Simple steps are all you need: Frank-Wolfe and generalized self-concordant functions

Alejandro Carderera *

Mathieu Besançon †

Sebastian Pokutta ‡

Abstract

Generalized self-concordance is a key property present in the objective function of many important learning problems. We establish the convergence rate of a simple Frank-Wolfe variant that uses the open-loop step size strategy $\gamma_t = 2/(t+2)$, obtaining a $\mathcal{O}(1/t)$ convergence rate for this class of functions in terms of primal gap and Frank-Wolfe gap, where t is the iteration count. This avoids the use of second-order information or the need to estimate local smoothness parameters of previous work. We also show improved convergence rates for various common cases, e.g., when the feasible region under consideration is uniformly convex or polyhedral.

1 Introduction

Constrained convex optimization is the cornerstone of many machine learning problems. We consider such problems, formulated as:

$$\min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}), \tag{1.1}$$

where $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is a generalized self-concordant function and $\mathcal{X} \subseteq \mathbb{R}^n$ is a compact convex set. When computing projections onto the feasible regions as required in, e.g., projected gradient descent, is prohibitive, Frank-Wolfe (FW) (Frank & Wolfe, 1956) algorithms (a.k.a. Conditional Gradients (CG) (Levitin & Polyak, 1966)) are the algorithms of choice, relying on Linear Minimization Oracles (LMO) at each iteration to solve Problem (1.1). The analysis of their convergence often relies on the assumption that the gradient is Lipschitz-continuous. This assumption does not necessarily hold for generalized self-concordant functions, an important class of functions for which the growth can be unbounded.

1.1 Related work

In the classical analysis of Newton's method, when the Hessian of f is assumed to be Lipschitz continuous and the function is strongly convex, one arrives at a convergence rate for the algorithm that depends on the Euclidean structure of $\mathbb{R}^n$, despite the fact that the algorithm is affine-invariant. This motivated the introduction of self-concordant functions in Nesterov & Nemirovskii (1994), functions for which the third derivative is bounded by the second-order derivative, with which one can obtain an affine-invariant convergence rate for the aforementioned algorithm. More importantly, many of the barrier functions used in interior-point methods are self-concordant, which extended the use of polynomial-time interior-point methods to many settings of interest.

*Georgia Institute of Technology. E-mail: alejandro.carderera@gatech.edu

†Zuse Institute Berlin. E-mail: besancon@zib.de

‡Zuse Institute Berlin and Technische Universität Berlin. E-mail: pokutta@math.tu-berlin.de

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

Self-concordant functions have received strong interest in recent years due to the attractive properties that they allow to prove for many statistical estimation settings (Marteau-Ferey et al., 2019; Ostrovskii & Bach, 2021). The original definition of self-concordance has been expanded and generalized since its inception, as many objective functions of interest have self-concordant-like properties without satisfying the strict definition of self-concordance. For example, the logistic loss function used in logistic regression is not strictly self-concordant, but it fits into a class of pseudo-self-concordant functions, which allows one to obtain similar properties and bounds as those obtained for self-concordant functions (Bach, 2010). This was also the case in Ostrovskii & Bach (2021) and Tran-Dinh et al. (2015), in which more general properties of these pseudo-self-concordant functions were established. This was fully formalized in Sun & Tran-Dinh (2019), in which the concept of *generalized self-concordant* functions was introduced, along with key bounds, properties, and variants of Newton methods for the unconstrained setting which make use of this property.

Most algorithms that aim to solve Problem (1.1) assume access to second-order information, as this often allows the algorithms to make monotonous progress, remain inside the domain of f, and often, converge quadratically when close enough to the optimum. Recently, several lines of work have focused on using Frank-Wolfe algorithm variants to solve these types of problems in the projection-free setting, for example constructing second-order approximations to a self-concordant f using first and second-order information, and minimizing these approximations over $\mathcal{X}$ using the Frank-Wolfe algorithm (Liu et al., 2020). Other approaches, such as the ones presented in Dvurechensky et al. (2020a) (later extended in Dvurechensky et al. (2020b)), apply the Frank-Wolfe algorithm to a generalized self-concordant f, using first and second-order information about the function to guarantee that the step sizes are so that the iterates do not leave the domain of f, and monotonous progress is made. An additional Frank-Wolfe variant in that work, in the spirit of Garber & Hazan (2016), utilizes first and second order information about f, along with a Local Linear Optimization Oracle for $\mathcal{X}$, to obtain a linear convergence rate in primal gap over polytopes given in inequality description. The authors in Dvurechensky et al. (2020b) also present an additional Frank-Wolfe variant which does not use second-order information, and uses the backtracking line search of Pedregosa et al. (2020) to estimate local smoothness parameters at a given iterate. Other specialized Frank-Wolfe algorithms have been developed for specific problems involving generalized self-concordant functions, such as the Frank-Wolfe variant developed for marginal inference with concave maximization (Krishnan et al., 2015), the variant developed in Zhao & Freund (2020) for θ-homogeneous barrier functions, or the application for phase retrieval in Odor et al. (2016), where the Frank-Wolfe algorithm is shown to converge on a self-concordant non-Lipschitz smooth objective.

1.2 Contribution

The contributions of this paper are detailed below and summarized in Table 1.

Simple FW for generalized self-concordant functions. We show that a small variation of the original Frank-Wolfe algorithm (Frank & Wolfe, 1956) with an open-loop step size of the form $\gamma_t = 2/(t+2)$, where t is the iteration count is all that is needed to achieve a convergence rate of $\mathcal{O}(1/t)$ in primal gap; this also answers an open question posed in Dvurechensky et al. (2020b). Our variation ensures monotonous progress while employing an open-loop strategy which, together with the iterates being convex combinations, ensures that we do not leave the domain of f. In contrast to other methods that depend on either a line search or second-order information, our variant uses only a linear minimization oracle, zeroth-order and first-order information and a domain oracle for $f(\mathbf{x})$. The assumption of the latter oracle is very mild and was also implicitly assumed in several of the algorithms presented in Dvurechensky et al. (2020b). As such, our iterations are much cheaper than those in previous work, while essentially achieving the same convergence rates for Problem (1.1).

Moreover, our variant relying on the open-loop step size $\gamma_t = 2/(t+2)$ allows us to establish a $\mathcal{O}(1/t)$ convergence rate for the Frank-Wolfe gap, is agnostic, i.e., does not need to estimate local smoothness parameters, and is parameter-free, leading to convergence rates and oracle complexities that are independent of any tuning parameters.

Algorithm	Convergence Primal gap	FW gap	Reference	1^{st}-order / LS free?	Requirements
`FW-GSC`	$\mathcal{O}(1/\varepsilon)$		[1, Alg.2]	✗ / ✓	SOO
`LBTFW-GSC`	$\mathcal{O}(1/\varepsilon)$		[1, Alg.3]	✓ / ✗	ZOO, DO
`MBTFW-GSC`	$\mathcal{O}(1/\varepsilon)$		[1, Alg.5]	✗ / ✓	ZOO, SOO, DO
`FW-LLOO`	$\mathcal{O}(\log 1/\varepsilon)$		[1, Alg.7]	✗ / ✓	polyh. $\mathcal{X}$, LLOO, SOO
`ASFW-GSC`	$\mathcal{O}(\log 1/\varepsilon)$		[1, Alg.8]	✗ / ✓	polyh. $\mathcal{X}$, SOO
`M-FW`	$\mathcal{O}(1/\varepsilon)$	$\mathcal{O}(1/\varepsilon)$	**This work**	✓ / ✓	ZOO,DO
`B-AFW`	$\mathcal{O}(\log 1/\varepsilon)$	$\mathcal{O}(\log 1/\varepsilon)$	**This work**	✓ / ✗	polyh. $\mathcal{X}$, ZOO,DO

Table 1: Number of iterations needed to achieve an ε-optimal solution for Problem 1.1. We denote Dvurechensky et al. (2020b) by [1], line search by LS, local linear optimization oracle by LLOO, and the assumption that $\mathcal{X}$ is polyhedral by polyh. $\mathcal{X}$. The oracles listed under **Requirements** are the additional oracles needed, other than the FOO and the LMO.

Faster rates in common special cases. We also obtain improved convergence rates when the optimum is contained in the interior of $\mathcal{X} \cap \operatorname{dom}(f)$, or when the set $\mathcal{X}$ is uniformly or strongly convex, using the backtracking line search of Pedregosa et al. (2020). We also show that the Away-Step Frank-Wolfe algorithm (Lacoste-Julien & Jaggi, 2015; Wolfe, 1970) can use the aforementioned line search to achieve linear rates over polytopes. For clarity we want to stress that any linear rate over polytopes has to depend also on the ambient dimension of the polytope; this applies to our linear rates and those in Table 1 established elsewhere (see Diakonikolas et al. (2020)). In contrast the $\mathcal{O}(1/\varepsilon)$ rates are dimension-independent.

Numerical experiments. We provide numerical experiments that showcase the performance of the algorithms on generalized self-concordant objectives to complement the theoretical results. In particular, they highlight that the simple step size strategy we propose is competitive with and sometimes outperforms other variants on many instances.

After publication of our initial draft, in a revision of their original work, Dvurechensky et al. (2020b) added an analysis of the Away-Step Frank-Wolfe algorithm which is complementary to ours (considering a slightly different setup and regimes) and was conducted independently; we have updated the tables to include these additional results.

1.3 Preliminaries and Notation

We denote the *domain* of f as $\operatorname{dom}(f) \stackrel{\text{def}}{=} \{\mathbf{x} \in \mathbb{R}^n, f(\mathbf{x}) < +\infty\}$ and the (potentially non-unique) minimizer of Problem (1.1) by $\mathbf{x}^*$. Moreover, we denote the *primal gap* and the *Frank-Wolfe gap* at $\mathbf{x} \in \mathcal{X} \cap \operatorname{dom}(f)$ as $h(\mathbf{x}) \stackrel{\text{def}}{=} f(\mathbf{x}) - f(\mathbf{x}^*)$ and $g(\mathbf{x}) \stackrel{\text{def}}{=} \max_{\mathbf{v} \in \mathcal{X}} \langle \nabla f(\mathbf{x}), \mathbf{x} - \mathbf{v} \rangle$, respectively. We use $\|\cdot\|$, $\|\cdot\|_H$, and $\langle \cdot, \cdot \rangle$ to denote the *Euclidean norm*, the *matrix norm* induced by a symmetric positive definite matrix $H \in \mathbb{R}^{n \times n}$, and the *Euclidean inner product*, respectively. We denote the *diameter* of $\mathcal{X}$ as $D \stackrel{\text{def}}{=} \max_{\mathbf{x}, \mathbf{y} \in \mathcal{X}} \|\mathbf{x} - \mathbf{y}\|$. Given a non-empty set $\mathcal{X} \subset \mathbb{R}^n$ we refer to its *boundary* as $\operatorname{Bd}(\mathcal{X})$ and to its *interior* as $\operatorname{Int}(\mathcal{X})$. We use Δ_n to denote the *probability simplex* of dimension n. Given a compact convex set $\mathcal{C} \subseteq \operatorname{dom}(f)$ we denote $L_f^{\mathcal{C}} = \max_{\mathbf{u} \in \mathcal{C}, \mathbf{d} \in \mathbb{R}^n} \|\mathbf{d}\|^2_{\nabla^2 f(\mathbf{u})} / \|\mathbf{d}\|_2^2$ and $\mu_f^{\mathcal{C}} = \min_{\mathbf{u} \in \mathcal{C}, \mathbf{d} \in \mathbb{R}^n} \|\mathbf{d}\|^2_{\nabla^2 f(\mathbf{u})} / \|\mathbf{d}\|_2^2$. We assume access to:

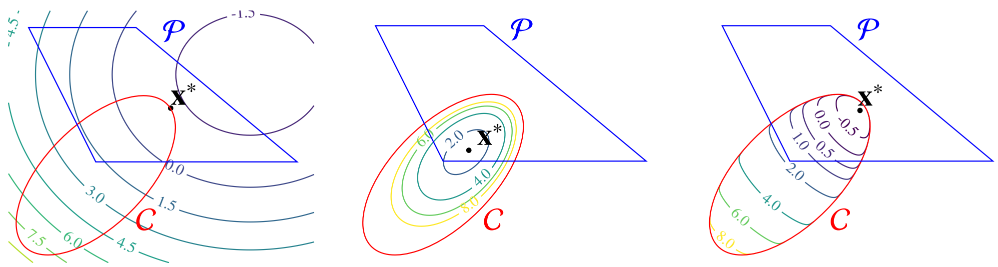
1. **Domain Oracle (DO):** Given $\mathbf{x} \in \mathcal{X}$, return `true` if $\mathbf{x} \in \operatorname{dom}(f)$, `false` otherwise.
2. **Zeroth-Order Oracle (ZOO)**: Given $\mathbf{x} \in \operatorname{dom}(f)$, return $f(\mathbf{x})$.
3. **First-Order Oracle (FOO)**: Given $\mathbf{x} \in \operatorname{dom}(f)$, return $\nabla f(\mathbf{x})$.
4. **Linear Minimization Oracle (LMO)**: Given $\mathbf{d} \in \mathbb{R}^n$, return $\operatorname{argmin}_{\mathbf{x} \in \mathcal{X}} \langle \mathbf{x}, \mathbf{d} \rangle$.

The FOO and LMO oracles are standard in the FW literature. The ZOO oracle is often implicitly assumed to be included with the FOO oracle; we make this explicit here for clarity. Finally, the DO oracle is motivated by the properties of generalized self-concordant functions. It is reasonable to assume the availability of the DO oracle: following the definition of the function codomain, one could simply evaluate f at $\mathbf{x}$ and assert $f(\mathbf{x}) < +\infty$, thereby combining the DO and ZOO oracles into one oracle. However, in many cases testing the membership of $\mathbf{x} \in \operatorname{dom}(f)$ is computationally less demanding than the function evaluation.

Remark 1.1. Requiring access to a zeroth-order and domain oracle are mild assumptions, that were also implicitly assumed in one of the three FW-variants presented in Dvurechensky et al. (2020b) when computing the step size according to the strategy from Pedregosa et al. (2020); see Line 3 in Algorithm 6 in the Appendix. The remaining two variants ensure that $\mathbf{x} \in \mathrm{dom}(f)$ by using second-order information about f, which we explicitly do not rely on.

The following example motivates the use of Frank-Wolfe algorithms in the context of generalized self-concordant functions. We present more examples in the computational results.

Example 1.2 (Intersection of a convex set with a polytope)**.** Consider Problem (1.1) where $\mathcal{X} = \mathcal{P} \cap C$, $\mathcal{P}$ is a polytope over which we can minimize a linear function efficiently, and C is a convex compact set for which one can easily build a barrier function.



(a) Plot of $f(\mathbf{x})$. (b) Plot of $f(\mathbf{x}) + \mu' \Phi_C(\mathbf{x})$. (c) Plot of $f(\mathbf{x}) + \mu \Phi_C(\mathbf{x})$.

Figure 1: Minimizing $f(\mathbf{x})$ over $\mathcal{P} \cap C$, versus minimizing the sum of $f(\mathbf{x})$ and $\Phi_C(\mathbf{x})$ over $\mathcal{P}$ for two different penalty values μ' and μ such that $\mu' \gg \mu$.

Solving a linear optimization problem over $\mathcal{X}$ may be extremely expensive. In light of this, we can incorporate C into the problem through the use of a barrier penalty in the objective function, minimizing instead $f(\mathbf{x})+\mu\Phi_C(\mathbf{x})$ where $\Phi_C(\mathbf{x})$ is a log-barrier function for C and μ is a parameter controlling the penalization. This reformulation is illustrated in Figure 1. Note that if the original objective function is generalized self-concordant, so is the new objective function (see Proposition 1 in Sun & Tran-Dinh (2019)). We assume that computing the gradient of $f(\mathbf{x}) + \mu\Phi_C(\mathbf{x})$ is roughly as expensive as computing the gradient for $f(\mathbf{x})$ and solving an LP over $\mathcal{P}$ is inexpensive relative to solving an LP over $\mathcal{P} \cap C$. The μ parameter can be driven down to 0 after a solution converges in a warm-starting procedure similar to interior-point methods, ensuring convergence to the true optimum.

An additional advantage of this transformation of the problem is the solution structure. Running Frank-Wolfe on the set $\mathcal{P} \cap C$ could potentially select a large number of extremal points from $\mathrm{Bd}(C)$ if C is non-polyhedral. In contrast, $\mathcal{P}$ has a finite number of vertices, a small subset of which will be selected throughout the optimization procedure. The same solution as that of the original problem can thus be constructed as a convex combination of a small number of vertices of $\mathcal{P}$, improving sparsity and interpretability in many applications.

The following definition formalizes the setting of Problem (1.1).

Definition 1.3 (Generalized self-concordant function)**.** Let $f \in C^3(\mathrm{dom}(f))$ be a closed convex function with $\mathrm{dom}(f) \subseteq \mathbb{R}^n$ open. Then f is (M, ν) *generalized self-concordant* if:

$$\left| \left\langle D^3 f(\mathbf{x})[\mathbf{w}]\mathbf{u}, \mathbf{u} \right\rangle \right| \leq M \left\|\mathbf{u}\right\|^2_{\nabla^2 f(\mathbf{x})} \left\|\mathbf{w}\right\|^{\nu-2}_{\nabla^2 f(\mathbf{x})} \left\|\mathbf{w}\right\|_2^{3-\nu},$$

for any $\mathbf{x} \in \mathrm{dom}(f)$ and $\mathbf{u}, \mathbf{w} \in \mathbb{R}^n$, where $D^3 f(\mathbf{x})[\mathbf{w}] = \lim_{\alpha \to 0} \alpha^{-1} \left(\nabla^2 f(\mathbf{x} + \alpha\mathbf{w}) - \nabla^2 f(\mathbf{x})\right)$.

2 Frank-Wolfe Convergence Guarantees

We establish convergence rates for a Frank-Wolfe variant with an open-loop step size strategy on generalized self-concordant functions. The *Monotonous Frank-Wolfe (`M-FW`)* algorithm presented in Algorithm 1 is a simple, but powerful modification of the standard Frank-Wolfe algorithm, with the only difference that before taking a step, we verify if $\mathbf{x}_t + \gamma_t(\mathbf{v}_t - \mathbf{x}_t) \in \mathrm{dom}(f)$, and if so, we check whether moving to the next iterate provides primal progress. Note, that the open-loop step size rule $2/(t + 2)$ does not guarantee monotonous primal progress for the vanilla Frank-Wolfe algorithm in general. If either of these two checks fails, we do not move: the algorithm sets $\mathbf{x}_{t+1} = \mathbf{x}_t$ in Line 6 of Algorithm 1.

Algorithm 1 Monotonous Frank-Wolfe (`M-FW`)

Input: Point $\mathbf{x}_0 \in \mathcal{X} \cap \mathrm{dom}(f)$, function f
Output: Iterates $\mathbf{x}_1, \ldots \in \mathcal{X}$

1: **for** $t = 0$ **to** $\ldots$ **do**
2: $\quad \mathbf{v}_t \leftarrow \mathrm{argmin}_{\mathbf{v} \in \mathcal{X}} \langle \nabla f(\mathbf{x}_t), \mathbf{v} \rangle$
3: $\quad \gamma_t \leftarrow 2/(t+2)$
4: $\quad \mathbf{x}_{t+1} \leftarrow \mathbf{x}_t + \gamma_t(\mathbf{v}_t - \mathbf{x}_t)$
5: $\quad$ **if** $\mathbf{x}_{t+1} \notin \mathrm{dom}(f)$ **or** $f(\mathbf{x}_{t+1}) > f(\mathbf{x}_t)$ **then**
6: $\qquad \mathbf{x}_{t+1} \leftarrow \mathbf{x}_t$

As customary, we assume short-circuit evaluation of the logical conditions in Algorithm 1, i.e., if the first condition in Line 5 is true, then the second condition is not even checked, and the algorithm directly goes to Line 6. This minor modification of the vanilla Frank-Wolfe algorithm enables us to use the monotonicity of the iterates in the proofs to come, at the expense of at most one extra function evaluation per iteration. Note that if we set $\mathbf{x}_{t+1} = \mathbf{x}_t$, we do not need to call the FOO or LMO oracle at iteration $t+1$, as we can simply reuse $\nabla f(\mathbf{x}_t)$ and $\mathbf{v}_t$. This effectively means that between successive iterations in which we search for an acceptable value of γ_t, we only need to call the zeroth-order and domain oracle.

Remark 2.1. In practice, one can search for a suitable step size logarithmically, that is, halving the value of γ_t if either $\mathbf{x}_{t+1} \notin \mathrm{dom}(f)$ or $f(\mathbf{x}_{t+1}) > f(\mathbf{x}_t)$. This would lead to a step size that is at most a factor of 2 smaller than the non-zero step size that would have been eventually accepted by the Monotonous Frank-Wolfe (`M-FW`) algorithm in Algorithm 1. Moreover, this strategy would allow us to obtain convergence rates that are very similar to those of the Monotonous Frank-Wolfe (`M-FW`) algorithm, with the exception of a factor of 2. The motivation and implications of this variant are detailed in Appendix A. For simplicity, we focus on the step size presented in Algorithm 1 in the main body of the paper.

In order to establish the main convergence results for the algorithm, we lower bound the progress per iteration with the help of Proposition 2.2.

Proposition 2.2. *(C.f., (Sun & Tran-Dinh, 2019, Proposition 10)) Given a (M, ν) generalized self-concordant function, then for $\nu \geq 2$, we have that:*

$$f(\mathbf{y}) - f(\mathbf{x}) - \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle \leq \omega_\nu(d_\nu(\mathbf{x} - \mathbf{y})) \left\| \mathbf{y} - \mathbf{x} \right\|^2_{\nabla^2 f(\mathbf{x})}, \tag{2.1}$$

where the inequality holds if and only if $d_\nu(\mathbf{x}, \mathbf{y}) < 1$ for $\nu > 2$, and we have that,

$$d_\nu(\mathbf{x}, \mathbf{y}) \stackrel{\text{def}}{=} \begin{cases} M \left\| \mathbf{y} - \mathbf{x} \right\| & \text{if } \nu = 2 \\ (\frac{\nu}{2} - 1) M \left\| \mathbf{y} - \mathbf{x} \right\|^{3-\nu} \left\| \mathbf{y} - \mathbf{x} \right\|^{\nu-2}_{\nabla^2 f(\mathbf{x})} & \text{if } \nu > 2, \end{cases}$$

where:

$$\omega_\nu(\tau) \stackrel{\text{def}}{=} \begin{cases} \frac{e^\tau - \tau - 1}{\tau^2} & \text{if } \nu = 2 \\ \frac{-\tau - ln(1-\tau)}{\tau^2} & \text{if } \nu = 3 \\ \frac{(1-\tau) ln(1-\tau) + \tau}{\tau^2} & \text{if } \nu = 4 \\ \left(\frac{\nu-2}{4-\nu} \right) \frac{1}{\tau} \left[\frac{\nu-2}{2(3-\nu)\tau} \left((1-\tau)^{\frac{2(3-\nu)}{2-\nu}} - 1 \right) - 1 \right] & \text{otherwise.} \end{cases}$$

The inequality shown in Equation (2.1) is very similar to the one that we would obtain if the gradient of f were Lipschitz continuous, however, while the Lipschitz continuity of the gradient leads to an inequality that holds globally for all $\mathbf{x}, \mathbf{y} \in \mathrm{dom}(f)$, the inequality in Equation (2.1) only holds for $d_\nu(\mathbf{x}, \mathbf{y}) < 1$. Moreover, there are two other important differences, the norm used in Equation (2.1) is now the norm defined by the Hessian at $\mathbf{x}_t$ instead of the ℓ_2 norm, and the term multiplying the norm is $\omega_\nu(d_\nu(\mathbf{x}, \mathbf{y}))$ instead of $1/2$. We deal with the latter issue by bounding $\omega_\nu(d_\nu(\mathbf{x}, \mathbf{y}))$ with a constant that depends on ν for any $\mathbf{x}, \mathbf{y} \in \mathrm{dom}(f)$ such that $d_\nu(\mathbf{x}, \mathbf{y}) \leq 1/2$, as shown in Remark 2.3.

Remark 2.3. As $\mathrm{d}\,\omega_\nu(\tau)/\mathrm{d}\,\tau > 0$ for $\tau < 1$ and $\nu \geq 2$, then $\omega_\nu(\tau) \leq \omega_\nu(1/2)$ for $\tau \leq 1/2$.

Due to the fact that we use a simple step size $\gamma_t = 2/(t+2)$, that we make monotonous progress, and we ensure that the iterates are inside $\mathrm{dom}(f)$, careful accounting allows us to bound the number of iterations until $d_\nu(\mathbf{x}_t, \mathbf{x}_t + \gamma_t(\mathbf{v}_t - \mathbf{x}_t)) \leq 1/2$. Before formalizing the convergence rate we first review a lemma that we will need in the proof.

Lemma 2.4. *(C.f., ([Sun & Tran-Dinh, 2019], Proposition 7)) Let f be a generalized self-concordant function with $\nu > 2$. If $d_\nu(\mathbf{x}, \mathbf{y}) < 1$ and $\mathbf{x} \in \mathrm{dom}(f)$ then $\mathbf{y} \in \mathrm{dom}(f)$. For the case $\nu = 2$ we have that $\mathrm{dom}(f) = \mathbb{R}^n$.*

Putting all these things together allows us to obtain a convergence rate for Algorithm 1.

Theorem 2.5. *Suppose $\mathcal{X}$ is a compact convex set and f is a (M, ν) generalized self-concordant function with $\nu \geq 2$. Then the Monotonous Frank-Wolfe algorithm (Algorithm 1) satisfies:*

$$h(\mathbf{x}_t) \leq \frac{4(T_\nu + 1)}{t+1} \max\left\{h(\mathbf{x}_0), L_f^{\mathcal{L}_0} D^2 \omega_\nu(1/2)\right\}. \tag{2.2}$$

for $t \geq T_\nu$, where $L_f^{\mathcal{L}_0} = \max_{\mathbf{u} \in \mathcal{L}_0, \mathbf{d} \in \mathbb{R}^n} \|\mathbf{d}\|^2_{\nabla^2 f(\mathbf{u})} / \|\mathbf{d}\|_2^2$ and T_ν is defined as:

$$T_\nu \stackrel{\text{def}}{=} \begin{cases} \lceil 4MD \rceil - 2 & \text{if } \nu = 2 \\ \left\lceil 2MD(L_f^{\mathcal{L}_0})^{\nu/2-1}(\nu - 2) \right\rceil - 2 & \text{otherwise.} \end{cases} \tag{2.3}$$

Otherwise it holds that $h(\mathbf{x}_t) \leq h(\mathbf{x}_0)$ for $t < T_\nu$.

Proof. Consider the compact set $\mathcal{L}_0 \stackrel{\text{def}}{=} \{\mathbf{x} \in \mathrm{dom}(f) \cap \mathcal{X} \mid f(\mathbf{x}) \leq f(\mathbf{x}_0)\}$. As the algorithm makes monotonous progress and moves towards points such that $\mathbf{x}_t \in \mathcal{X}$, then $\mathbf{x}_t \in \mathcal{L}_0$ for $t \geq 0$. As the smoothness parameter of f is bounded over $\mathcal{L}_0$, we have from the properties of smooth functions that the bound $\|\mathbf{d}\|^2_{\nabla^2 f(\mathbf{x}_t)} / \|\mathbf{d}\|^2 \leq L_f^{\mathcal{L}_0}$ holds for any $\mathbf{d} \in \mathbb{R}^n$. Particularizing for $\mathbf{d} = \mathbf{x}_t - \mathbf{v}_t$ and noting that $\|\mathbf{x}_t - \mathbf{v}_t\| \leq D$ leads to $\|\mathbf{x}_t - \mathbf{v}_t\|^2_{\nabla^2 f(\mathbf{x}_t)} \leq L_f^{\mathcal{L}_0} D^2$. We then define T_ν as in Equation (2.3). Note that for $t \geq T_\nu$ we have that $d(\mathbf{x}_t, \mathbf{x}_t + \gamma_t(\mathbf{v}_t - \mathbf{x}_t)) \leq 1/2$, and so as $\mathbf{x}_t \in \mathrm{dom}(f)$ we will have $\mathbf{x}_t + \gamma_t(\mathbf{v}_t - \mathbf{x}_t) \in \mathrm{dom}(f)$, by application of Lemma 2.4. This means that the non-zero step size γ_t will ensure that $\mathbf{x}_t + \gamma_t(\mathbf{v}_t - \mathbf{x}_t) \in \mathrm{dom}(f)$ in Line 5 of Algorithm 1. Moreover, it allows us to use the bound between points $\mathbf{x}_t$ and $\mathbf{x}_t + \gamma_t (\mathbf{v}_t - \mathbf{x}_t)$ in Proposition 2.2, which holds for $d(\mathbf{x}_t, \mathbf{x}_t + \gamma_t (\mathbf{v}_t - \mathbf{x}_t)) < 1$. With this we can estimate the primal progress we can guarantee for $t \geq T_\nu$ if we move from $\mathbf{x}_t$ to $\mathbf{x}_t + \gamma_t (\mathbf{v}_t - \mathbf{x}_t)$:

$$\begin{aligned} h(\mathbf{x}_t + \gamma_t (\mathbf{v}_t - \mathbf{x}_t)) &\leq h(\mathbf{x}_t) - \gamma_t g(\mathbf{x}_t) + \gamma_t^2 \omega_\nu \left(d_\nu (\mathbf{x}_t, \mathbf{x}_t + \gamma_t (\mathbf{v}_t - \mathbf{x}_t))\right) \|\mathbf{v}_t - \mathbf{x}_t\|^2_{\nabla^2 f(\mathbf{x}_t)} \\ &\leq h(\mathbf{x}_t)(1 - \gamma_t) + \gamma_t^2 L_f^{\mathcal{L}_0} D^2 \omega_\nu(1/2), \end{aligned}$$

where the second inequality follows from the upper bound on the primal gap via the Frank-Wolfe gap $g(\mathbf{x}_t)$, the application of Remark 2.3 as for $t \geq T_\nu$ we have that $d_\nu(\mathbf{x}_t, \mathbf{x}_t + \gamma_t (\mathbf{v}_t - \mathbf{x}_t)) \leq 1/2$, and from the fact that $\mathbf{x}_t \in \mathcal{L}_0$ for all $t \geq 0$. With the previous chain of inequalities we can bound the primal progress for $t \geq T_\nu$ as

$$h(\mathbf{x}_t) - h(\mathbf{x}_t + \gamma_t (\mathbf{v}_t - \mathbf{x}_t)) \geq \gamma_t h(\mathbf{x}_t) - \gamma_t^2 L_f^{\mathcal{L}_0} D^2 \omega_\nu(1/2). \tag{2.4}$$

From these facts we can prove the convergence rate shown in Equation (2.2) by induction. The base case $t = T_\nu$ holds trivially by the fact that using monotonicity we have that $h(\mathbf{x}_{T_\nu}) \leq h(\mathbf{x}_0)$. Assuming the claim is true for some $t \geq T_\nu$ we distinguish two cases.

Case $\gamma_t h(\mathbf{x}_t) - \gamma_t^2 L_f^{\mathcal{L}_0} D^2 \omega_\nu(1/2) > 0$: Focusing on the first case, we can plug the previous inequality into Equation (2.4) to find that γ_t guarantees primal progress, that is, $h(\mathbf{x}_t) > h(\mathbf{x}_t + \gamma_t (\mathbf{v}_t - \mathbf{x}_t))$ with the step size γ_t, and so we know that we will not go into Line 6 of Algorithm 1, and we have that $h(\mathbf{x}_{t+1}) = h(\mathbf{x}_t + \gamma_t (\mathbf{v}_t - \mathbf{x}_t))$. Thus using the induction hypothesis and plugging in the expression for $\gamma_t = 2/(t+2)$ into Equation (2.4) we have:

$$\begin{aligned} h(\mathbf{x}_{t+1}) &\leq 4 \max\left\{h(\mathbf{x}_0), L_f^{\mathcal{L}_0} D^2 \omega_\nu(1/2)\right\} \left(\frac{(T_\nu + 1)t}{(t+1)(t+2)} + \frac{1}{(t+2)^2}\right) \\ &\leq \frac{4(T_\nu + 1)}{t+2} \max\left\{h(\mathbf{x}_0), L_f^{\mathcal{L}_0} D^2 \omega_\nu(1/2)\right\}, \end{aligned}$$

where we use that $(T_\nu + 1)t/(t+1) + 1/(t+2) \leq T_\nu + 1$ for all $t \geq 0$ and any $t \geq T_\nu$.

Case $\gamma_t h(\mathbf{x}_t) - \gamma_t^2 L_f^{\mathcal{L}_0} D^2 \omega_\nu(1/2) \leq 0$: In this case, we cannot guarantee that the step size γ_t provides primal progress by plugging into Equation (2.4), and so we cannot guarantee if a step size of γ_t will be accepted and we will have $\mathbf{x}_{t+1} = \mathbf{x}_t + \gamma_t (\mathbf{v}_t - \mathbf{x}_t)$, or we will simply have $\mathbf{x}_{t+1} = \mathbf{x}_t$, that is, we may go into Line 6 of Algorithm 1. Nevertheless, if we reorganize the expression $\gamma_t h(\mathbf{x}_t) - \gamma_t^2 L_f^{\mathcal{L}_0} D^2 \omega_\nu(1/2) \leq 0$, by monotonicity we will have that:

$$h(\mathbf{x}_{t+1}) \leq h(\mathbf{x}_t) \leq \frac{2}{t+2} L_f^{\mathcal{L}_0} D^2 \omega_\nu(1/2) \leq \frac{4(T_\nu+1)}{t+2} \max\left\{h(\mathbf{x}_0), L_f^{\mathcal{L}_0} D^2 \omega_\nu(1/2)\right\}.$$

Where the last inequality holds as $2 \leq 4(T_\nu + 1)$ for any $T_\nu \geq 0$. □

Remark 2.6. In the case where $\nu = 2$ we can easily bound the primal gap $h(\mathbf{x}_1)$, as in this setting $\mathrm{dom}(f) = \mathbb{R}^n$, which leads to $h(\mathbf{x}_1) \leq L_f^{\mathcal{X}} D^2$ from Equation (2.4), regardless of if we set $\mathbf{x}_1 = \mathbf{x}_0$ or $\mathbf{x}_1 = \mathbf{v}_0$. Moreover, as the upper bound on the Bregman divergence holds for $\nu = 2$ regardless of the value of $d_2(\mathbf{x}, \mathbf{y})$, we can modify the proof of Theorem 2.5 to obtain a convergence rate of the form $h(\mathbf{x}_t) \leq 2/(t+1) L_f^{\mathcal{X}} D^2 w_2(MD)$ for $t \geq 1$, which is reminiscient of the $O(L_f^{\mathcal{X}} D^2/t)$ rate of the original Frank-Wolfe algorithm for the smooth and convex case.

Note that in the proof of Theorem 2.5 we explicitly use the progress bound from generalized self-concordance as opposed to the progress bound that arises from $L_f^{\mathcal{L}_0}$-smoothness, as there is no straightforward way to bound the number of iterations until the latter progress bound holds indefinitely for all $\mathbf{x}_t + \gamma_t(\mathbf{v}_t - \mathbf{x}_t)$, while there is a straightforward criterion on γ_t that allows us to ensure that the former holds from some point onward (see Remark A.1 in the Appendix for more details). Furthermore, with this simple step size we can also prove a convergence rate for the Frank-Wolfe gap, as shown in Theorem 2.7 (see Theorem A.2 in the Appendix for the proof).

Theorem 2.7. *Suppose $\mathcal{X}$ is a compact convex set and f is a (M, ν) generalized self-concordant function with $\nu \geq 2$. Then if the Monotonous Frank-Wolfe algorithm (Algorithm 1) is run for $T \geq T_\nu + 6$ iterations, we will have that $\min_{1 \leq t \leq T} g(\mathbf{x}_t) \leq O(1/T)$.*

Remark 2.8. Note that the Monotonous Frank-Wolfe algorithm (Algorithm 1) performs at most one ZOO, FOO, DO, and LMO oracle call per iteration. This means that Theorems 2.5 and 2.7 effectively bound the number of ZOO, FOO, DO, and LMO oracle calls needed to achieve a target primal gap or Frank-Wolfe gap accuracy ε as a function of T_ν and ε; note that T_ν is independent of ε. This is an important difference with respect to existing bounds, as the existing Frank-Wolfe-style first-order algorithms for generalized self-concordant functions in the literature that utilize various types of line searches may perform more than one ZOO or DO call per iteration in the line search. This means that the convergence bounds in terms of iteration count of these algorithms are only informative when considering the number of FOO and LMO calls that are needed to reach a target accuracy in primal gap, and do not directly provide any information regarding the number of ZOO or DO calls that are needed. In order to bound the latter two quantities one typically needs additional technical tools. For example, for the backtracking line search of Pedregosa et al. (2020), one can use Theorem 1 in Appendix C of Pedregosa et al. (2020), or a slightly modified version of Lemma 4 in Nesterov (2013), to find a bound for the number of ZOO or DO calls that are needed to find an ε-optimal solution. Note that these bounds depend on user-defined initialization or tuning parameters provided by the user.

In Table 2 we provide a detailed complexity comparison between the Monotonous Frank-Wolfe (`M-FW`) algorithm (Algorithm 1), and other comparable algorithms in the literature.

Algorithm	**SOO calls**	**FOO calls**	**ZOO calls**	**LMO calls**	**DO calls**
`FW-GSC` [1, Alg.2]	$O(1/\varepsilon)$	$O(1/\varepsilon)$		$O(1/\varepsilon)$	
`LBTFW-GSC`‡ [1, Alg.3]		$O(1/\varepsilon)$	$O(1/\varepsilon)$	$O(1/\varepsilon)$	$O(1/\varepsilon)$
`MBTFW-GSC`‡ [1, Alg.5]	$O(1/\varepsilon)$	$O(1/\varepsilon)$	$O(1/\varepsilon)$	$O(1/\varepsilon)$	$O(1/\varepsilon)$
`M-FW`† **[This work]**		$O(1/\varepsilon)$	$O(1/\varepsilon)$	$O(1/\varepsilon)$	$O(1/\varepsilon)$

Table 2: **Complexity comparison**: Number of iterations needed to reach a solution with $h(\mathbf{x})$ below ε for Problem 1.1. We denote Dvurechensky et al. (2020b) using [1]. We use the superscript † to indicate that the same complexities hold for reaching an ε-optimal solution in $g(\mathbf{x})$. The superscript ‡ is used to indicate that constants in the convergence bounds depend on user-defined inputs; the other algorithms are parameter-free.

We note that the LBTFW-GSC algorithm from Dvurechensky et al. (2020b) is in essence the Frank-Wolfe algorithm with a modified version of the backtracking line search of Pedregosa et al. (2020). In the next section, we provide improved convergence guarantees for various cases of interest for this algorithm, which we refer to as the Frank-Wolfe algorithm with Backtrack (B-FW) for simplicity.

2.1 Improved convergence guarantees

Algorithm 2 Frank-Wolfe with Backtrack of Pedregosa et al. (2020) (B-FW)

Input: $\mathbf{x}_0 \in \mathcal{X} \cap \operatorname{dom}(f)$, function f, estimate L_{-1}
Output: Iterates $\mathbf{x}_1, \ldots \in \mathcal{X}$

for $t = 0$ **to** $\ldots$ **do**
  $\mathbf{v}_t \leftarrow \operatorname{argmin}_{\mathbf{v}\in\mathcal{X}} \langle \nabla f(\mathbf{x}_t), \mathbf{v}\rangle$
  $\gamma_t, L_t \leftarrow \texttt{Backtrack}(f, \mathbf{x}_t, \mathbf{v}_t - \mathbf{x}_t, L_{t-1}, 1)$
  $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t + \gamma_t(\mathbf{v}_t - \mathbf{x}_t)$

We will now establish improved convergence rates for various special cases. We first focus on the assumption that $\mathbf{x}^* \in \operatorname{Int}(\mathcal{X} \cap \operatorname{dom}(f))$, obtaining improved rates when we use the FW algorithm coupled with the adaptive step size strategy from Pedregosa et al. (2020) (see Algorithm 6 in Appendix).This assumption is reasonable if for example $\operatorname{Bd}(\mathcal{X}) \not\subseteq \operatorname{dom}(f)$, and $\operatorname{Int}(\mathcal{X}) \subseteq \operatorname{dom}(f)$. That is to say, we will have that $\mathbf{x}^* \in \operatorname{Int}(\mathcal{X} \cap \operatorname{dom}(f))$ if for example we use logarithmic barrier functions to encode a set of constraint, and we have that $\operatorname{dom}(f)$ is a proper subset of $\mathcal{X}$. In this case the optimum is guaranteed to be in $\operatorname{Int}(\mathcal{X} \cap \operatorname{dom}(f))$.

The analysis in this case is reminiscent of the one in the seminal work of Guélat & Marcotte (1986), and is presented in Appendix A.2. Note that we can upper bound the value of L_t for $t \geq 0$ by $\tilde{L} \stackrel{\text{def}}{=} \max\{\tau L_f^{\mathcal{L}_0}, L_{-1}\}$, where $\tau > 1$ is the backtracking parameter and L_{-1} is the initial smoothness estimate in Algorithm 6 .

Theorem 2.9. *Let f be a (M, ν) generalized self-concordant function with $\nu \geq 2$ and let $\operatorname{dom}(f)$ not contain straight lines. Furthermore, we denote by $r > 0$ the largest value such that $\mathcal{B}(\mathbf{x}^*, r) \subseteq \mathcal{X} \cap \operatorname{dom}(f)$. Then the Frank-Wolfe algorithm with Backtrack (Algorithm 2) achieves a convergence rate for $t \geq 1$ of $h(\mathbf{x}_t) \leq h(\mathbf{x}_0)\left(1 - \mu_f^{\mathcal{L}_0}/(2\tilde{L})(r/D)^2\right)^t$.*

The assumption that $\operatorname{dom}(f)$ does not contain straight lines in Theorem 2.9 is related to the Hessian being positive definite over $\operatorname{dom}(f)$ (see the proof in the Appendix in Theorem A.5). Note that this is a very mild assumption as we can simply modify the function with a very small ℓ_2 regularizer, as e.g., in Nesterov (2012). Next, we recall the definition of uniformly convex sets, used in Kerdreux et al. (2021), which will allow us to obtain improved convergence rates for the FW algorithm over uniformly convex feasible regions.

Definition 2.10 ((κ, q)-uniformly convex set)**.** Given two positive numbers κ and q, we say the set $\mathcal{X} \subseteq \mathbb{R}^n$ is (κ, q)-*uniformly convex* with respect to a norm $\|\cdot\|$ if for any $\mathbf{x}, \mathbf{y} \in \mathcal{X}$, $0 \leq \gamma \leq 1$, and $\mathbf{z} \in \mathbb{R}^n$ with $\|\mathbf{z}\| = 1$ we have that $\mathbf{y} + \gamma(\mathbf{x} - \mathbf{y}) + \gamma(1-\gamma)\cdot\kappa\|\mathbf{x} - \mathbf{y}\|^q\,\mathbf{z} \in \mathcal{X}$.

Theorem 2.11. *Suppose $\mathcal{X}$ is a compact (κ, q)-uniformly convex set and f is a (M, ν) generalized self-concordant function with $\nu \geq 2$. Furthermore, assume that $\min_{\mathbf{x}\in\mathcal{X}} \|\nabla f(\mathbf{x})\| \geq C > 0$. Then the Frank-Wolfe algorithm with Backtrack (Algorithm 2) achieves a convergence rate of:*

$$h_t \leq \begin{cases} h(\mathbf{x}_0)\left(1 - \frac{1}{2}\min\left\{1, \frac{\kappa C}{\tilde{L}}\right\}\right)^t & \text{if } q = 2 \\ \frac{h(\mathbf{x}_0)}{2^t} & \text{if } q > 2, 1 \leq t \leq t_0 \\ \frac{\tilde{L}^{q/(q-2)}/(\kappa C)^{2/(q-2)}}{(1+(q-2)(t-t_0)/(2q))^{q/(q-2)}} = O\left(t^{-q/(q-2)}\right) & \text{if } q > 2, t > t_0, \end{cases}$$

for $t \geq 1$, where $t_0 = \max\{1, \lfloor \log_{1/2}((\tilde{L}^q/(\kappa C)^2)^{1/(q-2)} h(\mathbf{x}_0)) \rfloor\}$.

However, in the general case, we cannot assume that the norm of the gradient is bounded away from zero over $\mathcal{X}$. We deal with the general case in Theorem 2.12

Theorem 2.12. *Suppose $\mathcal{X}$ is a compact (κ, q)-uniformly convex set and f is a (M, ν) generalized self-concordant function with $\nu \geq 2$ for which the domain does not contain straight lines. Then* *the Frank-Wolfe algorithm with Backtrack (Algorithm 2) results in a convergence rate:*

$$h_t \leq \begin{cases} \frac{h(\mathbf{x}_0)}{2^t} & \text{if } 1 \leq t \leq t_0 \\ \frac{(\tilde{L}^q/(\kappa^2 \mu_f^{\mathcal{L}_0}))^{1/(q-1)}}{(1+(q-1)(t-t_0)/(2q))^{q/(q-1)}} = \mathcal{O}\left(t^{-q/(q-1)}\right) & \text{if } t > t_0, \end{cases}$$

for $t \geq 1$*, where* $t_0 = \max\{1, \lfloor \log_{1/2}((\tilde{L}^q/(\kappa^2 \mu_f^{\mathcal{L}_0}))^{1/(q-1)}/h(\mathbf{x}_0)) \rfloor\}$.

In Table 3 in Appendix A.2 we summarize the oracle complexity results shown in this paper for the `B-FW` algorithm when minimizing over a (κ, q)-uniformly convex set. Note that this algorithm is referred to as `LBTFW-GSC` in Dvurechensky et al. (2020b).

Remark 2.13. Contrary to previous claims, there is no obstacle for the *Away-step Frank-Wolfe* (AFW) algorithm (Guélat & Marcotte, 1986; Lacoste-Julien & Jaggi, 2015) together with the step size strategy in Algorithm 6 to obtain a linear convergence rate in primal and Frank-Wolfe gap when $\mathcal{X}$ is a polytope and f is generalized self-concordant. This is not surprising, as f is strongly convex and smooth over $\mathcal{L}_0$ if $\text{dom}(f)$ does not contain straight lines, and monotonicity ensures the feasibility of the iterates. We leave the analysis for this case to Appendix B, and the formal convergence statement to Theorem B.2 and B.3.

In Table 4 in Appendix B we provide a complexity comparison between the `B-AFW` algorithm, which can be found in Algorithm 7 in the appendix, and other comparable algorithms in the literature. Note that these complexities assume that $\mathcal{X}$ is polyhedral.

3 Computational experiments

We showcase the performance of the `M-FW` algorithm, the second-order step size and the LLOO algorithm from Dvurechensky et al. (2020b) (denoted by `GSC-FW` and `LLOO` in the figures) and the Frank-Wolfe and the Away-Step Frank-Wolfe algorithm with the backtracking stepsize of Pedregosa et al. (2020), denoted by `B-FW` and `B-AFW` respectively. All experiments are carried out in `Julia` using the `FrankWolfe.jl` package (Besançon et al., 2021), and the examples considered extend the ones presented in Dvurechensky et al. (2020b) and Liu et al. (2020). The code can be found in the `fw-generalized-selfconcordant` repository. We also use the vanilla FW algorithm denoted by `FW`, which is simply Algorithm 1 without Lines 5 and 6 using the traditional $\gamma_t = 2/(t+2)$ open-loop step size rule. Note that there are no formal convergence guarantees for this algorithm when applied to Problem (1.1). Details and remarks on the data and the experimental setup are provided in Appendix C. All figures show the evolution of the $h(\mathbf{x}_t)$ and $g(\mathbf{x}_t)$ against t and time with a log-log scale. As in Dvurechensky et al. (2020b) we implemented the LLOO based variant only for the portfolio optimization instance Δ_n; for the other examples, the oracle implementation was not implemented due to the need to estimate non-trivial parameters.

As can be seen in all experiments, the Monotonous Frank-Wolfe algorithm is very competitive, outperforming previously proposed variants in both in progress per iteration and time. The only other algorithm that is sometimes faster is the Away-Step Frank-Wolfe variant as detailed in Remark 2.13, which however depends on an active set, and can induce up to a quadratic overhead, making iterations progressively more expensive; this can also be observed in our experiments as the advantage in time is much less pronounced than in iterations.

Portfolio optimization. We consider $f(\mathbf{x}) = -\sum_{t=1}^{p} \log(\langle \mathbf{r}_t, \mathbf{x} \rangle)$, where p denotes the number of periods and $\mathcal{X} = \Delta_n$. The results are shown in Figure 2.

Signal recovery with KL divergence. We apply the aforementioned algorithms to the recovery of a sparse signal from a noisy linear image using the Kullback-Leibler divergence, expressed as $f(\mathbf{x}) = D(W\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{N} \left\{ \langle \mathbf{w}_i, \mathbf{x} \rangle \log\left(\frac{\langle \mathbf{w}_i, \mathbf{x} \rangle}{y_i}\right) - \langle \mathbf{w}_i, \mathbf{x} \rangle + y_i \right\}$, where $\mathbf{w}_i$ is the i^{th} row of a matrix W. In order to promote sparsity and enforce nonnegativity of the solution, we use the unit simplex of radius R as the feasible set $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}_+^d, \|\mathbf{x}\|_1 \leq R\}$. The results are shown in Figure 3. We used the same $M = 1$ choice for the second-order method as in Dvurechensky et al. (2020b) for comparison; its admissibility is unknown (see Remark C.1).

Logistic regression. We consider a logistic regression task with a design matrix with rows $\mathbf{a}_i \in \mathbb{R}^n$ with $1 \leq i \leq N$ and a vector $\mathbf{y} \in \{-1, 1\}^N$ and formulate the problem with elastic net regularization, in a similar fashion as is done in Liu et al. (2020), with $f(\mathbf{x}) = 1/N \sum_{i=1}^{N} \log(1 + \exp(-y_i \langle \mathbf{x}, \mathbf{a}_i \rangle)) + \mu/2 \|\mathbf{x}\|^2$, where μ is a regularization parameter and $\mathcal{X}$ is the ℓ_1 ball of radius ρ. The results can be seen in Figure 4 and Appendix C.

Birkhoff polytope. All previously considered applications have in common a feasible region possessing computationally inexpensive LMOs (probability/unit simplex and ℓ_1 norm ball). Additionally, each vertex returned from the LMO is highly sparse with at most one non-zero element. To complement the results, we consider a convex quadratic optimization problem over the Birkhoff polytope, where the LMO call uses the Hungarian method and is not as inexpensive as in the other examples. The results are shown in Figure 5.

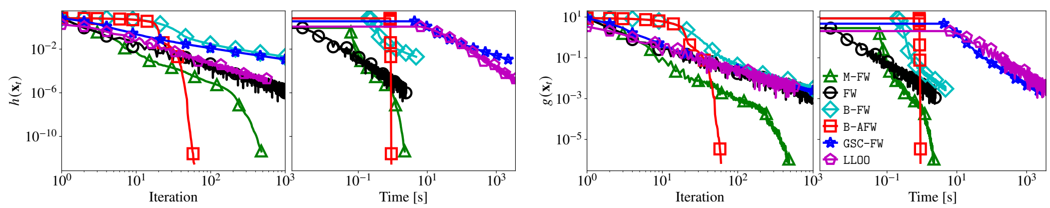


Figure 2: **Portfolio Optimization**: `LLOO` and `GSC-FW` perform similarly to `FW` on a per-iteration basis but the iterations are computationally more expensive. `B-AFW` is the fastest method both in terms iteration and runtime, followed by `M-FW` which is the only other method to terminate with the specified dual gap tolerance.

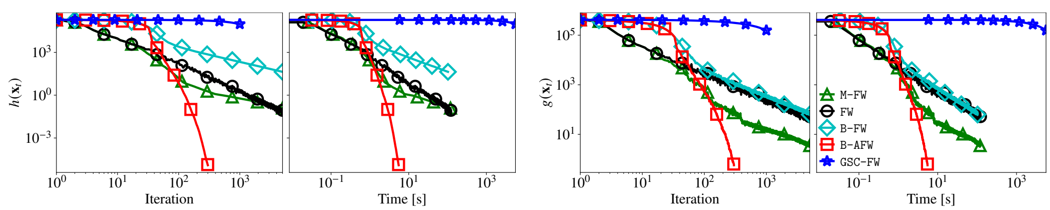


Figure 3: **Signal Recovery**: `B-AFW` significantly outperforms all other methods. `FW` and `B-FW` perform similarly in dual gap progress and converge slower than `M-FW`. In terms of primal gap progress, `M-FW` and `FW` perform similarly and outperform `B-FW`.

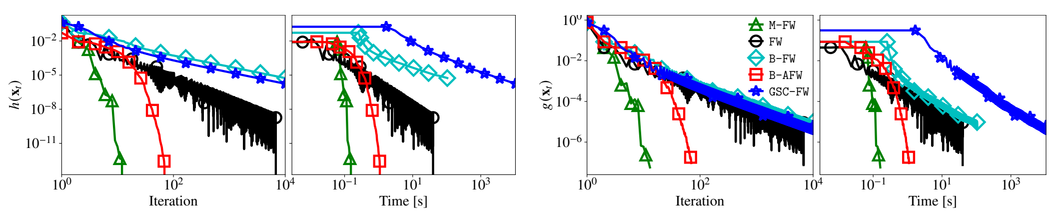


Figure 4: **Logistic Regression**: This instance shows that although simple in essence, `M-FW` can outperform other methods including `B-AFW` in terms of convergence. The primal and dual gaps for `B-FW` and `GSC-FW` converge at similar rates against iteration count.

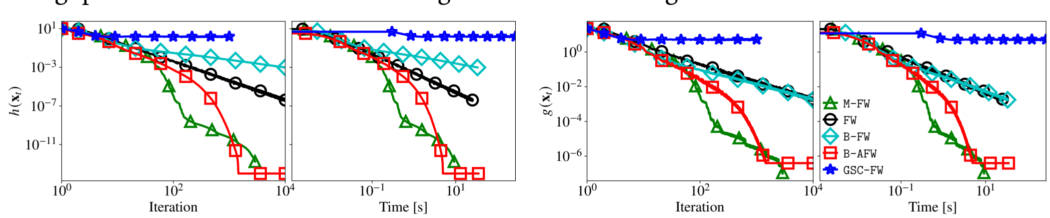


Figure 5: **Birkhoff Polytope**: `B-AFW` is the fastest-converging method for all measures. However, the dual gap reaches a plateau due to numerical issues above the termination threshold, unlike `M-FW` which reaches the dual gap tolerance. `GSC-FW` is run for 1000 iterations only given the longer runtime. Its slow progress is likely due to numerical instabilities in the Hessian computation which do not occur in first-order methods.

Acknowledgements

Research reported in this paper was partially supported through the Research Campus Modal funded by the German Federal Ministry of Education and Research (fund numbers 05M14ZAM,05M20ZBM) and the Deutsche Forschungsgemeinschaft (DFG) through the DFG Cluster of Excellence MATH+. We would like to thank the anonymous reviewers for their suggestions and comments.

References

Bach, F. Self-concordant analysis for logistic regression. *Electronic Journal of Statistics*, 4: 384–414, 2010.

Besançon, M., Carderera, A., and Pokutta, S. FrankWolfe.jl: a high-performance and flexible toolbox for Frank-Wolfe algorithms and conditional gradients. *arXiv preprint arXiv:2104.06675*, 2021.

Carderera, A., Besançon, M., and Pokutta, S. Frank-Wolfe for Generalized Self-Concordant Functions - Problem Instances, May 2021. URL https://doi.org/10.5281/zenodo.4836009.

Csiszar, I. Why least squares and maximum entropy? An axiomatic approach to inference for linear inverse problems. *The annals of statistics*, 19(4):2032–2066, 1991.

Diakonikolas, J., Carderera, A., and Pokutta, S. Locally accelerated conditional gradients. In *Proceedings of the 23th International Conference on Artificial Intelligence and Statistics*, pp. 1737–1747. PMLR, 2020.

Dvurechensky, P., Ostroukhov, P., Safin, K., Shtern, S., and Staudigl, M. Self-concordant analysis of Frank-Wolfe algorithms. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 2814–2824. PMLR, 2020a.

Dvurechensky, P., Safin, K., Shtern, S., and Staudigl, M. Generalized self-concordant analysis of Frank-Wolfe algorithms. *arXiv preprint arXiv:2010.01009*, 2020b.

Frank, M. and Wolfe, P. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.

Garber, D. and Hazan, E. A linearly convergent variant of the conditional gradient algorithm under strong convexity, with applications to online and stochastic optimization. *SIAM Journal on Optimization*, 26(3):1493–1528, 2016.

Guélat, J. and Marcotte, P. Some comments on Wolfe's 'away step'. *Mathematical Programming*, 35(1):110–119, 1986.

Jaggi, M. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning*, pp. 427–435. PMLR, 2013.

Kerdreux, T., d'Aspremont, A., and Pokutta, S. Restarting Frank-Wolfe. In *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*, pp. 1275–1283. PMLR, 2019.

Kerdreux, T., d'Aspremont, A., and Pokutta, S. Projection-free optimization on uniformly convex sets. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, pp. 19–27. PMLR, 2021.

Krishnan, R. G., Lacoste-Julien, S., and Sontag, D. Barrier Frank-Wolfe for Marginal Inference. In *Proceedings of the 28th Conference in Neural Information Processing Systems*. PMLR, 2015.

Lacoste-Julien, S. and Jaggi, M. On the global linear convergence of Frank-Wolfe optimization variants. In *Proceedings of the 29th Conference on Neural Information Processing Systems*, pp. 566–575. PMLR, 2015.

Levitin, E. S. and Polyak, B. T. Constrained minimization methods. *USSR Computational Mathematics and Mathematical Physics*, 6(5):1–50, 1966.

Liu, D., Cevher, V., and Tran-Dinh, Q. A Newton Frank-Wolfe method for constrained self-concordant minimization. *arXiv preprint arXiv:2002.07003*, 2020.

Marron, J. S., Todd, M. J., and Ahn, J. Distance-weighted discrimination. *Journal of the American Statistical Association*, 102(480):1267–1271, 2007.

Marteau-Ferey, U., Ostrovskii, D., Bach, F., and Rudi, A. Beyond least-squares: Fast rates for regularized empirical risk minimization through self-concordance. In *Proceedings of the 32nd Conference on Learning Theory*, pp. 2294–2340. PMLR, 2019.

Nesterov, Y. How to make the gradients small. *Optima. Mathematical Optimization Society Newsletter*, (88):10–11, 2012.

Nesterov, Y. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013.

Nesterov, Y. *Lectures on convex optimization*, volume 137. Springer, 2018.

Nesterov, Y. and Nemirovskii, A. *Interior-point polynomial algorithms in convex programming*. SIAM, 1994.

Odor, G., Li, Y.-H., Yurtsever, A., Hsieh, Y.-P., Tran-Dinh, Q., El Halabi, M., and Cevher, V. Frank-wolfe works for non-lipschitz continuous gradient objectives: scalable poisson phase retrieval. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6230–6234. Ieee, 2016.

Ostrovskii, D. M. and Bach, F. Finite-sample analysis of M-estimators using self-concordance. *Electronic Journal of Statistics*, 15(1):326–391, 2021.

Pedregosa, F., Negiar, G., Askari, A., and Jaggi, M. Linearly convergent Frank–Wolfe with backtracking line-search. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*. PMLR, 2020.

Sun, T. and Tran-Dinh, Q. Generalized self-concordant functions: a recipe for Newton-type methods. *Mathematical Programming*, 178(1):145–213, 2019.

Temlyakov, V. Greedy approximation in convex optimization. *Constructive Approximation*, 41 (2):269–296, 2015.

Tran-Dinh, Q., Li, Y.-H., and Cevher, V. Composite convex minimization involving self-concordant-like cost functions. In *Modelling, Computation and Optimization in Information Systems and Management Sciences*, pp. 155–168. Springer, 2015.

Wolfe, P. Convergence theory in nonlinear programming. In *Integer and Nonlinear Programming*, pp. 1–36. North-Holland, Amsterdam, 1970.

Zhao, R. and Freund, R. M. Analysis of the Frank-Wolfe method for logarithmically-homogeneous barriers, with an extension. *arXiv preprint arXiv:2010.08999*, 2020.

Simple steps are all you need: Frank-Wolfe and generalized self-concordant functions

Supplementary material

Outline. The appendix of the paper is organized as follows:

- Section A presents the full convergence proof of the Frank-Wolfe gap for the Monotonous Frank-Wolfe algorithm, and improved convergence bounds when using the Frank-Wolfe algorithm with the step size strategy of Pedregosa et al. (2020) when the optimum is contained in the interior of $\mathcal{X} \cap \mathrm{dom}(f)$, or when the feasible region is uniformly convex.
- Section B reviews the Away-step Frank-Wolfe algorithm (Guélat & Marcotte, 1986; Lacoste-Julien & Jaggi, 2015), and shows how using the step size strategy of Pedregosa et al. (2020) one can show a linear convergence in primal gap and in Frank-Wolfe gap when the feasible region is a polytope.
- Section C presents additional information about the experimental section of the paper.

A Monotonous Frank-Wolfe

This appendix contains the theoretical proofs that have not been included in the main body of the paper due to space constraints, as well as several remarks of interest. We start off with a remark regarding the convergence proof in Theorem 2.5, and continue by showing that the Monotonous Frank-Wolfe algorithm (restated for convenience in Algorithm 3) not only contracts the primal gap at a rate of $O(1/t)$ where t is the iteration count, but also ensures that the minimum of the Frank-Wolfe gap over the run of the algorithm is bounded by $O(1/t)$ from above.

Algorithm 3 Monotonous Frank-Wolfe (`M-FW`)

Input: Point $\mathbf{x}_0 \in \mathcal{X} \cap \mathrm{dom}(f)$, function f
Output: Iterates $\mathbf{x}_1, \ldots \in \mathcal{X}$

1: **for** $t = 0$ **to** $\ldots$ **do**
2: $\quad\mathbf{v}_t \leftarrow \mathrm{argmin}_{\mathbf{v}\in\mathcal{X}} \langle \nabla f(\mathbf{x}_t), \mathbf{v}\rangle$
3: $\quad\gamma_t \leftarrow 2/(t+2)$
4: $\quad\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t + \gamma_t(\mathbf{v}_t - \mathbf{x}_t)$
5: $\quad$**if** $\mathbf{x}_{t+1} \notin \mathrm{dom}(f)$ **or** $f(\mathbf{x}_{t+1}) > f(\mathbf{x}_t)$ **then**
6: $\quad\quad\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t$

Remark A.1 (Regarding the proof of Theorem 2.5)**.** One of the quantities that we have used in the proof of Theorem 2.5 is $L_f^{\mathcal{L}_0}$. Note that the function f is $L_f^{\mathcal{L}_0}$-smooth over $\mathcal{L}_0$. One could wonder then why we have bothered to use the bounds on the Bregman divergence in Proposition 2.2 for a (M, ν)-generalized self-concordant function, instead of simply using the bounds from the $L_f^{\mathcal{L}_0}$-smoothness of f over $\mathcal{L}_0$. The reason is that the upper bound on the Bregman divergence in Proposition 2.2 applies for any $\mathbf{x}, \mathbf{y} \in \mathrm{dom}(f)$ such that $d_\nu(\mathbf{x}, \mathbf{y}) < 1$, and we can easily bound the number of iterations T_ν it takes for the step size $\gamma_t = 2/(t+2)$ to verify both $\mathbf{x}_t, \mathbf{x}_t + \gamma_t(\mathbf{v}_t - \mathbf{x}_t) \in \mathrm{dom}(f)$ and $d_\nu(\mathbf{x}_t, \mathbf{x}_t + \gamma_t(\mathbf{v}_t - \mathbf{x}_t)) < 1$ for $t \geq T_\nu$. However, in order to apply the bounds on the Bregman divergence we need $\mathbf{x}_t, \mathbf{x}_t + \gamma_t(\mathbf{v}_t - \mathbf{x}_t) \in \mathcal{L}_0$, and while it is easy to show by monotonicity that $\mathbf{x}_t \in \mathcal{L}_0$, there is no straightforward way to prove that for some $\tilde{T}_\nu$ we have that $\mathbf{x}_t + \gamma_t(\mathbf{v}_t - \mathbf{x}_t) \in \mathcal{L}_0$ for all $t \geq \tilde{T}_\nu$.

As mentioned in Remark 2.1 in the main body of the text, in practice, a halving strategy for the step size is preferred for the implementation of the Monotonous Frank-Wolfe algorithm, as opposed to the step size implementation shown in Algorithm 3 (or Algorithm 1 in the main body). This halving strategy, which is shown in Algorithm 4, helps deal with the case in which a large number of **consecutive** step sizes γ_t are rejected either because $\mathbf{x}_t + \gamma_t(\mathbf{v}_t - \mathbf{x}_t) \notin \mathrm{dom}(f)$ or $f(\mathbf{x}_t + \gamma_t(\mathbf{v}_t - \mathbf{x}_t)) > f(\mathbf{x}_t)$, and helps avoid the need to potentially call the zeroth-order or domain oracle a large number of times in these cases. The halving strategy in Algorithm 4 results in a step size that is at most a factor of 2 smaller than the one that would have been accepted with the original strategy, i.e., that would have ensured that $\mathbf{x}_t + \gamma_t(\mathbf{v}_t - \mathbf{x}_t) \in \mathrm{dom}(f)$ and $f(\mathbf{x}_t + \gamma_t(\mathbf{v}_t - \mathbf{x}_t)) \leq f(\mathbf{x}_t)$, in the standard Monotonous Frank-Wolfe algorithm in Algorithm 3. However, the number of zeroth-order or domain oracles that would be needed to find this step size that satisfies both $\mathbf{x}_t + \gamma_t(\mathbf{v}_t - \mathbf{x}_t) \in \mathrm{dom}(f)$ and $f(\mathbf{x}_{t+1}) \leq f(\mathbf{x}_t)$ is logarithmic for the Monotonous Frank-Wolfe variant shown in Algorithm 4, when compared to the number needed for the Monotonous Frank-Wolfe variant with halving shown in Algorithm 3. Note that the convergence properties established throughout the paper for the Monotonous Frank-Wolfe algorithm in Algorithm 3 also hold for the variant in Algorithm 4; with the only difference being that we lose a very small constant factor (e.g., at most a factor of 2 for the standard case) in the convergence rate.

Algorithm 4 Halving Monotonous Frank-Wolfe

Input: Point $\mathbf{x}_0 \in \mathcal{X} \cap \mathrm{dom}(f)$, function f
Output: Iterates $\mathbf{x}_1, \ldots \in \mathcal{X}$

$\psi_{-1} \leftarrow 0$
for $t = 0$ **to** $\ldots$ **do**
  $\mathbf{v}_t \leftarrow \mathrm{argmin}_{\mathbf{v}\in\mathcal{X}} \langle \nabla f(\mathbf{x}_t), \mathbf{v}\rangle$
  $\psi_t \leftarrow \psi_{t-1}$
  $\gamma_t \leftarrow 2^{1-\psi_t}/(t+2)$
  $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t + \gamma_t(\mathbf{v}_t - \mathbf{x}_t)$
  while $\mathbf{x}_{t+1} \notin \mathrm{dom}(f)$ **or** $f(\mathbf{x}_{t+1}) > f(\mathbf{x}_t)$ **do**
    $\psi_t \leftarrow \psi_t + 1$
    $\psi_t \leftarrow 2^{1-\psi_t}/(t+2)$
    $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t + \gamma_t(\mathbf{v}_t - \mathbf{x}_t)$

A.1 Convergence of the Frank-Wolfe gap for the Monotonous Frank-Wolfe algorithm

In this section of the appendix, we show that when running the Monotonous Frank-Wolfe algorithm (Algorithm 1) the minimum of the Frank-Wolfe gap over the run of the algorithm converges at a rate of $O(1/t)$. The idea of the proof is very similar to the one in Jaggi (2013). In a nutshell, as the primal progress per iteration is directly related to the step size times the Frank-Wolfe gap, we know that the Frank-Wolfe gap cannot remain indefinitely above a given value, as otherwise we would obtain a large amount of primal progress, which would make the primal gap become negative. This is formalized in Theorem A.2.

Theorem A.2. *Suppose $\mathcal{X}$ is a compact convex set and f is a (M, ν) generalized self-concordant function with $\nu \geq 2$. Then if the Monotonous Frank-Wolfe algorithm (Algorithm 1) is run for $T \geq T_\nu + 6$ iterations, we will have that:*

$$\min_{1\leq t\leq T} g(\mathbf{x}_t) \leq O(1/T),$$

where T_ν is defined as:

$$T_\nu \stackrel{\text{def}}{=} \begin{cases} \lceil 4MD \rceil - 2 & \text{if } \nu = 2 \\ \left\lceil 2MD(L_f^{\mathcal{L}_0})^{\nu/2-1}(\nu-2) \right\rceil - 2 & \text{otherwise.} \end{cases} \tag{A.1}$$

Proof. In order to prove the claim, we focus on the iterations $T_\nu + \lceil (T - T_\nu)/3 \rceil - 2 \leq t \leq T - 2$, where T_ν is defined in Equation (A.1). Note that as we assume that $T \geq T_\nu + 6$, we know that $T_\nu \leq T_\nu + \lceil (T - T_\nu)/3 \rceil - 2$, and so for iterations $T_\nu + \lceil (T - T_\nu)/3 \rceil - 2 \leq t \leq T - 2$ we know that $d_\nu(\mathbf{x}_t, \mathbf{x}_{t+1}) \leq 1/2$, and so:

$$h(\mathbf{x}_{t+1}) \leq h(\mathbf{x}_t) - \gamma_t g(\mathbf{x}_t) + \gamma_t^2 L_f^{\mathcal{L}_0} D^2 \omega_\nu(1/2). \tag{A.2}$$

In a very similar fashion as was done in the proof of Theorem 2.5, we divide the proof into two different cases.

Case $-\gamma_t g(\mathbf{x}_t) + \gamma_t^2 L_f^{\mathcal{L}_0} D^2 \omega_\nu(1/2) \geq 0$ for some $T_\nu + \lceil (T - T_\nu)/3 \rceil - 2 \leq t \leq T - 2$: Reordering the inequality above we therefore know that there exists a $T_\nu + \lceil (T - T_\nu)/3 \rceil - 2 \leq K \leq T - 2$ such that:

$$
\begin{aligned}
g(\mathbf{x}_K) &\leq \frac{2}{2+K} L_f^{\mathcal{L}_0} D^2 \omega_\nu(1/2) \\
&\leq \frac{2}{T_\nu + \lceil (T - T_\nu)/3 \rceil} L_f^{\mathcal{L}_0} D^2 \omega_\nu(1/2) \\
&= \frac{6}{2T_\nu + T} L_f^{\mathcal{L}_0} D^2 \omega_\nu(1/2),
\end{aligned}
$$

where the second inequality follows from the fact that $T_\nu + \lceil (T - T_\nu)/3 \rceil - 2 \leq K$. This leads to $\min_{1 \leq t \leq T} g(\mathbf{x}_t) \leq g(\mathbf{x}_K) \leq \frac{6}{2T_\nu + T} L_f^{\mathcal{L}_0} D^2 \omega_\nu(1/2)$.

Case $-\gamma_t g(\mathbf{x}_t) + \gamma_t^2 L_f^{\mathcal{L}_0} D^2 \omega_\nu(1/2) < 0$ for all $T_\nu + T_\nu + \lceil (T - T_\nu)/3 \rceil - 2 \leq t \leq T - 2$: Using the inequality above and plugging into Equation (A.2) allows us to conclude that all steps $T_\nu + T_\nu + \lceil (T - T_\nu)/3 \rceil - 2 \leq t \leq T - 2$ will produce primal progress using the step size γ_t, and so as we know that $\mathbf{x}_{t+1} \in \mathrm{dom}(f)$ by Lemma 2.4, then for all $T_\nu + \lceil (T - T_\nu)/3 \rceil - 2 \leq t \leq T - 2$ we will take a non-zero step size determined by γ_t, as $\mathbf{x}_t + \gamma_t(\mathbf{v}_t - \mathbf{x}_t) \in \mathrm{dom}(f)$ and $f(\mathbf{x}_t + \gamma_t(\mathbf{v}_t - \mathbf{x}_t)) < f(\mathbf{x}_t)$ in Line 5 of Algorithm 1. Consequently, summing up Equation (A.2) from $t_{\min} \stackrel{\text{def}}{=} T_\nu + \lceil (T - T_\nu)/3 \rceil - 2$ to $t_{\max} \stackrel{\text{def}}{=} T - 2$ we have that:

$$
\begin{aligned}
h(\mathbf{x}_{t_{\max}+1}) &\leq h\left(\mathbf{x}_{t_{\min}}\right) - \sum_{t=t_{\min}}^{t_{\max}} \gamma_t g(\mathbf{x}_t) + L_f^{\mathcal{L}_0} D^2 \omega_\nu(1/2) \sum_{t=t_{\min}}^{t_{\max}} \gamma_t^2 && \text{(A.3)} \\
&\leq h\left(\mathbf{x}_{t_{\min}}\right) - 2 \min_{t_{\min} \leq t \leq t_{\max}} g(\mathbf{x}_t) \sum_{t=t_{\min}}^{t_{\max}} \frac{1}{2+t} + 4 L_f^{\mathcal{L}_0} D^2 \omega_\nu(1/2) \sum_{t=t_{\min}}^{t_{\max}} \frac{1}{(2+t)^2} && \text{(A.4)} \\
&\leq h\left(\mathbf{x}_{t_{\min}}\right) - 2 \min_{1 \leq t \leq T} g(\mathbf{x}_t) \frac{t_{\max} - t_{\min} + 1}{2 + t_{\max}} + 4 L_f^{\mathcal{L}_0} D^2 \omega_\nu(1/2) \frac{t_{\max} - t_{\min} + 1}{(2 + t_{\min})^2} && \text{(A.5)} \\
&\leq 4 \left(\frac{T_\nu + 1}{t_{\min} + 1} + \frac{t_{\max} - t_{\min} + 1}{(2 + t_{\min})^2} \right) \max\left\{ h(\mathbf{x}_0), L_f^{\mathcal{L}_0} D^2 \omega_\nu(1/2) \right\} && \text{(A.6)} \\
&\quad - 2 \min_{1 \leq t \leq T} g(\mathbf{x}_t) \frac{t_{\max} - t_{\min} + 1}{2 + t_{\max}}. && \text{(A.7)}
\end{aligned}
$$

Note that Equation A.4 stems from the fact that $\min_{t_{\min} \leq t \leq t_{\max}} g(\mathbf{x}_t) \leq g(\mathbf{x}_t)$ for any $t_{\min} \leq t \leq t_{\max}$, and from plugging $\gamma_t = 2/(2+t)$, and Equation A.5 follows from the fact that $-1/(2+t) \leq -1/(2+t_{\max})$ and $1/(2+t) \leq 1/(2+t_{\min})$ for all $t_{\min} \leq t \leq t_{\max}$. The last inequality, Equation (A.6) and (A.7) arises from plugging in the upper bound on the primal gap $h(\mathbf{x}_{t_{\min}})$ from Theorem 2.5 and collecting terms. If we plug in the specific values of $t_{\max}$ and $t_{\min}$ this leads to:

$$
\begin{aligned}
h(\mathbf{x}_{T-2}) &\leq 12 \left(\frac{T_\nu + 1}{2T_\nu + T - 3} + \frac{2T - 2T_\nu + 3}{(2T_\nu + T)^2} \right) \max\left\{ h(\mathbf{x}_0), L_f^{\mathcal{L}_0} D^2 \omega_\nu(1/2) \right\} && \text{(A.8)} \\
&\quad - \frac{2}{3} \min_{1 \leq t \leq T} g(\mathbf{x}_t) \frac{T - T_\nu}{T}. && \text{(A.9)}
\end{aligned}
$$

We establish our claim using proof by contradiction. Assume that:

$$
\min_{1 \leq t \leq T} g(\mathbf{x}_t) > \frac{18T}{T - T_\nu} \left(\frac{T_\nu + 1}{2T_\nu + T - 3} + \frac{2T - 2T_\nu + 3}{(2T_\nu + T)^2} \right) \max\left\{ h(\mathbf{x}_0), L_f^{\mathcal{L}_0} D^2 \omega_\nu(1/2) \right\}.
$$

Then by plugging into the bound in Equation (A.9) we have that $h(\mathbf{x}_{T-2}) < 0$, which is the desired contradiction, as the primal gap cannot be negative. Therefore we must have that:

$$
\min_{1 \leq i \leq T} g(\mathbf{x}_i) \leq \frac{18T}{T - T_\nu} \left(\frac{T_\nu + 1}{2T_\nu + T - 3} + \frac{2T - 2T_\nu + 3}{(2T_\nu + T)^2} \right) \max\left\{ h(\mathbf{x}_0), L_f^{\mathcal{L}_0} D^2 \omega_\nu(1/2) \right\} = O(1/T).
$$

This completes the proof. $\square$

A.2 Improved convergence bounds

We focus on two different settings to obtain improved convergence rates; in the first, we assume that $\mathbf{x}^* \in \operatorname{Int}(\mathcal{X} \cap \operatorname{dom}(f))$ (Section A.2), and in the second we assume that $\mathcal{X}$ is strongly or uniformly convex (Section A.2). In this section we focus on the combination of a slightly modified Frank-Wolfe algorithm with the adaptive line search technique of Pedregosa et al. (2020) (shown for reference in Algorithm 5 and 6). This is the same algorithm used in Dvurechensky et al. (2020b), however, we show improved convergence rates in several settings of interest.

Algorithm 5 Monotonous Frank-Wolfe with the adaptive step size of Pedregosa et al. (2020)

Input: Point $\mathbf{x}_0 \in \mathcal{X} \cap \operatorname{dom}(f)$, function f, initial smoothness estimate L_{-1}
Output: Iterates $\mathbf{x}_1, \ldots \in \mathcal{X}$

1: **for** $t = 0$ **to** ... **do**
2: $\quad \mathbf{v}_t \leftarrow \operatorname{argmin}_{\mathbf{v}\in\mathcal{X}} \langle \nabla f(\mathbf{x}_t), \mathbf{v}\rangle$
3: $\quad \gamma_t, L_t \leftarrow \texttt{Backtrack}(f, \mathbf{x}_t, \mathbf{v}_t - \mathbf{x}_t, L_{t-1}, 1)$
4: $\quad \mathbf{x}_{t+1} \leftarrow \mathbf{x}_t + \gamma_t(\mathbf{v}_t - \mathbf{x}_t)$

Algorithm 6 $\texttt{Backtrack}(f, \mathbf{x}, \mathbf{d}, L_{t-1}, \gamma_{\max})$ (line search of Pedregosa et al. (2020))

Input: Point $\mathbf{x} \in \mathcal{X} \cap \operatorname{dom}(f)$, $\mathbf{v} \in \mathbb{R}^n$, function f, estimate L_{t-1}, step size $\gamma_{\max}$
Output: γ, M

1: Choose $\tau > 1$, $\eta \leq 1$ and $M \in [\eta L_{t-1}, L_{t-1}]$
2: $\gamma = \min\{-\langle \nabla f(\mathbf{x}), \mathbf{d}\rangle / (M \|\mathbf{d}\|^2), \gamma_{\max}\}$
3: **while** $\mathbf{x} + \gamma\mathbf{d} \notin \operatorname{dom}(f)$ **or** $f(\mathbf{x}+\gamma\mathbf{d}) - f(\mathbf{x}) > \frac{M\gamma^2}{2}\|\mathbf{d}\|^2 + \gamma\langle\nabla f(\mathbf{x}), \mathbf{d}\rangle$ **do**
4: $\quad M = \tau M$
5: $\quad \gamma = \min\{-\langle \nabla f(\mathbf{x}), \mathbf{d}\rangle / (M \|\mathbf{d}\|^2), \gamma_{\max}\}$

Optimum contained in the interior

Before proving the main theoretical results of this section we first review some auxiliary results that allow us to prove the linear convergence in this setting.

Theorem A.3. *(C.f., (Nesterov, 2018, Theorem 5.1.6)) Let f be generalized self-concordant and $\operatorname{dom}(f)$ not contain straight lines, then the Hessian $\nabla^2 f(\mathbf{x})$ is non-degenerate at all points $\mathbf{x} \in \operatorname{dom}(f)$.*

Note that the assumption that $\operatorname{dom}(f)$ does not contain straight lines is without loss of generality as we can simply modify the function outside of our compact convex feasible region so that it holds.

Proposition A.4. *(C.f., Guélat & Marcotte (1986)) If there exists an $r > 0$ such that $\mathcal{B}(\mathbf{x}^*, r) \subseteq \mathcal{X} \cap \operatorname{dom}(f)$, then for all $\mathbf{x} \in \mathcal{X} \cap \operatorname{dom}(f)$ we have that:*

$$\frac{g(\mathbf{x})}{\|\mathbf{x}-\mathbf{v}\|} \geq \frac{r}{D}\|\nabla f(\mathbf{x})\| \geq \frac{r}{D}\frac{\langle \nabla f(\mathbf{x}), \mathbf{x}-\mathbf{x}^*\rangle}{\|\mathbf{x}-\mathbf{x}^*\|},$$

where $\mathbf{v} = \operatorname{argmin}_{\mathbf{y}\in\mathcal{X}} \langle \nabla f(\mathbf{x}), \mathbf{y}\rangle$ and $g(\mathbf{x})$ is the Frank-Wolfe gap.

With these tools at hand, we have that the Frank-Wolfe algorithm with the backtracking step size strategy converges at a linear rate.

Theorem A.5. *Let f be a (M, ν) generalized self-concordant function with $\nu \geq 2$ and let $\operatorname{dom}(f)$ not contain straight lines. Furthermore, we denote by $r > 0$ the largest value such that $\mathcal{B}(\mathbf{x}^*, r) \subseteq \mathcal{X} \cap \operatorname{dom}(f)$. Then the Frank-Wolfe algorithm (Algorithm 5) with the backtracking strategy of Pedregosa et al. (2020) results in a convergence:*

$$h(\mathbf{x}_t) \leq h(\mathbf{x}_0)\left(1 - \frac{\mu_f^{\mathcal{L}_0}}{2\tilde{L}}\left(\frac{r}{D}\right)^2\right)^t,$$

for $t \geq 1$, where $\tilde{L} \stackrel{\text{def}}{=} \max\{\tau L_f^{\mathcal{L}_0}, L_{-1}\}$, $\tau > 1$ is the backtracking parameter, L_{-1} is the initial smoothness estimate in Algorithm 6, $\mu_f^{\mathcal{L}_0} = \min_{\mathbf{u}\in\mathcal{L}_0,\mathbf{d}\in\mathbb{R}^n} \|\mathbf{d}\|^2_{\nabla^2 f(\mathbf{u})} / \|\mathbf{d}\|_2^2$ and $L_f^{\mathcal{L}_0} = \max_{\mathbf{u}\in\mathcal{L}_0,\mathbf{d}\in\mathbb{R}^n} \|\mathbf{d}\|^2_{\nabla^2 f(\mathbf{u})} / \|\mathbf{d}\|_2^2$.

Proof. Consider the compact set $\mathcal{L}_0 \stackrel{\text{def}}{=} \{\mathbf{x} \in \operatorname{dom}(f) \cap \mathcal{X} \mid f(\mathbf{x}) \leq f(\mathbf{x}_0)\}$. As the backtracking line search makes monotonous primal progress, we know that for $t \geq 0$ we will have that $\mathbf{x}_t \in \mathcal{L}_0$. Consequently we can define $\mu_f^{\mathcal{L}_0} = \min_{\mathbf{u}\in\mathcal{L}_0,\mathbf{d}\in\mathbb{R}^n} \|\mathbf{d}\|^2_{\nabla^2 f(\mathbf{u})} / \|\mathbf{d}\|_2^2$. As by our assumption $\operatorname{dom}(f)$ does not contain any straight lines we know that for all $\mathbf{x} \in \operatorname{dom}(f)$ the Hessian is non-degenerate, and therefore $\mu_f^{\mathcal{L}_0} > 0$. This allows us to claim that for any $\mathbf{x}, \mathbf{y} \in \mathcal{L}_0$ we have that:

$$f(\mathbf{x}) - f(\mathbf{y}) - \langle \nabla f(\mathbf{y}), \mathbf{x} - \mathbf{y} \rangle \geq \frac{\mu_f^{\mathcal{L}_0}}{2} \|\mathbf{x} - \mathbf{y}\|^2 . \tag{A.10}$$

The backtracking line search in Algorithm 6 will either output a point $\gamma_t = 1$ or $\gamma_t < 1$. In any case, Algorithm 6 will find and output a smoothness estimate L_t and a step size γ_t such that for $\mathbf{x}_{t+1} = \mathbf{x}_t + \gamma_t(\mathbf{v}_t - \mathbf{x}_t)$ we have that:

$$f(\mathbf{x}_{t+1}) - f(\mathbf{x}_t) \leq \frac{L_t \gamma_t^2}{2} \|\mathbf{x}_t - \mathbf{v}_t\|^2 - \gamma_t g(\mathbf{x}_t). \tag{A.11}$$

In the case where $\gamma_t = 1$ we know by observing Line 5 of Algorithm 6 that $g(\mathbf{x}_t) \geq L_t \|\mathbf{x}_t - \mathbf{v}_t\|^2$, and so plugging into Equation (A.11) we arrive at $h(\mathbf{x}_{t+1}) \leq h(\mathbf{x}_t)/2$. In the case where $\gamma_t = g(\mathbf{x}_t)/(L_t \|\mathbf{x}_t - \mathbf{v}_t\|^2) < 1$, we have that $g(\mathbf{x}_t) < L_t \|\mathbf{x}_t - \mathbf{v}_t\|^2$, which leads to $h(\mathbf{x}_{t+1}) \leq h(\mathbf{x}_k) - g(\mathbf{x}_t)^2/(2L_t \|\mathbf{x}_t - \mathbf{v}_t\|^2)$, when plugging the expression for the step size in the progress bound in Equation A.11. In this last case where $\gamma_t < 1$ we have the following contraction for the primal gap:

$$\begin{aligned} h(\mathbf{x}_t) - h(\mathbf{x}_{t+1}) &\geq \frac{g(\mathbf{x}_t)^2}{2L_t \|\mathbf{x}_t - \mathbf{v}_t\|^2} \\ &\geq \frac{r^2}{D^2} \frac{\|\nabla f(\mathbf{x}_t)\|^2}{2L_t} \\ &\geq \frac{\mu_f^{\mathcal{L}_0}}{\tilde{L}} \frac{r^2}{D^2} h(\mathbf{x}_t), \end{aligned}$$

where we have used the inequality that involves the central term and the leftmost term in Proposition A.4, and the last inequality stems from the bound $h(\mathbf{x}_t) \leq \|\nabla f(\mathbf{x}_t)\|^2 / (2\mu_f^{\mathcal{L}_0})$ for $\mu_f^{\mathcal{L}_0}$-strongly convex functions. Putting the above bounds together we have that:

$$\begin{aligned} h(\mathbf{x}_{t+1}) &\leq h(\mathbf{x}_t) \left(1 - \frac{1}{2} \min\left\{1, \frac{\mu_f^{\mathcal{L}_0}}{\tilde{L}} \left(\frac{r}{D}\right)^2\right\}\right) \\ &\leq h(\mathbf{x}_t) \left(1 - \frac{\mu_f^{\mathcal{L}_0}}{2\tilde{L}} \left(\frac{r}{D}\right)^2\right), \end{aligned}$$

which completes the proof. □

The previous bound depends on the largest positive r such that $\mathcal{B}(\mathbf{x}^*, r) \subseteq \mathcal{X} \cap \operatorname{dom}(f)$, which can be arbitrarily small. Note also that the previous proof uses the lower bound of the Bregman divergence from the $\mu_f^{\mathcal{L}_0}$-strong convexity of the function over $\mathcal{L}_0$ to obtain linear convergence. Note that this bound is local, and is only of use because the step size strategy of Algorithm 6 automatically ensures that if $\mathbf{x}_t \in \mathcal{L}_0$ and $\mathbf{d}_t$ is a direction of descent, then $\mathbf{x}_t + \gamma_t \mathbf{d}_t \in \mathcal{L}_0$. This is in contrast with Algorithm 3, in which the step size $\gamma_t = 2/(2+t)$ did not automatically ensure monotonicity in primal gap, and this had to be enforced by setting $\mathbf{x}_{t+1} = \mathbf{x}_t$ if $f(\mathbf{x}_t + \gamma_t \mathbf{d}_t) > f(\mathbf{x}_t)$, where $\mathbf{d}_t = \mathbf{v}_t - \mathbf{x}_t$. If we were to have used the lower bound on the Bregman divergence from Sun & Tran-Dinh (2019, Proposition 10) in the proof, which states that:

$$f(\mathbf{y}) - f(\mathbf{x}) - \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle \geq \omega_\nu(-d_\nu(\mathbf{x} - \mathbf{y})) \|\mathbf{y} - \mathbf{x}\|^2_{\nabla^2 f(\mathbf{x})},$$

for any $\mathbf{x}, \mathbf{y} \in \mathrm{dom}(f)$ and any $\nu \geq 2$, we would have arrived at a bound that holds over all $\mathrm{dom}(f)$. However, in order to arrive at a usable bound, and armed only with the knowledge that the Hessian is non-degenerate if $\mathrm{dom}(f)$ does not contain straight lines, and that $\mathbf{x}, \mathbf{y} \in \mathcal{L}_0$, we would have had to write:

$$\omega_\nu(-d_\nu(\mathbf{x} - \mathbf{y})) \|\mathbf{x} - \mathbf{y}\|^2_{\nabla^2 f(\mathbf{y})} \geq \mu_f^{\mathcal{L}_0} \omega_\nu(-d_\nu(\mathbf{x} - \mathbf{y})) \|\mathbf{x} - \mathbf{y}\|^2,$$

where the inequality follows from the definition of $\mu_f^{\mathcal{L}_0}$. It is easy to see that as $d\omega_\nu(\tau)/d\tau > 0$ by Remark 2.3, we have that $1/2 = \omega_\nu(0) \geq \omega_\nu(-d_\nu(\mathbf{x} - \mathbf{y}))$. This results in a bound:

$$f(\mathbf{y}) - f(\mathbf{x}) - \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x} \rangle \geq \mu_f^{\mathcal{L}_0} \omega_\nu(-d_\nu(\mathbf{x} - \mathbf{y})) \|\mathbf{x} - \mathbf{y}\|^2. \tag{A.12}$$

When we compare the bounds on Equation (A.10) and (A.12), we can see that the bound from $\mu_f^{\mathcal{L}_0}$-strong convexity is tighter than the bound from the properties of (M, ν)-generalized self-concordant functions, albeit local. This is the reason why we have used the former bound in the proof of Theorem A.5.

Strongly convex or uniformly convex sets

In order to prove convergence rate results for the case where the feasible region is (κ, p)-uniformly convex, we first review the definition of the (κ, p)-uniform convexity of a set (see Definition A.6), as well as a useful lemma that allows us to go from contractions to convergence rates.

Definition A.6 ((κ, q)-uniformly convex set)**.** Given two positive numbers κ and q, we say the set $\mathcal{X} \subseteq \mathbb{R}^n$ is (κ, q)-*uniformly convex* with respect to a norm $\|\cdot\|$ if for any $\mathbf{x}, \mathbf{y} \in \mathcal{X}$, $0 \leq \gamma \leq 1$, and $\mathbf{z} \in \mathbb{R}^n$ with $\|\mathbf{z}\| = 1$ we have that:

$$\mathbf{y} + \gamma(\mathbf{x} - \mathbf{y}) + \gamma(1 - \gamma) \cdot \kappa \|\mathbf{x} - \mathbf{y}\|^q \mathbf{z} \in \mathcal{X}.$$

The previous definition allows us to obtain a scaling inequality very similar to the one shown in Theorem A.4, which is key to proving the following convergence rates, and can be implicitly found in Kerdreux et al. (2021) and Garber & Hazan (2016).

Proposition A.7. *Let* $\mathcal{X} \subseteq \mathbb{R}^n$ *be* (κ, q)-uniformly convex*, then for all* $\mathbf{x} \in \mathcal{X}$*:*

$$\frac{g(\mathbf{x})}{\|\mathbf{x} - \mathbf{v}\|^q} \geq \kappa \|\nabla f(\mathbf{x})\|,$$

where $\mathbf{v} = \mathrm{argmin}_{\mathbf{u} \in \mathcal{X}} \langle \nabla f(\mathbf{x}), \mathbf{u} \rangle$*, and* $g(\mathbf{x})$ *is the Frank-Wolfe gap.*

The next lemma that will be presented is an extension of the one used in Kerdreux et al. (2021, Lemma A.1) (see also Temlyakov (2015)), and allows us to go from per iteration contractions to convergence rates.

Lemma A.8. *We denote a sequence of nonnegative numbers by* $\{h_t\}_t$*. Let* c_0*,* c_1*,* c_2 *and* α *be positive numbers such that* $c_1 < 1$*,* $h_1 \leq c_0$ *and* $h_t - h_{t+1} \geq h_t \min\{c_1, c_2 h_t^\alpha\}$ *for* $t \geq 1$*, then:*

$$h_t \leq \begin{cases} c_0 (1 - c_1)^{t-1} & \text{if } 1 \leq t \leq t_0 \\ \frac{(c_1/c_2)^{1/\alpha}}{(1 + c_1 \alpha (t - t_0))^{1/\alpha}} = O\left(t^{-1/\alpha}\right) & \text{otherwise.} \end{cases}$$

where

$$t_0 \stackrel{\text{def}}{=} \max\left\{1, \left\lfloor \log_{1-c_1}\left(\frac{(c_1/c_2)^{1/\alpha}}{c_0}\right) \right\rfloor \right\}.$$

This allows us to conveniently transform the per iteration contractions to convergence rates. Moving on to the proof of the convergence rate.

Theorem A.9. *Suppose $\mathcal{X}$ is a compact (κ, q)-uniformly convex set and f is a (M, ν) generalized self-concordant function with $\nu \geq 2$. Furthermore, assume that $\min_{\mathbf{x}\in\mathcal{X}} \|\nabla f(\mathbf{x})\| \geq C$. Then the Frank-Wolfe algorithm with Backtrack (Algorithm 5) results in a convergence:*

$$h_t \leq \begin{cases} h(\mathbf{x}_0)\left(1 - \frac{1}{2}\min\left\{1, \frac{\kappa C}{\tilde{L}}\right\}\right)^t & \text{if } q = 2 \\ \frac{h(\mathbf{x}_0)}{2^t} & \text{if } q > 2, 1 \leq t \leq t_0 \\ \frac{(\tilde{L}^q/(\kappa C)^2)^{1/(q-2)}}{(1+(q-2)(t-t_0)/(2q))^{q/(q-2)}} = \mathcal{O}\left(t^{-q/(q-2)}\right) & \text{if } q > 2, t > t_0, \end{cases}$$

for $t \geq 1$, where:

$$t_0 = \max\left\{1, \left\lfloor \log_{1/2}\left(\frac{(\tilde{L}^q/(\kappa C)^2)^{1/(q-2)}}{h(\mathbf{x}_0)}\right)\right\rfloor\right\}.$$

and $\tilde{L} \stackrel{\text{def}}{=} \max\{\tau L_f^{\mathcal{L}_0}, L_{-1}\}$, where $\tau > 1$ is the backtracking parameter, L_{-1} is the initial smoothness estimate in Algorithm 6, and $L_f^{\mathcal{L}_0} = \max_{\mathbf{u}\in\mathcal{L}_0, \mathbf{d}\in\mathbb{R}^n} \|\mathbf{d}\|^2_{\nabla^2 f(\mathbf{u})} / \|\mathbf{d}\|_2^2$.

Proof. At iteration t, the backtracking line search strategy finds through successive function evaluations a $L_t > 0$ such that:

$$h(\mathbf{x}_{t+1}) \leq h(\mathbf{x}_t) - \gamma_t g(\mathbf{x}_t) + \frac{L_t \gamma_t^2}{2} \|\mathbf{x}_t - \mathbf{v}_t\|^2 .$$

Finding the γ_t that maximizes the right-hand side of the previous inequality leads to $\gamma_t = \min\{1, g(\mathbf{x}_t)/(L_t \|\mathbf{x}_t - \mathbf{v}_t\|^2)\}$, which is the step size ultimately taken by the algorithm at iteration t. Note that if $\gamma_t = 1$ this means that $g(\mathbf{x}_t) \geq L_t \|\mathbf{x}_t - \mathbf{v}_t\|^2$, which when plugged into the inequality above leads to $h(\mathbf{x}_{t+1}) \leq h(\mathbf{x}_t)/2$. Conversely, for $\gamma_t < 1$ we have that $h(\mathbf{x}_{t+1}) \leq h(\mathbf{x}_t) - g(\mathbf{x}_t)^2/(2L_t \|\mathbf{x}_t - \mathbf{v}_t\|^2)$. Focusing on this case and using the bounds $g(\mathbf{x}_t) \geq h(\mathbf{x}_t)$ and $g(\mathbf{x}_t) \geq \kappa \|\nabla f(\mathbf{x}_t)\| \|\mathbf{x}_t - \mathbf{v}_t\|^q$ from Proposition A.7 leads to:

$$h(\mathbf{x}_{t+1}) \leq h(\mathbf{x}_t) - h(\mathbf{x}_t)^{2-2/q} \frac{(\kappa \|\nabla f(\mathbf{x}_t)\|)^{2/q}}{2L_t} \tag{A.13}$$

$$\leq h(\mathbf{x}_t) - h(\mathbf{x}_t)^{2-2/q} \frac{(\kappa C)^{2/q}}{2\tilde{L}}, \tag{A.14}$$

where the last inequality simply comes from the bound on the gradient norm, and the fact that $L_t \leq \tilde{L}$, for $\tilde{L} \stackrel{\text{def}}{=} \max\{\tau L_f^{\mathcal{L}_0}, L_{-1}\}$, where $\tau > 1$ is the backtracking parameter and L_{-1} is the initial smoothness estimate in Algorithm 6. Reordering this expression and putting together the two cases we have that:

$$h(\mathbf{x}_t) - h(\mathbf{x}_{t+1}) \geq h(\mathbf{x}_t) \min\left\{\frac{1}{2}, \frac{(\kappa C)^{2/q}}{2\tilde{L}} h(\mathbf{x}_t)^{1-2/q}\right\}.$$

For the case where $q = 2$ we get a linear contraction in primal gap. Using Lemma A.8 to go from a contraction to a convergence rate for $q > 2$ we have that:

$$h_t \leq \begin{cases} h(\mathbf{x}_0)\left(1 - \frac{1}{2}\min\left\{1, \frac{\kappa C}{\tilde{L}}\right\}\right)^t & \text{if } q = 2 \\ \frac{h(\mathbf{x}_0)}{2^t} & \text{if } q > 2, 1 \leq t \leq t_0 \\ \frac{(\tilde{L}^q/(\kappa C)^2)^{1/(q-2)}}{(1+(q-2)(t-t_0)/(2q))^{q/(q-2)}} = \mathcal{O}\left(t^{-q/(q-2)}\right) & \text{if } q > 2, t > t_0, \end{cases}$$

for $t \geq 1$, where:

$$t_0 = \max\left\{1, \left\lfloor \log_{1/2}\left(\frac{(\tilde{L}^q/(\kappa C)^2)^{1/(q-2)}}{h(\mathbf{x}_0)}\right)\right\rfloor\right\},$$

which completes the proof. □

Lastly, we deal with the general case in which the norm of the gradient is not bounded away from zero in $\mathcal{X}$.

Theorem A.10. *Suppose $\mathcal{X}$ is a compact (κ, q)-uniformly convex set and f is a (M, ν) generalized self-concordant function with $\nu \geq 2$ for which domain does not contain straight lines. Then the Frank-Wolfe algorithm with Backtrack (Algorithm 5) results in a convergence:*

$$h_t \leq \begin{cases} \frac{h(\mathbf{x}_0)}{2^t} & \text{if } 1 \leq t \leq t_0 \\ \frac{(\tilde{L}^q/(\kappa^2 \mu_f^{\mathcal{L}_0}))^{1/(q-1)}}{(1+(q-1)(t-t_0)/(2q))^{q/(q-1)}} = \mathcal{O}\left(t^{-q/(q-1)}\right) & \text{if } t > t_0, \end{cases}$$

for $t \geq 1$, where:

$$t_0 = \max\left\{1, \left\lfloor \log_{1/2}\left(\frac{(\tilde{L}^q/(\kappa^2 \mu_f^{\mathcal{L}_0}))^{1/(q-1)}}{h(\mathbf{x}_0)}\right)\right\rfloor\right\}.$$

and $\tilde{L} \stackrel{\text{def}}{=} \max\{\tau L_f^{\mathcal{L}_0}, L_{-1}\}$, where $\tau > 1$ is the backtracking parameter, L_{-1} is the initial smoothness estimate in Algorithm 6, $\mu_f^{\mathcal{L}_0} = \min_{\mathbf{u}\in\mathcal{L}_0, \mathbf{d}\in\mathbb{R}^n} \|\mathbf{d}\|^2_{\nabla^2 f(\mathbf{u})} / \|\mathbf{d}\|_2^2$ and $L_f^{\mathcal{L}_0} = \max_{\mathbf{u}\in\mathcal{L}_0, \mathbf{d}\in\mathbb{R}^n} \|\mathbf{d}\|^2_{\nabla^2 f(\mathbf{u})} / \|\mathbf{d}\|_2^2$.

Proof. Consider the compact set $\mathcal{L}_0 \stackrel{\text{def}}{=} \{\mathbf{x} \in \mathrm{dom}(f) \cap \mathcal{X} \mid f(\mathbf{x}) \leq f(\mathbf{x}_0)\}$. As the algorithm makes monotonous primal progress we have that $\mathbf{x}_t \in \mathcal{L}_0$ for $t \geq 0$. The proof proceeds very similarly as before, except for the fact that now we have to bound $\|\nabla f(\mathbf{x}_t)\|$ using $\mu_f^{\mathcal{L}_0}$-strong convexity for points $\mathbf{x}_t, \mathbf{x}_t + \gamma_t(\mathbf{v}_t - \mathbf{x}_t) \in \mathcal{L}_0$. Continuing from Equation (A.13) for the case where $\gamma_t < 1$ and using the fact that $h(\mathbf{x}_t) \leq \|\nabla f(\mathbf{x}_t)\|^2 / (2\mu_f^{\mathcal{L}_0})$ we have that:

$$\begin{aligned} h(\mathbf{x}_{t+1}) &\leq h(\mathbf{x}_t) - h(\mathbf{x}_t)^{2-2/q} \frac{(\kappa \|\nabla f(\mathbf{x}_t)\|)^{2/q}}{2L_t} \\ &\leq h(\mathbf{x}_t) - h(\mathbf{x}_t)^{2-1/q} \frac{\kappa^{2/q} (\mu_f^{\mathcal{L}_0})^{1/q} 2^{1/q-1}}{\tilde{L}}, \end{aligned}$$

where we have also used the bound $L_t \leq \tilde{L}$ in the last equation. This leads us to a contraction, together with the case where $\gamma_t = 1$, which is unchanged from the previous proofs, of the form:

$$h(\mathbf{x}_t) - h(\mathbf{x}_{t+1}) \geq h(\mathbf{x}_t) \min\left\{\frac{1}{2}, \frac{\kappa^{2/q} (\mu_f^{\mathcal{L}_0})^{1/q} 2^{1/q-1}}{\tilde{L}} h(\mathbf{x}_t)^{1-1/q}\right\}.$$

Using again Lemma A.8 to go from a contraction to a convergence rate for $q > 2$ we have that:

$$h_t \leq \begin{cases} \frac{h(\mathbf{x}_0)}{2^t} & \text{if } 1 \leq t \leq t_0 \\ \frac{(\tilde{L}^q/(\kappa^2 \mu_f^{\mathcal{L}_0}))^{1/(q-1)}}{(1+(q-1)(t-t_0)/(2q))^{q/(q-1)}} = \mathcal{O}\left(t^{-q/(q-1)}\right) & \text{if } t > t_0, \end{cases}$$

for $t \geq 1$, where:

$$t_0 = \max\left\{1, \left\lfloor \log_{1/2}\left(\frac{(\tilde{L}^q/(\kappa^2 \mu_f^{\mathcal{L}_0}))^{1/(q-1)}}{h(\mathbf{x}_0)}\right)\right\rfloor\right\},$$

which completes the proof. □

In Table 3 we provide an oracle complexity breakdown for the Frank-Wolfe algorithm with Backtrack (`B-FW`), also referred to as `LBTFW-GSC` in Dvurechensky et al. (2020b), when minimizing over a (κ, q)-uniformly convex set.

Algorithm	Assumptions	FOO/ZOO/LMO/DO calls	Reference
B-FW/LBTFW-GSC‡	$\mathbf{x}^* \in \mathrm{Int}\,(\mathcal{X} \cap \mathrm{dom}(f))$	$\mathcal{O}(\log 1/\varepsilon)$	**This work**
B-FW/LBTFW-GSC‡	$\min_{\mathbf{x}\in\mathcal{X}} \\|\nabla f(\mathbf{x})\\| > 0$, $q = 2$	$\mathcal{O}(\log 1/\varepsilon)$	**This work**
B-FW/LBTFW-GSC‡	$\min_{\mathbf{x}\in\mathcal{X}} \\|\nabla f(\mathbf{x})\\| > 0$, $q > 2$	$\mathcal{O}\left(\varepsilon^{-(q-2)/q}\right)$	**This work**
B-FW/LBTFW-GSC‡	No straight lines in $\mathcal{X}$	$\mathcal{O}\left(\varepsilon^{-(q-1)/q}\right)$	**This work**

Table 3: **Complexity comparison for B-FW (Algorithm 2) when minimizing over a (κ, q)-uniformly convex set**: Number of iterations needed to reach an ε-optimal solution in $h(\mathbf{x})$ for Problem 1.1 in several cases of interest. We use the superscript ‡ to indicate that constants in the convergence bounds depend on user-defined inputs.

B Away-step Frank-Wolfe

When the domain $\mathcal{X}$ is a polytope, one can obtain linear convergence in primal gap for a generalized self-concordant function using the well known *Away-step Frank-Wolfe* (AFW) algorithm (Guélat & Marcotte, 1986; Lacoste-Julien & Jaggi, 2015) with the adaptive step size of Pedregosa et al. (2020), shown in Algorithm 7. We use $\mathcal{S}_t$ to denote the *active set* at iteration t, that is, the set of vertices of the polytope that gives rise to $\mathbf{x}_t$ as a convex combination with positive weights. We can see that the algorithm either chooses to perform what is know as a *Frank-Wolfe* step in Line 6 of Algorithm 7 if the Frank-Wolfe gap $g(\mathbf{x})$ is greater than the *away gap* $\langle \nabla f(\mathbf{x}_t), \mathbf{a}_t - \mathbf{x}_t \rangle$ or an *Away*-step in Line 8 of Algorithm 7 otherwise.

Algorithm 7 Away-step Frank-Wolfe (B-AFW) with the step size of Pedregosa et al. (2020)

Input: Point $\mathbf{x}_0 \in \mathcal{X} \cap \mathrm{dom}(f)$, function f, initial smoothness estimate L_{-1}
Output: Iterates $\mathbf{x}_1, \ldots \in \mathcal{X}$

1: $\mathcal{S}_0 \leftarrow \{\mathbf{x}_0\}$, $\boldsymbol{\lambda}_0 \leftarrow \{1\}$
2: **for** $t = 0$ **to** $\ldots$ **do**
3: $\quad \mathbf{v}_t \leftarrow \mathrm{argmin}_{\mathbf{v}\in\mathcal{X}} \langle \nabla f(\mathbf{x}_t), \mathbf{v} \rangle$
4: $\quad \mathbf{a}_t \leftarrow \mathrm{argmax}_{\mathbf{v}\in\mathcal{S}_t} \langle \nabla f(\mathbf{x}_t), \mathbf{v} \rangle$
5: $\quad$ **if** $\langle \nabla f(\mathbf{x}_t), \mathbf{x}_t - \mathbf{v}_t \rangle \geq \langle \nabla f(\mathbf{x}_t), \mathbf{a}_t - \mathbf{x}_t \rangle$ **then**
6: $\qquad \mathbf{d}_t \leftarrow \mathbf{v}_t - \mathbf{x}_t$, $\gamma_{\max} \leftarrow 1$
7: $\quad$ **else**
8: $\qquad \mathbf{d}_t \leftarrow \mathbf{x}_t - \mathbf{a}_t$, $\gamma_{\max} \leftarrow \boldsymbol{\lambda}_t(\mathbf{a}_t)/(1 - \boldsymbol{\lambda}_t(\mathbf{a}_t))$
9: $\quad \gamma_t, L_t \leftarrow \texttt{Backtrack}(f, \mathbf{x}_t, \mathbf{d}_t, \nabla f(\mathbf{x}_t), L_{t-1}, \gamma_{\max})$
10: $\quad \mathbf{x}_{t+1} \leftarrow \mathbf{x}_t + \gamma_t \mathbf{d}_t$
11: $\quad$ Update $\mathcal{S}_t$ and $\boldsymbol{\lambda}_t$ to $\mathcal{S}_{t+1}$ and $\boldsymbol{\lambda}_{t+1}$

The proof of linear convergence follows closely from Pedregosa et al. (2020) and Lacoste-Julien & Jaggi (2015), with the only difference that we need to take into consideration that the function is generalized self-concordant as opposed to smooth and strongly convex. One of the key inequalities used in the proof is a scaling inequality from Lacoste-Julien & Jaggi (2015) very similar to the one shown in Proposition A.4 and Proposition A.7, which we state next:

Proposition B.1. *Let $\mathcal{X} \subseteq \mathbb{R}^n$ be a polytope, and denote by $\mathcal{S}$ the set of vertices of the polytope $\mathcal{X}$ that gives rise to $\mathbf{x} \in \mathcal{X}$ as a convex combination with positive weights, then for all $\mathbf{y} \in \mathcal{X}$:*

$$\langle \nabla f(\mathbf{x}), \mathbf{a} - \mathbf{v} \rangle \geq \delta \frac{\langle \nabla f(\mathbf{x}), \mathbf{x} - \mathbf{y} \rangle}{\|\mathbf{x} - \mathbf{y}\|},$$

where $\mathbf{v} = \mathrm{argmin}_{\mathbf{u}\in\mathcal{X}} \langle \nabla f(\mathbf{x}), \mathbf{u} \rangle$, $\mathbf{a} = \mathrm{argmax}_{\mathbf{u}\in\mathcal{S}} \langle \nabla f(\mathbf{x}), \mathbf{u} \rangle$, and $\delta > 0$ is the pyramidal width *of $\mathcal{X}$.*

Theorem B.2. *Suppose $\mathcal{X}$ is a polytope and f is a (M, ν) generalized self-concordant function with $\nu \geq 2$ for which the domain does not contain straight lines. Then the Away-step Frank-Wolfe*

(AFW) algorithm with Backtrack (Algorithm 7) results in a convergence:

$$h(\mathbf{x}_t) \leq h(\mathbf{x}_0)\left(1 - \frac{\mu_f^{\mathcal{L}_0}}{4\tilde{L}}\left(\frac{\delta}{D}\right)^2\right)^{\lceil (t-1)/2 \rceil},$$

where δ is the pyramidal width of the polytope $\mathcal{X}$, $\tilde{L} \stackrel{\text{def}}{=} \max\{\tau L_f^{\mathcal{L}_0}, L_{-1}\}$, $\tau > 1$ is the backtracking parameter, L_{-1} is the initial smoothness estimate in Algorithm 6, $\mu_f^{\mathcal{L}_0} = \min_{\mathbf{u}\in\mathcal{L}_0, \mathbf{d}\in\mathbb{R}^n} \|\mathbf{d}\|^2_{\nabla^2 f(\mathbf{u})} / \|\mathbf{d}\|_2^2$ and $L_f^{\mathcal{L}_0} = \max_{\mathbf{u}\in\mathcal{L}_0, \mathbf{d}\in\mathbb{R}^n} \|\mathbf{d}\|^2_{\nabla^2 f(\mathbf{u})} / \|\mathbf{d}\|_2^2$.

Proof. Proceeding very similarly as in the proof of Theorem A.5, we have that as the backtracking line search makes monotonous primal progress, we know that for $t \geq 0$ we will have that $\mathbf{x}_t \in \mathcal{L}_0$. As the function is $\mu_f^{\mathcal{L}_0}$-strongly convex over $\mathcal{L}_0$, we can use the appropriate inequalities from strong convexity in the progress bounds. Using this aforementioned property, together with the scaling inequality of Proposition B.1 results in:

$$\begin{aligned}
f(\mathbf{x}_t) - f(\mathbf{x}^*) &\leq \frac{\langle \nabla f(\mathbf{x}_t), \mathbf{x}_t - \mathbf{x}^*\rangle}{2\mu_f^{\mathcal{L}_0} \|\mathbf{x}_t - \mathbf{x}^*\|^2} && \text{(B.1)}\\
&\leq \frac{\langle \nabla f(\mathbf{x}_t), \mathbf{a}_t - \mathbf{v}_t\rangle^2}{2\mu_f^{\mathcal{L}_0}\delta^2} && \text{(B.2)}\\
&= \frac{(\langle \nabla f(\mathbf{x}_t), \mathbf{a}_t - \mathbf{x}_t\rangle + \langle \nabla f(\mathbf{x}_t), \mathbf{x}_t - \mathbf{v}_t\rangle)^2}{2\mu_f^{\mathcal{L}_0}\delta^2}, && \text{(B.3)}
\end{aligned}$$

where the first inequality comes from the $\mu_f^{\mathcal{L}_0}$-strong convexity over $\mathcal{L}_0$, and the second inequality comes from applying Proposition B.1 with $\mathbf{y} = \mathbf{x}^*$. Note that if the Frank-Wolfe step is chosen in Line 6, then $-\langle \nabla f(\mathbf{x}_t), \mathbf{d}_t\rangle = \langle \nabla f(\mathbf{x}_t), \mathbf{x}_t - \mathbf{v}_t\rangle \geq \langle \nabla f(\mathbf{x}_t), \mathbf{a}_t - \mathbf{x}_t\rangle$, otherwise, if an away step is chosen in Line 8, then $\langle \nabla f(\mathbf{x}_t), \mathbf{x}_t - \mathbf{v}_t\rangle < \langle \nabla f(\mathbf{x}_t), \mathbf{a}_t - \mathbf{x}_t\rangle = -\langle \nabla f(\mathbf{x}_t), \mathbf{d}_t\rangle$, in any case, plugging into Equation (B.3) we have that:

$$h(\mathbf{x}_t) = f(\mathbf{x}_t) - f(\mathbf{x}^*) \leq \frac{2\langle \nabla f(\mathbf{x}_t), \mathbf{d}_t\rangle^2}{\mu_f^{\mathcal{L}_0}\delta^2}. \tag{B.4}$$

Note that using a similar reasoning, as $\langle \nabla f(\mathbf{x}_t), \mathbf{x}_t - \mathbf{v}_t\rangle = g(\mathbf{x}_t)$, in both cases it holds that:

$$h(\mathbf{x}_t) \leq g(\mathbf{x}_t) \leq -\langle \nabla f(\mathbf{x}_t), \mathbf{d}_t\rangle. \tag{B.5}$$

As in the preceding proofs, the backtracking line search in Algorithm 6 will either output a point $\gamma_t = \gamma_{\max}$ or $\gamma_t < \gamma_{\max}$. In any case, and regardless of if a Frank-Wolfe step (Line 6) or an away step (Line 8) is chosen, Algorithm 6 will find and output a smoothness estimate L_t and a step size γ_t such that:

$$h(\mathbf{x}_{t+1}) - h(\mathbf{x}_t) \leq \frac{L_t\gamma_t^2}{2}\|\mathbf{d}_t\|^2 + \gamma_t\langle \nabla f(\mathbf{x}_t), \mathbf{d}_t\rangle. \tag{B.6}$$

As before, we will have two different cases. If $\gamma_t = \gamma_{\max}$ we know by observing Line 5 of Algorithm 6 that $-\langle \nabla f(\mathbf{x}_t), \mathbf{d}_t\rangle \geq \gamma_{\max}L_t\|\mathbf{d}_t\|^2$, and so plugging into Equation (B.6) we arrive at $h(\mathbf{x}_{t+1}) - h(\mathbf{x}_t) \leq \langle \nabla f(\mathbf{x}_t), \mathbf{d}_t\rangle \gamma_{\max}/2$. In the case where $\gamma_t < \gamma_{\max}$, we have that $-\langle \nabla f(\mathbf{x}_t), \mathbf{d}_t\rangle < \gamma_{\max}L_t\|\mathbf{d}_t\|^2$ and $\gamma_t = -\langle \nabla f(\mathbf{x}_t), \mathbf{d}_t\rangle / (L_t\|\mathbf{d}_t\|^2)$, and so plugging into Equation (B.6) we arrive at $h(\mathbf{x}_{t+1}) - h(\mathbf{x}_t) \leq -\langle \nabla f(\mathbf{x}_t), \mathbf{d}_t\rangle^2 / (2L_t\|\mathbf{d}_t\|^2)$. In any case, we can rewrite Equation (B.6) as:

$$h(\mathbf{x}_t) - h(\mathbf{x}_{t+1}) \geq \min\left\{\frac{-\langle \nabla f(\mathbf{x}_t), \mathbf{d}_t\rangle \gamma_{\max}}{2}, \frac{\langle \nabla f(\mathbf{x}_t), \mathbf{d}_t\rangle^2}{2L_t\|\mathbf{d}_t\|^2}\right\}. \tag{B.7}$$

We can now use the inequality in Equation (B.4) to bound the second term in the minimization component of Equation (B.7), and Equation (B.5) to bound the first term. This leads to:

$$h(\mathbf{x}_t) - h(\mathbf{x}_{t+1}) \geq h(\mathbf{x}_t) \min \left\{ \frac{\gamma_{\max}}{2}, \frac{\mu_f^{\mathcal{L}_0} \delta^2}{4L_t \|\mathbf{d}_t\|^2} \right\} \tag{B.8}$$

$$\geq h(\mathbf{x}_t) \min \left\{ \frac{\gamma_{\max}}{2}, \frac{\mu_f^{\mathcal{L}_0} \delta^2}{4\tilde{L}D^2} \right\}. \tag{B.9}$$

where in the last inequality we have used $\|d_t\| \leq D$ and $L_t \leq \tilde{L}$ for all t. It remains to bound $\gamma_{\max}$ away from zero to obtain the linear convergence bound. For Frank-Wolfe steps we immediately have $\gamma_{\max} = 1$, but for away steps there is no straightforward way of bounding $\gamma_{\max}$ away from zero. One of the key insights from Lacoste-Julien & Jaggi (2015) is that instead of bounding $\gamma_{\max}$ away from zero for all steps up to iteration t, we can instead bound the number of away steps with a step size $\gamma_t = \gamma_{\max}$ up to iteration t, which are steps that reduce the cardinality of the active set $\mathcal{S}_t$ and satisfy $h(\mathbf{x}_t) \leq h(\mathbf{x}_{t+1})$. This leads us to consider only the progress provided by the remaining steps, which are away steps with $\gamma_t < \gamma_{\max}$, and Frank-Wolfe steps. For a number of steps t, only at most half of these steps could have been away steps with $\gamma_t = \gamma_{\max}$, as we cannot drop more vertices from the active set than the number of vertices we could have potentially picked up with Frank-Wolfe steps. For the remaining $\lceil (t-1)/2 \rceil$ steps we know that $h(\mathbf{x}_t) - h(\mathbf{x}_{t+1}) \geq h(\mathbf{x}_t)\mu_f^{\mathcal{L}_0}\delta^2/(4\tilde{L}D^2)$. Therefore we have that the primal gap satisfies:

$$h(\mathbf{x}_t) \leq h(\mathbf{x}_0) \left(1 - \frac{\mu_f^{\mathcal{L}_0} \delta^2}{4\tilde{L}D^2}\right)^{\lceil (t-1)/2 \rceil}.$$

This completes the proof. □

We can make use of the proof of convergence in primal gap to prove linear convergence in Frank-Wolfe gap. In order to do so, we recall a quantity formally defined in Kerdreux et al. (2019) but already implicitly used earlier in Lacoste-Julien & Jaggi (2015) as:

$$\begin{aligned} w(\mathbf{x}_t, \mathcal{S}_t) &\stackrel{\text{def}}{=} \max_{\mathbf{u} \in \mathcal{S}_t, \mathbf{v} \in \mathcal{X}} \langle \nabla f(\mathbf{x}_t), \mathbf{u} - \mathbf{v} \rangle \\ &= \max_{\mathbf{u} \in \mathcal{S}_t} \langle \nabla f(\mathbf{x}_t), \mathbf{u} - \mathbf{x}_t \rangle + \max_{\mathbf{v} \in \mathcal{X}} \langle \nabla f(\mathbf{x}_t), \mathbf{x}_t - \mathbf{v} \rangle \\ &= \max_{\mathbf{u} \in \mathcal{S}_t} \langle \nabla f(\mathbf{x}_t), \mathbf{u} - \mathbf{x}_t \rangle + g(\mathbf{x}_t). \end{aligned}$$

Note that as the first term, the so-called *away gap* in the previous equation is positive and hence $w(\mathbf{x}_t, \mathcal{S}_t)$ provides an upper bound on the Frank-Wolfe gap.

Theorem B.3. *Suppose $\mathcal{X}$ is a polytope and f is a (M, ν) generalized self-concordant function with $\nu \geq 2$ for which the domain does not contain straight lines. Then the Away-step Frank-Wolfe (AFW) algorithm with Backtrack (Algorithm 7) contracts the Frank-Wolfe gap linearly, i.e., $\min_{1 \leq t \leq T} g(\mathbf{x}_t) \leq \varepsilon$ after $T = O(\log 1/\varepsilon)$ iterations.*

Proof. Note that the condition in Line 5 of Algorithm 7 means that regardless of if we chose to perform an away step of a Frank-Wolfe step, we have that $-2\langle \nabla f(\mathbf{x}_t), \mathbf{d}_t \rangle \geq \langle \nabla f(\mathbf{x}_t), \mathbf{x}_t - \mathbf{v}_t \rangle + \langle \nabla f(\mathbf{x}_t), \mathbf{a}_t - \mathbf{x}_t \rangle = w(\mathbf{x}_t, \mathcal{S}_t)$. On the other hand, we also have that $h(\mathbf{x}_t) - h(\mathbf{x}_{t+1}) \leq h(\mathbf{x}_t)$. Plugging these bounds into the right-hand side and the left hand side of Equation B.7 in Theorem B.2, and using the fact that $\|\mathbf{d}_t\| \leq D$ we have that:

$$\min \left\{ \frac{w(\mathbf{x}_t, \mathcal{S}_t)\gamma_{\max}}{4}, \frac{w(\mathbf{x}_t, \mathcal{S}_t)^2}{8L_t D^2} \right\} \leq h(\mathbf{x}_t) \leq h(\mathbf{x}_0) \left(1 - \frac{\mu_f^{\mathcal{L}_0}}{4\tilde{L}} \left(\frac{\delta}{D}\right)^2\right)^{\lceil (t-1)/2 \rceil},$$

where the second inequality follows from the convergence bound on the primal gap from Theorem B.2. Considering the steps that are not away steps with $\gamma_t = \gamma_{\max}$ as in the proof of Theorem B.2, leads us to:

$$g(\mathbf{x}_t) \leq w(\mathbf{x}_t, \mathcal{S}_t) \leq 4h(\mathbf{x}_0) \max\left\{1, \sqrt{\frac{\tilde{L}D^2}{2h(\mathbf{x}_0)}}\right\}\left(1 - \frac{\mu_f^{\mathcal{L}_0}}{4\tilde{L}}\left(\frac{\delta}{D}\right)^2\right)^{\lfloor (t-1)/4 \rfloor}.$$

□

In Table 4 we provide a detailed complexity comparison between the Backtracking AFW algorithm (`B-AFW`) algorithm, which can be found in Algorithm 7 in the appendix, and other comparable algorithms in the literature. Note that these complexities assume that the domain under consideration is polyhedral.

Algorithm	**SOO calls**	**FOO calls**	**ZOO calls**	**LMO calls**	**DO calls**
`FW-LLOO` [1, Alg.7]	$\mathcal{O}(\log 1/\varepsilon)$	$\mathcal{O}(\log 1/\varepsilon)$		$\mathcal{O}(\log 1/\varepsilon)^*$	
`ASFW-GSC` [1, Alg.8]	$\mathcal{O}(\log 1/\varepsilon)$	$\mathcal{O}(\log 1/\varepsilon)$		$\mathcal{O}(\log 1/\varepsilon)$	
`B-AFW`†‡ **[This work]**		$\mathcal{O}(\log 1/\varepsilon)$	$\mathcal{O}(\log 1/\varepsilon)$	$\mathcal{O}(\log 1/\varepsilon)$	$\mathcal{O}(\log 1/\varepsilon)$

Table 4: **Complexity comparison**: Number of iterations needed to reach an ε-optimal solution in $h(\mathbf{x})$ for Problem 1.1. We denote Dvurechensky et al. (2020b) by [1]. The asterisk on `FW-LLOO` highlights the fact that the oracle is different from the standard `LMO`. Lastly, the complexity shown for the `FW-LLOO`, `ASFW-GSC`, and `B-AFW` algorithms only apply to polyhedral domains, with the additional requirement that for the former two we need an explicit polyhedral representation of the domain. We use the superscript † to indicate that the same complexities hold when reaching an ε-optimal solution in $g(\mathbf{x})$, and the superscript ‡ to indicate that constants in the convergence bounds depend on user-defined inputs.

C Remarks and supplementary information on the experimental section

We ran all experiments on a server with 8 Intel Xeon 3.50GHz CPUs and 32GB RAM. All computations are run in single-threaded mode using Julia 1.6.0 with the `FrankWolfe.jl` package. The data sets used in the problem instances can be found in Carderera et al. (2021), the code used for the experiments can be found on `https://github.com/ZIB-IOL/fw-generalized-selfconcordant`. When running the adaptive step size from Pedregosa et al. (2020), the only parameter that we need to set is the initial smoothness estimate L_{-1}. We use the initialization proposed in the Pedregosa et al. (2020) paper, namely:

$$L_{-1} = \|\nabla f(\mathbf{x}_0) - \nabla f(\mathbf{x}_0 + \varepsilon(\mathbf{v}_0 - \mathbf{x}_0))\| / (\varepsilon \|\mathbf{v}_0 - \mathbf{x}_0\|)$$

with ε set to 10^{-3}. The scaling parameters $\tau = 2, \eta = 0.9$ are left at their default values as proposed in Pedregosa et al. (2020) and also used in Dvurechensky et al. (2020b).

We provide the full details of the experiments carried out in the paper:

Portfolio optimization. We consider $f(\mathbf{x}) = -\sum_{t=1}^{p} \log(\langle \mathbf{r}_t, \mathbf{x}\rangle)$, where p denotes the number of periods and $\mathcal{X} = \Delta_n$. The results are shown in Figure 2. We use the revenue data $\mathbf{r}_t$ from Dvurechensky et al. (2020b) and add instances generated in a similar fashion from independent Normal random entries with 1000, 2000, and 5000 dimensions, and from a Log-normal distribution with $(\mu = 0.0, \sigma = 0.5)$.

Signal recovery with KL divergence. We apply the aforementioned algorithms to the recovery of a sparse signal from a noisy linear image using the Kullback-Leibler divergence. Given a linear map W, we assume a signal $\mathbf{y}$ is generated by $\mathbf{y} = W\mathbf{x}_0 + \varepsilon$, where $\mathbf{x}_0$ is assumed to be a sparse unknown input signal and ε is a random error. Assuming W and $\mathbf{y}$ are entrywise positive, and that the signal to recover should also be entrywise positive, the minimizer of the KL divergence (or Kullback's I-divergence (Csiszar, 1991)) can be used as an estimator for $\mathbf{x}_0$. The KL divergence between the resulting output signals is expressed as $f(\mathbf{x}) = D(W\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{N} \left\{ \langle \mathbf{w}_i, \mathbf{x} \rangle \log\left(\frac{\langle \mathbf{w}_i, \mathbf{x} \rangle}{y_i}\right) - \langle \mathbf{w}_i, \mathbf{x} \rangle + y_i \right\}$, where $\mathbf{w}_i$ is the i^{th} row of W. In order to promote sparsity and enforce nonnegativity of the solution, we use the unit simplex of radius R as the feasible set $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}_+^d, \|\mathbf{x}\|_1 \leq R\}$. The results are shown in Figure 3. We used the same $M = 1$ choice for the second-order method as in Dvurechensky et al. (2020b) for comparison; whether this choice is admissible is unknown (see Remark C.1). We generate input signals $\mathbf{x}_0$ with 30% non-zeros elements following an exponential distribution of mean $\lambda = 1$. The entries of W are generated from a folded Normal distribution built from absolute values of Gaussian random numbers with standard deviation 5 and mean 0. The additive noise is generated from a Gaussian centered distribution with a standard deviation equal to a fraction of the standard deviation of $W\mathbf{x}_0$.

Logistic regression. One of the motivating examples for the development of a theory of generalized self-concordant function is the logistic regression problem, as it does not match the definition of a standard self-concordant function but shares many of its characteristics. We consider a design matrix with rows $\mathbf{a}_i \in \mathbb{R}^n$ with $1 \leq i \leq N$ and a vector $\mathbf{y} \in \{-1, 1\}^N$ and formulate a logistic regression problem with elastic net regularization, in a similar fashion as is done in Liu et al. (2020), with $f(\mathbf{x}) = 1/N \sum_{i=1}^{N} \log(1 + \exp(-y_i \langle \mathbf{x}, \mathbf{a}_i \rangle)) + \mu/2 \|\mathbf{x}\|^2$, and $\mathcal{X}$ is the ℓ_1 ball of radius ρ, where μ and ρ are two regularization parameters. The logistic regression loss is generalized self-concordant with $\nu = 2$. The results can be seen in Figure 4 and expanded in Section C. We use the `a1a-a9a` datasets from the LIBSVM classification data.

Birkhoff polytope. All applications previously considered all have in common a constraint set possessing computationally inexpensive LMOs (probability or unit simplex and ℓ_1 norm ball). Additionally, each vertex returned from the LMO is highly sparse with at most one non-zero element. To complement the results we consider a problem over the Birkhoff polytope, the polytope of doubly stochastic matrices, where the LMO is implemented through the Hungarian algorithm, and is not as inexpensive as in the other examples considered. We use a quadratic regularization parameter $\mu = 100/\sqrt{n}$ where n is the number of samples.

Remark C.1. Note that Proposition 2 in Sun & Tran-Dinh (2019), which deals with the composition of generalized self-concordant functions with affine maps, does not apply to the KL divergence objective function, reproduced here for reference:

$$f(\mathbf{x}) = D(W\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{N} \left\{ \langle \mathbf{w}_i, \mathbf{x} \rangle \log\left(\frac{\langle \mathbf{w}_i, \mathbf{x} \rangle}{y_i}\right) - \langle \mathbf{w}_i, \mathbf{x} \rangle + y_i \right\}.$$

Furthermore, the objective function is strongly convex if and only if $\operatorname{rank}(W) \geq n$, where n is the dimension of the problem.

Proof. (Sun & Tran-Dinh, 2019, Proposition 2) establishes certain conditions under which the composition of a generalized self-concordant function with an affine map results in a generalized self-concordant function. The objective is of the form:

$$\sum_{i=1}^{N} \phi_i(\langle \mathbf{w}_i, \mathbf{x} \rangle) \quad \text{with} \quad \phi_i(t) = t \log\left(\frac{t}{y_i}\right) - t + y_i = t \log t - t \log y_i - t + y_i.$$

Note that generalized self-concordant functions are closed under addition, and so we only focus on the individual terms in the sum. As first-order terms are $(0, \nu)$-generalized self-concordant for any $\nu > 0$, then we know that the composition of these first-order terms with an affine map results in a generalized self-concordant function Sun & Tran-Dinh (2019, Proposition 2). We therefore focus on the entropy function $t \log t$ which is $(1, 4)$ generalized self-concordant. The conditions which ensure that the composition of a (M, ν)-generalized self-concordant function with an affine map $x \mapsto Ax$ results in a generalized self-concordant function requires in the case $\nu > 3$ that $\lambda_{min}(A^T A) > 0$ (Sun & Tran-Dinh, 2019, Proposition 2). In the case of the KL divergence objective, $A = \mathbf{w}_i^T$ and $A^T A = \mathbf{w}_i \mathbf{w}_i^T$ is an outer product with only one positive eigenvalue, and 0 of multiplicity $n - 1$. Therefore we cannot guarantee that the function $\phi_i(\langle \mathbf{w}_i, \mathbf{x}\rangle)$ is generalized self-concordant by application of Proposition 2 in Sun & Tran-Dinh (2019).

Alternatively, in order to try to show that the function is generalized self-concordant we could consider $f(\mathbf{x}) := g(W\mathbf{x})$. Assuming $\operatorname{rank}(W) \geq n$, then $W^T W$ is positive definite, and only the generalized self-concordance of g is left to prove.

$$g(\mathbf{z}) = \sum_{i=1}^{N} \phi_i(\mathbf{z}_i).$$

Each term $\mathbf{z} \mapsto \phi_i(\mathbf{z}_i) = \phi_i(\mathbf{e}_i^T \mathbf{z})$ with $\mathbf{e}_i$ the i^{th} standard basis vector is the composition of a generalized self-concordant function composed with a rank-one affine transformation, this raises the same issues encountered in the paragraph above.

Regarding the strong-convexity of the objective function, we can express the gradient and the Hessian of the function as:

$$\begin{aligned} \nabla f(\mathbf{x}) &= \sum_{i=1}^{N} \mathbf{w}_i (\log \langle \mathbf{w}_i, \mathbf{x}\rangle - \log y_i) \\ \nabla^2 f(\mathbf{x}) &= \sum_{i=1}^{N} \frac{\mathbf{w}_i \mathbf{w}_i^T}{\langle \mathbf{w}_i, \mathbf{x}\rangle}, \end{aligned}$$

which is the sum of N outer products, each corresponding to a single eigenvector $\mathbf{w}_i$. If $\operatorname{rank}(W) \geq n$, the Hessian is definite positive and the objective is strongly convex. Otherwise, it possesses zero as an eigenvalue regardless of $\mathbf{x}$, and the function Hessian is positive semi-definite. □

Strong convexity parameter for the LLOO. The LLOO procedure explicitly requires a strong convexity parameter σ_f of the objective function, an underestimator of $\lambda_{min}(\nabla^2 f(\mathbf{x}))$ over $\mathcal{X}$. For the portfolio optimization problem, the Hessian is a sum of rank-one terms:

$$\nabla^2 f(\mathbf{x}) = \sum_t \frac{\mathbf{r}_t \mathbf{r}_t^T}{\langle \mathbf{r}_t, \mathbf{x}\rangle}.$$

The only non-zero eigenvalue associated with each t term is bounded below over $\mathcal{X}$ by:

$$\frac{\|\mathbf{r}_t\|^2}{\max_{\mathbf{x}\in\mathcal{X}} \langle \mathbf{r}_t, \mathbf{x}\rangle^2} = \frac{\|\mathbf{r}_t\|^2}{\max\{\max_{\mathbf{x}\in\mathcal{X}} \langle \mathbf{r}_t, \mathbf{x}\rangle, -\min_{\mathbf{x}\in\mathcal{X}} \langle \mathbf{r}_t, \mathbf{x}\rangle\}^2}.$$

The denominator can be solved by two calls to the LMO, and we will denote it by β_t for the t^{th} term. Each summation term contributes positively to one of the eigenvalues of the Hessian matrix, an underestimator of the the strong convexity parameter is then given by:

$$\lambda_{min}\left(\sum_t \frac{\mathbf{r}_t \mathbf{r}_t^T}{\beta_t}\right).$$

The second-order method `GSC-FW` has been implemented with an in-place Hessian matrix updated at each iteration, following the implementation of Dvurechensky et al. (2020b). The Hessian computation nonetheless adds significant cost in the runtime of each iteration, even if the local norm and other quadratic expressions $\left\langle \nabla^2 f(\mathbf{x})\mathbf{y}, \mathbf{z}\right\rangle$ can be computed allocation-free. A potential improvement for future work would be to represent Hessian matrices as functional linear operators mapping any $\mathbf{y}$ to $\nabla^2 f(\mathbf{x})\mathbf{y}$.

Monotonous step size: the numerical case. The computational experiments highlighted that the Monotonous Frank-Wolfe performs well in terms of iteration count and time against other Frank-Wolfe and Away-step Frank-Wolfe variants. Another advantage of a simple step size computation procedure is its numerical stability. On some instances, an ill-conditioned gradient can lead to a plateau of the primal and/or dual progress. Even worse, some step-size strategies do not guarantee monotonicity and can result in the primal value increasing over some iterations. The numerical issue that causes this phenomenon is illustrated by running the methods of the `FrankWolfe.jl` package over the same instance using 64-bits floating-point numbers and Julia `BigFloat` types (which support arithmetic in arbitrary precision to remove numerical issues).

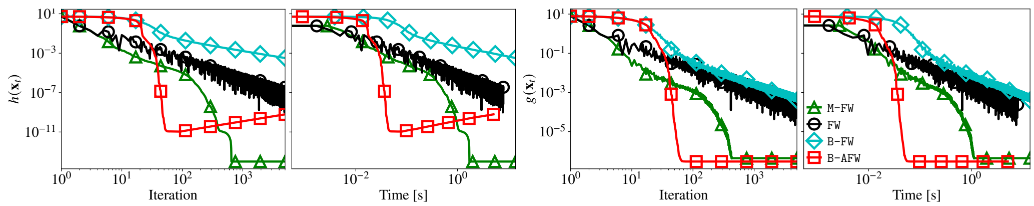


Figure 6: Ill-conditioned portfolio optimization problem solved using floating-point arithmetic.

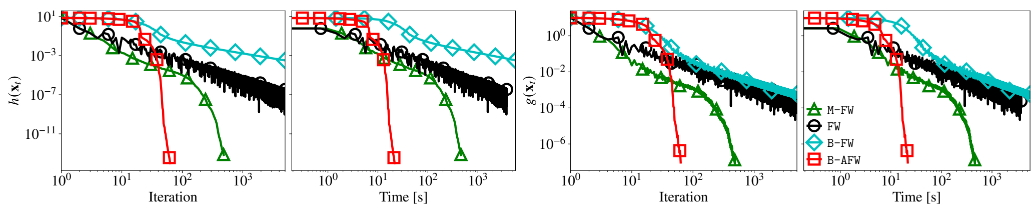


Figure 7: Ill-conditioned portfolio optimization problem solved using arbitrary precision.

In Figure 6, we observe a plateau of the dual gap for both `M-FW` and `B-AFW`. The primal value however worsens after the iteration where `B-AFW` reaches its dual gap plateau. In contrast, `M-FW` reaches a plateau in both primal and dual gap at a certain iteration. Note that the primal value at the point where the plateau is hit is already below $\sqrt{\varepsilon_{\text{float64}}}$, the square root of the machine precision. The same instance and methods operating in arbitrary precision arithmetic are presented Figure 7. Instead of reaching a plateau or deteriorating, `B-AFW` closes the dual gap tolerance and terminates before other methods. Although this observation (made on several instances of the portfolio optimization problem) only impacts ill-conditioned problems, it suggests `M-FW` may be a good candidate for a numerically robust default implementation of Frank-Wolfe algorithms.

Function domain in the constraints One of the arguments used to motivate the construction of FW algorithms for standard and generalized self-concordant minimization in prior work is the difficulty of handling objective functions with implicitly defined domains. We make several observations that highlight the relevance of this issue and justify the assumption of the availability of a Domain Oracle for $\mathrm{dom}(f)$. In the portfolio optimization example, all revenue vectors are assumed positive, $\mathbf{r}_t \in \mathbb{R}^n_{++}$ and $\mathbf{x} \in \Delta_n$, it follows that all feasible points lie in $\mathrm{dom}(f)$. More generally ,for the logarithm of an affine function, verifying that a candidate $\mathbf{x}$ lies in $\mathrm{dom}(f)$ consists of a single affine transformation and element-wise comparison $A\mathbf{x} + \mathbf{b} > \mathbb{R}^m_{++}$.

In the inverse covariance estimation problem, the information on $\mathrm{dom}(f)$ can be added to the constraints by imposing $\mathrm{mat}(\mathbf{x}) \in \mathbb{S}^n_+$, yielding a semi-definite optimization problem. The domain oracle consists of the computation of the smallest eigenvalue, which needs to be positive.

We can also modify the feasible region of the signal retrieval application using the KL divergence, resulting in a new feasible region $\mathcal{X}'$ so that $\mathrm{Int}\,(\mathcal{X}') \subset \mathrm{dom}(f)$. The objective is of the form:

$$f(\theta) = \sum_i \langle \mathbf{w}_i, \theta \rangle \log\left(\frac{\langle \mathbf{w}_i, \theta\rangle}{y_i}\right) - \langle \mathbf{w}_i, \theta \rangle$$

where the data W and y are assumed to be entrywise positive, thus $\mathrm{dom}(f) = \{x \in \mathbb{R}^n_+, x \neq 0\}$. Therefore we can define the set $\mathcal{X}'$ as the unit simplex. The domain of each function involved in the sum in f has an open domain $(0, +\infty)$.

However, the positivity assumption on all these components could be relaxed. Without the positivity assumption on W, the Domain Oracle would consist of verifying:

$$\langle \mathbf{w}_i, \theta \rangle > 0 \;\forall i. \tag{C.1}$$

This verification can however be simplified by a preprocessing step if the number of data points is large by finding the minimal set of supporting hyperplanes in the polyhedral cone (C.1), which we can find by solving the following linear problem:

$$\max_{\tau,\theta} \; \tau \tag{C.2a}$$

$$\text{s.t. } \tau \leq \langle \mathbf{w}_i, \theta \rangle \;\forall i \quad (\lambda_i) \tag{C.2b}$$

$$\|\theta\|_1 \leq R, \tag{C.2c}$$

where λ_i is the dual variable associated with the i^{th} inner product constraint. If the optimal solution of Problem (C.2a) is 0, the original problem is infeasible and the cone defined by (C.1) is empty. Otherwise the optimal θ will lie in the intersection of the closure of the polyhedral cone and the ℓ_1 norm ball. Furthermore, the support of λ provides us with the non-redundant inequalities of the cone. Let $\hat{W}$ be the matrix formed with the rows $\mathbf{w}_i$ such that $\lambda_i > 0$, then the Domain Oracle can be simplified to the verification that $\hat{W}\theta \in \mathbb{R}^n_{++}$. The Distance Weighted Discrimination (DWT) model also considered in Dvurechensky et al. (2020b) was initially presented in Marron et al. (2007), the denominator of each sum element ξ_i is initially constrained to be nonnegative, which makes $\operatorname{Int}(\mathcal{X}) \subseteq \operatorname{dom}(f)$ hold. Even without this additional constraint, the nonnegativity of all ξ_i can be ensured with a minimum set of linear constraints in a fashion similar to the signal retrieval application, thus simplifying the Domain Oracle.

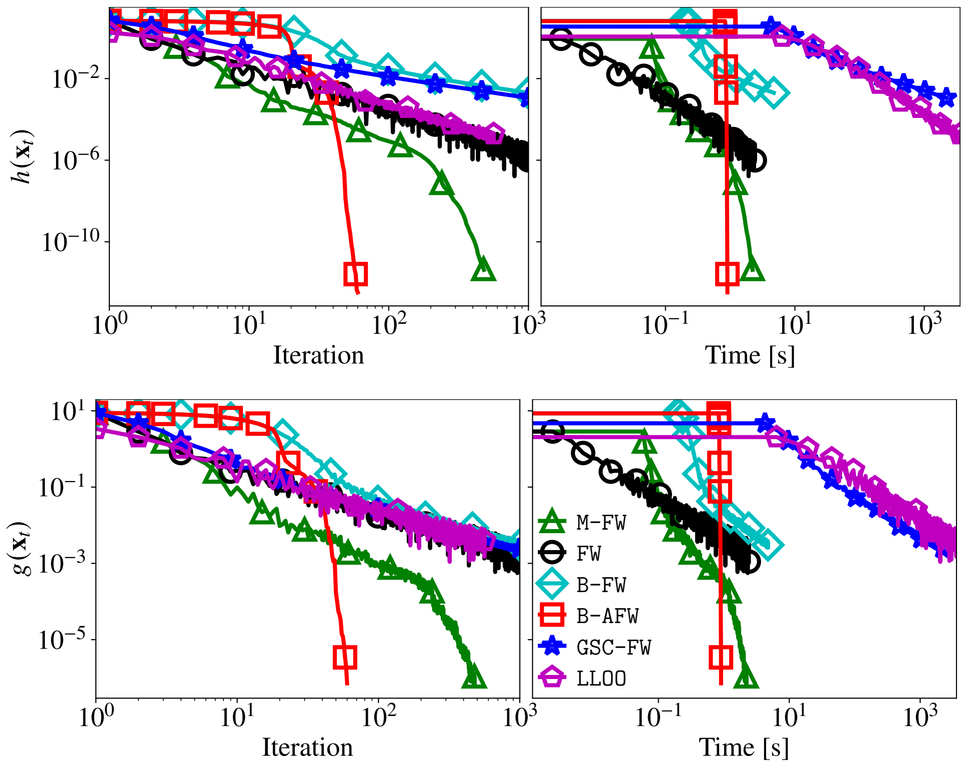


Figure 8: **Portfolio Optimization**: Convergence of $h(\mathbf{x}_t)$ and $g(\mathbf{x}_t)$ vs. t and wall-clock time.

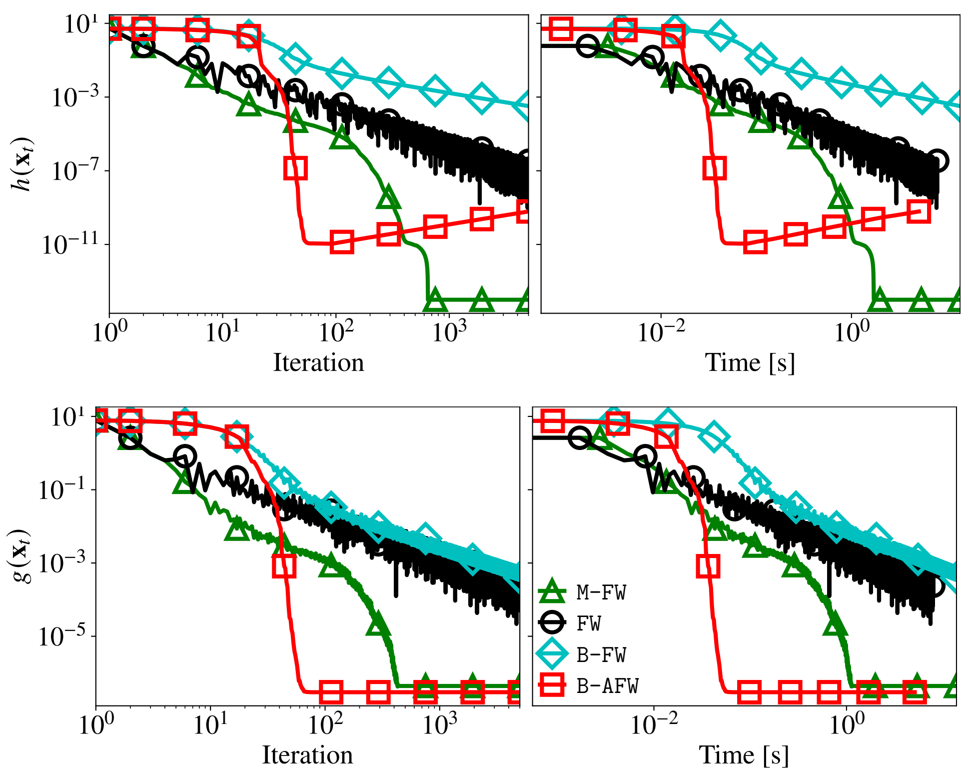


Figure 9: **Portfolio Optimization**: Convergence of $h(\mathbf{x}_t)$ and $g(\mathbf{x}_t)$ vs. t and wall-clock time.

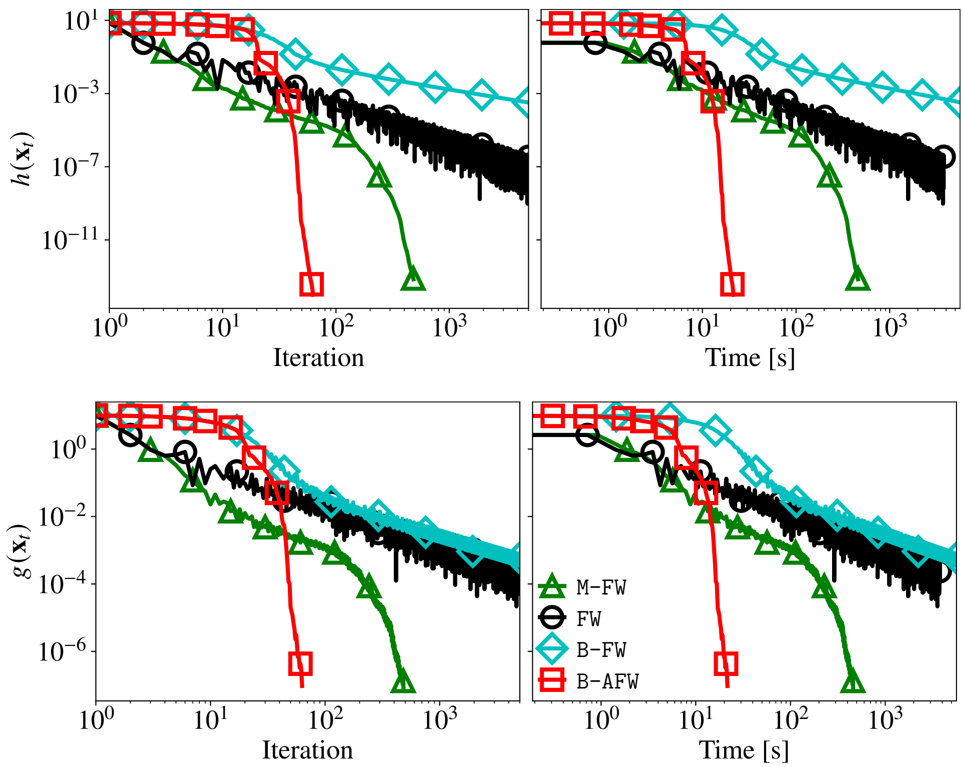


Figure 10: **Portfolio Optimization**: Convergence of $h(\mathbf{x}_t)$ and $g(\mathbf{x}_t)$ vs. t and wall-clock time.

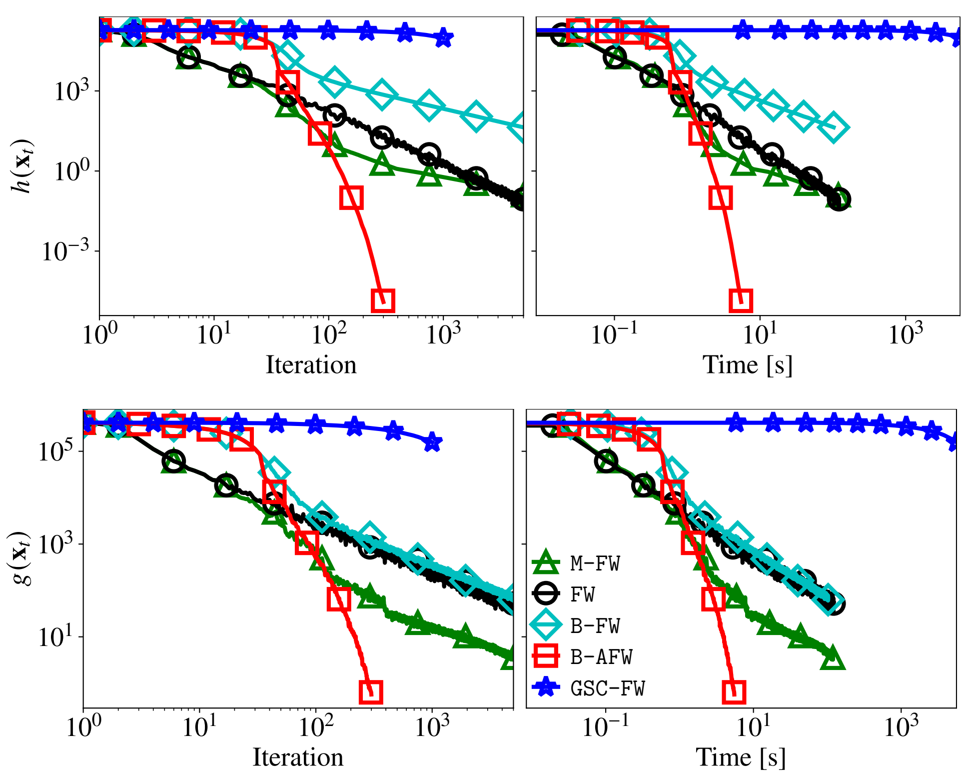


Figure 11: **Signal Recovery**: Convergence of $h(\mathbf{x}_t)$ and $g(\mathbf{x}_t)$ vs. t and wall-clock time.

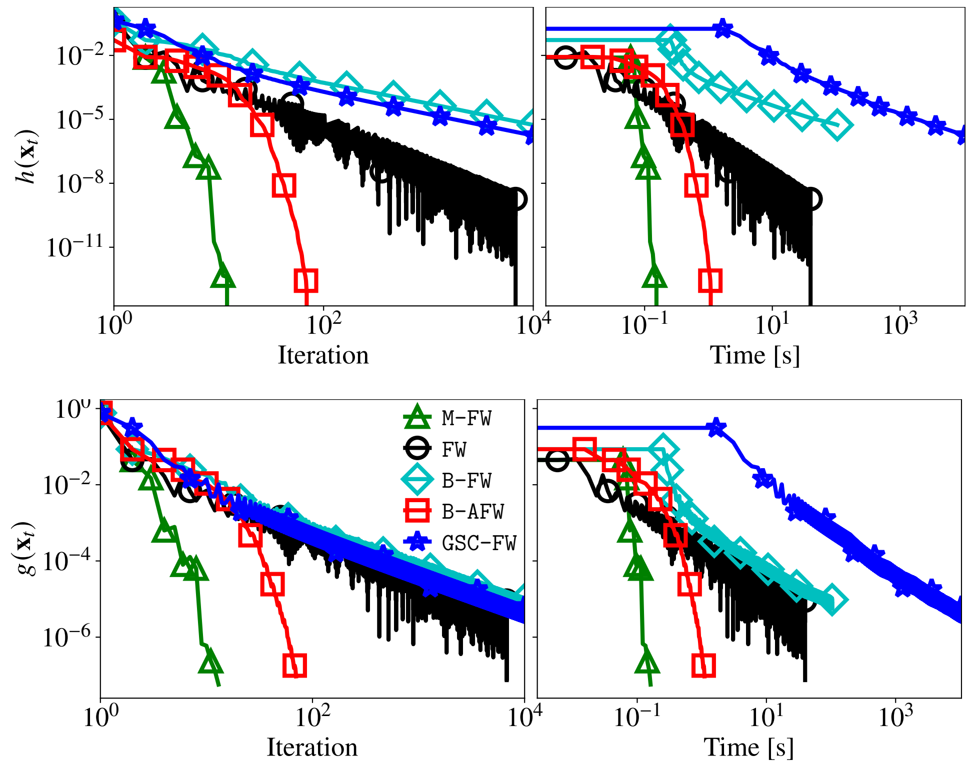


Figure 12: **Logistic Regression**: Convergence of $h(\mathbf{x}_t)$ and $g(\mathbf{x}_t)$ vs. t and wall-clock time for the a4a LIBSVM dataset.

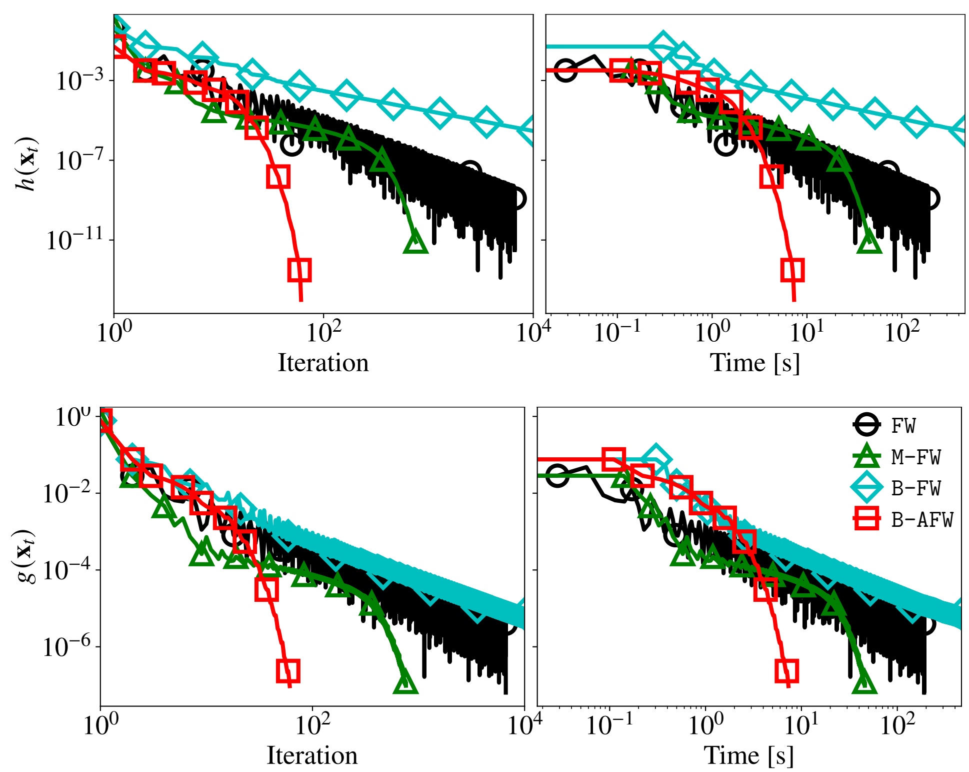


Figure 13: **Logistic Regression**: Convergence of $h(\mathbf{x}_t)$ and $g(\mathbf{x}_t)$ vs. t and wall-clock time for the a8a LIBSVM dataset.

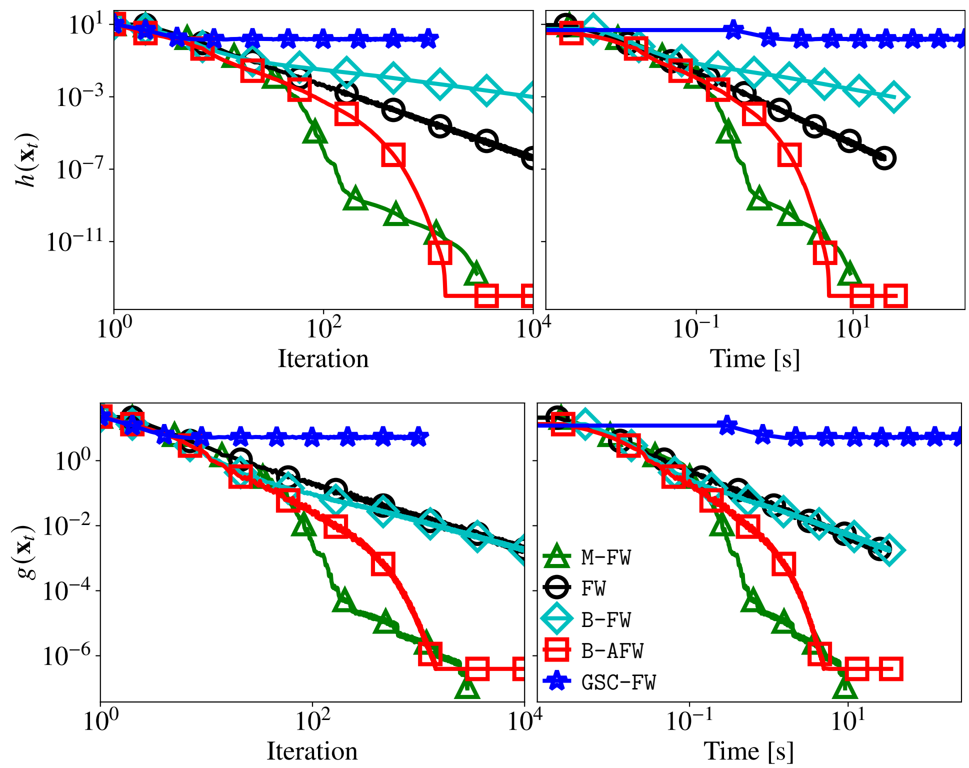


Figure 14: **Birkhoff Polytope**: Convergence of $h(\mathbf{x}_t)$ and $g(\mathbf{x}_t)$ vs. t and wall-clock time.