

# 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*

**Editors:** N. Matni, M. Morari, G. J. Pappas

## 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

---

\* Corresponding author

et al., 2016; Yang et al., 2016), and signal processing (Balavoine et al., 2015). Similarly, in multi-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*.<sup>1</sup>

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 the 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  $V_{\text{init}}^*$  and  $V_{\text{manifold}}^*$ , 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  $\mathcal{O}\left(M^{-\frac{1}{2}} + \max(V_{\text{init}}^*, c + V_{\text{manifold}}^*)\right)$ , 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  $V_{\text{init}}^*$  and  $V_{\text{manifold}}^*$  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

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

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

Our goal is to learn meta-parameters  $\theta \in \Theta$  that produce a good task-specific solution  $\theta_i$  after a few steps of random search. In particular, let  $\text{Alg}^t(\theta, \square f_i)$  corresponds to performing  $t$  steps of DFO initialized with  $\theta$ . For example, if one step of random search is taken and  $\theta$  corresponds to the initial point, we have  $\theta_i \equiv \text{Alg}^1(\theta, \square f_i) = \theta - \alpha \hat{g}_i$ , where  $\alpha$  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  $S^{d-1}$  and  $B^d$  denote the  $d$ -dimensional unit sphere and unit ball, and  $\omega$  be a random vector sampled from the uniform distribution over  $S^{d-1}$ . For a function  $f : \mathbb{R}^d \rightarrow \mathbb{R}$ , its  $\delta$ -smoothed version is  $\hat{f}(\theta) =_{\omega} [f(\theta + \delta\omega)]$ . Then,

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

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

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 a DFO  $\text{Alg}^t(\theta, \square f)$ .

**Notations.** We use  $\|\cdot\|_1$ ,  $\|\cdot\|_2$ , and  $\|\cdot\|_\infty$  to denote  $\ell_1$ ,  $\ell_2$ , and  $\ell_\infty$ -norms, respectively. We use the shorthand notation  $[M] = \{1, \dots, M\}$ .

### 2.2. Assumptions

We assume that the stochastic function is bounded ( $|F(\theta, \xi)| \leq \Omega$ ),  $L$ -Lipschitz, and  $\mu$ -smooth with respect to  $\theta$  for all  $\xi$ , and has uniformly bounded variance ( $E_{\xi}[(F(\theta, \xi) - f(\theta))^2] \leq V_F$ ). Furthermore, we assume that given  $\xi$ ,  $F(\cdot, \xi)$  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):  $F(\theta, \xi) = h(r(\theta; \psi_r^*); \psi_h^*)$  for all  $\theta \in \mathbb{R}^d$ . The above assumptions are adopted by Sener and Koltun (2020). For simplicity, we denote  $\psi = (\psi_r, \psi_h) \in \Psi$ , where  $\Psi$  is assumed to have a bounded  $\ell_\infty$  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 ?? 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  $\nabla f_\psi(\cdot) = (; \psi_r) \nabla h(\cdot; \psi_h)$ , where  $(; \psi_r) = \partial r(\cdot; \psi_r) / \partial \cdot = r(\cdot, \psi_r)$ . LMRS first orthonormalizes the Jacobian  $(; \psi_r)$  using the Gram–Schmidt procedure for numerical stability and then performs the random search in the column space of this orthonormal matrix  $q(\cdot; \psi_r)$ , which has lower dimensionality than the full space. For each step, an  $n$ -dimensional vector  $\tilde{\omega}$  is sampled uniformly from  $n-1$  and lifted to the input space via  $q(\cdot; \psi_r) \tilde{\omega}$ . 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)):

$$\xi, \tilde{\omega} \in^{n-1} [\hat{g}(\cdot, q(\cdot; \psi_r) \tilde{\omega})] = \frac{2\delta}{n} \nabla \tilde{f}_\psi(\cdot),$$

where  $\tilde{f}_\psi(\cdot) \approx_n [f(\cdot + \delta q(\cdot; \psi_r))]$  is the smoothed function. In each iteration, exploration is added by sampling directions from the manifold  $\gg_m$  and the full space  $\gg_e$ , which are mixed to obtain the final estimate  $\gg = (1 - \beta) \gg_m + \beta \gg_e$ .

**Manifold learning.** At each iteration, we observe  $y(t, \omega^t) = f(t + \delta q(t; \psi_r^t) \omega^t) - f(t - \delta q(t; \psi_r^t) \omega^t)$ , which is the projection of the gradient onto the chosen directions. Thus, the one-step loss can be defined as  $\mathcal{L}(t, \omega^t, \psi^t) = \left( \frac{y(t, \omega^t)}{2\delta} - \omega^{t\top} \nabla h(r(t; \psi_r^t); \psi_h^t) \right)^2$ , 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](#)):

$$\psi^{t+1} =_\psi \sum_{k=1}^t \mathcal{L}(k, \omega^k, \psi) + \lambda \mathcal{R}(\psi), \quad (2)$$

where  $\mathcal{R}(\psi) = \|\nabla h(r(t; \psi_r^t); \psi_h^t) - \nabla h(r(t; \psi_r); \psi_h)\|_2^2$  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 <sup>1</sup>.

**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  $1 - 4\gamma \ln(T)$ ,*

$$\frac{1}{T} \sum_{t=1}^T \|\nabla f(t)\|^2 \leq \frac{c_1}{\sqrt{T}} \sqrt{\sum_{t=2}^T \|\cdot\|_2} + \frac{c_2}{T^{\frac{1}{2}}} + \frac{c_3}{T^{\frac{1}{3}}}, \quad (3)$$

where  $c_0 = \left( L^2 n^2 \mu \Omega^{-1} + 2^{-\frac{2}{3}} n^{\frac{4}{3}} V_F^{\frac{2}{3}} \mu^2 \Omega^{-\frac{4}{3}} T^{\frac{1}{3}} \right)^{-\frac{1}{2}}$ ,  $c_1 = \Omega \sqrt{dc'_1}$ ,  $c_2 = 4Ln\sqrt{\Omega\mu} + \Omega\sqrt{d}(\sqrt{c'_5} + \sqrt{c'_1 c'_4})$ ,  $c_3 = 5\mu(V_F\Omega)^{\frac{1}{3}}n^{\frac{2}{3}}$ , 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<sup>1</sup> and random trajectory data  $\{t\}_{t=1}^T$ , 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  $\sqrt{\sum_{t=1}^T \|t^{-1}\|_2}$  in (3) by  $\sqrt{\|T^{-1}\|_2}$  (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  $\alpha = b_0 T^{-\frac{1}{2}}$ ,  $\beta = d^{-1}$ , 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

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

where  $b_0 = \left( L^2 n^2 \mu \Omega^{-1} + n^{\frac{4}{3}} V_F^{\frac{2}{3}} \mu^2 \Omega^{-\frac{4}{3}} T^{\frac{1}{3}} \right)^{-\frac{1}{2}}$ ,  $b_1 = \Omega \sqrt{db'_1}$ ,  $b_2 = 4Ln\sqrt{\Omega\mu} + \Omega\sqrt{d}(\sqrt{b'_5} + \sqrt{b'_1 b'_4})$ ,  $b_3 = 6\mu V_F^{\frac{1}{3}} \Omega^{\frac{1}{3}} n^{\frac{2}{3}}$ , 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

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

where  $i$  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:

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

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  $U_i(\frac{1}{i})$ . Thus, summing from  $i = 1$  to  $M$  and dividing by  $M$ , we have that

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

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  $\{i^t\}_{t=1}^T$ . Let  $\bar{U}^{\text{param}}(M)$  be the upper bound on the regret with respect to static initialization ,

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

where the sequence of initialization  $\{i^t\}_{t=1}^M$  are obtained by some online learning algorithm. Thus, we can proceed to bound  $\bar{R}(M, T)$  by  $\bar{U}^{\text{param}}(M) + \frac{1}{M} \sum_{i=1}^M U_i(\cdot)$ . 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  $\{\ell_i : \mathcal{X} \rightarrow \mathbb{R}\}_{i \geq 1}$ , follow-the-regularized leader (FTRL) plays

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

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  $\{U_i : \mathcal{X} \rightarrow \mathbb{R}\}_{i \in [M]}$ . 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,

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

where  $U_i(X; \{\xi^t\}_{t \in [T]})$  makes the dependence of (6) on the trajectory data  $\{\xi^t\}_{t \in [T]}$  explicit. Specifically, the constraint  $\xi_i^{t+1} \sim \text{Alg}^t(x, \square f_i)$  requires that the trajectory data  $\{\xi^t\}_{t \in [T]}$  is generated by  $\text{Alg}$  starting from the initial point  $x$ . We take the supremum over all possible realizations of the trajectories to remove the randomness in the definition of  $V_{\text{init}}^*$ . As seen from the definition above,  $V_{\text{init}}^*$  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  $V_{\text{init}}^*$ ).

**Theorem 4** *Let  $\{\xi_i^t\}_{i \in [M]}$  be obtained by running FTRML on the sequence of loss functions  $\{U_i\}_{i=1, \dots, M}$ , with initial point  $\phi \in \mathcal{X}$  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  $\phi$ . 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  $V_{\text{init}}^*$  to run FTRML. Note that similar results as above also hold if we replace FTRML 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:

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

where  $\psi_i^t = (\psi_{r,i}^t, \psi_{h,i}^t)$  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  $\psi_i^0$  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  $(\gamma, \tau)$ -approximate optimization oracle, which, for a given function  $\ell : \Psi \rightarrow \mathbb{R}$  and a  $d'$ -dimensional vector  $\sigma$ , 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  $\{\ell_i : \mathcal{X} \rightarrow \mathbb{R}\}_{i \geq 1}$ , follow-the-perturbed leader (FTPL) plays  $\psi_i = \mathcal{Q}_{\gamma,\tau}(\sum_{s < i} \ell_s(\psi) - \langle \sigma_s, \cdot \rangle)$ , where  $\sigma_s$  is a random perturbation such that its  $j$ -th coordinate  $\sigma_{s,j}$  is sampled from the exponential distribution  $\text{Exp}(\eta)$  with parameter  $\eta$  (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:

$$V_{\text{manifold}}(\psi) := \sup_{x \in \mathcal{X}, \{\zeta_i^{t+1} \sim \text{Alg}^t(\zeta, \square f_i)\}_{i \in [M], t \in [T-1]}} \frac{1}{M} \sum_{i=1}^M U'_i(\psi; \{\zeta_i^{t+1}\}_{t \in [T-1]}), \quad (11)$$

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  $\zeta_i^{t+1} \sim \text{Alg}^t(\zeta, \square f_i)$  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  $\{U_i : \Psi \rightarrow \mathbb{R}\}_{i \in [M]}$ , with appropriately chosen  $\eta$ , 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  $\gamma$  and  $\tau$  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  $\psi_i$ . This is possible since the upper bounds of (6) and (10) can be combined:

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

where  $\kappa \in [0, 1]$  is the weight. For a fixed  $\kappa$ , 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  $\psi_i^1$  (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  $\kappa$  to be an arbitrary number in  $[0, 1]$ , it is unclear if a better setting exists. Indeed, a higher value of  $\kappa$  weighs more on meta-learning the initial point (i.e., the  $\{U_i\}_{i \in [M]}$  sequence) while a lower value of  $\kappa$  weighs more on manifold meta-learning (i.e., the  $\{U'_i\}_{i \in [M]}$  sequence).

To this end, we consider adapting the weights  $\kappa$  together with FTRL on  $x_i^1$  and FTPML on  $\psi_i^1$ . Let  $U_i^{\text{sim}}(\kappa) = \kappa U_i(\frac{1}{i}) + (1 - \kappa) U'_i(\psi_i^1)$  be a function of  $\kappa$  conditioning on the values of  $U_i(\frac{1}{i})$  and  $U'_i(\psi_i^1)$ . Hence,  $U_i^{\text{sim}}$  is simply a linear function of  $\kappa$ . In the  $i$ -th meta update, we run the following parallel processes: (1) FTRL on  $\{U_m^{\text{sim}}\}_{m < i}$  to obtain  $\kappa_i$ , (2) FTRL on  $\{\kappa_m U_m\}_{m < i}$  to obtain  $x_i^1$ , and (3) FTRL on  $\{\kappa_m U'_m\}_{m < i}$  to obtain  $\psi_i^1$ . 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, FTRL, 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. For any  $\kappa \in (0, 1)$ , we have that*

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

From the bound, it may not seem obvious the theoretical improvement over meta-learning initialization or manifold independently, as one can change  $\kappa$  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)

---

- 1: **for**  $i = 1, \dots, M$  **do**
- 2:   **for**  $t = 1, \dots, T$  **do**
- 3:      $x_i^{t+1}, \psi_i^{t+1} = \text{LMRS}(x_i^t, \psi_i^t)$
- 4:   **end for**
- 5:   Evaluate  $U_i(\frac{1}{i})$  and  $U'_i(\psi_i^1)$  using (6) and (10), respectively
- 6:   Update the next task parameter and manifold initialization

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

- 7: **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. FTRL and FTPML algorithms are used to meta initialize  $x_i^1$  and  $\psi_i^1$  at the start of each new task, respectively. We use multi-layered perceptrons as parametric functions for the manifold. We set the manifold dimension to 50 for both environments without extensive tuning (the rationale for choosing this

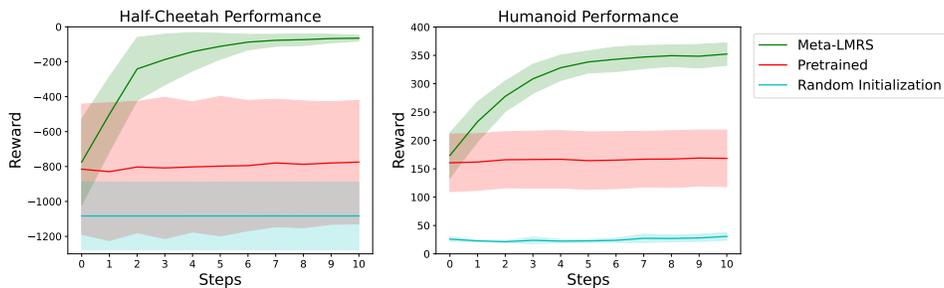


Figure 1: Performance comparison of Meta-LMRS with baselines. Both the average and standard deviation (shades) are reported across 10 independent runs.

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 and discussion about task similarity in 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.

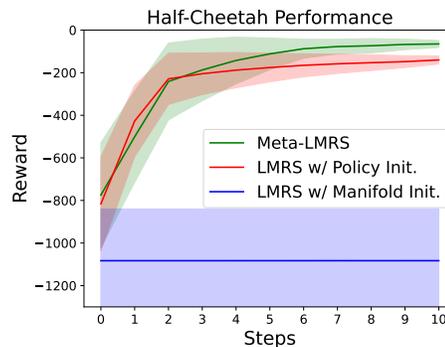


Figure 2: Comparison of incremental benefits of meta-initializing policy and manifold parameters.

## 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.
- 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.
- Brandon Amos. Tutorial on amortized optimization for learning to optimize over continuous domains. *arXiv preprint arXiv:2202.00665*, 2022.
- 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.
- 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.
- 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 Grazi, 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.
- 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.
- 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.
- 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, Subhansu 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.
- 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.
- 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.
- 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.
- 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](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.
- Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.
- 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.
- 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 Freitas. 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.
- 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.
- Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 2021.