Approximations to worst-case data dropping: unmasking failure modes

Jenny Y. Huang, David R. Burt, Tin D. Nguyen, Yunyi Shen, Tamara Broderick
MIT EECS and MIT-IBM Watson AI Lab
{jhuang9,dburt,tdn,yshen99,tbroderick}@mit.edu

Abstract

A data analyst would worry about generalization if dropping a very small fraction of data points from a study could change its substantive conclusions. Finding the worst-case data subset to drop poses a combinatorial optimization problem. To overcome this intractability, recent works propose using additive approximations, which treat the contribution of a collection of data points as the sum of their individual contributions, and greedy approximations, which iteratively select the point with the highest impact to drop and re-runs the data analysis without that point [Broderick et al., 2020, Kuschnig et al., 2021]. We identify that, even in a setting as simple as OLS linear regression, many of these approximations can break down in realistic data arrangements. Several of our examples reflect masking, where one data point may hide or conceal the effect of another data point. We provide recommendations for users and suggest directions for future development.

1 Introduction

Researchers typically run a data analysis with the goal of applying conclusions of the analysis in the future. For instance, economists run randomized controlled trials (RCTs) of microcredit with a particular set of people. If the resulting data analysis demonstrates that microcredit increases business profit, a policymaker might distribute microcredit to people in the future, on the assumption that microcredit will help these people too. We might worry whether this assumption is warranted if we could drop a very small fraction of people from the original trial and instead conclude that microcredit decreases business profit. As a concrete example, Broderick et al. [2020] show that it is possible to drop 15 households out of over 16,500 in an influential microcredit RCT and change the result to a statistically significant conclusion of the opposite sign.

In many cases, then, it behooves us to check: can we find a small fraction of data that, if dropped, would change the conclusion of the analysis? A brute force approach to answering this question requires enumerating every possible small data subset and re-running the analysis a combinatorially large number of times. E.g., running $\binom{16500}{15}$ 1-second-long data analyses would take over 10^{43} years. This brute force approach is computationally prohibitive [Moitra and Rohatgi, 2023].

Broderick et al. [2020] instead suggest using an approximation based on instantiating continuous weights on the data points and differentiating with respect to these weights. The authors use this approximation to identify small data subsets that, when dropped, change conclusions in multiple landmark papers in economics [e.g. Angelucci and De Giorgi, 2009, Finkelstein et al., 2012]. In follow up work focused on linear regression with ordinary least squares (OLS), Kuschnig et al. [2021] introduced two additional ideas for finding the worst-case data subset: (1) approximating the impact of removing a group of points by the sum of the impacts of exactly removing individual data points and (2) greedily removing one data point at a time.

Preprint. Under review.

Recently, scientists and social scientists have used these approximations to assess the robustness of important findings in econometrics [Martinez, 2022], epidemiology [Di and Xu, 2022], and the social sciences [Davies et al., 2024, Burton and Roach, 2023]. Given the deployment of these approximations in practice, we ask when and how they can fail in realistic data analyses – to alert practitioners and motivate further approximation development. Previous works have identified particular instances of failure modes, without a full characterization of the explanation behind the failures [Broderick et al., 2020, Nguyen et al., 2024]. Other works have illustrated failure modes in adversarial constructions or settings where a large fraction of the data needed to be removed [Moitra and Rohatgi, 2023, Freund and Hopkins, 2023].

In the present work, we systematically explore failure modes for approximations to dropping worst-case very-small fractions (< 1%) of data. We focus on natural data settings with no adversary. Like [Kuschnig et al., 2021, Moitra and Rohatgi, 2023, Freund and Hopkins, 2023], we focus on linear regression fit with OLS. Before the present work and contemporaneous work by Hu et al. [2024], there had not been studies systematically characterizing non-adversarial failure modes of approximations to the Most Influential Set and comparing the prominent existing approximations.

In many aspects, Hu et al. [2024] and our present work are complementary. For instance, while Hu et al. [2024] focus on exact recovery of the most influential data subset with size at most equal to a stated value, we focus on finding whether there exists a small subset of data that could change substantive conclusions. Given our different focuses, Hu et al. [2024] finds it useful to separate masking into two phenomena: amplification and cancellation. Meanwhile, we find it useful to point out failure modes due to poor conditioning of the bulk of data. Hu et al. [2024]'s theory assumes a particular data-generating process [their Equation 7]; while we don't make this assumption, we instead need to take a limit of an outlier data point's position to derive our results (Proposition D.1).

Both Hu et al. [2024] and our work focus on OLS for theory and illustration of failure modes. While linear regression is less common in engineering disciplines, Castro Torres and Akbaritabar [2024] demonstrate that, as recently as 2022, (often well) over half of all papers reporting any methods in Medical and Health Sciences, Agricultural Sciences, Social Sciences, and the Humanities use linear regression. Indeed, most of the applied papers cited in the discussion above use linear regression [Angelucci and De Giorgi, 2009, Finkelstein et al., 2012, Martinez, 2022, Davies et al., 2024, Burton and Roach, 2023]. We suspect OLS is the most common form of linear regression used in practice. In the present work, we identify failure modes in cases of OLS-fit linear regression with one covariate. In light of the surfaced failure modes, we end with some recommendations for the user and directions for future work.

2 Setup

We formally describe the linear regression problem we consider and review data dropping. We observe N data points, $d_{1:N}$, where $d_n = (x_n, y_n)$ consists of covariates $x_n \in \mathbb{R}^P$ and response $y_n \in \mathbb{R}$. We estimate an unknown parameter θ by minimizing a sum of losses f: $\hat{\theta} = \arg\min_\theta \sum_{n=1}^N f(d_n, \theta)$. All of our examples focus on OLS solutions to linear regression with $P = 2$ including an all-ones covariate, so $f(d_n, \theta) = (y_n - \theta_1 x_{n,1} - \theta_0)^2$.

We might be concerned if dropping a small fraction $\alpha \in (0, 1)$ of our data changed our substantive conclusions. The value of α is user-defined; we focus on $\alpha = 0.01$ (i.e., 1% of the data) as a small fraction; Broderick et al. [2020] pointed out several influential economics papers could be overturned by removing less than 1% of the sample. Broderick et al. [2020] define the *Maximum Influence Perturbation* as the largest possible change induced in some quantity of interest by dropping at most α fraction of the data. We focus on decisions made from the sign of the pth parameter dimension, so our quantity of interest will be the effect size θ_p. Without loss of generality, we assume $\hat{\theta}_p(w) > 0$, and we ask whether we can change the result to a negative sign. Our examples focus on the slope, θ_1. Our development before the examples allows a more general loss f and quantity of interest θ_p.

To write the optimization problem formally (see Equation (1)), let w_n represent a weight on the nth data point, and collect $w = w_{1:N}$. Define $\hat{\theta}(w) := \arg\min_\theta \sum_{n=1}^N w_n f(\theta, d_n)$. Setting $w = 1_N$, the all-ones vector of length N, recovers the original data analysis, and setting w_n to zero corresponds to dropping the nth point. We collect all weightings that correspond to dropping at most α fraction of the data in $W_\alpha := \{w \in \{0, 1\}^N : \sum_{n=1}^N (1 - w_n) \leq \alpha N\}$. Finally, the Maximum Influence

Perturbation in our case[1] can be written

$$\max_{w \in W_\alpha} \hat{\theta}_p(1_N) - \hat{\theta}_p(w). \tag{1}$$

W_α has $\binom{N}{\lfloor \alpha N \rfloor}$ elements. The *Most Influential Set* is defined to be the set of dropped data corresponding to the maximizing w value. As detailed in Section 1, directly solving for $\hat{\theta}(w)$ this many times is computationally prohibitive. Even in the special case of linear regression, under a commonly assumed complexity result, a continuous relaxation of this optimization problem (Equation (1)) cannot be solved in time that is $N^{o(P)}$ [Moitra and Rohatgi, 2023, Theorem 1.3]; a more thorough discussion of this computational complexity result appears in Appendix B.2).

3 Approximations

We next review various approximations to Equation (1) and also discuss new potential approximations. We start with what we call *additive approximations*. In particular, additive approximations (1) approximate the change (to a quantity of interest) due to dropping a single data point and (2) add up the individual changes to approximate the change of dropping a group of data points. Then we discuss *greedy approximations.* Greedy approximations iteratively (1) select the point that results in the biggest change (to the quantity of interest) when dropped and (2) re-run the data analysis without that point [Belsley et al., 1980, Section 2.1]. While many authors have considered approximating dropping a pre-defined subset of data from an analysis, we here focus on dropping the worst-case subset of data as in Equation (1); see Appendix B.3 for further discussion.

3.1 Additive approximations

Approximate Maximum Influence Perturbation (AMIP). Broderick et al. [2020] propose relaxing w to allow continuous values and replacing the w-specific quantity of interest with a first-order Taylor series expansion with respect to w around 1_N. In our case (cf. Appendix C.1), this approximation amounts to replacing Equation (1) with

$$\max_{w \in W_\alpha} \sum\nolimits_{n=1}^{N} (w_n - 1) \frac{\partial \hat{\theta}_p(w)}{\partial w_n}\Big|_{w=1_N} \tag{2}$$

$$\frac{\partial \hat{\theta}(w)}{\partial w_n}\Big|_{w=1_N} = -H(1_N)^{-1} \nabla_\theta f(\theta(1_N), d_n) \tag{3}$$

and $H(w) := \sum_{n=1}^{N} w_n \nabla_\theta^2 f(\theta, d_n)$ is the Hessian of the weighted loss. Equation (2) can be solved by (a) computing the *influence scores* (Equation (3)), (b) finding the largest $\lfloor \alpha N \rfloor$ values, and (c) removing those. Step (a) costs one data analysis and step (b) costs $O(N \log N)$ to sort the scores.

Additive One-Exact. Kuschnig et al. [2021] approximate the change in effect size that results from dropping a group of data points in OLS by the sum of the impacts of dropping individual points. We call this approach *Additive One-Exact* and observe that the idea can be applied more broadly (e.g. for a more general loss f). The broad idea would be to (a) compute the exact effect of dropping a single data point, (b) find the $\lfloor \alpha N \rfloor$ data points that, when dropped individually, yield the largest changes, and (c) remove those. For general losses, Additive One-Exact can cost N times the cost of a single data analysis and need not be exact for $\lfloor \alpha N \rfloor > 1$. In the special case of OLS with an effect-size quantity of interest, Additive One-Exact requires just a single data analysis but still need not be exact for $\lfloor \alpha N \rfloor > 1$.

Additive One-step Newton. Past work has proposed using the one-step Newton (1sN) approximation to estimate how much dropping a pre-defined subset of data changes the loss, for general losses [Beirami et al., 2017, Sekhari et al., 2021, Koh et al., 2019, Ghosh et al., 2020]. When we simultaneously consider (a) OLS linear regression, (b) our particular (effect-size) quantity of interest, and (c) $\lfloor \alpha N \rfloor = 1$, Additive One-Exact is equivalent to the Additive-1sN approximation. So for the experiments below, there is no distinction. Nonetheless, we develop a general form of the Additive-1sN approximation here since it may be useful in models beyond OLS; in particular, in such models, a practitioner may be unwilling to incur the cost of Additive One-Exact. Our more general

[1] See Broderick et al. [2020] for a more general definition, including other quantities of interest.

approximation applies to general differentiable losses, though we continue to focus on a quantity of interest equal to a particular parameter value. (So our approximation does not include a decision based on statistical significance.) To the best of our knowledge, the Additive-1sN approximation has not been previously proposed (beyond the OLS special case).

The 1sN approximation works by optimizing a second-order Taylor expansion to the loss around $w = 1_N$; see Appendix C.2 for more details. In the case where we must search for the worst-case data subset (the Most Influential Set) to drop, we approximate $\hat{\theta}(w)$ with

$$\hat{\theta}^{\text{1sN}}(w) := \hat{\theta}(1_N) + H(w)^{-1} \sum_{n=1}^{N} (w_n - 1) \nabla_\theta f(\theta(1_N), d_n). \tag{4}$$

Analogous to the AMIP, we might think to approximate the objective in Equation (1) by the second term in Equation (4). But now the $H(w)^{-1}$ factor precludes the fast solution we saw for the AMIP. As a result, the general 1sN approximation has not been proposed in the context of the Maximum Influence Perturbation problem. For the problem of worst-case data dropping, we instead propose to approximate the effect of dropping a group of points by the sum of the individual points' approximated contributions; that is, we approximate Equation (1) with $\max_{w \in W_\alpha} \sum_{n=1}^{N} (w_n - 1) \hat{\theta}_p^{\text{1sN}}(1^{(-n)})$, where $1^{(-n)}$ is the all-ones vector except for a zero in the nth entry.

3.2 Greedy approximations

Greedy One-Exact. The outlier detection literature has highlighted the combinatorial cost of finding influential subsets exactly. This literature also describes the *masking problem* that can arise in additive approximations of influence: namely, when one outlier hides the effect of another [Belsley et al., 1980, Atkinson, 1986]. To address these issues, Belsley et al. [1980, Section 2.1] suggest to greedily remove one outlier point at a time in a stepwise procedure. Kuschnig et al. [2021] propose a similar greedy procedure for approximating the Maximum Influence Perturbation; namely, they iteratively: (1) choose one data point that makes the largest exact change when dropped (equivalently, run Additive One-Exact for removing just one data point), (2) re-run the full analysis without that point. In general, Greedy One-Exact requires $\lfloor \alpha N \rfloor$ times the cost of an additive approximation. So in the special case of OLS with an effect-size quantity of interest, Greedy One-Exact costs only $\lfloor \alpha N \rfloor$ data analyses, which is not prohibitive in our examples below. For more general losses though, we expect Greedy One-Exact to cost $\lfloor \alpha N^2 \rfloor$ times the cost of a single data analysis, which will often be prohibitive, especially as the analyses themselves become more complex and costly.

Greedy AMIP. We define Greedy AMIP analogously to Greedy One-Exact, replacing the exact effect of removing a point with the influence score approximation. It requires $\lfloor \alpha N \rfloor$ re-runs of the analysis.

3.3 Lower bound algorithms

A line of recent works provide lower bounds on the Maximum influence Perturbation [Moitra and Rohatgi, 2023, Freund and Hopkins, 2023, Rubinstein and Hopkins, 2024]. The algorithms that provide a lower bound on the number of points that need to be removed to change a conclusion do not also identify a Most Influential Set. In addition to the lower bound algorithms, Moitra and Rohatgi [2023] also introduce algorithms that do identify the Most Influential Set and that rely on mathematical program solvers to do so; Freund and Hopkins [2023] later refine their algorithms. While we do not compare to these methods in the present work, we hope to do so in future. For further discussion on lower bound algorithms, see Appendix B.5.

4 Failure modes

We next demonstrate some failure modes of the approximations above when estimating the sign of an effect with OLS. First, we define two types of failure. We say there is a failure of *Type 1* if there exists a small fraction of data that we can drop to change conclusions, but the method reports such a data subset does not exist. Second, we recognize that, especially in linear regression, users are often willing to run their data analysis one additional time after the approximation. We say there is a failure of *Type 2* if (a) there exists a small fraction of data that we can drop to change conclusions and (b) we remove the points suggested by the method and re-run the analysis, but we do not see an actual change in conclusions upon re-running. We consider various α values, all less than 0.01, in what

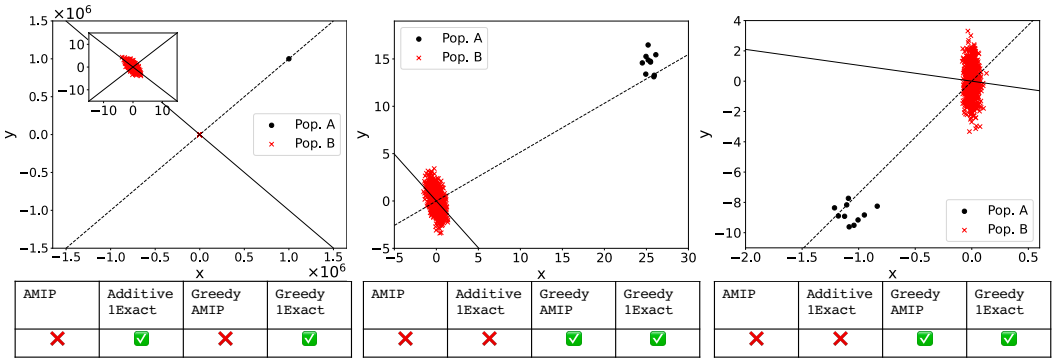

Figure 1: Our examples: one-outlier (left), Simpson's paradox (middle), and poor conditioning (right). A dashed line represents the OLS-estimated slope on the entire dataset while a solid line represents the slope with black dots removed. The tables display the performance of the approximations for the corresponding example: a red X indicates a Type 2 failure, and a green check indicates a success. The left plot includes an inset zoomed in on the bulk of the data.

follows. Note that these failure modes focus on underestimation of sensitivity; if the data analyst is willing to re-run their analysis once, any non-robustness found from that re-run is conclusive.

4.1 One outlier example

By definition, Additive One-Exact is exact for removing a single point. However, the AMIP faces Type 1 and 2 failures. There is no distinction between greedy and additive algorithms when $\lfloor \alpha N \rfloor = 1$. In realistic data settings, we may have a single data point far from the bulk of the data; this outlier may arise due to data-entry errors, machine-measurement errors, or heavy tails in both the covariate and response. To construct the plot in Figure 1 (left), we draw 1,000 red crosses by taking $x_n \sim \mathcal{N}(0, 1)$ i.i.d. and $y_n = -x_n + \epsilon_n$ with $\epsilon_n \sim \mathcal{N}(0, 1)$ i.i.d. Throughout, we use $\mathcal{N}(\mu, \sigma^2)$ to denote the normal distribution with mean μ and variance σ^2. The black dot appears at $x_n = y_n = 10^6$. We fit OLS with an intercept. The OLS-estimated slope on the full dataset is (nearly) 1; after dropping the black point (0.02% of the dataset), the estimate is (nearly) -1, representing a sign change. AMIP chooses a red cross to drop; it predicts that no sign change will be achieved, and dropping the chosen point and re-running also does not achieve a sign change (Table 1). It follows that AMIP suffers both types of failure in this example.

Intuition. To see what goes wrong, consider that in linear regression, the influence score factorizes into two terms: the residual times a leverage-like term [Hampel, 1974]; see Appendix D.2 and Equation (13) for full details. A sufficiently-far outlier will have a very low residual. Meanwhile the leverage-like term goes to zero as the outlier gets farther away. So a sufficiently-far outlier will have a vanishingly low influence score. See Appendix D.2 for details. Although previous work has noted that high-leverage observations can lead to problems for algorithms that approximate the Most Influential Set [Kuschnig et al., 2021], they do not demonstrate this result in a setting with one outlier. This one-outlier setting allows us to mathematically analyze why certain points lead to errors in the influence-function approximation. For instance, our theory suggests that we can achieve similar behavior for outliers in other directions. Specifically, in Appendix D.2, we consider a point with $y_n = cx_n$ for any constant $c > 0$. We show that, for sufficiently large $|x_n|$, the AMIP approximation makes Type 1 errors.

4.2 Simpson's paradox

We next see that, if we have a small, noisy group of outliers instead of a single outlier, both AMIP and Additive One-Exact can suffer Type 1 and 2 failures. The greedy algorithms succeed in this setting. It is common to have (at least) two noisy subpopulations within a single dataset; we consider the case where one subpopulation represents a small fraction of the total. For instance, we might have heterogeneity in the population that the regression model does not account for; such a situation is commonly known as Simpson's paradox. In the particular example in Figure 1 (center), the overall

slope (across all the data) has a different sign than the slope in just the red data or just the black data. To create the illustration in Figure 1 (center), we draw 1,000 red crosses with $x_n \sim \mathcal{N}(0, 0.25)$ iid, $y_n = -x_n + \epsilon_n$, and $\epsilon_n \sim \mathcal{N}(0, 1)$ iid. We draw 10 black dots with $x_n \sim \mathcal{N}(25, 0.25)$, $y_n = -x_n + 40 + \epsilon_n$ iid, and ϵ_n as before. We fit OLS with an intercept. The OLS-estimated slope on the full dataset is 0.52. Dropping the black dots (0.99% of the data) yields a slope of -0.99, a sign change.

When asked to find the worst-case 0.99% (i.e., 10 data points) of the dataset to drop, both AMIP and Additive One-Exact choose some red-cross points and some black-dot points to drop; see Table 2. Both methods predict there will be no sign change (a Type-1 failure). Upon removing the flagged data points and re-running the data analysis, in both cases we find no sign change (a Type-2 failure). At extra computational expense, both greedy methods flag exactly the black-dot data points as the points to drop, so neither suffers a failure.

Intuition. Once we leave the regime of one data point, we see that both AMIP and Additive One-Exact can fail. We see that errors can arise when we approximate the change in dropping a group of data points by the sum of the changes of dropping individual data points. This phenomenon is known more broadly as masking, where one outlier can hide the effect of another [Belsley et al., 1980, Atkinson, 1986]. We analyze this mechanism in more detail in Appendix D.4. To overcome masking problems, previous work has noted the success of using stepwise, or greedy, procedures both in problem of outlier detection [Hadi and Simonoff, 1993, Lawrence, 1995] as well as in the problem of identifying the Maximum Influence Perturbation [Kuschnig et al., 2021]. Although both demonstrate the phenomenon of masking, the failure modes we surface are distinct from the simulation studies of Kuschnig et al. [2021]; our examples demonstrate settings where removing a small fraction of the data can lead to a change in sign of the regression coefficient – a failure of the approximation, as defined in Section 4. The examples in Kuschnig et al. [2021] demonstrate when approximation methods can become inaccurate in the face of masking, but these examples demonstrate neither a Type 1 or Type 2 failure. See Appendix B.4 for more discussion on masking.

4.3 Poor conditioning

We next give an example where AMIP and Additive One-Exact suffer both Type 1 and 2 failures due to the group action of a small subset of data; the greedy algorithms succeed in this setting. Here, we adapt an example presented in Moitra and Rohatgi [2023]; in the example of Moitra and Rohatgi [2023], the data points lie perfectly along two straight lines (see Figure 3); moreover, the small subset of outlier data points lie perfectly along the OLS-estimated slope for the entire dataset. The removal of the black points causes all variation along feature space to be lost and the OLS solution to become ill-defined. We attempt to alter this adversarial setup into one that might arise in natural data settings with no adversary (see Figure 1 (right)). To that end, we find that approximation methods still break down when we add generous amounts of noise to both red-cross points and black-dot points and translate the black-dot points to no longer lie along the OLS-estimated slope. This setting points out a setting in regression, where the covariance matrix is poorly conditioned, and where an OLS solution may become near or completely ill-defined upon dropping a small data subset.

In Figure 1 (right), we generate the red crosses so as to have poor conditioning; since there is much more noise around the (zero) trend than variation in the covariates, there is no clear regression solution. In particular, we generate the 1,000 red crosses with $x_n \sim \mathcal{N}(0, 0.001)$ iid, $y_n = \epsilon_n$, and $\epsilon_n \sim \mathcal{N}(0, 1)$ iid. We draw the 10 black dots as $x_n \sim \mathcal{N}(-1, 0.01)$ iid, $y_n = -x_n - 10 - \epsilon_n$, and ϵ_n as before. We fit OLS with an intercept. When we consider both black dots and red crosses together as a single dataset, there is no poor conditioning. The OLS-estimated slope on the full dataset is around 7.40; dropping the black dots (0.99% of the data) yields a slope of about -1.04, a sign change.

We ask each method to find the worst-case 0.99% (10 data points) of the dataset to drop. Both AMIP and Additive One-Exact choose some red crosses and some black dots; see Table 3. Both methods suffer Type 1 and 2 failures. The greedy methods are more computationally expensive but succeed.

Intuition. In this example, the design matrix is poorly conditioned without the black dots. Although Moitra and Rohatgi [2023] also find that modest, constant-factor instability in the sample covariance can lead to approximations breaking down, their example [Moitra and Rohatgi, 2023, Appendix G.4] relies on an adversarial construction where each element of the sample covariance matrix is specifically chosen in order for the approximation to break down. In contrast, we find through our

example here that constructions need not be adversarial for a failure mode to arise. See Appendix D.4 for a further discussion contrasting the mechanisms at play in Sections 4.2 and 4.3.

4.4 Greedy failures

Moitra and Rohatgi [2023] construct a failure mode of greedy approximations in an adversarial way (see Figure 3). We surface similar adversarial examples where greedy algorithms fail (see Figure 4 and Figure 5) and have yet to identify realistic non-adversarial examples of greedy failures.

5 Discussion

In the present work, we highlight failure modes of approximations to the Maximum Influence Perturbation. For users interested in the Maximum Influence Perturbation for OLS linear regression with a single covariate and a slope quantity of interest, we recommend (1) the use of Greedy One-Exact among the methods described in Section 3 if one is willing to incur the computational expense but importantly (2) that users visualize their data with diagnostic plots (scatter plots, leverage plots, residual plots).

Many opportunities remain for methodological development. For instance, it remains to investigate how alternative models might interact with different data arrangements to affect approximation quality. Geometries of interest include those arising from high dimensional covariates, generalized linear models (and other cases with constrained residuals), constrained parameter spaces (e.g., for variance parameters), mixed-effect models (related to Bayesian hierarchies), and other more-complex models. It also remains to extend additive 1sN to other quantities of interest (e.g., the endpoint of a confidence interval) and more general Z-estimators. These are cases where the AMIP is currently defined, so we might expect to be able to use ideas from the AMIP's development to extend additive 1sN. For models beyond OLS, where there does not exist a closed-form update for the removal of a data point, Greedy One-Exact costs $O(\alpha N^2)$ analyses to run. In these scenarios, it is worth exploring the use of the one-step Newton approximation, which coincides with One-Exact in OLS (see Section 3). Finally, it remains to investigate the speed and accuracy trade-offs in additive and greedy approaches for data analyses more computationally expensive than the low-dimensional linear regression examples considered here; there exist many cases where a practitioner is not willing to re-run their data analysis $\lfloor \alpha N \rfloor$ times. We discuss open problems in greater detail in Appendix E.

Acknowledgments

This work was supported in part by an ONR Early Career Grant, the MIT-IBM Watson AI Lab, the NSF TRIPODS program (award DMS-2022448), and a MachineLearningApplications@CSAIL Seed Award. We are grateful to Tom Rainforth for helpful discussions.

References

Manuela Angelucci and Giacomo De Giorgi. Indirect effects of an aid program: how do cash transfers affect ineligibles' consumption? *American economic review*, 99(1):486–508, 2009.

AC Atkinson. Masking unmasked. *Biometrika*, 73(3):533–541, 1986.

Ahmad Beirami, Meisam Razaviyayn, Shahin Shahrampour, and Vahid Tarokh. On optimal generalizability in parametric learning. *Advances in neural information processing systems*, 30, 2017.

David A. Belsley, Edwin Kuh, and Roy E. Welsch. *Regression diagnostics: Identifying influential data and sources of collinearity*. John Wiley & Sons, 1980.

SM Bendre. Masking and swamping effects on tests for multiple outliers in normal sample. *Communications in statistics-Theory and Methods*, 18(2):697–710, 1989.

Olivier Bousquet and André Elisseeff. Stability and generalization. *The Journal of Machine Learning Research*, 2:499–526, 2002.

Tamara Broderick, Ryan Giordano, and Rachael Meager. An automatic finite-sample robustness metric: When can dropping a little data make a big difference? *arXiv preprint arXiv:2011.14999v1*, 2020.

Anne M. Burton and Travis Roach. Negative externalities of temporary reductions in cognition: Evidence from particulate matter pollution and fatal car crashes. Technical report, University of Texas at Dallas, 2023. Working Paper, https://annemburton.com/pages/working_papers/Burton_Roach_Pollution.pdf.

Andrés F Castro Torres and Aliakbar Akbaritabar. The use of linear models in quantitative research. *Quantitative Science Studies*, pages 1–21, 2024.

R. Dennis Cook. Detection of influential observation in linear regression. *Technometrics*, 42(1): 65–68, 1977.

Elwyn Davies, Peter Deffebach, Leonardo Iacovone, and David Mckenzie. Training microentrepreneurs over zoom: Experimental evidence from Mexico. *Journal of Development Economics*, 167:103244, 2024.

Michael Di and Ke Xu. Covid-19 vaccine and post-pandemic recovery: Evidence from bitcoin cross-asset implied volatility spillover. *Finance Research Letters*, 50:103289, 2022.

Amy Finkelstein, Sarah Taubman, Bill Wright, Mira Bernstein, Jonathan Gruber, Joseph P. Newhouse, Heidi Allen, Katherine Baicker, and the Oregon Health Study Group. The Oregon health insurance experiment: evidence from the first year. *The Quarterly journal of economics*, 127(3):1057–1106, 2012.

Ronald A. Fisher. *Statistical Methods for Research Workers*. Oliver and Boyd, Edinburgh, UK, 1st edition, 1925.

Daniel Freund and Samuel B. Hopkins. Towards practical robustness auditing for linear regression. *arXiv preprint arXiv:2307.16315v1*, 2023.

Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *International Conference on Machine Learning*. JMLR, 2019.

Soumya Ghosh, Will Stephenson, Tin D. Nguyen, Sameer Deshpande, and Tamara Broderick. Approximate cross-validation for structured models. *Advances in neural information processing systems*, 33:8741–8752, 2020.

Ryan Giordano, William Stephenson, Runjing Liu, Michael Jordan, and Tamara Broderick. A Swiss Army infinitesimal jackknife. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1139–1147. PMLR, 2019.

Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens Van Der Maaten. Certified data removal from machine learning models. In *International Conference on Machine Learning*, pages 3832—-3842. JMLR, 2020.

Ali S. Hadi and Jeffrey S. Simonoff. Procedures for the identification of multiple outliers in linear models. *Journal of the American Statistical Association*, 88(424):1264–1272, 1993.

Frank R. Hampel. The influence curve and its role in robust estimation. *Journal of the American Statistical Association*, 69(346), 1974.

Yuzheng Hu, Pingbang Hu, Han Zhao, and Jiaqi W. Ma. Most influential subset selection: Challenges, promises, and beyond. In *Advances in Neural Information Processing Systems*, volume 38, 2024.

Andrew Ilyas, Sung Min Park, Logan Engstrom, Guillaume Leclerc, and Aleksander Madry. Datamodels: Predicting predictions from training data. In *International Conference on Machine Learning*. PMLR, 2022.

Louis A. Jaeckel. The infinitesimal jackknife, memorandum. Technical report, Bell Lab, 1972.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pages 1885–1894. PMLR, 2017.

Pang Wei Koh, Kai-Siang Ang, Hubert H.K. Teo, and Percy Liang. On the accuracy of influence functions for measuring group effects. *Advances in neural information processing systems*, 32, 2019.

Nikolas Kuschnig, Gregor Zens, and Jesús Crespo Cuaresma. Hidden in plain sight: Influential sets in linear models. Technical report, Vienna University of Economics and Business, 2021. CESifo Working Paper, http://dx.doi.org/10.2139/ssrn.3819102.

AJ Lawrence. Deletion influence and masking in regression. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 57(1):181–189, 1995.

Allen Liu and Ankur Moitra. Learning GMMs with nearly optimal robustness guarantees. In *Conference on Learning Theory*, pages 2815–2895. PMLR, 2022.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *stat*, 1050(9), 2017.

Luis R. Martinez. How much should we trust the dictator's GDP growth estimates? *Journal of Political Economy*, 130(10):2731–2769, 2022.

Ankur Moitra and Dhruv Rohatgi. Provably auditing ordinary least squares in low dimensions. *The 11th International Conference on Learning Representations (ICLR 2023)*, 2023.

Tin D. Nguyen, Ryan James Giordano, Rachael Meager, and Tamara Broderick. Using gradients to check sensitivity of MCMC-based analyses to removing data. In *ICML 2024 Workshop on Differentiable Almost Everything: Differentiable Relaxations, Algorithms, Operators, and Simulators*, 2024. URL https://openreview.net/forum?id=bCPhVcq9Mj.

Sung Min Park, Kristian Georgiev, Andrew Ilyas, Guillaume Leclerc, and Aleksander Madry. Trak: Attributing model behavior at scale. In *International Conference on Machine Learning*, pages 27074–27113. PMLR, 2023.

Daryl Pregibon. Logistic Regression Diagnostics. *The Annals of Statistics*, 9(4), 1981.

Kamiar Rad and Arian Maleki. A scalable estimate of the out-of-sample prediction error via approximate leave-one-out cross-validation. *Journal of the Royal Statistical Society, Series B*, 82 (4):965–996, 2020.

Ittai Rubinstein and Samuel Hopkins. Robustness auditing for linear regression: To singularity and beyond. *arXiv preprint arXiv:2410.07916*, 2024.

Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. Remember what you want to forget: Algorithms for machine unlearning. *Advances in Neural Information Processing Systems*, 34:18075–18086, 2021.

William Stephenson and Tamara Broderick. Approximate cross-validation in high dimensions with guarantees. In *International conference on artificial intelligence and statistics*, pages 2424–2434. PMLR, 2020.

Vinith Suriyakumar and Ashia C. Wilson. Algorithms that approximate data removal: New results and limitations. *Advances in Neural Information Processing Systems*, 35:18892–18903, 2022.

A Code

Code is available at https://github.com/JennyHuang19/gradientBasedDataDroppingFailureModes, including all scripts for reproducing the results in this paper.

B Related works

B.1 Related notions of robustness

In this section, we discuss related notions of robustness and explain why robustness to worst-case data dropping provides a new and useful check on generalizability. Many tools in statistics, such as p-values and confidence intervals, are meant to measure the generalizability of sample-based conclusions [Fisher, 1925]. Similarly, works on algorithmic stability in the learning theory literature quantify an algorithm's generalization error [Bousquet and Elisseeff, 2002]. However, these tools rely on an assumption that the data are drawn independently and identically (IID) from the underlying target population. In most real world settings, we cannot assume the population that the samples are drawn from is identical to the target population. For instance, we might look to apply the conclusions from a specific sample to a slightly different future population. Departing from the IID regime, we can no longer rely on the theory behind classical tools alone to tell us something definitive about the generalizability of sample-based conclusions. Prior works have also studied the robustness of estimators to gross outliers and adversarially corrupted samples, arbitrary corruptions of a data-point or small collection of data points [Hampel, 1974, Madry et al., 2017, Liu and Moitra, 2022]. The influence function has played a central role in the study of gross outlier sensitivity since the pioneering work of Hampel [1974]. Specific to linear regression, Cook [1977] introduced Cook's Distance, for detecting outliers and gross errors. However, conclusions may still fail to generalize in the absence of gross outliers [Broderick et al., 2020]. As these related notions of robustness alone do not provide a comprehensive check, a data analyst might still worry about generalizability if dropping a very small fraction of the sample can lead to drastically different conclusions.

B.2 Computational difficulties of the Maximum Influence Perturbation problem

An exact computation of the Maximum Influence Perturbation is computationally intractable. A brute force approach involves enumerating over every possible data subset, which amounts to rerunning $O(2^N)$ data analyses, where N is the number of points in the dataset. Inspired by the Maximum Influence Perturbation problem, Moitra and Rohatgi [2023] prove computational complexity results for finding lower bounds on the maximum influence perturbation within the specific setting of detecting a sign change of a regression coefficient. In particular, $Stability(X, y)$ is the minimum number of points that can be dropped to change the sign. Moitra and Rohatgi [2023] improve on the results of the brute force computation, showing that the exponential dependence on N is not necessary while an exponential dependence on P is (P being the dimension of feature space) [Moitra and Rohatgi, 2023, Theorem 1.1 and 1.2]. Under the Exponential Time Hypothesis, Moitra and Rohatgi [2023] proved that, for a given integer $k > 0$, there is no $N^{o(P)}$ time algorithm that can determine whether $Stability(X, y) \leq k$ [Moitra and Rohatgi, 2023, Theorem 1.2][2]. Note, as computing $Stability(X, y)$ is strictly harder than computing $Stability(X, y) \leq k$ for a given k, the Most Influential Set problem (and corresponding Maximum Influence Perturbation problem) must require at least $N^{\Omega(P)}$ computations. As the computation of the Maximum Influence Perturbation is expensive, even in cases of OLS linear regression, this prompts the need for approximations.

B.3 Approximations to (non-worst-case) data-dropping

In this section, we discuss tangential works that use approximations to data dropping in settings where the subset to drop is known. While we are concerned with developing algorithms to overcome the combinatorial problem of searching for some worst-case subset to induce the largest change in a quantity of interest, the works discussed in this section do not provide a fast way to search for the worst-case data subset to drop.

The idea of using approximations to data dropping goes as far back as Jaeckel [1972] and Hampel [1974], who introduced the influence function in the context of robust statistics. Cook [1977] then introduced influence measures, such as Cook's Distance, in the context of detecting outliers and gross

[2] Moitra and Rohatgi [2023] worked with a fractionally relaxed version of this problem, where the weight of a data point can take on non-integer values. This result precludes the existence of a faster solution in the integer version, as one could use the integer version to solve the weighted version (up to an approximation) by making several copies of the dataset.

errors. Pregibon [1981] introduced the one-step Newton approximation in the context of logistic regression diagnostics.

Recently, approximations for data dropping have been used as fast alternatives to model retraining, which can be expensive when using complex models on large datasets. Several works have developed gradient-based approximations to cross validation [Beirami et al., 2017, Rad and Maleki, 2020, Giordano et al., 2019, Stephenson and Broderick, 2020, Ghosh et al., 2020]. Works in the data privacy space have used approximations for deleting user data from models [Guo et al., 2020, Sekhari et al., 2021, Suriyakumar and Wilson, 2022]. Within the model interpretability and data attribution space, methods such as Shapley value estimators [Ghorbani and Zou, 2019] and datamodels [Ilyas et al., 2022] require retraining the model a large number of times on different subsets of the data in order to quantify the impact of particular training points on the model output. In response, several works have proposed using approximations to retraining based on the influence function [Koh and Liang, 2017, Koh et al., 2019, Park et al., 2023]. These gradient-based based approximations achieve great gains in computation while maintaining comparable accuracy to methods that rely on model retraining, as shown in Park et al. [2023, Figure 1]. While these works [Koh and Liang, 2017, Koh et al., 2019, Park et al., 2023] investigate the performance of gradient-based approximations to data-dropping for the task of dropping a pre-defined subset, we investigate the performance of gradient-based approximations to data-dropping for the task of dropping a worst-case data subset, noting that the performance of an approximation may be quite different in the worst-case compared to an average-case subset.

B.4 Masking

We identify cases where masking, when one outlier hides the effect of another outlier, can interfere with finding the Most Influential Set. Masking itself has been a widely studied phenomenon in the context of multi-outlier detection. Hampel [1974], a foundational text in robust statistics, notes that masking can make the identification of multiple outliers cumbersome and erroneous [Hampel, 1974, Section 1.4]. Belsley et al. [1980, Section 2.1] also discussed the masking phenomenon and proposed a stepwise procedure for identifying groups of influential outliers. Bendre [1989] noted that masking can change the results for some common multiple outlier tests. Lawrence [1995] noted that masking can create errors in common diagnostic quantities, such as Cook's Distance. In the context of outlier detection, Atkinson [1986] proposed a solution that is able to mitigate the effects of masking: it is based on a two-step procedure that first fits subsamples of the data using least median of squares regression, which identifies potential groups of outliers, then uses single-point influence measures to confirm whether the points identified are indeed outliers. To address masking effects in this work, we consider a stepwise approach (Greedy AMIP and One-Exact) similar to the one taken in Belsley et al. [1980, Section 2.1] to overcome a combinatorial problem of searching for a Most Influential Set. Most recently, Kuschnig et al. [2021] compare greedy and additive approximations for the Most Influential Set problem; we build on their work by (1) identifying instances where these methods result in conclusive failures through the definitions in Section 4, and (2) provide mathematical insight into these failure modes, as we present in Appendix D.4.

B.5 Lower bound algorithms

Following Broderick et al. [2020], a line of works [Moitra and Rohatgi, 2023, Freund and Hopkins, 2023, Rubinstein and Hopkins, 2024] provides lower bounds on the Maximum Influence Perturbation in the specific context of detecting a sign change in a particular component of a regression coefficient vector in OLS linear regression. However, these lower bound algorithms (1) do not identify a Most Influential Set and (2) are not easily adaptable to cases outside of detecting sign changes in OLS linear regression, so we do not compare to these methods in this work.

Moitra and Rohatgi [2023] introduce an additional upper bound algorithm that does identify the Most Influential Set by solving a mathematical program and Freund and Hopkins [2023] make refinements to the algorithm. Specifically, Moitra and Rohatgi [2023] consider a version of the problem in which the data weights do not have to be binary but may instead be fractional, while Freund and Hopkins [2023] enforces the weights to be integral. In contrast to the additive and greedy approximations examined in the current work, these mathematical programming algorithms provide PAC-style guarantees for approximating the Maximum Influence Perturbation [Moitra and Rohatgi, 2023]. We hope to compare to these methods in future.

Within the specific context of OLS, the line of algorithms providing lower bounds may inspire future methodological development for the Most Influential Set problem. For example, Freund and Hopkins [2023] introduce a spectral algorithm that takes a more global approach than those taken by either the additive or greedy approximations. As another example, Rubinstein and Hopkins [2024] introduce a lower bound algorithm based on analyzing the error term between the AMIP approximation and the true effect of rerunning an analysis without the dropped subset. They then provide new ways to upper bound the error term expression; this may offer mathematical insight into ways in which the additive and greedy algorithms break down. Despite these connections, there is currently no direct way to use these algorithms to identify a Most Influential Set, so it is not the focus of this paper.

B.6 Failure modes

Past works have pointed out cases where worst-case data dropping approximations may perform poorly, but these works do not define notions of failure within the context of generalization of sample-based conclusions. We define these notions of failure concretely in Section 4 and then surface examples of failures with respect to these specific definitions. Whereas previous works have pointed out failures in adversarial examples and in settings of dropping larger data fractions $> 1\%$ [Moitra and Rohatgi, 2023, Freund and Hopkins, 2023], we are concerned with failures that can arise in natural data settings without an adversary and (for purposes of generalization) where dropping a surprisingly small fraction of data leads to changes in conclusions. Other past works have surfaced failure modes in real-world settings without further investigation into the properties linking the mathematical error in these approximations to the data [Broderick et al., 2020, Nguyen et al., 2024, Moitra and Rohatgi, 2023, Freund and Hopkins, 2023, Kuschnig et al., 2021], which we do in Appendix D.4. Specifically, Broderick et al. [2020, Section 4.3] and Nguyen et al. [2024, Section 6.2] point to settings where the approximation performs poorly on a study on microcredit [Angelucci and De Giorgi, 2009]. Broderick et al. [2020, Section 4.3] points out a failure of the AMIP in a setting where the quantity of interest is a hypervariance parameter in a hierarchical model. Here, AMIP approximates a positive effect while the actual refit gives a negative effect. This failure modes has another layer of distinction from our problem, as it points to a failure that may arise due to a constrained parameter space. Nguyen et al. [2024, Section 6.2] point out a setting where the approximations perform poorly for a component of a hierarchical model fitted with MCMC; in particular, they identify a setting where the confidence interval for AMIP undercovers. Finally, for the same microcredit study, Kuschnig et al. [2021], Moitra and Rohatgi [2023], Freund and Hopkins [2023] compare the performance of different approximation algorithms but do not further investigate these settings or generalize properties in the data that may be linked to these failure modes.

C Approximation supplementals

C.1 AMIP supplementals

Broderick et al. [2020] consider a linear approximation to dropping data that can be used in any setting where the loss function $f(d_n; \theta)$ is twice continuously differentiable in θ. They define a quantity-of-interest, $\phi(\theta, w)$, to be a scalar related to the conclusion of a data analysis, which one is concerned about observing a change in upon dropping a very small fraction of data. Common quantities of interest in a data analysis include the sign or significance of a regression coefficient.

Specifically, they linearize the quantity-of-interest as a function of the data-weights

$$\phi^{\text{lin}}(w) = \phi(1_N) + \sum_{n=1}^{N} (w_n - 1) \frac{\partial \phi(w)}{\partial w_n}\Big|_{w=1_N}. \tag{5}$$

The derivative $\frac{\partial \phi(w)}{\partial w_n}\big|_{w=1_N}$ is known as the *influence score* of data point n for ϕ at 1_N.

Although the AMIP methodology has been developed and used for general quantities of interest, $\phi(\hat{\theta}(w), w)$, we focus on the change in sign of a particular regression coefficient, p; thus, $\phi(\hat{\theta}(w), w) = -\hat{\theta}_p(w)$. Under the general setting where $\hat{\theta}(1_N)$ is the solution to the equation $(\sum_{n=1}^{N} \nabla_\theta f(\hat{\theta}(1_N), d_n)) = 0_p$ (which is the case in our setup as $\hat{\theta}(1_N)$ is a minimizer of a loss function) the implicit function theorem allows us to transform a derivative in w space into a derivative

in θ space

$$\frac{\partial \hat{\theta}(\vec{w})}{\partial w_n}\Big|_{\vec{w}=1_N} = -H(1_N)^{-1}\nabla_\theta f(\hat{\theta}(1_N), d_n) \tag{6}$$

where for $\vec{v} \in \mathbb{R}^N$, $H(\vec{v}) := \sum_{n=1}^N v_n \nabla_\theta^2 f(\hat{\theta}(1_N), d_n)$ is the Hessian of the weighted loss. See Broderick et al. [2020] for a detailed derivation of Equation (6).

Let e_p be the pth standard basis vector. Then the linear approximation in the setting where the quantity of interest is the sign of the pth regression coefficient becomes

$$\hat{\theta}_p^{\text{lin}}(w) = \hat{\theta}_p(1_N) + e_p^\top H(1_N)^{-1} \sum_{n=1}^N (w_n - 1)\nabla_\theta f(\hat{\theta}(1_N), d_n). \tag{7}$$

C.2 One-step Newton supplementals

Let w be a vector such that $w_n = 0$ for the dropped data points and $w_n = 1$ for the remaining data points. The one-step Newton approximation allows us to approximate $\hat{\theta}(w) = \arg\min \sum_{n=1}^N w_n f(\theta, d_n)$ by approximating the weighted loss function with a second-order Taylor series expansion (in θ) centered at the estimate for the full data, $\hat{\theta}(1_N)$.

$$\begin{aligned}\sum_{n=1}^N w_n f(\theta, d_n) \approx f(\hat{\theta}(1_N), d_n) &+ \sum_{n=1}^N w_n \nabla f(\hat{\theta}(1_N), d_n)(\theta - \hat{\theta}(1_N)) \\ &+ \frac{1}{2}(\theta - \hat{\theta}(1_N))^\top \sum_{n=1}^N w_n \nabla^2 f(\hat{\theta}(1_N), d_n)(\theta - \hat{\theta}(1_N))\end{aligned} \tag{8}$$

In order to solve for $\arg\min \sum_{n=1}^N w_n f(\theta, d_n)$, we can minimize the quadratic approximation to get

$$\hat{\theta}^{\text{1sN}}(w) = \hat{\theta}(1_N) + (\sum_{n=1}^N w_n \nabla^2 f(\hat{\theta}(1_N), d_n))^{-1} \sum_{n=1}^N w_n \nabla f(\hat{\theta}(1_N), d_n). \tag{9}$$

In the recent machine learning literature, this 1sN approximation (Equation (9)) has been proposed to estimate the effect of dropping known subsets of data [Beirami et al., 2017, Sekhari et al., 2021, Koh et al., 2019, Ghosh et al., 2020] in the context of general twice-differentiable losses.

The 1sN approximation has not been proposed in the context of the Maximum Influence Perturbation problem because (unlike influence scores) the approximation is not additive. This non-additivity precludes the fast solution of approximating the Most Influential Set by ranking and taking a sum of the top individual scores [Broderick et al., 2020]. As a solution, we adapt ideas from AMIP to consider an approximation that uses a sum of 1sN scores for leaving out individual data points (Equation (10)). We call this approximation the *Additive 1sN (Add-1sN)*. See Section 3 for more details.

$$\hat{\theta}^{\text{Add-1sN}}(w) = \hat{\theta}(1_N) + \sum_{i \in S} \left(\Big(\sum_{\substack{n=1,\\ n \neq i}}^N \nabla^2 f(\hat{\theta}(1_N), d_n) \Big)^{-1} \nabla f(\hat{\theta}(1_N), d_i) \right). \tag{10}$$

C.2.1 The 1sN approximation gives the exact reweighted OLS estimate

In the setting of simple linear regression, the one-step Newton approximation gives the exact solution to the reweighted ordinary least squares estimate of a regression coefficient [Pregibon, 1981, Equation 3]. Let $\mathbf{X} \in \mathbb{R}^{N \times P}$ denote the design matrix and $\vec{y} \in \mathbb{R}^N$ denote the response vector. Let S denote the dropped set (i.e. the observations indexed by S in the design matrix and response vector) and $\setminus S$ denote its complement. Let $\hat{\beta}^{\text{1sN}}(w)$ denote the one-step Newton approximation of $\hat{\beta}(w)$ given in Equation (4).

$$
\begin{aligned}
\hat{\beta}^{\text{1sN}}(w) &= (\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top \vec{y} + (\mathbf{X}_{\setminus S}^\top \mathbf{X}_{\setminus S})^{-1}(\mathbf{X}_S^\top \vec{y}_S - \mathbf{X}_S^\top \mathbf{X}_S (\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top \vec{y}) \\
&= (\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top \vec{y} - (\mathbf{X}_{\setminus S}^\top \mathbf{X}_{\setminus S})^{-1}(\mathbf{X}_{\setminus S}^\top \vec{y}_{\setminus S} - \mathbf{X}_{\setminus S}^\top \mathbf{X}_{\setminus S} (\mathbf{X}^\top \mathbf{X})^{-1}\mathbf{X}^\top \vec{y}) \\
&= (\mathbf{X}_{\setminus S}^\top \mathbf{X}_{\setminus S})^{-1}\mathbf{X}_{\setminus S}^\top \vec{y}_{\setminus S}
\end{aligned} \tag{11}
$$

such that

$$
\hat{\beta}(w) - \hat{\beta}^{\text{1sN}}(w) = (\mathbf{X}_{\setminus S}^\top \mathbf{X}_{\setminus S})^{-1}\mathbf{X}_{\setminus S}^\top \vec{y}_{\setminus S} - (\mathbf{X}_{\setminus S}^\top \mathbf{X}_{\setminus S})^{-1}\mathbf{X}_{\setminus S}^\top \vec{y}_{\setminus S} = 0. \tag{12}
$$

D Failure modes supplementals

D.1 Tables demonstrating results of approximation methods

Table 1 shows that, in the One Outlier example, AMIP fails in the type 1 sense. While there exists one point (in black) that can change the sign of the regression coefficient, the method reports that no subsets of size 1 exist that can change the sign. AMIP also fails in the type 2 sense because, upon removal of the point suggested by AMIP (which is a red point) and refitting the model, we still do not see a change in sign. Greedy AMIP fails in the type 2 sense because the sign does not change upon refitting after we remove the point identified by the algorithm. In this example, Additive 1sN and Greedy 1sN succeed.

Table 2 shows that in the Simpson's Paradox example, AMIP and Additive 1sN fail in both the type 1 and type 2 sense. While there exists a group of ten points ($\alpha < 0.01$) (specifically, the group of points in black) such that, upon removal, the sign of the regression coefficient changes from positive to negative, both AMIP and Additive 1sN report that no such subset of this size or smaller exists. The sign also does not change upon refitting, after removing the points identified by the algorithms. In this example, the greedy versions of both approximations succeed.

Similar to the Simpson's Paradox example, Table 3 shows that in the Poor Conditioning example, AMIP and Additive 1sN fail in both the type 1 and type 2 sense, while the greedy versions of both approximations succeed.

Table 1: Performance of methods under the One Outlier example, where an outlier is placed at (X, Y) = (1e6, 1e6). We know that there exists a subset (one data point!) such that, upon removal, the sign of the regression coefficient changes from positive (1.000) to negative (-1.000). Hence, $\alpha = \frac{1}{N}$ is sufficient to lead to a failure mode. The "Predicted Estimate" column shows the estimate predicted by the approximation algorithm, and the "Refit Estimate" column shows the result of refitting the model after removing the approximate Most Influential Subset specified by the algorithm. The "Points Dropped" column shows the number of red (R) and black (B) points that the algorithm drops. The values highlighted in green indicate that the algorithm succeeded. Non-highlighted values under "Predicted Estimate" indicate a failure of type (i), while non-highlighted values under "Refit Estimate" indicate a failure of type (ii).

Method	Predicted Estimate	Refit Estimate	Points Dropped
Removing Population A	—	-1.000	(R: 0, B: 1)
AMIP	0.999	0.999	(R: 1, B: 0)
Additive 1sN	-1.000	-1.000	(R: 0, B: 1)
Greedy AMIP	—	0.999	(R: 1, B: 0)
Greedy 1sN	—	-1.000	(R: 0, B: 1)

Table 2: Performance of methods under the Simpson's Paradox example. We know that there exists a subset (namely, the 10 points in Population A ($\alpha < 0.01$) in Figure 1) such that, upon removal, the sign of the regression coefficient changes from positive (0.586) to negative (-0.990). The "Predicted Estimate" column shows the estimate predicted by the approximation algorithm, and the "Refit Estimate" column shows the result of refitting the model after removing the approximate Most Influential Subset specified by the algorithm. The "Points Dropped" column shows the number of red (R) and black (B) points that the algorithm drops. The values highlighted in green indicate that the algorithm succeeded. Non-highlighted values under "Predicted Estimate" indicate a failure of type (i), while non-highlighted values under "Refit Estimate" indicate a failure of type (ii).

Method	Predicted Estimate	Refit Estimate	Points Dropped
Removing Population A	—	-0.990	(R: 0, B: 10)
AMIP	0.462	0.279	(R: 2, B: 8)
Additive 1sN	0.456	0.279	(R: 2, B: 8)
Greedy AMIP	—	-0.990	(R: 0, B: 10)
Greedy 1sN	—	-0.990	(R: 0, B: 10)

Table 3: Performance of methods under the Poor Conditioning example. We know that there exists a subset (namely, the 10 points in Population A ($\alpha < 0.01$) in Figure 1) such that, upon removal, the sign of the regression coefficient changes from positive (8.452) to negative (-1.049). The "Predicted Estimate" column shows the estimate predicted by the approximation algorithm, and the "Refit Estimate" column shows the result of refitting the model after removing the approximate Most Influential Subset specified by the algorithm. The "Points Dropped" column shows the number of red (R) and black (B) points that the algorithm drops. The values highlighted in green indicate that the algorithm succeeded. Non-highlighted values under "Predicted Estimate" indicate a failure of type (i), while non-highlighted values under "Refit Estimate" indicate a failure of type (ii).

Method	Predicted Estimate	Refit Estimate	Indices Dropped
Removing Population A	—	-1.049	(R: 0, B: 10)
AMIP	6.724	5.376	(R:3, B:7)
Additive 1sN	6.667	5.376	(R: 3, B: 7)
Greedy AMIP	—	-1.049	(R: 0, B: 10)
Greedy 1sN	—	-1.049	(R: 0, B: 10)

D.2 One-outlier example

We saw that both AMIP and Greedy AMIP break down in the one-outlier setting. This is because the influence score of the outlier vanishes as the point approaches infinity in the x and y directions. In Proposition D.1, we will examine the mathematics behind this phenomenon.

Proposition D.1. *Let (x_i, y_i) denote the outlier point, where $x_i = \lambda v \in \mathbb{R}^P$, $y_i = \lambda$. Let $\|v\| = 1$, $e \in \mathbb{R}^P$, and $\mathbf{X} \in \mathbb{R}^{N \times P}$, and assume that the OLS estimator remains well-defined upon the removal of the outlier (the covariance matrix is non-degenerate). As $\lambda \to \infty$, the influence with respect to $e^\top \beta$ goes to zero. Note, this result holds more generally for $y_i = c\lambda$, for any constant $c > 0$.*

Proof. The formula for the influence function in OLS linear regression is

$$IF((x_i, y_i); \hat{\beta}) = \underbrace{(\mathbf{X}^\top \mathbf{X})^{-1} x_i}_{\text{leverage-like term}} \underbrace{(y_i - \hat{\beta}^\top x_i)}_{\text{residual term}} \tag{13}$$

Let $A = (\mathbf{X}_{-i}^\top \mathbf{X}_{-i})$. The leverage-like term, $(\mathbf{X}^\top \mathbf{X})^{-1} x_i$, approaches $0 \in \mathbb{R}^P$ as $\lambda \to \infty$.

$$
\begin{aligned}
(\mathbf{X}^\top \mathbf{X})^{-1} x_i &= (A + x_i x_i^\top)^{-1} x_i \\
&= (A^{-1} - \frac{A^{-1} x_i x_i^\top A^{-1}}{1 + x_i^\top A^{-1} x_i}) x_i && \text{(Sherman-Morrison formula)} \\
&= \frac{A^{-1} x_i}{1 + x_i^\top A^{-1} x_i} && \text{(algebraic simplification)} \\
&= \frac{\lambda A^{-1} v}{1 + \lambda^2 v^\top A^{-1} v} && \text{(substituting } x_i = \lambda v \in \mathbb{R}^P, y_i = \lambda)
\end{aligned}
$$

Hence, the leverage-like term vanishes at rate $O(\frac{1}{\lambda})$.

The residual term approaches 0 as $\lambda \to \infty$.

$$
\begin{aligned}
\hat{\beta}^\top x_i &= y^\top \mathbf{X} (\mathbf{X}^\top \mathbf{X})^{-1} x_i && \text{(OLS solution)} \\
&= y^\top \mathbf{X} A^{-1} x_i - \frac{y^\top \mathbf{X} A^{-1} x_i x_i^\top A^{-1} x_i}{1 + x_i^\top A^{-1} x_i} && \text{(Sherman-Morrison formula)} \\
&= y^\top \mathbf{X} A^{-1} x_i (1 - \frac{x_i^\top A^{-1} x_i}{1 + x_i^\top A^{-1} x_i}) && \text{(algebraic simplification)} \\
&= \frac{y^\top \mathbf{X} A^{-1} x_i}{1 + x_i^\top A^{-1} x_i} \\
&= \frac{(y_i x_i^\top + \sum_{j \neq i} y_j x_j^\top) A^{-1} x_i}{1 + x_i^\top A^{-1} x_i} \\
&= \frac{(\lambda^2 v^\top + \sum_{j \neq i} y_j x_j^\top) A^{-1} \lambda v}{1 + \lambda^2 v^\top A^{-1} v} && \text{(substituting } x_i = \lambda v \in \mathbb{R}^P, y_i = \lambda) \\
&= \frac{\lambda^3 v^\top A^{-1} v + \lambda \sum_{j \neq i} y_j x_j^\top A^{-1} v}{1 + \lambda^2 v^\top A^{-1} v}
\end{aligned} \tag{14}
$$

$$
y_i - \hat{\beta}^\top x_i = \frac{\lambda - \lambda \sum_{j \neq i} y_j x_j^\top A^{-1} v}{1 + \lambda^2 v^\top A^{-1} v} \tag{15}
$$

Applying L'Hopital's rule, we get that $\lim_{h \to \infty} \frac{f(h)}{g(h)} = \lim_{h \to \infty} \frac{f'(h)}{g'(h)}$.

$$
\frac{\partial}{\partial \lambda} (\frac{\lambda - \lambda \sum_{j \neq i} y_j x_j^\top A^{-1} v}{1 + \lambda^2 v^\top A^{-1} v}) = \frac{1 - \sum_{j \neq i} y_j x_j^\top A^{-1} v}{\lambda v^\top A^{-1} v} \tag{16}
$$

Hence, the residual term vanishes at rate $O(\frac{1}{\lambda})$.

Since both $(\mathbf{X}^\top \mathbf{X})^{-1} x_i$ and $(y_i - \hat{\beta}^\top x_i)$ approach zero as $\lambda \to \infty$, the influence score also approaches zero. □

In contrast, the 1sN score for one data point is equivalent to the leave-one-out deletion formula for linear regression, $\beta - \beta(-i) = \frac{(\mathbf{X}^\top \mathbf{X})^{-1} x_i^\top (y_i - \beta^\top x_i)}{(1 - h_i)}$ [Belsley et al., 1980]. This fact can be verified in the following Proposition D.2.

Proposition D.2. *For all values of λ, the One-step Newton approximation is equivalent to $e^\top(\beta - \beta(-i))$.*

Proof. The One-step Newton approximation differs from the Influence Function approximation by an additional leverage-based term (Equation (17)).

$$
\begin{aligned}
&\frac{1}{1 - x_i^\top A^{-1} x_i} \\
&\qquad = \frac{1}{1 - \frac{1}{(\lambda^2 v^\top A^{-1} v)^{-1} + 1}} \qquad \text{(substituting } x_i = \lambda v \in \mathbb{R}^P\text{)} \\
&\qquad = \lambda^2 v^\top A^{-1} v + 1. \qquad \text{(algebraic simplification)}
\end{aligned} \tag{17}
$$

Taking the product of each term in the one-step Newton score (where $A = (\mathbf{X}_{-i}^\top \mathbf{X}_{-i})$), we get $(\beta - \beta_{-i})$.

$$
(1 + \lambda^2 v^\top A^{-1} v)(\frac{\lambda A^{-1} v}{1 + \lambda^2 v^\top A^{-1} v})(\frac{\lambda - \lambda \sum_{j \neq i} y_j x_j^\top A^{-1} v}{1 + \lambda^2 v^\top A^{-1} v})
$$

$$
\begin{aligned}
&= \lambda A^{-1} v (\frac{\lambda - \lambda \sum_{j \neq i} y_j x_j^\top A^{-1} v}{1 + \lambda^2 v^\top A^{-1} v}) && \text{(algebraic simplification)} \\
&= \frac{\lambda^2 A^{-1} v - \lambda^2 A^{-1} v \sum_{j \neq i} y_j x_j^\top A^{-1} v}{1 + \lambda^2 v^\top A^{-1} v} \\
&= (\frac{y_i A^{-1} x_i - A^{-1} x_i (\sum_{j \neq i} y_j x_j^\top A^{-1} x_i)}{1 + x_i^\top A^{-1} x_i}) && \text{(substituting } x_i = \lambda v \in \mathbb{R}^P, y_i = \lambda\text{)} \\
&= (\frac{y_i A^{-1} x_i + y_i A^{-1} x_i x_i^\top A^{-1} x_i - A^{-1} x_i x_i^\top A^{-1} (\sum y_j x_j + y_j x_j)}{1 + x_i^\top A^{-1} x_i}) && \text{(adding zero)} \\
&= (A^{-1} y_i x_i - \frac{A^{-1} x_i x_i^\top A^{-1}}{1 + x_i^\top A^{-1} x_i} (\sum_{i \neq j} y_j x_j + y_j x_j)) && \text{(algebraic simplification)} \\
&= ((A^{-1} - \frac{A^{-1} x_i x_i^\top A^{-1}}{1 + x_i^\top A^{-1} x_i})(\sum_{j \neq i} y_j x_j + y_i x_i) - A^{-1} \sum_{j \neq i} y_j x_j) \\
&= ((\mathbf{X}^\top \mathbf{X})^{-1} X^\top y - (\mathbf{X}_{-i}^\top \mathbf{X}_{-i})^{-1} \mathbf{X}_{-i}^\top y_{-i})) && \text{(Sherman-Morrison formula)} \\
&= (\beta - \beta_{-i}) && \text{(OLS Solution)}
\end{aligned} \tag{18}
$$

□

The 1sN score differs from the influence score by a factor of $\frac{1}{1-h_i}$, where h_i is the leverage score for data point i [Belsley et al., 1980, Section 2.1]. Hence, for all values of λ, the 1sN approximation is equivalent to $e^\top(\beta - \beta(-i))$ (Proposition D.2).

Table 4: This table shows the influence and 1Exact scores for the black-dot point at various values of (x_i, y_i) for the data generating process described in Section 4.1 (see plot for the setting where $(x_i, y_i) = (10^6, 10^6)$ in Figure 1 (left)). In order to obtain the influence with respect to β_1, we let $e_1 = (0, 1)$, the standard basis vector corresponding to the x term in our linear regression setup. The Influence Score, $e_1^\top (\mathbf{X}^\top \mathbf{X})^{-1} x_i (y_i - \hat{\beta} x_i)$, is a product between the quantity in column 3 (which we call the leverage-like term, see Equation (13)) and column 4 (the residuals). The 1Exact Score is the change in effect size that results from dropping the data point at (x_i, y_i) and refitting OLS. The influence scores highlighted in yellow are those that are smaller than the influence score of a red-cross point, leading AMIP and Greedy AMIP to misidentify the Most Influential Set of size 1, resulting in a Type 2 failure.

BLACK DOT:					
x_i	y_i	$e_1^\top (\mathbf{X}^\top \mathbf{X})^{-1} x_i$	$(y_i - \hat{\beta} x_i)$	INFLUENCE SCORE	1EXACT SCORE
1E1	1E1	9.36E-3	18.390	1.72E-1	1.90E-1
1E2	1E2	9.11E-3	17.981	1.64E-1	1.85
1E4	1E4	1.00E-5	1.97E-1	1.97E-5	2.03
1E6	1E6	1.00E-6	1.97E-3	1.97E-9	2.03
1E8	1E8	1.00E-8	2.00E-5	1.97E-13	2.18
1E10	1E10	1.00E-10	1.00E-5	9.54E-16	2.03

Table 5: This table shows the influence and 1Exact scores for the red-cross point with the largest influence score when the black-dot point (see Figure 1 (left)) is placed at the (x_i, y_i) position shown in the corresponding row of Table 4. In order to obtain the influence with respect to β_1, we let $e_1 = (0, 1)$, the standard basis vector corresponding to the x term in our linear regression setup. The Influence Score, $e_1^\top (\mathbf{X}^\top \mathbf{X})^{-1} x_j (y_j - \hat{\beta} x_j)$, is a product between the quantity in column 3 (which we call the leverage-like term, see Equation (13)) and column 4 (the residuals). The 1Exact Score is the change in effect size that results from dropping the data point at (x_j, y_j) and refitting OLS. The influence scores highlighted in yellow are those that are larger than the influence score of the black-dot point, leading AMIP and Greedy AMIP to misidentify the Most Influential Set of size 1, resulting in a Type 2 failure.

RED CROSS:					
x_j	y_j	$e_1^\top (\mathbf{X}^\top \mathbf{X})^{-1} x_j$	$(y_j - \hat{\beta} x_j)$	INFLUENCE SCORE	1EXACT SCORE
2.13	-0.09	2.02E-3	1.66	3.38E-3	3.40E-3
-0.72	-2.05	7.10E-5	-1.57	1.12E-4	1.12E-4
2.70	-4.85	7.25E-5	-7.66	5.55E-7	5.55E-7
2.70	-4.85	7.25E-8	-7.65	7.63E-9	7.64E-9
2.70	-4.85	9.97E-10	-7.65	7.65E-11	7.66E-11
2.70	-4.85	1.00E-14	-7.65	7.65E-13	7.66E-13

Table 4 and Table 5 display empirical findings for the data generating process described in Section 4.1, showing that a sufficiently far outlier will have vanishingly low influence score (see Proposition D.1). As the black-dot point moves far from the group of red-cross points in both the x and y directions, both the leverage-like term and the residual term of the influence function approach zero at rate $O(\frac{1}{\lambda})$ (see columns 3 and 4 of Table 4). When observing the behavior of the corresponding red-cross point with the largest influence score in Table 5, we see that the leverage-like term approaches zero while the residual term stays relatively constant (within the same order of magnitude). Thus, for sufficiently large values of (x_i, y_i), the influence score for the black-dot point becomes smaller than that of a red-cross point (see the highlighted values in Table 4 and Table 5); for the (x_i, y_i) values with highlighted influence scores (see Table 4), both AMIP and Greedy AMIP fail (in both the Type 1 and 2 sense). This occurs at $x = y = 10^6$ for the data generating process described in Section 4.1.

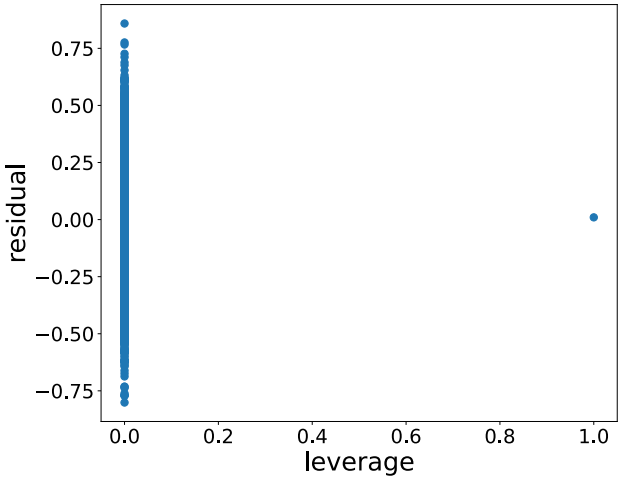

Figure 2: The leverage plotted against the residuals of the data points. The leverage of the extreme outlier is much larger than the leverage of any other data point, while the residual of the outlier is close to zero. In linear regression, AMIP tends to fail in the presence of high leverage points.

D.3 Additional multi-outlier examples

Adversarial Example. Moitra and Rohatgi [2023] presents an adversarially constructed example in which there exists a small fraction of points that can be dropped such that the covariance matrix becomes singular. This leads to a failure mode of approximation algorithms. In Section 4, we attempt to alter this adversarial setup into one that might arise in natural data settings with no adversary (see Figure 1 (right)). We see that even modest levels of instability in the covariance matrix can lead to failure modes in the approximation methods.

To visualize the example presented in Moitra and Rohatgi [2023], we generate the red crosses so as to have a singular covariance matrix (see Figure 3). In particular, we generate the 1,000 red crosses with $x_n = 0$, $y_n = \epsilon_n$, and $\epsilon_n \overset{\text{iid}}{\sim} \mathcal{N}(0, 1)$. We draw the 10 black dots as $x_n \overset{\text{iid}}{\sim} \mathcal{N}(-1, 0.01)$, $y_n = x_n$. When we consider both black dots and red crosses together as a single dataset, there is no poor conditioning. However, when we drop the red population, a pathological change occurs in the covariance matrix; it becomes singular. The OLS-estimated slope on the full dataset is about 1.00; dropping the black dots (0.99% of the data) yields a slope of exactly 0. Note, although the removal of the black-dot points do not induce a sign change in this example, going from a positive signed coefficient to 0 still constitutes a conceivable conclusion change in a data analysis.

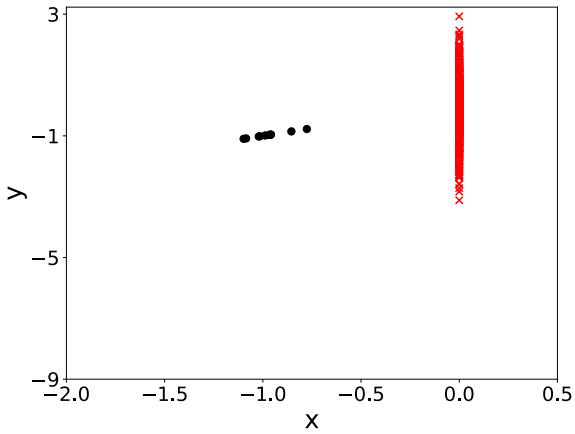

Figure 3: Example of poor conditioning presented in Section 5.1 of Moitra and Rohatgi [2023]

Greedy AMIP Failure Example. In the following example, we illustrate a case in which Greedy AMIP fails (See Figure 4). In particular, Greedy AMIP fails; in particular, when there is one black dot left to remove, Greedy AMIP is unable to identify the black dot as the point to remove. This is because the residual of the last remaining black dot becomes vanishingly small when the second to last black dot is removed in the previous iteration.

In this example, we generate the 1,000 red crosses with $x_n = 0$, $y_n = \epsilon_n$, and $\epsilon_n \overset{\text{iid}}{\sim} \mathcal{N}(0, 1)$. We draw the 10 black dots as $x_n \overset{\text{iid}}{\sim} \mathcal{N}(-1, 0.01)$, $y_n = -5x_n - 10$. The OLS-estimated slope on the full dataset is about 4.94; dropping the black dots (0.99% of the data) yields a slope of about 0.

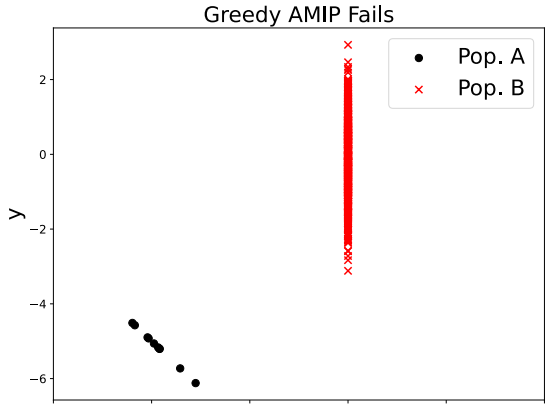

Figure 4: This is an example where Greedy AMIP fails but Greedy 1sN succeeds.

Greedy AMIP and Greedy 1sN Failure Example. In the next example (See Figure 5), by clustering k outliers tightly into a small clump, we can construct an instance where both greedy AMIP and 1sN fail to identify the k outlier cluster. This repeated k points centered around one clump (where k is large) produces an example where the residuals are vanishingly small in the outlier cluster, while the leverage can only be as big as 1/k. Hence, the leave-one-out 1sN scores of certain points in the red inlier population (population B) will be larger. In this instance, if we computed the exact 1sN score for every subset of size k, however, we would be able to correctly identify population A as the Most Influential Set.

In this example, we generate the 1,000 red crosses with $x_n = 0$, $y_n = \epsilon_n$, and $\epsilon_n \overset{\text{iid}}{\sim} \mathcal{N}(0, 1)$. We draw the 10 black dots as $x_n \overset{\text{iid}}{\sim} \mathcal{N}(-1, 10^{-7})$, $y_n = -5x_n - 10$. The OLS-estimated slope on the full dataset is about 4.94; dropping the black dots (0.99% of the data) yields a slope of about 0.

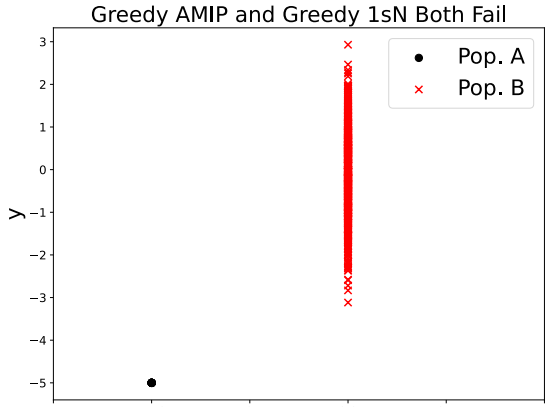

Figure 5: This is an example where Greedy AMIP and Greedy 1sN both fail.

D.4 Approximation error in OLS

As the approximation methods presented in Section 3 are local approximations based on removing individual observations, errors may accrue when there exists subsets of points with high joint influence measures but low individual influence measures.

In linear regression, we can formalize this intuition by looking at an analytic expression for the ordinary least squares estimator of the pth regression coefficient, β. Let $e_p \in \mathbb{R}^d$ denote the pth standard basis vector. Let $\mathbf{X} \in \mathbb{R}^{n \times p}$ denote the design matrix and $\vec{y} \in \mathbb{R}^n$ denote the response vector. Let S denote the dropped set (i.e. the observations indexed by S in the design matrix and response vector) and $\setminus S$ denote its complement. Let $\hat{\beta}^{\text{IF}}(w)$ denote the influence function approximation of $\hat{\beta}$ given in Equation (7) and let $\hat{\beta}^{\text{1sN}}(w)$ denote the 1sN approximation given in Equation (4).

Let $\mathbf{X}_{\setminus n}$ denote the design matrix leaving out data point n, $x_n \in \mathbb{R}^d$ denote the x value of the nth data point, and $r_n = (y_n - \hat{\beta}(1_N)^\top x_n)$ denote the residual value of the nth data point.

The error incurred by AMIP can be written as

$$\begin{aligned}
\hat{\beta}^{\text{AMIP}}(w) - \hat{\beta}(w) &= \sum_{n \in S} e_p^\top (\mathbf{X}^\top \mathbf{X})^{-1} x_n r_n - \sum_{n \in S} e_p^\top (\mathbf{X}_{\setminus S}^\top \mathbf{X}_{\setminus S})^{-1} x_n r_n \\
&= e_p^\top ((\mathbf{X}^\top \mathbf{X})^{-1} - (\mathbf{X}_{\setminus S}^\top \mathbf{X}_{\setminus S})^{-1}) \sum_{n \in S} x_n r_n \\
&= e_p^\top ((\mathbf{X}^\top \mathbf{X})^{-1} - (\mathbf{X}_{\setminus S}^\top \mathbf{X}_{\setminus S})^{-1}) \mathbf{X}_S^\top r_S
\end{aligned} \tag{19}$$

and the error in Additive 1sN can be written as

$$\begin{aligned}
\hat{\beta}^{\text{Add-1sN}}(w) - \hat{\beta}(w) &= \sum_{n \in S} e_p^\top (\mathbf{X}_{\setminus n}^\top \mathbf{X}_{\setminus n})^{-1} x_n r_n - \sum_{n \in S} e_p^\top (\mathbf{X}_{\setminus S}^\top \mathbf{X}_{\setminus S})^{-1} x_n r_n \\
&= \sum_{n \in S} e_p^\top ((\mathbf{X}_{\setminus n}^\top \mathbf{X}_{\setminus n})^{-1} - (\mathbf{X}_{\setminus S}^\top \mathbf{X}_{\setminus S})^{-1}) x_n r_n.
\end{aligned} \tag{20}$$

In contrast, the 1sN approximation gives the exact reweighted ordinary least squares solution (See Appendix C.2.1 and Pregibon [1981, Equation 3])

$$\begin{aligned}
\hat{\beta}^{\text{1sN}}(w) &= e_p^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \vec{y} + e_p^\top (\mathbf{X}_{\setminus S}^\top \mathbf{X}_{\setminus S})^{-1} (\mathbf{X}_S^\top \vec{y}_S - \mathbf{X}_S^\top \mathbf{X}_S (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \vec{y}) \\
&= e_p^\top (\mathbf{X}_{\setminus S}^\top \mathbf{X}_{\setminus S})^{-1} \mathbf{X}_{\setminus S}^\top \vec{y}_{\setminus S}
\end{aligned}$$

such that

$$\begin{aligned}\hat{\beta}^{1sN}(w) - \hat{\beta}(w) &= e_p^\top (\mathbf{X}_{\setminus S}^\top \mathbf{X}_{\setminus S})^{-1} \mathbf{X}_{\setminus S}^\top \vec{y}_{\setminus S} - e_p^\top (\mathbf{X}_{\setminus S}^\top \mathbf{X}_{\setminus S})^{-1} \mathbf{X}_{\setminus S}^\top \vec{y}_{\setminus S} \\ &= 0.\end{aligned} \tag{21}$$

Observe from Equation (19) that the error arises from a failure to correctly reweight the inverse Hessian term by the dropped subset, S. While AMIP disregards this reweighting entirely, Additive 1sN reduces this error by reweighting the Hessian on an individual data point basis, which is still incorrect (See Equation (20)). The exact 1sN (Equation (21)), however, is out of reach for the Maximum Influence Perturbation problem because we do not have access to the dropped subset beforehand.

The error in the AMIP approximation (Equation (19)), can be expressed as an inner product between two vectors, $\vec{m}_{\text{Err}} = e_p^\top ((\mathbf{X}^\top \mathbf{X})^{-1} - (\mathbf{X}_{\setminus S}^\top \mathbf{X}_{\setminus S})^{-1})$ and $\vec{v}_{\text{Err}} = \mathbf{X}_S^\top r_S$. This inner product can be written as $\vec{m}_{\text{Err}} \cdot \vec{v}_{\text{Err}} = \|\vec{m}_{\text{Err}}\| \|\vec{v}_{\text{Err}}\| \cos(\theta)$, where θ is the angle between $\vec{m}_{\text{Err}}$ and $\vec{v}_{\text{Err}}$. A large AMIP approximation error implies at least one of the following properties about the data:

1. The product $\|\vec{m}_{\text{Err}}\| \|\vec{v}_{\text{Err}}\|$ is large.
2. The angle between $\vec{m}_{\text{Err}}$ and $\vec{v}_{\text{Err}}$ is small.

We observe empirically (see Figure 6) that property (2) is more prominent in the Simpson's Paradox Example while property (1) is more prominent in the Poor Conditioning Example. That is, the product of norms is large in the Poor Conditioning Example, 11.15 (this product is notably smaller in the Simpson's Paradox Example), while the angle between vectors is small in the Simpson's Paradox Example, $0.23°$ (this angle is notably larger in the Poor Conditioning Example).

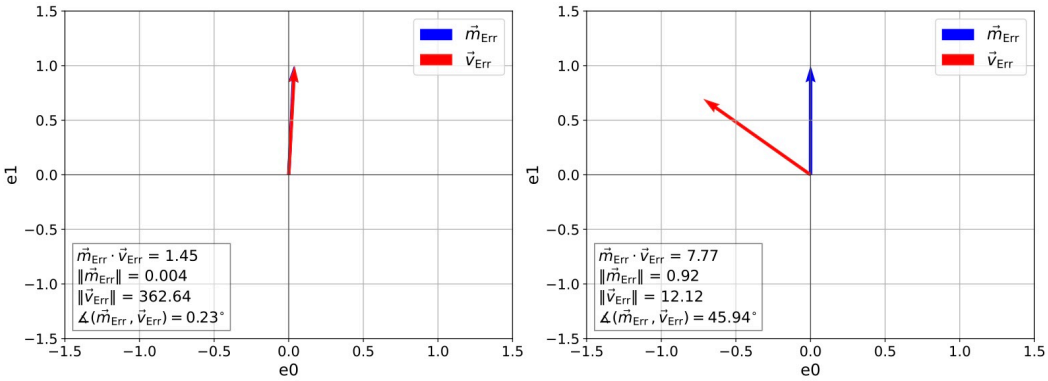

Figure 6: This plot shows the vectors $\vec{m}_{\text{Err}}$ (blue) and $\vec{v}_{\text{Err}}$ (red), against e_0 and e_1 (the standard basis vectors corresponding to the intercept term and x term in our linear regression setup) for the Simpson's Paradox Examples (left) and the Poor Conditioning Example (right). The angle between the vectors is small in the Simpson's Paradox Example (property (2)) while product of norms is large in the Poor Conditioning Example (property (1)).

D.5 Linking failure modes to data arrangements

In order to link the failure modes presented in Appendix D.4 to specific properties of the data arrangements, we present modifications of the failure modes illustrated in Section 4 in order to reduce the effects of either property (1) or property (2).

D.5.1 Simpson's paradox example (modified)

In this section, we modify the Simpson's Paradox Example to increase the angle between $\vec{m}_{\text{Err}}$ and $\vec{v}_{\text{Err}}$ (thereby reducing the prominence of property (2) in Appendix D.4). To increase the angle between the vectors, we shift the outlier cluster in Figure 1 closer to zero (see Figure 7). In the original Simpson's Paradox Example, both $\vec{m}_{\text{Err}}$ and $\vec{v}_{\text{Err}}$ aligned closely with e_1. Note that $\vec{v}_{\text{Err}} = (\sum_{n \in S} r_n, \sum_{n \in S} r_n x_n)$, so $\vec{v}_{\text{Err}}$ aligns with e_1 because the outlier cluster has large x relative

to residual values (see Figure 6). If we shift the outlier cluster (Figure 1) closer to zero, we can reduce the magnitude of the x values and the alignment of $\vec{v}_{\text{Err}}$ with e_1 and $\vec{m}_{\text{Err}}$.

Upon shifting the outlier cluster closer to zero, we do observe that all approximation methods succeed (see Figure 7).

In Figure 7, we generate the 1,000 red crosses with $x_n \overset{\text{iid}}{\sim} \mathcal{N}(0, 0.25)$, $y_n = -x_n + \epsilon_n$, and $\epsilon_n \overset{\text{iid}}{\sim} \mathcal{N}(0, 1)$. We draw the 10 black dots as $x_n \overset{\text{iid}}{\sim} \mathcal{N}(1, 0.25)$, $y_n = -x_n + 40$, and $\epsilon_n \overset{\text{iid}}{\sim} \mathcal{N}(0, 1)$. The OLS-estimated slope on the full dataset is about 1.11; dropping the black dots (0.99% of the data) yields a slope of about -0.99.

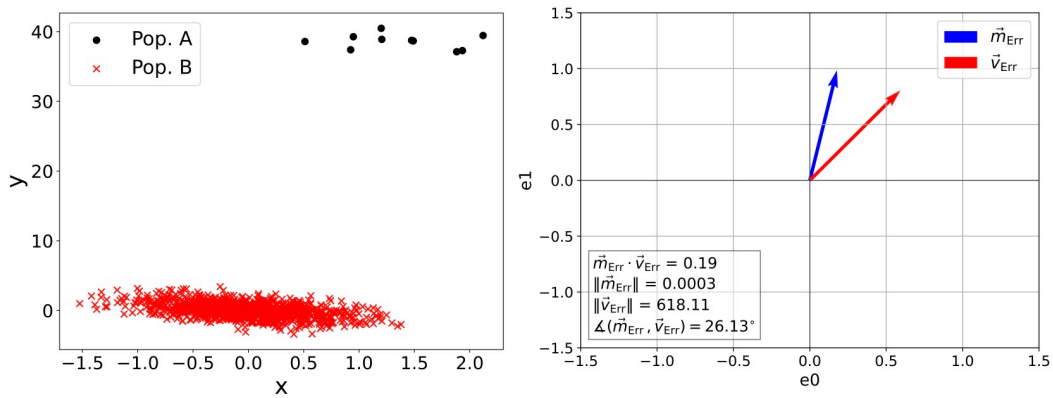

Figure 7: By shifting the outlier cluster (black) closer to zero, we can modify the Simpson's Paradox Example to reduce the alignment of $\vec{v}_{\text{Err}}$ with $\vec{m}_{\text{Err}}$ and find that all approximation methods succeed.

D.5.2 Poor conditioning example (modified)

In this section, we modify the Poor Conditioning Example to decrease $\|\vec{m}_{\text{Err}}\|\|\vec{v}_{\text{Err}}\|$ (property (1)) in Appendix D.4. Specifically, by increasing the variance of x in the red cluster (making the covariance matrix of the red cluster well-conditioned),[3] we can decrease $\|\vec{m}_{\text{Err}}\|$. Indeed, by performing this modification, we see a three magnitudes decrease in $\|\vec{m}_{\text{Err}}\|$, which results in a decrease of the overall product, $\|\vec{m}_{\text{Err}}\|\|\vec{v}_{\text{Err}}\|$, from 11.15 to 0.115. Upon making this modification, we observe that all approximation methods presented succeed.

In the example in Figure 8, we generate 1,000 red crosses with $x_n \overset{\text{iid}}{\sim} \mathcal{N}(0, 0.1)$, $y_n = \epsilon_n$, and $\epsilon_n \overset{\text{iid}}{\sim} \mathcal{N}(0, 1)$ (notice the 100 fold increase in variance from the original example). We draw the 10 black dots as $x_n \overset{\text{iid}}{\sim} \mathcal{N}(-1, 0.01)$, $y_n = -x_n - 10 - \epsilon_n$, and ϵ_n as before. When we remove the black dots now, there is no longer the issue of poor conditioning in the red crosses Figure 8. The condition number of the design matrix in the failure mode example presented in Section 4 is 31.68, whereas the condition number within this example is ten times smaller, at 3.17. The OLS-estimated slope on the full dataset is 0.73; dropping the black dots (0.99% of the data) yields a slope of -0.11, which is a sign change.

$$\begin{aligned}\vec{m}_{\text{Err}} &= e_p^\top((\mathbf{X}^\top\mathbf{X})^{-1} - (\mathbf{X}_{\setminus S}^\top\mathbf{X}_{\setminus S})^{-1}) \\ &= e_p^\top(\mathbf{X}_{\setminus S}^\top\mathbf{X}_{\setminus S})^{-1}\mathbf{X}_S^\top(I + \mathbf{X}_S(\mathbf{X}_{\setminus S}^\top\mathbf{X}_{\setminus S})^{-1}\mathbf{X}_S^\top)^{-1}\mathbf{X}_S(\mathbf{X}_{\setminus S}^\top\mathbf{X}_{\setminus S}^\top)^{-1} \text{ (Woodbury Matrix identity)}\end{aligned} \tag{22}$$

[3]Equation (22) shows us that cases of poor conditioning of $\mathbf{X}_{\setminus S}^\top\mathbf{X}_{\setminus S}$ presents opportunities for large errors in the approximations. Specifically, if the smallest eigenvalue of $\mathbf{X}_{\setminus S}^\top\mathbf{X}_{\setminus S}$ is large (i.e. the matrix is well-conditioned), then $\|\vec{m}_{\text{Err}}\|$ cannot be large (see Equation (22)).

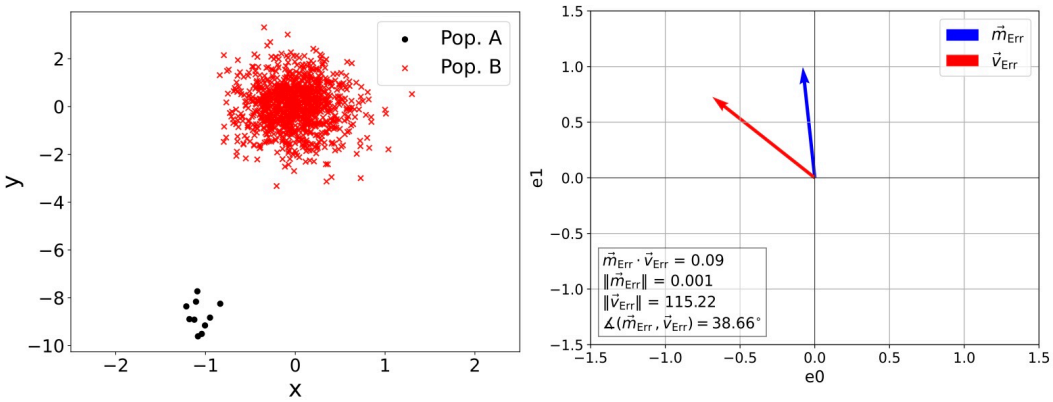

Figure 8: By increasing the variance in the red population (making the design matrix of the red population well-conditioned), we can modify the Poor Conditioning Example to reduce the error arising from $\|\vec{m}_{\text{Err}}\|\|\vec{v}_{\text{Err}}\|$ and find that all approximation methods succeed.

E Limitations and next steps

We here discuss open problems and future work in greater detail.

More complex models. The analysis we consider in the present paper is very specific; we expect some insights will hold in other cases, and some will not. For instance, 1sN has not been developed for other quantities of interest (such as the endpoint of a confidence interval) – and if it were developed, we do not necessarily expect it to be exact when dropping a single data point; we likewise do not expect it to be exact when dropping a single data point but using a different loss. On the other hand, we observe that the poor conditioning example might be expected to hold in other common cases with poor conditioning, such as multicollinearity when more covariates are present. There are a number of interesting directions for future work. For one, we hope to understand how high-dimensional covariates affect failure. For another, we hope to understand how the geometries of more complex models affect failure. For instance, in linear regression, many examples seemed to hinge on the square loss favoring small residuals for an outlier point; we would not see this behavior even in a basic logistic regression. But multicollinearity can still plague generalized linear models.

Before testing beyond linear regression, we first need to extend the Additive 1sN approximation to more general quantities of interest and general Z-estimators. The goal would be an extension that applies automatically, via the use of autodiff, as in AMIP. We anticipate that we can borrow techniques from AMIP's development for the former extension. The latter extension may be more difficult.

There are already other known failure modes of AMIP in more complex scenarios – for instance, in constrained parameter spaces [Broderick et al., 2020] or in hierarchical models under an extension of AMIP to Markov chain Monte Carlo-based analyses [Nguyen et al., 2024]. It remains to better understand the mechanisms of these failures and whether an alternative approximation could solve them.

While greedy methods offer accuracy advantages for relatively simple data analyses, we expect that for sufficiently complex data analyses, the $\lfloor \alpha N \rfloor$ cost multiplier will become prohibitive. One potential option would be to run the AMIP or Additive 1sN, offer the user an estimate of how much longer a greedy approach would take, and let the user decide if they are willing to incur the cost. The tradeoffs remain to be explored.

This work highlights the importance of visualizing data in conjunction with automated robustness checks and highlights room for methodological development in approximating the Maximum Influence Perturbation.

F Experiments compute resources

All experiments were conducted on a personal computer equipped with an Apple M1 Pro CPU at 3200 MHz and 16 GB of RAM. Each experiment took approximately 5 to 10 seconds to run.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: In this work, we compare existing approximations for worst-case data dropping (Section 3) and identify failure modes of these approximations in OLS linear regression (Section 4). Certain examples we identify reflect masking (Section 4). We provide recommendations for the user and suggest way forward for methodological development (Section 5).

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss the limitations of our current work and next steps for future work in Section 5 and Appendix E.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: See Appendix D.2 for the assumptions and proof associated with the One-Outlier example.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The information needed to reproduce the experimental results are detailed before each result in Section 4 as well as in Appendix D.3.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We provide a link to the code, to reproduce all experiments, in Appendix A.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: All details on how to recreate experiments are given in Section 4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: This question does not apply, given the nature of our experimental setup. We identify example cases where data-dropping approximations break down.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We detail the computer resources for the experiments in the paper in Appendix F.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the code of ethics, and the paper does conform with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: In Section 1, we discuss potential positive societal impacts of our work. As we do not introduce any new model or technologies, we do not anticipate this work to have any negative societal impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: The paper does not use existing assets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.