We thank all reviewers for their comments. Minor points will be dealt with in the revised version of our article.

R#2: My major concern...provided that m = O(k log n). The problem is actually much more subtle. It is not hard to show that the sparse vector is, in fact, *not identifiable even with infinitely many samples*, if the only assumption one makes is that a constant fraction of the measurements survive truncation. Here's a simple example: consider sampling (a) a Gaussian N(0,1), versus (b) a Gaussian N(0,2). We can define a randomized truncation procedure for situation (a), and for situation (b), such that the truncated samples from (a) are identically distributed to those from (b). Only a constant fraction of the observations are removed. Nonetheless, no estimation of the variance is possible. This can be recast as a linear regression problem with n=k=1; even for m arbitrarily large, recovery is impossible.

Another point we want to make here is why a two-step argument of the following form fails: (1) first show that, if a constant fraction of the measurements survive, then the measurement matrix corresponding to the surviving samples is incoherent (RIP/REC/whatever); (2) then use existing results in the literature implying that such incoherence conditions are sufficient for sparse recovery. This is, indeed, a natural approach but there is a subtle reason why it fails. In order to black box existing recovery results we need that the noise added to the measurements is independent of the measurement matrix. Unfortunately, in our case the measurement matrix of the surviving measurements depends on the noise, breaking the independence assumption, and thus invalidating this simple approach. It is true, however, that without measurement noise this approach works as we point out in Theorem 3.4.

It is not convincing...expression of the function f? We did not think about trying to improve the rate, since there are generic methods to amplify to $1/\text{poly}(n)$ by repeatedly running the algorithm on independent samples, at the cost of $O(\log n)$ extra multiplicative factor in sample complexity. However, you are correct that this is suboptimal. The suboptimality is entirely due to a single bound (Lemma A.1), and it turns out that it is fairly straightforward to modify Lemma A.1 to achieve $1/\text{poly}(n)$ failure rate, which leads to $1/\text{poly}(n)$ failure rate in Theorem 3.1, matching the exact type of bounds known for the untruncated case. Specifically, Lemma A.1 relies on a first moment bound and Markov's inequality. We can instead prove sub-Gaussianity of the truncated Gaussian $y - \mathbb{E}[y]$ conditioned on A. We can then bound the sub-Gaussian variance proxy of $(A^{\mathrm{T}})_i \cdot (y - \mathbb{E}[y])$ (holding A fixed) with high probability over A. Finally apply sub-Gaussian tail bounds, and sum over $i \in U$. With this modification, the algorithm in our paper and all statements that use the phrase "with high probability" achieve success probability $1 - n^{-100}$ with $O(k \log n)$ samples.

I am disappointed that the proof...high probability event. Theorem 3.1 follows immediately from the concatenation of Propositions 3.2 and 3.3, which we provide proofs for. We will include a statement to this effect in our next revision. Again, with the above modification the probability of success is $1 - 1/\text{poly}(n)$ when $m = O(k \log n)$.

The term "adversarial noise" is misleading... We could say "arbitrary bounded noise."

PDW in Wainwright 2009 requires a lower bound...difference? We are not sure where Wainwright requires a lower bound on $|\hat{x} - x|$. Our paper proves an upper bound on the reconstruction error as a function of n, k, m.

I find Lemmas 5.7 and 5.8...sparse directions? Due to lack of (unrestricted) strong convexity, SGD only guarantees that $f(x^t)$, where x^t is the t-th iterate, converges to $f(\hat{x})$ ($\hat{x}$ is optimum of regularized NLL program). Section 5.2 shows that in fact x^t converges to $\hat{x}$. Specifically, Lemma 5.8 shows that if $f(x)$ is close to $f(\hat{x})$, then x is close to $\hat{x}$. The algorithm never needs to test that $f(x) - f(\hat{x}) < \log n/m^3$. This condition is guaranteed by SGD. We do lower bound the Hessian's smallest singular value in the sparse support of x^* (this is Lemma 5.7). However, this is not sufficient to prove that SGD converges, since SGD's iterates are dense.

My overall feeling...Wainwright 2009 We are very familiar with the results of both [Daskalakis et al. 2019] and [Wainwright 2009]. While we use primal-dual witness and log-likelihood, there are a number of innovations required for high-dimensional truncated regression but not in the prior work, as explained in Section 1.1 (overview of proofs and techniques). On a technical level, we cite three lemmas from [Daskalakis et al. 2019]. Aside from these, the results of the prior work do not have useful instantiations in the high-dimensional setting. The structure of our proof is quite distinct from that of [Daskalakis et al. 2019], since in their case the special projection set yields all the properties needed for SGD convergence, but in our case there can be no such projection set.

R# 3: Can the authors comment...running time? The algorithm is fairly straightforward to implement; each update of the SGD requires a sampling step and a projection. In comparison to [Daskalakis et al. 2019], the sampling step is trickier (described in Section K of the supplementary); however the projection is simpler. From our theoretical results we achieve polynomial dependence of $\tilde{O}(nk^6)$ for $m = O(k \log n)$. We expect that in practice fewer iterations are needed, but we do not have experimental results to confirm or deny this belief.

R# 4: (regarding experimental work) After this work in which we established the necessary theoretical framework for this problem, we agree that the next relevant step is to try our method both on simulated and on real-world data where truncation occurs.