We thank all three reviewers for their thorough reviews and constructive feedback.

Why not Newton?:
As remarked by reviewers #6 and #7, retaining the diagonal blocks of the Hessian in the approximation results in a regularized and damped Newton method $(\Delta x, \Delta y) = (\mathrm{Id} + \eta J)^{-1}(\nabla_x f, \nabla_y g)$, where J is the Jacobian of the vector field $(\nabla_x f, \nabla_y g)$. We agree that the present version of the paper lacks a thorough discussion of the reasons for ignoring the diagonal parts of the Hessian and will therefore add the following three reasons.
Blow-up of condition number: Including the diagonal blocks of the Hessian in the non-convex-concave setting can make the matrix inverse arbitrarily ill-conditioned as $\|\eta D^2_{xx} f\|$ or $\|\eta D^2_{yy} f\|$ approach or exceed 1, greatly increasing the cost of the linear system solve. In contrast, for zero-sum games the condition number of the matrix inverse in CGD is always bounded from above by $(1 + \eta^2 \|D^2_{xy} f\|^2)$.
Irrational updates: For $\eta\|D^2_{xx} f\|$ or $\eta\|D^2_{yy} f\|$ bigger than 1 and f non-convex-concave, the regularized Newton update can loose its game-theoretic interpretation as a local strategic equilibrium, allowing for convergence to highly non-optimal critical points. While we leave the full characterization of the attractors of CGD for future work, we expect them to always be game-theoretically meaningful since the updates of CGD arise as local Nash equilibria.
Lack of regularity: For the diagonal blocks of the Hessian to be useful in optimization, we need to make additional assumptions on the regularity of the loss function, for example by bounding the Lipschitz constants of $D^2_{xx} f, D^2_{yy} f$. Otherwise, including additional second order information can make the results worse. Consider for instance, minimizing $x \mapsto x^2 + \epsilon^{3/2} \sin(x/\epsilon)$ for $\epsilon \ll 1$. Many minimax problems, for example GANs, have the form $f(x, y) = \Phi(\mathcal{G}(x), \mathcal{D}(y))$ where Φ is *smooth* and *simple* but $\mathcal{G}$ and $\mathcal{D}$ might only have first order regularity. In this setting, the bilinear approximation has the advantage of fully exploiting first order information of $\mathcal{G}, \mathcal{D}$, without assuming them to have higher degrees of regularity. This is because the bilinear approximation of f then contains only the first derivatives of $\mathcal{G}$ and $\mathcal{D}$, while the quadratic approximation contains second derivatives $D^2_{xx}\mathcal{G}$ and $D^2_{yy}\mathcal{D}$, and therefore needs stronger regularity assumptions on $\mathcal{G}$ and $\mathcal{D}$ to be effective.

Reviewer #3:
***"...convergence rate results for CGD..."*:** Under lower (upper) bounds on $D_{xx} f$ ($D_{yy} f$), global exponential convergence can be derived from Theorem 2.2, and we are happy to include this result with the revisions. A special case of this is strong convex-concavity. We are not aware of existing work on minimaximization that provides global convergence proofs without either convex-concavity/monotonicity, or strong additional assumptions.
***"...CGD still requires that the step-size is bounded by one over the max diagonal entry of the Hessian..."*:** Correct! This is the analogous requirement to applying gradient descent to the single player game (keeping the other player fixed). For problems like GANs the problem of optimizing one player while keeping the other player fixed can be solved reliably via gradient descent while the two-player game becomes unstable under alternating gradient descent. The purpose of CGD is to solve two-player games with similar step sizes and stability properties as when using gradient descent to optimize one player while the other player is kept fix.

Reviewer #6:
Concern 1: ***Why not use full second order?*** See first paragraph.
Concern 2: ***Complexity of matrix inverse?*** We are sorry for the misunderstanding and will try to make line 289 more precise: Figure 5 does **not** show the convergence as a function of the iteration count, but as a function of the number of gradient evaluations and Hessian-vector products. Thus, a single step of CGD that needs k iterations of conjugate gradient to solve the linear system in the update rule will amount to an x-value of $(4 + 2k)$, while a single step of optimistic gradient descent ascent (OGDA) corresponds to an x-value of 2 in the plot. Thus, this measure of cost fairly accounts for the complexity of the matrix inverse in CGD. We find that the convergence rate of CGD is competitive throughout a wide range of step sizes and we explain this fact with the *graceful reduction to linearized CGD* described in Line 274: If the gain from the matrix inverse is small, the matrix will be well-conditioned and thus easy to invert. See also our answer to Reviewer #7.

Reviewer #7:
Concerns 1 & 2: ***Why drop diagonal blocks of Hessian? Why use bilinear approximation?*** See first paragraph.
Concern 3: ***Is CGD scalable?*** Since mixed mode automatic differentiation allows to compute Hessian vector products with minimal overhead compared to gradient computations using reverse mode automatic differentiation, we see no reason why CGD should be restricted to small problems. While larger problems will tend to require more iterations of conjugate gradient, they also tend to negatively affect the existing methods, with the experiments in Figure 5 suggesting that the advantage of CGD over existing methods *increases* as the problems size increases. We are presently working on an implementation of CGD using JAX (which provides GPU accelerated mixed mode automatic differentiation) to then apply to large GAN problems.