We thank all the reviewers for their detailed reviews and valuable comments. We very much appreciate the positive feedback and are very happy that the reviewers generally like our contribution. In what follows, we focus on the most significant comments raised by the reviewers; we will carefully implement all minor suggestions as well as improve the exposition in Section 5 as pointed out by R2 and R3. Code for experiments and proofs can be found in the supplementary material for reproducibility (R2). Further, we will add a clarification in the related work regarding first linear convergence results by Garber and Hazan 2014.

Q1. Framework to unify and derive a subset of prior CG/FW results (Reviewer 3, 4). We highlighted some connections to prior CG/FW results in Sections 3, 4, Table 1 (appendix), however we will make these more explicit. We consider the shadow of the gradient, i.e., its directional derivative, as a descent direction. We show that (i) the continuous time dynamics of moving along the shadow at any point is equivalent to the continuous time dynamics of projected gradient descent (PGD). (ii) In PGD, a constrained movement along the gradient (e.g., $-\frac{1}{L}\nabla f(\mathbf{x}_t)$) is projected back to the polytope. We show that the limit of infinite movement along the gradient (i.e., $\lim_{w\to\infty} -w\nabla f(\mathbf{x})$) followed by a projection recovers the Frank-Wolfe (FW) vertex (R4: at a high level, we called this "maximal wrap" around the polytope), thereby establishing a novel interpretation of FW steps [Frank & Wolfe 1956]. (iii) We show the PGD iterate (i.e., $\Pi_P(\mathbf{x}_t - \frac{1}{L}\nabla f(\mathbf{x}_t))$) lies on the projections curve starting from $\mathbf{x}_t$. (iv) Next, we show that the shadow direction is the best normalized (unit norm) local direction for descent in terms of the dot product with the gradient. This implies it is the best direction in the convex hull of all possible away directions, i.e., the set considered by [Bashiri et. al 2017]. (v) Pairwise-steps considered by [Jaggi & Lacoste-Julien (2015)] are the sum of FW steps and away directions, therefore, the best pairwise-steps could similarly be obtained using our framework. (vi) [Bashiri et. al 2017] and [Garber & Hazan 2016] both compute the best away vertex in the minimal face containing the current iterate, whereas the shadow step recovers the best convex combination of such vertices aligned with the negative gradient. Therefore, these previously mentioned CG methods are *both* approximations of SHADOW-CG. (vii) Moreover, [Garber & Hazan 2014] restrict the FW vertex to a ball around the current iterate, thereby normalizing the norm of the descent direction. Their algorithm can be therefore understood as an approximation of SHADOW-WALK.

Q2. Theoretical differences between SHADOW-CG and SHADOW-WALK (Reviewers 1, 3, 4). SHADOW-WALK moves along the shadow direction and traces the projection curve until a "non-boundary step" is taken. SHADOW-CG, on the other hand, either moves along the FW or shadow direction depending on the dual gap test (choose FW direction whenever $\langle -\nabla f(\mathbf{x}_t), \mathbf{d}_t^{\text{FW}} \rangle \geq \langle -\nabla f(\mathbf{x}_t), \mathbf{d}_{\mathbf{x}_t}^{\Pi}/\|\mathbf{d}_{\mathbf{x}_t}^{\Pi}\| \rangle = \|\mathbf{d}_{\mathbf{x}_t}^{\Pi}\|$, and shadow direction otherwise). Theoretical bound on iteration complexity for a given fixed accuracy is better for SHADOW-WALK, since it is closer to PGD. However, the computational complexity for SHADOW-CG is better compared to SHADOW-WALK since FW steps are much cheaper to compute compared to the shadow direction and we can avoid the potentially expensive computation via the TRACE-routine. This is also observed in the experiments (see Q4).

Q3. Independence of the convergence rates on geometric constants (Reviewers 1, 4). As the reviewers note, we prove an *algebraic* upper bound of $O(2^m)$ breakpoints in the projections curve, for polytopes with m facets in n dimensions; however computationally we see much fewer oracle calls (e.g., rarely exceeding 10 in 100 dimensions). Similar in nature to the PAIRWISE-FW (which is arguably one of the fastest CG variants in practice, yet lacks a more sophisticated analysis), we unfortunately have to resort to a highly pessimistic worst-case bound. We have talked to various discrete geometers and there was no consensus on what the right answer for the order of number of breakpoints should even be. We believe that once the projection curve leaves a facet it can never reenter, and hence we conjecture $O(mn)$ breakpoints in lines 311-313 of the paper, which would significantly improve upon the $O(2^m)$ bound. Although our linear convergence rate depends on the the number of facet inequalities m and in fact the combinatorial structure of the face-lattice of the polytope, *it is invariant under any deformations of the actual geometry of the polytope preserving the face-lattice* (in contrast to vertex-facet distance and pyramidal width), e.g., Figure 4's discussion shows pyramidal width can become arbitrarily small while the number of facets is invariant. We will make this distinction more precise.

Q4. Computational differences in SHADOW-CG and SHADOW-WALK (Reviewers 1, 2, 3, 4). We provide two ways to compute the shadow: (i) using equation (8), i.e., $\mathbf{d}_{\mathbf{x}}^{\Pi} = (\Pi_P(\mathbf{x} - \epsilon\nabla f(\mathbf{x})) - \mathbf{x})/\epsilon$ for ϵ sufficiently small; (ii) as stated in line 155 $\mathbf{d}_{\mathbf{x}}^{\Pi} = \arg\min_{\mathbf{d}}\{\|-\nabla f(\mathbf{x}) - \mathbf{d}\|^2 : A_{I(\mathbf{x})}\mathbf{d} \leq \mathbf{0}\}$. Hence, either way, computing the shadow reduces to a convex quadratic program. Note, that the cost of the shadow computation has to be non-trivial, as it was shown in [Garber 2020, Diakonikolas et al. 2019] that we cannot eliminate the dependence on the dimension in convergence rates of CG variants *that solely rely on linear optimization as oracle*, whereas our rates for SHADOW-WALK are independent of the dimension (i.e., $(1 - \mu/L)$ contraction). That being said, even though computing the shadow is expensive in the worst case, it can be cheap for specific setups and there are many existing algorithms that provide cheap approximation to this oracle, e.g., DICG variants [Garber and Hazan 2016, Bashiri et. al 2017]. Our computations confirm that SHADOW-CG interpolates between CG variants and PGD. In particular, Figure 2 for the Video Co-Localization problem shows SHADOW-CG has lower iteration count than CG variants (slightly higher than PGD), while also improving on wall-clock time compared to PGD and SHADOW-WALK being almost as fast as CG without assuming shadow oracle access. Addition of FW steps in SHADOW-CG significantly reduces the total number of shadow oracle calls and number of times we enter TRACE-routine as demonstrated for the Lasso regression problem in Figures 10 and 13 (appendix).