Response to the reviewers

Manuscript: The Constrained LASSO is a Portfolio: A numerical check of the CLA–LASSO identity under constraints **Journal:** Journal of Reproducible Statistics **Decision:** minor revision

I thank the editor and the three reviewers for careful reports. Every required and recommended item is adopted below except one, Reviewer 1's runtime comparison, which is answered with a reason in the text. One recommended item, more seeds, found something new: three defects in `cvxcla`, fixed in release 2.7.1, and two further declined draws. Both are reported under R2.2. Section and table numbers refer to the revised manuscript.

Required

R2 (required). State the exact objective of the linear program behind the KKT certificate. Adopted (Section 5.1). The text now states what the program does. Its unknowns are the multipliers κ of $A\beta = b$ and η of the active rows of $C\beta \le h$. It minimises one number τ, the largest violation of all the optimality conditions taken jointly, subject to $\eta \ge 0$. Each condition $\varphi_i(\kappa, \eta) \le 0$ enters as the row $\varphi_i(\kappa, \eta) \le \tau$. On a segment the conditions at both ends enter one program. The optimum is zero exactly when β is optimal. The violation is then re-evaluated exactly at the multipliers found. A small reported value therefore certifies optimality to that size, and a real violation cannot be masked, since no choice of multipliers brings it below the program's optimum. This is what the code has always done (`best_multipliers` in `python/make_figures.py`); the paper now says so.

R3 (required). Record the seeds and add a determinism attestation. Adopted (Code and data availability; README). Every random design is drawn inside `python/make_figures.py` from a seed recorded there. Seed 7 is used for Table 2, seeds 7–9 for Table 3, and seeds 7–16 for Tables 4 and 5. The paper now states that, on a fixed platform, the knot counts, the set of declined problems and the certificate outcomes reproduce exactly, and that only residuals near rounding vary, in their last digits. A new test, `test_deterministic`, runs the problems of Tables 3 and 4 twice and asserts that the discrete outputs are identical.

Recommended

[visualisation] Figure 3: colourblind-safe palette and direct labels. Adopted. The eight genera are drawn in the Okabe–Ito palette and each line is labelled at its right-hand end ($\lambda = 0$). The labels are spread apart only as far as needed to avoid overlap, so the series can be read without relying on hue.

[visualisation] State the axis transform of Figures 1 and 3. Adopted. Both captions, and the axis label of Figure 3, now say that the horizontal axis is λ decreasing to the right on a symmetric-log scale. It is logarithmic above λ_max/50 and linear below, so that the end of the path at $\lambda = 0$ is shown.

[software-landscape] One feature table. Adopted as Table 6 (Section 6), placed there so that Tables 2–5 keep their numbers. It lists, for `lars_path`, `c-lasso`, comlasso, Gaines et al., CLASSO and `cvxcla`, whether each accepts caps, per-coefficient signs, equality rows, inequality rows, zero sums and $p > n$, and whether it does so for $p > n$ without a ridge. CLASSO's column is taken from its paper, since its code is not released.

R1.1. The diabetes example: an illustration, u chosen post hoc, incident diabetes versus progression. Adopted (Section 1). The example is now introduced as an illustration of what a cap does to offsetting coefficients, "not offered as a better model of the disease". The text says that $u = 396.1$ is chosen after

seeing the fit. It separates the two claims: [20] supports low HDL as a risk factor for *incident* diabetes, while the response here is *progression*, so that finding "suggests a negative sign but does not establish one". The robustness to the cap is described as "a property of these data, not a finding about HDL".

R1.2. Runtime against comlasso, Gaines et al. and c-lasso. Answered with a reason (Section 6), not with timings. None of the three runs here as its authors would run it today. comlasso is R code built for R 3.2 and read under R 4. The algorithm of Gaines et al. is our port of Julia 0.6 code. `c-lasso` returns its own grid with a fixed ridge. A timing across R, Julia and Python would measure the languages and our ports as much as the algorithms. The text says so, and points a user to Table 6 to choose by the constraints each tool accepts, then to time candidates on their own problem.

R1.3. "Sum to one" versus synthetic control. Adopted (Section 5.1). The text now says that, unlike the classic synthetic control estimator (Abadie, Diamond and Hainmueller, 2010, now cited), which also requires nonnegative weights, the weights here may be negative, and that adding the signs would be one more bound.

R1.4. sCD14 and the other real-data paths are computational demonstrations. Adopted, where sCD14 is introduced (Section 5.1) and in the caption of Figure 3: "a computational demonstration on real data. No biological inference is intended, and none should be drawn from its path."

R2.1. Name the test whose path fails the segment check, and what the check returns. Adopted (Section 5.1). It is `test_segment_check_sees_between_samples`. With $p = 1$, $y = 1$ and the cap $\beta \le ½$, the exact path has knots at $\lambda = 1$ and $\lambda = ½$. The single segment from $(\lambda, \beta) = (5/4, 0)$ to $(1/4, ½)$ misses both knots, yet meets the path at its two ends and its midpoint. There the optimality conditions hold exactly (largest violation 0), while the segment check returns 0.25, relative to $\|c\|_\infty$. Both numbers are now written into the text by the script.

R2.2. Tables 3 and 4 over more than three seeds, keeping the worst case. Adopted. One correction first: Table 3 was drawn from a single seed, not three. Table 3 now uses three seeds per size. Tables 4 and 5 now use ten (seeds 7–16). Knot counts are reported as ranges and every other column as the worst case.

The extra seeds did what the reviewer intended and moved the boundary between "traced exactly" and "declined":

- *Duplicate 10^{-6} with the cap* is now declined on 2 of the 10 draws. The cause is the same as without the cap: on those draws both near-duplicate columns become free before the cap binds, and the free block is singular to working precision. The earlier sentence "under the cap they never are" was true of three seeds only and has been replaced. Every path that is traced still meets the optimality conditions to rounding.
- *Correlated* failed on one draw (seed 14). This was not a principled decline but defects in `cvxcla` 2.7.0, which the author maintains. Two were in the check that a bound held along a segment is still optimal. With no active rows its feasibility LP had no variables and raised an error, and its noise floor ignored the benchmark term q, so round-off at $\lambda = 0$, where Σw and q cancel, read as a wrongly held bound. Testing the fix on other platforms exposed a third, in the plain LASSO homotopy, which is not used in the paper: when two events coincide, as when nearly collinear columns swap, it lost the second. All three are fixed in `cvxcla` 2.7.1, which the revision uses, each with a regression test. On seed 14 *Correlated* is now traced and meets the optimality conditions to rounding, and over the ten seeds it reports 43–49 knots and a residual of at most 4.1×10^{-13}. Section 1 reports the three defects as it reports the two fixed in 2.6.0.
- On *Wide* ($p > n$) the worst KKT residual over ten seeds is 1.0×10^{-9}, against 2.7×10^{-13} over three. It occurs on one draw of *Zero-sum, signs*, at the very end of the path ($\lambda \approx 10^{-4}$ λ_max), where the fit

almost interpolates and cond(X) ≈ 6×10^{15}. Everywhere else the residuals stay at rounding, and the interior-point objective is never better than the path's by more than 7.5×10^{-12}, relative. The abstract and Section 5.5 now say this, and no longer claim rounding-level residuals there.

R2.3. The values of ε, and that the certificate guarantees the path. Adopted (Section 4). In `cvxcla` 2.7.1, ε is 10^{-6} times the coefficient scale $\|c\|_\infty$ / $\max_j G_{jj}$, then 10^{-7} and 10^{-8} times it if the certificate fails. A path is accepted at 10^{-9} relative to $\|c\|_\infty$. The text adds: "The extrapolation only proposes a path; the certificate is what guarantees the path returned."

R2.4. The gap against LARS where the minimiser is not unique, and the five Gaines problems. Adopted (Sections 5.4 and 5.6). Where the minimiser is not unique, LARS and the CLA may trace different paths of minimisers. A gap between them is no defect as long as both meet the optimality conditions, and the certificate, not the agreement, is what each path is held to. The five Table 2 problems on which the ported Gaines algorithm, started at the exact first knot, matches the CLA are *Plain*, *Box*, *Cholesterol*, *Zero-sum* and *sCD14*. The Julia check now writes these names into the text.

R3.1. Which R runs comlasso 0.2, and how a package built for R 3 is loaded. Adopted (Code and data availability). R 4.6.1 runs the check, locally and in CI. The package is not installed, since R 4 refuses packages built before R 4.0. The script fetches the zip from the authors' repository at a pinned commit, checks its SHA-256, and reads the functions from the package's serialized code with `lazyLoad`, unchanged. comlasso ships no compiled code, so nothing in it depends on the R 3 build. Its one dependency, Rglpk 0.6.5.1, comes from a dated CRAN snapshot.

R3.2. Timing methodology. Adopted (Section 5.3; README). Each step is now run once to warm up and then timed 11 times (3 times for the p = 500 grid, which takes minutes). The paper quotes the ratio of the medians together with the range of the ratio over paired runs, instead of the fastest of several runs. The ratios barely move: about 90× the path at p = 100, 340× at p = 500, and 14–110× on the constrained problems of Table 2, with every paired run within 8% of the median ratio.

R3.3. Fallback for the sCD14 source. Adopted. `python/fetch_scd14.py` tries CRAN's `src/contrib`, then the CRAN archive, then the Posit Package Manager snapshot of 2026-09-01. Posit repacks the tarball, so each source's tarball digest is pinned separately. The script also checks the SHA-256 of `data/sCD14.rda` inside the tarball, which we verified to be byte-identical across CRAN and Posit. The data that reach the analysis are therefore pinned whichever source serves them.

Optional

[statistical-rigor] More seeds to characterise the gap as a distribution. Partly taken up through R2.2 (ten seeds). Characterising the dependence of the gap on conditioning as a distribution is left, as the editor suggests, to later work.

R1.5. A predictive comparison (cross-validation). Declined as beyond the scope of a verification note, as the reviewer also says. The paper claims nothing about predictive performance, and the revised framing of the diabetes and sCD14 examples (R1.1, R1.4) removes any suggestion that it does.