On the Convergence of IRLS and Its Variants in Outlier-Robust Estimation

Liangzu Peng
Johns Hopkins University
lpeng25@jhu.edu

Christian Kümmerle
UNC Charlotte
kuemmerle@uncc.edu

René Vidal
University of Pennsylvania
vidalr@seas.upenn.edu

Abstract

Outlier-robust estimation involves estimating some parameters (e.g., 3D rotations) from data samples in the presence of outliers, and is typically formulated as a non-convex and non-smooth problem. For this problem, the classical method called iteratively reweighted least-squares (IRLS) and its variants have shown impressive performance. This paper makes several contributions towards understanding why these algorithms work so well. First, we incorporate majorization and graduated non-convexity (GNC) into the IRLS framework and prove that the resulting IRLS variant is a convergent method for outlier-robust estimation. Moreover, in the robust regression context with a constant fraction of outliers, we prove this IRLS variant converges to the ground truth at a global linear and local quadratic rate for a random Gaussian feature matrix with high probability. Experiments corroborate our theory and show that the proposed IRLS variant converges within 5*-*10 *iterations for typical problem instances of outlier-robust estimation, while state-of-the-art methods need at least* 30 *iterations. A basic implementation of our method is provided: https://github.com/liangzu/IRLS-CVPR2023*

> *... attempts to analyze this difficulty [*caused by infinite weights of IRLS for the ℓ_p-loss*] have a long history of proofs and counterexamples to incorrect claims.*
>
> Khurrum Aftab & Richard Hartley [1]

1. Introduction

1.1. The Outlier-Robust Estimation Problem

Many parameter estimation problems can be stated in the following general form. We are given some function $r : \mathcal{C} \times \mathcal{D} \to [0, \infty)$, called the *residual* function. Here $\mathcal{D}$ is the domain of data samples $d_1, \dots, d_m$, and $\mathcal{C} \subset \mathbb{R}^n$ is the *constraint set* where our (ground truth) *variable* v^* lies; $\mathcal{C}$ can be convex such as an affine subspace, or non-convex such as a special orthogonal group $\mathrm{SO}(3)$. We aim to recover v^* from data d_i's. A simple example is *linear regression*, where a sample $d_i = (\boldsymbol{a}_i, y_i)$ consists of a feature vector $\boldsymbol{a}_i \in \mathbb{R}^n$ and a scalar response $y_i \in \mathbb{R}$, and the residual function is $r(v, d_i) := |\boldsymbol{a}_i^\top v - y_i|$.

The sample d_i is called an *inlier*, if $r(v^*, d_i) \approx 0$. It is called an *outlier*, if the residual $r(v^*, d_i)$ is *large* (vaguely speaking). If all samples are inliers, one usually prefers solving the following problem as a means to estimate v^*:

$$\min_{v \in \mathcal{C}} \sum\nolimits_{i=1}^{m} r(v, d_i)^2 \tag{1}$$

Problem (1) is called *least-squares*, and is known since Legendre [41] and Gauss [29] in the linear regression context. Even before that, Boscovich [13] suggested to minimize (1) without the square. This unsquared version is called *least absolute deviation*, and is more robust to outliers than (1).

Consider the following formulation for *outlier-robust estimation* (*i.e.*, a specific type of *M-estimators* [35, 55]):

$$\min_{v \in \mathcal{C}} \sum\nolimits_{i=1}^{m} \rho\big(r(v, d_i)\big) \tag{2}$$

Here $\rho : \mathbb{R} \to \mathbb{R}$ is some outlier-robust loss (the unsquared version of (1) corresponds to $\rho(r) = |r|$ in (2)). Among many possible losses ρ [19, 23], we discuss two particular choices. The first is the ℓ_p-loss $\rho(r) = |r|^p/p$, $p \in (0, 1]$; it has been used in several research fields, *e.g.*, *geometric vision* [1, 19], *compressed sensing* [17, 22, 36], *matrix recovery* [37, 44, 45], and *subspace clustering* [26]. The other loss is due to Huber [35]: $\rho(r) = \min\{r^2, c^2\}$, with $c > 0$ a hyper-parameter; it has later been named as *Talwar* [21, 48], *Huber-type skipped mean* [30], *truncated quadratic* [6, 11], and *truncated least-squares* (TLS) [4, 68, 74]. Both losses are highly robust to outliers but make solving (2) difficult, *e.g.*, the objective of (2) becomes non-smooth or non-convex. This motivates the need to develop efficient and provably correct solvers for (2) with either of the two losses.

1.2. IRLS and Its Variants in Vision & Optimization

The General Principle of IRLS. As its name suggests, *iteratively reweighted least-squares* (IRLS) is a general algorithmic paradigm that alternates between defining a weight for each sample and solving a weighted least squares problem. Specifically, IRLS initializes a variable $v^{(0)} \in \mathcal{C}$, and, for $t = 0, 1, \ldots$, alternates between the following two steps:

$$\text{Update weights } w_i^{(t+1)} \text{ based on } v^{(t)}, \forall i = 1, \ldots, m \quad (3)$$

$$\text{Solve: } v^{(t+1)} \leftarrow \underset{v \in \mathcal{C}}{\operatorname{argmin}} \sum_{i=1}^{m} w_i^{(t+1)} r(v, d_i)^2 \quad (4)$$

This basic idea dates back to the seminal work of Weiszfeld [66]; see [9,32] for some historical accounts. A well-known and general rule for the weight update is (*cf*. [1,21,47])

$$w_i^{(t+1)} \leftarrow \rho'\big(r_i^{(t)}\big)/r_i^{(t)}, \quad r_i^{(t)} := r(v^{(t)}, d_i). \qquad (5)$$

In a nutshell, the rationale behind rule (5) is to "connect" weighted least-squares (4) to outlier-robust estimation (2), allowing IRLS to optimize the latter (2). Indeed, [1] shows that IRLS with the weight update in (5) results in a non-increasing objective (2). Moreover, [1] gives conditions under which IRLS with (5) converges to a stationary point of (2). This confirms that one can apply IRLS to problem (2), as long as one can solve weighted least-squares[1] (4).

However, the conditions of the theorem of [1] are hard to verify, *e.g.*, one condition requires the minimizer of (4) to be a continuous function of weights $w_i^{(t+1)}$. Moreover, as [1] commented, directly applying (5) to non-smooth or non-convex losses (*e.g.*, ℓ_p or TLS) might create *significant theoretical and practical difficulties*, *e.g.*, (5) is undefined at non-differentiable points. This suggests that rule (5) needs to be improved if the ℓ_p or TLS loss is to be minimized.

IRLS in A Tale of Two Losses. For the non-smooth ℓ_p-loss, (5) *results in*[2] $w_i^{(t+1)} \leftarrow (r_i^{(t)})^{p-2}$, which tends to infinity as $r_i^{(t)} \to 0$. A workaround is to truncate the residual by some positive number ϵ, *i.e.*, $w_i^{(t+1)} \leftarrow \max\{r_i^{(t)}, \epsilon\}^{p-2}$ [19, 24–26, 42, 43, 64]. While [1, 59] considered this to be "*an ad-hoc procedure*", in the optimization literature, there do exist theoretical guarantees for IRLS with this revised weight update to converge, at least for some specific residual functions r, see, *e.g.*, [8, 16, 42, 43].

For the non-smooth and non-convex TLS loss $\rho(r) = \min\{r^2, c^2\}$, (5) *results in*[2] a *hard thresholding* scheme: set $w_i^{(t+1)}$ to 1 if $r_i^{(t)} \leq c$, or set it to 0 otherwise. IRLS fails with such a weight update if the outlier rate exceeds 10% for *category-level perception* as reported in [57]. This could be remedied in two ways, discussed next.

The first is to adopt a different hard thresholding method [10] from the optimization literature, which sets $w_i^{(t+1)}$ to 1 if $r_i^{(t)}$ is among the s-smallest of all residuals (s is a hyper-parameter), or set $w_i^{(t+1)}$ to 0 otherwise; this method is robust up to 50% outliers for *robust regression*, and converges globally linearly under some conditions. Note that this IRLS variant is not meant to minimize the TLS loss.

The second remedy manifests itself if one applies rule (5) to some *smoothing* approximation $\rho_\mu(r)$ of the TLS loss $\rho(r) = \min\{r^2, c^2\}$. The approximation of [12] is

$$\rho_\mu(r) = \begin{cases} r^2, & \text{if } r^2 \leq \frac{\mu c^2}{\mu+1}, \\ c^2, & \text{if } r^2 \geq \frac{\mu+1}{\mu} c^2, \\ 2c|r|\sqrt{\mu(\mu+1)} - \mu(c^2 + r^2), & \text{o/w.} \end{cases} \qquad (6)$$

Since $\rho_\mu \to \rho$ as $\mu \to \infty$, a natural strategy, called *graduated non-convexity* (GNC) [12], is to alternate between optimizing ρ_μ and increasing μ at each iteration t. The method used for increasing μ is called a GNC *schedule* and the default schedule has been a *linear* one, *i.e.*, $\mu^{(t+1)} \leftarrow \gamma\mu^{(t)}$ with some hyper-parameter $\gamma > 1$ [39, 46, 57, 60, 68, 74]. For example, the `GNC-TLS` method [12, 68] incorporates this linear schedule within the IRLS framework (3)-(5) to approximate the TLS loss via ρ_μ.

However, the great engineering intuition of [12] and its follow-up works [4, 39, 62, 68, 75] on GNC comes with the lack of theoretical guarantees, thus [69, 71] refer to GNC as a "*fast heuristic*" strategy. On the other hand, in the optimization literature, similar GNC twists for the ℓ_p-loss have been empirically investigated [18, 65, 67] for compressed sensing and related problems, and empowered with global linear or local superlinear convergence rates [22, 36, 46, 52].

For outlier-robust estimation (2), either with general [4, 39, 68] or specific residual functions [26, 52], either with the ℓ_p [26, 46, 52], TLS [4, 39, 57, 68], or even other losses [59, 73], combining IRLS and GNC has pushed the empirical performance to a certain limit, which other types of methods (*e.g.*, `RANSAC` [28]) can hardly attain given the same time budget. On the other hand, theoretical guarantees for IRLS offered in the optimization literature are limited to specific problems (*e.g.*, compressed sensing [22]), and, though related, cannot be applied directly to outlier-robust estimation (2). An intriguing but under-explored theoretical question is why IRLS, GNC, and the like work so well for outlier-robust estimation (2)—can we extend, not just apply, the insights from optimization to answer this question?

1.3. Our Contribution

We present an IRLS variant called $\texttt{GNC-IRLS}_p$ (Algorithm 1) for the outlier-robust estimation problem (2) and establish general convergence properties for general constraint sets $\mathcal{C}$, providing a well-founded framework for empirically successful GNC methods. We further elucidate how appropriately chosen update rules for the smoothing parameter $\epsilon^{(t)}$ (Line 7) of $\texttt{GNC-IRLS}_p$ lead to a global and fast local convergence for outlier-robust estimation problems. More specifically, our contributions are as follows:

- In Section 2, we consider outlier-robust estimation (2) for a general class of residual functions and constraints, and we prove that $\texttt{GNC-IRLS}_p$ converges to stationary points

[1] While solving weighted least-squares (4) can be hard, many solvers for geometric vision exist, see, *e.g.*, [2, 5, 15, 33, 34, 49, 56, 57, 68, 73].

[2] Pretending that the ℓ_p or TLS losses are differentiable everywhere.

Algorithm 1: GNC-IRLS$_p$

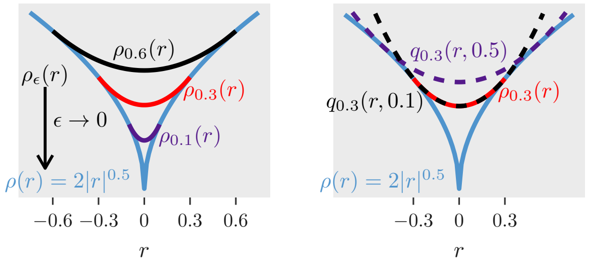
```
1 Input: data $d_1, \dots, d_m$, $p \in (0, 1]$;
2 Let $v^{(0)} \in \mathcal{C}$ with $\|v^{(0)}\|_2 < \infty$ and $\epsilon^{(0)} \in (0, \infty)$;
3 For $t \leftarrow 0, 1, 2, \dots$:
4     Compute the residual $r_i^{(t)} \leftarrow r(v^{(t)}, d_i), \forall i$;
5     $w_i^{(t+1)} \leftarrow \max\{r_i^{(t)}, \epsilon^{(t)}\}^{p-2}$ ;   // Sec. 2
6     Solve problem (4) and get $v^{(t+1)}$ ;   // Sec. 2
7     Calculate $\epsilon^{(t+1)}$ based on a GNC schedule ;   // Sec. 3
```

of (some majorizer of) the ℓ_p-loss under suitable assumptions (Theorem 1). Moreover, the assumptions are easy to verify and satisfied by many geometric vision problems (see the appendix). This challenges the viewpoint of [1, 59] that truncating the residual (Line 4, Algorithm 1) is "*ad-hoc*". Our proof is enabled by a *majorization* interpretation of GNC-IRLS$_p$, and is motivated by [22, 45, 54]. As we will discuss, our result is more general than those of [22, 45, 54].

- In Section 3, we propose a *superlinear* GNC schedule for GNC-IRLS$_p$, as opposed to a *linear* one. We prove that GNC-IRLS$_p$ with such a schedule converges to the ground truth at a global linear and local superlinear rate, with high probability (Theorem 2). Moreover, GNC-IRLS$_p$ provably enjoys quadratic rates starting from the first iteration. A theoretical drawback of this powerful result is that it has a "burn-in" period and is limited to the *robust regression* setting; this is harmless though when it comes to practical use. Our proof is motivated by [46]. Their result holds only for $p = 1$, and our contribution lies not only in overcoming the non-convexity for the case of $p < 1$, but in leveraging the non-convexity to obtain a faster convergence rate.
- In Section 4 we compare the performance of GNC-TLS and GNC-IRLS$_p$ for *point cloud registration*. GNC-IRLS$_p$ terminates in 10 iterations while GNC-TLS takes 30. This is because GNC-IRLS$_p$ uses a superlinear GNC schedule, while GNC-TLS uses a linear schedule.
- In Section 5, we endow the TLS loss with a *majorization* strategy and a *superlinear* GNC schedule, leading to an IRLS method that we call MS-GNC-TLS. With majorization we prove MS-GNC-TLS converges, which challenges the viewpoint of [69, 71] that GNC is "*heuristic*". With the superlinear schedule, MS-GNC-TLS converges, say, at iteration 6, whereas GNC-TLS does so only at iteration 30.

2. GNC-IRLS$_p$: Interpretation & Convergence

In this section we show that GNC-IRLS$_p$ is a convergent method, each iteration making steady progress towards minimizing (some majorizer of) the ℓ_p-loss. We first show GNC-IRLS$_p$ involves *two-level majorization* (Section 2.1). Then we state our convergence result (Section 2.2).



(a) Smooth Majorizer $\rho_\epsilon(r)$ (b) Quadratic Majorizer $q_\epsilon(r, u)$

Figure 1. Two majorizers of $\rho(r) = 2|r|^{0.5}$, ρ_ϵ (7) and q_ϵ (8).

2.1. Interpretation of GNC-IRLS$_p$

For two functions f and g defined on $\mathbb{R}$, if $f(r) \geq g(r)$ ($\forall r \in \mathbb{R}$), we say f *majorizes* g or f is a *majorizer* of g. Behind the apparent alternating nature of GNC-IRLS$_p$, it involves *two-level majorization*, as signified by the *smooth majorizer* and *quadratic majorizer*, introduced next.

Smooth Majorizer. As the main player in the first level of majorization, we define the *smooth majorizer* $\rho_\epsilon : \mathbb{R} \to \mathbb{R}_{\geq 0}$ for each $\epsilon > 0$ [52, 61] such that

$$\rho_\epsilon(r) = \begin{cases} \frac{1}{p}|r|^p, & |r| > \epsilon, \\ \frac{1}{2}\frac{r^2}{\epsilon^{2-p}} + (\frac{1}{p} - \frac{1}{2})\epsilon^p, & |r| \leq \epsilon. \end{cases} \tag{7}$$

The smooth majorizer ρ_ϵ is a Huber-like loss [35] which coincides with the ℓ_p-loss if $|r| \geq \epsilon$ and is otherwise quadratic in r. Figure 1a shows that ρ_ϵ majorizes the ℓ_p-loss for $p = 0.5$ and different values of ϵ. More formally, we have:

Lemma 1 ($\rho_\epsilon(\cdot)$ is Smooth ℓ_p-Majorizer)**.** *For $\rho(r) = \frac{1}{p}|r|^p$ and $\rho_\epsilon(r)$ defined in (7), the following holds: (i) $\rho_\epsilon(\cdot)$ is continuously differentiable, (ii) $\rho(r) \leq \rho_\epsilon(r), \forall r \in \mathbb{R}$, (iii) $\epsilon' \leq \epsilon \Rightarrow \rho_{\epsilon'}(r) \leq \rho_\epsilon(r)$, (iv) $\rho(r) = \lim_{\epsilon \to 0} \rho_\epsilon(r)$.*

Remark 1 (*Rethink Weight Update*). The weight update of Algorithm 1 coincides with rule (5) with $\rho = \rho_{\epsilon^{(t)}}$.

Remark 2 (*GNC for the ℓ_p-Loss*). Lemma 1 prompts a GNC strategy of minimizing ρ_ϵ (7) or even the ℓ_p-loss: decrease $\epsilon^{(t)}$ at each iteration t (Line 7, Algorithm 1).

Quadratic Majorizer. The smooth majorizer (7) is non-convex, and directly minimizing it can be hard. This is why the second level of majorization comes into play; the quadratic majorizer is the following quadratic function q_ϵ:

$$q_\epsilon(r, u) = \rho_\epsilon(u) + \frac{1}{2} \cdot \frac{r^2 - u^2}{\max\{|u|, \epsilon\}^{2-p}}. \tag{8}$$

Note that $q_\epsilon(r, u)$ is a shifted version of $\rho_\epsilon(u)$ by a carefully chosen amount, which makes $q_\epsilon(\cdot, u)$ into a majorizer of $\rho_\epsilon(\cdot)$. Indeed, Figure 1b shows that $q_{0.3}(\cdot, u)$ majorizes $\rho_{0.3}(\cdot)$ for $u = 0.1$ and 0.5. More formally, we have:

Lemma 2 ($q_\epsilon(\cdot, u)$ is Quadratic ℓ_p-Majorizer)**.** *With $\rho(r) = \frac{1}{p}|r|^p$, $\rho_\epsilon(r)$ and $q_\epsilon(r,u)$ defined respectively in* (7) *and* (8)*, we have $\rho_\epsilon(u) = q_\epsilon(u,u)$ and $\rho_\epsilon(r) \leq q_\epsilon(r,u)$, $\forall r, u \in \mathbb{R}$.*

Remark 3 (*Rethink Weighted Least-Squares*)*.* Recall $r_i^{(t)} := r(v^{(t)}, d_i)$. The WLS step (4) of Algorithm 1 minimizes the quadratic majorizer $\sum_{i=1}^m q_{\epsilon^{(t)}}\big(r(\cdot, d_i), r_i^{(t)}\big)$:

$$\begin{aligned} v^{(t+1)} &\in \operatorname*{argmin}_{v \in \mathcal{C}} \sum_{i=1}^m \frac{r(v, d_i)^2}{\max\{|r_i^{(t)}|, \epsilon^{(t)}\}^{2-p}} \\ &= \operatorname*{argmin}_{v \in \mathcal{C}} \sum\nolimits_{i=1}^m q_{\epsilon^{(t)}}\big(r(v, d_i), r_i^{(t)}\big) \end{aligned}$$

GNC-IRLS$_p$ differs from the *majorization-minimization* paradigm [27, 54, 63] in that, at different iterations, GNC-IRLS$_p$ minimizes different quadratic majorizers, as controlled by the smoothing parameter $\epsilon^{(t)}$; in so doing, it blends (quadratic) majorization-minimization with GNC.

2.2. Convergence of GNC-IRLS$_p$

To obtain convergence results, we need appropriate assumptions on the constraint set $\mathcal{C}$ and residual function r. The first assumption is standard (*cf*. [3, Section 4.2]):

Assumption 1. $\mathcal{C}$ is non-empty and closed. The residual function $r(v, d) : \mathcal{C} \times \mathcal{D} \to \mathbb{R}_{\geq 0}$ is *weakly coercive* in v:

$$\text{Either } \mathcal{C} \text{ is bounded or } \lim_{v \in \mathcal{C}, \|v\|_2 \to \infty} r(v, d) \to \infty. \tag{9}$$

Moreover, if $\|v\|_2 \neq \infty$ then $r(v, d) \neq \infty$.

The next assumption is about *differentiability*:

Assumption 2. The residual function $r(v, d)$ is continuous in v everywhere, and differentiable in v if $r(v, d) \neq 0$. Moreover, $r(v, d)^2$ is continuously differentiable in v.

Assumptions 1 and 2 are mild and easy to verify. With these assumptions, we prove the following:

Theorem 1 (Convergence of GNC-IRLS$_p$)**.** *Let $\{v^{(t)}\}_t$ be the iterates of Algorithm 1 with $\epsilon^{(t)}$ non-increasing and $\epsilon := \lim_{t\to\infty} \epsilon^{(t)} > 0$. Under Assumptions 1 and 2, every accumulation point of $\{v^{(t)}\}_t$ is a stationary point*[3] *of*

$$\min_{v \in \mathcal{C}} \sum\nolimits_{i=1}^m \rho_\epsilon\big(r(v, d_i)\big). \tag{10}$$

With a GNC schedule that creates a non-increasing sequence $\{\epsilon^{(t)}\}_t$ convergent to ϵ, GNC-IRLS$_p$ finds a stationary point of ρ_ϵ (Theorem 1), and ρ_ϵ approximates the ℓ_p-loss very well if ϵ is small (Lemma 1, Figure 1a). The convergence statement of "*accumulation points are stationary points*" in Theorem 1 is standard, and similar results can be found in optimization papers on IRLS or majorization-minimization, *e.g.*, [22, Thm 5.3 (ii)], [45, Thm 3.2], [54, Thm 1], [61, Thm 11], [47, Proposition 5], [42, Thm 1]. However, to our knowledge, Theorem 1 is the only result that holds for a *general* constraint set $\mathcal{C}$ and for minimizing *a sequence of majorizers* within the GNC framework.

Theorem 1 is proved by combing ideas of [22, 45] and [54], while generalizing their results. Unlike in Theorem 1, $\mathcal{C}$ is assumed to be convex and $\epsilon^{(t)} = \epsilon$ for all t in [54]. In [22, 45], $\mathcal{C}$ is defined by linear equality constraints and the residual function r is very specific, unlike in Theorem 1. Finally, as reviewed in Section 1.2, the result of [1] requires a condition that is hard to verify and their result does not apply to IRLS with the GNC strategy.

While stationary points are not necessarily local minimizers[3], convergence to them is perhaps the best one could guarantee in the setting where the objective (7) and constraint set $\mathcal{C}$ can both be non-convex. That said, a stronger convergence theory is possible given more assumptions on the problem and data. We will explore this in Section 3.

3. Convergence Rates for Robust Regression

While Theorem 1 is general, it does not reveal any convergence *speed*. Here, we compromise on generality and prove that GNC-IRLS$_p$ converges rapidly for *robust regression* [48]. Consider the following problem setup:

Problem 1 (Robust Regression)**.** For a *feature matrix* $\boldsymbol{A} = [\boldsymbol{a}_1, \dots, \boldsymbol{a}_m]^\top \in \mathbb{R}^{m \times n}$ and a *response vector* $\boldsymbol{y} = [y_1, \dots, y_m]^\top \in \mathbb{R}^m$, assume there is a ground truth vector $v^* = \boldsymbol{x}^* \in \mathbb{R}^n$ such that the residual vector $\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{y}$ has k non-zero entries; *i.e.*, there are k outliers and $m - k$ inliers among data $\{d_i\}_{i=1}^m = \{(\boldsymbol{a}_i, y_i)\}_{i=1}^m$. The goal of robust regression is to recover $\boldsymbol{x}^*$ from data $\boldsymbol{A}$ and $\boldsymbol{y}$.

In Problem 1 we assume all inliers $(\boldsymbol{a}_i, y_i)$ are noiseless, *i.e.*, $r(v^*, d_i) = |\boldsymbol{a}_i^\top \boldsymbol{x}^* - y_i| = 0$. The extension to the noisy case is not hard (*cf*. [46, Thm 2], [36, Thm A.1]).

The GNC schedule is closely related to the convergence rates of IRLS. Informally, [46] suggests that the *linear* GNC schedule (as is commonly seen) leads to a *linear* rate. However, it is possible for IRLS to attain *superlinear* rates. In particular, defining the *superlinear GNC schedule*

$$\epsilon^{(t+1)} \leftarrow \beta(\epsilon^{(t)})^{2-p}, \quad \beta > 0, \tag{11}$$

we prove the following result:

Theorem 2. *Assume $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ has i.i.d. $\mathcal{N}(0,1)$ entries. Initialize Algorithm 1 at $\boldsymbol{x}^{(0)}$ and $\epsilon^{(0)} > 0$ such that $\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^*\|_2 \leq \epsilon^{(0)}$. Denote by $r^*_{\min+}$ the smallest non-zero number among the set of residuals $\{|\boldsymbol{a}_i^\top \boldsymbol{x}^* - y_i|\}_{i=1}^m$. Define*

$$\alpha := \frac{\sqrt{5} \cdot 2^{2-p}}{0.99 \cdot 0.516} \cdot \frac{1}{\left(r^*_{\min+}\right)^{1-p}} \cdot \frac{\sqrt{k} \cdot \left(1.01\sqrt{k} + \sqrt{n}\right)}{(m-k)}. \tag{12}$$

[3] Stationary points are in the sense of [3, Section 5.3]; they satisfy a certain geometric condition that every local minimizer of (10) fulfills.

Then the iterates $\{\boldsymbol{x}^{(t)}\}_{t\geq 0}$ *produced by* `GNC-IRLS`$_p$ *with* $p \in [0,1]$[4] *and the GNC schedule* (11) *with* $\beta \geq \alpha$ *satisfy*

$$\|\boldsymbol{x}^{(t)} - \boldsymbol{x}^*\|_2 \leq \begin{cases} \beta^t \cdot \epsilon^{(0)} & p = 1 \\ \beta^{\frac{(2-p)^t - 1}{1-p}} \cdot (\epsilon^{(0)})^{(2-p)^t} & p \in [0,1) \end{cases} \tag{13}$$

with probability at least $1 - (P_0 + P_1 + P_2)$*, where*

$$\begin{aligned} &P_0 := \exp\big(-\tilde{\Omega}(n)\big), \;\; P_1 := \exp(-\tilde{\Omega}(k-n)), \\ &P_2 := \exp\big(-\tilde{\Omega}(m-k-n\log n)\big). \end{aligned} \tag{14}$$

We discuss several aspects of Theorem 2: **(i)** the probabilities (14), **(ii)** the condition $\beta \geq \alpha$ (12), **(iii)** its relation to prior works, and **(iv)** the interaction of the GNC schedule (11), error bound (13) and condition $\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^*\|_2 \leq \epsilon^{(0)}$.

(i) In the probability terms of (14), $\tilde{\Omega}$ stands for the standard big-Ω notation, with the difference that $\tilde{\Omega}$ also suppresses logarithmic terms. We wish P_0, P_1, P_2 of (14) to be small so that (13) holds with high probability. This is true whenever n is large (P_0), $k \gg n$ (P_1), and $m-k \gg n\log n$ (P_2). It seems counterintuitive to ask for the number k of outliers to be far larger than n, but the challenging case of Problem 1 occurs exactly when k is large. If k were small, then an alternative proof would give $P_1 = \exp(-\tilde{\Omega}(n))$. Such proof is much simpler, which is why we omit it.

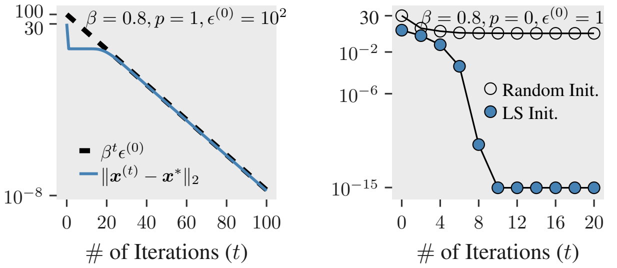
(ii) We wish α to be as small as possible as this would make it easier to set the factor β in the GNC schedule (11). Ignoring the values of the constants in (12), α mainly involves two terms, $r^*_{\text{min+}}$ and $O((k+\sqrt{kn})/(m-k))$. Since $r^*_{\text{min+}}$ measures the minimum residual of outliers at the ground truth $\boldsymbol{x}^*$, we expect it to be a *large* constant. Since $p \in [0,1]$, a large $(r^*_{\text{min+}})^{1-p}$ would make α small; on the other hand, for $p = 1$, α does not depend on $r^*_{\text{min+}}$ at all. Then note that we require a large k in **(i)**, but this might make the second term $O((k+\sqrt{kn})/(m-k))$ and therefore α very large. The rescue is in the denominator: α is small if the number $m-k$ of inliers (or the inlier rate) is large.

(iii) Theorem 2 is motivated by [46, Thm 1], over which we make some improvements. First, the GNC schedule of [46] sets $\epsilon^{(t+1)} \leftarrow \beta\epsilon^{(t)}$ if $\|\boldsymbol{x}^{(t+1)} - \boldsymbol{x}^{(t)}\|_2 \leq 2\beta\epsilon^{(t)}$, or otherwise $\epsilon^{(t+1)} \leftarrow \epsilon^{(t)}$. We simplify and generalize it into (11). Also, [46] is limited to the case $p = 1$, but Theorem 2 holds for any $p \in [0,1]$; we derive some technical lemmas that overcome the challenges of the non-convex case $p < 1$.

The final point **(iv)** has more delicate interpretations and ramifications, and we discuss it in Sections 3.1-3.4 next.

3.1. Global Linear Convergence at $p = 1$?

For the error bound $\|\boldsymbol{x}^{(t)} - \boldsymbol{x}^*\|_2 \leq \beta^t \cdot \epsilon^{(0)}$ of (13) to make sense, one needs to set $\beta < 1$, then the condition $\alpha \leq \beta$ in Theorem 2 implies $\alpha < 1$. As discussed, we have $\alpha < 1$ if the inlier rate is large. Indeed, assuming $k, m \gg n$ and bringing now the constant of (12) into the picture, we see that $\alpha < 1$ amounts to $m - k > 2.02\sqrt{5}/(0.99 \times 0.516)k$. This defines an outlier rate below which Theorem 2 holds. This also implies Theorem 2 is optimal in an information-theoretical sense (*e.g.*, it only requires m to be linear in k).



(a) Decay of Error (Bound) (13)

(b) Decay of Error $\|\boldsymbol{x}^{(t)} - \boldsymbol{x}^*\|_2$

Figure 2. (2a, Section 3.1): Error bound $\|\boldsymbol{x}^{(t)} - \boldsymbol{x}^*\|_2 \leq \beta^t\epsilon^{(0)}$ (13) with initialization $\boldsymbol{x}^{(0)} \sim \mathcal{N}(0, 100\boldsymbol{I}_n)$. (2b, Section 3.2): Errors of `GNC-IRLS`$_0$ at each iteration with least-squares versus random initialization. 100 trials, $k = 400, m = 1000, n = 10$.

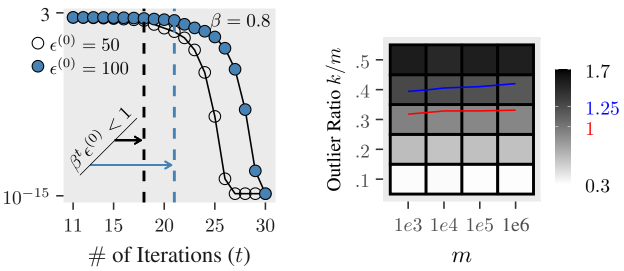
For $\|\boldsymbol{x}^{(t)} - \boldsymbol{x}^*\|_2 \leq \beta^t \cdot \epsilon^{(0)}$ to be true, Theorem 2 requires $\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^*\|_2 \leq \epsilon^{(0)}$ (among other assumptions). Given any initialization $\boldsymbol{x}^{(0)}$, one can choose a large $\epsilon^{(0)}$ such that $\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^*\|_2 \leq \epsilon^{(0)}$, so [46] claimed this is a *global* linear convergence. But this claim is imprecise, e.g., if $\boldsymbol{x}^{(0)}$ is the least-squares initialization and $\epsilon^{(0)}$ is larger than all residuals $|\boldsymbol{a}_i^\top \boldsymbol{x}^{(0)} - y_i|$, then all weights $w_i^{(1)}$ are equal to $\epsilon^{(0)}$, and we would get $\boldsymbol{x}^{(1)} = \boldsymbol{x}^{(0)}$. As such, the error would not decrease until $\epsilon^{(t)}$ becomes smaller: Figure 2a shows that $\|\boldsymbol{x}^{(t)} - \boldsymbol{x}^*\|_2$ "waits" for almost 20 iterations to decay together with the bound $\beta^t\epsilon^{(0)}$ ($\epsilon^{(0)} = 100$). This overlooked phenomenon caused by large $\epsilon^{(0)}$ is what we call a *burn-in* period. Interestingly, the burn-in period does not mean that our bound (13) is incorrect, but just that it might be loose for large $\epsilon^{(0)}$ in early iterations.

Figure 2a shows that `GNC-IRLS`$_1$ needs more than 100 iterations to reach machine accuracy. We improve this next, by considering $p \in [0,1)$ (Sections 3.2-3.4).

3.2. Local Quadratic Convergence at $p = 0$

Theorem 2 with $p \in [0,1)$ is better elaborated in the case $p = 0$, for which (13) gives $\|\boldsymbol{x}^{(t)} - \boldsymbol{x}^*\|_2 \leq (\beta\epsilon^{(0)})^{2^t}/\beta$. This corresponds to a quadratic convergence rate. Again, the error bound $(\beta\epsilon^{(0)})^{2^t}/\beta$ only makes sense if $\beta\epsilon^{(0)} < 1$, or if we set $\epsilon^{(0)}$ small (note that this time we do not require $\beta < 1$). In turn, Theorem 2 would demand an initialization $\boldsymbol{x}^{(0)}$ such that $\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^*\|_2 \leq \epsilon^{(0)}$. As corroborated by Figure 2b, `GNC-IRLS`$_0$ with random ("bad") initialization and small $\epsilon^{(0)}$ fails, but the least-squares initialization seems to be good enough, allowing `GNC-IRLS`$_0$ to converge at a quadratic rate, within 10 iterations, where "*the number of correct digits doubles at each iteration*" [14, Section 9.5.3].

[4] It is valid to run `GNC-IRLS`$_p$ with $p = 0$, as we justified in [52].



(a) Decay of $\|\boldsymbol{x}^{(t)} - \boldsymbol{x}^*\|_2$

(b) Error of Least-Squares

Figure 3. (3a, Section 3.3): From linear to quadratic rates, vertical lines indicating the transition takes place; $k = 400, m = 1000$, $\boldsymbol{x}^{(0)} \sim \mathcal{N}(0, \boldsymbol{I}_n)$. (3b, Section 3.4): Error $\|\boldsymbol{x}^\dagger - \boldsymbol{x}^*\|_2$ of the least-squares estimator $\boldsymbol{x}^\dagger$. We set 100 trials, $n = 10$.

Powerful as it might seem, quadratic (and superlinear) convergence is doomed to be local and in general cannot hold for all initializations (*cf*. Newton's method); we refer the reader to our prior work [52] for different insights into the quadratic rates of IRLS for robust regression.

We believe the *next best* convergence guarantees are these two: (i) prove that some IRLS variant has two-phase convergence, first global linear and then local quadratic, (ii) derive a suitable choice of $\epsilon^{(0)}$, β, and $\boldsymbol{x}^{(0)}$ for which quadratic convergence happens starting from the first iteration. We discuss these next in Section 3.3 and 3.4.

3.3. Graduated Rates From Linear to Quadratic?

Consider the following slight twist over Algorithm 1:

(a) With some initialization $\boldsymbol{x}^{(0)}$ and a sufficiently large $\epsilon^{(0)}$ such that $\|\boldsymbol{x}^{(0)} - \boldsymbol{x}^*\|_2 \leq \epsilon^{(0)}$, run Algorithm 1 with $p = 1$ and GNC schedule (11), until $\beta^t \epsilon^{(0)} < 1$. Theorem 2 suggests that $\|\boldsymbol{x}^{(t)} - \boldsymbol{x}^*\|_2 \leq \beta^t \epsilon^{(0)}$.

(b) Re-run Algorithm 1 with $\boldsymbol{x}^{(0)} := \boldsymbol{x}^{(t)}$, $\epsilon^{(0)} := \beta^t \epsilon^{(0)}$, $p = 0$, and schedule (11). Quadratic convergence (13) of Theorem 2 is now meaningful, since $\beta \epsilon^{(0)} < 1$.

Simply put, the above twist switches from $p = 1$ to $p = 0$ if $\beta^t \epsilon^{(0)} < 1$, resulting in a graduated rate from global linear to local quadratic. Such a graduated rate guarantee seems rare; we can only find it in [20]. Figure 3a shows that when we switch to $p = 0$, the convergence ensues in the next 10 iterations. A deficiency is that this twist also comes with a burn-in period (*cf*. Section 3.1), after which it is possible that the linear convergence phase is skipped and the quadratic convergence takes place directly (Figure 3a).

3.4. Quadratic Rates From The First Iteration?

The IRLS twist of Section 3.3 can take 30 iterations to converge if $\epsilon^{(0)}$ is large (Figure 3a). But we also saw that, with the least-squares initialization and $\beta \epsilon^{(0)} = 0.8 < 1$, GNC-IRLS$_0$ converges within 10 iterations, at a quadratic rate (Figure 2b). We now argue that it is theoretically possible for GNC-IRLS$_0$ to have quadratic rates *starting from as early as the first iteration*. For this, we first prove:

Proposition 1. *Assume $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ has i.i.d. $\mathcal{N}(0,1)$ entries with $m \geq n$. Let $\boldsymbol{x}^\dagger := (\boldsymbol{A}^\top \boldsymbol{A})^{-1} \boldsymbol{A}^\top \boldsymbol{y}$. With probability at least $1 - \exp(-\Omega(k)) - \exp(-\Omega(m))$, we have*

$$\|\boldsymbol{x}^\dagger - \boldsymbol{x}^*\|_2 \leq \frac{(1.01\sqrt{k} + \sqrt{n}) \cdot \|\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{y}\|_2}{(0.99\sqrt{m} - \sqrt{n})^2}. \tag{15}$$

We wish $\|\boldsymbol{x}^\dagger - \boldsymbol{x}^*\|_2 \leq 1$; if so we can set $\epsilon^{(0)} = 1$ and $\beta < 1$, achieving quadratic rates with initialization $\boldsymbol{x}^\dagger$ (Theorem 2). This is possible if k/m is small and $m, k \gg n$; see (15). This is also empirically confirmed in Figure 3b, where $\|\boldsymbol{x}^\dagger - \boldsymbol{x}^*\|_2 \leq 1$ for fewer than 30% outliers.

Implementation Details. The discussions so far suggest the following implementation of GNC-IRLS$_p$. Set $\epsilon^{(0)} = 1, p = 0$. Initialize it via least-squares. Set β smaller than 1; we always use $\beta = 0.8$. Theorem 1 suggests to let $\{\epsilon^{(t)}\}_t$ converge to some $\epsilon > 0$. In the noiseless case, we set $\epsilon = 10^{-16}$. Otherwise, if we are given an *inlier threshold* c such that $r(v^*, d_i) \leq c$ for all inliers d_i, then we set $\epsilon = c$.

4. Experiments: Lp Versus TLS

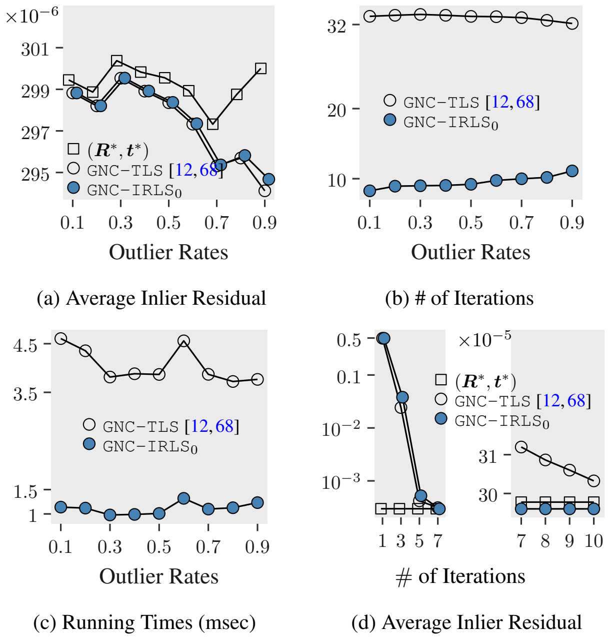
Here we compare GNC-TLS [12, 68] and GNC-IRLS$_p$. For more extensive experiments of IRLS and its variants, see, *e.g*., [1, 19, 24, 26, 40, 57, 58, 62, 68].

Experimental Setup. We contextualize our experiment in the application of *point cloud registration*. In this application, each sample d_i is a 3D point pair $(\boldsymbol{y}_i, \boldsymbol{x}_i)$, the variable v consists of a 3D rotation $\boldsymbol{R}$ and translation $\boldsymbol{t}$, and the residual function is $r(v; d_i) = \|\boldsymbol{y}_i - \boldsymbol{R}\boldsymbol{x}_i - \boldsymbol{t}\|_2$. The corresponding weighted least-squares problem (4) is solved by eliminating the translation first and then applying SVD [34].

Data. We randomly sample k outlier point pairs $(\boldsymbol{y}_j, \boldsymbol{x}_j)$ so that $\boldsymbol{y}_j \sim \mathcal{N}(0, \boldsymbol{I}_3)$ and $\boldsymbol{x}_j \sim \mathcal{N}(0, \boldsymbol{I}_3)$; here $\boldsymbol{I}_3$ denotes the 3×3 identity matrix. To get $m - k$ inlier pairs $(\boldsymbol{y}_i, \boldsymbol{x}_i)$, we randomly sample $\boldsymbol{x}_i$ from $\mathcal{N}(0, \boldsymbol{I}_3)$ and compute $\boldsymbol{y}_i = \boldsymbol{R}^*\boldsymbol{x}_i + \boldsymbol{t}^* + \boldsymbol{\epsilon}_i$. Here, $\boldsymbol{R}^*$ and $\boldsymbol{t}^*$ are randomly generated ground truth rotation and translation respectively, and $\boldsymbol{\epsilon}_i \sim \mathcal{N}(0, 0.01^2 \boldsymbol{I}_3)$ is some Gaussian noise. We set $c^2 = 0.01^2 \times 5.54^2$, so each inlier $(\boldsymbol{y}_i, \boldsymbol{x}_i)$ satisfies $\|\boldsymbol{y}_i - \boldsymbol{R}^*\boldsymbol{x}_i - \boldsymbol{t}^*\|_2 \leq c^2$ with probability $\geq 1 - 10^{-6}$.

Metric. Given a rotation $\boldsymbol{R}$, translation $\boldsymbol{t}$, and ground truth inlier index set $\mathcal{I}^*$, we can calculate the *average inlier residual* $\sum_{i \in \mathcal{I}^*} \|\boldsymbol{y}_i - \boldsymbol{R}\boldsymbol{x}_i - \boldsymbol{t}\|_2 / (m - k)$. This is used to measure the errors made by the algorithms to evaluate.

Results. As the outlier rate varies from 10% to 90%, GNC-IRLS$_0$ and GNC-TLS entail almost the same average inlier residual (Figure 4a). Their errors are even smaller



(a) Average Inlier Residual

(b) # of Iterations

(c) Running Times (msec)

(d) Average Inlier Residual

Figure 4. Comparison of GNC-IRLS$_0$ and GNC-TLS for point cloud registration. 100 trials, $m = 1000$. 4d: 900/1000 outliers.

than those at the ground truth $(\boldsymbol{R}^*, \boldsymbol{t}^*)$, which suggests that the performance of both algorithms cannot be further improved for such experiments. But note that they could fail for more than 900/1000 outliers (which was reported in prior works, so we did not provide a plot here) and that the breakdown points will change for different data distributions and different geometric problems.

What can actually be improved is the convergence rate: GNC-IRLS$_0$ terminates in 10 iterations, while GNC-TLS takes 32 (Figure 4b), indicating that GNC-IRLS$_0$ is 3 times faster (Figure 4c). For fair[5] comparison, both methods are implemented to terminate under the same condition, that is whenever the difference of the minimum values of (4) between two consecutive iterations is smaller than 10^{-10} (other thresholds, *e.g.*, $10^{-6}, 10^{-16}$, lead to similar results).

The errors of GNC-TLS decrease as fast as GNC-IRLS$_0$ (Figure 4d, left). At first glance, this seems counterintuitive because GNC-TLS comes with a linear GNC schedule (*cf*. Section 1.2) and is thus expected to converge linearly (*cf*. [46]), as opposed to the quadratic rate of GNC-IRLS$_0$ (*cf*. Theorem 2). With hindsight, this might be a natural consequence of the weighting strategy of GNC-TLS (*cf*. [68, Eq. (14)], (5), (6)): Weight 0 is set if the residual is *particularly large*, and this could completely rule out some *obvious* outliers at early iterations (and similarly for *particularly small* residuals), resulting in a fast decrease of errors. But this weighting scheme brings diminishing gains in later iterations, where the errors of GNC-TLS decrease only linearly (Figure 4d, right). The final observation is that GNC-IRLS$_0$ reaches an error smaller than that of $(\boldsymbol{R}^*, \boldsymbol{t}^*)$ at iteration 7, but it requires a few more iterations to terminate (similarly for GNC-TLS). This implies the termination criterion is sub-optimal (it is hard to design a provably better one).

Finally, Figures 4b and 4d show that GNC-TLS has an error of $\leq 10^{-3}$ already at iteration 10, but, unnecessarily, it terminates at iteration 32. We improve this in Section 5, without even changing the termination criterion.

5. MS-GNC-TLS: Improving GNC-TLS

> *... it indicates that GNC can fail, and that there is therefore no point in looking for a general proof of correctness.*
>
> Andrew Blake & Andrew Zisserman [12]

In this section, we improve GNC-TLS [12, 68] from two aspects, as respectively motivated by two ideas that we have developed for the ℓ_p-loss, namely *majorization* (Section 2) and *superlinear GNC schedule* (Section 3). Majorization guarantees a monotonic decrease of the objective and the eventual convergence (*cf*. Theorem 1), and the superlinear GNC schedule speeds up convergence (*cf*. Theorem 2).

Majorization. To motivate the need for majorizing the TLS loss $\rho(r) = \min\{r^2, c^2\}$, recall GNC-TLS uses ρ_μ (6) to approximate $\rho(\cdot)$. The issue is that $\rho_\mu(\cdot)$ relaxes $\rho(\cdot)$ and approximates it *from below*, and hence $\rho_\mu(r) \leq \rho(r), \forall \mu > 0$ (Figure 5a). This makes a convergence analysis difficult.

We propose the following smooth function

$$\overline{\rho}_\mu(r) = \begin{cases} r^2, & \text{if } |r| \leq c, \\ \frac{\mu+1}{\mu} c^2, & \text{if } |r| \geq \frac{\mu+1}{\mu} c, \\ -\mu r^2 + 2(1+\mu)c|r| - (1+\mu)c^2, & \text{o/w}, \end{cases} \tag{16}$$

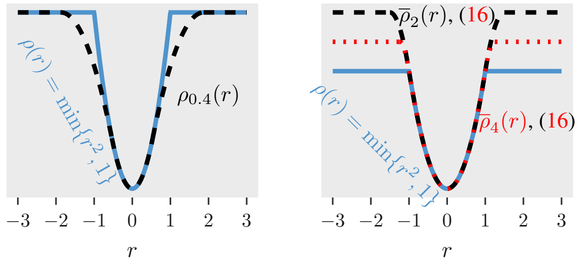
to majorize the TLS loss $\rho(r)$; see Figure 5b. Since both $\rho_\mu(r)$ (6) and $\overline{\rho}_\mu(r)$ approach $\rho(r)$ as $\mu \to \infty$, one might expect comparable performance. However, a crucial difference is that, with the majorizer $\overline{\rho}_\mu(r)$, convergence guarantees easily ensue. Indeed, $\overline{\rho}_\mu(r)$ is akin to the smooth majorizer (7) of the ℓ_p-loss, and one could construct a quadratic majorizer for $\overline{\rho}_\mu(r)$, which enables an IRLS + GNC scheme (*cf*. Remarks 1-3, Section 1.2). In particular, this IRLS variant involves (i) weight update using (5) with $\rho = \overline{\rho}_{\mu^{(t)}}$, *i.e.*,

$$w_i^{(t+1)} = \begin{cases} 1, & \text{if } r_i^{(t)} \leq c, \\ 0, & \text{if } r_i^{(t)} \geq \frac{\mu^{(t)}+1}{\mu^{(t)}} c, \\ \frac{c(1+\mu^{(t)})}{r_i^{(t)}} - \mu^{(t)}, & \text{o/w}, \end{cases} \tag{17}$$

and (ii) updating $\mu^{(t+1)}$ based on some GNC schedule.

We prove the following result to accompany Theorem 1.

[5] It is slightly unfair to GNC-IRLS$_0$ as its weights are typically larger.



(a) Approximation $\rho_\mu(r)$ of [12]

(b) Smooth Majorizer (16)

Figure 5. The TLS loss $\rho(r)$ and its surrogates.

Theorem 3 (Convergence of majorized GNC-TLS). *Let $\{v^{(t)}\}_t$ be the iterates of IRLS with weight update (17) and a GNC schedule $\{\mu^{(t)}\}_t$. Assume $\{v^{(t)}\}_t$ is bounded, i.e., $\|v^{(t)}\|_2 < \infty$ ($\forall t$). Suppose $\{\mu^{(t)}\}_t$ is non-decreasing and converges to $\mu < \infty$. Under Assumptions 1 and 2, every accumulation point of $\{v^{(t)}\}_t$ is a stationary point of*

$$\min_{v \in \mathcal{C}} \sum\nolimits_{i=1}^{m} \bar{\rho}_\mu\big(r(v, d_i)\big). \tag{18}$$

Superlinear Schedule. Motivated by (11) (with $p = 0$) and discussions in Sections 3.2-3.3, we propose the update rule

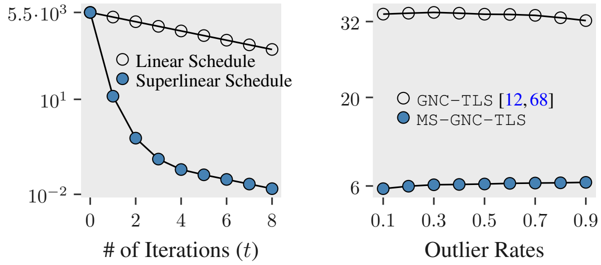
$$\mu^{(t+1)} \leftarrow \begin{cases} \gamma\sqrt{\mu^{(t)}} & \mu^{(t)} \leq 1 \\ \gamma\mu^{(t)} & \mu^{(t)} > 1 \end{cases}, \quad \gamma > 1, \tag{19}$$

as our GNC schedule. Denote by MS-GNC-TLS the resulting IRLS method that optimizes (16) with schedule (19).

The intuition behind the superlinear schedule (19) is as follows. With (19), the interval $(c, c + c/\mu^{(t)})$ of (17) that produces non-binary weights shrinks faster than the linear schedule $\mu^{(t+1)} \leftarrow \gamma\mu^{(t)}$ (Figure 6a), thus the superlinear schedule makes it happen earlier that all weights become binary, which is a good indicator for convergence. Note though that this argument does not prove (MS-)GNC-TLS converges, as it does not exclude the case that (MS-)GNC-TLS could produce different binary weights at consecutive iterations (*cf*. [10] and [4, Thm 15]).

Under the setting of Figure 4c, MS-GNC-TLS takes 6 iterations to converge (Figure 6b). It is even faster than GNC-IRLS$_0$ as it benefits from combining soft and hard thresholding (17). In this experiment, MS-GNC-TLS and GNC-TLS result in basically the same error upon convergence; it is just that GNC-TLS does not monotonically decrease the objective, and that its linear GNC schedule is more conservative than the proposed superlinear one.

Implementation Details. With the superlinear schedule (19), $\mu^{(t)}$ increases very fast, so one could set $\mu^{(0)} \leftarrow 10^{-15}$ such that MS-GNC-TLS can still terminate within 10 iterations. However, schedule (19) is *double-edged*: If $\mu^{(t)}$ increases so fast that all residuals are larger than $\frac{\mu^{(t)}+1}{\mu^{(t)}}c$, then all weights would be zero as per (17) and MS-GNC-TLS might fail. Fortunately, this situation can be prevented if we *slow down*: replace $\mu^{(t+1)} \leftarrow \gamma\sqrt{\mu^{(t)}}$ of (19) with $\mu^{(t+1)} \leftarrow \gamma(\mu^{(t)})^{1/(2-p)}$ for a larger $p \in (0, 1]$.



(a) $c/\mu^{(t)}$

(b) # of Iterations (t)

Figure 6. 6a: Length $c/\mu^{(t)}$ of the interval that corresponds to non-binary weights (17) with $c = 0.0554$, $\mu^{(0)} = 10^{-5}$, $\gamma = 1.4$. 6b: Number of iterations at which the algorithms terminate.

6. Conclusion, Limitations, and Future Work

Conclusion. While IRLS and GNC have often been viewed as different techniques [38,39,57,68,72], we reconcile them with an emphasis on a theoretical understanding of convergence properties and their relation with GNC schedules. Two messages are (i) that a majorization strategy should be constructed for guaranteeing *convergence* (Theorems 1 and 3), and (ii) that a superlinear GNC schedule should be considered for guaranteeing *convergence rates* (Theorem 2).

Limitations & Future Work. IRLS and its variants would break down if the number $m - k$ of inliers is close to the number n of variables, say if $m - k < 3n$. For geometric vision problems, n is small (*e.g.*, $n = 6$ for point cloud registration), so IRLS might fail if, for example, $m - k < 18$. In fact, for small m, other methods (*e.g.*, RANSAC [7, 28], outlier removal [50], or semidefinite relaxations [31, 51, 70]) are efficient, accurate, and are thus recommended.

A limitation of the TLS loss $\rho(r) = \min\{r^2, c^2\}$ is the need to choose a threshold parameter c. Ideally, it should be chosen as small as possible but larger than every inlier residual; see [53] for a related discussion. Prior works [4, 62] tried to dispense with c^2, but it was at the expense of introducing other parameters. This issue might be solved by changing c in a GNC style at each iteration, which implies future work of designing a GNC schedule for c and studying its interplay with another GNC parameter $\mu^{(t)}$. On the theory side, we note that extending the analysis of Theorem 2 beyond ℓ_p-losses remains to be studied in future work.

Acknowledgements. This work was supported by grants NSF 1704458, NSF 1934979, ONR MURI 503405-78051, and the Northrop Grumman Mission Systems Research in Applications for Learning Machines (REALM) initiative.

References

[1] Khurrum Aftab and Richard Hartley. Convergence of iteratively re-weighted least squares to robust M-estimators. In *IEEE Winter Conference on Applications of Computer Vision*, 2015. 1, 2, 3, 4, 6

[2] Chris Aholt, Sameer Agarwal, and Rekha Thomas. A QCQP approach to triangulation. In *European Conference on Computer Vision*, 2012. 2

[3] Niclas Andréasson, Anton Evgrafov, and Michael Patriksson. *An Introduction to Continuous Optimization: Foundations and Fundamental Algorithms*. Courier Dover Publications, 2020. 4

[4] Pasquale Antonante, Vasileios Tzoumas, Heng Yang, and Luca Carlone. Outlier-robust estimation: Hardness, minimally tuned algorithms, and applications. *IEEE Transactions on Robotics*, 2021. 1, 2, 8

[5] K Somani Arun, Thomas S Huang, and Steven D Blostein. Least-squares fitting of two 3D point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (5):698–700, 1987. 2

[6] Erik Ask, Olof Enqvist, and Fredrik Kahl. Optimal geometric fitting under the truncated L_2-norm. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 1

[7] Daniel Barath, Jana Noskova, Maksym Ivashechkin, and Jiri Matas. MAGSAC++, a fast, reliable and accurate robust estimator. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 8

[8] Amir Beck. On the convergence of alternating minimization for convex programming with applications to iteratively reweighted least squares and decomposition schemes. *SIAM Journal on Optimization*, 25(1):185–209, 2015. 2

[9] Amir Beck and Shoham Sabach. Weiszfeld's method: Old and new results. *Journal of Optimization Theory and Applications*, 164(1):1–40, 2015. 2

[10] Kush Bhatia, Prateek Jain, and Purushottam Kar. Robust regression via hard thresholding. *Advances in Neural Information Processing Systems*, 2015. 2, 8

[11] Michael J Black and Anand Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *International Journal of Computer Vision*, 19(1):57–91, 1996. 1

[12] Andrew Blake and Andrew Zisserman. *Visual Reconstruction*. MIT Press, 1987. 2, 6, 7, 8

[13] Roger Joseph Boscovich. De litteraria expeditione per pontificiam ditionem, et synopsis amplioris operis, ac habentur plura ejus ex exemplaria etiam sensorum impessa. *Bononiensi Scientiarum et Artum Instuto Atque Academia Commentarii*, 4:353–396, 1757. 1

[14] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. 5

[15] Jesus Briales and Javier Gonzalez-Jimenez. Convex global 3D registration with lagrangian duality. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[16] Tony F Chan and Pep Mulet. On the convergence of the lagged diffusivity fixed point method in total variation image restoration. *SIAM Journal on Numerical Analysis*, 36(2):354–367, 1999. 2

[17] Rick Chartrand. Exact reconstruction of sparse signals via nonconvex minimization. *IEEE Signal Processing Letters*, 14(10):707–710, 2007. 1

[18] Rick Chartrand and Wotao Yin. Iteratively reweighted algorithms for compressive sensing. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008. 2

[19] Avishek Chatterjee and Venu Madhav Govindu. Robust relative rotation averaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):958–972, 2017. 1, 2, 6

[20] Bintong Chen and Naihua Xiu. A global linear and local quadratic noninterior continuation method for nonlinear complementarity problems based on Chen–Mangasarian smoothing functions. *SIAM Journal on Optimization*, 9(3):605–623, 1999. 6

[21] David Coleman, Paul Holland, Neil Kaden, Virginia Klema, and Stephen C Peters. A system of subroutines for iteratively reweighted least squares computations. *ACM Transactions on Mathematical Software*, 6(3):327–336, 1980. 1, 2

[22] Ingrid Daubechies, Ronald DeVore, Massimo Fornasier, and C Sinan Güntürk. Iteratively reweighted least squares minimization for sparse recovery. *Communications on Pure and Applied Mathematics*, 63(1):1–38, 2010. 1, 2, 3, 4

[23] DQF De Menezes, Diego Martinez Prata, Argimiro R Secchi, and José Carlos Pinto. A review on robust M-estimators for regression analysis. *Computers & Chemical Engineering*, 147:107254, 2021. 1

[24] Tianjiao Ding, Yunchen Yang, Zhihui Zhu, Daniel P Robinson, René Vidal, Laurent Kneip, and Manolis C Tsakiris. Robust homography estimation via dual principal component pursuit. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2, 6

[25] Tianyu Ding, Zhihui Zhu, Tianjiao Ding, Yunchen Yang, René Vidal, Manolis C. Tsakiris, and Daniel Robinson. Noisy dual principal component pursuit. In *International Conference on Machine Learning*, 2019. 2

[26] Wenhua Dong, Xiao-jun Wu, and Josef Kittler. Sparse subspace clustering via smoothed ℓ_p minimization. *Pattern Recognition Letters*, 125:206–211, 2019. 1, 2, 6

[27] Taosha Fan and Todd Murphey. Majorization minimization methods for distributed pose graph optimization with convergence guarantees. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020. 4

[28] Martin A Fischler and Robert C Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. 2, 8

[29] Carl Friedrich Gauss. *Theoria motus corporum coelestium in sectionibus conicis solem ambientium auctore Carolo Friderico Gauss*. sumtibus Frid. Perthes et IH Besser, 1809. 1

[30] Frank R Hampel. The breakdown points of the mean combined with some rejection rules. *Technometrics*, 27(2):95–107, 1985. 1

[31] Linus Härenstam-Nielsen, Niclas Zeller, and Daniel Cremers. Semidefinite relaxations for robust multiview triangulation. Technical report, arXiv:2301.11431 [cs.CV], 2023. 8

[32] Richard Hartley. Tutorial notes on IRLS. http://users.cecs.anu.edu.au/~hartley/Papers/PDF/Hartley:IRLS14.pdf, 2014. Accessed: September 2022. 2

[33] Joel A Hesch and Stergios I Roumeliotis. A direct least-squares (DLS) method for PnP. In *International Conference on Computer Vision*, 2011. 2

[34] Berthold KP Horn, Hugh M Hilden, and Shahriar Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America A*, 5(7):1127–1135, 1988. 2, 6

[35] Peter J Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964. 1, 3

[36] Christian Kümmerle, Claudio Mayrink Verdun, and Dominik Stöger. Iteratively reweighted least squares for basis pursuit with global linear convergence rate. *Advances in Neural Information Processing Systems*, 2021. 1, 2, 4

[37] Christian Kümmerle and Claudio M Verdun. A scalable second order method for ill-conditioned matrix completion from few samples. In *International Conference on Machine Learning*, 2021. 1

[38] Huu Le and Christopher Zach. A graduated filter method for large scale robust estimation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. 8

[39] Huu Le and Christopher Zach. Robust fitting with truncated least squares: A bilevel optimization approach. In *International Conference on 3D Vision*, 2021. 2, 8

[40] Seong Hun Lee and Javier Civera. HARA: A hierarchical approach for robust rotation averaging. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 6

[41] Adrien Marie Legendre. *Nouvelles méthodes pour la détermination des orbites des comètes: avec un supplément contenant divers perfectionnemens de ces méthodes et leur application aux deux comètes de 1805*. Courcier, 1806. 1

[42] Gilad Lerman and Tyler Maunu. Fast, robust and non-convex subspace recovery. *Information and Inference: A Journal of the IMA*, 7(2):277–336, 2018. 2, 4

[43] Gilad Lerman, Michael B. McCoy, Joel A. Tropp, and Teng Zhang. Robust computation of linear models by convex relaxation. *Foundations of Computational Mathematics*, 15(2):363–410, 2015. 2

[44] Goran Marjanovic and Victor Solo. On ℓ_q optimization and matrix completion. *IEEE Transactions on Signal Processing*, 60(11):5714–5724, 2012. 1

[45] Karthik Mohan and Maryam Fazel. Iterative reweighted algorithms for matrix rank minimization. *The Journal of Machine Learning Research*, 13(1):3441–3473, 2012. 1, 3, 4

[46] Bhaskar Mukhoty, Govind Gopakumar, Prateek Jain, and Purushottam Kar. Globally-convergent iteratively reweighted least squares for robust regression problems. In *International Conference on Artificial Intelligence and Statistics*, 2019. 2, 3, 4, 5, 7

[47] Peter Ochs, Alexey Dosovitskiy, Thomas Brox, and Thomas Pock. On iteratively reweighted algorithms for nonsmooth nonconvex optimization in computer vision. *SIAM Journal on Imaging Sciences*, 8(1):331–372, 2015. 2, 4

[48] Dianne P O'Leary. Robust regression computation using iteratively reweighted least squares. *SIAM Journal on Matrix Analysis and Applications*, 11(3):466–480, 1990. 1, 4

[49] Frank C Park and Bryan J Martin. Robot sensor calibration: Solving $AX = XB$ on the Euclidean group. *IEEE Transactions on Robotics and Automation*, 10(5):717–721, 1994. 2

[50] Álvaro Parra Bustos and Tat-Jun Chin. Guaranteed outlier removal for point cloud registration with correspondences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2868–2882, 2018. 8

[51] Liangzu Peng, Mahyar Fazlyab, and René Vidal. Semidefinite relaxations of truncated least-squares in robust rotation search: Tight or not. In *European Conference on Computer Vision*, 2022. 8

[52] Liangzu Peng, Christian Kümmerle, and René Vidal. Global linear and local superlinear convergence of IRLS for non-smooth robust regression. In *Advances in Neural Information Processing Systems*, 2022. 2, 3, 5, 6

[53] Liangzu Peng, Manolis C. Tsakiris, and René Vidal. ARCS: Accurate rotation and correspondences search. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 8

[54] Meisam Razaviyayn, Mingyi Hong, and Zhi-Quan Luo. A unified convergence analysis of block successive minimization methods for nonsmooth optimization. *SIAM Journal on Optimization*, 23(2):1126–1153, 2013. 3, 4

[55] William J.J. Rey. *Introduction to Robust and Quasi-Robust Statistical Methods*. Springer Science & Business Media, 1983. 1

[56] David M Rosen, Luca Carlone, Afonso S Bandeira, and John J Leonard. SE-Sync: A certifiably correct algorithm for synchronization over the special Euclidean group. *The International Journal of Robotics Research*, 38(2-3):95–125, 2019. 2

[57] Jingnan Shi, Heng Yang, and Luca Carlone. Optimal and robust category-level perception: Object pose and shape estimation from 2D and 3D semantic keypoints. Technical report, arXiv:2206.12498 [cs.CV], 2022. 2, 6, 8

[58] Yunpeng Shi and Gilad Lerman. Message passing least squares framework and its application to rotation synchronization. In *International Conference on Machine Learning*, 2020. 6

[59] Chitturi Sidhartha and Venu Madhav Govindu. It is all in the weights: Robust rotation averaging revisited. In *International Conference on 3D Vision*, 2021. 2, 3

[60] Torbjorn Smith and Olav Egeland. Dynamical pose estimation with graduated non-convexity for outlier robustness. *Modeling, Identification and Control*, 43(2):79–89, 2022. 2

[61] Junxiao Song, Prabhu Babu, and Daniel P Palomar. Sparse generalized eigenvalue problem via smooth optimization. *IEEE Transactions on Signal Processing*, 63(7):1627–1642, 2015. 3, 4

[62] Lei Sun. IMOT: General-purpose, fast and robust estimation for spatial perception problems with outliers. Technical report, arXiv:2204.01324v1 [cs.CV], 2022. 2, 6, 8

[63] Ying Sun, Prabhu Babu, and Daniel P Palomar. Majorization-minimization algorithms in signal processing, communications, and machine learning. *IEEE Transactions on Signal Processing*, 65(3):794–816, 2016. 4

[64] Manolis C. Tsakiris and René Vidal. Dual principal component pursuit. *Journal of Machine Learning Research*, 19(18):1–50, 2018. 2

[65] Sergey Voronin and Ingrid Daubechies. An iteratively reweighted least squares algorithm for sparse regularization. Technical report, arXiv:1511.08970v3 [math.NA], 2015. 2

[66] Endre Weiszfeld. Sur le point pour lequel la somme des distances de n points donnés est minimum. *Tohoku Mathematical Journal*, 43:355–386, 1937. 2

[67] David Wipf and Srikantan Nagarajan. Iterative reweighted ℓ_1 and ℓ_2 methods for finding sparse solutions. *IEEE Journal of Selected Topics in Signal Processing*, 4(2):317–329, 2010. 2

[68] Heng Yang, Pasquale Antonante, Vasileios Tzoumas, and Luca Carlone. Graduated non-convexity for robust spatial perception: From non-minimal solvers to global outlier rejection. *IEEE Robotics and Automation Letters*, 5(2):1127–1134, 2020. 1, 2, 6, 7, 8

[69] Heng Yang and Luca Carlone. One ring to rule them all: Certifiably robust geometric perception with outliers. In *Advances in Neural Information Processing Systems*, 2020. 2, 3

[70] Heng Yang and Luca Carlone. Certifiable outlier-robust geometric perception: Exact semidefinite relaxations and scalable global optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 8

[71] Heng Yang, Jingnan Shi, and Luca Carlone. TEASER: Fast and certifiable point cloud registration. *IEEE Transactions on Robotics*, 37(2):314–333, 2021. 2, 3

[72] Christopher Zach and Guillaume Bourmaud. Descending, lifting or smoothing: Secrets of robust cost optimization. In *European Conference on Computer Vision*, 2018. 8

[73] Ji Zhao. An efficient solution to non-minimal case essential matrix estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 2

[74] Pengwei Zhou, Xuexun Guo, Xiaofei Pei, and Ci Chen. T-TOAM: Truncated least squares Lidar-only odometry and mapping in real time. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–13, 2021. 1, 2

[75] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. In *European Conference on Computer Vision*, 2016. 2