We would like to thank the reviewers for their detailed and insightful comments.

[Performance of SGD and robustness of optimization algorithms.] We have resolved the concerns with SGD. By increasing the batch size towards the last iterations and averaging the last iterates, SGD on the adaptive Gaussian sketch problem performs better in terms of time vs accuracy performance, compared to SGD on problem (1) or on the oblivious Gaussian sketch problem. We have similar results with Adam. As reported in the submission, SVRG on the adaptive Gaussian sketch performs better than SVRG on problem (1), and is robust to the choice of hyperparameters. Further, Sub-sampled Newton (with mini-batch Hessian and full-batch gradient) has a strong time vs accuracy performance on adaptive Gaussian sketch. In the revised version, we will include our new results for SGD and Adam, and a sensitivity analysis to sketching, batch and step sizes, for all algorithms applied to the sketched problems (adaptive and oblivious).

[Comparison with other sketching baselines.] We carried out extensive numerical evaluations of oblivious Gaussian sketching and adaptive sketching with uniform column sub-sampling matrix (Nystrom method) on MNIST and CIFAR10. For a wide range of values of sketching size m and regularization parameter λ, adaptive Gaussian sketching always strongly beats oblivious sketching, and, outperforms Nystrom method, both in terms of final test accuracy (see Table 1 below), and, time vs accuracy performance for the following algorithms: SGD, SVRG, Sub-sampled Newton and Adam. Further, adaptive Gaussian sketching matches the performance of x^* for relatively small values of m. We will include all these results in the revised version. **[Computational issues with $(S^\top S)^{-\frac{1}{2}}$ for large m.]** Thanks to this question,

Table 1: Test classification error on MNIST and CIFAR10, for 10-classes classification. "AG": Adaptive Gaussian sketch, "Ob": Oblivious Gaussian sketch, "N": Nystrom method, x_m: solution obtained from problem (2) with sketching size m. We mapped MNIST (resp. CIFAR10) images through 10000 (resp. 60000) random cosines.

λ	x^*_{MNIST}	x^{AG}_{256}	x^{AG}_{1024}	x^{Ob}_{256}	x^{Ob}_{1024}	x^{N}_{256}	x^{N}_{1024}	x^*_{CIFAR}	x^{AG}_{256}	x^{AG}_{1024}	x^{Ob}_{1024}	x^{N}_{256}	x^{N}_{1024}
$5 \cdot 10^{-5}$	4.6 %	4.0 %	4.5 %	25.2 %	8.5%	5.0 %	4.6 %	51.6 %	50.6%	51.0%	70.5%	55.8%	53.1%
$5 \cdot 10^{-6}$	2.5%	2.8%	2.4%	30.1%	9.4%	3.0%	2.7%	47.6%	51.9%	45.8%	80.1%	57.2%	55.8%

we have improved our results and we can show that the matrix $(S^\top S)^{-\frac{1}{2}}$ can be replaced by any other pre-conditioner Q, and in particular, for large m, a matrix Q obtained by approximate SVD. Provided $\|Q - (S^\top S)^{-\frac{1}{2}}\|_2$ is small, then the condition number of (7) remains close to that of (1). Importantly, it does not affect any of the bounds on $\widetilde{x}$. We will include these results in the revised version.

[For small m, would dynamically modifying the sketching matrix lead to tighter bounds?] For small m, we tried numerically to refresh the sketching matrix at each iteration and it did not yield good results. However, our Algorithm 2 refreshes the sketching matrix at the end of each optimization, and gives tighter bounds.

[Results on CIFAR10 far from state-of-the-art. Other optimization problems for which the method could be demonstrated?] We did additional experiments with features extracted from a pre-trained neural network, and $\widetilde{x}$ matched exactly the test error of x^* ($\sim 10\%$). We will include these results in the final version. Beyond classification, large-scale generalized linear models (other than least squares) can be addressed with our method.

[Unclear if SGD (without sketching) converges to the same solution or performs better]. The reported results correspond to the best SGD solution we obtained (with grid search of the batch and step sizes), even through longer time horizons.

[Value of λ used for synthetic experiments?] We used $\lambda = 10^{-4}$. Thank you for pointing this out, we will fix it.

[Title suggestion.] We agree with the relevant title suggestion. **[Non-convexity].** We derived a new result regarding non-convex, smooth functions f: if α^* is a nearly-stationary point for the sketched problem (2), then $\widetilde{x}$ is a nearly ε-stationary point for problem (1), where ε controlled again by the tail spectral decay of A. **[Regularity assumptions on f].** We will add some discussion in the main body of the paper. In Appendix E, guarantees are provided for convex, non-smooth objectives f. **[Other regularizers.]** We have extended our analysis to smooth, strongly convex regularizers. However, extension to the L_1-norm is an open question.

[Invoking the representer theorem not necessary in Eq. (11).][Notation 5.10^{-5} non-standard.] We will simplify the argument for Eq. (11) in the revised version, and correct the notation. Thank you for pointing this out.

[Intuition for why the proposed approach works.] We will discuss more carefully some intuition in the revised version. In a nutshell, the kernelized version of optimization problem (1) is well approximated by $\min_w f(AP_S A^\top w) + \lambda \|P_S A^\top w\|^2$, provided that $AA^\top \approx AP_S A^\top$. Adaptive sketching works better than oblivious one, since $\|AA^\top - AP_{A^\top \widetilde{S}} A^\top\|_2 \ll \|AA^\top - AP_{\widetilde{S}} A^\top\|_2$, for $\widetilde{S}$ i.i.d. Gaussian.

[Comparison of the proposed method with approximate kernel methods] Random Fourier features lead to problems of type (1). Standard Nystrom methods approximates the matrix K in (11). But both problems (1) and (11) are typically high-dimensional. Our method is a dimension-reduction tool, that can be used on top of approximate kernel methods.