We thank the reviewers for their time and attention. Below, we address each review in turn. In addition, we describe a subtle technical correction we made to our paper shortly after submission, which does not change our runtime guarantees.

Reviewer 1 Thank you for your review; we are glad you found our paper interesting and well-written. Below, we address in detail the two concerns raised in the review, starting with a comparison to Sherman (2017). We hope this comparison meets your requirement for raising our paper's score.

Comparison to algorithms for ℓ_∞-ℓ_1 games. The recent papers by Sherman (2017) and Sidford and Tian (2018) consider a different setting than we do, and their developments do not imply runtime improvements for our setting. Specifically, these papers consider bilinear saddle-point problems $\min_{x\in\mathcal{X}}\max_{y\in\mathcal{Y}} y^\top Ax$ where the domain $\mathcal{X}$ is the box (ℓ_∞ ball) while $\mathcal{Y}$ is the simplex. As Sherman explains in his introduction, the ℓ_∞ domain is challenging because no distance generating function has both 1-strong-convexity w.r.t. ℓ_∞ and range sublinear in dimension. Sherman's development side-steps this challenge using finer-grained notions of convexity and Nesterov's dual extrapolation method to obtain improved runtime guarantees for ℓ_∞-ℓ_1 games. Sidford and Tian attack this challenge using local notions of smoothness and a randomized coordinate method, and obtain improved runtimes for column-sparse A.

In contrast, this challenge does not exist in the ℓ_1-ℓ_1 and ℓ_1-ℓ_2 games that we study, because in these settings suitable distance generating functions are readily available: negative entropy for ℓ_1 and Euclidean norm for ℓ_2. Consequently, the ideas in Sherman (2017) and Sidford and Tian (2018) do not imply runtime improvements for our settings. These works do not consider variance reduction, on which our paper crucially relies. In future work, we intend to explore whether our variance reduction techniques can provide benefits in the ℓ_∞-ℓ_1 setting. When revising our paper we will be sure to cite Sherman (2017) and Sidford and Tian (2018) and compare them to our development—thank you for pointing out the importance of this comparison.

The range of the distance generating function (dgf). For ℓ_1-ℓ_2 games the range of our dgf is $\frac{1}{2}+\log m$. More generally, in our paper the range Θ does not introduce polynomial dependence on problem dimension; the runtime bounds in Theorems 1 and 2 account for Θ and also include logarithmic terms. For the n-dimensional simplex (ℓ_1 domain) we use negative entropy as the dgf, and it has range $\Theta = \log n$. For the unit Euclidean ball (ℓ_2 domain), our dgf is half the Euclidean norm, so that $\Theta = 1/2$, regardless of the dimension. (Note that, as is standard in the literature, we consider simplices and Euclidean balls of unit norm. This is without loss of generality as scaling of the domain is equivalent to scaling of the matrix A, and we account for its norm via the parameter L.) In Eqs. (1) and (2) in the introduction we substituted the relevant values of Θ into the runtime guarantees. However, since this creates confusion, we will revise the introduction to clarify the contribution of the range Θ to the runtime bound.

Reviewer 2 Thank you for the kind review; we are pleased that our development came across clearly. We hope that our repackaging of Nemirovski's ideas and our "sampling from the difference" technique will inspire and assist future improvements in minimax optimization and variational inequalities.

Reviewer 3 Thank you for the generous review and for recognizing the novelty and significance of our results. Indeed, extending the regime in which stochastic (and possibly variance-reduced) methods aid minimax game solution is an exciting direction for further research, in which we are currently engaged.

Averaged vs. random iterates In convex optimization, returning a random iterate and returning the average of the iterates often result in equivalent guarantees. However, this is not the case in our paper, and we must do the latter. Therefore, we changed Algorithm 1 (`OuterLoop`) to run for the full K iterations (no random stopping) and return the average $\bar{z}_K = \frac{1}{K}\sum_{k=1}^{K} z_{k-1/2}$. This way, the proof of Lemma 1 implies a bound on the duality gap at $\bar{z}_K$ as defined in line 154, via standard convexity arguments (cf. [23] page 8); we do not get this guarantee with a random iterate.

Similarly, we changed Algorithm 2 (`InnerLoop`) to run for T iterations and return $\bar{w}_T = \frac{1}{T}\sum_{t=1}^{T} w_t$. This way, for bilinear games $\bar{w}_T$ satisfies the α-proximal oracle property. To see this, note that when $g(w) = (A^\top w^{\mathsf{y}}, -Aw^{\mathsf{x}})$ we have $\frac{1}{T}\sum_{t=1}^{T}\langle g(w_t), w_t - u\rangle = \langle g(\bar{w}_T), \bar{w}_T - u\rangle$ and therefore the α-proximal oracle property holds by the bound in Lemma 2; we do not get this property with a random iterate. In the revised paper we also include a proximal oracle implementation valid for general games (i.e. not only bilinear), that returns the last iterate (i.e. does not perform averaging) but requires a number of restarts logarithmic in $1/\epsilon$.

We note that the proofs of Lemmas 1 and 2 in the submitted paper are already fully compatible with these algorithmic modifications, and that our main results (Theorem 1, Corollary 1 and Theorem 2) hold unchanged.