OPT2019: 11th Annual Workshop on Optimization for Machine Learning

Near-Optimal Methods for Minimizing Star-Convex Functions and Beyond

Oliver Hinder OHINDER@GOOGLE.COM
Google, New York, NY 10011
Stanford University, Stanford, CA 94305

Aaron Sidford SIDFORD@STANFORD.EDU
Nimit Sohoni NIMS@STANFORD.EDU
Stanford University, Stanford, CA 94305

Abstract

In this paper, we provide near-optimal accelerated first-order methods for minimizing a broad class of smooth nonconvex functions that are strictly unimodal on all lines through a minimizer. This function class, which we call the class of smooth *quasar-convex* functions, is parameterized by a constant $\gamma \in (0, 1]$, where $\gamma = 1$ encompasses the classes of smooth convex and star-convex functions, and smaller values of γ indicate that the function can be "more nonconvex." We develop a variant of accelerated gradient descent that computes an ϵ-approximate minimizer of a smooth γ-quasar-convex function with at most $O(\gamma^{-1}\epsilon^{-1/2}\log(\gamma^{-1}\epsilon^{-1}))$ total function and gradient evaluations. We also derive a lower bound of $\Omega(\gamma^{-1}\epsilon^{-1/2})$ on the number of gradient evaluations required by any deterministic first-order method in the worst case, showing that, up to a logarithmic factor, no deterministic first-order algorithm can improve upon ours.

1. Introduction

Acceleration [38, 39] is one of the most powerful tools for improving the performance of first-order optimization methods. Nesterov's accelerated gradient descent method obtains asymptotically optimal runtimes for minimizing smooth convex functions [39]. Furthermore, acceleration is prevalent in stochastic optimization [2, 23, 28, 51, 52], is useful in coordinate descent methods [18, 25, 41, 46], can improve proximal methods [20, 33, 35], and yields tight rates for higher-order optimization [9, 21, 27]. Acceleration has shown to be successful in practical applications such as image deblurring [6] and neural network training [47].

More recently, acceleration techniques have been applied to compute ϵ-stationary points (i.e., points where the gradient has norm at most ϵ) of nonconvex functions with smooth derivatives [1, 10–12]. Worst-case runtime bounds in this setting are at least $\Omega(\epsilon^{-8/5})$, significantly worse than the corresponding $O(\epsilon^{-1/2})$ bound that accelerated gradient descent (AGD) achieves for smooth convex functions [39]. Still, in practice it is often possible to find approximate stationary points, and even approximate global minimizers, of nonconvex functions faster than these lower bounds suggest. This performance gap stems from the fairly weak assumptions underpinning these generic bounds. For example, Carmon et al. [10, 13] only assume Lipschitz continuity of the gradient and some higher-order derivatives. However, functions minimized in practice often admit significantly more structure, even if they are not convex. For example, under suitable assumptions on their inputs,

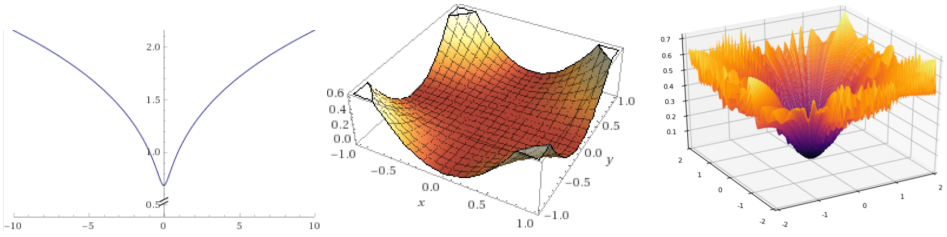

Figure 1: Examples of quasar-convex functions.

several popular nonconvex optimization problems, including matrix completion, deep learning, and phase retrieval, display "convexity-like" properties, e.g. that all local minimizers are global [5, 22]. Much more research is needed to characterize structured sets of functions for which minimizers can be efficiently found; our work is a step in this direction.

Basic notation We use $\|\cdot\|$ to denote the Euclidean norm (i.e. $\|\cdot\|_2$). We say that a function $f : \mathbb{R}^n \to \mathbb{R}$ is L-smooth, or L-Lipschitz differentiable, if $\|\nabla f(x) - \nabla f(y)\| \leq L \|x - y\|$ for all $x, y \in \mathbb{R}^n$. (We say a function is *smooth* if it is L-smooth for some $L \in [0, \infty)$.) We denote a minimizer of f by x^*, and we say that a point x is "ϵ-optimal" or an "ϵ-approximate minimizer" if $f(x) \leq f(x^*) + \epsilon$. We use $\log$ to denote the natural logarithm and $\log^+(\cdot)$ to denote $\max\{\log(\cdot), 1\}$.

In this paper, we improve upon the state-of-the-art complexity of first-order methods for minimizing smooth *quasar-convex* functions,[1] defined as follows.

Definition 1 *Let $\gamma \in (0, 1]$ and let x^* be a minimizer of the differentiable function $f : \mathbb{R}^n \to \mathbb{R}$. The function f is* γ-quasar-convex *with respect to x^* if for all $x \in \mathbb{R}^n$,*

$$f(x^*) \geq f(x) + \frac{1}{\gamma} \nabla f(x)^\top (x^* - x). \tag{1}$$

We simply say that f is quasar-convex if (1) holds for some minimizer x^* of f and constant $\gamma \in (0, 1]$. We refer to x^* as the "quasar-convex point" of f. Assuming differentiability, in the case $\gamma = 1$, condition (1) is equivalent to what is known as star-convexity [42]; if in addition the condition holds for all $y \in \mathbb{R}^n$ instead of just for x^*, it becomes the standard definition of convexity [7].

There are several other 'convexity-like' conditions in the literature related to quasar-convexity [3, 15, 17, 37, 45, 49, 53, 56].[2] For example, star-convexity is a condition that relaxes convexity, and is a strict subset of quasar-convexity in the differentiable case. Star-convexity is an interesting property because there is some evidence to suggest the loss function of neural networks might conform to this structure in large neighborhoods of the minimizers [31, 55]. Therefore, understanding acceleration for quasar-convex functions is pertinent to understanding acceleration for neural network training. Furthermore, Hardt, Ma, and Recht [26] show that, under mild assumptions, the objective

1. The concept of quasar-convexity was first introduced by Hardt et al. [26], who used the term 'weak quasi-convexity'. We decided to introduce the term quasar-convexity because we believe it is linguistically clearer; in particular, 'weak quasi-convexity' is a misnomer because it does not strictly subsume quasi-convexity.
2. A more in-depth discussion is presented in Appendix K.

for learning linear dynamical systems is quasar-convex; the problem of learning dynamical systems is closely related to the training of recurrent neural networks.

We are not the first to study acceleration on quasar-convex functions. Recent work by Guminov and Gasnikov [24] and Nesterov et al. [43] shows how to achieve accelerated rates for minimizing quasar-convex functions. For a function that is L-smooth and γ-quasar-convex with respect to a minimizer x^*, with initial distance to x^* bounded by R, the algorithm of Guminov and Gasnikov [24] yields an ϵ-optimal point in $O(\gamma^{-1}L^{1/2}R\epsilon^{-1/2})$ iterations, while the algorithm of Nesterov et al. [43] does so in $O(\gamma^{-3/2}L^{1/2}R\epsilon^{-1/2})$ iterations. For convex functions (which have $\gamma = 1$), these bounds match the *iteration* bounds achieved by AGD [39], but use a different oracle model. In particular, to achieve these iteration bounds, Guminov and Gasnikov [24] rely on a low-dimensional subspace optimization method within each iteration, while Nesterov et al. [43] use a one-dimensional line search over the function value in each iteration. However, quasar-convex functions are not necessarily unimodal along the arbitrary low-dimensional regions or line segments being searched over. Therefore, even finding an approximate global minimizer within these subregions may be computationally expensive, and thus the total number of *function and gradient evaluations* required by these methods may be large. Independently, recent work by Zhang et al. [54] uses a differential equation discretization to approach the accelerated $O(\kappa^{1/2}\log(\epsilon^{-1}))$ rate for minimization of smooth strongly quasar-convex functions in a neighborhood of the optimum, in the special case $\gamma = 1$.[3] Similarly, in the $\gamma = 1$ case, geometric descent [8] achieves $O(\kappa^{1/2}\log(\epsilon^{-1}))$ running times in terms of the number of calls to a one-dimensional line search oracle (although, as previously noted, the number of function and gradient evaluations required may still be large).

2. Theoretical results

For functions that are L-smooth and γ-quasar-convex, we provide Algorithm 1 which finds an ϵ-optimal solution in $O(\gamma^{-1}L^{1/2}R\epsilon^{-1/2})$ iterations (where, as before, R is an upper bound on the initial distance to the quasar-convex point x^*). Our iteration bound, given in Theorem 1, is the same as that of Guminov and Gasnikov [24], and a factor of $\gamma^{1/2}$ better than the $O(\gamma^{-3/2}L^{1/2}R\epsilon^{-1/2})$ bound of Nesterov et al. [43]. Additionally, we are the first to provide bounds on the total number of function and gradient evaluations required; our algorithm uses $O(\gamma^{-1}L^{1/2}R\epsilon^{-1/2}\log(\gamma^{-1}\epsilon^{-1}))$ function and gradient evaluations to find a ϵ-optimal solution.

Algorithm 1: AGD for quasar-convex functions
input : L-smooth $f : \mathbb{R}^n \to \mathbb{R}$, initial point $x^{(0)} \in \mathbb{R}^n$, number of iterations K

Define $\omega^{(-1)} = 1$, $\omega^{(k)} = \frac{\omega^{(k-1)}}{2}\left(\sqrt{(\omega^{(k-1)})^2 + 4} - \omega^{(k-1)}\right)$ for $k \geq 0$; set $v^{(0)} = x^{(0)}$

for $k = 0, 1, 2, \ldots, K-1$ **do**
  Set $\alpha^{(k)} = \texttt{BinaryLineSearch}(f, x^{(k)}, v^{(k)})$ `// Details omitted, see Appendix E`
  Set $y^{(k)} = \alpha^{(k)}x^{(k)} + (1-\alpha^{(k)})v^{(k)}$
  Set $x^{(k+1)} = y^{(k)} - \frac{1}{L}\nabla f(y^{(k)})$
  Set $v^{(k+1)} = v^{(k)} - \frac{\gamma}{L\omega^{(k)}}\nabla f(y^{(k)})$
end
return $x^{(K)}$

3. $\kappa = L/\mu$ denotes the *condition number* of an L-smooth (γ, μ)-strongly quasar-convex function.

Theorem 1 *If f is L-smooth and γ-quasar-convex with respect to a minimizer x^*, with $\gamma \in (0, 1]$ and $\left\|x^{(0)} - x^*\right\| \leq R$, then Algorithm 1 produces an ϵ-optimal point after $O\left(\gamma^{-1}L^{1/2}R\epsilon^{-1/2}\log^+\left(\gamma^{-1}L^{1/2}R\epsilon^{-1/2}\right)\right)$ function and gradient evaluations.*

In the appendix, we also analyze the *strongly quasar-convex* regime (a condition analogous to strong convexity) and provide an algorithm that produces an ϵ-optimal point using $O(\gamma^{-1}\kappa^{1/2}\log(\gamma^{-1}\kappa^{1/2})\log(\epsilon^{-1}))$ evaluations. In the $\gamma = 1$ (star-convex) case, our respective bounds are within a logarithmic factor of the runtimes of AGD on *convex* or *strongly convex* functions (AGD is asymptotically optimal in the convex and strongly convex settings [39]).

The key idea behind our algorithm is to take a close look at which essential invariants need to hold during the momentum step of AGD, and use this insight to redesign the algorithm to accelerate on general smooth quasar-convex functions. By observing how the function behaves along the line segment between current iterates $x^{(k)}$ and $v^{(k)}$, we show that for any smooth quasar-convex function, there always exists a point $y^{(k)}$ along this segment with the properties needed for acceleration. Further, we show that an efficient binary search can be used to find such a point.

To complement our upper bounds, we provide lower bounds of $\Omega(\gamma^{-1}L^{1/2}R\epsilon^{-1/2})$ for the number of gradient evaluations that *any* deterministic first-order method requires to find an ϵ-approximate minimizer of a quasar-convex function. This shows that up to logarithmic factors, our lower and upper bounds are tight. Our lower bounds extend the techniques of Carmon, Duchi, Hinder, and Sidford [10] to the class of smooth quasar-convex functions.

Theorem 2 *Let $\epsilon, R, L \in (0, \infty)$, $\gamma \in (0, 1]$, and assume $L^{1/2}R\epsilon^{-1/2} \geq 1$. Let $\mathcal{F}$ denote the set of L-smooth functions that are γ-quasar-convex with respect to some point with Euclidean norm $\leq R$. Then, given any deterministic first-order method, there exists a function $f \in \mathcal{F}$ such that the method requires at least $\Omega(\gamma^{-1}L^{1/2}R\epsilon^{-1/2})$ gradient evaluations to find an ϵ-optimal point of f.*

We derive our lower bounds by constructing a particular type of quasar-convex function known as a *zero-chain* and showing that any deterministic first-order method must use $\Omega(\gamma^{-1}L^{1/2}R\epsilon^{-1/2})$ gradient evaluations to optimize some function of this type. Full details are given in Appendices I-J.

3. Preliminary experiments

The main contribution of this work is theoretical. However, we also include preliminary experiments.

We first consider optimizing a "hard instance" - an example of the type of function used to construct the lower bound in Theorem 2. This function class is parameterized by σ and the dimension T; we denote these functions by $\bar{f}_{T,\sigma}$ (see Appendix I for the definition). We compare our method to other commonly used first-order methods: gradient descent (GD), [standard] accelerated gradient descent (AGD), nonlinear conjugate gradients (CG), and the limited-memory BFGS (L-BFGS) algorithm. (Out of all these algorithms, only our method and GD have theoretical guarantees for quasar-convex function minimization.) We next evaluate our algorithm on real-world tasks: we use our algorithm to train a support vector machine (SVM) on the nine LIBSVM UCI binary classification datasets [14] (which are derived from the UCI "Adult" datasets [16]). The SVM loss function we use is a

smoothed version of the hinge loss: $f(x) = \sum_{i=1}^n \phi_\alpha(1 - b_i a_i^\top x)$, where $a_i \in \mathbb{R}^d, b_i = \pm 1$ are given by the training data (the a_i's are the covariates and the b_i's are the labels), and $\phi_\alpha(t) = 0$ for $t \le 0$, $\frac{t^2}{2}$ for $t \in [0, 1]$, and $\frac{t^\alpha - 1}{\alpha} + \frac{1}{2}$ for $t \ge 1$. When $\alpha = 1$, $\phi_\alpha = \frac{t^2}{2}$ for all $t \ge 0$, and thus ϕ_α and f are convex. For all $\alpha \in (0, 1]$, ϕ_α is smooth and α-quasar-convex. Line searches for this function are inexpensive, as the quantities $b_i a_i^\top x$ need only be calculated once per outer loop iteration. Results are given in Table 1.

Finally, we evaluate on the problem of learning linear dynamical systems, which was shown to be quasar-convex (under certain assumptions) by Hardt et al. [26]. In this problem, we are given observations $\{(x_t, y_t)\}_{i=1}^T$ generated by the time-invariant linear system $h_{t+1} = Ah_t + Bx_t; y_t = Ch_t + Dx_t$, where $x_t, y_t \in \mathbb{R}$; $h_t \in \mathbb{R}^n$ is the *hidden state* at time t; and $\Theta = (A, B, C, D)$ are the (unknown) parameters of the system. Informally, we seek to learn $\hat{\Theta}$ to minimize $\frac{1}{T}\sum_{i=1}^T (y_t - \hat{y}_t)^2$, where $\hat{h}_{t+1} = \hat{A}\hat{h}_t + \hat{B}x_t; \hat{y}_t = \hat{C}\hat{h}_t + \hat{D}x_t$, and $\hat{h}_0 = 0$. When parameterized in *controllable canonical form*, this problem was shown to be quasar-convex on a subset of the domain near the optimum in [26]. We describe this problem and our experimental approach in more detail in Appendix D. Representative plots are given in Figure 2. Despite the nonconvexity, AGD performs quite well on this problem. Nonetheless, we observe that our method is competitive with AGD in terms of *iteration* count; we use more *function evaluations* due to the line search, but gradient evaluations are about twice as expensive in this setting, and the line search can also be parallelized. The design of better heuristics to speed up our method (for example, using the standard AGD value of α as an initial guess for the line search) is an interesting question for future investigation.

In all experiments, we use adaptive step sizes for our method, as well as GD and AGD, as in practice L may not be known *a priori*. We note that our algorithm can be straightforwardly generalized to this adaptive step-size setting, for which Theorem 1 can be rederived.

↓ Function / Algorithm →	Ours (Alg. 4)	Gradient Descent (GD)	Standard AGD	Nonlinear CG	L-BFGS
$\bar{f}_{T,\sigma}$ ($\sigma = 10^{-1}, T = 10^2; \epsilon = 10^{-4}$)	422; 1,451	336; 738	272; 869	312; 1,599	354; 1,778
$\bar{f}_{T,\sigma}$: ($\sigma = 10^{-4}, T = 10^3; \epsilon = 10^{-6}$)	12,057; 55,357	18,607; 40,684	3,891; 12,399	1,251; 3,647	1,093; 6,554
$\bar{f}_{T,\sigma}$: ($\sigma = 10^{-6}, T = 10^3; \epsilon = 10^{-8}$)	17,135; 167,447	275,572; 602,561	55,623; 177,247	10,007; 30,023	2,079; 12,476
LIBSVM UCI datasets ($\alpha = 1; \epsilon = 10^{-4}$)	0.92; +0.017%	4.65; +0.036%	—	0.46; +0.001%	0.29; +0.010%
LIBSVM UCI datasets ($\alpha = 0.5; \epsilon = 10^{-4}$)	1.32; +0.016%	4.78; +0.033%	—	0.48; +0.001%	0.30; +0.011%

Table 1: Experimental results. The stopping criterion used is $\|\nabla f(x)\|_\infty \le \epsilon$. For $\bar{f}_{T,\sigma}$ we report *(# iterations; # function+gradient evals)*; the initial point is $x_0 = \mathbf{0}$. For LIBSVM UCI datasets, we report: the *ratio* of the total number of iterations required compared to standard AGD, averaged over all 9 datasets and 3 different random initializations (shared across algorithms) per dataset, and the average final *test classification accuracy difference* compared to AGD.

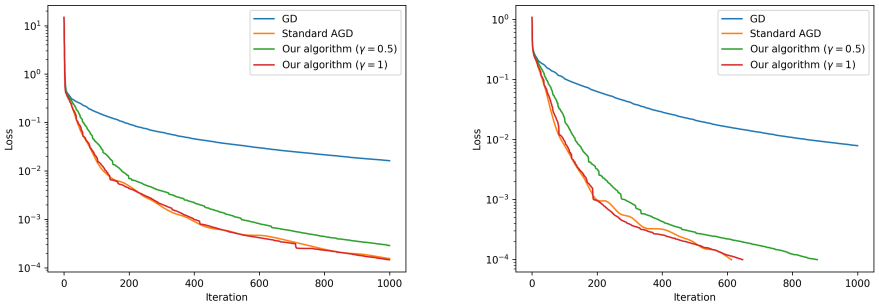

Figure 2: Results on learning linear dynamical systems, for two different problem instances. We evaluate our method with $\gamma = \{0.5, 1\}$, and compare to GD and AGD. We run until the loss is $< 10^{-4}$ or 1000 iterations have been reached. Our method uses ≈4x as many total evaluations as AGD; for instance, in the first setting all methods run for 1000 iterations and use 2195, 3195, 13562 and 14626 total evaluations respectively (out of which 1000 are gradient evaluations).

References

[1] Naman Agarwal, Zeyuan Allen-Zhu, Brian Bullins, Elad Hazan, and Tengyu Ma. Finding approximate local minima faster than gradient descent. In *Symposium on Theory of Computing (STOC)*, pages 1195–1199. ACM, 2017.

[2] Zeyuan Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. *Journal of Machine Learning Research*, 18(1):8194–8244, 2017.

[3] Mihai Anitescu. Degenerate nonlinear programming with a quadratic growth condition. *SIAM Journal on Optimization*, 10(4):1116–1135, 2000.

[4] Kenneth J. Arrow and Alain C. Enthoven. Quasi-concave programming. *Econometrica*, 16(5):779–800, 1961.

[5] Peter L. Bartlett, David P. Helmbold, and Philip M. Long. Gradient descent with identity initialization efficiently learns positive-definite linear transformations by deep residual networks. *Neural Computation*, 31(3):477–502, 2019.

[6] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[7] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[8] Sébastien Bubeck, Yin Tat Lee, and Mohit Singh. A geometric alternative to Nesterov's accelerated gradient descent. *arXiv preprint arXiv:1506.08187*, 2015.

[9] Sébastien Bubeck, Qijia Jiang, Yin Tat Lee, Yuanzhi Li, and Aaron Sidford. Near-optimal method for highly smooth convex optimization. In *Conference on Learning Theory (COLT)*, pages 492–507, 2019.

[10] Yair Carmon, John C. Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points II: First-order methods. *arXiv preprint arXiv:1711.00841*, 2017.

[11] Yair Carmon, John C. Duchi, Oliver Hinder, and Aaron Sidford. Convex until proven guilty: Dimension-free acceleration of gradient descent on non-convex functions. In *International Conference on Machine Learning (ICML)*, pages 654–663, 2017.

[12] Yair Carmon, John C. Duchi, Oliver Hinder, and Aaron Sidford. Accelerated methods for nonconvex optimization. *SIAM Journal on Optimization*, 28(2):1751–1772, 2018.

[13] Yair Carmon, John C. Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points I. *Mathematical Programming*, pages 1–50, 2019.

[14] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[15] Bruce D. Craven and Barney M. Glover. Invex functions and duality. *Journal of the Australian Mathematical Society*, 39(1):1–20, 1985.

[16] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL http://archive.ics.uci.edu/ml.

[17] Marian J. Fabian, René Henrion, Alexander Y. Kruger, and Jiří V. Outrata. Error bounds: Necessary and sufficient conditions. *Set-Valued and Variational Analysis*, 18(2):121–149, 2010.

[18] Olivier Fercoq and Peter Richtárik. Accelerated, parallel, and proximal coordinate descent. *SIAM Journal on Optimization*, 25(4):1997–2023, 2015.

[19] Roger Fletcher and Colin M. Reeves. Function minimization by conjugate gradients. *The Computer Journal*, 7(2):149–154, 1964.

[20] Roy Frostig, Rong Ge, Sham Kakade, and Aaron Sidford. Un-regularizing: approximate proximal point and faster stochastic algorithms for empirical risk minimization. In *International Conference on Machine Learning (ICML)*, pages 2540–2548, 2015.

[21] Alexander Gasnikov, Pavel Dvurechensky, Eduard Gorbunov, Evgeniya Vorontsova, Daniil Selikhanovych, and César A. Uribe. The global rate of convergence for optimal tensor methods in smooth convex optimization. *Computer Research and Modeling*, 10(6):737–753, 2018.

[22] Rong Ge, Jason D. Lee, and Tengyu Ma. Matrix completion has no spurious local minimum. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2973–2981, 2016.

[23] Saeed Ghadimi and Guanghui Lan. Accelerated gradient methods for nonconvex nonlinear and stochastic programming. *Mathematical Programming*, 156(1-2):59–99, 2016.

[24] Sergey Guminov and Alexander Gasnikov. Accelerated methods for α-weakly-quasi-convex problems. *arXiv preprint arXiv:1710.00797*, 2017.

[25] Filip Hanzely and Peter Richtárik. Accelerated coordinate descent with arbitrary sampling and best rates for minibatches. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 304–312, 2019.

[26] Moritz Hardt, Tengyu Ma, and Benjamin Recht. Gradient descent learns linear dynamical systems. *Journal of Machine Learning Research*, 19(29):1–44, 2018.

[27] Bo Jiang, Haoyue Wang, and Shuzhong Zhang. An optimal high-order tensor method for convex optimization. In *Conference on Learning Theory (COLT)*, pages 1799–1801, 2019.

[28] Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems (NIPS)*, pages 315–323, 2013.

[29] Pooria Joulani, András György, and Csaba Szepesvári. A modular analysis of adaptive (non-)convex optimization: Optimism, composite objectives, and variational bounds. In *International Conference on Algorithmic Learning Theory (ALT)*, 2017.

[30] Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the Polyak-Łojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD)*, pages 795–811. Springer, 2016.

[31] Robert Kleinberg, Yuanzhi Li, and Yang Yuan. An alternative view: When does SGD escape local minima? In *International Conference on Machine Learning (ICML)*, pages 2698–2707, 2018.

[32] Jasper C.H. Lee and Paul Valiant. Optimizing star-convex functions. In *Symposium on Foundations of Computer Science (FOCS)*, pages 603–614. IEEE, 2016.

[33] Huan Li and Zhouchen Lin. Accelerated proximal gradient methods for nonconvex programming. In *Advances in Neural Information Processing Systems (NIPS)*, pages 379–387, 2015.

[34] Yuanzhi Li and Yang Yuan. Convergence analysis of two-layer neural networks with ReLU activation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 597–607, 2017.

[35] Hongzhou Lin, Julien Mairal, and Zaid Harchaoui. A universal catalyst for first-order optimization. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3384–3392, 2015.

[36] Olvi L. Mangasarian. Pseudo-convex functions. *Journal of the Society for Industrial and Applied Mathematics Series A Control*, 3(2):281–290, 1965.

[37] Ion Necoara, Yurii Nesterov, and François Glineur. Linear convergence of first order methods for non-strongly convex optimization. *Mathematical Programming*, 175(1):69–107, 2019.

[38] Arkadi S. Nemirovski. Orth-method for smooth convex optimization. *Izvestia AN SSSR, Ser. Tekhnicheskaya Kibernetika*, 2, 1982.

[39] Yurii Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

[40] Yurii Nesterov. *Introductory Lectures on Convex Optimization*. Kluwer Academic Publishers, 2004.

[41] Yurii Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

[42] Yurii Nesterov and Boris T. Polyak. Cubic regularization of Newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

[43] Yurii Nesterov, Alexander Gasnikov, Sergey Guminov, and Pavel Dvurechensky. Primal-dual accelerated gradient descent with line search for convex and nonconvex optimization problems. *Proceedings of the Russian Academy of Sciences (RAS)*, 485(1):15–18, 2019.

[44] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *Advances in Neural Information Processing Systems (NeurIPS) - Autodiff Workshop*, 2017.

[45] Boris T. Polyak. Gradient methods for minimizing functionals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, 3(4):643–653, 1963.

[46] Shai Shalev-Shwartz and Tong Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. In *International Conference on Machine Learning (ICML)*, pages 64–72, 2014.

[47] Ilya Sutskever, James Martens, George E. Dahl, and Geoffrey E. Hinton. On the importance of initialization and momentum in deep learning. In *International Conference on Machine Learning (ICML)*, pages 1139–1147, 2013.

[48] Alexander Tyurin. Mirror version of similar triangles method for constrained optimization problems. *arXiv preprint arXiv:1705.09809*, 2017.

[49] Huynh Van Ngai and Jean-Paul Penot. Approximately convex functions and approximately monotonic operators. *Nonlinear Analysis: Theory, Methods & Applications*, 66(3):547–564, 2007.

[50] Jean-Philippe Vial. Strong and weak convexity of sets and functions. *Mathematics of Operations Research*, 8(2):231–259, 1983.

[51] Blake E. Woodworth and Nati Srebro. Tight complexity bounds for optimizing composite objectives. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3639–3647, 2016.

[52] Peng Xu, Bryan He, Christopher De Sa, Ioannis Mitliagkas, and Christopher Ré. Accelerated stochastic power iteration. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 58–67, 2018.

[53] Hui Zhang and Wotao Yin. Gradient methods for convex minimization: better rates under weaker conditions. *arXiv preprint arXiv:1303.4645*, 2013.

[54] Jingzhao Zhang, Suvrit Sra, and Ali Jadbabaie. Acceleration in first order quasi-strongly convex optimization by ODE discretization. *arXiv preprint arXiv:1905.12436*, 2019.

[55] Yi Zhou, Junjie Yang, Huishuai Zhang, Yingbin Liang, and Vahid Tarokh. SGD converges to global minimum in deep learning via star-convex path. In *International Conference on Learning Representations (ICLR)*, 2019.

[56] Zhengyuan Zhou, Panayotis Mertikopoulos, Nicholas Bambos, Stephen Boyd, and Peter Glynn. Stochastic mirror descent in variationally coherent optimization problems. In *Advances in Neural Information Processing Systems (NIPS)*, pages 7040–7049, 2017.

Appendix

D. Additional Experimental Details

We implement our algorithm, as well as AGD and GD, in Julia and Python. We run our experiments on learning linear dynamical systems (LDS) using the PyTorch framework [44]. We generate the true parameters and the dynamical model inputs the same way as in [26], using the same parameters $n = 20, T = 500$. However, differently from this paper, we do not generate fresh sequences $\{(x_t, y_t)\}$ at each iteration, but instead generate 100 sequences at the beginning which are used throughout (so, it is no longer a stochastic optimization problem). As in [26], we actually minimize the loss $\frac{1}{|\mathcal{B}|}\sum_{(x,y)\in\mathcal{B}}\left(\frac{1}{T-T_1}\sum_{i>T_1}(y_t-\hat{y}_t)^2\right)$, where the outer summation is over the batch $\mathcal{B}$ of 100 sequences and the inner summation starts at time $T_1 := T/4$, to mitigate the fact that the initial hidden state is not known. In addition, we generate the initial point $(\hat{A}_0, \hat{C}_0, \hat{D}_0)$ by perturbing the true dynamical system parameters (A, C, D) with random noise; we additionally ensure that the spectral radius of $\hat{A}_0$ remains less than 1.

The quasar-convexity parameter γ derived in [26] for the LDS objective is defined as the supremum of the real part of a ratio of two degree-n univariate polynomials over the complex unit circle. Therefore, it is difficult to calculate in practice. We instead simply select different values of γ to use in our experiments; we find that, while the choice of γ does affect performance somewhat, our method does not break down even if the "wrong" choice is used.

Hardt, Ma, and Recht [26] presented two better-performing alternatives to fixed-stepsize SGD: SGD with gradient clipping or projected SGD. By contrast, as we use an adaptive step size, there is no need to clip gradients; in addition, we find projection to be unnecessary as the initial iterate we generate already has $\rho(\hat{A}_0) < 1$ by construction.

In the LDS experiments, we use forward difference to approximate the 1D gradients in the line search (as described in Appendix E), since gradient evaluations are more expensive than function evaluations in this case; fortunately, this does not incur significant numerical error.

In all experiments, we do not use any initial guess for the line search to compute $\alpha^{(k)}$ (Algorithm 3). For the adaptive step sizes, we use a standard scheme in which the step size at iteration $k > 0$ [which we denote $\frac{1}{L^{(k)}}$] is initialized to the previous step size $\frac{1}{L^{(k-1)}}$ times a fixed value $\zeta_1 > 1$, and then multiplied by a fixed value $\zeta_2 \in (0, 1)$ until it is small enough so that the function value decrease is sufficient,[4] where ζ_1, ζ_2 are constant hyperparameters. In all experiments for GD, AGD, and our method, we used $\zeta_1 = 1.1, \zeta_2 = 0.6$, and $L^{(0)} = 1$ (these values were not tuned; the algorithms are fairly insensitive to them when reasonable settings are used).

E. Quasar-Convex Minimization Framework

In this section, we provide and analyze a general algorithmic template for accelerated minimization of smooth quasar-convex functions. In Section F.1 we show how to leverage this framework to achieve accelerated rates for minimizing *strongly* quasar-convex functions, and in Section F.2 we

4. Specifically, for GD, we decrease the step size $\frac{1}{L^{(k)}}$ until the criterion $f(x^{(k+1)}) \le f(x^{(k)}) - \frac{1}{2L^{(k)}}||\nabla f(x^{(k)})||^2$ is satisfied; for AGD and our method, the criterion is $f(x^{(k+1)}) \le f(y^{(k)}) - \frac{1}{2L^{(k)}}||\nabla f(y^{(k)})||^2$. These criteria are guaranteed to hold when $L^{(k)} \ge L$.

show how to achieve accelerated rates for minimizing *non-strongly* quasar-convex functions (i.e. when $\mu = 0$), as in Algorithm 1. For simplicity, we assume the domain is $\mathbb{R}^n$.

Our algorithm (Algorithm 2) is a simple generalization of accelerated gradient descent. Given a differentiable function $f \in \mathbb{R}^n \to \mathbb{R}$ with smoothness parameter $L > 0$ and initial point $x^{(0)} = v^{(0)} \in \mathbb{R}^n$, the algorithm iteratively computes points $x^{(k)}, v^{(k)} \in \mathbb{R}^n$ of improving "quality." However, it is challenging to argue that Algorithm 2 actually performs optimally *without the assumption of convexity*. The crux of circumventing convexity is to show that there exists a way to efficiently compute the momentum parameter $\alpha^{(k)}$ to yield convergence at the desired rate. In this section, we provide general tools for analyzing this algorithm; in Section F, we leverage this analysis with specific choices of the parameters $\alpha^{(k)}, \beta$, and $\eta^{(k)}$ to derive our fully-specified accelerated schemes for both quasar-convex and strongly quasar-convex functions.

Algorithm 2: General AGD Framework
input: L-smooth $f : \mathbb{R}^n \to \mathbb{R}$, initial point $x^{(0)} \in \mathbb{R}^n$, number of iterations K

Parameter $\beta \in [0, 1]$ and sequences $\{\alpha^{(k)}\}_{k=0}^{K-1}$, $\{\eta^{(k)}\}_{k=0}^{K-1}$ are computed as defined by the particular algorithm instance, where $\alpha^{(k)} \in [0, 1]$, $\eta^{(k)} \geq \frac{\gamma}{L}$. Set $v^{(0)} = x^{(0)}$.

for $k = 0, 1, 2, \ldots, K-1$ **do**
  Set $y^{(k)} = \alpha^{(k)} x^{(k)} + (1 - \alpha^{(k)}) v^{(k)}$
  Set $x^{(k+1)} = y^{(k)} - \frac{1}{L} \nabla f(y^{(k)})$
  Set $v^{(k+1)} = \beta v^{(k)} + (1 - \beta) y^{(k)} - \eta^{(k)} \nabla f(y^{(k)})$
end
return $x^{(K)}$

We first define notation that will be used throughout Sections E and F:

Definition 2 *Let* $\epsilon^{(k)} \triangleq f(x^{(k)}) - f(x^*), \epsilon_y^{(k)} \triangleq f(y^{(k)}) - f(x^*), r^{(k)} \triangleq \left\|v^{(k)} - x^*\right\|^2$, $r_y^{(k)} \triangleq \left\|y^{(k)} - x^*\right\|^2, Q^{(k)} \triangleq \beta \left(2\eta^{(k)} \alpha^{(k)} \nabla f(y^{(k)})^\top (x^{(k)} - v^{(k)}) - (\alpha^{(k)})^2 (1 - \beta) \left\|x^{(k)} - v^{(k)}\right\|^2\right)$.

In the remainder of this section, we analyze Algorithm 2. We assume that f is L-smooth and (γ, μ) strongly quasar-convex (possibly with $\mu = 0$) with respect to a minimizer x^*. First, we use Lemma 1 to bound how much the function error of $x^{(k)}$ and the distance from $v^{(k)}$ to x^* decrease at each iteration. To prove this lemma we use the following elementary fact (see [40] for proof).

Fact 1 *If f is L-smooth and $x = y - \frac{1}{L} \nabla f(y)$, then $f(x) \leq f(y) - \frac{1}{2L} \|\nabla f(y)\|^2$ for all y. Additionally, if x^* is a minimizer of f, then $f(y) \leq f(x^*) + \frac{L}{2} \|y - x^*\|^2$ for all y.*

Lemma 1 (One Step Framework Analysis) *Suppose f is L-smooth and (γ, μ)-quasar-convex with respect to a minimizer x^*. Then, in each iteration $k \geq 0$ of Algorithm 2 applied to f, it is the case that*

$$2(\eta^{(k)})^2 L \epsilon^{(k+1)} + r^{(k+1)} \leq \beta r^{(k)} + \left[(1 - \beta) - \gamma \mu \eta^{(k)}\right] r_y^{(k)} + 2\eta^{(k)} \left[L\eta^{(k)} - \gamma\right] \epsilon_y^{(k)} + Q^{(k)}.$$

Proof Let $z^{(k)} \triangleq \beta v^{(k)} + (1-\beta)y^{(k)}$. Since $v^{(k+1)} = z^{(k)} - \eta^{(k)}\nabla f(y^{(k)})$, direct algebraic manipulation yields that

$$\begin{aligned} r^{(k+1)} &= \left\|v^{(k+1)} - x^*\right\|^2 = \left\|z^{(k)} - x^* - \eta^{(k)}\nabla f(y^{(k)})\right\|^2 \\ &= \left\|z^{(k)} - x^*\right\|^2 + 2\eta^{(k)}\nabla f(y^{(k)})^\top(x^* - z^{(k)}) + (\eta^{(k)})^2\left\|\nabla f(y^{(k)})\right\|^2 . \end{aligned} \tag{2}$$

Using the definitions of $z^{(k)}$ and $y^{(k)}$, we have

$$\begin{aligned} \left\|z^{(k)} - x^*\right\|^2 &= \beta\left\|v^{(k)} - x^*\right\|^2 + (1-\beta)\left\|y^{(k)} - x^*\right\|^2 - \beta(1-\beta)\left\|v^{(k)} - y^{(k)}\right\|^2 \\ &= \beta r^{(k)} + (1-\beta)r_y^{(k)} - \beta(1-\beta)(\alpha^{(k)})^2\left\|v^{(k)} - x^{(k)}\right\|^2 . \end{aligned} \tag{3}$$

Further, since $v^{(k)} = y^{(k)} + \alpha^{(k)}(v^{(k)} - x^{(k)})$ and $z^{(k)} = \beta v^{(k)} + (1-\beta)y^{(k)} = y^{(k)} + \alpha^{(k)}\beta(v^{(k)} - x^{(k)})$, it follows that

$$\nabla f(y^{(k)})^\top(x^* - z^{(k)}) = \nabla f(y^{(k)})^\top(x^* - y^{(k)}) + \alpha^{(k)}\beta\nabla f(y^{(k)})^\top(x^{(k)} - v^{(k)}) . \tag{4}$$

Since (γ,μ)-strong quasar-convexity of f implies $-\epsilon_y^{(k)} \geq \frac{1}{\gamma}\nabla f(y^{(k)})^\top(x^* - y^{(k)}) + \frac{\mu}{2}r_y^{(k)}$ and the definition of $x^{(k+1)}$ implies $\left\|\nabla f(y^{(k)})\right\|^2 \leq 2L[\epsilon_y^{(k)} - \epsilon^{(k+1)}]$ by Fact 1, combining with (2), (3), and (4) yields the result. ■

Lemma 1 provides our main bound on how the error $\epsilon^{(k)}$ changes between successive iterations of Algorithm 2. The key step necessary to apply this lemma is to relate $f(y^{(k)})$ and $\nabla f(y^{(k)})^\top(x^{(k)} - v^{(k)})$ to $f(x^{(k)})$, in order to bound $Q^{(k)}$. In the standard analysis of accelerated gradient descent, convexity is used to obtain such a connection. In our algorithms, we instead perform binary search to compute the momentum parameter $\alpha^{(k)}$ for which the necessary relationship holds without assuming convexity. The following lemma shows that there always exists a setting of $\alpha^{(k)}$ that satisfies the necessary relationship.

Lemma 2 (Existence of "Good" α) *Let $f : \mathbb{R}^n \to \mathbb{R}$ be differentiable and let $x, v \in \mathbb{R}^n$. For $\alpha \in \mathbb{R}$ define $y_\alpha \triangleq \alpha x + (1-\alpha)v$. For any $c \geq 0$ there exists $\alpha \in [0,1]$ such that*

$$\alpha\nabla f(y_\alpha)^\top(x - v) \leq c\left[f(x) - f(y_\alpha)\right] . \tag{5}$$

Proof Define $g(\alpha) \triangleq f(y_\alpha)$. Then for all $\alpha \in \mathbb{R}$ we have $g'(\alpha) = \nabla f(y_\alpha)^\top(x - v)$. Consequently, (5) is equivalent to the condition $\alpha g'(\alpha) \leq c[g(1) - g(\alpha)]$.

If $g'(1) \leq 0$, inequality (5) trivially holds at $\alpha = 1$; if $f(v) = g(0) \leq g(1) = f(x)$, the inequality trivially holds at $\alpha = 0$. If neither of these conditions hold, $g'(1) > 0$ and $g(0) > g(1)$, so Fact 2 from Section G.1 implies that there is a value of $\alpha \in (0,1)$ such that $g'(\alpha) = 0$ and $g(\alpha) \leq g(1)$, and therefore this value of α satisfies (5). Figure E illustrates this third case graphically. ■

In our algorithms we will not seek $\alpha \in [0,1]$ satisfying (5) exactly, but instead $\alpha \in [0,1]$ such that

$$\alpha\nabla f(y_\alpha)^\top(x - v) - \alpha^2 b\left\|x - v\right\|^2 \leq c\left[f(x) - f(y_\alpha)\right] + \tilde{\epsilon} , \tag{6}$$

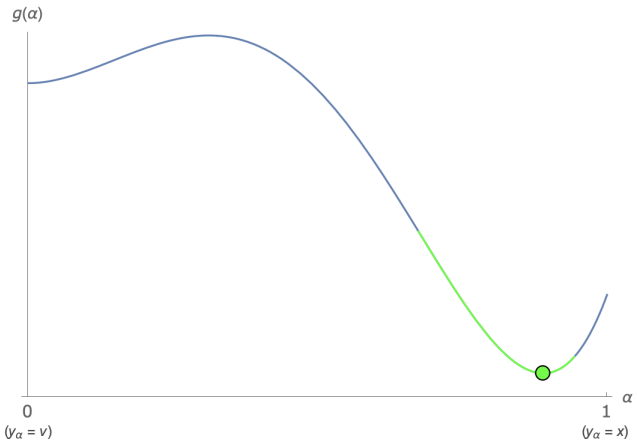

Figure 3: Illustration of Lemma 2. $g(\alpha)$ is defined as in the proof of the lemma; here, we depict the case where $g(0) > g(1)$ and $g'(1) > 0$. The points highlighted in green satisfy inequality (5); the circled point has $g'(\alpha) = 0$ and $g(\alpha) \le g(1)$. Here $c = 10$.

for some $b, c, \tilde{\epsilon} \ge 0$. As (6) is a weaker statement than (5), the existence of α satisfying (6) directly follows from Lemma 2. Moreover, we will show how to lower bound the size of the set of points satisfying (6), which we use to bound the time required to compute such a point.

We can thus bound the quantity $Q^{(k)}$ from Lemma 1 by selecting $\alpha^{(k)}$ to satisfy (6) with appropriate settings of $b, c, \tilde{\epsilon}$, which we do in Lemma 3.

Lemma 3 *If $\beta > 0$ and $\alpha^{(k)} \in [0,1]$ satisfies (6) with $x = x^{(k)}, v = v^{(k)}$, $b = \frac{1-\beta}{2\eta^{(k)}}, c = \frac{L\eta^{(k)}-\gamma}{\beta}$, or if $\beta = 0$ and $\alpha^{(k)} = 1$, then*

$$Q^{(k)} \le 2\eta^{(k)} \left[(L\eta^{(k)} - \gamma) \cdot (\epsilon^{(k)} - \epsilon_y^{(k)}) + \beta\tilde{\epsilon} \right]. \tag{7}$$

Proof First suppose $\beta > 0$. As by definition $y^{(k)} = \alpha^{(k)} x^{(k)} + (1 - \alpha^{(k)}) v^{(k)}$ and $L\eta^{(k)} \ge \gamma$, applying (6) yields

$$\begin{aligned} Q^{(k)} &= 2\beta\eta^{(k)} \left(\alpha^{(k)} \nabla f(y^{(k)})^\top (x^{(k)} - v^{(k)}) - \left(\alpha^{(k)}\right)^2 \frac{(1-\beta)\left\|x^{(k)} - v^{(k)}\right\|^2}{2\eta^{(k)}} \right) \\ &\le 2\beta\eta^{(k)} \left(\frac{L\eta^{(k)} - \gamma}{\beta} [f(x^{(k)}) - f(y^{(k)})] + \tilde{\epsilon} \right) = 2\eta^{(k)} \left([L\eta^{(k)} - \gamma] \cdot [\epsilon^{(k)} - \epsilon_y^{(k)}] + \beta\tilde{\epsilon} \right). \end{aligned}$$

Alternatively, suppose $\beta = 0$. Then $Q^{(k)} = 0$ as well; if we select $\alpha^{(k)} = 1$, then $y^{(k)} = x^{(k)}$ and (7) trivially holds for any $\tilde{\epsilon}$, as $\epsilon_y^{(k)} = \epsilon^{(k)}$. ■

Now, in Algorithm 3 we show how to efficiently compute an α satisfying inequality (6).

Algorithm 3: `BinaryLineSearch`$(f, x, v, L, b, c, \tilde{\epsilon}$, [guess])
Assumptions: f is L-smooth; $x, v \in \mathbb{R}^n$; $b, c, \tilde{\epsilon} \geq 0$; "guess" (optional) is in $[0, 1]$ if provided.
Define $g(\alpha) \triangleq f(\alpha x + (1-\alpha)v)$ and $p \triangleq b\|x - v\|^2$.
[**if** cg(guess) + guess $\cdot$ (g'(guess) $-$ guess $\cdot\, p$) $\leq cg(1) + \tilde{\epsilon}$ **then return** guess]
if $g'(1) \leq \tilde{\epsilon} + p$ **then return** 1;
else if $c = 0$ or $g(0) \leq g(1) + \tilde{\epsilon}/c$ **then return** 0;
$\tau \leftarrow 1 - \frac{\tilde{\epsilon}+p}{L\|x-v\|^2}$
lo $\leftarrow 0$, **hi** $\leftarrow \tau, \alpha \leftarrow \tau$
while $cg(\alpha) + \alpha(g'(\alpha) - \alpha p) > cg(1) + \tilde{\epsilon}$ **do**
  $\alpha \leftarrow$ (**lo** + **hi**)$/2$
  if $g(\alpha) \leq g(\tau)$ **then hi** $\leftarrow \alpha$;
  else lo $\leftarrow \alpha$;
end
return α

The basic idea behind Algorithm 3 is as follows: as in the proof of Lemma 2, let $g(\alpha) \triangleq f(\alpha x + (1-\alpha)v)$ be the restriction of the function f to the line from v to x. If either $g(0) \leq g(1)$, or g is decreasing at $\alpha = 1$, then (5) is immediately satisfied. If this does not happen, then $g(0)$ is greater than $g(1)$ but $g'(1) > 0$, which means that at some $\alpha \in (0, 1)$ with $g(\alpha) < g(1)$, the function g must switch from decreasing to increasing, and so $g'(\alpha) = 0$. Such a value of α also satisfies (5). Algorithm 3 uses binary search to exploit this type of relationship and thereby efficiently compute a value of α *approximately* satisfying (5) (i.e., satisfying (6)). In Lemma 4 (proved in Section G.1), we bound the maximum number of iterations that this algorithm can take until (6) holds and it thereby terminates. Note that derivatives of g may be computed cheaply by a simple finite difference scheme (e.g. approximating $g'(\alpha)$ by $h^{-1}(g(\alpha + h) - g(\alpha))$ for sufficiently small h); L-smoothness of f allows us to bound this approximation error as a function of h.

"guess" is an optional argument to the function. If used, the value of "guess" will be tested first, and chosen as the value of α if it satisfies (6).

Lemma 4 (Line Search Runtime) *For L-smooth $f : \mathbb{R}^n \to \mathbb{R}$, points $x, v \in \mathbb{R}^n$ and scalars $b, c, \tilde{\epsilon} \geq 0$, Algorithm 3 computes $\alpha \in [0, 1]$ satisfying (6) with at most*

$$7 + 2\left\lceil \log_2^+ \left((4 + c) \min\left\{ \frac{L^3}{b^3}, \frac{L\|x-v\|^2}{2\tilde{\epsilon}} \right\} \right) \right\rceil$$

function and gradient evaluations.

In summary, we achieve our accelerated quasar-convex minimization procedures by setting $\eta^{(k)}, \beta$, and ϵ appropriately and computing an $\alpha^{(k)}$ satisfying (6) via binary search (Algorithm 3). By carefully lower bounding the length of the interval of values of $\alpha^{(k)}$ satisfying (6), we ultimately show that this binary search only costs a logarithmic factor in the algorithm's overall runtime.

F. Algorithms

In this section, we develop algorithms for accelerated minimization of strongly quasar-convex functions and quasar-convex functions, respectively, and analyze their running times in terms of the number of function and gradient evaluations required.

F.1. Strongly Quasar-Convex Minimization

First, we provide and analyze our algorithm for (γ, μ)-strongly quasar-convex function minimization, where $\mu > 0$. The algorithm (Algorithm 4) is a carefully constructed instance of the general AGD framework (Algorithm 2).

As in the general AGD framework, the algorithm maintains two current points denoted $x^{(k)}$ and $v^{(k)}$ and at each step appropriately selects $y^{(k)} = \alpha^{(k)} x^{(k)} + (1 - \alpha^{(k)}) v^{(k)}$ as a convex combination of these two points. Intuitively, the algorithm iteratively seeks to decrease quadratic upper and lower bounds on the function value. L-smoothness of f implies for all $x, y \in \mathbb{R}^n$ that $f(x) \leq U_y(x) \triangleq f(y) + \nabla f(y)^\top (x - y) + \frac{L}{2} \|x - y\|^2$; we set $x^{(k+1)}$ to be the minimizer $y^{(k)} - \frac{1}{L} \nabla y^{(k)}$ of the upper bound $U_{y^{(k)}}$. Similarly, by (γ, μ) quasar-convexity, $f(x) \geq f(x^*) \geq \min_z L_y(z)$ for all $x, y \in \mathbb{R}^n$, where $L_y(x) \triangleq f(y) + \frac{1}{\gamma} \nabla f(y)^\top (x - y) + \frac{\mu}{2} \|x - y\|^2$. The minimizer of the lower bound $L_{y^{(k)}}$ is $y^{(k)} - \frac{1}{\gamma\mu} \nabla f(y^{(k)})$; we set $v^{(k+1)}$ to be a convex combination of $v^{(k)}$ and the minimizer of $L_{y^{(k)}}$.

Algorithm 4: Accelerated Strongly Quasar-Convex Function Minimization
input: L-smooth $f : \mathbb{R}^n \to \mathbb{R}$ that is (γ, μ)-strongly quasar-convex, with $\mu > 0$
input: Initial point $x^{(0)} \in \mathbb{R}^n$, number of iterations K, solution tolerance $\epsilon > 0$
return output of Algorithm 2 on f with initial point $x^{(0)}$ and parameter $\beta = 1 - \gamma \sqrt{\frac{\mu}{L}}$,
where for all k, $\eta^{(k)} = \eta \triangleq \frac{1}{\sqrt{\mu L}}$,
and $\alpha^{(k)} = \texttt{BinaryLineSearch}(f, x^{(k)}, v^{(k)}, L, b = \frac{1-\beta}{2\eta}, c = \frac{L\eta - \gamma}{\beta}, \tilde{\epsilon} = 0)$ **if** $\beta > 0$ **else** 1.

We leverage the analysis from Section E to analyze Algorithm 4. First, in Lemma 5 we show that the algorithm converges at the desired rate, by building off of Lemma 1 and using the specific parameter choices in Algorithm 4.

Lemma 5 (Strongly Quasar-Convex Convergence) *If f is L-smooth and (γ, μ)-strongly quasar-convex with minimizer x^*, $\gamma \in (0, 1]$, and $\mu > 0$, then in each iteration $k \geq 0$ of Algorithm 4,*

$$\epsilon^{(k+1)} + \frac{\mu}{2} r^{(k+1)} \leq \left(1 - \frac{\gamma}{\sqrt{\kappa}}\right) \left[\epsilon^{(k)} + \frac{\mu}{2} r^{(k)}\right], \tag{8}$$

where $\epsilon^{(k)} \triangleq f(x^{(k)}) - f(x^*)$, $r^{(k)} \triangleq \|v^{(k)} - x^*\|^2$, and $\kappa \triangleq \frac{L}{\mu}$. Therefore, if the number of iterations $K \geq \left\lceil \frac{\sqrt{\kappa}}{\gamma} \log^+ \left(\frac{3\epsilon^{(0)}}{\gamma\epsilon}\right) \right\rceil$, then the output $x^{(K)}$ of Algorithm 4 satisfies $f(x^{(K)}) \leq f(x^*) + \epsilon$.

Proof For all k, $\eta^{(k)} = \eta = \frac{1}{\sqrt{\mu L}} \geq \sqrt{\frac{\gamma}{(2-\gamma)L^2}} \geq \frac{\gamma}{L}$ as required by Algorithm 2, since $\frac{(2-\gamma)L}{\gamma} \geq \mu > 0$ by Observation 1 and $\frac{x}{2-x} \geq x^2$ for all $x \in [0, 1]$. Similarly, since $0 < \frac{\mu}{L} \leq \frac{2-\gamma}{\gamma}$

and $\gamma \in (0,1]$, we have $0 < \gamma\sqrt{\kappa} = \gamma\sqrt{\frac{\mu}{L}} \le \sqrt{\gamma(2-\gamma)} \le 1$, meaning that $\beta \in [0,1)$. Additionally, by construction, either $\beta = 0$ and $\alpha^{(k)} = 1$, or $\beta > 0$, $\alpha^{(k)} \in [0,1]$, and $(\alpha, x, y_\alpha, v) = (\alpha^{(k)}, x^{(k)}, y^{(k)}, v^{(k)})$ satisfies (6) with $b = \frac{1-\beta}{2\eta^{(k)}}$, $c = \frac{L\eta^{(k)}-\gamma}{\beta}$, $\tilde{\epsilon} = 0$. Consequently, by combining Lemmas 1 and 3, for each iteration $k \ge 0$ of Algorithm 4 we have

$$2\eta^2 L\epsilon^{(k+1)} + r^{(k+1)} \le \beta r^{(k)} + [(1-\beta) - \gamma\mu\eta]\, r_y^{(k)} + 2\eta\,[L\eta - \gamma]\,\epsilon^{(k)} + 2\beta\eta\tilde{\epsilon}$$

Substituting in $\eta = \frac{1}{\sqrt{\mu L}} = \frac{1-\beta}{\gamma\mu}$ and $\tilde{\epsilon} = 0$, this implies that

$$\frac{2}{\mu}\epsilon^{(k+1)} + r^{(k+1)} \le \beta r^{(k)} + \frac{2}{\sqrt{\mu L}}\left[\sqrt{\frac{L}{\mu}} - \gamma\right]\epsilon^{(k)} = \beta\left[r^{(k)} + \frac{2}{\mu}\epsilon^{(k)}\right].$$

Multiplying by $\mu/2$ and using the definition of β as $1 - \frac{\gamma}{\sqrt{\kappa}}$ yields (8). Now, by (8) and induction,

$$\epsilon^{(k)} + \frac{\mu}{2}r^{(k)} \le \left(1 - \frac{\gamma}{\sqrt{\kappa}}\right)^k \left[\epsilon^{(0)} + \frac{\mu}{2}r^{(0)}\right] \le \exp\left(-\frac{k\gamma}{\sqrt{\kappa}}\right)\left[\epsilon^{(0)} + \frac{\mu}{2}r^{(0)}\right].$$

Therefore, whenever $k \ge \frac{\sqrt{\kappa}}{\gamma}\log^+\left(\frac{\epsilon^{(0)}+\frac{\mu}{2}r^{(0)}}{\epsilon}\right)$ we have $\epsilon^{(k)} = f(x^{(k)}) - f(x^*) \le \epsilon$, as $r^{(k)} \ge 0$ always. By Corollary 1, $\frac{2\epsilon^{(0)}}{\gamma} \ge \frac{\mu}{2}r^{(0)}$, so it suffices to run $k \ge \left\lceil \frac{\sqrt{\kappa}}{\gamma}\log^+\left(\frac{3\epsilon^{(0)}}{\gamma\epsilon}\right)\right\rceil$ iterations. ■

Note that when f is $(1,\mu)$-strongly quasar-convex with $\mu > 0$, Lemma 5 implies that the number of *iterations* Algorithm 4 needs to find an ϵ-approximate minimizer of f is of the same order as the number of iterations required by standard AGD to find an ϵ-approximate minimizer of a μ-strongly convex function [40]. In each iteration of Algorithm 4, we compute $\alpha^{(k)}$ and then simply perform $O(1)$ vector operations to compute $y^{(k)}$, $x^{(k+1)}$, and $v^{(k+1)}$. Consequently, to obtain a complete bound on the overall complexity of Algorithm 4, all that remains is to bound the cost of computing $\alpha^{(k)}$, which we do using Lemma 4. This leads to Theorem 3.

Theorem 3 *If f is L-smooth and (γ,μ)-strongly quasar-convex with $\gamma \in (0,1]$ and $\mu > 0$, then Algorithm 4 produces an ϵ-optimal point after $O\left(\left(\gamma^{-1}\kappa^{1/2}\log\left(\gamma^{-1}\kappa\right)\log^+\left(\frac{f(x^{(0)})-f(x^*)}{\gamma\epsilon}\right)\right)\right)$ function and gradient evaluations.*

Proof Lemma 5 implies that $O\left(\frac{\sqrt{\kappa}}{\gamma}\log^+\left(\frac{\epsilon^{(0)}}{\gamma\epsilon}\right)\right)$ iterations are needed to get an ϵ-optimal point. Lemma 4 implies that each iteration uses $O\left(\log^+\left((1+c)\min\left\{\frac{L\|x-v\|^2}{\tilde{\epsilon}}, \frac{L^3}{b^3}\right\}\right)\right)$ function and gradient evaluations. In this case, $b = \frac{1-\beta}{2\eta} = \frac{\gamma\sqrt{\mu/L}}{2/\sqrt{\mu L}} = \frac{\gamma\mu}{2}$, $c = \frac{L\eta-\gamma}{\beta} = \frac{\sqrt{L/\mu}-\gamma}{1-\gamma\sqrt{\mu/L}} = \sqrt{\kappa} \ge \sqrt{\frac{\gamma}{2}}$, and $\tilde{\epsilon} = 0$. Thus, this reduces to $O(\log^+(\sqrt{\kappa}\frac{L^3}{\gamma^3\mu^3})) = O(\log(\frac{\kappa}{\gamma}))$. So, the total number of required function and gradient evaluations is $O\left(\frac{\sqrt{\kappa}}{\gamma}\log\left(\frac{\kappa}{\gamma}\right)\log^+\left(\frac{\epsilon^{(0)}}{\gamma\epsilon}\right)\right)$ as claimed.

Note that Lemma 5 shows that $x^{(k)}$ will be ϵ-optimal if $k = \left\lceil \frac{\sqrt{\kappa}}{\gamma}\log^+\left(\frac{3\epsilon^{(0)}}{\gamma\epsilon}\right)\right\rceil$, while the above argument shows that $O\left(\frac{\sqrt{\kappa}}{\gamma}\log\left(\frac{\kappa}{\gamma}\right)\log^+\left(\frac{\epsilon^{(0)}}{\gamma\epsilon}\right)\right)$ function and gradient evaluations are required

to compute such an $x^{(k)}$. Thus, Algorithm 4 produces *an* ϵ-optimal point using at most this many evaluations; however, of course, the algorithm need not return instantly and may still continue to run if the specified number of iterations K is larger. (Future iterates will also be ϵ-optimal.) ■

Standard AGD on L-smooth μ-strongly-convex functions requires $O\left(\kappa^{1/2}\log^+\left(\frac{f(x^{(0)})-f(x^*)}{\epsilon}\right)\right)$ function and gradient and evaluations to find an ϵ-optimal point [40]. Thus, as the class of L-smooth $(1,\mu)$-strongly quasar-convex functions contains the class of L-smooth μ-strongly convex functions, our algorithm requires only a $O(\log(\kappa))$ factor extra function and gradient evaluations in the smooth strongly convex case, while also being able to efficiently minimize a much broader class of functions than standard AGD.

F.2. Non-Strongly Quasar-Convex Minimization

Now, we provide and analyze our algorithm (Algorithm 5) for *non-strongly* quasar-convex function minimization, i.e. when $\mu = 0$. (This is exactly Algorithm 1 from the body, written precisely.) Once again, this algorithm is an instance of Algorithm 2, the general AGD framework, with a different choice of parameters. We assume $L > 0$, since otherwise quasar-convexity implies f is constant.

Algorithm 5: Accelerated Non-Strongly Quasar-Convex Function Minimization
input: L-smooth $f : \mathbb{R}^n \to \mathbb{R}$ that is γ-quasar-convex
input: Initial point $x^{(0)} \in \mathbb{R}^n$, number of iterations K, solution tolerance $\epsilon > 0$
Define $\omega^{(-1)} = 1$, $\omega^{(k)} = \frac{\omega^{(k-1)}}{2}\left(\sqrt{(\omega^{(k-1)})^2+4} - \omega^{(k-1)}\right)$ for $k \geq 0$.
return output of Algorithm 2 on f with initial point $x^{(0)}$ and parameter $\beta = 1$,
where for all k, $\eta^{(k)} = \frac{\gamma}{L\omega^{(k)}}$,
and $\alpha^{(k)} = \texttt{BinaryLineSearch}(f, x^{(k)}, v^{(k)}, L, b = 0, c = [L\eta^{(k)} - \gamma], \tilde{\epsilon} = \frac{\gamma\epsilon}{2})$.

Lemma 6 (Non-Strongly Quasar-Convex AGD Convergence) *If f is L-smooth and γ-quasar-convex with respect to a minimizer x^*, with $\gamma \in (0,1]$, then in each iteration $k \geq 0$ of Algorithm 5,*

$$\epsilon^{(k)} \leq \frac{8}{(k+2)^2}\left[\epsilon^{(0)} + \frac{L}{2\gamma^2}r^{(0)}\right] + \frac{\epsilon}{2}, \tag{9}$$

where $\epsilon^{(k)} \triangleq f(x^{(k)}) - f(x^*)$ and $r^{(k)} \triangleq \left\|v^{(k)} - x^*\right\|^2$. Therefore, if $R \geq \left\|x^{(0)} - x^*\right\|$ and the number of iterations $K \geq \left\lfloor 4\gamma^{-1}L^{1/2}R\epsilon^{-1/2}\right\rfloor$, then the output $x^{(K)}$ of Algorithm 5 satisfies $f(x^{(K)}) \leq f(x^*) + \epsilon$.

Combining the bound on the number of iterations from Lemma 6, and the bound from Lemma 4 on the number of function and gradient evaluations during the line search, leads to the bound in Theorem 1 on the total number of function and gradient evaluations required to find an ϵ-optimal point. For ease of exposition, the proofs of Lemma 6 and Theorem 1 are given in Section G.2.

Theorem 1 *If f is L-smooth and γ-quasar-convex with respect to a minimizer x^*, with $\gamma \in (0,1]$ and $\left\|x^{(0)} - x^*\right\| \leq R$, then Algorithm 1 produces an ϵ-optimal point after $O\left(\gamma^{-1}L^{1/2}R\epsilon^{-1/2}\log^+\left(\gamma^{-1}L^{1/2}R\epsilon^{-1/2}\right)\right)$ function and gradient evaluations.*

Note that standard AGD on the class of L-smooth *convex* functions requires $O\left(L^{1/2}R\epsilon^{-1/2}\right)$ function and gradient evaluations to find an ϵ-optimal point; so, again, our algorithm requires only a logarithmic factor more evaluations than does standard AGD.

F.3. Line Search Initial Guess

In the above analysis, we do not assume that an initial guess is passed to the line search (Algorithm 3). However, in some cases, a reasonable initial guess can speed up the algorithm by allowing the line search to be circumvented a large portion of the time. For instance, at each step k we could use the $\alpha^{(k)}$ specified in the standard version of AGD as a guess: $\frac{\sqrt{\kappa}}{\sqrt{\kappa}+1}$ in the strongly quasar-convex case (Algorithm 4), and $1-\omega^{(k)}$ in the non-strongly quasar-convex case (Algorithm 5). In this case, if f is convex or strongly convex (and thus $\gamma = 1$), the respective algorithms are equivalent to standard AGD [40], since the initial guess always satisfies the necessary condition (6) by convexity [in fact, it satisfies the stronger (5)] and will thus be chosen as the value of $\alpha^{(k)}$, and, aside from the choice of $\alpha^{(k)}$, the algorithms are otherwise equivalent to standard AGD when $\gamma = 1$. Moreover, even if f is nonconvex, checking this initial guess costs at most one extra function and gradient evaluation each per invocation of Algorithm 3.

G. Algorithm analysis

Here, we provide proofs deferred from Sections E-F.

G.1. Line search analysis

We first present a simple fact that is useful in our proofs of Lemmas 2 and 4.

Fact 2 *Suppose that $a < b$, $g : \mathbb{R} \to \mathbb{R}$ is differentiable, and that $g(a) \geq g(b)$. Then, there is a $c \in (a, b]$ such that $g(c) \leq g(b)$ and either $g'(c) = 0$, or $c = b$ and $g'(c) \leq 0$.*

Proof If $g'(b) \leq 0$, the claim is trivially true. If not, then $g'(b) > 0$, so the minimum value of g on $[a, b]$ is strictly less than $g(b)$ (and therefore strictly less than $g(a)$ as well). By continuity of g and the extreme value theorem, g must therefore attain its minimum on $[a, b]$ at some point in $c \in (a, b)$. By differentiability of g and the fact that c minimizes g, we then have $g'(c) = 0$. ■

Using Lemma 2 and another simple fact, we can prove Lemma 4.

Fact 3 *Suppose f is L-smooth. Define $g(\alpha) \triangleq f(\alpha x + (1-\alpha)v)$; then, g is $L\|x - v\|^2$-smooth.*

Proof By L-smoothness of f, $\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$ for all x, y. So,

$$\begin{aligned}\|\nabla f(y(\alpha_1)) - \nabla f(y(\alpha_2))\| &= \|\nabla f(\alpha_1 x + (1-\alpha_1)v) - \nabla f(\alpha_2 x + (1-\alpha_2)v)\| \\ &\leq L\|(\alpha_1 - \alpha_2)x - (\alpha_1 - \alpha_2)v\| = L|\alpha_1 - \alpha_2|\,\|x - v\|\,.\end{aligned}$$

By definition of g and the Cauchy-Schwarz inequality,

$$|g'(\alpha_1)-g'(\alpha_2)| = |\nabla f(y(\alpha_1))^\top (x-v) - \nabla f(y(\alpha_2))^\top (x-v)| \le \|\nabla f(y(\alpha_1)) - \nabla f(y(\alpha_2))\| \, \|x-v\| \,,$$

so $|g'(\alpha_1) - g'(\alpha_2)| \le L \|x-v\|^2 |\alpha_1 - \alpha_2|$ as desired. ■

Lemma 4 (Line Search Runtime) *For L-smooth $f : \mathbb{R}^n \to \mathbb{R}$, points $x, v \in \mathbb{R}^n$ and scalars $b, c, \tilde{\epsilon} \ge 0$, Algorithm 3 computes $\alpha \in [0,1]$ satisfying (6) with at most*

$$7 + 2\left\lceil \log_2^+ \left((4+c) \min\left\{ \tfrac{L^3}{b^3}, \tfrac{L\|x-v\|^2}{2\tilde{\epsilon}} \right\} \right) \right\rceil$$

function and gradient evaluations.

Proof Define $\hat{L} \triangleq L\|x-v\|^2$; by Fact 3, g is $\hat{L}$-smooth. Thus, if $g'(1) > \tilde{\epsilon} + p$, then $g'(t) > \tilde{\epsilon} + p - \hat{L}|1-t|$ for all $t \in \mathbb{R}$, so, recalling that $\tau = 1 - \frac{\tilde{\epsilon}+p}{\hat{L}}$, we have $g'(\tau) > 0$ and

$$\begin{aligned} g(\tau) - g(1) &= -\int_\tau^1 g'(t)\,dt < -\int_\tau^1 (\tilde{\epsilon} + p - \hat{L}(1-t))\,dt = (\tau-1)(\tilde{\epsilon}+p-\hat{L}) + \tfrac{\hat{L}(\tau^2-1)}{2} \\ &= -\tfrac{(\tilde{\epsilon}+p)^2}{\hat{L}} + (\tilde{\epsilon}+p) + \tfrac{(\tilde{\epsilon}+p)^2}{2\hat{L}} - (\tilde{\epsilon}+p) = -\tfrac{(\tilde{\epsilon}+p)^2}{2\hat{L}} \,. \end{aligned}$$

Recall that the loop termination condition in Algorithm 3 is $\alpha(g'(\alpha) - \alpha p) \le c(g(1) - g(\alpha)) + \tilde{\epsilon}$. First, we claim that the invariants $g(\mathbf{lo}) > g(\tau)$, $g(\mathbf{hi}) \le g(\tau)$, and $g'(\mathbf{hi}) > \tilde{\epsilon}$ hold at the start of every loop iteration. This is true at the beginning of the loop, since otherwise the algorithm would return before entering it. In the loop body, **hi** is only ever set to a new value α if $g(\alpha) \le g(\tau)$. If the loop does not subsequently terminate, this also implies $g'(\alpha) > \tilde{\epsilon}$ since then

$$\alpha(g'(\alpha) - \alpha p) > c(g(1) - g(\alpha)) + \tilde{\epsilon} \ge c(g(1) - g(\tau)) + \tilde{\epsilon} \ge \tilde{\epsilon} \,.$$

Similarly, **lo** is only ever set to a new value α if $g(\alpha) > g(\tau)$. Thus, these invariants indeed hold at the start of each loop iteration.

Now, suppose $\alpha = (\mathbf{lo} + \mathbf{hi})/2$ does not satisfy the termination condition. If $g(\alpha) \le g(\tau)$, this implies $g'(\alpha) > \tilde{\epsilon}$. As $g(\mathbf{lo}) > g(\tau) \ge g(\alpha)$, by Fact 2, there must be an $\hat{\alpha} \in (\mathbf{lo}, \alpha)$ with $g'(\hat{\alpha}) = 0$ and $g(\hat{\alpha}) \le g(\tau)$ [and thus satisfying the termination condition]. The algorithm sets **hi** to α, which will keep $\hat{\alpha}$ in the new search interval $[\mathbf{lo}, \alpha]$.

Similarly, if $g(\alpha) > g(\tau)$, then since $g(\tau) \ge g(\mathbf{hi})$ and $g'(\mathbf{hi}) > 0$, there must be an $\hat{\alpha} \in (\alpha, \mathbf{hi})$ with $g'(\hat{\alpha}) = 0$ and $g(\hat{\alpha}) \le g(\tau)$ [and thus satisfying the termination condition], by applying Fact 2. The algorithm sets **lo** to α, which will keep $\hat{\alpha}$ in the search interval. Thus, there is always at least one point $\hat{\alpha} \in [\mathbf{lo}, \mathbf{hi}]$ satisfying the termination condition.

In addition, note that if an interval $[z_1, z_2] \subseteq [0, \tau]$ of points satisfies the termination condition, then at every loop iteration, either the entire interval lies in $[\mathbf{lo}, \mathbf{hi}]$ or none of the interval does, i.e. either $[z_1, z_2] \subseteq [\mathbf{lo}, \mathbf{hi}]$ or $[z_1, z_2] \cap [\mathbf{lo}, \mathbf{hi}] = \emptyset$. The reason is that if a point α satisfies the termination

condition we terminate immediately. If not, then α is not in an interval of points satisfying the termination condition, so either $z_2 < \alpha$ or $z_1 > \alpha$. Thus, all intervals of points satisfying the termination condition either disjointly lie in the set of points that remain in our search interval, or the set of points we throw away (i.e. an interval of satisfying points never gets split).

Suppose that $\alpha \in [0, \tau]$, $g'(\alpha) = 0$, and $g(\alpha) \leq g(\tau)$. Note that if $p + \tilde{\epsilon} \geq \hat{L}$ and $g'(\alpha) = 0$, then by $\hat{L}$-smoothness of g, we have $g'(1) \leq \tilde{\epsilon} + p$. So, it must be the case that $p + \tilde{\epsilon} < \hat{L}$ if Algorithm 3 enters the binary search phase. By $\hat{L}$-Lipschitz continuity of g', we have that for all t, $|g'(t)| = |g'(t) - g'(\alpha)| \leq \hat{L}|t - \alpha|$ and $g(t) - g(\tau) \leq g(t) - g(\alpha) \leq \frac{\hat{L}}{2}(t - \alpha)^2$. So, for all $t \in [\alpha/2, \tau]$,

$$\begin{aligned} t(g'(t) - tp) + c(g(t) - g(\tau)) &\leq t(\hat{L}|t - \alpha| - (t - \alpha)p) + \tfrac{c\hat{L}}{2}(t - \alpha)^2 - \alpha tp \\ &\leq \left(\hat{L}(1 + \tfrac{c}{2}) + p\right) \cdot |t - \alpha| - \alpha^2 p/2 \, . \end{aligned}$$

Suppose $|t - \alpha| \leq \dfrac{\alpha^2 p/2 + \tilde{\epsilon}}{\hat{L}(1 + \frac{c}{2}) + p}$. Then, $\left(\hat{L}(1 + \frac{c}{2}) + p\right) \cdot |t - \alpha| - \alpha^2 p/2 \leq \tilde{\epsilon}$.

So, if $\alpha \in [0, \tau]$, $g'(\alpha) = 0$, and $g(\alpha) \leq g(\tau)$, then all points in the interval $\left[\alpha - \frac{\alpha^2 p/2 + \tilde{\epsilon}}{\hat{L}(1+c/2)+p}, \alpha + \frac{\alpha^2 p/2 + \tilde{\epsilon}}{\hat{L}(1+c/2)+p}\right] \cap [\alpha/2, \tau]$ also satisfy the termination condition. If $\frac{\alpha^2 p/2 + \tilde{\epsilon}}{\hat{L}(1+c/2)+p} \leq \alpha/2$, the lower bound of the first interval is $\geq \alpha/2$ and the intersection of the two intervals contains $[\alpha - \frac{\alpha^2 p/2 + \tilde{\epsilon}}{\hat{L}(1+c/2)+p}, \alpha]$. If not, then the first interval contains $[\alpha/2, \alpha]$ as does the second interval, so the intersection of the two intervals contains $[\alpha/2, \alpha]$. Therefore, the length of the interval is at least $\min\{\frac{\alpha}{2}, \frac{\alpha^2 p/2 + \tilde{\epsilon}}{\hat{L}(1+c/2)+p}\}$.

If $g'(\alpha) = 0$ and $g(\alpha) \leq g(\tau)$, then $g(0) \leq g(\tau) + \frac{\hat{L}}{2}\alpha^2$ by $\hat{L}$-smoothness. Since $g(\tau) + \frac{(p+\tilde{\epsilon})^2}{2\hat{L}} \leq g(1) < g(0)$, this implies $\alpha \geq \frac{p+\tilde{\epsilon}}{\hat{L}}$. Therefore, the interval length is at least $\min\left\{\dfrac{p + \tilde{\epsilon}}{2\hat{L}}, \dfrac{p^3/(2\hat{L}^2) + \tilde{\epsilon}}{(1 + c/2)\hat{L} + p}\right\}$ $\geq \min\left\{\dfrac{p + \tilde{\epsilon}}{2\hat{L}}, \dfrac{p^3/(2\hat{L}^2) + \tilde{\epsilon}}{(2 + c/2)\hat{L}}\right\} = \dfrac{p^3/(2\hat{L}^2) + \tilde{\epsilon}}{(2 + c/2)\hat{L}}$.

Note that $\dfrac{p^3/(2\hat{L}^2) + \tilde{\epsilon}}{(2 + c/2)\hat{L}} \geq \max\left\{\dfrac{p^3}{(4 + c)\hat{L}^3}, \dfrac{\tilde{\epsilon}}{(2 + c/2)\hat{L}}\right\}$. Recall that $\hat{L} = L\|x - v\|^2$ and $p = b\|x - v\|^2$. So, the interval length is at least

$$\begin{aligned} \tfrac{p^3/\hat{L}^2 + \tilde{\epsilon}}{(2+c/2)\hat{L}} &\geq \max\left\{\tfrac{p^3}{(4+c)\hat{L}^3}, \tfrac{\tilde{\epsilon}}{(2+c/2)\hat{L}}\right\} \\ &= \max\left\{\tfrac{b^3}{(4+c)L^3}, \tfrac{\tilde{\epsilon}}{(2+c/2)\hat{L}}\right\} \, . \end{aligned}$$

Since we know at least one such interval of points satisfying the termination condition is always contained within our current search interval, this implies that if we run the algorithm until the current search interval has length at most $\max\left\{\frac{b^3}{(4+c)L^3}, \frac{\tilde{\epsilon}}{(2+c/2)L\|x-v\|^2}\right\}$, we will terminate with a point satisfying the necessary condition. As we halve our search interval (which is initially $[0, \tau] \subset [0, 1]$) at every iteration, we must therefore terminate in $\leq \left\lceil \log_2^+ \left((4 + c) \min\left\{\frac{L^3}{b^3}, \frac{L\|x-v\|^2}{2\tilde{\epsilon}}\right\}\right)\right\rceil$ iterations.

Before each loop iteration (including the last which does not get executed when the termination condition is satisfied), we compute $g(\alpha)$ and $g'(\alpha)$, so there are two function and gradient evaluations per iteration; there are also (at most) five before the loop, to evaluate $g(0), g(1), g'(1), g(\text{guess}), g'(\text{guess})$. Thus, the total number of function and gradient evaluations made is at most $7 + 2\left\lceil \log_2^+ \left((4+c) \min\left\{ \frac{L^3}{b^3}, \frac{L\|x-v\|^2}{2\tilde{\epsilon}} \right\} \right) \right\rceil$.

Note that we define $\min\{x, +\infty\} = x$ for any $x \in \mathbb{R} \cup \{\pm\infty\}$. Note also that if $b = 0$ and $L = 0$, or if $\tilde{\epsilon} = 0$ and either $L = 0$ or $x = v$, the above expression is technically indeterminate; however, observe that g is constant in all of these cases, so one gradient evaluation is performed and the point $\alpha = 1$ is returned (or, if an initial guess is passed in, then there are three evaluations — $g(\text{guess})$, $g'(\text{guess})$, and $g(1)$ — and the point "guess" is returned). ■

G.2. Quasar-convex algorithm analysis

Lemma 7 *Suppose* $\omega^{(-1)} = 1$ *and* $\omega^{(k)} = \frac{1}{2}\left(\omega^{(k-1)}\left(\sqrt{\left(\omega^{(k-1)}\right)^2 + 4} - \omega^{(k-1)}\right)\right)$ *for* $k \geq 0$. *In the following sub-lemmas, we prove various simple properties of this sequence:*

Lemma 7.1 $\omega^{(k)} \leq \dfrac{4}{k+6}$ *for all* $k \geq 0$.

Proof The case $k = 0$ is clearly true as $\omega^{(0)} = \frac{\sqrt{5}-1}{2} < \frac{2}{3}$. Suppose that $\omega^{(i-1)} \leq \dfrac{4}{i+5}$ for some $i \geq 1$. $\omega^{(i)} = \dfrac{\omega^{(i-1)}}{2}\left(\sqrt{\left(\omega^{(i-1)}\right)^2 + 4} - \omega^{(i-1)}\right)$. Using the fact that $\sqrt{x^2+1} \leq 1 + \frac{x^2}{2}$ for all x and the fact that $\omega^{(i-1)} \in (0, 1)$,

$$\omega^{(i)} \leq \frac{\omega^{(i-1)}}{2}\left(2 - \omega^{(i-1)} + \frac{\left(\omega^{(i-1)}\right)^2}{2}\right) \leq \omega^{(i-1)}\left(1 - \frac{\omega^{(i-1)}}{4}\right).$$

If $y > 0$, then $x(1 - \frac{x}{4}) < \frac{4}{y+1}$ for all $0 \leq x \leq \frac{4}{y}$. Thus, setting $y = i + 5$ yields that $\omega^{(i)} \leq \frac{4}{i+6}$ by the inductive hypothesis. ■

Lemma 7.2 $\omega^{(k)} \geq \dfrac{1}{k+2}$ *for all* $k \geq 0$.

Proof The case $k = 0$ is clearly true as $\omega^{(0)} = \frac{\sqrt{5}-1}{2} > \frac{1}{2}$. Suppose that $\omega^{(i-1)} \geq \dfrac{1}{i+1}$ for some $i \geq 1$. Observe that the function $h(x) = \frac{1}{2}(x(\sqrt{x^2+4} - x))$ is increasing for all x. Therefore, $\omega^{(i)} = h(\omega^{(i-1)}) \geq h(\frac{1}{i+1}) = \frac{1}{2(i+1)}\left(\sqrt{\frac{1}{(i+1)^2} + 4} - \frac{1}{i+1}\right) = \frac{1}{2(i+1)^2}\left(\sqrt{4(i+1)^2 + 1} - 1\right)$.

Now, it just remains to show that $\sqrt{4x^2+1} \geq \frac{2x^2}{x+1} + 1$ for all $x \geq 0$. To prove this, note that $4x^2(x+1)^2 = 4x^4 + 8x^3 + 4x^2$, so

$$4x^2 + 1 = \frac{4x^4 + 8x^3 + 4x^2}{(x+1)^2} + 1 \geq \frac{4x^4 + 4x^3 + 4x^2}{(x+1)^2} + 1 = \left(\frac{2x^2}{x+1} + 1\right)^2 .$$

Thus, $\omega^{(i)} \geq \frac{1}{2(i+1)^2}\left(\sqrt{4(i+1)^2+1} - 1\right) \geq \frac{1}{2(i+1)^2} \cdot \frac{2(i+1)^2}{(i+2)} = \frac{1}{i+2}$. ■

Lemma 7.3 *$\omega^{(k)} \in (0,1)$ for all $k \geq 0$. Additionally, $\omega^{(k)} < \omega^{(k-1)}$ for all $k \geq 0$.*

Proof The fact that $\omega^{(k)} > 0$ follows from Lemma 7.2. To show the rest, we simply observe that $\frac{1}{2}(\sqrt{x^2+4} - x) < \frac{2}{2} = 1$ for all $x > 0$; as $\omega^{(-1)} = 1$ and $\omega^{(k)} = \frac{1}{2}(\sqrt{(\omega^{(k-1)})^2 + 4} - \omega^{(k-1)}) \cdot \omega^{(k-1)}$ for all $k \geq 0$, the result follows. ■

Lemma 7.4 *Define $s^{(k)} = 1 + \sum_{i=0}^{k-1} \frac{1}{\omega^{(i)}}$. Then, $\left(s^{(k)}\right)^{-1} \leq \frac{8}{(k+2)^2}$ for all $k \geq 0$.*

Proof Applying Lemma 7.1, $s^{(k)} \geq 1 + \sum_{i=0}^{k-1}\left(\frac{i+6}{4}\right) = \frac{k(k+11)+8}{8} \geq \frac{k(k+4)+4}{8} = \frac{1}{8}(k+2)^2$, and so $\left(s^{(k)}\right)^{-1} \leq \frac{8}{(k+2)^2}$. ■

Lemma 6 (Non-Strongly Quasar-Convex AGD Convergence) *If f is L-smooth and γ-quasar-convex with respect to a minimizer x^*, with $\gamma \in (0,1]$, then in each iteration $k \geq 0$ of Algorithm 5,*

$$\epsilon^{(k)} \leq \frac{8}{(k+2)^2}\left[\epsilon^{(0)} + \frac{L}{2\gamma^2} r^{(0)}\right] + \frac{\epsilon}{2} , \tag{9}$$

where $\epsilon^{(k)} \triangleq f(x^{(k)}) - f(x^*)$ and $r^{(k)} \triangleq \left\|v^{(k)} - x^*\right\|^2$. Therefore, if $R \geq \left\|x^{(0)} - x^*\right\|$ and the number of iterations $K \geq \left\lfloor 4\gamma^{-1} L^{1/2} R \epsilon^{-1/2} \right\rfloor$, then the output $x^{(K)}$ of Algorithm 5 satisfies $f(x^{(K)}) \leq f(x^*) + \epsilon$.

Proof In the non-strongly quasar-convex case, $\mu = 0$ and $\beta = 1$. For all k, $\eta^{(k)} = \frac{\gamma}{L\omega^{(k)}} \geq \frac{\gamma}{L}$ since $\omega^{(k)} \in (0,1)$ by Lemma 7.3. Additionally, $\alpha^{(k)}$ is in $[0,1]$ and $(\alpha, x, y_\alpha, v) = (\alpha^{(k)}, x^{(k)}, y^{(k)}, v^{(k)})$ satisfies (6) with $b = \frac{1-\beta}{2\eta^{(k)}} = 0$, $c = \frac{L\eta^{(k)} - \gamma}{\beta} = L\eta^{(k)} - \gamma$ by construction. Lemmas 1 and 3 thus imply that for all $k \geq 0$,

$$2(\eta^{(k)})^2 L \epsilon^{(k+1)} + r^{(k+1)} \leq r^{(k)} + 2\eta^{(k)}\left(L\eta^{(k)} - \gamma\right)\epsilon^{(k)} + 2\eta^{(k)}\tilde{\epsilon} . \tag{10}$$

Define $A^{(k)} \triangleq 2\left(\eta^{(k)}\right)^2 L - 2\eta^{(k)}\gamma$. So, $(A^{(k)} + 2\eta^{(k)}\gamma)\epsilon^{(k+1)} + r^{(k+1)} \leq A^{(k)}\epsilon^{(k)} + r^{(k)} + 2\eta^{(k)}\tilde{\epsilon}$.

Notice that $(\omega^{(k+1)})^2 = (1-\omega^{(k+1)})(\omega^{(k)})^2$ and $\omega^{(k)} \in (0,1)$ for all $k \geq 0$. So,

$$A^{(k+1)} - (A^{(k)} + 2\eta^{(k)}\gamma) = 2(\eta^{(k+1)})^2 L - 2\eta^{(k+1)}\gamma - 2(\eta^{(k)})^2 L \quad =$$
$$2\left(\frac{\gamma^2 L}{L^2(\omega^{(k+1)})^2} - \frac{\gamma^2}{L\omega^{(k+1)}} - \frac{\gamma^2 L}{L^2(\omega^{(k)})^2}\right) = \frac{2\gamma^2}{L}\left(\frac{1-\omega^{(k+1)}}{(\omega^{(k+1)})^2} - \frac{1}{(\omega^{(k)})^2}\right) \qquad = 0.$$

So, $A^{(k+1)} = A^{(k)} + 2\eta^{(k)}\gamma = 2(\eta^{(k)})^2 L$.

Also, $A^{(0)} = 2(\eta^{(0)})^2 L - 2\eta^{(0)}\gamma = 2\frac{\gamma^2}{L(\omega^{(0)})^2} - 2\frac{\gamma^2}{L\omega^{(0)}} = \frac{2\gamma^2}{L}$ as $\omega^{(0)} = \frac{\sqrt{5}-1}{2}$.

Thus, by induction on k, $A^{(k)} = \frac{2\gamma^2}{L} + 2\gamma\sum_{i=0}^{k-1}\eta^{(i)} = \frac{2\gamma^2}{L}s^{(k)}$, where $s^{(k)} \triangleq \left(1 + \sum_{i=0}^{k-1}\frac{1}{\omega^{(i)}}\right)$.

From (10) and the fact that $A^{(k+1)} = 2(\eta^{(k)})^2 L$, we have

$$A^{(k)}\epsilon^{(k)} + r^{(k)} \leq A^{(k-1)}\epsilon^{(k-1)} + r^{(k-1)} + 2\eta^{(k-1)}\tilde{\epsilon} \leq \cdots \leq A^{(0)}\epsilon^{(0)} + r^{(0)} + 2\tilde{\epsilon}\sum_{i=0}^{k-1}\eta^{(i)}. \quad (11)$$

So, as $r^{(k)} \geq 0$,

$$\begin{aligned}
\epsilon^{(k)} &\leq (A^{(k)})^{-1}\left(A^{(0)}\epsilon^{(0)} + r^{(0)}\right) + 2(A^{(k)})^{-1}\tilde{\epsilon}\sum_{i=0}^{k-1}\eta^{(i)} \\
&= \frac{L}{2\gamma^2}(s^{(k)})^{-1}\left(\frac{2\gamma^2}{L}\epsilon^{(0)} + r^{(0)}\right) + \left(2\gamma\left(\frac{\gamma}{L} + \sum_{i=0}^{k-1}\eta^{(i)}\right)\right)^{-1}\left(2\tilde{\epsilon}\sum_{i=0}^{k-1}\eta^{(i)}\right) \\
&\leq (s^{(k)})^{-1}\left(\epsilon^{(0)} + \frac{L}{2\gamma^2}r^{(0)}\right) + \gamma^{-1}\tilde{\epsilon}
\end{aligned}$$

Now, $\tilde{\epsilon} = \frac{\gamma\epsilon}{2}$ by definition and $\left(s^{(k)}\right)^{-1} \leq \frac{8}{(k+2)^2}$ by Lemma 7.1, which yields the bound on $\epsilon^{(k)}$.

For the iteration bound, we simply require K large enough such that $\frac{8}{(K+2)^2}\left(\epsilon^{(0)} + \frac{L}{2\gamma^2}r^{(0)}\right) \leq \frac{\epsilon}{2}$. Observe that as $f(x^{(0)}) \leq f(x^*) + \frac{L}{2}\left\|x^{(0)} - x^*\right\|^2$ by Fact 1, $\epsilon^{(0)} \leq \frac{L}{2}r^{(0)} \leq \frac{L}{2\gamma^2}r^{(0)}$.

So, it suffices to have $\frac{8}{(K+2)^2}\left(\frac{L}{\gamma^2}r^{(0)}\right) \leq \frac{\epsilon}{2}$. Rearranging, this is equivalent to $K + 2 \geq 4\gamma^{-1}L^{1/2}R\epsilon^{-1/2}$, as $r^{(0)} = R^2$. As K must be a nonnegative integer, it suffices to have $K \geq \left\lfloor 4\gamma^{-1}L^{1/2}R\epsilon^{-1/2}\right\rfloor$. ■

Theorem 1 *If f is L-smooth and γ-quasar-convex with respect to a minimizer x^*, with $\gamma \in (0,1]$ and $\left\|x^{(0)} - x^*\right\| \leq R$, then Algorithm 1 produces an ϵ-optimal point after $O\left(\gamma^{-1}L^{1/2}R\epsilon^{-1/2}\log^+\left(\gamma^{-1}L^{1/2}R\epsilon^{-1/2}\right)\right)$ function and gradient evaluations.*

Proof Lemma 6 implies $O(\gamma^{-1}L^{1/2}R\epsilon^{-1/2})$ iterations are needed to get an ϵ-optimal point. Lemma 4 implies that each line search uses $O\left(\log^+\left((1+c)\min\left\{\frac{L\left\|x^{(k)}-v^{(k)}\right\|^2}{\tilde{\epsilon}}, \frac{L^3}{b^3}\right\}\right)\right)$ function and gradient evaluations. In this case, $b = 0$, $c = L\eta^{(k)} - \gamma = \gamma\left(\frac{1}{\omega^{(k)}} - 1\right)$, and $\tilde{\epsilon} = \frac{\gamma\epsilon}{2}$. By Lemma 7.2 and 7.3, $1 < \frac{1}{\omega^{(k)}} \leq k+2$ for all $k \geq 0$. Thus, the number of function and gradient evaluations required for the line search at iteration k of Algorithm 5 is $O\left(\log^+\left((\gamma k+1)\frac{L\left\|x^{(k)}-v^{(k)}\right\|^2}{\gamma\epsilon}\right)\right)$.

Now, we bound $\left\|x^{(k)} - v^{(k)}\right\|^2$. To do so, we first bound $\left\|v^{(k)} - x^*\right\|^2 = r^{(k)}$. Recall that equation (11) in the proof of Lemma 6 says that $A^{(k)}\epsilon^{(k)} + r^{(k)} \leq A^{(0)}\epsilon^{(0)} + r^{(0)} + 2\tilde{\epsilon}\sum_{i=0}^{k-1}\eta^{(i)}$, where $A^{(j)} \triangleq \frac{2\gamma^2}{L}\left(1 + \sum_{i=0}^{j-1}\frac{1}{\omega^{(i)}}\right)$. As $A^{(k)}, \epsilon^{(k)} \geq 0$, this means that

$$r^{(k)} \leq A^{(0)}\epsilon^{(0)} + r^{(0)} + 2\tilde{\epsilon}\sum_{i=0}^{k-1}\eta^{(i)} = \frac{2\gamma^2}{L}\epsilon^{(0)} + r^{(0)} + \frac{\gamma^2\epsilon}{L}\sum_{i=0}^{k-1}\frac{1}{\omega^{(i)}}\,,$$

using that $\eta^{(i)} = \frac{\gamma}{L\omega^{(i)}}$, $\tilde{\epsilon} = \frac{\gamma\epsilon}{2}$, and $A^{(0)} = \frac{2\gamma^2}{L}$ (as previously shown in the proof of Lemma 6). Now, by Lemma 7.2 we have that $\sum_{i=0}^{k-1}\frac{1}{\omega^{(i)}} \leq \sum_{i=0}^{k-1}(i+2) = \frac{k(k+3)}{2}$, and by L-smoothness of f and Fact 1 we have that $\epsilon^{(0)} \leq \frac{L}{2}r^{(0)} \leq \frac{L}{2\gamma^2}r^{(0)}$. Thus, for all $k \geq 1$, we have

$$r^{(k)} \leq 2r^{(0)} + \frac{\gamma^2\epsilon k(k+3)}{2L} \leq 2(R^2 + \frac{\gamma^2\epsilon k^2}{L})\,,$$

as $r^{(0)} = R^2$ and $k+3 \leq 4k$ for all $k \geq 1$. In fact, the above holds for $k = 0$ as well, because $r^{(k)}$ is simply $r^{(0)}$ in this case.

By the triangle inequality, $\left\|v^{(k)} - v^{(k-1)}\right\| \leq \left\|v^{(k)} - x^*\right\| + \left\|v^{(k-1)} - x^*\right\| \leq 2\sqrt{2(R^2 + \frac{\gamma^2\epsilon k^2}{L})}$. Since $\beta = 1$, we have that $v^{(k-1)} - \eta^{(k-1)}\nabla f(y^{(k-1)})$ and so $\left\|v^{(k)} - v^{(k-1)}\right\| = \eta^{(k-1)}\left\|\nabla f(y^{(k-1)})\right\|$. Thus,

$$\left\|\nabla f(y^{(k-1)})\right\| \leq (\eta^{(k-1)})^{-1}\cdot 2\sqrt{2(R^2 + \tfrac{\gamma^2\epsilon k^2}{L})} = L\omega^{(k-1)}\gamma^{-1}\sqrt{8(R^2 + \tfrac{\gamma^2\epsilon k^2}{L})}\,. \tag{12}$$

Now, by definition of $x^{(k)}$, $v^{(k)}$, and $y^{(k-1)}$,

$$\begin{aligned}
x^{(k)} - v^{(k)} &= y^{(k-1)} - \tfrac{1}{L}\nabla f(y^{(k-1)}) - v^{(k)} \\
&= \alpha^{(k-1)}x^{(k-1)} + (1-\alpha^{(k-1)})v^{(k-1)} - \tfrac{1}{L}\nabla f(y^{(k-1)}) - v^{(k)} \\
&= \alpha^{(k-1)}x^{(k-1)} + (1-\alpha^{(k-1)})v^{(k-1)} - \tfrac{1}{L}\nabla f(y^{(k-1)}) - \left(v^{(k-1)} - \eta^{(k-1)}\nabla f(y^{(k-1)})\right) \\
&= \alpha^{(k-1)}(x^{(k-1)} - v^{(k-1)}) + (\eta^{(k-1)} - \tfrac{1}{L})\nabla f(y^{(k-1)})\,.
\end{aligned}$$

Therefore,

$$\begin{aligned}\left\|x^{(k)}-v^{(k)}\right\| &\le \alpha^{(k-1)}\left\|x^{(k-1)}-v^{(k-1)}\right\| + \left|\eta^{(k-1)}-\tfrac{1}{L}\right|\cdot\left\|\nabla f(y^{(k-1)})\right\| \\ &\le \left\|x^{(k-1)}-v^{(k-1)}\right\| + \left(\eta^{(k-1)}+\tfrac{1}{L}\right)\cdot\left\|\nabla f(y^{(k-1)})\right\| \\ &\le \left\|x^{(k-1)}-v^{(k-1)}\right\| + \tfrac{2}{L\omega^{(k-1)}}\cdot\left\|\nabla f(y^{(k-1)})\right\| \\ &\le \left\|x^{(k-1)}-v^{(k-1)}\right\| + \gamma^{-1}\sqrt{32(R^2+\tfrac{\gamma^2\epsilon k^2}{L})} \\ &\le \left\|x^{(k-1)}-v^{(k-1)}\right\| + \sqrt{32}\gamma^{-1}\left(R+\gamma k\sqrt{\tfrac{\epsilon}{L}}\right)\ ,\end{aligned}$$

where the first inequality is the triangle inequality, the third inequality uses that $\eta^{(k-1)} = \frac{\gamma}{L\omega^{(k-1)}}$ and that $\gamma, \omega^{(k-1)} \in (0,1]$, the fourth inequality uses (12), and the final inequality uses that $\sqrt{a+b} \le \sqrt{a}+\sqrt{b}$ for any $a, b \ge 0$.

As this holds for all $k \ge 1$, we have by induction that for all $k \ge 0$,

$$\left\|x^{(k)}-v^{(k)}\right\| \le \left\|x^{(0)}-v^{(0)}\right\| + \sum_{j=1}^{k}\sqrt{32}\gamma^{-1}\left(R+\gamma j\sqrt{\tfrac{\epsilon}{L}}\right) = \sqrt{32}\gamma^{-1}\sum_{j=1}^{k}\left(R+\gamma j\sqrt{\tfrac{\epsilon}{L}}\right)\ ,$$

since $x^{(0)} = v^{(0)}$. Simplification yields $\left\|x^{(k)}-v^{(k)}\right\| \le \sqrt{32}k\gamma^{-1}R + \sqrt{8}k(k+1)\sqrt{\frac{\epsilon}{L}}$. For all $k \ge 1$, it is the case that $k+1 \le 2k$, so $\left\|x^{(k)}-v^{(k)}\right\| \le \sqrt{32}\left(k\gamma^{-1}R + k^2\sqrt{\frac{\epsilon}{L}}\right)$; this inequality holds for $k = 0$ as well, as $\left\|x^{(0)}-v^{(0)}\right\| = 0$ in this case.

Suppose $k \le \left\lfloor 4\gamma^{-1}L^{1/2}R\epsilon^{-1/2}\right\rfloor$. Then

$$\begin{aligned}\left\|x^{(k)}-v^{(k)}\right\| &\le \sqrt{32}\left(4\gamma^{-1}L^{1/2}R\epsilon^{-1/2}\cdot\gamma^{-1}R + 16\gamma^{-2}LR^2\epsilon^{-1}\cdot\sqrt{\tfrac{\epsilon}{L}}\right) \\ &= 80\sqrt{2}\cdot\gamma^{-2}L^{1/2}R^2\epsilon^{-1/2}\ .\end{aligned}$$

Recall that the line search at iteration k requires $O\left(\log^+\left((\gamma k+1)\frac{L\left\|x^{(k)}-v^{(k)}\right\|^2}{\gamma\epsilon}\right)\right)$ function and gradient evaluations. $(\gamma k+1)\frac{L\left\|x^{(k)}-v^{(k)}\right\|^2}{\gamma\epsilon} \le (4L^{1/2}R\epsilon^{-1/2}+1)\cdot 12800(\gamma^{-5}L^2R^4\epsilon^{-2})$. Therefore, each line search indeed requires $O\left(\log^+\left(\gamma^{-1}L^{1/2}R\epsilon^{-1/2}\right)\right)$ function and gradient evaluations.

As the number of iterations k is $O(\gamma^{-1}L^{1/2}R\epsilon^{-1/2})$, the total number of function and gradient evaluations required is thus $O\left(\gamma^{-1}L^{1/2}R\epsilon^{-1/2}\log^+\left(\gamma^{-1}L^{1/2}R\epsilon^{-1/2}\right)\right)$, as claimed.

As in the strongly convex case, the algorithm may continue to run if the specified number of iterations K is larger; however, this theorem combined with Lemma 6 shows that $x^{(k)}$ will be ϵ-optimal if $k = \left\lfloor 4\gamma^{-1}L^{1/2}R\epsilon^{-1/2}\right\rfloor$, and this $x^{(k)}$ will be produced using $O\left(\gamma^{-1}L^{1/2}R\epsilon^{-1/2}\log^+\left(\gamma^{-1}L^{1/2}R\epsilon^{-1/2}\right)\right)$ function and gradient evaluations. (Iterates $x^{(k')}$ with $k' > \left\lfloor 4\gamma^{-1}L^{1/2}R\epsilon^{-1/2}\right\rfloor$ will also be ϵ-optimal.) ■

Remark 1 *If f is L-smooth and γ-quasar-convex with $\gamma \in (0,1]$ and $\left\|x^{(0)} - x^*\right\| \leq R$, then gradient descent with step size $\frac{1}{L}$ returns a point x with $f(x) \leq f(x^*) + \epsilon$ after $O\left(\gamma^{-1} L R^2 \epsilon^{-1}\right)$ function and gradient evaluations.*

Proof See Theorem 1 in [24]. ■

H. The structure of quasar-convex functions

In this section, we state and prove some useful properties of quasar-convex functions.

Lemma 8 *Let $f : \mathbb{R}^n \to \mathbb{R}$ be differentiable with a minimizer x^*. Then, the following two statements:*

$$f(tx^* + (1-t)x) + t\left(1 - \frac{t}{2-\gamma}\right)\frac{\gamma\mu}{2}\|x^* - x\|^2 \leq \gamma t f(x^*) + (1-\gamma t) f(x) \;\; \forall x \in \mathbb{R}^n,\, t \in [0,1] \tag{13}$$

$$f(x^*) \geq f(x) + \frac{1}{\gamma}\nabla f(x)^\top (x^* - x) + \frac{\mu}{2}\|x^* - x\|^2 \;\; \forall x \in \mathbb{R}^n \tag{14}$$

are equivalent for all $\mu \geq 0$, $\gamma \in (0,1]$.

Proof First, we prove that (14) implies (13).

Suppose (14) holds and $\mu = 0$. Let $x \in \mathbb{R}^n$ be arbitrary and for all $t \in [0,1]$ let $x_t \triangleq (1-t)x^* + tx$ and let $g(t) \triangleq f(x_t) - f(x^*)$. Since $g'(t) = \nabla f(x_t)^\top (x - x^*)$ and $x^* - x_t = -t(x^* - x)$, substituting these equalities into (14) yields that $g(t) \leq \frac{t}{\gamma} g'(t)$ for all $t \in [0,1]$.

Rearranging, we see that the inequality in (13) [for fixed x] is equivalent to the condition that $g(t) \leq \ell(t)$ for all $t \in [0,1]$, where $\ell(t) \triangleq (1 - \gamma(1-t))g(1)$. We proceed by contradiction: suppose that for some $\alpha \in [0,1]$ it is the case that $g(\alpha) > \ell(\alpha)$. Note that $\alpha > 0$ necessarily. Let β be the minimum element of the set $\{t \in [\alpha, 1] : g(t) = \ell(t)\}$. Since $g(1) = \ell(1)$, such a β exists with $\alpha < \beta$. Consequently, for all $t \in (\alpha, \beta)$ we have $g(t) \geq \ell(t)$ and so

$$\int_\alpha^\beta g'(t)\, dt = g(\beta) - g(\alpha) < \ell(\beta) - \ell(\alpha) = \gamma(\beta - \alpha) g(1) \tag{15}$$

and

$$(\beta - \alpha) g(1) = \int_\alpha^\beta \frac{\ell(t)}{1 - \gamma(1-t)}\, dt \leq \int_\alpha^\beta \frac{g(t)}{1 - \gamma(1-t)}\, dt\,. \tag{16}$$

Combining (15) and (16) and using that $g(t) \leq \frac{t}{\gamma} g'(t)$, we have

$$\int_\alpha^\beta \left[\frac{1}{t} - \frac{1}{1 - \gamma(1-t)}\right] g(t)\, dt \leq \int_\alpha^\beta \frac{g'(t)}{\gamma}\, dt - \int_\alpha^\beta \frac{g(t)}{1 - \gamma(1-t)}\, dt < 0$$

As $g(t) = f(x_t) - f(x^*) \geq 0$ and $1/t \geq 1/(1 - \gamma(1-t))$ for all $t \in [\alpha, \beta] \subset (0,1]$, we have a contradiction.

Now, suppose $\mu > 0$. Define $h(x) \triangleq f(x) - \frac{\gamma\mu}{2(2-\gamma)}\|x^* - x\|^2$. Observe that $h(x^*) = f(x^*)$, $\nabla h(x) = \nabla f(x) - \frac{\gamma\mu}{2-\gamma}(x - x^*)$, and $\nabla h(x)^\top (x^* - x) = \nabla f(x)^\top (x^* - x) + \frac{\gamma\mu}{2-\gamma}\|x^* - x\|^2$. Thus, by algebraic simplification and then application of (14) by assumption,

$$\begin{aligned} h(x) + \frac{1}{\gamma}\nabla h(x)^\top (x^* - x) &= f(x) - \frac{\gamma\mu}{2(2-\gamma)}\|x^* - x\|^2 + \frac{1}{\gamma}\nabla f(x)^\top (x^* - x) + \frac{\mu}{2-\gamma}\|x^* - x\|^2 \\ &= f(x) + \frac{1}{\gamma}\nabla f(x)^\top (x^* - x) + \frac{\mu}{2}\|x^* - x\|^2 \left(-\frac{\gamma}{2-\gamma} + \frac{2}{2-\gamma}\right) \\ &= f(x) + \frac{1}{\gamma}\nabla f(x)^\top (x^* - x) + \frac{\mu}{2}\|x^* - x\|^2 \\ &\leq f(x^*) = h(x^*) \,. \end{aligned}$$

As we earlier showed that (14) implies (13) in the $\mu = 0$ case, we have that

$$h(tx^* + (1-t)x) \leq \gamma t h(x^*) + (1 - \gamma t)h(x) \,.$$

Substituting in the definition of h:

$$\begin{aligned} &f(tx^* + (1-t)x) - \frac{\gamma\mu}{2(2-\gamma)}\|x^* - tx^* - (1-t)x\|^2 \\ &\leq \gamma t f(x^*) + (1-\gamma t)f(x) - (1-\gamma t)\frac{\gamma\mu}{2(2-\gamma)}\|x^* - x\|^2 \,. \end{aligned}$$

Rearranging terms and simplifying yields

$$\begin{aligned} &f(tx^* + (1-t)x) + \frac{\gamma\mu}{2(2-\gamma)}\left((1-\gamma t)\|x^* - x\|^2 - (1-t)^2\|x^* - x\|^2\right) \\ &\leq \gamma t f(x^*) + (1-\gamma t)f(x) \,. \end{aligned}$$

Finally, $(1-\gamma t) - (1-t)^2 = t((2-\gamma) - t)$, which gives the desired result.

Now, we prove that (13) implies (14).

This time, define $g(t) \triangleq f(tx^* + (1-t)x)$. For $t \in [0, 1)$, $g'(t) = \nabla f(tx^* + (1-t)x)^\top (x^* - x)$. By assumption, $g(t) + t\left(1 - \frac{t}{2-\gamma}\right)\frac{\gamma\mu}{2}\|x^* - x\|^2 \leq \gamma t g(1) + (1-\gamma t)g(0)$ for all $t \in [0, 1]$, so $g(1) \geq g(0) + \frac{g(t) - g(0)}{\gamma t} + \left(1 - \frac{t}{2-\gamma}\right)\frac{\mu}{2}\|x^* - x\|^2$ for all $t \in (0, 1]$. Taking the limit as $t \downarrow 0$ yields $f(x^*) = g(1) \geq g(0) + \frac{1}{\gamma}g'(0) + \frac{\mu}{2}\|x^* - x\|^2 = f(x) + \frac{1}{\gamma}\nabla f(x)^\top (x^* - x) + \frac{\mu}{2}\|x^* - x\|^2$. ■

Note that when $\gamma = 1, \mu = 0$, and (13) is required to hold for *all* minimizers of f, it becomes the standard definition of star-convexity [42].

Corollary 1 *If f is (γ, μ)-strongly quasar-convex with minimizer x^*, then*

$$f(x) \geq f(x^*) + \frac{\gamma\mu}{2(2-\gamma)}\|x^* - x\|^2 \,, \ \forall x$$

Proof Plug in $t = 1$ to (13) to get

$$f(x^*) + \left(1 - \frac{1}{2-\gamma}\right) \frac{\gamma\mu}{2} \|x^* - x\|^2 \le \gamma f(x^*) + (1-\gamma) f(x) \,.$$

Simplifying yields

$$f(x) \ge f(x^*) + \left(1 - \frac{1}{2-\gamma}\right) \frac{\gamma\mu}{2(1-\gamma)} \|x^* - x\|^2 = f(x^*) + \frac{\gamma\mu}{2(2-\gamma)} \|x^* - x\|^2 \,. \quad \blacksquare$$

Observation 1 *If f is (γ, μ)-strongly quasar-convex, then f is not L-smooth for any $L < \frac{\gamma\mu}{2-\gamma}$.*

Proof If f is (γ, μ)-strongly quasar-convex, Corollary 1 says that $f(x) \ge f(x^*) + \frac{\gamma\mu}{2(2-\gamma)} \|x^* - x\|^2$ for all x. If f is L-smooth, Fact 1 says that $f(x) \le f(x^*) + \frac{L}{2} \|x^* - x\|^2$ for all x.

Thus, if f is (γ, μ)-strongly quasar-convex and L-smooth, we have $\frac{\gamma\mu}{2(2-\gamma)} \|x^* - x\|^2 \le \frac{L}{2} \|x^* - x\|^2$ for all x, which means that we must have $L \ge \frac{\gamma\mu}{2-\gamma}$. $\blacksquare$

Observation 2 *If f is (γ, μ)-strongly quasar-convex with $\mu > 0$, f has a unique minimizer.*

Proof By Corollary 1, $f(x) > f(x^*)$ if $\mu > 0$ and $x \ne x^*$, implying that x minimizes f iff $x = x^*$. $\blacksquare$

Observation 3 *Suppose f is differentiable and (γ, μ)-strongly quasar-convex. Then f is also $(\theta\gamma, \mu/\theta)$-strongly quasar-convex for any $\theta \in (0, 1]$.*

Proof (γ, μ)-strong quasar-convexity states that $0 \ge f(x^*) - f(x) \ge \frac{1}{\gamma} \nabla f(x)^\top (x^* - x) + \frac{\mu}{2} \|x^* - x\|^2$ for some x^* and all x in the domain of f. Multiplying by $\frac{1}{\theta} - 1 \ge 0$, it follows that

$f(x^*) \ge f(x) + \frac{1}{\gamma} \nabla f(x)^\top (x^* - x) + \frac{\mu}{2} \|x - x^*\|^2 \ge f(x) + \frac{1}{\gamma\theta} \nabla f(x)^\top (x^* - x) + \frac{\mu}{2\theta} \|x^* - x\|^2$.

Note that any (γ, μ)-strongly quasar-convex function is also $(\gamma, \tilde{\mu})$-strongly quasar-convex for any $\tilde{\mu} \in [0, \mu]$. Thus, the restriction $\gamma \in (0, 1]$ in the definition of quasar-convexity may be made without any loss of generality compared to the restriction $\gamma > 0$. $\blacksquare$

Observation 4 *The parameter γ is a dimensionless quantity, in the sense that if f is γ-quasar-convex on $\mathbb{R}^n$, the function $g(x) \triangleq a \cdot f(bx)$ is also γ-quasar-convex on $\mathbb{R}^n$, for any $a \ge 0, b \in \mathbb{R}$.*

Proof If a or b is 0, then g is constant so the claim is trivial. Now suppose $a, b \ne 0$. Let x^* denote the quasar-convex point of f. Observe that as x^* minimizes f, x^*/b minimizes g. By (13), for all

$x \in \mathbb{R}^n$ we have

$$\begin{aligned}\tfrac{1}{a}g((tx^* + (1-t)x)/b) &= f(tx^* + (1-t)x) \\ &\le \gamma t f(x^*) + (1-\gamma t) f(x) \\ &= \gamma t \cdot \tfrac{1}{a} g(x^*/b) + (1-\gamma t) \cdot \tfrac{1}{a} g(x/b)\ .\end{aligned}$$

Multiplying by a, we have $g(t(x^*/b)+(1-t)(x/b)) \le \gamma t g(x^*/b)+(1-\gamma t)g(x/b)$ for all $x \in \mathbb{R}^n$. Since x/b can take on any value in $\mathbb{R}^n$, this means that g is γ-quasar-convex with respect to x^*/b. ■

I. Lower bounds

In this section, we construct lower bounds which demonstrate that the algorithms we presented in Section F obtain, up to logarithmic factors, the best possible worst-case iteration bounds for deterministic first-order methods. We use the ideas of Carmon et al. [13], who mechanized the process of constructing such lower bounds. Their idea is to construct a *zero-chain*, which is defined as a function f for which if $x_j = 0, \forall j \ge t$ then $\frac{\partial f(x)}{\partial x_{t+1}} = 0$. On these zero-chains, one can provide lower bounds for a particular class of methods known as *first-order zero-respecting algorithms*. First-order zero-respecting algorithms [13] are algorithms that only query the gradient at points $x^{(t)}$ with $x_i^{(t)} \ne 0$ if there exists some $j < t$ with $\nabla_i f(x^{(j)}) \ne 0$. Examples of zero-respecting first-order methods include gradient descent, accelerated gradient descent, and nonlinear conjugate gradient [19]. It is relatively easy to form lower bounds for zero-respecting algorithms applied to zero-chains, because one can prove that if the initial point is $x^{(0)} = \mathbf{0}$, then $x^{(T)}$ has at most T nonzeros [13, Observation 1]. The particular first-order zero-chain we use to derive our lower bounds is

$$\bar{f}_{T,\sigma}(x) \triangleq q(x) + \sigma \sum_{i=1}^{T} \Upsilon(x_i)$$

where

$$\begin{aligned}\Upsilon(\theta) &\triangleq 120 \int_1^\theta \frac{t^2(t-1)}{1+t^2}\,dt \\ q(x) &\triangleq \frac{1}{4}(x_1 - 1)^2 + \frac{1}{4}\sum_{i=1}^{T-1}(x_i - x_{i+1})^2.\end{aligned}$$

This function $\bar{f}_{T,\sigma}$ is very similar to the function $\bar{f}_{T,\mu,r}$ of Carmon, Duchi, Hinder, and Sidford [10]. However, the lower bound proof is different because the primary challenge is to show $\bar{f}_{T,\sigma}$ is quasar-convex, rather than showing that $\|\nabla \bar{f}_{T,\sigma}(x)\| \ge \epsilon$ for all x with $x_T \ne 0$. Our main lemma is as follows, and applies to an appropriately rescaled version of $\bar{f}_{T,\sigma}$ denoted by $\hat{f}$. Our main lemma shows that this function is in fact $\frac{1}{100T\sqrt{\sigma}}$-quasar-convex.

Lemma 9 *Let $\sigma \in (0, 10^{-4}], T \in \left[\sigma^{-1/2}, \infty\right) \cap \mathbb{Z}$. The function $\bar{f}_{T,\sigma}$ is $\frac{1}{100T\sqrt{\sigma}}$-quasar-convex and 3-smooth, with unique minimizer $x^* = \mathbf{1}$. Furthermore, if $x_t = 0$ for all $t = \lceil T/2 \rceil, \ldots, T$, then $\bar{f}_{T,\sigma}(x) - \bar{f}_{T,\sigma}(\mathbf{1}) \ge 2T\sigma$.*

The proof of Lemma 9 appears in Appendix J.1. The argument rests on showing that the quasar-convexity inequality $\frac{1}{100T\sqrt{\sigma}}(\bar{f}_{T,\sigma}(x) - \bar{f}_{T,\sigma}(\mathbf{1})) \leq \nabla \bar{f}_{T,\sigma}(x)^T(x - \mathbf{1})$ holds for all $x \in \mathbb{R}^T$. The nontrivial situation is when there exists some $j_1 < j_2$ such that $x_{j_1} \geq 0.9$, $x_{j_2} \leq 0.1$, and $0.1 \leq x_i \leq 0.9$ for $i \in \{j_1 + 1, \ldots, j_2 - 1\}$. In this situation, we use ideas closely related to the transition region arguments made in Lemma 3 of Carmon, Duchi, Hinder, and Sidford [10]. The intuition is as follows. If the gaps $x_{i+1} - x_i$ are large, then the convex function $q(x)$ dominates the function value and gradient of $\bar{f}_{T,\sigma}(x)$, allowing us to establish quasar-convexity. Conversely, if the $x_{i+1} - x_i$'s are small, then a large portion of the x_i's must lie in the quasar-convex region of Υ, and the corresponding $\Upsilon'(x_i)(x_i - 1)$ terms make $\nabla \bar{f}_{T,\sigma}(x)^\top(x - \mathbf{1})$ sufficiently positive.

Lemma 10 *Let $\epsilon \in (0, \infty)$, $\gamma \in (0, 10^{-2}]$, $T = \lceil 10^{-3}\gamma^{-1}L^{1/2}R\epsilon^{-1/2} \rceil$, and $\sigma = \frac{1}{10^4 T^2 \gamma^2}$, and assume $L^{1/2}R\epsilon^{-1/2} \geq 10^3$. Consider the function*

$$\hat{f}(x) \triangleq \tfrac{1}{3}LR^2T^{-1} \cdot \bar{f}_{T,\sigma}(xT^{1/2}R^{-1}). \tag{17}$$

This function is L-smooth and γ-quasar-convex, and its minimizer x^* is unique and has $\|x^*\| = R$. Furthermore, if $x_t = 0$ $\forall t \in \mathbb{Z} \cap [T/2, T]$, then $\hat{f}(x) - \inf_z \hat{f}(z) > \epsilon$.

The proof of Lemma 10 appears in Appendix J.1. Combining Lemma 10 with Observation 1 from Carmon et al. [13] yields a lower bound for first-order zero-respecting algorithms. Furthermore, we can use the argument from [13] to extend our lower bounds for first-order zero-respecting methods to the class of all deterministic first-order methods. This leads to Theorem 2, whose proof appears in Appendix J.2.

Theorem 2 *Let $\epsilon, R, L \in (0, \infty)$, $\gamma \in (0, 1]$, and assume $L^{1/2}R\epsilon^{-1/2} \geq 1$. Let $\mathcal{F}$ denote the set of L-smooth functions that are γ-quasar-convex with respect to some point with Euclidean norm $\leq R$. Then, given any deterministic first-order method, there exists a function $f \in \mathcal{F}$ such that the method requires at least $\Omega(\gamma^{-1}L^{1/2}R\epsilon^{-1/2})$ gradient evaluations to find an ϵ-optimal point of f.*

Theorem 2 demonstrates that the worst-case bound for our algorithm for quasar-convex minimization is tight within logarithmic factors.

Although the construction of the lower bounds in [10] is quite similar to our construction, there are some important differences between our lower bounds and those in [10]. First, the assumptions differ significantly; we assume quasar-convexity and Lipschitz continuity of the first derivative, while Carmon et al. [10] assume Lipschitz continuity of the first *three* derivatives. Next, we have only logarithmic gaps between our lower and upper bounds, whereas there is a gap of $\tilde{O}(\epsilon^{-1/15})$ between the lower bound of $\Omega(\epsilon^{-8/5})$ given by [10] and the best known upper bound of $O(\epsilon^{-5/3}\log(\epsilon^{-1}))$ given by [11] for the minimization of functions satisfying the assumptions in [10]. Another key difference is that the bounds in [10] and [11] apply to finding ϵ-stationary points, rather than ϵ-optimal points. Finally, we require $x_t = 0$ for all $t > T/2$ to guarantee $\hat{f}(x) - \inf_z \hat{f}(z) > \epsilon$, whereas Carmon et al. [10, 13] only need $x_T = 0$ to guarantee $\|\nabla \hat{f}(x)\| > \epsilon$.

J. Lower bound proofs

In this section, we use $\mathbf{0}$ to denote a vector with all entries equal to 0 and $\mathbf{1}$ to denote a vector with all entries equal to 1.

J.1. Proof of Lemma 10

Before we prove Lemma 10, we prove two useful results related to the properties of q and Υ. For convenience, these functions are restated below:

$$\Upsilon(\theta) \triangleq 120 \int_1^\theta \frac{t^2(t-1)}{1+t^2}\, dt$$
$$q(x) \triangleq \frac{1}{4}(x_1-1)^2 + \frac{1}{4}\sum_{i=1}^{T-1}(x_i - x_{i+1})^2.$$

Observation 5 *q is convex and 2-smooth with minimizer $x^* = \mathbf{1}$. Also, for any $1 \le j_1 < j_2 \le T$,*

$$q(x) = \frac{1}{2}\nabla q(x)^\top (x - x^*) \ge \max\left\{\frac{1}{4}(x_1-1)^2, \frac{(x_{j_1} - x_{j_2})^2}{4(j_2-j_1)}\right\}.$$

Proof Convexity and 2-smoothness of q follow from definitions. It is easy to see that q is always nonnegative and $q(\mathbf{1}) = 0$, so $\mathbf{1}$ minimizes q. In fact $\mathbf{1}$ is the unique minimizer, since q is strictly positive for all nonconstant vectors and all vectors with $x_1 \neq 1$.

Notice that as q is a convex quadratic, $q(x) = \frac{1}{2}(x-x^*)^\top \nabla^2 q(x)(x-x^*)$ where $\nabla^2 q(x)$ is a constant matrix. Therefore $\nabla q(x) = \nabla^2 q(x)(x-x^*)$. It follows that $q(x) = \frac{1}{2}\nabla q(x)^\top (x-x^*)$.

By definition $q(x) \ge \frac{1}{4}(x_1-1)^2$. Furthermore, $\frac{1}{j_2-j_1}\sum_{i=j_1}^{j_2}(x_i - x_{i+1})^2 \ge \left(\frac{1}{j_2-j_1}\sum_{i=j_1}^{j_2}(x_i - x_{i+1})\right)^2 = \frac{(x_{j_1}-x_{j_2})^2}{(j_2-j_1)^2}$, where the inequality uses that the expectation of the square of a random variable is greater than the square of its expectation. The result follows. ■

Properties of Υ that we will use are listed below.

Lemma 11 *The function Υ satisfies the following.*

1. $\Upsilon'(0) = \Upsilon'(1) = 0$.
2. *For all $\theta \le 1$, $\Upsilon'(\theta) \le 0$, and for all $\theta \ge 1$, $\Upsilon'(\theta) \ge 0$.*
3. *For all $\theta \in \mathbb{R}$ we have $\Upsilon(\theta) \ge \Upsilon(1) = 0$, and $\Upsilon(0) \le 10$.*
4. $\Upsilon'(\theta) < -1$ *for all* $\theta \in (-\infty, -0.1] \cup [0.1, 0.9]$.
5. Υ *is* 180*-smooth.*

6. *For all* $\theta \in \mathbb{R}$ *we have* $\Upsilon(\theta) \le \min\{30\theta^4 - 40\theta^3 + 10,\ 60(\theta-1)^2\}$, *and* $\Upsilon(0) \ge 5$.
7. *For all* $\theta \notin (-0.1, 0.1)$ *we have* $40(\theta-1)\Upsilon'(\theta) \ge \Upsilon(\theta)$.

Proof Properties 1-4 were proved in [10, Lemma 2].

Property 5. $|\Upsilon''(\theta)| = 120\left|\frac{\theta(\theta^3+3\theta-2)}{(1+\theta^2)^2}\right| \le 120 \cdot \frac{3}{2} = 180$ for all $\theta \in \mathbb{R}$. Thus, for any $\theta_1, \theta_2 \in \mathbb{R}$, $|\Upsilon'(\theta_1) - \Upsilon'(\theta_2)| \le \max\limits_{\theta \in [\theta_1, \theta_2]} |\Upsilon''(\theta)| \cdot |\theta_1 - \theta_2| \le 180|\theta_1 - \theta_2|$.

Property 6. We have $\Upsilon(0) = 120\int_0^1 \frac{t^2(1-t)}{1+t^2}\,dt \ge 120\int_0^1 \frac{t^2(1-t)}{2}\,dt = \frac{120}{2\cdot 12} = 5$. For all $\theta \in \mathbb{R}$ we have $\Upsilon(\theta) = 120\int_1^\theta \frac{t^2(t-1)}{1+t^2}\,dt \le 120\int_1^\theta t^2(t-1)\,dt = 120((\theta^4/4 + \theta^3/3) - (1/4 - 1/3)) = 30\theta^4 - 40\theta^3 + 10$. In addition, since $\frac{t^2}{1+t^2} \le 1$ for all t, we have for all $\theta \in \mathbb{R}$ that $\Upsilon(\theta) \le 120\int_1^\theta (t-1)\,dt = 120(\theta-1)^2/2$.

Property 7. If $\theta \in (\infty, -1.0] \cup [1.0, \infty)$ then $\frac{\theta^2}{1+\theta^2} \ge \frac{1}{2}$, so by property 6 we have

$$\begin{aligned}\Upsilon(\theta) + 40(1-\theta)\Upsilon'(\theta) &\le 60(\theta-1)^2 - 40 \cdot 120\frac{\theta^2(\theta-1)^2}{1+\theta^2} \\ &\le 60(\theta-1)^2 - 40 \cdot 60(\theta-1)^2 \\ &= -60 \cdot 39(\theta-1)^2 \le 0.\end{aligned}$$

Alternatively, if $\theta \in [-1.0, -0.1] \cup [0.1, 1.0]$ then $\frac{1}{1+\theta^2} \ge \frac{1}{2}$, so by property 6 we have

$$\begin{aligned}\Upsilon(\theta) + 40(1-\theta)\Upsilon'(\theta) &\le 10 + 30\theta^4 - 40\theta^3 - 40 \cdot 120\frac{\theta^2(\theta-1)^2}{1+\theta^2} \\ &\le 10\left(1 + \theta^2\left(3\theta^2 - 4\theta - 240(\theta-1)^2\right)\right) \\ &= 10\left(1 - 237\theta^4 + 476\theta^3 - 240\theta^2\right) = 10P(\theta)\,,\end{aligned}$$

where we define $P(\theta) \triangleq 1 - 237\theta^4 + 476\theta^3 - 240\theta^2$. $P'(\theta) = -12\theta(40 - 119\theta + 79\theta^2)$ has exactly three roots: at $\theta = 0, \theta = 1$ and $\theta = 40/79$. Furthermore, at $\theta = 1$, $\theta = 40/79$ and $\theta = 0.1$ we have $P(\theta) \le 0$, which implies $P(\theta) \le 0$ for $\theta \in [0.1, 1]$. We conclude that $\Upsilon(\theta) + 40(1-\theta)\Upsilon'(\theta) \le 0$ for $\theta \in [0.1, 1]$. In addition, $P(\theta)$ is negative while $P'(\theta)$ is positive for $\theta = -0.1$, which means that $P(\theta)$ and thus $\Upsilon(\theta) + 40(1-\theta)\Upsilon'(\theta)$ are also negative on $[-1.0, -0.1]$. ■

Now, we prove Lemma 9, which we later use to prove Lemma 10 (a "scaled version").

Lemma 9 *Let* $\sigma \in (0, 10^{-4}], T \in \left[\sigma^{-1/2}, \infty\right) \cap \mathbb{Z}$. *The function* $\bar{f}_{T,\sigma}$ *is* $\frac{1}{100T\sqrt{\sigma}}$*-quasar-convex and* 3*-smooth, with unique minimizer* $x^* = \mathbf{1}$. *Furthermore, if* $x_t = 0$ *for all* $t = \lceil T/2 \rceil, \ldots, T$, *then* $\bar{f}_{T,\sigma}(x) - \bar{f}_{T,\sigma}(\mathbf{1}) \ge 2T\sigma$.

Proof Since $\sigma \in (0, 10^{-4}]$, Υ is 180-smooth, and q is 2-smooth, we deduce $\bar{f}_{T,\sigma}$ is 3-smooth. By Observation 5 and Lemma 11.3 we deduce $\bar{f}_{T,\sigma}(\mathbf{1}) = 0 < \bar{f}_{T,\sigma}(x)$ for all $x \ne \mathbf{1}$. Therefore, $x^* = \mathbf{1}$ is the unique minimizer of $\bar{f}_{T,\sigma}$.

Now, we will show $\bar{f}_{T,\sigma}$ is $\frac{1}{100T\sqrt{\sigma}}$-quasar-convex, i.e. that $\nabla \bar{f}_{T,\sigma}(x)^\top (x - \mathbf{1}) \geq \frac{\bar{f}_{T,\sigma}(x) - \bar{f}_{T,\sigma}(\mathbf{1})}{100T\sqrt{\sigma}}$ for all $x \in \mathbb{R}^T$. Define

$$\mathcal{A} \triangleq \{i : x_i \in (-\infty, -0.1] \cup (0.9, \infty)\};\ \mathcal{B} \triangleq \{i : x_i \in (-0.1, 0.1)\};\ \mathcal{C} \triangleq \{i : x_i \in [0.1, 0.9]\}.$$

First, we derive two useful inequalities. By Observation 5 and the fact that $\Upsilon'(x_i) \leq 0$ for $i \in \mathcal{B}$,

$$\begin{aligned}\nabla \bar{f}_{T,\sigma}(x)^\top (x - \mathbf{1}) &= \nabla q(x)^\top (x - \mathbf{1}) + \sigma \sum_{i \in \mathcal{A} \cup \mathcal{B} \cup \mathcal{C}} (x_i - 1)\Upsilon'(x_i) \\ &\geq 2q(x) + \sigma \sum_{i \in \mathcal{A} \cup \mathcal{C}} (x_i - 1)\Upsilon'(x_i)\,. \end{aligned} \tag{18}$$

By Lemma 11.2 and 11.6 we deduce $\sum_{i \in \mathcal{B} \cup \mathcal{C}} \Upsilon(x_i) \leq |\mathcal{B} \cup \mathcal{C}| \Upsilon(-0.1) \leq 11T$, so $\bar{f}_{T,\sigma}(x) \leq q(x) + 11T\sigma + \sigma \sum_{i \in \mathcal{A}} \Upsilon(x_i)$, and therefore using $T \geq \sigma^{-1/2}$ and nonnegativity of Υ and q,

$$\begin{aligned}\frac{\bar{f}_{T,\sigma}(x) - \bar{f}_{T,\sigma}(\mathbf{1})}{100T\sqrt{\sigma}} = \frac{\bar{f}_{T,\sigma}(x)}{100T\sqrt{\sigma}} &\leq \frac{11T\sigma}{100T\sqrt{\sigma}} + \frac{\sigma}{100T\sqrt{\sigma}} \sum_{i \in \mathcal{A}} \Upsilon(x_i) + \frac{1}{100T\sqrt{\sigma}} q(x) \\ &\leq \frac{11}{100}\sigma^{1/2} + \frac{\sigma}{100} \sum_{i \in \mathcal{A}} \Upsilon(x_i) + \frac{1}{100} q(x) \\ &\leq \frac{11}{100}\sigma^{1/2} + \frac{\sigma}{40} \sum_{i \in \mathcal{A}} \Upsilon(x_i) + q(x) \end{aligned} \tag{19}$$

We now consider three possible cases for the values of x.

1. Consider the case that $x_1 \notin [0.9, 1.1]$. We have

$$\begin{aligned}\nabla \bar{f}_{T,\sigma}(x)^\top (x - \mathbf{1}) &\geq 2q(x) + \frac{\sigma}{40} \sum_{i \in \mathcal{A} \cup \mathcal{C}} \Upsilon(x_i) \\ &\geq \frac{0.1^2}{4} + q(x) + \frac{\sigma}{40} \sum_{i \in \mathcal{A} \cup \mathcal{C}} \Upsilon(x_i) \\ &= \frac{1}{\sqrt{10^4 \sigma}} \cdot \frac{\sqrt{\sigma}}{4} + \frac{\sigma}{40} \sum_{i \in \mathcal{A} \cup \mathcal{C}} \Upsilon(x_i) + q(x) \\ &\geq \frac{\sqrt{\sigma}}{4} + \frac{\sigma}{40} \sum_{i \in \mathcal{A} \cup \mathcal{C}} \Upsilon(x_i) + q(x) \\ &\geq \frac{\bar{f}_{T,\sigma}(x) - \bar{f}_{T,\sigma}(\mathbf{1})}{100T\sqrt{\sigma}} \end{aligned}$$

where the first inequality uses (18) and Lemma 11.7, the second inequality uses Observation 5 and $x_1 \notin [0.9, 1.1]$, the penultimate inequality uses $\sigma \in (0, 10^{-4}]$, and the final inequality uses (19) and

nonnegativity of Υ. 2. Consider the case that $\mathcal{B} = \emptyset$. By Lemma 11.7 and convexity of $q(x)$,

$$\begin{aligned}\nabla \bar{f}_{T,\sigma}(x)^\top (x - \mathbf{1}) &= \nabla q(x)^\top (x - \mathbf{1}) + \sigma \sum_{i \in \mathcal{A} \cup \mathcal{C}} (x_i - 1)\Upsilon'(x_i) \\ &\geq q(x) - q(\mathbf{1}) + \frac{\sigma}{40} \sum_{i \in \mathcal{A} \cup \mathcal{C}} \Upsilon(x_i) \\ &= \frac{1}{40}\left(q(x) + \sigma \sum_{i=1}^{T} \Upsilon(x_i)\right) - \bar{f}_{T,\sigma}(\mathbf{1}) + \frac{39}{40} q(x) \\ &\geq \frac{\bar{f}_{T,\sigma}(x) - \bar{f}_{T,\sigma}(\mathbf{1})}{40} \geq \frac{\bar{f}_{T,\sigma}(x) - \bar{f}_{T,\sigma}(\mathbf{1})}{100T\sqrt{\sigma}}.\end{aligned}$$

3. Suppose cases 1-2 do not hold, i.e. $x_1 \in [0.9, 1.1]$ and $\mathcal{B} \neq \emptyset$. Then there exist $m \geq 1$ and $1 \leq j \leq T - m$ such that $x_j \geq 0.9$, $x_{j+m} \leq 0.1$, and $x_i \in \mathcal{C}$ $\forall i \in \{j+1, \ldots, j+m-1\}$. Then,

$$\begin{aligned}\nabla \bar{f}_{T,\sigma}(x)^\top (x - \mathbf{1}) &\geq q(x) + \sigma \sum_{i \in \mathcal{A} \cup \mathcal{C}} (x_i - 1)\Upsilon'(x_i) + q(x) \\ &\geq \frac{0.8^2}{4m} + \sigma \sum_{i \in \mathcal{C}} (x_i - 1)\Upsilon'(x_i) + \sigma \sum_{i \in \mathcal{A}} (x_i - 1)\Upsilon'(x_i) + q(x) \\ &\geq \frac{0.8^2}{4m} + 0.1\sigma(m-2) + \frac{\sigma}{40} \sum_{i \in \mathcal{A}} \Upsilon(x_i) + q(x) \\ &\geq \frac{0.16}{\sqrt{1.6}} \sigma^{1/2} + \frac{\sigma}{40} \sum_{i \in \mathcal{A}} \Upsilon(x_i) + q(x) \geq \frac{\bar{f}_{T,\sigma}(x) - \bar{f}_{T,\sigma}(\mathbf{1})}{100T\sqrt{\sigma}}\end{aligned}$$

where the the first inequality holds by (18), the second inequality uses Observation 5, the third inequality uses Lemma 11.4 and 11.7, the fourth inequality uses that $m = \sqrt{1.6}\sigma^{-0.5} \geq 2$ minimizes the previous expression, and the final inequality uses (19) [and the fact that $0.16/\sqrt{1.6} > 0.11$].

Finally, suppose $x_t = 0$ for all $t = \lceil T/2 \rceil, \ldots, T$. Then we have $\bar{f}_{T,\sigma}(x) - \bar{f}_{T,\sigma}(\mathbf{1}) = \bar{f}_{T,\sigma}(x) \geq \sigma \lceil T/2 \rceil \Upsilon(0) \geq 2T\sigma$, where the first inequality uses that $\Upsilon \geq 0$ and $q \geq 0$, and the last inequality uses that $T \geq 1$ and $\Upsilon(0) \geq 5$. ■

With Lemma 9 in hand, we are able to establish Lemma 10 which is a scaled version of Lemma 9.

Lemma 10 *Let $\epsilon \in (0, \infty)$, $\gamma \in (0, 10^{-2}]$, $T = \lceil 10^{-3}\gamma^{-1}L^{1/2}R\epsilon^{-1/2} \rceil$, and $\sigma = \frac{1}{10^4 T^2 \gamma^2}$, and assume $L^{1/2}R\epsilon^{-1/2} \geq 10^3$. Consider the function*

$$\hat{f}(x) \triangleq \tfrac{1}{3} L R^2 T^{-1} \cdot \bar{f}_{T,\sigma}(xT^{1/2}R^{-1}). \tag{17}$$

This function is L-smooth and γ-quasar-convex, and its minimizer x^* is unique and has $\|x^*\| = R$. Furthermore, if $x_t = 0$ $\forall t \in \mathbb{Z} \cap [T/2, T]$, then $\hat{f}(x) - \inf_z \hat{f}(z) > \epsilon$.

Proof We have $\sigma^{-1/2} = 10^2 T\gamma \le T$ and $\sigma = \frac{1}{10^4 T^2 \gamma^2} \le \frac{10^2}{(L^{1/2} R \epsilon^{-1/2})^2} \le 10^{-4}$, so $\bar{f}_{T,\sigma}$ satisfies the conditions of Lemma 9. Let us verify the properties of $\hat{f}$. The optimum of $\bar{f}_{T,\sigma}$ is $\mathbf{1}$, but after this rescaling it becomes $x^* = \frac{R}{\sqrt{T}}\mathbf{1}$, for which $\|x^*\| = R$. For all $x, y \in \mathbb{R}^T$, by 3-smoothness of $\bar{f}_{T,\sigma}$,

$$\begin{aligned}\left\|\nabla \hat{f}(x) - \nabla \hat{f}(y)\right\| &= \tfrac{1}{3}(LR^2T^{-1}) \cdot (T^{1/2}R^{-1}) \left\|\nabla \bar{f}_{T,\sigma}(xT^{1/2}R^{-1}) - \nabla \bar{f}_{T,\sigma}(yT^{1/2}R^{-1})\right\| \\ &\le (LR^2T^{-1}) \cdot (T^{1/2}R^{-1})^2 \|x - y\| \\ &= L\|x - y\| \ .\end{aligned}$$

Therefore $\hat{f}$ is L-smooth. By the definition of σ we have $\frac{1}{100T\sqrt{\sigma}} = \gamma$, so $\bar{f}_{T,\sigma}$ is γ-quasar-convex. As quasar-convexity is invariant to scaling (Observation 4), we deduce that $\hat{f}$ is γ-quasar-convex as well. Finally, given $x_t^{(k)} = 0$ for $t = \lceil T/2 \rceil, \ldots, T$, we have

$$\hat{f}(x^{(k)}) - \inf_z \hat{f}(z) \ge 2T\sigma \cdot \frac{LR^2}{3T} = \tfrac{2}{3}LR^2\sigma = \tfrac{2}{3}(10^{-2}\gamma^{-1}L^{1/2}RT^{-1})^2 \ge \tfrac{50}{3}\epsilon,$$

where the first transition uses Lemma 9, the third transition uses $\sigma = \frac{1}{10^4 T^2 \gamma^2}$, and the last uses that $T = \left\lceil 10^{-3}\gamma^{-1}L^{1/2}R\epsilon^{-1/2}\right\rceil \le 2 \cdot 10^{-3}\gamma^{-1}L^{1/2}R\epsilon^{-1/2}$, as $10^{-3}\gamma^{-1}L^{1/2}R\epsilon^{-1/2} \ge 1$. ■

J.2. Proof of Theorem 2

Before proving Theorem 2 we recap definitions that were originally provided in Carmon, Duchi, Hinder, and Sidford [13].

Definition 3 *A function f is a first-order zero-chain if for every $x \in \mathbb{R}^n$,*

$$x_i = 0 \quad \forall i \ge t \quad \Rightarrow \quad \nabla_i f(x) = 0 \quad \forall i > t.$$

Definition 4 *An algorithm is a first-order zero-respecting algorithm (FOZRA) if, for all $i \in \{1, \ldots, n\}$, its iterates $x^{(0)}, x^{(1)}, \ldots \in \mathbb{R}^n$ satisfy*

$$\nabla_i f(x^{(k)}) = 0 \quad \forall k \le t \quad \Rightarrow \quad x_i^{(t+1)} = 0$$

Definition 5 *An algorithm $\mathcal{A}$ is a first-order deterministic algorithm (FODA) if there exists a sequence of functions $\mathcal{A}_k$ such the algorithm's iterates satisfy*

$$x^{(k+1)} = \mathcal{A}_k(x^{(0)}, \ldots, x^{(k)}, \nabla f(x^{(0)}), \ldots, \nabla f(x^{(k)}))$$

for all $k \in \mathbb{N}$, input functions f, and starting points $x^{(0)}$.

Observation 6 *Consider $\epsilon > 0$, a function class $\mathcal{F}$, and $K \in \mathbb{N}$. If $f : \mathbb{R}^n \to \mathbb{R}$ satisfies*

1. *f is a first-order zero-chain,*
2. *f belongs to the function class $\mathcal{F}$, i.e. $f \in \mathcal{F}$, and*
3. *$f(x) - \inf_z f(z) \geq \epsilon$ for every x such that $x_t = 0$ for all $t \in \{K, K+1, \ldots, n\}$;*

then it takes at least K iterations for any FOZRA to find an ϵ-optimal solution of f.

Proof Cosmetic modification of the proof of Observation 2 in [13]. ■

Theorem 2 *Let $\epsilon, R, L \in (0, \infty)$, $\gamma \in (0, 1]$, and assume $L^{1/2} R \epsilon^{-1/2} \geq 1$. Let $\mathcal{F}$ denote the set of L-smooth functions that are γ-quasar-convex with respect to some point with Euclidean norm $\leq R$. Then, given any deterministic first-order method, there exists a function $f \in \mathcal{F}$ such that the method requires at least $\Omega(\gamma^{-1} L^{1/2} R \epsilon^{-1/2})$ gradient evaluations to find an ϵ-optimal point of f.*

Proof Applying Lemma 10 and Observation 6 implies this result for any first-order zero-respecting method. Applying Proposition 1 from [13], which states that lower bounds for first-order zero-respecting methods also apply to deterministic first-order methods, gives the result. ■

K. Related function classes

In this section, we provide a brief taxonomy of related conditions (relaxations of convexity or strong convexity), and describe how they relate to quasar-convexity. For simplicity, here we assume f is L-smooth with domain $\mathcal{X} = \mathbb{R}^n$. We denote the minimum of f by f^* and the set of minimizers of f by $\mathcal{X}^*$; when $\mathcal{X}^*$ consists of a single point, we denote the point by x^*.

First, we review the definitions of quasar-convexity, star-convexity, and convexity. Recall that (strong) quasar-convexity is a generalization of (strong) star-convexity, which itself generalizes (strong) convexity.

- *(Strong) quasar-convexity* (with parameters $\gamma \in (0, 1]$, $\mu \geq 0$): for some $x^* \in \mathcal{X}^*$, $f(x^*) \geq f(x) + \frac{1}{\gamma} \nabla f(x)^\top (x^* - x) + \frac{\mu}{2} \|x^* - x\|^2$ for all $x \in \mathcal{X}$.
  - When $\mu = 0$, this is merely referred to as *quasar-convexity*, which is also known as *weak quasi-convexity* [26].
  - When $\mu > 0$, f has exactly one minimizer x^*.
- *(Strong) star-convexity* (with parameter $\mu \geq 0$): for some $x^* \in \mathcal{X}^*$, $f(x^*) \geq f(x) + \nabla f(x)^\top (x^* - x) + \frac{\mu}{2} \|x^* - x\|^2$ for all $x \in \mathcal{X}$.
  - When $\mu = 0$, this is merely referred to as *star-convexity*.
  - When $\mu > 0$, this is also known as *quasi-strong convexity* [37].

  - When $\mu = 0$, f may not have a unique minimizer; some authors require the condition to hold for *all* $x^* \in \mathcal{X}^*$ [42], while others only require it for *some* $x^* \in \mathcal{X}^*$ [32]; we use the latter definition.
  - When $\mu > 0$, f has exactly one minimizer x^*.
- *(Strong) convexity* (with parameter $\mu \geq 0$): $f(y) \geq f(x) + \nabla f(x)^\top (y - x) + \frac{\mu}{2} \|y - x\|^2$ for all $x, y \in \mathcal{X}$.
  - When $\mu = 0$, this is merely referred to as *convexity*.

Next, we enumerate some other generalizations of strong convexity from the literature, and state whether they generalize quasar-convexity, are generalized by quasar-convexity, or neither.

- *Weak convexity* [50] (with parameter $\mu > 0$): $f(y) \geq f(x) + \nabla f(x)^\top (y - x) - \frac{\mu}{2} \|y - x\|^2$ for all $x, y \in \mathcal{X}$.
  - Neither implies nor is implied by quasar-convexity.
- *Quadratic growth condition* (with parameter $\mu > 0$) [3]: $f(x) \geq f(x^*) + \frac{\mu}{2} \|x^* - x\|^2$ for all $x \in \mathcal{X}$.
  - Neither implies nor is implied by quasar-convexity.
- *Restricted secant condition* (with parameter $\mu > 0$) [53]: $0 \geq \nabla f(x)^\top (x^* - x) + \frac{\mu}{2} \|x^* - x\|^2$ for all $x \in \mathcal{X}$.
  - Implied by $(\gamma, \frac{\mu}{\gamma})$-strong quasar-convexity (for any choice of $\gamma \in (0, 1]$).
- *One-point strong convexity* (with parameter $\mu > 0$) [34]: for some $y \in \mathcal{X}$, $0 \geq \nabla f(x)^\top (y - x) + \frac{\mu}{2} \|y - x\|^2$ for all $x \in \mathcal{X}$.
  - This is a generalization of the restricted secant property (which is one-point strong convexity in the special case $y = x^*$), and is therefore likewise implied by strong quasar-convexity.
- *Variational coherence* [56]: $0 \geq \nabla f(x)^\top (x^* - x)$ for all $x \in \mathcal{X}$, $x^* \in \mathcal{X}^*$, with equality iff $x \in \mathcal{X}^*$.
  - Implied by strong quasar-convexity (for any $\mu > 0$ and $\gamma \in (0, 1]$). The closely related weaker condition "for some $x^* \in \mathcal{X}^*$, $0 \geq \nabla f(x)^\top (x^* - x)$ for all $x \in \mathcal{X}$, with equality iff $x \in \mathcal{X}^*$" is implied by quasar-convexity (for any $\mu \geq 0, \gamma \in (0, 1]$).
- *Polyak-Łojasiewicz condition* [45] (with parameter $\mu > 0$): $\frac{1}{2} \|\nabla f(x)\|^2 \geq \mu (f(x) - f_*)$ for all $x \in \mathcal{X}$.
  - This is implied by the restricted secant property [30], and therefore by strong quasar-convexity.
- *Quasiconvexity* [4]: $f(\lambda x + (1 - \lambda) y) \leq \max\{f(x), f(y)\}$ for all $x, y \in \mathcal{X}$ and $\lambda \in [0, 1]$.

  - Neither implies nor is implied by quasar-convexity.

- *Pseudoconvexity* [36]: $f(y) \geq f(x)$ for all $x, y \in \mathcal{X}$ such that $\nabla f(x) \cdot (y - x) \geq 0$.
  - Neither implies nor is implied by quasar-convexity.
- *Invexity* [15]: $x \in \mathcal{X}^*$ for all $x \in \mathcal{X}$ such that $\nabla f(x) = \mathbf{0}$.
  - Implied by quasar-convexity (for any $\mu \geq 0, \gamma \in (0, 1]$).