Learning-to-Learn to Guide Random Search: Derivative-Free Meta Blackbox Optimization on Manifold

Bilgehan Sel BSEL@VT.EDU
Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24061, USA

Ahmad Al-Tawaha ATAWAHA@VT.EDU
Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24061, USA

Yuhao Ding YUHAO_DING@BERKELEY.EDU
Industrial Engineering and Operations Research, University of California, Berkeley, CA 94720, USA

Ruoxi Jia RUOXIJIA@VT.EDU
Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24061, USA

Bo Ji BOJI@VT.EDU
Computer Science, Virginia Tech, Blacksburg, VA 24061, USA

Javad Lavaei LAVAEI@BERKELEY.EDU
Industrial Engineering and Operations Research, University of California, Berkeley, CA 94720, USA

Ming Jin* JINMING@VT.EDU
Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24061, USA

Abstract

Solving a sequence of high-dimensional, nonconvex, but potentially similar optimization problems poses a computational challenge in engineering applications. We propose the *first* meta-learning framework that leverages the shared structure among sequential tasks to improve the computational efficiency and sample complexity of derivative-free optimization. Based on the observation that most practical high-dimensional functions lie on a latent low-dimensional manifold, which can be further shared among instances, our method jointly learns the meta-initialization of a search point and a meta-manifold. Theoretically, we establish the benefit of meta-learning in this challenging setting. Empirically, we demonstrate the effectiveness of the proposed algorithm in two high-dimensional reinforcement learning tasks.

Keywords: meta-learning, derivative-free optimization, manifold learning

1. Introduction

Solving a sequence of optimization problems with similar structures often arises in engineering applications. For instance, in time-varying optimization, the problem usually has a fixed structure, but its instantiations depend on time-varying data obtained at predefined sampling times (Zavala and Anitescu, 2010). Real-time computation of the solution is often required for problems in power systems (Dall'Anese et al., 2017; Hauswirth et al., 2018; Tang and Low, 2017; Ding et al., 2021), communication systems (Chen and Lau, 2011; Low and Lapsley, 1999), online learning (Mokhtari et al., 2016; Yang et al., 2016), and signal processing (Balavoine et al., 2015). Similarly, in multi-

* Corresponding author

task learning, multiple loss functions corresponding to similar tasks are jointly optimized for improved generalization performance (Zhang and Yang, 2021). Despite advances in computing, resolving each instance of single-task optimization separately can be impractical. For example, a reinforcement learning (RL) algorithm can potentially take millions of steps before converging to a good policy from scratch, which is restrictive in real-world settings (Dulac-Arnold et al., 2021).

A promising direction, as exemplified by meta-learning, or learning-to-learn, is to leverage prior and similar experiences to accelerate optimization (Hospedales et al., 2021). Notwithstanding the empirical success (Finn et al., 2017), most efforts to analyze initialization-based meta-learning focus on the setting with decomposable single-task loss functions that are often convex for theoretical tractability (Finn et al., 2019; Denevi et al., 2019; Balcan et al., 2019); nonconvex single-task settings are studied usually for multi-task representation learning (Balcan et al., 2015; Maurer et al., 2016; Du et al., 2020; Tripuraneni et al., 2020). Therefore, the first challenge is *to analyze the theoretical benefits of meta-learning for nonconvex optimizations with shared structures*.

Furthermore, while gradient-based meta-learning has gained traction, there are many applications such as hyperparameter tuning (Real et al., 2017), reinforcement learning (Kirsch et al., 2022), simulation-based optimization (Gosavi et al., 2015), and generating adversarial examples (Chen et al., 2017), which resist access to the gradient, let alone the Hessian of the objective function. In general, derivative-free optimization (DFO) has been employed for blackbox optimizations (Larson et al., 2019). However, contrary to first-order techniques, where the convergence rates are independent of the problem dimensionality, DFO methods, such as Bayesian optimization (Snoek et al., 2012) and random search (Mania et al., 2018), often scale poorly with the dimensionality (Ghadimi and Lan, 2013). Adaptivity to the latent low-dimensional structure of the search space has been pursued by subspace methods (Wang et al., 2016; Choromanski et al., 2019). In particular, Sener and Koltun (2020) proposed the learned manifold random search (LMRS), which learns a latent manifold while performing the optimization. The second challenge, therefore, is *to enable meta-learning of DFO that exploits the latent structure of a potentially high-dimensional problem*.[1]

In view of the aforementioned challenges, we propose the *first* framework that leverages the shared structure among potentially high-dimensional, nonconvex, but similar problem instances to improve computational efficiency and sample complexity of repeated application of DFO. Specifically, using LMRS as an exemplary base algorithm, we develop Meta-LMRS that adaptively and jointly learns a meta-initialization of a search point and a meta-manifold. Preliminary results on two high-dimensional RL problems demonstrate that Meta-LMRS facilitates each task to be solved with a small handful of iterations. For analysis, we introduce two notions of similarity among optimization tasks, namely V^*_{init} and V^*_{manifold}, which measure closeness based on the initial point and optimization landscape as captured by some shared manifold, respectively. We establish a key theoretical benefit of the proposed Meta-LMRS. Notably, the task-averaged regret on stationarity (Def. 3) can be bounded as $\mathcal{O}\left(M^{-\frac{1}{2}} + \max\left(V^*_{\text{init}}, c + V^*_{\text{manifold}}\right)\right)$, where M is the number of tasks and c is some constant. This bound improves upon single-task learning when M is large enough or V^*_{init} and V^*_{manifold} are small enough (the tasks are sufficiently similar). To contextualize the contribution, our technique can also be viewed as a step forward for semi-amortized models over a latent space without requiring gradients (Amos, 2022, Sec. 3.2.2).

1. Note that here we aim to learn a low-dimensional manifold to improve sample complexity or computational efficiency, which is conceptually different from the classical field of manifold optimization (Absil et al., 2009), where the manifold is formed by predetermined constraints.

The rest of the paper is organized as follows. Sec. 2 introduces the problem formulation; we also include a recapitulation of LMRS along with tailored bounds to facilitate subsequent development. The proposed method is presented in Sec. 3 and evaluated in Sec. 4. Conclusion is drawn in Sec. 5. Due to space limitations, we provide all the proofs and additional experimentation details in this online document (Sel et al., 2022).

2. Problem Formulation and Preliminaries

2.1. Problem setup

Consider an online sequence of tasks $\{\mathcal{T}_i\}_{i=1}^M$ that arrive sequentially. Each task $\mathcal{T}_i$ is to solve a high-dimensional stochastic optimization problem of the form

$$\min_{x \in \mathbb{R}^d} f_i(x) = \mathbb{E}_\xi[F_i(x, \xi)],$$

where x is the optimization variable and $f_i : \mathbb{R}^d \to \mathbb{R}$ is the function of task i, which is defined as expectation over some noise variable ξ. In DFO, instead of evaluating the gradients, we only have zeroth-order access to the objective function through the sampling operator $\square$, i.e., $\square f_i(x) \sim F_i(x, \xi)$ is a random variable for the input x and some noise variable ξ.

Our goal is to learn meta-parameters $\theta \in \Theta$ that produce a good task-specific solution x_i after a few steps of random search. In particular, let $\mathcal{A}lg^t(\theta, \square f_i)$ corresponds to performing t steps of DFO initialized with θ. For example, if one step of random search is taken and θ corresponds to the initial point, we have $x_i \equiv \mathcal{A}lg^1(\theta, \square f_i) = \theta - \alpha \hat{g}_i$, where α is the stepsize and $\hat{g}_i$ is the estimated gradient. To estimate the gradient using function evaluations, recall the classical result by Flaxman et al. (2004); Nesterov and Spokoiny (2017). Let $\mathbb{S}^{d-1}$ and $\mathbb{B}^d$ denote the d-dimensional unit sphere and unit ball, and ω be a random vector sampled from the uniform distribution over $\mathbb{S}^{d-1}$. For a function $f : \mathbb{R}^d \to \mathbb{R}$, its δ-smoothed version is $\hat{f}(x) = \mathbb{E}_{v \sim \mathbb{B}^d}[f(x + \delta v)]$. Then,

$$\hat{g}(x, \omega) = (F(x + \delta\omega, \xi) - F(x - \delta\omega, \xi))\, \omega \tag{1}$$

is an unbiased estimate of the gradient of the smoothed function $\mathbb{E}_{\xi, \omega \in \mathbb{S}^{d-1}}[\hat{g}(x, \omega)] = \frac{2\delta}{d} \nabla_x \hat{f}(x)$.

The goal of gradient-based meta-learning is to find an inductive bias that enables solving a new instance of optimization f with a few steps of adaptation (t is small) by an DFO $\mathcal{A}lg^t(\theta, \square f)$.

Notations. We use $\|\cdot\|_1$, $\|\cdot\|_2$, and $\|\cdot\|_\infty$ to denote ℓ_1, ℓ_2, and ℓ_∞-norms, respectively. We use the shorthand notation $[M] = \{1, ..., M\}$.

2.2. Assumptions

We assume that the stochastic function is bounded ($|F(x, \xi)| \leq \Omega$), L-Lipschitz, and μ-smooth with respect to x for all ξ, and has uniformly bounded variance ($\mathbb{E}_\xi[(F(x, \xi) - f(x))^2] \leq V_F$). Furthermore, we assume that given ξ, $F(\cdot, \xi)$ lies on an n-dimensional manifold (e.g., $n \ll d$) and this manifold can be defined via a nonlinear parametric family (e.g. a neural network): $F(x, \xi) = h(r(x; \psi_r^\star); \psi_h^\star)$ for all $x \in \mathbb{R}^d$. The above assumptions are adopted by Sener and Koltun (2020). For simplicity, we denote $\psi = (\psi_r, \psi_h) \in \Psi$, where Ψ is assumed to have a bounded ℓ_∞ diameter: $D_\Psi = \sup_{\psi, \psi' \in \Psi} \|\psi - \psi'\|_\infty$. Unless otherwise specified, we assume the above for each task.

2.3. Learning to guide random search in a single task

Based on the observations that the function of interest in many practical problems lies on a low-dimensional nonlinear manifold, Sener and Koltun (2020) proposed LMRS to minimize the function and learn the manifold jointly. For completeness, we provide a succinct review of LMRS (see Algorithm 2 for the pseudo-code).

Random search over a manifold. Denote $f_\psi = h(r(\cdot;\psi_r);\psi_h)$, then, by the chain rule, we have that $\nabla_x f_\psi(x) = J(x;\psi_r)\nabla_{\hat{r}} h(\hat{r};\psi_h)$, where $J(x;\psi_r) = {\partial r(x;\psi_r)}/{\partial x}$ and $\hat{r} = r(x,\psi_r)$. LMRS first orthonormalizes the Jacobian $J(x;\psi_r)$ using the Gram–Schmidt procedure for numerical stability and then performs the random search in the column space of this orthonormal matrix $J_q(x;\psi_r)$, which has lower dimensionality than the full space. For each step, an n-dimensional vector $\tilde{\omega}$ is sampled uniformly from $\mathbb{S}^{n-1}$ and lifted to the input space via $J_q(x;\psi_r)\tilde{\omega}$. Then, by the manifold Stokes' theorem, using the lifted vector as a random direction gives an unbiased estimate of the gradient of the smoothed function as (c.f., (1)):

$$\mathbb{E}_{\xi,\tilde{\omega}\in\mathbb{S}^{n-1}}[\hat{g}(x, J_q(x;\psi_r)\tilde{\omega})] = \frac{2\delta}{n}\nabla_x \tilde{f}_\psi(x),$$

where $\tilde{f}_\psi(x) = \mathbb{E}_{\tilde{v}\sim\mathbb{B}^n}[f(x+\delta J_q(x;\psi_r)\tilde{v})]$ is the smoothed function. In each iteration, exploration is added by sampling directions from the manifold g_m and the full space g_e, which are mixed to obtain the final estimate $g = (1-\beta)g_m + \beta g_e$.

Manifold learning. At each iteration, we observe $y(x^t,\omega^t) = f(x^t + \delta J_q(x^t;\psi_r^t)\omega^t) - f(x^t - \delta J_q(x^t;\psi_r^t)\omega^t)$, which is the projection of the gradient onto the chosen directions. Thus, the one-step loss can be defined as $\mathcal{L}(x^t,\omega^t,\psi^t) = \left(\frac{y(x^t,\omega^t)}{2\delta} - \omega^{t\mathsf{T}}\nabla_x h(r(x^t;\psi_r^t);\psi_h^t)\right)^2$, and the manifold parameters can be learned by minimizing the aforementioned loss along the trajectory, in the same vein as Follow the Regularized Leader (FTRL) algorithm (Hazan et al., 2016):

$$\psi^{t+1} = \arg\min_{\psi}\sum_{k=1}^{t}\mathcal{L}(x^k,\omega^k,\psi) + \lambda\mathcal{R}(\psi), \tag{2}$$

where $\mathcal{R}(\psi) = \|\nabla_x h(r(x^t;\psi_r^t);\psi_h^t) - \nabla_x h(r(x^t;\psi_r);\psi_h)\|_2^2$ is a temporal smoothness regularizer that penalizes sudden changes in the gradient estimates. In the theoretical analysis, it is assumed that (2) can be solved optimally. Although apparently strong, the assumption is supported by the experimental results of Sener and Koltun (2020) since neural networks can have high capacity. We note that further relaxation is possible by adopting the result from Suggala and Netrapalli (2020).

2.4. Single-task performance guarantee

The following result reveals the dependence of the convergence rate on the initial parameter x^1.

Lemma 1 *Consider running learned manifold random search (Sel et al., 2022, Algorithm. C) for T steps. Let $k_e = 1$ and $k_m = 1$ for simplicity and set $\alpha = c_0 T^{-\frac{1}{2}}$, $\beta = d^{-1}$, and $\delta = (2n)^{\frac{1}{3}}(V_F\Omega)^{\frac{1}{6}}\mu^{-\frac{1}{2}}T^{-\frac{1}{6}}$. Then, with probability $1-4\gamma\ln(T)$,*

$$\frac{1}{T}\sum_{t=1}^{T}\|\nabla_x f(x^t)\|^2 \le \frac{c_1}{\sqrt{T}}\sqrt{\sum_{t=2}^{T}\|x^t - x^1\|_2} + \frac{c_2}{T^{\frac{1}{2}}} + \frac{c_3}{T^{\frac{1}{3}}}, \tag{3}$$

where $c_0 = \left(L^2n^2\mu\Omega^{-1} + 2^{-\frac{2}{3}}n^{\frac{4}{3}}V_F^{\frac{2}{3}}\mu^2\Omega^{-\frac{4}{3}}T^{\frac{1}{3}}\right)^{-\frac{1}{2}}$, $c_1 = \Omega\sqrt{dc_1'}$, $c_2 = 4Ln\sqrt{\Omega\mu}+\Omega\sqrt{d}(\sqrt{c_5'}+\sqrt{c_1'}c_4')$, $c_3 = 5\mu(V_F\Omega)^{\frac{1}{3}}n^{\frac{2}{3}}$, *with* $\{c_j'\}_{j\in[5]}$ *defined in (Sel et al., 2022, (22)-(26)).*

Strictly speaking, the above result is not a standard result on convergence since the bound on the right-hand side of (3) depends on the initial parameter x^1 and random trajectory data $\{x^t\}_{t=1}^T$, which is also the key difference with (Sener and Koltun, 2020, Theorem 1). This development is the key step toward enabling principled meta-initialization. It is possible to replace the term $\sqrt{\sum_{t=1}^T \|x^t - x^1\|_2}$ in (3) by $\sqrt{\|x^T - x^1\|_2}$ (to reflect the relation between the initial point and the final point) with a slight change of constants; in this case, we can recover the rate of $\mathcal{O}(T^{-\frac{1}{3}})$ as in (Sener and Koltun, 2020, Theorem 1). However, we introduce the dependence on the trajectory for the purpose of trajectory-based meta-learning; see the discussion on task-similarity in Sec. 3.1.

The next result reveals the dependence of the convergence rate on the manifold parameter.

Lemma 2 *Consider running learned manifold random search (Sel et al., 2022, Algorithm. 2) in for T steps. Let $k_e = 1$ and $k_m = 1$ for simplicity and set $\alpha = b_0T^{-\frac{1}{2}}$, $\beta = d^{-1}$, and $\delta = 2^{\frac{1}{2}}n^{\frac{1}{3}}(V_F\Omega)^{\frac{1}{6}}\mu^{-\frac{1}{2}}T^{-\frac{1}{6}}$. Then, with probability $1-4\gamma\ln(T)$, we have that*

$$\frac{1}{T}\sum_{t=1}^{T}\|\nabla_x f(x^t)\|^2 \le \frac{b_1}{\sqrt{T}}\sqrt{\sum_{t=2}^{T}\|\nabla_x h(r(x^t;\psi_r^t);\psi_h^t) - \nabla_x h(r(x^t;\psi_r^1);\psi_h^1)\|_2} + \frac{b_2}{T^{\frac{1}{2}}} + \frac{b_3}{T^{\frac{1}{3}}}. \quad (4)$$

where $b_0 = \left(L^2n^2\mu\Omega^{-1} + n^{\frac{4}{3}}V_F^{\frac{2}{3}}\mu^2\Omega^{-\frac{4}{3}}T^{\frac{1}{3}}\right)^{-\frac{1}{2}}$, $b_1 = \Omega\sqrt{db_1'}$, $b_2 = 4Ln\sqrt{\Omega\mu} + \Omega\sqrt{d}(\sqrt{b_5'} + \sqrt{b_1'}b_4')$, $b_3 = 6\mu V_F^{\frac{1}{3}}\Omega^{\frac{1}{3}}n^{\frac{2}{3}}$, *with* $\{b_j'\}_{j\in[5]}$ *defined in (Sel et al., 2022, (33)-(37))*

2.5. Task-averaged regret on stationarity

The meta-manifold search aims to learn a meta-initialization model that facilitates each task to be solved after a few rounds of adaptation. Therefore, we will seek to minimize the task-averaged regret on stationarity defined as follows.

Definition 3 *The **task-averaged regret on stationarity (TARS)** $\bar{R}$ after M tasks is*

$$\bar{R}(M,T) = \frac{1}{M}\sum_{i=1}^{M}\mathbb{E}_T\|\nabla f_i(x_i)\|_2^2, \quad (5)$$

where x_i is the returned by running some within-task algorithm for T timesteps at task i and the expectation is taken with respect to the meta and within-task algorithms and the environment.

We can expect that the upper bounds on the task-averaged regrets depend on the meta-initialization for each task. However, unlike in the standard regret, one cannot achieve TARS decreasing in M without further assumptions on the environment because the set of first-order stationary points may change arbitrarily from task to task. Generally, we expect TARS to improve with the similarity among the online optimization tasks. Furthermore, the notion of similarity not only affects the evaluation of the meta-learning algorithm, but also impacts the quality of the meta initialization being learned and, eventually, the performance on an unseen task.

3. Methodology

3.1. Meta-learning the initial search point

The meta-learner's goal is to make a sequence of decisions on the initial search points that can lead to quick adaptation to each task. To begin with, recall the convergence rate for a single-task LMRS:

$$U_i(x_i^1) := \frac{c_{i,1}}{\sqrt{T}} \sqrt{\sum_{t=2}^{T} \|x_i^t - x_i^1\|_2} + \frac{c_{i,2}}{T^{\frac{1}{2}}} + \frac{c_{i,3}}{T^{\frac{1}{3}}}, \tag{6}$$

where the constants $\{c_{i,j}\}_{j=1,\dots,3}$ are given in Lemma 1. A key observation is that we can bound each term of the dynamic regret in (5) corresponding to task i by a loss term based on the initial policy $U_i(x_i^1)$. Thus, summing from $i = 1$ to M and dividing by M, we have that

$$\bar{R}(M,T) \leq \frac{1}{M} \sum_{i=1}^{M} U_i(x_i^1). \tag{7}$$

Therefore, we have transformed the original problem of bounding the dynamic regret TARS into a relaxed problem of bounding the right-hand side of the inequality above, which can be formulated as a standard online learning problem. In particular, we can treat U_i as a loss function, which is revealed after the completion of each task and instantiated with the past trajectory $\{x_i^t\}_{t=1}^T$. Let $\bar{U}^{\text{param}}(M)$ be the upper bound on the regret with respect to static initialization z,

$$\frac{1}{M} \sum_{i=1}^{M} U_i(x_i^1) - U_i(z) \leq \bar{U}^{\text{param}}(M), \tag{8}$$

where the sequence of initialization $\{x_i^1\}_{i=1}^M$ are obtained by some online learning algorithm. Thus, we can proceed to bound $\bar{R}(M,T)$ by $\bar{U}^{\text{param}}(M) + \frac{1}{M}\sum_{i=1}^{M} U_i(z)$. In the following, we discuss

Follow-the-regularized meta leader. Given a starting point x_0 and a fixed learning rate $\eta > 0$, for a sequence of functions $\{\ell_i : \mathcal{X} \to \mathbb{R}\}_{i \geq 1}$, follow-the-regularized leader (FTRL) plays

$$x_i = \arg\min_{x \in \mathcal{X}} \frac{1}{2}\|x - x_0\|_2^2 + \eta \sum_{s<i} \ell_s(x),$$

where $\mathcal{X}$ is a compact convex set considered for initialization; we denote the radius of the set as $D = \sup_{x,x' \in \mathcal{X}} \|x - x'\|_2$. In our follow-the-regularized meta leader (FTRML) algorithm, we perform FTRL on the loss functions $\{U_i : \mathcal{X} \to \mathbb{R}\}_{i \in [M]}$. We assume that U_i is G_i-Lipschitz with respect to $\|\cdot\|_2$, which is satisfied when $\mathcal{X}$ is compact and U_i is bounded away from zero.

Task-similarity measure. In gradient-based meta-learning, we aim to find an initial point that performs well for a new task after a few updates (Finn et al., 2017; Lee et al., 2019). Hence, it is natural to measure similarity between tasks based on *initial points* and *optimization trajectories*. As we tread beyond standard gradient descent, it is meaningful to define such a measure with respect to specific within-task algorithms, i.e., LMRS, which play a role in shaping search behavior. To this end, we introduce a new notion of task-similarity,

$$V_{\text{init}}^* = \min_x \left\{ V_{\text{init}}(x) := \sup_{\{x_i^{t+1} \sim \mathcal{A}lg^t(x, \Box f_i)\}_{i \in [M], t \in [T-1]}} \frac{1}{M} \sum_{i=1}^{M} U_i(x; \{x_i^{t+1}\}_{t \in [T-1]}) \right\}, \tag{9}$$

where $U_i(X; \{x_i^t\}_{t\in[T]})$ makes the dependence of (6) on the trajectory data $\{x_i^t\}_{t\in[T]}$ explicit. Specifically, the constraint $x_i^{t+1} \sim \mathcal{A}lg^t(x, \Box f_i)$ requires that the trajectory data $\{x_i^t\}_{t\in[T]}$ is generated by $\mathcal{A}lg$ starting from the initial point x. We take the supremeum over all possible realizations of the trajectories to remove the randomness in the definition of V^*_{init}. As seen from the definition above, V^*_{init} depends on both the initialization and the optimization trajectories, and is specific to the chosen algorithm (LMRS in our case).

Theoretical bound of TARS. In the following, we show that TARS can be reduced with an increasing number of tasks (i.e., increasing M) or more task-similarity (i.e., lower V^*_{init}).

Theorem 4 *Let $\{x_i^1\}_{i\in[M]}$ be obtained by running FTRML on the sequence of loss functions $\{U_i\}_{i=1,\dots,M}$, with initial point $x_0 \in \mathcal{X}$ and learning rate $\eta = \sqrt{\frac{D}{G^2 M}}$, where $G^2 \geq \frac{1}{M}\sum_{i=1}^M G_i^2$, we have that*

$$\bar{R}(M,T) \lesssim \frac{1}{\sqrt{M}} + V^*_{\text{init}}.$$

The above result reveals a *key theoretical benefit of meta-learning*. Suppose we treat each task as independent and start from the same initial point ϕ. By (7) and (9), we can bound TARS $\bar{R}(M,T)$ by $V_{\text{init}}(\phi)$. For meta-learning to improve upon the single-task learning TARS, we need to have one of the following conditions: **1)** the number of tasks is large enough: $M \gtrsim \frac{G^2 D}{(V_{\text{init}}(\phi)-V^*_{\text{init}})^2}$, or **2)** the tasks are sufficiently similar: $V^*_{\text{init}} \lesssim V_{\text{init}}(\phi) - \frac{G\sqrt{D}}{\sqrt{M}}$, where we have omitted some constant factors. For practical relevance, we do not need to access the exact values of V^*_{init} to run FTRML. Note that similar results as above also hold if we replace FTRL with online mirror descent or other online algorithms (Hazan et al., 2016).

3.2. Meta-learning the search manifold

High-dimensional problems that arise in real-world settings often lie in latent low-dimensional manifolds. Our critical insight is that *problems of similar nature also share this latent space, and thus can in principle be meta-learned*. In the context of blackbox random search, this is of particular interest because the manifold is typically not known prior to the optimization, and learning such a manifold is both data- and computation-intensive.

Our strategy is parallel to the one outlined in Sec. 3.1. Thus, we will only highlight the key differences. First, recall the manifold-dependent convergence rate for a single-task LMRS:

$$U_i'(\psi_i^0) \coloneqq \frac{b_{i,1}}{\sqrt{T}}\sqrt{\sum_{t=2}^{T} \|\nabla_x h(r(x_i^t;\psi_{r,i}^t);\psi_{h,i}^t) - \nabla_x h(r(x_i^t;\psi_{r,i}^1);\psi_{h,i}^1)\|_2} + \frac{b_{i,2}}{T^{\frac{1}{2}}} + \frac{b_{i,3}}{T^{\frac{1}{3}}}, \tag{10}$$

where $\psi_i^t = (\psi_{r,i}^t, \psi_{r,i}^t)$ and the constants $\{b_{i,j}\}_{j=1,\dots,3}$ are given in Lemma 2. Different from (6), U_i' is a function of manifold parameters ψ_i^0 and is in general *nonconvex*. Hence, FTRML that runs on the sequence of $\{U_i'\}_{i\in[M]}$ cannot achieve sublinear regret (Suggala and Netrapalli, 2020, Prop. 3). Thus, a different strategy is entailed.

Follow-the-perturbed meta leader. Consider the (γ,τ)-approximate optimization oracle, which, for a given function $\ell : \Psi \to \mathbb{R}$ and a d'-dimensional vector σ, returns an approximate minimizer $\psi^* \in \Psi$ such that

$$\ell(\psi^*) - \langle \sigma, \psi^* \rangle \leq \inf_{\psi\in\Psi} \{\ell(\psi) - \langle \sigma, \psi\rangle\} + (\gamma + \tau\|\sigma\|_1).$$

We denote such an oracle by $\mathcal{Q}_{\gamma,\tau}(\ell - \langle\sigma,\cdot\rangle)$. For a sequence of functions $\{\ell_i : \mathcal{X} \to \mathbb{R}\}_{i\geq 1}$, follow-the-perturbed leader (FTPL) plays $\psi_i = \mathcal{Q}_{\gamma,\tau}\left(\sum_{s<i}\ell_s(\psi) - \langle\sigma_s,\cdot\rangle\right)$, where σ_s is a random perturbation such that its j-th coordinate $\sigma_{s,j}$ is sampled from the exponential distribution $\mathrm{Exp}(\eta)$ with parameter η (Agarwal et al., 2019). In our follow-the-perturbed meta leader (FTPML) algorithm, we perform FTPL on the loss functions $\{U_i'\}_{i\in[M]}$.

Task similarity based on the manifold. Similar to meta-initializing the starting point, we aim to define a notion of similarity that depends on manifold parameters and the single task optimization procedure. For simplicity, let $\zeta = (x,\psi)$. To this end, consider the following metric:

$$V_{\text{manifold}}(\psi) \coloneqq \sup_{x\in\mathcal{X},\{\zeta_i^{t+1}\sim\mathcal{A}lg^t(\zeta,\Box f_i)\}_{i\in[M],t\in[T-1]}} \frac{1}{M}\sum_{i=1}^{M} U_i'(\psi;\{\zeta_i^{t+1}\}_{t\in[T-1]}), \tag{11}$$

where $U_i'(\psi;\{\zeta_i^{t+1}\}_{t\in[T-1]}$ makes the dependence of (10) on the trajectory data $\{\zeta_i^t\}_{t=1}^T$ explicit. The constraint $\zeta_i^{t+1} \sim \mathcal{A}lg^t(\zeta,\Box f_i)$ can be satisfied where LMRS is chosen as a within-task algorithm, which conducts joint optimization and manifold learning. Hence, we define $V^*_{\text{manifold}} = \min_\psi V_{\text{manifold}}(\psi)$, which depends on both the initialization and the optimization trajectories.

Theoretical bound of TARS. We assume that U_i' is G_i'-Lipschitz with respect to $\|\cdot\|_1$, which is satisfied when U_i' is bounded away from zero. We state the following result for FTPML.

Theorem 5 *Let $\{\psi_i^1\}_{i\in[M]}$ be obtained by running FTPML on the sequence of loss functions $\{U_i : \Psi \to \mathbb{R}\}_{i\in[M]}$, with appropriately chosen η, we have that*

$$\bar{R}(M,T) \lesssim \sqrt{\frac{d_\Psi^3 D_\Psi G'^2(\tau M + D_\Psi)}{M}} + \gamma + \tau d_\Psi G' + V^*_{\text{manifold}}.$$

The above result indicates that FTPML achieves $\mathcal{O}\left(M^{-\frac{1}{2}} + \gamma + \sqrt{\tau} + V^*_{\text{manifold}}\right)$. Hence, in the ideal case when both γ and τ are equal to zero, the theoretical benefits of meta-learning can be shown to be similar to the discussion after Theorem 4 under the conditions that either M is large or the tasks are sufficiently similar. In fact, the advantage over single-task learning persists as long as $\gamma = \mathcal{O}\left(M^{-\frac{1}{2}}\right)$ and $\tau = \mathcal{O}\left(M^{-1}\right)$, which is reasonable given that heuristics such as stochastic gradient descent seem to find approximate global optima for a variety of optimization landscapes (including training deep neural networks).

3.3. Joint meta-learning of the initial search point and the manifold

To gain the most benefits from meta-learning, we can jointly learn the initial search point x_i^1 and manifold parameters ψ_i. This is possible since the upper bounds of (6) and (10) can be combined:

$$\bar{R}(M,T) \leq \frac{1}{M}\sum_{i=1}^{M} \kappa U_i(x_i^1) + (1-\kappa)U_i'(\psi_i^1), \tag{12}$$

where $\kappa \in [0,1]$ is the weight. For a fixed κ, we can conduct online learning on the sequence of $\{\kappa U_i + (1-\kappa)U_i'\}_{i\in[M]}$ by running two independent processes on the two sequences $\{U_i\}_{i\in[M]}$ and $\{U_i'\}_{i\in[M]}$ with FTRML and FTPML, respectively. This is based on the observation that U_i depends only on x_i^1 and U_i' depends only on ψ_i^1 (note that the summation in (10) starts from $t = 2$). As a result, it is straightforward to see that TARS can be bounded by the weighted sum of bounds from Theorems 4 and 5. While in practice, we can set κ to be an arbitrary number in $[0, 1]$, it is unclear if a better setting exists. Indeed, a higher value of κ weighs more on meta-learning the initial point (i.e., the $\{U_i\}_{i\in[M]}$ sequence) while a lower value of κ weighs more on manifold meta-learning (i.e., the $\{U_i'\}_{i\in[M]}$ sequence).

To this end, we consider adapting the weights κ together with FTRML on x_i^1 and FTPML on ψ_i^1. Let $U_i^{\text{sim}}(\kappa) = \kappa U_i(x_i^1) + (1-\kappa)U_i'(\psi_i^1)$ be a function of κ conditioning on the values of $U_i(x_i^1)$ and $U_i'(\psi_i^1)$. Hence, U_i^{sim} is simply a linear function of κ. In the i-th meta update, we run the following parallel processes: *(1)* FTRL on $\{U_m^{\text{sim}}\}_{m<i}$ to obtain κ_i, *(2)* FTRML on $\{\kappa_m U_m\}_{m<i}$ to obtain x_i^1, and *(3)* FTRML on $\{\kappa_m U_m'\}_{m<i}$ to obtain ψ_i^1. We prove the following bound on TARS.

Theorem 6 *Let $\{\kappa_i, x_i^1, \psi_i^1\}_{i\in[M]}$ be obtained by running FTRL, FTRML, and FTPML on sequences $\{U_m^{\text{sim}}\}_{m<i}$, $\{U_i\}_{i\in[M]}$, and $\{U_i'\}_{i\in[M]}$, respectively and in parallel. we have that*

$$\bar{R}(M,T) \lesssim \kappa\left(\frac{1}{\sqrt{M}} + V_{\text{init}}^*\right) + (1-\kappa)\left(\sqrt{\frac{d_\Psi^3 D_\Psi G'^2(\tau M + D_\Psi)}{M}} + \gamma + \tau d_\Psi G' + V_{\text{manifold}}^*\right)$$

for any $\kappa \in (0, 1)$.

From the bound, it may not seem obvious the theoretical improvement over meta-learning initialization or manifold independently, as one can change κ to either 0 or 1 depending on which of the two bounds from Theorems 4 and 5 is smaller. However, the bound in Theorem 6 is adaptive in the sense that it holds for any $\kappa \in (0, 1)$, thus obviating the need to access the exact values of the bounds in Theorems 4 and 5. We introduce the meta-learned manifold random search algorithm (Algorithm 1). In this framework, a within-task algorithm (Sel et al., 2022, Algorithm 2: LMRS) is included in the online learning framework.

Algorithm 1 Meta - Learned Manifold Random Search (Meta-LMRS)

for $i = 1, \ldots, M$ **do**
  for $t = 1, \ldots, T$ **do**
    $x_i^{t+1}, \psi_i^{t+1} = \text{LMRS}\left(x_i^t, \psi_i^t\right)$
  end for
  Evaluate $U_i(x_i^1)$ and $U_i'(\psi_i^1)$ using (6) and (10), respectively
  Update the next task parameter and manifold initialization

$$\kappa_{i+1} = \text{FTRL}\big(\{U_m^{\text{sim}}\}_{m<i}\big), x_{i+1}^1 = \text{FTRML}(\{\kappa_m U_m\}_{m<i}), \psi_{i+1}^1 = \text{FTPML}\big(\{\kappa_m U_m'\}_{m<i}\big)$$

end for

4. Experiments

We evaluate the proposed approach on two MuJoCo control problems (Todorov et al., 2012).

Experimental setup. For each task $i \in [M]$, LMRS is used as a within-task solver. FTRML and FTPML algorithms are used to meta initialize x_i^1 and ψ_i^1 at the start of each new task, respectively. We use multi-layered perceptrons as parametric functions for the manifold. We set the manifold

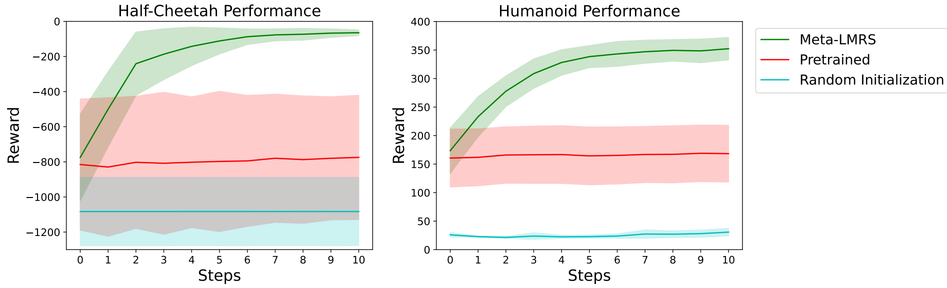

Figure 1: Performance comparison of Meta-LMRS with baselines. Both the average and standard deviation (shades) are reported across 10 independent runs.

dimension to 50 for both environments without extensive tuning (the rationale for choosing this value is discussed in (Sel et al., 2022, Sec. C)). A linear policy is used for RL agents. Two baselines are considered: *1)* the random initialization baseline uses random initialization for the within-task algorithm, and *2)* the pretrained baseline is trained on a task from the task distribution. We run the algorithm online where tasks are encountered sequentially. Further details on the Half-Cheetah and Humanoid environments can be found in (Sel et al., 2022, Sec. C).

Result and discussion. As shown in Fig. 1, at the beginning of a new task, the pretrained baseline performs similarly to Meta-LMRS. However, after a few within-task steps, Meta-LMRS is able to achieve higher rewards in both environments compared to baselines. The lack of noticeable improvement in the pre-trained baseline suggests that knowledge of a single task is not quickly transferred to other tasks.

Ablation analysis. Fig. 2 shows the incremental benefits of search point and manifold meta-initializations. Manifold initialization alone does not appear to have an apparent improvement over random initializations. Intuitively, while a well-learned manifold increases the accuracy of the estimated gradients, a few steps are not enough to reliably improve the policy if the search starts far from a good solution. Joint initialization of both policy and manifold parameters (Meta-LMRS) provides the best performance, indicating that meta-manifold benefits are enhanced when combined with a good initial point.

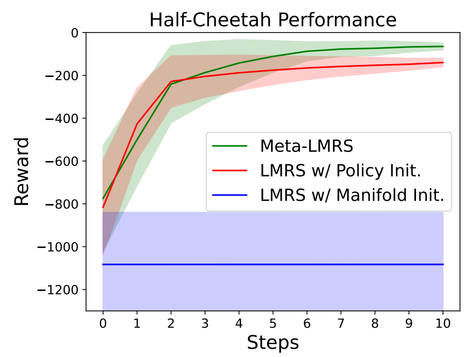

Figure 2: Comparison of incremental benefits of meta-initializing policy and manifold parameters.

5. Conclusion

In this paper, we introduce Meta-LMRS for gradient-based meta-learning over a sequence of optimization tasks without access to gradients. We demonstrate the empirical performance and theoretical benefits of such an approach for repeatedly solving similar instances of the same problem. This work is aligned with the emerging directions of learning to optimize (Chen et al., 2021) and amortized optimization (Amos, 2022) and opens up opportunities to deal with problems that may involve human feedback formulated as blackbox inquiries.

References

P-A Absil, Robert Mahony, and Rodolphe Sepulchre. Optimization algorithms on matrix manifolds. In *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2009.

Bernardetta Addis, Marco Locatelli, and Fabio Schoen. Local optima smoothing for global optimization. *Optimization Methods and Software*, 20(4-5):417–437, 2005.

Naman Agarwal, Alon Gonen, and Elad Hazan. Learning in non-convex games with an optimization oracle. In *Conference on Learning Theory*, pages 18–29. PMLR, 2019.

Heni Ben Amor, Oliver Kroemer, Ulrich Hillenbrand, Gerhard Neumann, and Jan Peters. Generalization of human grasping for multi-fingered robot hands. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2043–2050. IEEE, 2012.

Brandon Amos. Tutorial on amortized optimization for learning to optimize over continuous domains. *arXiv preprint arXiv:2202.00665*, 2022.

Antreas Antoniou, Harrison Edwards, and Amos Storkey. How to train your maml. *arXiv preprint arXiv:1810.09502*, 2018.

Aurele Balavoine, Christopher J Rozell, and Justin Romberg. Discrete and continuous-time soft-thresholding for dynamic signal recovery. *IEEE Transactions on Signal Processing*, 63(12):3165–3176, 2015.

Maria-Florina Balcan, Avrim Blum, and Santosh Vempala. Efficient representations for lifelong learning and autoencoding. In *Conference on Learning Theory*, pages 191–210. PMLR, 2015.

Maria-Florina Balcan, Mikhail Khodak, and Ameet Talwalkar. Provable guarantees for gradient-based meta-learning. In *International Conference on Machine Learning*, pages 424–433. PMLR, 2019.

Albert S Berahas, Liyuan Cao, Krzysztof Choromanski, and Katya Scheinberg. Linear interpolation gives better gradients than gaussian smoothing in derivative-free optimization. *arXiv preprint arXiv:1905.13043*, 2019.

Junting Chen and Vincent KN Lau. Convergence analysis of saddle point problems in time varying wireless systems—control theoretical approach. *IEEE Transactions on Signal Processing*, 60(1): 443–452, 2011.

Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pages 15–26, 2017.

Tianlong Chen, Xiaohan Chen, Wuyang Chen, Howard Heaton, Jialin Liu, Zhangyang Wang, and Wotao Yin. Learning to optimize: A primer and a benchmark. *arXiv preprint arXiv:2103.12828*, 2021.

Krzysztof M Choromanski, Aldo Pacchiano, Jack Parker-Holder, Yunhao Tang, and Vikas Sindhwani. From complexity to simplicity: Adaptive es-active subspaces for blackbox optimization. *Advances in Neural Information Processing Systems*, 32, 2019.

Adria Colomé and Carme Torras. Dimensionality reduction for dynamic movement primitives and application to bimanual manipulation of clothes. *IEEE Transactions on Robotics*, 34(3):602–615, 2018.

Andrew R Conn, Katya Scheinberg, and Luis N Vicente. *Introduction to derivative-free optimization*. SIAM, 2009.

Paul G Constantine. *Active subspaces: Emerging ideas for dimension reduction in parameter studies*. SIAM, 2015.

Ana Luísa Custódio, Katya Scheinberg, and Luís Nunes Vicente. Methodologies and software for derivative-free optimization. *Advances and trends in optimization with engineering applications*, pages 495–506, 2017.

Emiliano Dall'Anese, Swaroop S Guggilam, Andrea Simonetto, Yu Christine Chen, and Sairaj V Dhople. Optimal regulation of virtual power plants. *IEEE Transactions on Power Systems*, 33 (2):1868–1881, 2017.

Giulia Denevi, Carlo Ciliberto, Riccardo Grazzi, and Massimiliano Pontil. Learning-to-learn stochastic gradient descent with biased regularization. In *International Conference on Machine Learning*, pages 1566–1575. PMLR, 2019.

Yuhao Ding, Javad Lavaei, and Murat Arcak. Time-variation in online nonconvex optimization enables escaping from spurious local minima. *IEEE Transactions on Automatic Control*, 2021.

Simon S Du, Wei Hu, Sham M Kakade, Jason D Lee, and Qi Lei. Few-shot learning via learning the representation, provably. *arXiv preprint arXiv:2002.09434*, 2020.

Yan Duan, John Schulman, Xi Chen, Peter L Bartlett, Ilya Sutskever, and Pieter Abbeel. Rl˜2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*, 2016.

John C Duchi, Michael I Jordan, Martin J Wainwright, and Andre Wibisono. Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Transactions on Information Theory*, 61(5):2788–2806, 2015.

Gabriel Dulac-Arnold, Nir Levine, Daniel J Mankowitz, Jerry Li, Cosmin Paduraru, Sven Gowal, and Todd Hester. Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. *Machine Learning*, 110(9):2419–2468, 2021.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

Chelsea Finn, Aravind Rajeswaran, Sham Kakade, and Sergey Levine. Online meta-learning. In *International Conference on Machine Learning*, pages 1920–1930. PMLR, 2019.

Abraham D Flaxman, Adam Tauman Kalai, and H Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. *arXiv preprint cs/0408007*, 2004.

Luca Franceschi, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimiliano Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In *International Conference on Machine Learning*, pages 1568–1577. PMLR, 2018.

Zhi Gao, Yuwei Wu, Mehrtash T Harandi, and Yunde Jia. Curvature-adaptive meta-learning for fast adaptation to manifold data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

Saeed Ghadimi and Guanghui Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.

Abhijit Gosavi et al. *Simulation-based optimization*. Springer, 2015.

Adrian Hauswirth, Irina Subotić, Saverio Bolognani, Gabriela Hug, and Florian Dörfler. Time-varying projected dynamical systems with applications to feedback optimization of power systems. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 3258–3263. IEEE, 2018.

Elad Hazan et al. Introduction to online convex optimization. *Foundations and Trends® in Optimization*, 2(3-4):157–325, 2016.

Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(9): 5149–5169, 2021.

Max Jaderberg, Wojciech M Czarnecki, Iain Dunning, Luke Marris, Guy Lever, Antonio Garcia Castaneda, Charles Beattie, Neil C Rabinowitz, Ari S Morcos, Avraham Ruderman, et al. Human-level performance in 3d multiplayer games with population-based reinforcement learning. *Science*, 364(6443):859–865, 2019.

Kevin G Jamieson, Robert Nowak, and Ben Recht. Query complexity of derivative-free optimization. *Advances in Neural Information Processing Systems*, 25, 2012.

Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences*, 71(3):291–307, 2005.

Louis Kirsch, Sebastian Flennerhag, Hado van Hasselt, Abram Friesen, Junhyuk Oh, and Yutian Chen. Introducing symmetries to black box meta reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7202–7210, 2022.

Jeffrey Larson, Matt Menickelly, and Stefan M Wild. Derivative-free optimization methods. *Acta Numerica*, 28:287–404, 2019.

Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10657–10665, 2019.

Zhenguo Li, Fengwei Zhou, Fei Chen, and Hang Li. Meta-sgd: Learning to learn quickly for few-shot learning. *arXiv preprint arXiv:1707.09835*, 2017.

Steven H Low and David E Lapsley. Optimization flow control. i. basic algorithm and convergence. *IEEE/ACM Transactions on networking*, 7(6):861–874, 1999.

Niru Maheswaranathan, Luke Metz, George Tucker, Dami Choi, and Jascha Sohl-Dickstein. Guided evolutionary strategies: escaping the curse of dimensionality in random search. 2018.

Niru Maheswaranathan, Luke Metz, George Tucker, Dami Choi, and Jascha Sohl-Dickstein. Guided evolutionary strategies: Augmenting random search with surrogate gradients. In *International Conference on Machine Learning*, pages 4264–4273. PMLR, 2019.

Horia Mania, Aurelia Guy, and Benjamin Recht. Simple random search of static linear policies is competitive for reinforcement learning. *Advances in Neural Information Processing Systems*, 31, 2018.

Andreas Maurer, Massimiliano Pontil, and Bernardino Romera-Paredes. The benefit of multitask representation learning. *Journal of Machine Learning Research*, 17(81):1–32, 2016.

Paul Micaelli and Amos Storkey. Non-greedy gradient-based hyperparameter optimization over long horizons. 2020.

Aryan Mokhtari, Shahin Shahrampour, Ali Jadbabaie, and Alejandro Ribeiro. Online optimization in dynamic environments: Improved regret rates for strongly convex problems. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 7195–7201. IEEE, 2016.

Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17(2):527–566, 2017.

Esteban Real, Sherry Moore, Andrew Selle, Saurabh Saxena, Yutaka Leon Suematsu, Jie Tan, Quoc V Le, and Alexey Kurakin. Large-scale evolution of image classifiers. In *International Conference on Machine Learning*, pages 2902–2911. PMLR, 2017.

Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

Bilgehan Sel, Ahmad Al-Tawaha, Yuhao Ding, Ruoxi Jia, Ruoxi Jia, Bo Ji, Javad Lavaei, and Ming Jin. Learning-to-learn to guide random search: Derivative-free meta blackbox optimization on manifold. 2022. URL `http://www.jinming.tech/papers/Meta-LMRS.pdf`.

Ozan Sener and Vladlen Koltun. Learning to guide random search. In *International Conference on Learning Representations*, 2020.

Ohad Shamir. An optimal algorithm for bandit and zero-order convex optimization with two-point feedback. *The Journal of Machine Learning Research*, 18(1):1703–1713, 2017.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.

James C Spall. *Introduction to stochastic search and optimization: estimation, simulation, and control*. John Wiley & Sons, 2005.

Arun Sai Suggala and Praneeth Netrapalli. Online non-convex learning: Following the perturbed leader is optimal. In *Algorithmic Learning Theory*, pages 845–861. PMLR, 2020.

Yujie Tang and Steven Low. Distributed algorithm for time-varying optimal power flow. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 3264–3270. IEEE, 2017.

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*, pages 5026–5033. IEEE, 2012.

Samuele Tosatto, Georgia Chalvatzaki, and Jan Peters. Contextual latent-movements off-policy optimization for robotic manipulation skills. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10815–10821. IEEE, 2021.

Nilesh Tripuraneni, Michael Jordan, and Chi Jin. On the theory of transfer learning: The importance of task diversity. *Advances in Neural Information Processing Systems*, 33:7852–7862, 2020.

Ziyu Wang, Frank Hutter, Masrour Zoghi, David Matheson, and Nando De Feitas. Bayesian optimization in a billion dimensions via random embeddings. *Journal of Artificial Intelligence Research*, 55:361–387, 2016.

Tianbao Yang, Lijun Zhang, Rong Jin, and Jinfeng Yi. Tracking slowly moving clairvoyant: Optimal dynamic regret of online learning with true and noisy gradient. In *International Conference on Machine Learning*, pages 449–457. PMLR, 2016.

Kenny Young, Baoxiang Wang, and Matthew E Taylor. Metatrace: Online step-size tuning by meta-gradient descent for reinforcement learning control. *arXiv preprint arXiv:1805.04514*, 2018.

Victor M Zavala and Mihai Anitescu. Real-time nonlinear optimization as a generalized equation. *SIAM Journal on Control and Optimization*, 48(8):5444–5467, 2010.

Jiaxin Zhang, Hoang Tran, Dan Lu, and Guannan Zhang. A novel evolution strategy with directional gaussian smoothing for blackbox optimization. *arXiv preprint arXiv:2002.03001*, 2020.

Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 2021.

Appendix A. Related work

Meta-learning. Existing works on meta-learning include learning initial conditions (Finn et al., 2017; Li et al., 2017), hyperparameters (Jaderberg et al., 2019; Micaelli and Storkey, 2020), step directions (Li et al., 2017), stepsizes (Young et al., 2018; Franceschi et al., 2018; Antoniou et al., 2018; Micaelli and Storkey, 2020), and manifolds (Gao et al., 2022); recurrent neural networks can also be learned to embed previous task experience (Duan et al., 2016) (see (Hospedales et al., 2021) for review). Most initialization-based meta-learning methods focus on the setting with decomposable within-task loss functions that are often convex (Finn et al., 2019; Denevi et al., 2019; Balcan et al., 2019); nonconvex within-task settings are usually studied for multitask representation learning (Du et al., 2020; Tripuraneni et al., 2020).

Derivative-free optimization. DFO has extensive literature (see (Custódio et al., 2017; Conn et al., 2009; Larson et al., 2019) for thorough reviews). Approaches for DFO related to this work include local gradient estimation (Berahas et al., 2019; Choromanski et al., 2019; Maheswaranathan et al., 2019; Salimans et al., 2017), smoothing techniques (Flaxman et al., 2004; Addis et al., 2005), and random search (Mania et al., 2018). Random search methods estimate the directional derivative by performing perturbations to the current iterate (Spall, 2005). Convergence analysis has been established in (Nesterov and Spokoiny, 2017; Ghadimi and Lan, 2013; Shamir, 2017) for convex, nonconvex, and nonsmooth loss functions. Duchi et al. (2015); Jamieson et al. (2012) provide a lower bound on sample complexity for the convex case.

High-dimensional analysis with low-dimensional structure. Several approaches have been developed to alleviate the curse dimensionality problem and improve gradient estimation in high-dimensional search spaces (Constantine, 2015). These approaches include direct projection of directional primitives into a lower dimensional space using variants of principal component analysis (Amor et al., 2012; Colomé and Torras, 2018; Tosatto et al., 2021), guided evolutionary search (Maheswaranathan et al., 2018; Zhang et al., 2020), active subspace method limited to linear space (Choromanski et al., 2019; Maheswaranathan et al., 2019), and learning the low-dimensional manifold using neural networks (Sener and Koltun, 2020), which is extended to the meta-learning setting in the present work.

Appendix B. Missing proofs in the main text

B.1. Proof of Lemma 1

The statement will be proved by following three steps. As the proof is similar to Sener and Koltun (2020), we simplify the presentation and only highlight the differentiated parts.

Step 1: Analysis of SGD with bias. Denote g^t as the gradient at the iteration t and b^t as its bias, i.e., $b^t = \mathbb{E}[g^t] - \nabla_x \tilde{F}(x, \xi)$. Then,

$$\begin{aligned} \tilde{F}(x^{t+1}, \xi) = \tilde{F}(x^t - \alpha g^t, \xi) &\leq \tilde{F}(x^t, \xi) - \alpha \nabla_x \tilde{F}(x^t, \xi) g^t + \frac{\mu \alpha^2}{2} \|g^t\|_2^2 \\ &\leq \tilde{F}(x^t, \xi) - \alpha \nabla_x \tilde{F}(x^t, \xi)[g^t - b^t] + \frac{\mu \alpha^2}{2} \|g^t\|_2^2 + \alpha B^t, \end{aligned}$$

where the first inequality is due to the $\mu-$smoothness of the function $\tilde{F}$ and the second inequality is by assuming a bound on the bias: $|\nabla_x \tilde{F}(x^t, \xi)^\top b^t| \leq B^t$. Taking the expectation with respect to ω and ξ, we get

$$\alpha\|\nabla_x \tilde{f}(x^t)\|_2^2 \leq \mathbb{E}_{\omega,\xi}[\tilde{f}(x^t)] - \mathbb{E}_{\omega,\xi}[\tilde{f}(x^{t+1})] + \frac{\mu\alpha^2 V_g}{2} + \alpha B^t$$

where $V_g = \mathbb{E}[\|g^t\|_2^2]$. Summing up from $t = 1$ to T and dividing by α, we obtain

$$\sum_{t=1}^{T} \|\nabla_x \tilde{f}(x^t)\|_2^2 \leq \frac{\mathbb{E}_{\omega,\xi}[\tilde{f}(x^1)] - \mathbb{E}_{\omega,\xi}[\tilde{f}(x^{T+1})]}{\alpha} + \frac{\mu\alpha T V_g}{2} + \sum_{t=1}^{T} B^t. \tag{13}$$

By (Sener and Koltun, 2020, Eq. 35), we can bound the bias term $B^t \leq \Omega\|\nabla_x \tilde{F}(x^t, \xi) - \nabla_x h(r(x^t; \psi_r^t); \psi_h^t)\|_2 + \Omega\delta^2\mu^2$. Hence, for $0 \leq \beta \leq 1$,

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T} \|\nabla_x f(x^t)\|_2^2 \leq & \frac{\mathbb{E}_{\omega,\xi}[\tilde{f}(x^1)] - \mathbb{E}_{\omega,\xi}[\tilde{f}(x^{T+1})]}{\alpha T} + \frac{\mu\alpha V_g}{2} \\ & + \frac{\Omega}{T}\sum_{t=1}^{T} \|\nabla_x \tilde{F}(x^t, \xi) - \nabla_x h(r(x^t; \psi_r^t); \psi_h^t)\|_2 + \delta^2\mu^2.\end{aligned} \tag{14}$$

Step 2: Bounding the total bias $\sum_{t=1}^{T} \|\nabla_x \tilde{F}(x^t, \xi) - \nabla_x h(r(x^t; \psi_r^t); \psi_h^t)\|_2$. The total bias term is the sum of the differences between the gradients of the true function ($F(x, \xi)$) and the estimated one ($h(r(x; \psi_r); \psi_h)$). On the other hand, the empirical information we have is the projection of this loss to a random direction (ω) with an additional noise term. In this section, we will analyze the difference between the bias and the empirical loss without the noise.

Let $\Delta(\omega, x, \xi, \psi) = \left(\omega^\top\left(\nabla_x \tilde{F}(x, \xi) - \nabla_x h((r(x; \psi_r); \psi_h)\right)\right)^2$ denote the bandit feedback that we receive by projecting on the direction of ω, then, by (Sener and Koltun, 2020, Sec. A.4.1),

$$\mathbb{E}_{\omega\in\mathbb{S}^{d-1}}\Big[\Delta(\omega, x, \xi, \psi)\Big] = \frac{1}{d}\Big\|\nabla_x \tilde{F}(x, \xi) - \nabla_x h((r(x; \psi_r); \psi_h)\Big\|_2^2. \tag{15}$$

Hence, our goal is to bound the difference $\big|\Delta(\omega, x, \xi, \psi) - \mathbb{E}_\omega[\Delta(\omega, x, \xi, \psi)]\big|$. Following (Sener and Koltun, 2020, Sec. A.3.2), we have that with probability $1 - 4\gamma\ln(T)$,

$$\sum_{t=1}^{T}\mathbb{E}[\Delta^t] \leq \sum_{t=1}^{T}\Delta^t + 2\sqrt{\frac{2L^2+d}{d}}\sqrt{\sum_{t=1}^{T}\Delta^t}\sqrt{\ln(1/\gamma)} + \max\left\{\frac{8L^2+4d}{d}, 6L^2\left(\frac{1+d}{d}\right)\right\}\ln(1/\gamma) \tag{16}$$

While we are one step closer to the actual bound, note that we do not have direct access to $\Delta(\omega, x, \xi, \psi)$ as it requires evaluation of $\omega^\top\nabla_x \tilde{F}(x, \xi)$. Instead, we only have unbiased estimation $\frac{y(x^t,\omega^t,\xi^t)}{2\delta}$ included in $\mathcal{L}^t = \mathcal{L}(\omega^t, x^t, \xi^t, \psi^t)$. Thus, we proceed by bounding Δ^t by $\mathcal{L}^t$:

$$\begin{aligned}\Delta^t &= \left({\omega^t}^\top[\nabla_x \hat{F}(x^t, \xi) - \nabla_x h(r(x^t; \psi_r^t); \psi_h^t)]\right)^2 \\ &\leq \left({\omega^t}^\top\nabla_x \hat{F}(x^t, \xi) - \frac{y(x^t, \omega^t, \xi^t)}{2\delta}\right)^2 + \left(\frac{y(x^t, \omega^t, \xi^t)}{2\delta} - {\omega^t}^\top\nabla_x h(r(x^t; \psi_r^t); \psi_h^t)\right)^2 \\ &\leq \left(\mathbb{E}_{v\in\mathbb{B}^d}\left[{\omega^t}^\top\nabla_x F(x^t + \delta v, \xi) - \frac{y(x^t, \omega^t, \xi^t)}{2\delta}\right]\right)^2 + \mathcal{L}(\omega^t, x^t, \xi^t, \psi^t) \\ &\leq \mu^2\delta^2 + \mathcal{L}(\omega^t, x^t, \xi^t, \psi^t)\end{aligned} \tag{17}$$

At this point, recall that (Sener and Koltun, 2020, Eq. 46) states that

$$\sum_{t=1}^{T} \mathcal{L}(\omega^t, x^t, \xi^t, \psi^t) \leq 8\mu L \sum_{t=1}^{T} \|x^t - x^{t-1}\|_2 + 2L. \tag{18}$$

Hence, by combining the above, we have

$$\sum_{t=1}^{T} \Delta^t \leq 8\mu L \sum_{t=1}^{T} \|x^t - x^{t-1}\|_2 + \mu^2\delta^2 T + 2L \leq 16\mu L \sum_{t=1}^{T} \|x^t - x^0\|_2 + \mu^2\delta^2 T + 2L, \tag{19}$$

where we use the triangle inequality in the last relation.

Step 3: Putting it together. From (14) and (15), we have that

$$\begin{aligned}\frac{1}{T}\sum_{t=1}^{T} \|\nabla_x f(x^t)\|_2^2 &\leq \frac{\mathbb{E}_{\omega,\xi}[\tilde{f}(x^1)] - \mathbb{E}_{\omega,\xi}[\tilde{f}(x^{T+1})]}{\alpha T} + \frac{\mu\alpha V_g}{2} + \frac{\Omega}{T}\sum_{t=1}^{T} \sqrt{d\mathbb{E}[\Delta^t]} + \delta^2\mu^2 \\ &\leq \frac{2\Omega}{\alpha T} + \frac{\mu\alpha V_g}{2} + \Omega\sqrt{d}\sqrt{\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[\Delta^t]} + \delta^2\mu^2 \end{aligned} \tag{20}$$

where the second inequality uses the Jensen's inequality and the assumption that the stochastic function is bounded ($|F(x,\xi)| \leq \Omega$). From (16) and (19), we have

$$\begin{aligned}\sum_{t=1}^{T}\mathbb{E}[\Delta^t] &\leq \sum_{t=1}^{T}\Delta^t + 2\sqrt{\frac{2L^2+d}{d}}\sqrt{\sum_{t=1}^{T}\Delta^t}\sqrt{\ln(1/\gamma)} + \max\left\{\frac{8L^2+4d}{d}, 6L^2\left(\frac{1+d}{d}\right)\right\}\ln(1/\gamma) \\ &\leq c_1' \sum_{t=1}^{T}\|x^t - x^0\|_2 + c_2'\sqrt{\sum_{t=1}^{T}\|x^t - x^0\|_2} + c_3' \\ &= c_1'\left(\sqrt{\sum_{t=1}^{T}\|x^t - x^0\|_2} + c_4'\right)^2 + c_5' \end{aligned} \tag{21}$$

where

$$c_1' = 16\mu L \tag{22}$$

$$c_2' = 8\sqrt{\frac{2\mu L^3 + \mu d L}{d}}\sqrt{\ln(1/\gamma)} \tag{23}$$

$$c_3' = 2\sqrt{\frac{2\mu L^2 + d}{d}}\sqrt{\ln(1/\gamma)}\sqrt{\mu^2\delta^2 T + 2L} + \max\left\{\frac{8L^2+4d}{d}, 6L^2\left(\frac{1+d}{d}\right)\right\}\ln(1/\gamma) \tag{24}$$

$$c_4' = \frac{1}{4}\sqrt{\frac{2L^2+d}{d\mu L}}\sqrt{\ln(1/\gamma)} \tag{25}$$

$$c_5' = c_3' - c_1' {c_4'}^2 \tag{26}$$

Combine (21) with (20), we obtain that:

$$\frac{1}{T}\sum_{t=1}^{T}\|\nabla_x f(x^t)\|_2^2 \leq \frac{2\Omega}{\alpha T} + \frac{\mu\alpha V_g}{2} + \Omega\sqrt{d}\sqrt{\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[\Delta^t] + \delta^2\mu^2}$$
$$\leq \frac{2\Omega}{\alpha T} + \frac{\mu\alpha V_g}{2} + \frac{\Omega\sqrt{dc_1'}}{\sqrt{T}}\sqrt{\sum_{t=1}^{T}\|x^t - x^0\|_2 + \frac{\Omega\sqrt{d}(\sqrt{c_5'} + \sqrt{c_1'}c_4')}{\sqrt{T}} + \delta^2\mu^2}.$$

We bound V_g by choosing $\beta = 1/d$ as,

$$\mathbb{E}[V_g] \leq \mathbb{E}\left[\left(\frac{1}{d}g_e + \left(1 - \frac{1}{d}\right)g_m\right)\right] \leq 4L^2n^2 + \frac{4n^2V_F}{\delta^2}.$$

Hence, with probability $1 - 4\gamma\ln(T)$,

$$\frac{1}{T}\sum_{t=1}^{T}\|\nabla_x f(x^t)\|^2 \leq \frac{4\sqrt{\Omega\left(L^2n^2\mu + \frac{n^2V_F\mu}{\delta^2}\right)}}{\sqrt{T}}$$
$$+ \frac{\Omega\sqrt{dc_1'}}{\sqrt{T}}\sqrt{\sum_{t=1}^{T}\|x^t - x^0\|_2 + \frac{\Omega\sqrt{d}(\sqrt{c_5'} + \sqrt{c_1'}c_4')}{\sqrt{T}} + \delta^2\mu^2}.$$

where we chose $\alpha = \frac{4\Omega}{4\sqrt{\Omega\left(L^2n^2\mu + \frac{n^2V_F\mu}{\delta^2}\right)}\sqrt{T}}$. By choosing $\delta = (2n)^{\frac{1}{3}}\left(\frac{V_F\Omega}{T}\right)^{\frac{1}{6}}\frac{1}{\sqrt{\mu}}$, we have that

$$\frac{1}{T}\sum_{t=1}^{T}\|\nabla_x f(x^t)\|^2 \leq \frac{c_1}{\sqrt{T}}\sqrt{\sum_{t=1}^{T}\|x^t - x^0\|_2 + \frac{c_2}{T^{\frac{1}{2}}} + \frac{c_3}{T^{\frac{1}{3}}}},$$

where

$$c_1 = \Omega\sqrt{dc_1'}$$
$$c_2 = 4Ln\sqrt{\Omega\mu} + \Omega\sqrt{d}(\sqrt{c_5'} + \sqrt{c_1'}c_4')$$
$$c_3 = 5\mu V_F^{\frac{1}{3}}\Omega^{\frac{1}{3}}n^{\frac{2}{3}}.$$

B.2. Proof of the Lemma 2

The proof is similar to that of Lemma 1. However, in Step 2, instead of bounding $\sum_{t=1}^{T}\mathbb{E}[\Delta^t]$ by (19), we seek to bound $\sum_{t=1}^{T}\Delta^t$ by the manifold learning errors. In particular, we directly sum over (17) to obtain:

$$\sum_{t=1}^{T}\Delta^t \leq \sum_{t=1}^{T}\mathcal{L}(\omega^t, x^t, \xi^t, \psi^t) + \mu^2\delta^2. \tag{27}$$

Next, by the FTL-BTL Lemma (Kalai and Vempala, 2005),

$$\sum_{t=1}^{T} \mathcal{L}(\omega^t, x^t, \xi^t, \psi^t) \leq \sum_{t=1}^{T} \left[\mathcal{L}(\omega^t, x^t, \xi^t, \psi^t) - \mathcal{L}(\omega^t, x^t, \xi^t, \psi^{t+1})\right] + 2L.$$

We use the Lipschitz smoothness property to convert this into distance travelled by the learner as

$$\begin{aligned}
&\mathcal{L}(\omega^t, x^t, \xi^t, \psi^t) - \mathcal{L}(\omega^t, x^t, \xi^t, \psi^{t+1}) \\
&= \left(\frac{y(x^t, \omega^t, \xi^t)}{2\delta} - {\omega^t}^\top \nabla_x h(r(x^t; \psi_r^t); \psi_h^t)\right)^2 \\
&\qquad\qquad - \left(\frac{y(x^t, \omega^t, \xi^t)}{2\delta} - {\omega^t}^\top \nabla_x h(r(x^t; \psi_r^{t+1}); \psi_h^{t+1})\right)^2 \\
&\leq 4L \sum_{t=1}^{T} {\omega^t}^\top (\nabla_x h(r(x^t; \psi_r^t); \psi_h^t) - \nabla_x h(r(x^t; \psi_r^{t+1}); \psi_h^{t+1})) \\
&\leq 4L \sum_{t=1}^{T} \|\nabla_x h(r(x^t; \psi_r^t); \psi_h^t) - \nabla_x h(r(x^t; \psi_r^{t+1}); \psi_h^{t+1})\|_2 \\
&\leq 8L \sum_{t=1}^{T} \|\nabla_x h(r(x^t; \psi_r^t); \psi_h^t) - \nabla_x h(r(x^t; \psi_r^0); \psi_h^0)\|_2
\end{aligned}$$

where the last relation is due to triangle inequality. Hence, we have (c.f., (19)):

$$\sum_{t=1}^{T} \Delta^t \leq 8L \sum_{t=1}^{T} \|\nabla_x h(r(x^t; \psi_r^t); \psi_h^t) - \nabla_x h(r(x^t; \psi_r^0); \psi_h^0)\|_2 + \mu^2\delta^2 T + 2L. \tag{28}$$

By following Step 3 as the proof of Lemma 1, we have that with probability $1 - 4\gamma \ln(T)$,

$$\frac{1}{T}\sum_{t=1}^{T} \|\nabla_x f(x^t)\|^2 \leq \frac{b_1}{\sqrt{T}} \sqrt{\sum_{t=1}^{T} \|\nabla_x h(r(x^t; \psi_r^t); \psi_h^t) - \nabla_x h(r(x^t; \psi_r^0); \psi_h^0)\|_2} + \frac{b_2}{T^{\frac{1}{2}}} + \frac{b_3}{T^{\frac{1}{3}}}.$$

where $\alpha = b_0 T^{-\frac{1}{2}}$, $\delta = (2n)^{\frac{1}{3}} \left(\frac{2V_F\Omega}{T}\right)^{\frac{1}{6}} \frac{1}{\sqrt{\mu}}$,

$$b_0 = \left(L^2 n^2 \mu \Omega^{-1} + n^{\frac{4}{3}} V_F^{\frac{2}{3}} \mu^2 \Omega^{-\frac{4}{3}} T^{\frac{1}{3}}\right)^{-\frac{1}{2}} \tag{29}$$

$$b_1 = \Omega\sqrt{d b_1'} \tag{30}$$

$$b_2 = 4Ln\sqrt{\Omega\mu} + \Omega\sqrt{d}(\sqrt{b_5'} + \sqrt{b_1' b_4'}) \tag{31}$$

$$b_3 = 6\mu V_F^{\frac{1}{3}} \Omega^{\frac{1}{3}} n^{\frac{2}{3}}. \tag{32}$$

with

$$b'_1 = 8L \tag{33}$$

$$b'_2 = 4\sqrt{\frac{4L^3 + 2dL}{d}}\sqrt{\ln(1/\gamma)} \tag{34}$$

$$b'_3 = 2\sqrt{\frac{2\mu L^2 + d}{d}}\sqrt{\ln(1/\gamma)}\sqrt{\mu^2\delta^2 T + 2L} + \max\left\{\frac{8L^2 + 4d}{d}, 6L^2\left(\frac{1+d}{d}\right)\right\}\ln(1/\gamma) \tag{35}$$

$$b'_4 = \frac{1}{4}\sqrt{\frac{4L^2 + 2d}{dL}}\sqrt{\ln(1/\gamma)} \tag{36}$$

$$b'_5 = b'_3 - b'_1 {b'_4}^2 \tag{37}$$

B.3. Proof of Theorem 4

Note that U_i is convex for all $i = 1, ..., M$. By applying the standard result on the regret bound (Hazan et al., 2016), we have that $\bar{U}^{\text{param}}(M) \leq 2\frac{G\sqrt{D}}{\sqrt{M}}$. Combining the above with (7), (8), and (9) proves the claim.

B.4. Proof of Theorem 5

By (Suggala and Netrapalli, 2020, Thm. 2) and with the choice of $\eta = \sqrt{\frac{\tau M + D_\Psi}{M d_\Psi D_\Psi G'^2}}$, we have that $\frac{1}{M}\sum_{i=1}^{M} U'_i(\psi_i^1) - U'_i(\psi) \lesssim \sqrt{\frac{d_\Psi^3 D_\Psi G'^2(\tau M + D_\Psi)}{M}} + \gamma + \tau d_\Psi G'$. The result follows by the definition of V^*_{manifold}.

B.5. Proof of Theorem 6

Let $R_M^{\text{init}}(x) \geq \frac{1}{M}\sum_{i=1}^{M} U_i(x_i^1) - U_i(x)$ be the regret upper bound compared to a static decision $x \in \mathcal{X}$ when playing FTRML on $\{U_i\}_{i\in[M]}$. Similarly, let $R_M^{\text{mani}}(\psi)$ and $R_M^{\text{sim}}(\kappa)$ be the regret upper bounds against comparators ψ and κ when playing FTPML and FTRL on $\{U'_i\}_{i\in[M]}$ and $\{U_i^{\text{sim}}\}_{i\in[M]}$, respectively. Then, the following holds:

$$\begin{aligned}
\bar{R}(M,T) &\leq \frac{1}{M}\sum_{i=1}^{M} \kappa_i U_i(x_i^1) + (1-\kappa_i)U'_i(\psi_i^1) \\
&\leq \min_{\kappa\in[0,1]} R_M^{\text{sim}}(\kappa) + \sum_{i=1}^{M} \kappa U_i(x_i^1) + (1-\kappa)U'_i(\psi_i^1) \\
&\leq \min_{\kappa\in[0,1]} R_M^{\text{sim}}(\kappa) + \kappa \underbrace{\left(\min_{x\in\mathcal{X}} R_M^{\text{init}}(x) + \sum_{i=1}^{M} U_i(x)\right)}_{(i)} + (1-\kappa)\underbrace{\left(\min_{\psi\in\Psi} R_M^{\text{mani}}(\psi) + \sum_{i=1}^{M} U'_i(\psi)\right)}_{(ii)},
\end{aligned}$$

where the first inequality is due to (6) and (10), the second and third inequalities are due to the definition of $R_M^{\text{sim}}(\kappa)$, $R_M^{\text{init}}(x)$, and $R_M^{\text{mani}}(\psi)$. By Theorems 4 and 5, we can bound $(i) \lesssim \frac{1}{\sqrt{M}} + V^*_{\text{init}}$ and $(ii) \lesssim \sqrt{\frac{d_\Psi^3 D_\Psi G'^2(\tau M + D_\Psi)}{M}} + \gamma + \tau d_\Psi G' + V^*_{\text{manifold}}$. Since $R_M^{\text{sim}}(\kappa) \lesssim \frac{1}{\sqrt{M}}$ for FTRL, combining the above proves the result.

B.6. Base algorithm (LMRS)

Algorithm 2 Learned Manifold Random Search (Sener and Koltun, 2020)

for $t = 1, \ldots, T$ **do**
 $g_e^t = \textsc{GradEst}\left(x^t, \delta\right)$ (Sener and Koltun, 2020, Alg. 1).
 $g_m^t = \textsc{ManifoldGradEst}\left(x^t, J_q(x^t; \psi_r^t)\right)$ (Sener and Koltun, 2020, Alg. 2).
 $g^t = (1-\beta)g_m^t + \beta g_e^t$
 $x^{t+1} = x^t - \alpha g^t$
 $\psi^{t+1} = \arg\min_\psi \sum_{k=1}^t \mathcal{L}(x^k, \omega^k, \psi) + \lambda \mathcal{R}(\psi)$
end for

Appendix C. Experiment details

Humanoid. Humanoid is a simulation environment of a 3D bipedal robot designed to simulate a human. The object consists of a torso (abdomen) with two arms and two legs. Each leg/arm consists of two links connected by knees/elbows. The manifold dimension n is chosen to be 50, which is much lower than the ambient dimension $d = 6392$. Task variations are introduced with different desired forward directions, randomly selected from a uniform distribution $\mathcal{U}[-\frac{\pi}{2}, \frac{\pi}{2}]$. The maximum episode length is chosen to be 250.

Half-Cheetah. Half-Cheetah is a 2-dimensional robot that consists of 9 links and 8 joints. The manifold dimension is chosen to be $n = 50$, while the ambient dimension $d = 119$. Task variations are introduced with different goal velocities, which are randomly sampled from a uniform distribution $\mathcal{U}[0, 2]$. The episode length is 1000.

Rationale of setting the manifold dimension. In our experiments, we set the manifold dimension $n = 50$, which matches the number of zeroth-order blackbox queries used to estimate search gradients. This is based on the intuition that without using any simultaneous perturbation techniques, such as when coordinate-wise perturbation is employed, the number of queries should be roughly equal to the dimension of the problem. In our experiments, because we query the environment 50 times (i.e., performing 50 rollouts) for each estimate of policy gradients, we choose the manifold dimension accordingly.