Optimistic Meta-Gradients

Anonymous Author(s)
Affiliation
Address
email

Abstract

We study the connection between gradient-based meta-learning and convex optimisation. We observe that gradient descent with momentum is as a special case of meta-gradients, and building on recent results in optimisation, we prove convergence rates for meta-learning in the single task setting. While a meta-learned update rule can yield faster convergence up to constant factor,it is not sufficient for acceleration. Instead, some form of optimism is required. We show that optimism in meta-learning can be captured through the recently proposed Bootstrapped Meta-Gradient [9] method, providing deeper insight into its underlying mechanics.

1 Introduction

In meta-learning, a learner is using a parameterised algorithm to adapt to a given task. The parameters of the algorithm are then meta-learned by evaluating the learner's resulting performance [24, 10, 2]. As such, meta-learning features a complex interaction between the learner and the meta-learner. The **learner's problem** is to minimize the expected loss f of a stochastic objective by adapting its parameters $x \in \mathbb{R}^n$. The learner has an update rule φ at its disposal that generates new parameters $x_t = x_{t-1} + \varphi(x_{t-1}, w_t)$; we suppress data dependence to simplify notation. A simple example is when φ represents gradient descent with $w_t = \eta$ its step size, that is $\varphi(x_{t-1}, \eta) = -\eta \nabla f(x_{t-1})$ [16, 25]; several works have explored meta-learning other aspects of a gradient-based update rule [6, 20, 7, 29, 30, 9, 14, 21]. φ need not be limited to a gradient-based update, it can represent some algorithm implemented within a Recurrent Neural Network [24, 11, 1, 28].

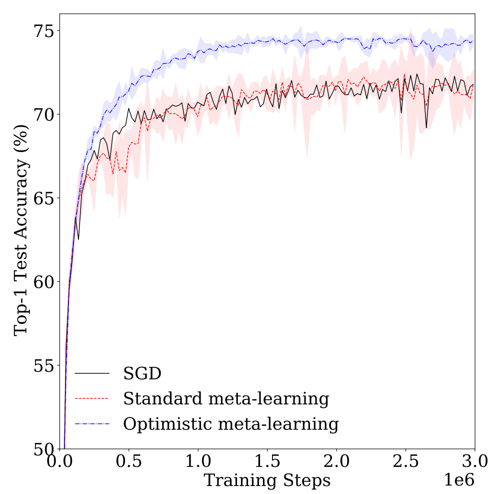

Figure 1: ImageNet. We compare training a 50-layer ResNet using SGD against variants that tune an element-wise learning rate online using standard meta-learning or optimistic meta-learning. Shading depicts 95% confidence intervals over 3 seeds.

The meta-learner's problem is to optimise the meta-parameters w_t to yield effective updates. In a typical (gradient-based) meta-learning setting, it does so by treating x_t as a function of w. Let h_t, defined by $h_t(w) = f(x_{t-1} + \varphi(x_{t-1}, w))$, denote the learner's post-update performance as a function of w. The learner and the meta-learner co-evolve according to

$$x_t = x_{t-1} + \varphi(x_{t-1}, w_t), \qquad w_{t+1} = w_t - \nabla h_t(w_t) = w_t - D\varphi(x_{t-1}, w_t)^T \nabla f(x_t),$$

where $D\varphi(x, w)$ denotes the Jacobian of φ with respect to w. The nested structure between these two updates makes it challenging to analyse meta-learning, in particular it depends heavily on the properties of the Jacobian. In practice, φ is highly complex and so $D\varphi$ is almost always intractable.

Submitted to the 6th Workshop on Meta-Learning at NeurIPS 2022, New Orleans. Do not distribute.

For this reason, the only theoretical results we are aware of specialise to the multi-task setting, where the learner must adapt to a new task f_t. Acceleration in these guarantees are driven entirely by the task distribution. That is, if all tasks are sufficiently similar, a meta-learned update can accelerate convergence. However, they do not yield acceleration in the absence of a task distribution.

This paper provides an alternative view. We study the classical convex optimisation setting of approximating the minimiser $\min_x f(x)$. We observe that setting the update rule equal to the gradient, i.e. $\varphi : (x, w) \mapsto w\nabla f(x)$, recovers gradient descent. Similarly, we show in Section 3 that φ can be chosen to recover gradient descent with momentum. This offers another view of meta-learning as a non-linear transformation of classical optimisation. An implication thereof is that a task distribution is not necessary for meta-learning. While there is ample empirical evidence to that effect [29, 30, 9, 15], we are only aware of theoretical results in the special case of meta-learned step sizes [16, 25].

Given f convex with Lipschitz smooth gradients, meta-learning affects the rate of convergence $O(\lambda/T)$ by a multiplicative factor λ that captures the smoothness of the update rule. To achieve accelerated convergence, $O(1/T^2)$, some form of *optimism* is required, typically in the form of a prediction of the next gradient. We consider optimism with meta-learning in the convex setting and prove accelerated rates of convergence, $O(\lambda/T^2)$. Again, meta-learning affects these bounds by a multiplicative factor. Our main contributions are as follows:

1. We show that meta-learning contains gradient descent with momentum (Heavy Ball [22]; Section 3) and Nesterov Acceleration [19] as special cases (Section 4).
2. We show that gradient-based meta-learning can be understood as a non-linear transformation of an underlying optimisation method (Section 3).
3. We establish rates of convergence for meta-learning in the convex setting (Section 3).
4. We show that optimism can be expressed through the recently proposed Bootstrapped Meta-Gradient method [BMG; 9]. Our analysis provides a first proof of convergence for BMG and highlights the underlying mechanics that enable faster learning with BMG (Section 4).

2 Meta-learning meets convex optimisation

Problem definition. This section defines the problem studied in this paper and introduces our notation. Let $f : \mathcal{X} \to \mathbb{R}$ be a proper and convex function. The problem of interest is to approximate the global minimum $\min_{x\in\mathcal{X}} f(x)$. We assume a global minimiser exists and is unique, defined by

$$x^* = \operatorname*{arg\,min}_{x\in\mathcal{X}} \; f(x). \tag{1}$$

We assume that $\mathcal{X} \subseteq \mathbb{R}^n$ is a closed, convex and non-empty set. f is differentiable and has Lipschitz smooth gradients with respect to a norm $\|\cdot\|$, meaning that there exists $L \in (0, \infty)$ such that $\|\nabla f(x) - \nabla f(y)\|_* \le L\|x - y\|$ for all $x, y \in \mathcal{X}$, where $\|\cdot\|_*$ is the dual norm of $\|\cdot\|$. We consider the noiseless setting for simplicity; our results carry over to the stochastic setting by replacing the key online-to-batch bound used in our analysis by its stochastic counterpart [13].

Algorithm. Let $[T] = \{1, 2, \ldots, T\}$. We are given weights $\{\alpha_t\}_{t=1}^T$, each $\alpha_t > 0$, and an initialisation $(\bar{x}_0, w_1) \in \mathcal{X} \times \mathcal{W}$. At each time $t \in [T]$, an update rule $\varphi : \mathcal{X} \times \mathcal{W} \to \mathcal{X}$ generates the update $x_t = \varphi(\bar{x}_{t-1}, w_t)$, where $\mathcal{W} \subseteq \mathbb{R}^m$ is closed, convex, and non-empty. We discuss φ momentarily. The algorithm maintains the online average

$$\bar{x}_t = \frac{x_{1:t}}{\alpha_{1:t}} = (1 - \rho_t)\bar{x}_{t-1} + \rho_t x_t, \tag{2}$$

where $x_{1:t} = \sum_{s=1}^t \alpha_s x_s$, $\alpha_{1:t} = \sum_{s=1}^t \alpha_s$, and $\rho_t = \alpha_t/\alpha_{1:t}$. Our goal is to establish conditions under which $\{\bar{x}_t\}_{t=1}^T$ converges to the minimiser x^*. While this moving average is not always used in practical applications, it is required for accelerated rates in online-to-batch conversion [26, 3, 13].

Convergence depends on how meta-parameters w_t are chosen. The meta-learner faces a sequence of losses $h_t : \mathcal{W} \to \mathbb{R}$ defined by the composition $h_t(w) = f((1 - \rho_t)\bar{x}_{t-1} + \rho_t\varphi(\bar{x}_{t-1}, w))$. This makes meta-learning a form of online gradient descent [17], which we can model under Follow-The-Regularized-Leader (FTRL; reviewed in Appendix D): given w_0, each w_t is chosen according to

$$w_{t+1} = \operatorname*{arg\,min}_{w\in\mathcal{W}} \left(\sum_{s=1}^t \alpha_s \langle \nabla h_s(w_s), w\rangle + \frac{1}{2\beta}\|w\|^2 \right). \tag{3}$$

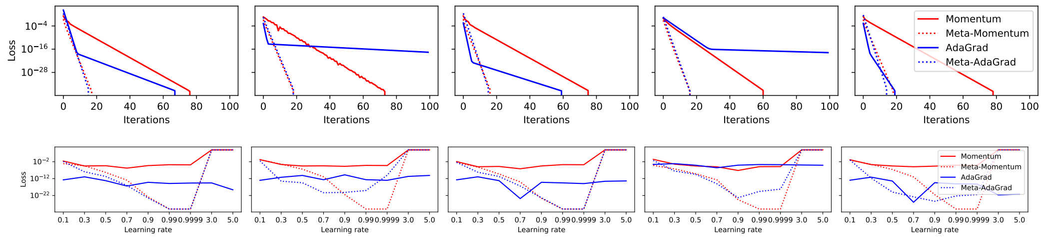

Figure 2: Convex Quadratic. We generate convex quadratic loss functions with ill-conditioning and compare gradient descent with momentum and AdaGrad to meta-learning variants. Meta-Momentum uses $\varphi : (x, w) \mapsto w \odot \nabla f(x)$ while Meta-AdaGrad uses $\varphi : (x, w) \mapsto \nabla f(x)/\sqrt{w}$, where division is element-wise. *Top:* loss per iteration for randomly sampled loss functions. *Bottom:* cumulative loss (regret) at the end of learning as a function of learning rate; details in Appendix B.

Note that this subsumes the standard meta-gradient; if $\|\cdot\|$ is the Euclidean norm, an interior solution to Eq. 3 yields $w_{t+1} = w_t - \alpha_t \beta \nabla h_t(w_t)$. It is straightforward to extend Eq. 3 to account for meta-updates that use AdaGrad-like [5] acceleration by altering the norms [12].

Update rule. It is not possible to prove convergence outside of the convex setting, since φ may reach a local minimum where it cannot yield better updates, but the updates are not sufficient to converge. Convexity means that each h_t must be convex, which requires that φ is affine in w (but may vary non-linearly in x). We also assume that φ is smooth with respect to $\|\cdot\|$, in the sense that it has bounded norm; for all $x \in \mathcal{X}$ and all $w \in \mathcal{W}$ we assume that there exists $\lambda \in (0, \infty)$ for which

$$\|D\varphi(x, w)^T \nabla f(x)\|_*^2 \leq \lambda \|\nabla f(x)\|_*^2.$$

These assumptions hold for any update rule up to first-order Taylor approximation error.

3 Meta-Gradients without Optimism

The main difference between classical optimisation and meta-learning is the introduction of the update rule φ. To see how this acts on optimisation, consider two special cases. If the update rule just return the gradient, $\varphi = \nabla f$, Eq. 3 reduces to gradient descent (with averaging). This inductive bias is fixed and does not change with experience, so acceleration is not possible: the rate of convergence is $O(1/\sqrt{T})$ [27]. The other extreme is an update rule that only depends on the meta-parameters, $\varphi(x, w) = w$. Here, the meta-learner has ultimate control and selects the next update without constraints. The only relevant inductive bias is contained in w. To see how this inductive bias is formed, suppose $\|\cdot\| = \|\cdot\|_2$ so that Eq. 3 yields $w_{t+1} = w_t - \alpha_t \rho_t \beta \nabla f(\bar{x}_t)$ (assuming an interior solution). Combining this with the moving average in Eq. 2, we may write the learner's iterates as

$$\bar{x}_t = \bar{x}_{t-1} + \tilde{\rho}_t \left(\bar{x}_{t-1} - \bar{x}_{t-2}\right) - \tilde{\beta}_t \nabla f(\bar{x}_{t-1}),$$

where each $\tilde{\rho}_t = \rho_t \frac{1-\rho_{t-1}}{\rho_{t-1}}$ and $\tilde{\beta}_t = \alpha_t \rho_t \beta$; setting $\beta = 1/(2L)$ and each $\alpha_t = t$ yields $\tilde{\rho}_t = \frac{t-2}{t+1}$ and $\tilde{\beta}_t = t/(4(t+1)L)$. Hence, the canonical momentum algorithm, Polyak's Heavy-Ball method [22], is obtained as the special case of meta-learning under the update rule $\varphi : (x, w) \mapsto w$. Because Heavy Ball carries momentum from past updates, it can encode a model of the learning dynamics that leads to faster convergence, on the order $O(1/T)$. The implication of this is that the dynamics of meta-learning are fundamentally momentum-based and thus learns an inductive bias in the same cumulative manner. This similarity is clear from our theoretical analysis, summarised in the following result.

Theorem 1 (*Informal*)**.** *Set* $\alpha_t = 1$ *and* $\beta = \frac{1}{\lambda L}$. *Then* $f(\bar{x}_T) - f(x^*) \leq \frac{\lambda L \operatorname{diam}(\mathcal{W})}{T}$.

Details: Appendix E. Compared to Heavy Ball, meta-learning introduces a constant λ that captures the smoothness of the update rule. Hence, while meta-learning does not achieve better scaling in T through φ, it can improve upon classical optimisation by a constant factor if $\lambda < 1$.

That meta-learning can improve upon momentum is borne out experimentally. In Figure 2, we consider the problem of minimizing a convex quadratic $f : x \mapsto \langle x, Qx \rangle$, where $Q \in \mathbb{R}^{n \times n}$ is PSD but ill-conditioned. We compare momentum to a meta-learned step-size, i.e. $\varphi : (x, w) \mapsto w \odot \nabla f(x)$, where $\odot$ is the Hadamard product. Across randomly sampled Q matrices (details: Appendix B), we find that introducing a non-linearity φ leads to a sizeable improvement in the rate of convergence. We also compare AdaGrad to a meta-learned version, $\varphi : (x, w) \mapsto \nabla f(x)/\sqrt{w}$, where division is element-wise. While AdaGrad is a stronger baseline on account of being parameter-free, we find that meta-learning the scale vector consistently leads to faster convergence.

4 Meta-Gradients with Optimism

It is well known that minimizing a smooth convex function admits convergence rates of $O(1/T^2)$. Our analysis of meta-learning does not achieve these rates. Previous work indicate that we should not expect it to either; to achieve the theoretical lower-limit of $O(1/T^2)$, some form of *optimism* (reviewed in Appendix D) is required. A typical form of optimism is to predict the next gradient. This is how Nesterov Acceleration operates [19], and is the reason for its $O(1/T^2)$ convergence guarantee.

From our perspective, meta-learning is a non-linear transformation of the iterate x. Hence, we should expect optimism to play a similarly crucial role. Formally, optimism comes in the form of *hint functions* $\{\tilde{g}_t\}_{t=1}^T$, each $\tilde{g}_t \in \mathbb{R}^m$, that are revealed to the meta-learner prior to selecting w_{t+1}. These hints give rise to *Optimistic Meta-Learning* (OML) via meta-updates

$$w_{t+1} = \arg\min_{w \in \mathcal{W}} \left(\alpha_{t+1} \tilde{g}_{t+1} + \sum_{s=1}^{t} \alpha_s \langle \nabla h_s(w_s), w \rangle + \frac{1}{2\beta_t} \|w\|^2 \right). \tag{4}$$

If the hints are accurate, meta-learning with optimism can achieve an accelerated rate of $O(\tilde{\lambda}/T^2)$, where $\tilde{\lambda}$ is a constant that characterises the smoothness of φ, akin to λ. Again, we find that meta-learning behaves as a non-linear transformation of classical optimism and its rate of convergence is governed by the geometry it induces. We summarise this result in the following result.

Theorem 2 (Informal)**.** *Let each hint be given by $\tilde{g}_{t+1} = D\varphi(\bar{x}_{t-1}, w_t)^T \nabla f(\bar{x}_t)$. Assume that φ is sufficiently smooth. Set $\alpha_t = t$ and $\beta_t = \frac{t-1}{2t\tilde{\lambda}L}$, then $f(\bar{x}_T) - f(x^*) \leq \frac{4\tilde{\lambda}L \operatorname{diam}(\mathcal{W})}{T^2-1}$.*

Details: Appendix E. These predictions hold empirically in a non-convex setting. We train a 50-layer ResNet using either SGD with a fixed learning rate, or an update rule that adapts a per-parameter learning rate online, $\varphi : (x, w) \mapsto w \odot \nabla f(x)$. We compare the standard meta-learning approach without optimism to optimistic meta-learning. Figure 1 shows that optimism is critical for meta-learning to achieve acceleration, as predicted by theory (experiment details in Appendix C).

5 Bootstrapped Meta-Gradients as a form of Optimism

Given Theorem 2, it is of interest to study practical ways of implementing optimism in meta-learning. We study a recently proposed variant of meta-gradients, *Bootstrapped Meta-Gradients (BMG)* [8]. Here, we present an informal comparison, see Appendix G for a complete derivation. Instead of directly minimising the loss f, the meta-objective in BMG is the distance between the meta-learner's output x_t and a desired *target* z_t. The target is computed by unrolling the meta-learner for a further number of steps, thus implicitly embodying a form of optimism, before a gradient step is taken: $z_t = x_t + \varphi(x_t, w_t) - \nabla f(x_t + \varphi(x_t, w_t))$. This encodes optimism via φ because it encourages the meta-learner to build up momentum (i.e. to accumulate past updates). To see how BMG arises as a form of optimism, we turn to AO-FTRL (Eq. 4). Choose hints $\tilde{g}_{t+1} = D\varphi(\bar{x}_{t-1}, w_t)^T \tilde{y}_{t+1}$ for some $\tilde{y}_{t+1} \in \mathbb{R}^n$ and set $\|\cdot\| = \|\cdot\|_2$; assuming an interior solution, Eq. 4 yields

$$w_{t+1} = w_t - \underbrace{D\varphi(\bar{x}_{t-1}, w_t)^T (\alpha_{t+1} \tilde{y}_{t+1} + \alpha_t \nabla f(\bar{x}_t))}_{\text{BMG update}} + \underbrace{\alpha_t D\varphi(\bar{x}_{t-2}, w_{t-1})^T \tilde{y}_t}_{\text{FTRL error correction}}. \tag{5}$$

Hence, BMG encodes very similar dynamics to those of AO-FTRL in Eq. 4. An immediate implication of this is that the hints in Corollary 1 can be expressed as targets in BMG, and hence if BMG satisfies the assumptions involved, it converges at a rate $O(\tilde{\lambda}/T^2)$.

6 Conclusion

This paper explores a connection between convex optimisation and meta-learning. We find that a meta-learned update rule cannot generate a better dependence on the horizon T, it can improve upon classical optimisation up to a constant factor. An implication of our analysis is that some form of optimism is required for acceleration. The recently proposed BMG method provides one way of incorporating optimism in practical applications.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

References

[1] M. Andrychowicz, M. Denil, S. Gómez, M. W. Hoffman, D. Pfau, T. Schaul, and N. de Freitas. Learning to Learn by Gradient Descent by Gradient Descent. In *Advances in Neural Information Processing Systems*, 2016.

[2] Y. Bengio, S. Bengio, and J. Cloutier. *Learning a Synaptic Learning Rule*. Université de Montréal, Département d'informatique et de recherche opérationnelle, 1991.

[3] A. Cutkosky. Anytime Online-to-Batch, Optimism and Acceleration. In *International Conference on Machine Learning*, 2019.

[4] O. Dekel, A. Flajolet, N. Haghtalab, and P. Jaillet. Online learning with a hint. In *Advances in Neural Information Processing Systems*, 2017.

[5] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(61):2121–2159, 2011.

[6] C. Finn, P. Abbeel, and S. Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *International Conference on Machine Learning*, 2017.

[7] S. Flennerhag, P. G. Moreno, N. D. Lawrence, and A. Damianou. Transferring Knowledge across Learning Processes. In *International Conference on Learning Representations*, 2019.

[8] S. Flennerhag, Y. Schroecker, T. Zahavy, H. van Hasselt, D. Silver, and S. Singh. Bootstrapped Meta-Learning. *arXiv preprint arXiv:2109.04504*, 2021.

[9] S. Flennerhag, Y. Schroecker, T. Zahavy, H. van Hasselt, D. Silver, and S. Singh. Bootstrapped Meta-Learning. In *International Conference on Learning Representations*, 2022.

[10] G. E. Hinton and D. C. Plaut. Using Fast Weights to Deblur Old Memories. In *Cognitive Science Society*, 1987.

[11] S. Hochreiter, A. S. Younger, and P. R. Conwell. Learning To Learn Using Gradient Descent. In *International Conference on Artificial Neural Networks*, 2001.

[12] P. Joulani, A. György, and C. Szepesvári. A modular analysis of adaptive (non-) convex optimization: Optimism, composite objectives, and variational bounds. *Journal of Machine Learning Research*, 1:40, 2017.

[13] P. Joulani, A. Raj, A. Gyorgy, and C. Szepesvári. A Simpler Approach to Accelerated Optimization: Iterative Averaging Meets Optimism. In *International Conference on Machine Learning*, 2020.

[14] L. Kirsch, S. van Steenkiste, and J. Schmidhuber. Improving Generalization in Meta Reinforcement Learning Using Learned Objectives. *arXiv preprint arXiv:1910.04098*, 2019.

[15] J. Luketina, S. Flennerhag, Y. Schroecker, D. Abel, T. Zahavy, and S. Singh. Meta-gradients in non-stationary environments. In *ICLR Workshop on Agent Learning in Open-Endedness*, 2022.

[16] A. R. Mahmood, R. S. Sutton, T. Degris, and P. M. Pilarski. Tuning-Free Step-Size Adaptation. In *ICASSP*, 2012.

[17] H. B. McMahan. A survey of algorithms and analysis for adaptive online learning. *The Journal of Machine Learning Research*, 18(1):3117–3166, 2017.

[18] M. Mohri and S. Yang. Accelerating Online Convex Optimization via Adaptive Prediction. In *International Conference on Artificial Intelligence and Statistics*, 2016.

[19] Y. E. Nesterov. A method for solving the convex programming problem with convergence rate o (1/kˆ 2). In *Dokl. akad. nauk Sssr*, volume 269, pages 543–547, 1983.

[20] A. Nichol, J. Achiam, and J. Schulman. On First-Order Meta-Learning Algorithms. *arXiv preprint ArXiv:1803.02999*, 2018.

[21] J. Oh, M. Hessel, W. M. Czarnecki, Z. Xu, H. P. van Hasselt, S. Singh, and D. Silver. Discovering Reinforcement Learning Algorithms. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

[22] B. T. Polyak. Some Methods of Speeding up the Convergence of Iteration Methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

[23] S. Rakhlin and K. Sridharan. Optimization, Learning, and Games with Predictable Sequences. In *Advances in Neural Information Processing Systems*, 2013.

[24] J. Schmidhuber. *Evolutionary Principles in Self-Referential Learning*. PhD thesis, Technische Universität München, 1987.

[25] T. van Erven and W. M. Koolen. MetaGrad: Multiple Learning Rates in Online Learning. In *Advances in Neural Information Processing Systems*, 2016.

[26] J.-K. Wang and J. Abernethy. Acceleration through Optimistic No-Regret Dynamics. *arXiv preprint arXiv:1807.10455*, 2018.

[27] J.-K. Wang, J. Abernethy, and K. Y. Levy. No-regret dynamics in the fenchel game: A unified framework for algorithmic convex optimization. *arXiv preprint arXiv:2111.11309*, 2021.

[28] J. X. Wang, Z. Kurth-Nelson, D. Tirumala, H. Soyer, J. Z. Leibo, R. Munos, C. Blundell, D. Kumaran, and M. Botvinick. Learning to Reinforcement Learn. In *Annual Meeting of the Cognitive Science Society*, 2016.

[29] Z. Xu, H. P. van Hasselt, and D. Silver. Meta-Gradient Reinforcement Learning. In *Advances in Neural Information Processing Systems*, 2018.

[30] T. Zahavy, Z. Xu, V. Veeriah, M. Hessel, J. Oh, H. P. van Hasselt, D. Silver, and S. Singh. A Self-Tuning Actor-Critic Algorithm. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

[31] M. Zinkevich. Online Convex Programming and Generalized Infinitesimal Gradient Ascent. In *International Conference on Machine Learning*, 2003.

Appendix

A Notation

Table 1: Notation

Indices	
t	Iteration index: $t \in \{1, ..., T\}$.
T	Total number of iterations.
$[T]$	The set $\{1, 2, \ldots, T\}$.
i	Component index: x^i is the ith component of $x = (x^1, \ldots, x^n)$.
$\alpha_{a:b}$	Sum of weights: $\alpha_{a:b} = \sum_{s=a}^{b} \alpha_s$
$x_{a:b}$	Weighted sum: $x_{a:b} = \sum_{s=a}^{b} \alpha_s x_s$
$\bar{x}_{a:b}$	Weighted average: $\bar{x}_{a:b} = x_{a:b}/\alpha_{a:b}$
Parameters	
$x^* \in \mathcal{X}$	Minimiser of f.
$x_t \in \mathcal{X}$	Parameter at time t
$\bar{x}_t \in \mathcal{X}$	Moving average of $\{x_s\}_{s=1}^{t}$ under weights $\{\alpha_s\}_{s=1}^{t}$.
$\rho_t \in (0, \infty)$	Moving average coefficient $\alpha_t/\alpha_{1:t}$.
$w_t \in \mathcal{W}$	Meta parameters
$w^* \in \mathcal{X}$	$w \in \mathcal{W}$ that retains regret with smallest norm $\|w\|$.
$\alpha_t \in (0, \infty)$	Weight coefficients
$\beta_t \in (0, \infty)$	Meta-learning rate
Maps	
$f : \mathcal{X} \to \mathbb{R}$	Objective function
$\|\cdot\| : \mathcal{X} \to \mathbb{R}$	Norm on $\mathcal{X}$.
$\|\cdot\|_* : \mathcal{X}^* \to \mathbb{R}$	Dual norm of $\|\cdot\|$.
$h_t : \mathcal{W} \to \mathbb{R}$	Online loss faced by the meta learner
$R^x(T)$	Regret of $\{x_t\}_{t=1}^{T}$ against x^*: $R^x(T) \coloneqq \sum_{t=1}^{T} \alpha_t \langle \nabla f(\bar{x}_t), x_t - x^* \rangle$.
$R^w(T)$	$R^w(T) \coloneqq \sum_{t=1}^{T} \alpha_t \langle \nabla f(\bar{x}_t), \varphi(\bar{x}_{t-1}, w_t) - \varphi(\bar{x}_{t-1}, w^*) \rangle$.
$\varphi : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$	Generic update rule used in practice
$D\varphi(x, \cdot) : \mathbb{R}^m \to \mathbb{R}^{n \times m}$	Jacobian of φ w.r.t. its second argument, evaluated at $x \in \mathbb{R}^n$.
$\varphi : \mathcal{X} \times \mathcal{W} \to \mathcal{X}$	Update rule in convex setting
$D\varphi(x, \cdot) : \mathcal{W} \to \mathbb{R}^{n \times m}$	Jacobian of φ w.r.t. its second argument, evaluated at $x \in \mathcal{X}$.
$B^\mu : \mathbb{R}^n \times \mathbb{R}^n \to [0, \infty)$	Bregman divergence under $\mu : \mathbb{R}^n \to \mathbb{R}$.
$\mu : \mathbb{R}^n \to \mathbb{R}$	Convex distance generating function.

Table 2: Hyper-parameter sweep on Convex Quadratics. All algorithms are tuned for learning rate and initialisation of w. Baselines are tuned for decay rate; meta-learned variant are tuned for the meta-learning rate.

Learning rate	[.1, .3, .7, .9, 3., 5.]
w init scale	[0., 0.3, 1., 3., 10., 30.]
Decay rate / Meta-learning rate	[0.001, 0.003, 0.01, .03, .1, .3, 1., 3., 10., 30.]

B Convex Quadratic Experiments

Loss function. We consider the problem of minimising a convex quadratic loss functions $f : \mathbb{R}^2 \to \mathbb{R}$ of the form $f(x) = x^T Q x$, where Q is randomly sampled as follows. We sample a random orthogonal matrix U from the Haar distribution `scipy.stats.ortho_group`. We construct a diagonal matrix of eigenvalues, ranked smallest to largest, with $\lambda_i = i^2$. Hence, the first dimension has an eigenvalue 1 and the second dimension has eigenvalue 4. The matrix Q is given by $U^T \operatorname{diag}(\lambda_1, \dots, \lambda_n) U$.

Protocol. Given that the solution is always $(0, 0)$, this experiment revolves around understanding how different algorithms deal with curvature. Given symmetry in the solution and ill-conditioning, we fix the initialisation to $x_0 = (4, 4)$ for all sampled Qs and all algorithms and train for 100 iterations. For each Q and each algorithm, we sweep over the learning rate, decay rate, and the initialization of w see Table 2. For each method, we then report the results for the combination of hyper parameters that performed the best.

Results. We report the learning curves for the best hyper-parameter choice for 5 randomly sampled problems in the top row of Figure 2 (columns correspond to different Q). We also study the sensitivity of each algorithm to the learning rate in the bottom row Figure 2. For each learning rate, we report the cumulative loss during training. While baselines are relatively insensitive to hyper-parameter choice, meta-learned improve for certain choices, but are never worse than baselines.

C Imagenet Experiments

Protocol. We train a 50-layer ResNet following the Haiku example, available at https://github.com/deepmind/dm-haiku/blob/main/examples/imagenet. We modify the default setting to run with SGD. We compare default SGD to variants that meta-learn an element-wise learning rate online, i.e. $(x, w) \mapsto w \odot \nabla f(x)$. For each variant, we sweep over the learning rate (for SGD) or meta-learning rate. We report results for the best hyper-parameter over three independent runs.

Standard meta-learning. In the standard meta-learning setting, we apply the update rule once before differentiating w.r.t. the meta-parameters. That is, the meta-update takes the form $w_{t+1} = w_t - \beta \nabla h_t(w_t)$, where $h_t = f(x_t + w_t \odot \nabla f(x_t))$. Because the update rule is linear in w, we can compute the meta-gradient analytically:

$$\nabla h_t(w_t) = \nabla_w f(x + \varphi(x, w)) = D\varphi(x, w)^T \nabla f(x') = \nabla f(x) \odot \nabla f(x'),$$

where $x' = x + \varphi(x, w)$. Hence, we can compute the meta-updates in Algorithm 1 manually as $w_{t+1} = \max\{w_t - \beta \nabla f(x_t) \odot \nabla f(x_{t+1}), 0.\}$, where we introduce the $\max$ operator on an element-wise basis to avoid negative learning rates. Empirically, this was important to stabilize training.

Optimistic meta-learning. For optimistic meta-learning, we proceed much in the same way, but include a gradient prediction $\tilde{g}_{t+1}$. For our prediction, we use the previous gradient, $\nabla f(x_{t+1})$, as our prediction. Following Eq. 5, this yields meta-updates of the form

$$w_{t+1} = \max \Big\{ w_t - \beta \nabla f(x_{t+1}) \odot (\nabla f(x_{t+1}) + \nabla f(x_t)) - \nabla f(x_t) \odot \nabla f(x_t), 0. \Big\}.$$

Results. We report Top-1 accuracy on the held-out test set as a function of training steps in Figure 1. Tuning the learning rate does not yield any statistically significant improvements under standard meta-learning. However, with optimistic meta-learning, we obtain a significant acceleration as well as improved final performance, increasing the mean final top-1 accuracy from 72% to 75%.

Table 3: Hyper-parameter sweep on Imagenet.

(Meta-)learning rate	[0.001, 0.01, 0.02, 0.05, 0.1]

D Background

In this section, we present analytical tools from the optimisation literature that we build upon. In a standard optimisation setting, there is no update rule φ; instead, the iterates x_t are generated by a gradient-based algorithm, akin to Eq. 3. In particular, our setting reduces to standard optimisation if φ is defined by $\varphi : (x, w) \mapsto w$, in which case $x_t = w_t$. A common approach to analysis is to treat the iterates $x_1, x_2, \ldots$ as generated by an online learning algorithm over online losses, obtain a regret guarantee for the sequence, and use online-to-batch conversion to obtain a rate of convergence.

Online Optimisation. In online convex optimisation [31], a learner is given a convex decision set $\mathcal{U}$ and faces a sequence of convex loss functions $\{\alpha_t f_t\}_{t=1}^T$. At each time $t \in [T]$, it must make a prediction u_t prior to observing $\alpha_t f_t$, after which it incurs a loss $\alpha_t f_t(u_t)$ and receives a signal—either $\alpha_t f_t$ itself or a (sub-)gradient of $\alpha_t f_t(u_t)$. The learner's goal is to minimise *regret*, $R(T) \coloneqq \sum_{t=1}^T \alpha_t (f_t(u_t) - f_t(u))$, against a comparator $u \in \mathcal{U}$. An important property of a convex function f is $f(u') - f(u) \leq \langle \nabla f(u'), u' - u \rangle$. Hence, the regret is largest under linear losses: $\sum_{t=1}^T \alpha_t (f_t(u_t) - f_t(u)) \leq \sum_{t=1}^T \alpha_t \langle \nabla f_t(u_t), u_t - u \rangle$. For this reason, it is sufficient to consider regret under linear loss functions. An algorithm has sublinear regret if $\lim_{T \to \infty} R(T)/T = 0$.

FTRL & AO-FTRL. The meta-update in Eq. 3 is an instance of Follow-The-Regularised-Leader (FTRL) under linear losses. In Appendix G, we show that BMG is an instance of the Adaptive-Optimistic FTRL (AO-FTRL), which is an extension due to [23, 18, 13, 27]. In AO-FTRL, we have a strongly convex regulariser $\|\cdot\|^2$. FTRL and AO-FTRL sets the first prediction u_1 to minimise $\|\cdot\|^2$. Given linear losses $\{g_s\}_{s=1}^{t-1}$ and learning rates $\{\beta_t\}_{t=1}^T$, each $\beta_t > 0$, the algorithm proceeds according to

$$u_t = \operatorname*{arg\,min}_{u \in \mathcal{U}} \left(\alpha_t \langle \tilde{g}_t, u \rangle + \sum_{s=1}^{t-1} \alpha_s \langle g_s, u \rangle + \frac{1}{2\beta_t} \|u\|^2 \right), \tag{6}$$

where each $\tilde{g}_t$ is a "hint" that enables optimistic learning [23, 18]; setting $\tilde{g}_t = 0$ recovers the original FTRL algorithm. The goal of a hint is to predict the next loss vector g_t; if the predictions are accurate AO-FTRL can achieve lower regret than its non-optimistic counter-part. Since $\|\cdot\|^2$ is strongly convex, FTRL is well defined in the sense that the minimiser exists, is unique and finite [17]. The regret of FTRL and AO-FTRL against any comparator $u \in \mathcal{U}$ can be upper-bounded by

$$R(T) = \sum_{t=1}^T \alpha_t \langle g_t, u_t - u \rangle \leq \frac{\|u\|^2}{2\beta_T} + \frac{1}{2} \sum_{t=1}^T \alpha_t^2 \beta_t \, \|g_t - \tilde{g}_t\|_*^2 \,. \tag{7}$$

Hence, hints that predict g_t well can reduce the regret substantially. Without hints, FTRL can guarantee $O(\sqrt{T})$ regret (for non strongly convex loss functions). However, [4] show that under linear losses, if hints are weakly positively correlated—defined as $\langle g_t, \tilde{g}_t \rangle \geq \epsilon \|g_t\|^2$ for some $\epsilon > 0$—then the regret guarantee improves to $O(\log T)$, even for non strongly-convex loss functions. We believe optimism provides an exciting opportunity for novel forms of meta-learning. Finally, we note that these regret bounds (and hence our analysis) can be extended to stochastic optimisation [18, 12].

Online-to-batch conversion. The main idea behind online to batch conversion is that, for f convex, Jensen's inequality gives $f(\bar{x}_T) - f(x^*) \leq \sum_{t=1}^T \alpha_t \langle \nabla f(x_t), x_t - x^* \rangle / \alpha_{1:T}$. Hence, one can provide a convergence rate by first establishing the regret of the algorithm that generates x_t, from which one obtains the convergence rate of the moving average of iterates. Applying this naively yields $O(1/T)$ rate of convergence. In recent work, [3] shows that one can upper-bound the sub-optimality gap by instead querying the gradient gradient at the average iterate, $f(\bar{x}_T) - f(x^*) \leq \sum_{t=1}^T \alpha_t \langle \nabla f(\bar{x}_t), x_t - x^* \rangle / \alpha_{1:T}$, which can yield faster rates of convergence. Recently, [13] tightened the analysis and proved that the sub-optimality gap can be bounded by

$$\begin{aligned} &f(\bar{x}_T) - f(x^*) \leq \\ &\frac{1}{\alpha_{1:T}} \left(R^x(T) - \frac{\alpha_t}{2L} \|\nabla f(\bar{x}_t) - \nabla f(x^*)\|_*^2 - \frac{\alpha_{1:t-1}}{2L} \|\nabla f(\bar{x}_{t-1}) - \nabla f(\bar{x}_t)\|_*^2 \right), \end{aligned} \tag{8}$$

were we define $R^x(T) \coloneqq \sum_{t=1}^{T} \alpha_t \langle \nabla f(\bar{x}_t), x_t - x^* \rangle$ as the regret of the sequence $\{x_t\}_{t=1}^T$ against the comparator x^*. With this machinery in place, we now turn to deriving our main results.

Algorithm 1: Meta-learning in practice.
input : Weights $\{\beta_t\}_{t=1}^T$
input : Update rule φ
input : Initialisation (x_0, w_1)
for $t = 1, 2, \dots, T$:
$\quad x_t = x_{t-1} + \varphi(x_{t-1}, w_t)$
$\quad h_t(\cdot) = f(x_{t-1} + \rho_t \varphi(x_{t-1}, \cdot))$
$\quad w_{t+1} = w_t - \beta_t \nabla h_t(w_t)$
return x_T

Algorithm 2: Meta-learning in the convex setting.
input : Weights $\{\alpha_t\}_{t=1}^T, \{\beta_t\}_{t=1}^T$
input : Update rule φ
input : Initialisation $(\bar{x}_0, w_1)$
for $t = 1, 2, \dots, T$:
$\quad x_t = \varphi(\bar{x}_{t-1}, w_t)$
$\quad \bar{x}_t = (1 - \alpha_t/\alpha_{1:t})\bar{x}_{t-1} + (\alpha_t/\alpha_{1:t})x_t$
$\quad g_t = D\varphi(\bar{x}_{t-1}, w_t)^T \nabla f(\bar{x}_t)$
$\quad w_{t+1} = \arg\min_{w \in \mathcal{W}} \sum_{s=1}^t \alpha_s \langle g_s, w \rangle + \frac{1}{2\beta_t}\|w\|^2$
return $\bar{x}_T$

E Analysis

The central challenge in applying Eq. 8 to Algorithm 2 is that the iterates x_t are generated under the update rule φ. Hence, we cannot apply standard regret bounds directly. Instead, observe that

$$
\begin{aligned}
R^x(T) &= \sum_{t=1}^{T} \alpha_t \langle \nabla f(\bar{x}_t), x_t - x^* \rangle = \sum_{t=1}^{T} \alpha_t \langle \nabla f(\bar{x}_t), \varphi(\bar{x}_{t-1}, w_t) - x^* \rangle \\
&= \sum_{t=1}^{T} \alpha_t \langle \nabla f(\bar{x}_t), \varphi(\bar{x}_{t-1}, w_t) - \varphi(\bar{x}_{t-1}, w^*) \rangle + \sum_{t=1}^{T} \alpha_t \langle \nabla f(\bar{x}_t), \varphi(\bar{x}_{t-1}, w^*) - x^* \rangle.
\end{aligned}
$$

The first term in the final inequality can be understood as the regret under convex losses $\ell_t(\cdot) = \alpha_t \langle \nabla f(\bar{x}_t), \varphi(\bar{x}_{t-1}, \cdot) \rangle$. Since φ is affine, ℓ_t is convex and thus this regret can be upper-bounded by linearising the losses. The linearisation reads $\langle D\varphi(\bar{x}_{t-1}, w_t)^T \nabla f(\bar{x}_t), \cdot \rangle$, which is identical the linear losses $\langle \nabla h_t(w_t), \cdot \rangle$ faced by the meta-learner in Eq. 3. Hence, we can upper-bound this term by the of the meta-learner:

$$
R^w(T) \coloneqq \sum_{t=1}^{T} \alpha_t \langle \nabla h_t(w_t), w_t - w^* \rangle \geq \sum_{t=1}^{T} \alpha_t \langle \nabla f(\bar{x}_t), \varphi(\bar{x}_{t-1}, w_t) - \varphi(\bar{x}_{t-1}, w^*) \rangle.
$$

Hence, we have that

$$
R^x(T) \leq R^w(T) + \sum_{t=1}^{T} \alpha_t \langle \nabla f(\bar{x}_t), \varphi(\bar{x}_{t-1}, w^*) - x^* \rangle. \tag{9}
$$

For the last term to be negative we need the relative power of the comparator w^* to be greater than that the comparator x^*. Intuitively, the comparator x^* is non-adaptive. It must make one choice x^* and suffer the average loss. In contrast, the comparator w^* becomes adaptive under the update rule; it can only choose one w^*, but on each round it plays $\varphi(\bar{x}_{t-1}, w^*)$. If φ is sufficiently flexible, this gives the comparator w^* more power than x^*, and hence it can force the meta-learner to suffer greater regret. When this is the case, we say that regret is retained when moving from x^* to w^*. As long as φ is not degenerate, this is typically easy to satisfy by making $\mathcal{W}$ sufficiently large.

Definition 1. *Given f, $\{\alpha_t\}_{t=1}^T$, and $\{x_t\}_{t=1}^T$, an update rule $\varphi : \mathcal{X} \times \mathcal{W} \to \mathcal{X}$* preserves regret *if there exists a comparator $w \in \mathcal{W}$ that satisfies*

$$
\sum_{t=1}^{T} \alpha_t \langle \varphi(\bar{x}_{t-1}, w), \nabla f(\bar{x}_t) \rangle \leq \sum_{t=1}^{T} \alpha_t \langle x^*, \nabla f(\bar{x}_t) \rangle. \tag{10}
$$

If such w exists, let w^* denote the comparator with smallest norm $\|w\|$.

Lemma 1. *Given f, $\{\alpha_t\}_{t=1}^T$, and $\{x_t\}_{t=1}^T$, if φ preserves regret, then*

$$
R^x(T) = \sum_{t=1}^{T} \alpha_t \langle \nabla f(\bar{x}_t), x_t - x^* \rangle \leq \sum_{t=1}^{T} \alpha_t \langle \nabla f(\bar{x}_t), \varphi(\bar{x}_{t-1}, w_t) - \varphi(\bar{x}_{t-1}, w^*) \rangle = R^w(T).
$$

Proof: Appendix F. From Eq. 10, it is clear that for φ to retain regret, it must admit a parameterisation that correlates negatively with the gradient. In other words, φ must be able to behave as a gradient descent algorithm. However, this must not hold on every step, only sufficiently often. For instance, $\varphi(x, \cdot)$ affine can be made to satisfy this condition if $\mathcal{X}$ and $\mathcal{W}$ are chosen appropriately.

Theorem 3. *Let φ preserve regret and satisfies the assumptions in Section 2. Then*

$$f(\bar{x}_T) - f(x^*) \leq \frac{1}{\alpha_{1:T}} \left(\frac{\|w^*\|^2}{\beta} + \sum_{t=1}^{T} \frac{\lambda \beta \alpha_t^2}{2} \|\nabla f(\bar{x}_t)\|_*^2 - \frac{\alpha_t}{2L} \|\nabla f(\bar{x}_t) - \nabla f(x^*)\|_*^2 - \frac{\alpha_{1:t-1}}{2L} \|\nabla f(\bar{x}_{t-1}) - \nabla f(\bar{x}_t)\|_*^2 \right).$$

If x^* is a global minimiser of f, setting $\alpha_t = 1$ and $\beta = \frac{1}{\lambda L}$ yields $f(\bar{x}_T) - f(x^*) \leq \frac{\lambda L \operatorname{diam}(\mathcal{W})}{T}$.

The proof formalises the example given above and is deferred to Appendix F.

Algorithm 3: BMG in practice.

input : Weights $\{\beta_t\}_{t=1}^T$
input : Update rule φ
input : Target oracle
input : Initialisation (x_0, w_1)
for $t = 1, 2, \ldots, T$:
  $x_t = x_{t-1} + \varphi(x_{t-1}, w_t)$
  Query z_t from target oracle
  $d_t(\cdot) = \|z_t - x_t + \varphi(x_t, \cdot)\|^2$
  $w_{t+1} = w_t - \beta_t \nabla d_t(w_t)$
return x_T

Algorithm 4: Convex optimistic meta-learning.

input : Weights $\{\alpha_t\}_{t=1}^T, \{\beta_t\}_{t=1}^T$
input : Update rule φ
input : Hints $\{\tilde{g}_t\}_{t=1}^T$
input : Initialisation $(\bar{x}_0, w_1)$
for $t = 1, 2, \ldots, T$:
  $x_t = \varphi(\bar{x}_{t-1}, w_t)$
  $\bar{x}_t = (1 - \alpha_t/\alpha_{1:t})\bar{x}_{t-1} + (\alpha_t/\alpha_{1:t})x_t$
  $g_t = D\varphi(\bar{x}_{t-1}, w_t)^T \nabla f(\bar{x}_t)$
  $v_t = \alpha_{t+1}\tilde{g}_{t+1} + \sum_{s=1}^t \alpha_s g_s$
  $w_{t+1} = \arg\min_{w \in \mathcal{W}} \langle v_t, w \rangle + \frac{1}{2\beta_t}\|w\|^2$
return $\bar{x}_T$

In Theorem 3, that the reason we cannot achieve acceleration is because the negative terms $-\|\nabla f(\bar{x}_{t-1}) - \nabla f(\bar{x}_t)\|_*^2$ do not come into play. This is because the positive term in the summation involves $\|\nabla f(\bar{x}_t)\|_*^2$, which is typically a larger quantity. To obtain acceleration, we need some form of optimism. In this section, we consider an alteration to Algorithm 2 that uses AO-FTRL for the meta-updates. Given some sequence of hints $\{\tilde{g}_t\}_{t=1}^T$, each $\tilde{g}_t \in \mathbb{R}^m$, each w_{t+1} is given by

$$w_{t+1} = \arg\min_{w \in \mathcal{W}} \left(\alpha_{t+1}\tilde{g}_{t+1} + \sum_{s=1}^{t} \alpha_s \langle \nabla h_s(w_s), w \rangle + \frac{1}{2\beta_t}\|w\|^2 \right). \tag{11}$$

For a complete description, see Algorithm 4. These updates do not correspond to the typical meta-update in Algorithm 1; however, we show momentarily that they can be interpreted as the targets in the BMG method, summarised in Algorithm 3. Before turning to BMG, we establish that optimistic meta-learning in the convex setting does indeed yield acceleration.

Theorem 4. *Let φ preserve regret and assume Algorithm 4 satisfy the assumptions in Section 2. Then*

$$f(\bar{x}_T) - f(x^*) \leq \frac{1}{\alpha_{1:T}} \left(\frac{\|w^*\|^2}{\beta_T} + \sum_{t=1}^{T} \frac{\alpha_t^2 \beta_t}{2} \|D\varphi(\bar{x}_{t-1}, w_t)^T \nabla f(\bar{x}_t) - \tilde{g}_t\|_*^2 - \frac{\alpha_t}{2L} \|\nabla f(\bar{x}_t) - \nabla f(x^*)\|_*^2 - \frac{\alpha_{1:t-1}}{2L} \|\nabla f(\bar{x}_{t-1}) - \nabla f(\bar{x}_t)\|_*^2 \right).$$

Proof. The proof follows the same lines as that of Theorem 3. The only difference is that the regret of the $\{w_t\}_{t=1}^T$ sequence can be upper bounded by $\frac{\|w^*\|^2}{\beta_T} + \frac{1}{2}\sum_{t=1}^T \alpha_t^2 \beta_t \|\nabla h_t(w_t) - \tilde{g}_t\|_*^2$ instead of $\frac{\|w^*\|^2}{\beta_T} + \frac{1}{2}\sum_{t=1}^T \alpha_t^2 \beta_t \|\nabla h_t(w_t)\|_*^2$, as per the AO-FTRL regret bound in Eq. 7. ■

From Theorem 4, it is clear that if $\tilde{g}_t$ is a good predictor of $D\varphi(\bar{x}_{t-1}, w_t)^T \nabla f(\bar{x}_t)$, then the positive term in the summation can be cancelled by the negative term. In a classical optimisation setting, $D\varphi = I_n$, and hence it is easy to see that simply choosing $\tilde{g}_t$ to be the previous gradient is sufficient to achieve the cancellation [13]. Indeed, this choice gives us Nesterov's Accelerated rate [27]. The upshot of this is that we can specialise Algorithm 4 to capture Nesterov's Accelerated method by choosing $\varphi : (x, w) \mapsto w$—as in the reduction to Heavy Ball—and setting the hints to $\tilde{g}_t = \nabla f(\bar{x}_{t-1})$. Hence, while the standard meta-update without optimism contains Heavy Ball as a special case, the optimistic meta-update contains Nesterov Acceleration as a special case.

In the meta-learning setting, $D\varphi$ is not an identity matrix, and hence the best targets for meta-learning are different. Naively, choosing $\tilde{g}_t = D\varphi(\bar{x}_{t-1}, w_t)^T \nabla f(\bar{x}_{t-1})$ would lead to a similar cancellation, but this is not allowed. At iteration t, we have not computed w_t when $\tilde{g}_t$ is chosen, and hence $D\varphi(\bar{x}_{t-1}, w_t)$ is not available. The nearest term that is accessible is $D\varphi(\bar{x}_{t-2}, w_{t-1})$.

Corollary 1. *Let each $\tilde{g}_{t+1} = D\varphi(\bar{x}_{t-1}, w_t)^T \nabla f(\bar{x}_t)$. Assume that φ satisfies*

$$\left\| D\varphi(x', w)^T \nabla f(x) - D\varphi(x'', w')^T \nabla f(x') \right\|_*^2 \leq \tilde{\lambda} \left\| \nabla f(x') - \nabla f(x) \right\|_*^2$$

for all $x'', x', x \in \mathcal{X}$ and $w, w' \in \mathcal{W}$, for some $\tilde{\lambda} > 0$. If each $\alpha_t = t$ and $\beta_t = \frac{t-1}{2t\tilde{\lambda}L}$, then $f(\bar{x}_T) - f(x^*) \leq \frac{4\tilde{\lambda}L \operatorname{diam}(\mathcal{W})}{T^2-1}$.

Proof: Appendix F.

F Proofs

This section provides complete proofs. We restate the results for convenience.

Lemma 1. *Given f, $\{\alpha_t\}_{t=1}^T$, and $\{x_t\}_{t=1}^T$, if φ preserves regret, then*

$$R^x(T) = \sum_{t=1}^T \alpha_t \langle \nabla f(\bar{x}_t), x_t - x^* \rangle \leq \sum_{t=1}^T \alpha_t \langle \nabla f(\bar{x}_t), \varphi(\bar{x}_{t-1}, w_t) - \varphi(\bar{x}_{t-1}, w^*) \rangle = R^w(T).$$

Proof. Starting from R^x in Eq. 9, if the update rule preserves regret, there exists $w^* \in \mathcal{W}$ for which

$$\begin{aligned}
R^x(T) =& \sum_{t=1}^T \alpha_t \langle \nabla f(\bar{x}_T), \varphi(\bar{x}_{t-1}, w_t) - x^* \rangle \\
=& \sum_{t=1}^T \alpha_t \langle \nabla f(\bar{x}_T), \varphi(\bar{x}_{t-1}, w_t) - \varphi(\bar{x}_{t-1}, w^*) \rangle + \sum_{t=1}^T \alpha_t \langle \nabla f(\bar{x}_T), \varphi(\bar{x}_{t-1}, w^*) - x^* \rangle \\
\leq& \sum_{t=1}^T \alpha_t \langle \nabla f(\bar{x}_T), \varphi(\bar{x}_{t-1}, w_t) - \varphi(\bar{x}_{t-1}, w^*) \rangle = R^w(T),
\end{aligned}$$

since w^* is such that $\sum_{t=1}^T \alpha_t \langle \nabla f(\bar{x}_T), \varphi(\bar{x}_{t-1}, w^*) - x^* \rangle \leq 0$. ∎

Theorem 3. *Let φ preserve regret and assume Algorithm 2 satisfy the assumptions in Section 2. Then*

$$\begin{aligned}
f(\bar{x}_T) - f(x^*) \leq& \frac{1}{\alpha_{1:T}} \left(\frac{\|w^*\|^2}{\beta} + \sum_{t=1}^T \frac{\lambda \beta \alpha_t^2}{2} \|\nabla f(\bar{x}_t)\|_*^2 \right. \\
& \left. - \frac{\alpha_t}{2L} \|\nabla f(\bar{x}_t) - \nabla f(x^*)\|_*^2 - \frac{\alpha_{1:t-1}}{2L} \|\nabla f(\bar{x}_{t-1}) - \nabla f(\bar{x}_t)\|_*^2 \right).
\end{aligned}$$

If x^* is a global minimiser of f, setting $\alpha_t = 1$ and $\beta = \frac{1}{\lambda L}$ yields $f(\bar{x}_T) - f(x^*) \leq \frac{\lambda L \operatorname{diam}(\mathcal{W})}{T}$.

Proof. Since φ preserves regret, by Lemma 1, the regret term $R^x(T)$ in Eq. 8 is upper bounded by $R^w(T)$. We therefore have

$$\begin{aligned}
& f(\bar{x}_T) - f(x^*) \leq \\
& \frac{1}{\alpha_{1:T}} \left(R^w(T) - \frac{\alpha_t}{2L} \|\nabla f(\bar{x}_t) - \nabla f(x^*)\|_*^2 - \frac{\alpha_{1:t-1}}{2L} \|\nabla f(\bar{x}_{t-1}) - \nabla f(\bar{x}_t)\|_*^2 \right).
\end{aligned} \tag{12}$$

Next, we need to upper-bound $R^w(T)$. Since, $R^w(T) = \sum_{t=1}^T \alpha_t \langle \nabla f(\bar{x}_T), \varphi(\bar{x}_{t-1}, w_t) - \varphi(\bar{x}_{t-1}, w^*) \rangle$, the regret of $\{w_t\}_{t=1}^T$ is defined under loss functions $h_t : \mathcal{W} \to \mathbb{R}$ given by $h_t = \alpha_t \langle \nabla f(\bar{x}_T), \varphi(\bar{x}_{t-1}, w) \rangle$. By assumption of convexity in φ, each h_t is convex in w. Hence, the regret under $\{\alpha_t h_t\}_{t=1}^T$ can be upper bounded by the regret under the linear losses $\{\alpha_t \langle \nabla h_t(w_t), \cdot \rangle\}_{t=1}^T$. These linear losses correspond to the losses used in the meta-update in Eq. 3. Since the meta-update is an instance of FTRL, we may upper-bound $R^w(T)$ by Eq. 7 with each $\tilde{g}_t = 0$. Putting this together along with smoothness of φ,

$$
\begin{aligned}
R^x(T) &\le R^w(T) \\
&= \sum_{t=1}^T \alpha_t \langle \nabla f(\bar{x}_T), \varphi(\bar{x}_{t-1}, w_t) - \varphi(\bar{x}_{t-1}, w^*) \rangle \\
&\le \sum_{t=1}^T \alpha_t \langle \nabla h_t(w_t), w_t - w^* \rangle \\
&\le \frac{\|w^*\|^2}{\beta} + \frac{\beta}{2} \sum_{t=1}^T \alpha_t^2 \|\nabla h_t(w_t)\|_*^2 \\
&= \frac{\|w^*\|^2}{\beta} + \frac{\beta}{2} \sum_{t=1}^T \alpha_t^2 \|D\varphi(\bar{x}_{t-1}, w_t)^T \nabla f(\bar{x}_t)\|_*^2 \\
&\le \frac{\|w^*\|^2}{\beta} + \frac{\lambda\beta}{2} \sum_{t=1}^T \alpha_t^2 \|\nabla f(\bar{x}_t)\|_*^2. \qquad (13)
\end{aligned}
$$

Putting Eq. 12 and Eq. 13 together gives the stated bound. Next, if x^* is the global optimiser, $\nabla f(x^*) = 0$ by first-order condition. Setting $\beta = 1/(L\lambda)$ and $\alpha_t = 1$ means the first two norm terms in the summation cancel. The final norm term in the summation is negative and can be ignored. We are left with $f(\bar{x}_T) - f(x^*) \le \frac{\lambda L \|w^*\|^2}{T} \le \frac{\lambda L \operatorname{diam}(\mathcal{W})}{T}$. ∎

Corollary 1. *Let each $\tilde{g}_{t+1} = D\varphi(\bar{x}_{t-1}, w_t)^T \nabla f(\bar{x}_t)$. Assume that φ satisfies*

$$\left\| D\varphi(x', w)^T \nabla f(x) - D\varphi(x'', w')^T \nabla f(x') \right\|_*^2 \le \tilde{\lambda} \left\| \nabla f(x') - \nabla f(x) \right\|_*^2$$

for all $x'', x', x \in \mathcal{X}$ and $w, w' \in \mathcal{W}$, for some $\tilde{\lambda} > 0$. If each $\alpha_t = t$ and $\beta_t = \frac{t-1}{2t\tilde{\lambda}L}$, then $f(\bar{x}_T) - f(x^*) \le \frac{4\tilde{\lambda}L \operatorname{diam}(\mathcal{W})}{T^2 - 1}$.

Proof. Plugging in the choice of $\tilde{g}_t$ and using that

$$\left\| D\varphi(\bar{x}_{t-1}, w_t)^T \nabla f(\bar{x}_t) - D\varphi(x_{t-2}, w_{t-1})^T \nabla f(\bar{x}_{t-1}) \right\|_*^2 \le \tilde{\lambda} \left\| \nabla f(\bar{x}_{t-1}) - \nabla f(\bar{x}_t) \right\|_*^2,$$

the bound in Theorem 4 becomes

$$f(\bar{x}_T) - f(x^*) \le \frac{1}{\alpha_{1:T}} \left(\frac{\|w^*\|^2}{\beta_T} + \frac{1}{2} \sum_{t=1}^T \left(\tilde{\lambda} \alpha_t^2 \beta_t - \frac{\alpha_{1:t-1}}{L} \right) \|\nabla f(\bar{x}_t) - \nabla f(\bar{x}_{t-1})\|_*^2 \right),$$

where we drop the negative terms $\|\nabla f(\bar{x}_t) - \nabla f(x^*)\|_*^2$. Setting $\alpha_t = t$ yields $\alpha_{1:t-1} = \frac{(t-1)t}{2}$, while setting $\beta_t = \frac{t-1}{2t\tilde{\lambda}L}$ means $\tilde{\lambda}\alpha_t^2\beta_t = \frac{(t-1)t}{2L}$. Hence, $\tilde{\lambda}\alpha_t^2\beta_t - \alpha_{1:t-1}/L$ cancels and we get

$$f(\bar{x}_T) - f(x^*) \le \frac{\|w^*\|^2}{\beta_T \alpha_{1:T}} = \frac{4\|w^*\|^2 \tilde{\lambda} L}{(T-1)(T+1)} \le \frac{4\tilde{\lambda} L \operatorname{diam}(\mathcal{W})}{(T-1)(T+1)} = \frac{4\tilde{\lambda} L \operatorname{diam}(\mathcal{W})}{T^2 - 1}.$$

∎

Corollary 3. *Let each $\tilde{g}_{t+1} = D\varphi(\bar{x}_{t-1}, w_t)^T \tilde{y}_{t+1}$, for some $\tilde{y}_{t+1} \in \mathbb{R}^n$. If each $\tilde{y}_{t+1}$ is a better predictor of the next gradient than $\nabla f(\bar{x}_{t-1})$, in the sense that*

$$\|D\varphi(\bar{x}_{t-2}, w_{t-1})^T \tilde{y}_t - D\varphi(\bar{x}_{t-1}, w_t)^T \nabla f(\bar{x}_t)\|_* \le \tilde{\lambda} \|\nabla f(\bar{x}_t) - \nabla f(\bar{x}_{t-1})\|_*,$$

then Algorithm 4 guarantees convergence at a rate $O(\tilde{\lambda}/T^2)$.

Proof. The proof follows the same argument as Corollary 1. ∎

Algorithm 5: BMG in practice (general version).

input : Weights $\{\rho_t\}_{t=1}^T, \{\beta_t\}_{t=1}^T$
input : Update rule φ
input : Matching function B^μ
input : Target oracle
input : Initialisation (x_0, w_1)
for $t = 1, 2, \ldots, T$:
  $x_t = x_{t-1} + \varphi(x_{t-1}, w_t)$
  Query z_t from target oracle
  $d_t : w \mapsto B^\mu_{z_t}(x_{t-1} + \varphi(x_{t-1}, w))$
  $w_{t+1} = w_t - \beta_t \nabla d_t(w_t)$
return x_T

G BMG as an instance of Optimism

In this section, we provide a more comprehensive reduction of BMG to AO-FTRL. First, we provide a more general definition of BMG. Let $\mu : \mathcal{X} \to \mathbb{R}$ be a convex distance generating function and define the Bregman Divergence $B^\mu : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ by

$$B^\mu_z(x) = \mu(x) - \mu(z) - \langle \nabla \mu(z), x - z \rangle.$$

Given initial condition (x_0, w_1), the BMG updates proceed according to

$$\begin{aligned} x_t &= x_{t-1} + \varphi(x_{t-1}, w_t) \\ w_{t+1} &= w_t - \beta_t \nabla d_t(w_t), \end{aligned} \tag{14}$$

where $d_t : \mathbb{R}^n \to \mathbb{R}$ is defined by $d_t(w) = B^\mu_{z_t}(x_{t-1} + \varphi(x_{t-1}, w_t))$, where each $z_t \in \mathbb{R}^n$ is referred to as a target. See Algorithm 5 for an algorithmic summary. A bootstrapped target uses the meta-learner's most recent update, x_t, to compute the target, $z_t = x_t + y_t$ for some tangent vector $y_t \in \mathbb{R}^n$. This tangent vector represents a form of optimism, and provides a signal to the meta-learner as to what would have been a more efficient update. In particular, the author's consider using the meta-learned update rule to construct y_t; $y_t = \varphi(x_t, w_t) - \nabla f(x_t \varphi(x_t, w - t))$. Note that $x_t = x_{t-1} + \varphi(x_{t-1}, w_t)$, and hence this tangent vector is obtained by applying the update rule again, but now to x_t. For this tangent to represent an improvement, it must be *assumed that w_t is a good parameterisation*. Hence, bootstrapping represents a form of optimism. To see how BMG relates to Algorithm 4, and in particular, Eq. 11, expand Eq. 14 to get

$$w_{t+1} = w_t - \beta_t D\varphi(x_{t-1}, w_t)^T \left(\nabla \mu(x_t) - \nabla \mu(z_t) \right). \tag{15}$$

In contrast, AO-FTRL reduces to a slightly different type of update.

Lemma 2. *Consider Algorithm 4. Given online losses $h_t : \mathcal{W} \to \mathbb{R}$ defined by $\{\langle D\varphi(\bar{x}_{t-1}, w_t)^T \nabla f(\bar{x}_t), \cdot \rangle\}_{t=1}^T$ and hint functions $\{\langle \tilde{g}_t, \cdot, \}\rangle_{t=1}^T$, with each $\tilde{g}_t \in \mathbb{R}^m$. If $\| \cdot \| = (1/2)\| \cdot \|_2$, an interior solution to Eq. 11 is given by*

$$w_{t+1} = \frac{\beta_t}{\beta_{t-1}} w_t - \beta_t \left(\alpha_{t+1} \tilde{g}_{t+1} + \alpha_t (D\varphi(\bar{x}_{t-1}, w_t)^T \nabla f(\bar{x}_t) - \tilde{g}_t) \right).$$

Proof. By direct computation:

$$\begin{aligned}
w_{t+1} &= \arg\min_{w\in\mathcal{W}} \left(\alpha_{t+1}\langle \tilde{g}_{t+1}, w\rangle + \sum_{s=1}^{t} \alpha_s \langle D\varphi(\bar{x}_{s-1}, w_s)^T \nabla f(\bar{x}_s), w\rangle + \frac{1}{2\beta_t}\|w\|_2^2\right) \\
&= -\beta_t \left(\alpha_{t+1}\tilde{g}_{t+1} + \sum_{s=1}^{t} \alpha_t D\varphi(\bar{x}_{s-1}, w_s)^T \nabla f(\bar{x}_s))\right) \\
&= -\beta_t \left(\alpha_{t+1}\tilde{g}_{t+1} + \alpha_t D\varphi(\bar{x}_{t-1}, w_t)^T \nabla f(\bar{x}_t) + \left(\sum_{s=1}^{t-1} \alpha_t D\varphi(\bar{x}_{s-1}, w_s)^T \nabla f(\bar{x}_s))\right)\right) \\
&= -\beta_t \left(\alpha_{t+1}\tilde{g}_{t+1} + \alpha_t (D\varphi(\bar{x}_{t-1}, w_t)^T \nabla f(\bar{x}_t) - \tilde{g}_t)\right) \\
&\quad - \beta_t \left(\alpha_t \tilde{g}_t + \sum_{s=1}^{t-1} \alpha_t D\varphi(\bar{x}_{s-1}, w_s)^T \nabla f(\bar{x}_s))\right) \\
&= \frac{\beta_t}{\beta_{t-1}} w_t - \beta_t \left(\alpha_{t+1}\tilde{g}_{t+1} + \alpha_t (D\varphi(\bar{x}_{t-1}, w_t)^T \nabla f(\bar{x}_t) - \tilde{g}_t)\right).
\end{aligned}$$

■

AO-FTRL includes a decay rate β_t/β_{t-1}; this decay rate can be removed by instead using optimistic online mirror descent [23, 12]—to simplify the exposition we consider only FTRL-based algorithms in this paper. An immediate implication of Lemma 2 is the error-corrected version of BMG.

Corollary 2. *Setting $\tilde{g}_{t+1} = D\varphi(\bar{x}_{t-1}, w_t)^T \tilde{g}_{t+1}$ for some $\tilde{y}_{t+1} \in \mathbb{R}^n$ yields an error-corrected version of the BMG meta-update in Eq. 14. Specifically, the meta-updates in Lemma 2 becomes*

$$w_{t+1} = \frac{\beta_t}{\beta_{t-1}} w_t - \underbrace{\beta_t D\varphi(\bar{x}_{t-1}, w_t)^T (\alpha_{t+1}\tilde{y}_{t+1} + \alpha_t \nabla f(\bar{x}_t))}_{\textit{BML update}} + \underbrace{\beta_t \alpha_t D\varphi(\bar{x}_{t-2}, w_{t-1})^T \tilde{y}_t}_{\textit{FTRL error correction}}.$$

Proof. Follows immediately by substituting for each $\tilde{g}_{t+1}$ in Lemma 2. ■

To illustrate this connection, Let $\mu = f$. In this case, the BMG update reads $w_{t+1} = w_t - \beta_t D\varphi(x_{t-1}, w_t)^T(\nabla f(z_t) - \nabla f(x_t))$. The equivalent update in the convex optimisation setting (i.e. Algorithm 4) is obtained by setting $\tilde{y}_{t+1} = \nabla f(z_t)$, in which case Corollary 2 yields

$$w_{t+1} = \frac{\beta_{t+1}}{\beta_t} w_t - \beta_t D\varphi(\bar{x}_{t-1}, w_t)^T (\alpha_{t+1}\nabla f(z_t) - \alpha_t \nabla f(\bar{x}_t)) + \xi_t,$$

where $\xi_t = \beta_t \alpha_t D\varphi(\bar{x}_{t-2}, w_{t-1})^T \nabla f(\bar{x}_t - 1)$ denotes the error correction term we pick up through AO-FTRL. Since Algorithm 5 does not average its iterates—while Algorithm 4 does—we see that these updates (ignoring ξ_t) are identical up to scalar coefficients (that can be controlled for by scaling each β_t and each $\tilde{g}_{t+1}$ accordingly).

More generally, the mapping from targets in BMG and hints in AO-FTRL takes on a more complicated pattern. Our next results show that we can always map one into the other. To show this, we need to assume a certain recursion. It is important to notice however that at each iteration introduces an unconstrained variable and hence the assumption on the recursion is without loss of generality (as the free variable can override it).

Theorem 5. *Targets in Algorithm 5 and hints in algorithm 4 commute in the following sense.* ***BMG → AO-FTRL.*** *Let BMG targets $\{z_t\}_{t=1}^T$ by given. A sequence of hints $\{\tilde{g}\}_{t=1}^T$ can be constructed recursively by*

$$\alpha_{t+1}\tilde{g}_{t+1} = D\varphi(\bar{x}_{t-1}, w_t)^T (\nabla\mu(\bar{x}_t) - \nabla\mu(z_t) - \alpha_t \nabla f(\bar{x}_t)) + \alpha_t \tilde{g}_t, \qquad t \in [T], \tag{16}$$

so that interior updates for Algorithm 4 are given by

$$w_{t+1} = \frac{\beta_t}{\beta_{t-1}} w_t - \beta_t (\nabla\mu(z_t) - \nabla\mu(\bar{x}_t)).$$

AO-FTRL → BMG. *Conversely, assume a sequence $\{\tilde{y}_t\}_{t=1}^T$ are given, each $\tilde{y}_t \in \mathbb{R}^n$. If μ strictly convex, a sequence of BMG targets $\{z_t\}_{t=1}^T$ can be constructed recursively by*

$$z_t = \nabla\mu^{-1}\left(\nabla\mu(x_t) - (\alpha_{t+1}\tilde{y}_{t+1} + \alpha_t \nabla f(x_t))\right) \qquad t \in [T],$$

so that BMG updates in Eq. 14 are given by

$$w_{t+1} = w_t - \beta_t \left(\alpha_{t+1}\tilde{g}_{t+1} + \alpha_t (D\varphi(\bar{x}_{t-1}, w_t)^T \nabla f(\bar{x}_t) - \tilde{g}_t)\right),$$

where each $\tilde{g}_{t+1}$ is the BMG-induced hint function, given by

$$\alpha_{t+1}\tilde{g}_{t+1} = \alpha_{t+1} D\varphi(x_{t-1}, w_t)^T \tilde{y}_{t+1} + \alpha_t \tilde{g}_t.$$

Proof. First, consider BMG → AO-FTRL. First note that $\tilde{g}_1$ is never used and can thus be chosen arbitrarily—here, we set $\tilde{g}_1 = 0$. For w_2, Lemma 2 therefore gives the interior update

$$w_2 = \frac{\beta_2}{\beta_1} w_1 - \beta_1 (\alpha_2 \tilde{g}_2 + \alpha_1 D\varphi(\bar{x}_0, w_1)^T \nabla f(\bar{x}_1)).$$

Since the formulate for $\tilde{g}_2$ in Eq. 16 only depends on quantities with iteration index $t = 0, 1$, we may set $\alpha_2 \tilde{g}_t = D\varphi(\bar{x}_0, w_1)^T (\nabla\mu(\bar{x}_1) - \nabla\mu(z_t) - \alpha_t \nabla f(\bar{x}_1))$. This gives the update

$$w_2 = \frac{\beta_2}{\beta_1} w_1 - \beta_1 D\varphi(\bar{x}_0, w_1)^T (\nabla\mu(\bar{x}_1) - \nabla\mu(z_1)).$$

Now assume the recursion holds up to time t. As before, we may choose $\alpha_{t+1}\tilde{g}_{t+1}$ according to the formula in Eq. 16 since all quantities on the right-hand side depend on quantities computed at iteration t or $t-1$. Subtituting this into Lemma 2, we have

$$\begin{aligned}
w_{t+1} &= \frac{\beta_t}{\beta_{t-1}} w_t - \beta_t \left(\alpha_{t+1}\tilde{g}_{t+1} + \alpha_t (D\varphi(\bar{x}_{t-1}, w_t)^T \nabla f(\bar{x}_t) - \tilde{g}_t)\right) \\
&= \frac{\beta_t}{\beta_{t-1}} w_t - \beta_t \left(D\varphi(\bar{x}_{t-1}, w_t)^T (\nabla\mu(\bar{x}_t) - \nabla\mu(z_t) - \alpha_t \nabla f(\bar{x}_t)) + \alpha_t \tilde{g}_t \right.\\
&\quad \left. + \alpha_t (D\varphi(\bar{x}_{t-1}, w_t)^T \nabla f(\bar{x}_t) - \tilde{g}_t)\right) \\
&= \frac{\beta_t}{\beta_{t-1}} w_t - \beta_t D\varphi(\bar{x}_{t-1}, w_t)^T (\nabla\mu(\bar{x}_t) - \nabla\mu(z_t)).
\end{aligned}$$

AO-FTRL → BMG. The proof in the other direction follows similarly. First, note that for μ strictly convex, $\nabla\mu$ is invertible. Then, $z_1 = \nabla\mu^{-1}(\nabla\mu(x_1) - (\alpha_2 \tilde{y}_2 + \alpha_1 \nabla f(x_1)))$. This target is permissible since x_1 is already computed and $\{\tilde{y}_t\}_{t=1}^T$ is given. Substituting this into the BMG meta-update in Eq. 14, we find

$$\begin{aligned}
w_2 &= w_1 - \beta_1 D\varphi(x_0, w_1)^T (\nabla\mu(x_1) - \nabla\mu(\nabla\mu^{-1}(\nabla\mu(x_1) - (\alpha_2 \tilde{y}_2 + \alpha_1 \nabla f(x_1))))) \\
&= w_1 - \beta_1 D\varphi(x_0, w_1)^T (\alpha_2 \tilde{y}_2 + \alpha_1 \nabla f(x_1)) \\
&= w_1 - \beta_1 \left(\alpha_2 \tilde{g}_2 + \alpha_1 (D\varphi(\bar{x}_0, w_1)^T \nabla f(\bar{x}_1) - \tilde{g}_1)\right),
\end{aligned}$$

where the last line uses that $\tilde{g}_2$ is defined by $\alpha_2 \tilde{g}_2 - \alpha_1 \tilde{g}_1 = D\varphi(\bar{x}_0, w_1)^T \tilde{y}_2$ and $\tilde{g}_1$ is arbitrary. Again, assume the recursion holds to time t. We then have

$$\begin{aligned}
w_{t+1} &= w_t - \beta_t D\varphi(x_{t-1}, w_t)^T \left(\nabla\mu(x_t) - \nabla\mu(z_t)\right) \\
&= w_t - \beta_t D\varphi(x_{t-1}, w_t)^T (\nabla\mu(x_t) \\
&\quad - \nabla\mu(\nabla\mu^{-1}(\nabla\mu(x_t) - (\alpha_{t+1}\tilde{y}_{t+1} + \alpha_t \nabla f(x_t))))) \\
&= w_t - \beta_t D\varphi(x_{t-1}, w_t)^T (\alpha_{t+1}\tilde{y}_{t+1} + \alpha_t \nabla f(x_t)) \\
&= w_t - \beta_t (\alpha_{t+1}\tilde{g}_{t+1} + \alpha_t (D\varphi(x_{t-1}, w_t)^T \nabla f(x_t) - \tilde{g}_t)).
\end{aligned}$$

■

More generally, Theorem 4 provides a sufficient condition for any target bootstrap in BMG to achieve acceleration. This is captured in the following corollary.

Corollary 3. *Let each $\tilde{g}_{t+1} = D\varphi(\bar{x}_{t-1}, w_t)^T \tilde{y}_{t+1}$, for some $\tilde{y}_{t+1} \in \mathbb{R}^n$. If each $\tilde{y}_{t+1}$ is a better predictor of the next gradient than $\nabla f(\bar{x}_{t-1})$, in the sense that*

$$\|D\varphi(\bar{x}_{t-2}, w_{t-1})^T \tilde{y}_t - D\varphi(\bar{x}_{t-1}, w_t)^T \nabla f(\bar{x}_t)\|_* \leq \tilde{\lambda} \|\nabla f(\bar{x}_t) - \nabla f(\bar{x}_{t-1})\|_*,$$

then Algorithm 4 guarantees convergence at a rate $O(\tilde{\lambda}/T^2)$.

Proof. The proof follows the same argument as Corollary 1. ■