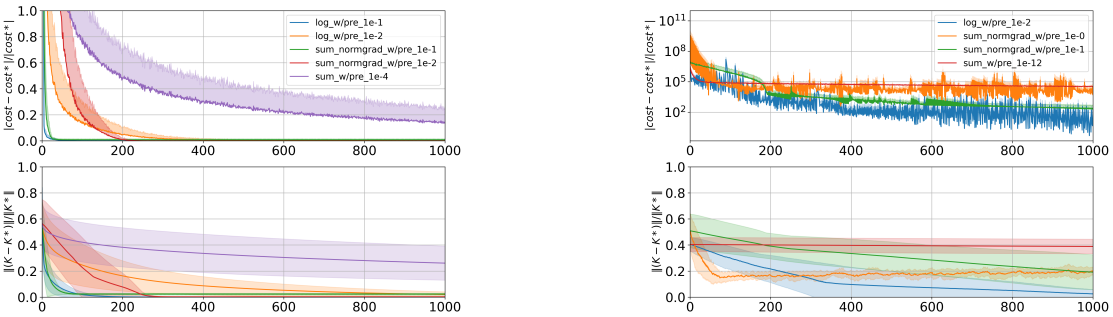

Figure 4. LQR loss difference to optimal, left: $\rho_{max}(A) = 2$, right: $\rho_{max}(A) = 10$

F. General Unstable RL

Figure 5 shows 3 customized unstable environments: unstable cart-pole, unstable mountain car and Planar Vertical Take-off and Landing (PVTOL) aircraft. We use a single hidden layer neural network with 64 hidden neurons and ReLU activation functions. The input layer and output layer has the same dimension of environment state and action space respectively. Similar to LQR experiments, we search a largest learning rate without crashing the optimization. Each experiment is performed under 3 random seeds. The lower half of variance is omitted for visualization in log-scale plots.

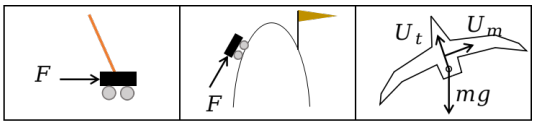

Figure 5. Unstable RL examples: modified cart-pole, modified mountain car, PVTOL aircraft

F.1. Modified cart-pole

Compared with standard cart-pole problem from OpenAI Gym package, we use a continuous force input and enlarged its force magnitude to introduce more instability to the input-output system (a small amount of control feedback could dramatically change the system behavior). Besides, we allow the agent to simulate fixed 20 time steps instead of terminating the episode if the agent runs into an undesired zone. The cost function is defined in the quadratic form of the distance between current state towards target position, instead of using the 0/1 reward depending on whether the episode is done or not.

Figure 6 shows the cost against epochs for cart-pole problem with and without pre-process. For vanilla sum loss without pre-process, $\eta = 1e\text{-}8$ is the maximum allowed learning rate and there is a significant difference in convergence speed compared with other two. The logarithmic mapped cost is higher but close to sum loss with normalizing gradient. With a pre-processed policy, the system is more stable at the beginning and therefore larger learning rates are allowed. All three methods could reach the optimal. To remark on the cart-pole problem, the instability mostly comes from the large force magnitude instead of the unbounded state space because there exists local equilibrium when the pole sticks downward. Therefore, compared to the following 2 environments, it is less challenging and could be addressed with vanilla sum loss with a simple pre-process.

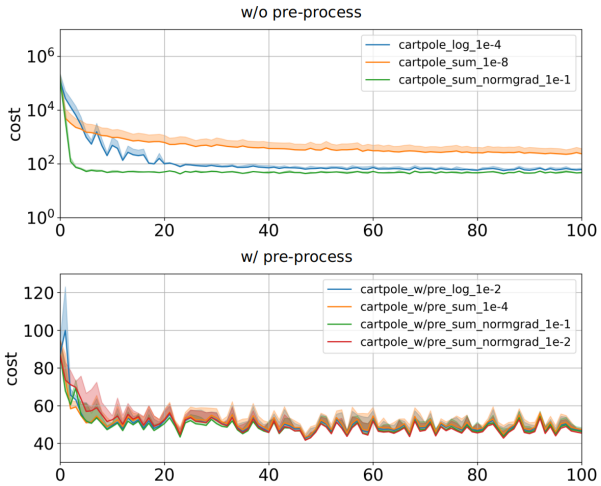

Figure 6. Unstable cart-pole

F.2. Modified mountain car

Similar to the cart-pole treatment, we remove the terminal conditions and re-define the cost function in the quadratic form. The control target is to drive the car to a certain location and stabilize it. We manipulate a steep slope by adding an acceleration term proportional to the cube of horizontal displacement from the peak and there does not exist any local equilibrium point.

The results are shown in Figure 7. Both vanilla sum loss and normalizing gradient require small learning rate, the logarithmic mapping outperforms the other two methods. When pre-processing is engaged, the vanilla sum loss still converges slowly, the other two methods share similar performance and achieve a smaller cost compared with the optimal results without the pre-process.

F.3. PVTOL Aircraft

The Planar Vertical Take-off and Landing (PVTOL) aircraft (Lin et al., 1999) is a simplified 2D model of realistic aircraft maneuver. The aircraft state includes the lateral/vertical displacement of the gravity center and roll angle. The control feedback U_t and U_m are longitudinal thrust and lateral rolling force. Notice U_m provides both force and rolling moment to the airplane. The target is to control and airplane to certain state and cost function is also in the quadratic form.

Similar to the unstable mountain car example, both vanilla sum loss and normalizing gradient show slow convergence when pre-process is not engaged. The logarithmic mapping is capable to achieve optimal results regardless of the pre-treatment.

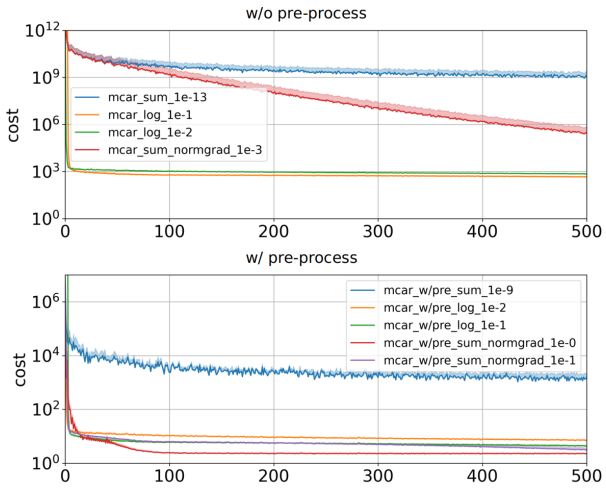

Figure 7. Unstable mountain car

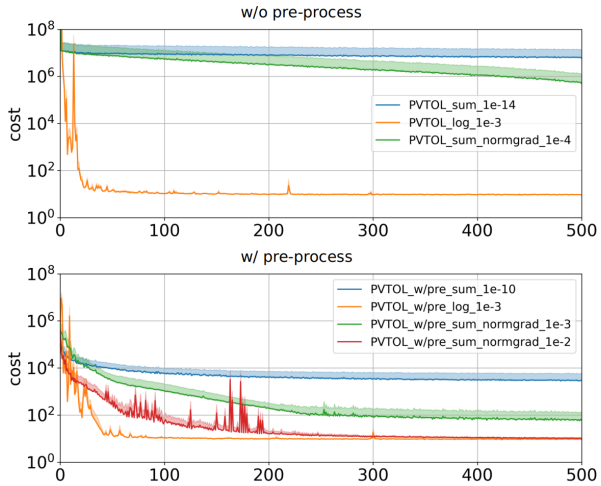

Figure 8. PVTOL

G. More theoretical results on unstable RL

Lemma G.1. *Update the value function $V_T(\theta)$ by policy gradient method with $\theta \leftarrow \theta - \eta \nabla_\theta V_T(\theta)$, choose step size $\eta < 2/\max_\xi \rho_{max}(J_T(\theta + \xi\eta\nabla J_T(\theta)))$ for $\xi \in [0,1]$, then $V_T(\theta)$ is monotonically decreasing.*

Proof. Let $\xi \in [0,1]$ be a scalar, denote $s = -\eta\nabla V_T(\theta), g(\xi) = V_T(\theta + \xi s)$, we have

$$
\begin{aligned}
V_T(\theta + \xi s) - V_T(\theta) = g(1) - g(0) &= \int_0^1 \frac{dg}{d\xi}\, d\xi \\
&= \int_0^1 s^\top \nabla V_T(\theta + \xi s)\, d\xi \\
&\leq \int_0^1 s^\top \nabla V_T(\theta)\, d\xi + |\int_0^1 s^\top (\nabla V_T(\theta) - \nabla V_T(\theta + \xi s))\, d\xi| \\
&\leq s^\top \nabla V_T(\theta) + \int_0^1 \|s\| \|(\nabla V_T(\theta) - \nabla V_T(\theta + \xi s))\|\, d\xi \\
&\leq s^\top \nabla V_T(\theta) + \|s\|^2 \max_\xi \rho_{max}(J_T(\theta + \xi\eta s))/2.
\end{aligned}
$$

substitute $s = \eta \nabla V_T(\theta)$ into the equation, we have

$$V_T(\theta + \xi s) - V_T(\theta) \leq -\eta(1 - \frac{\eta}{2}\max_{\xi} \rho_{max}(J_T(\theta + \xi\eta\nabla J_T(\theta))))\|\nabla V_T(\theta)\|^2 < 0.$$

when $(1 - \frac{\eta}{2}\max_{\xi} \rho_{max}(J_T(\theta + \eta\nabla J_T(\theta))))$ term is negative.

□

Theorem G.2. *If $V_T(\theta)$ satisfies Assumption 3.1 and Assumption 3.4, using the vanilla gradient descent algorithm from Equation (2), then $\rho_{max}(J_T(\theta)) < \sum_{k=1}^{m} d_k[L_1(\sum_{t=0}^{T} t\phi_k(\theta)^{t-1}) + {L_2}^2(\sum_{t=0}^{T} t(t-1)\phi_k(\theta)^{t-2}]$*

Proof.

$$\begin{aligned}
\nabla_\theta V_T(\theta) &= \sum_{k=1}^{m} d_k(\sum_{t=0}^{T} t\phi_k(\theta)^{t-1})\frac{\partial \phi_k(\theta)}{\partial \theta}, \\
\text{Hessian } J_T(\theta) &= \nabla_\theta^2 V_T(\theta) \\
&= \sum_{k=1}^{m} J_T^k(\theta).
\end{aligned}$$

where

$$\begin{aligned}
J_T^k(\theta) = d_k[&(\sum_{t=0}^{T} t\phi_k(\theta)^{t-1})\frac{\partial^2 \phi_k(\theta)}{\partial \theta^2} \\
&+ (\sum_{t=0}^{T} t(t-1)\phi_k(\theta)^{t-2})\frac{\partial \phi_k(\theta)}{\partial \theta}\frac{\partial \phi_k(\theta)}{\partial \theta}^\top].
\end{aligned} \tag{7}$$

Denote the eigenvalues of $\frac{\partial^2 \phi_k(\theta)}{\partial \theta^2}$, $\frac{\partial \phi_k(\theta)}{\partial \theta}\frac{\partial \phi_k(\theta)}{\partial \theta}^\top$, $J_T^k(\theta)$, as $\mu_1^k > ... > \mu_n^k$, $\upsilon_1^k > ... > \upsilon_n^k$, $\nu_1^k > ... > \nu_n^k$ respectively. Notice $\frac{\partial \phi_k(\theta)}{\partial \theta}\frac{\partial \phi_k(\theta)}{\partial \theta}^\top$ is positive semi-definite and has same non-zero eigenvalue with $\frac{\partial \phi_k(\theta)}{\partial \theta}^\top\frac{\partial \phi_k(\theta)}{\partial \theta}$, then $\upsilon_1^k \leq {L_2}^2$ by Lipschitz condition. To bound the eigenvalues of $\frac{\partial^2 \phi_k(\theta)}{\partial \theta^2}$,

$$\begin{aligned}
|\mu_i^k| &\leq \|\frac{\partial^2 \phi_k(\theta)}{\partial \theta^2} v\|/\|v\| \\
&= \lim_{h \to 0} \frac{\|\nabla \phi_k(\theta + hv) - \nabla \phi_k(\theta)\|}{|h|\|v\|} \\
&\leq \frac{L_1\|hv\|}{|h|\|v\|} \\
&\leq L_1.
\end{aligned}$$

Notice $\frac{\partial^2 \phi_k(\theta)}{\partial \theta^2}$ and $\frac{\partial \phi_k(\theta)}{\partial \theta}\frac{\partial \phi_k(\theta)}{\partial \theta}^\top$ are Hermitian, by Weyl's inequality to bound:

$$\begin{aligned}
\nu_1^k &\leq d_k[L_1(\sum_{t=0}^{T} t\phi_k(\theta)^{t-1}) + {L_2}^2(\sum_{t=0}^{T} t(t-1)\phi_k(\theta)^{t-2})], \\
\nu_n^k &\geq d_k[-L_1(\sum_{t=0}^{T} t\phi_k(\theta)^{t-1})].
\end{aligned}$$

$$\rho_{max}(J_T^k(\theta)) = \max(|\nu_1^k|, |\nu_n^k|)$$

$$\leq d_k[L_1(\sum_{t=0}^{T} t\phi_k(\theta)^{t-1}) + {L_2}^2(\sum_{t=0}^{T} t(t-1)\phi_k(\theta)^{t-2})].$$

$$\rho_{max}(J_T(\theta)) \leq \sum_{k=1}^{m} d_k[L_1(\sum_{t=0}^{T} t\phi_k(\theta)^{t-1}) + {L_2}^2(\sum_{t=0}^{T} t(t-1)\phi_k(\theta)^{t-2})].$$

□

Lemma G.3. *If function $f(x) : \mathbb{R}^m \to \mathbb{R}+$ is L_1 smooth and L_2 Lipschitz and non-negative for $x \in \mathbb{S} \subset \mathbb{R}^m$, then its polynomial $f(x)^n$ is $(n\overline{f_{\mathbb{S}}}^{n-1}L_1 + n(n-1)\overline{f_{\mathbb{S}}}^{n-2}{L_2}^2)$ smooth on $\mathbb{S}$, where $\overline{f_{\mathbb{S}}} = \max_{x\in\mathbb{S}}[f(x)]$*

Proof. With function $f(x)$ being L_1 smooth, equivalently

$$\begin{aligned}
&\|\nabla f(x) - \nabla f(y)\| \leq L_1\|x-y\| \\
\Longleftrightarrow &g(x) = \frac{L_1}{2}x^\top x - f(x) \text{ is convex} \\
\Longleftrightarrow &L_1 I \succeq \frac{\partial^2 f(x)}{\partial x^2}.
\end{aligned}$$

With function $f(x)$ being L_2 Lipschitz, $\|\nabla f(x)\| \leq L_2$,

for polynomial $f(x)^n$,

$$\begin{aligned}
\frac{\partial^2[f(x)^n]}{\partial x^2} &= nf(x)^{n-1}\frac{\partial^2 f(x)}{\partial x^2} + n(n-1)f(x)^{n-2}\nabla f(x)\nabla f(x)^\top \\
&\preceq (n\overline{f_{\mathbb{S}}}^{n-1}L_1 + n(n-1)\overline{f_{\mathbb{S}}}^{n-2}{L_2}^2)I.
\end{aligned}$$

where the ${L_2}^2$ term comes from the fact that $\nabla f(x)\nabla f(x)^\top$ has same non-zero eigenvalue with $\nabla f(x)^\top\nabla f(x)$.

Therefore, $f(x)^n$ is locally $(n\overline{f_{\mathbb{S}}}^{n-1}L_1 + n(n-1)\overline{f_{\mathbb{S}}}^{n-2}{L_2}^2)$ smooth on the support $\mathbb{S}$. □

Lemma G.4.

$$\frac{V_T(\theta) - V_T(\theta_*)}{\|\nabla_\theta V_T(\theta)\|^2} \geq \frac{1}{2L'}.$$

where

$$L' = \sum_{k=1}^{m} d_k \sum_{t=0}^{T}[t\overline{\phi_k}(\theta)^{t-1}L_1 + t(t-1)\overline{\phi_k}(\theta)^{t-2}{L_2}^2].$$

where

$$\overline{\phi_k}(\theta) = \max_{\xi}[\phi_k(\theta_* + \xi(\theta - \theta_*))] \textit{ for } \xi \in [0,1].$$

Proof. With Assumption 3.4 on $\phi_k(\theta)$, apply Lemma G.3 on the straight line from θ to θ_*, $V_T(\theta) \sim \sum_{k=1}^{m} d_k \sum_{t=0}^{T} \phi_k(\theta)^t$ is L' smooth on the straight line.

$$\begin{aligned}
V_T(\theta_*) &\le \min_{\xi} V_T(\theta - \xi \nabla V_T(\theta)) \\
&\le \min_{\xi} [V_T(\theta) - \xi \|\nabla V_T(\theta)\|^2 + \frac{L'}{2}\xi^2 \|\nabla V_T(\theta)\|^2] \\
&\le \min_{\xi} [V_T(\theta) + \|\nabla V_T(\theta)\|^2 (\frac{L'}{2}(\xi - \frac{1}{L'})^2 - \frac{1}{2L'})] \\
&\le V_T(\theta) - \frac{1}{2L'} \|\nabla V_T(\theta)\|^2.
\end{aligned}$$

where second inequality comes from the L' smoothness on the straight line from θ to θ_*.
therefore,

$$\frac{V_T(\theta) - V_T(\theta_*)}{\|\nabla_\theta V_T(\theta)\|^2} \ge \frac{1}{2L'}.$$

□

Remark G.5. If $\phi_k(\theta_* + \xi(\theta - \theta_*))$ is monotonically increasing on ξ, then

$$\begin{aligned}
\overline{\phi_k}(\theta) &\le \phi_k(\theta), \\
L' &\le \sum_{k=1}^{m} d_k \sum_{t=0}^{T} [\phi_k(\theta)^{t-1} L_1 + t(t-1)\phi_k(\theta)^{t-2} {L_2}^2].
\end{aligned}$$

then the smoothness along the updated step is bounded by the spectral radius of the Hessian on θ, as $\rho_{max}(J_T(\theta))$ in Theorem G.2.

Proposition G.6. *For twice-differentiable f, f is α-strong convexity function if and only if $\nabla^2 f(x) \succcurlyeq \alpha I$ for some $\alpha > 0$ and $x \in \mathbb{R}^d$.*

Proof. For a twice-differentiable function, α-strong convexity is equivalent to the smallest eigenvalue of Hessian of f being lower bounded by α. □

Theorem 3.7 (Restated)**.** *Suppose $V_T(\theta)$ satisfies Assumption 3.1 and Assumption 3.4, using the vanilla gradient descent algorithm from Equation (2), if $\eta < 1/\sum_{k=1}^{m} d_k[L_1(\sum_{t=0}^{T} t\phi_k(\theta)^{t-1}) + {L_2}^2(\sum_{t=0}^{T} t(t-1)\phi_k(\theta)^{t-2}]$, then $\eta < 2/\rho_{max}(J_T(\theta))$ is satisfied for monotonic decrease of value function.*

The proof is completed by substituting Theorem G.2 into Lemma G.1 and taking $\xi = 0$.

Theorem 3.8 (Restated)**.** *Assume $\phi_k(\theta)$ is local strong convex as stated in Assumption 3.5 and the fixed learning rate η satisfies the conditions in Theorem 3.7, then if we run gradient descent for $V_T(\theta)$, it yields a solution:*

$$\|\theta_l - \theta_*\|^2 \le q^l \|\theta_0 - \theta_*\|^2, \tag{8}$$

where $\omega^* = \min_{\theta} \sum_{k=0}^{m} d_k[(\sum_{t=0}^{T} t\phi_k(\theta)^{t-1})]$, $\sqrt{q}$ denotes the convergence rate and its square q is lower bounded s.t. $q \ge (1 - \frac{2\omega^*\alpha}{\rho_{max}(J_T(\theta_0))})$

Proof. by Proposition G.6, $\frac{\partial^2 \phi_k(\theta)}{\partial \theta^2} \succcurlyeq \alpha I$. From (7),

$$\begin{aligned}
J_T^k(\theta) &\succcurlyeq d_k[(\sum_{t=0}^{T} t\phi_k(\theta)^{t-1})]\alpha I, \\
J_T(\theta) &\succcurlyeq \sum_{k=0}^{m} d_k[(\sum_{t=0}^{T} t\phi_k(\theta)^{t-1})]\alpha I \\
&\succcurlyeq \min_{\theta} \sum_{k=0}^{m} d_k[(\sum_{t=0}^{T} t\phi_k(\theta)^{t-1})]\alpha I = \omega^* \alpha I.
\end{aligned}$$

by Proposition G.6, $V_T(\theta)$ is $\omega^*\alpha$ strong convex:

$$V_T(\theta_1+\theta_2) \geq V_T(\theta_1) + \theta_2^\top \nabla_\theta V_T(\theta_1) + \frac{\omega^*\alpha}{2}\|\theta_2\|^2. \tag{9}$$

$$\begin{aligned}
&\|\theta_{l+1}-\theta_*\|^2 \\
&= \|\theta_l - \eta_l \nabla_\theta V_T(\theta_l) - \theta_*\|^2 \\
&= \|\theta_l-\theta_*\|^2 - 2\eta_l \nabla_\theta V_T(\theta_l)^\top(\theta_l-\theta_*) + \eta_l^2\|\nabla_\theta V_T(\theta_l)\|^2 \\
&\overset{(9)}{\leq} \|\theta_l-\theta_*\|^2(1-\eta_l\omega^*\alpha) - 2\eta_l(V_T(\theta_l)-V_T(\theta_*)) \\
&\quad + \eta_l^2\|\nabla_\theta V_T(\theta_l)\|^2 \\
&\leq \|\theta_l-\theta_*\|^2(1-\eta_l\omega^*\alpha) \text{ when } \eta_l < 2\frac{V_T(\theta_l)-V_T(\theta_*)}{\|\nabla_\theta V_T(\theta_l)\|^2}
\end{aligned} \tag{10}$$

where the inequality condition is satisfied with our analysis in Lemma G.4 and Remark G.5 when

$$\begin{aligned}
\eta_l &< 1/\sum_{k=1}^m d_k[L_1(\sum_{t=0}^T t\phi_k(\theta_l)^{t-1}) + {L_2}^2(\sum_{t=0}^T t(t-1)\phi_k(\theta_l)^{t-2}] \text{ (conditions in Theorem 3.7)} \\
&\leq \frac{1}{L'} \\
&\leq 2\frac{V_T(\theta_l)-V_T(\theta_*)}{\|\nabla_\theta V_T(\theta_l)\|^2}.
\end{aligned}$$

Since we have a non-increasing step size, η_l is upper bounded by $\frac{2}{\rho_{max}(J_T(\theta_0))}$, $(1-\eta_l\omega^*\alpha)$ is greater than $(1-\frac{2\omega^*\alpha}{\rho_{max}(J_T(\theta_0))})$.

□

Theorem 3.10 (Restated)**.** *Assume $\phi_k(\theta)$ is local strong convex as stated in Assumption 3.5 and the fixed learning rate $\eta_l < \frac{C}{L_1T+{L_2}^2T(T-1)}$, then if we run gradient descent for logarithmic mapped $V_T(\theta)$ with Equation (4), it yields a solution:*

$$\|\theta_{l+1}-\theta_*\|^2 \leq q_l\|\theta_l-\theta_*\|^2.$$

where the the square of the step convergence rate q_l has a varying lower bound s.t. $q_l \geq (1-\frac{2\omega^*\alpha}{\rho_{max}(J_T(\theta_l))})$

Proof.

$$\begin{aligned}
&\|\theta_{l+1}-\theta_*\|^2 \\
&= \|\theta_l - \frac{\eta_l}{V_T(\theta_l)}\nabla_\theta V_T(\theta_l) - \theta_*\|^2 \\
&= \|\theta_l-\theta_*\|^2 - 2\frac{\eta_l}{V_T(\theta_l)}\nabla_\theta V_T(\theta_l)^\top(\theta_l-\theta_*) + \frac{\eta_l}{V_T(\theta_l)}^2\|\nabla_\theta V_T(\theta_l)\|^2 \\
&\leq \|\theta_l-\theta_*\|^2(1-\frac{\eta_l}{V_T(\theta_l)}\omega^*\alpha) - 2\frac{\eta_l}{V_T(\theta_l)}(V_T(\theta)-V_T(\theta_*)) + \frac{\eta_l}{V_T(\theta_l)}^2\|\nabla_\theta V_T(\theta_l)\|^2 \\
&\leq \|\theta_l-\theta_*\|^2(1-\frac{\eta_l}{V_T(\theta_l)}\omega^*\alpha) \text{ when } \frac{\eta_l}{V_T(\theta_l)} \leq 2\frac{V_T(\theta_l)-V_T(\theta_*)}{\|\nabla_\theta V_T(\theta_l)\|^2}
\end{aligned}$$

where the inequality condition is satisfied with our analysis in Lemma G.4 and Remark G.5 when $\frac{\eta_l}{V_T(\theta_l)} < 1/\sum_{k=1}^m d_k[L_1(\sum_{t=0}^T t\phi_k(\theta_l)^{t-1}) + {L_2}^2(\sum_{t=0}^T t(t-1)\phi_k(\theta_l)^{t-2}]$. The latter is valid because

$V_T(\theta_l)/\sum_{k=1}^m d_k[L_1(\sum_{t=0}^T t\phi_k(\theta_l)^{t-1}) + {L_2}^2(\sum_{t=0}^T t(t-1)\phi_k(\theta_l)^{t-2}]$ is lower bounded by a constant

$$\begin{aligned}
&\min_l \frac{V_T(\theta_l)}{\sum_{k=1}^m d_k[L_1(\sum_{t=0}^T t\phi_k(\theta_l)^{t-1}) + {L_2}^2(\sum_{t=0}^T t(t-1)\phi_k(\theta_l)^{t-2}]} \\
&= \min_l \frac{\sum_{k=1}^m d_k[(\sum_{t=0}^T \phi_k(\theta_l)^t)}{\sum_{k=1}^m d_k[L_1(\sum_{t=0}^T t\phi_k(\theta_l)^{t-1}) + {L_2}^2(\sum_{t=0}^T t(t-1)\phi_k(\theta_l)^{t-2})} \\
&\geq \min_l \min_k \frac{\phi_k(\theta_l)}{L_1 T + {L_2}^2 T(T-1)} \\
&> \frac{\underline{C}}{L_1 T + {L_2}^2 T(T-1)} > \eta_l.
\end{aligned}$$

Because

$$\begin{aligned}
\frac{\eta_l}{V_T(\theta_l)} &< 1/\sum_{k=1}^m d_k[L_1(\sum_{t=0}^T t\phi_k(\theta_l)^{t-1}) + {L_2}^2(\sum_{t=0}^T t(t-1)\phi_k(\theta_l)^{t-2}] \\
&< \frac{2}{\rho_{max}(J_T(\theta_l))}.
\end{aligned} \tag{11}$$

we have $(1 - \frac{\eta_l}{V_T(\theta_l)}\omega^*\alpha) > (1 - \frac{2\omega^*\alpha}{\rho_{max}(J_T(\theta_l))})$. □