SUPPLEMENTARY MATERIALS: AN EFFICIENT ALGORITHM FOR INTEGER LATTICE REDUCTION[*]

FRANÇOIS CHARTON[†], KRISTIN LAUTER[‡], CATHY LI[§], AND MARK TYGERT[¶]

SM1. Poorly performing alternatives. This supplementary section mentions two possible modifications that lack the firm theoretical grounding of the algorithm presented in the main text and performed rather poorly in numerical experiments. These modifications may be natural, yet seem not to work well. Subsection SM1.1 considers adding to a basis vector multiple other basis vectors simultaneously, such that the full linear combination would minimize the Euclidean norm if the coefficients in the linear combination did not have to be rounded to the nearest integers. Subsection SM1.2 considers a modified Gram-Schmidt procedure.

SM1.1. Combining multiple vectors simultaneously. One possibility is to choose a basis vector at random, say a_j^i, and add to that vector the linear combination of all other basis vectors which minimizes the Euclidean norm of the result, with the coefficients in the linear combination rounded to the nearest integers. That is, choose an index j uniformly at random, and calculate real-valued coefficients $c_{j,k}^i$ such that the Euclidean norm $\|a_j^i - \sum_{k=1}^n c_{j,k}^i \cdot a_k^i\|$ is minimal, where $c_{j,j}^i = 0$. Then, construct $a_j^{i+1} = a_j^i - \sum_{k=1}^n \operatorname{nint}(c_{j,k}^i) \cdot a_k^i$.

Repeating the process for multiple iterations, $i = 0, 1, 2, \dots$, would appear reasonable. However, this scheme worked well empirically only when the number n of basis vectors was very small, at least when the number m of entries in each of the basis vectors was equal to n. Rounding the coefficients $c_{j,k}^i$ to the nearest integers is too harsh for this process to work well.

SM1.2. Modified Gram-Schmidt process. Another possibility is to run the classical Gram-Schmidt procedure on the basis vectors, while subtracting off from all vectors not yet added to the orthogonal basis the projections onto the current pivot vector. In this modified Gram-Schmidt scheme, each iteration chooses as the next pivot vector the residual basis vector for which adding that basis vector to the orthogonal basis would minimize the sum of the p-th powers of the Euclidean norms of the reduced basis vectors. The iteration orthogonalizes the pivot vector against all previously chosen pivot vectors and then subtracts off (from all residual basis vectors not yet chosen as pivots) the projection onto the orthogonalized pivot vector, with the coefficients in the projections rounded to the nearest integers.

This scheme strongly resembles the LLL algorithm of [SM2], but with a different pivoting strategy (using modified Gram-Schmidt). Numerical experiments indicate that the modified Gram-Schmidt performs somewhat similarly to yet significantly worse than the classical LLL algorithm. Omitting the bubble-sorting of the LLL algorithm via the so-called "Lovász criterion" spoils the scheme.

This scheme is also reminiscent of the variants of the LLL algorithm with so-called "deep insertions," as developed by [SM4], [SM1], [SM5], and others. LLL with deep

[*] Submitted to the editors DATE.

Funding: This work was funded by Meta Platforms, Inc.

[†] Meta Platforms, Inc., 6 Rue Ménars 75002, Paris, France (fcharton@meta.com).

[‡] Meta Platforms, Inc., 1101 Dexter Ave N, Seattle, WA (klauter@meta.com).

[§] Meta Platforms, Inc., 1101 Dexter Ave N, Seattle, WA (cathyli@meta.com).

[¶] Meta Platforms, Inc., 1 Facebook Way, Menlo Park, CA (tygert@meta.com).

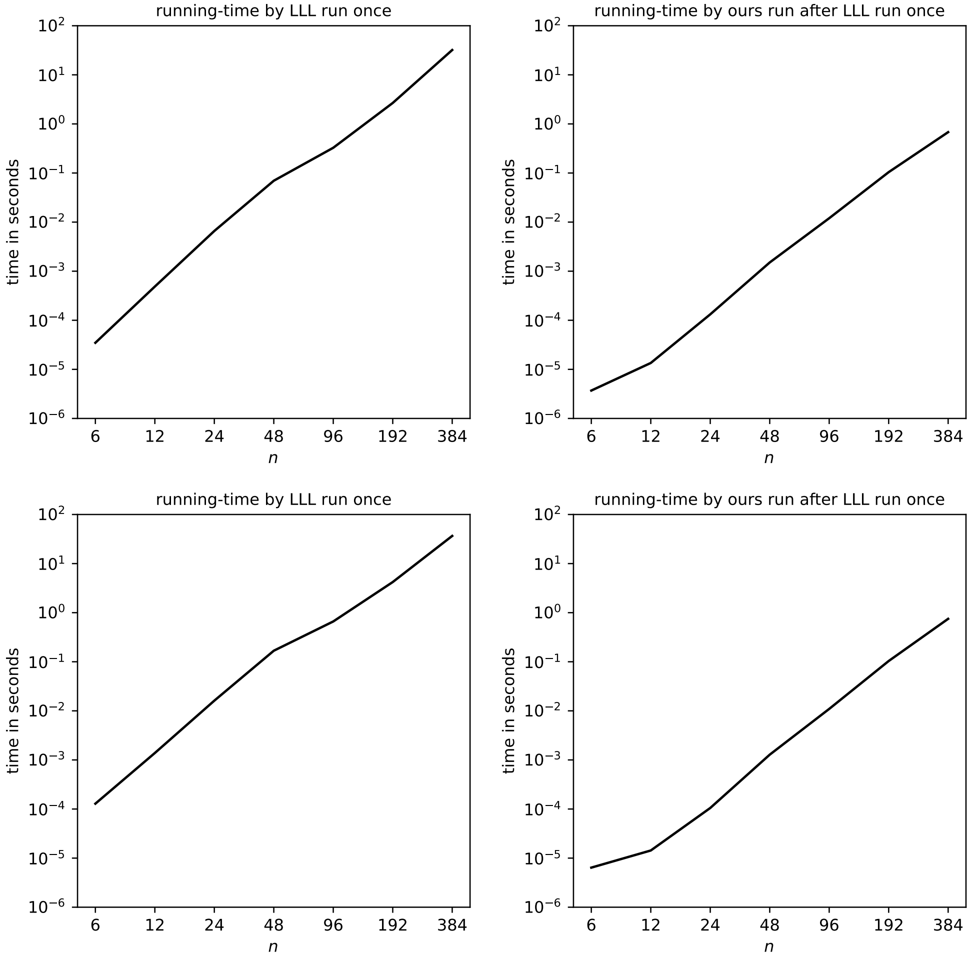

FIG. SM1. *$\delta = 1 - 10^{-1}$, $p = 2$; the upper plots are for $q = 2^{13} - 1$, the lower plots are for $q = 2^{31} - 1$*

insertions performs much better, however, both theoretically and practically. Other modifications to the LLL algorithm, notably the BKZ and BKW methods reviewed by [SM3] and others, also perform much better than the modified Gram-Schmidt.

SM2. Further figures. This supplementary section presents figures analogous to those of Section 3, but using different values of the parameters δ and p detailed in Subsection 3.1. Figures SM1–SM5 are the same as Figures 1–5, but with $\delta = 1 - 10^{-1}$ instead of $\delta = 1 - 10^{-15}$. Figures SM6–SM15 are the same as Figures 1–5 and Figures SM1–SM5 for $n = 192$, but with varying values of p rather than just $p = 2$.

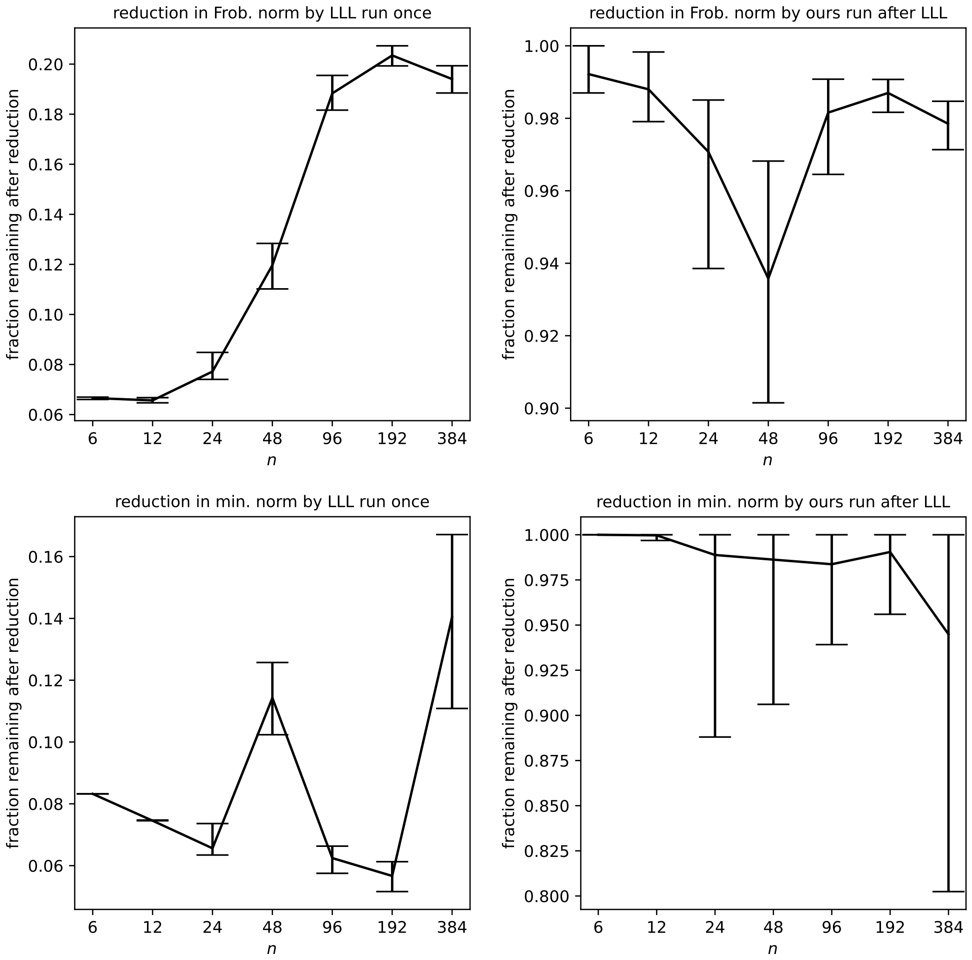

FIG. SM2. $\delta = 1 - 10^{-1}$, $p = 2$, $q = 2^{13} - 1$

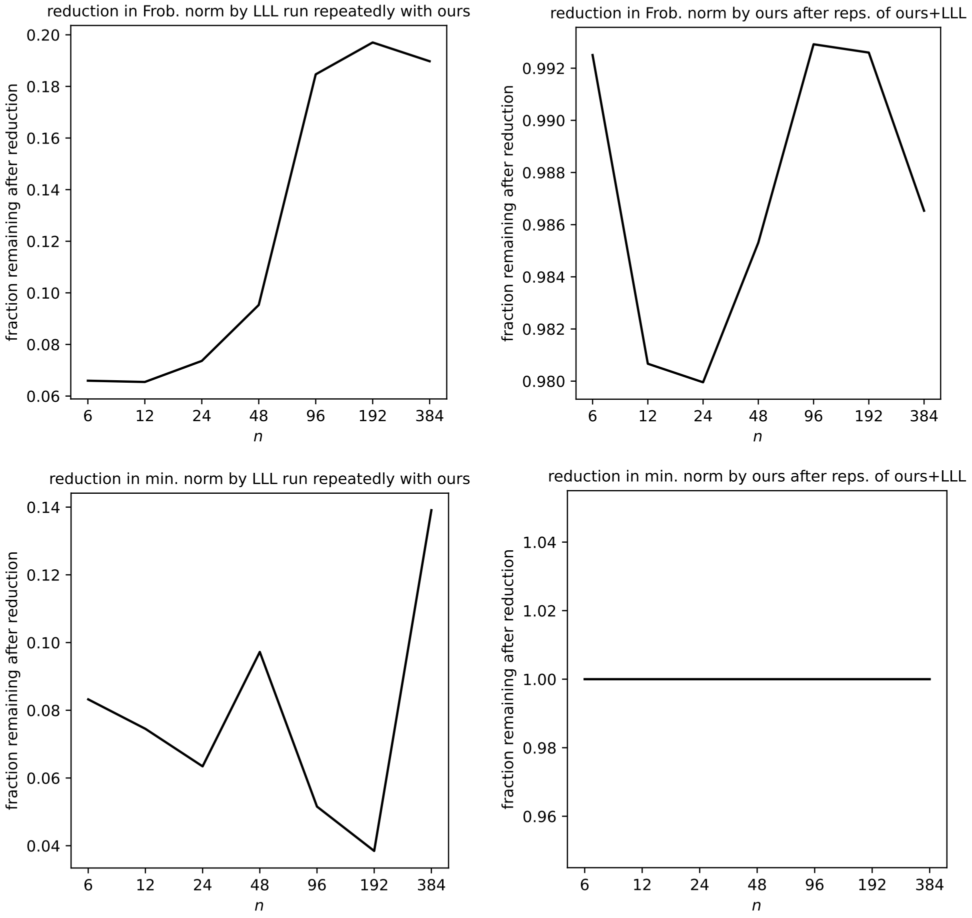

FIG. SM3. $\delta = 1 - 10^{-1}$, $p = 2$, $q = 2^{13} - 1$

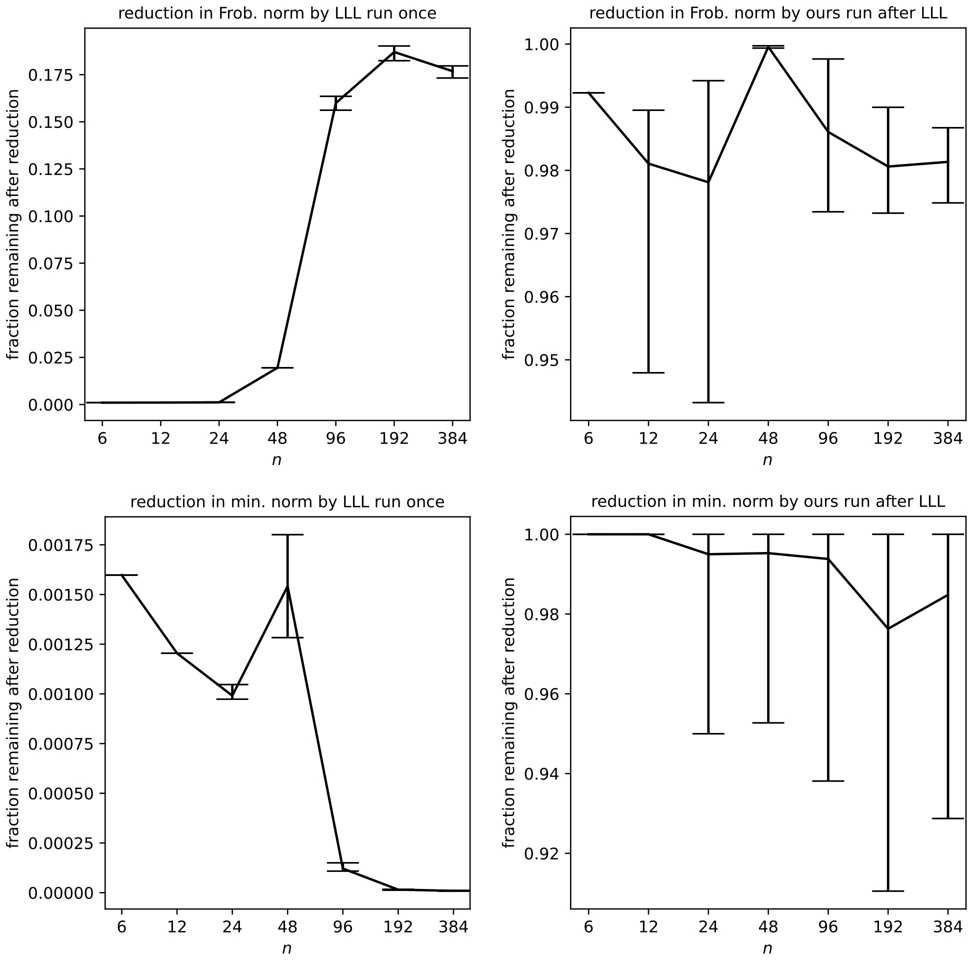

Fig. SM4. $\delta = 1 - 10^{-1}$, $p = 2$, $q = 2^{31} - 1$

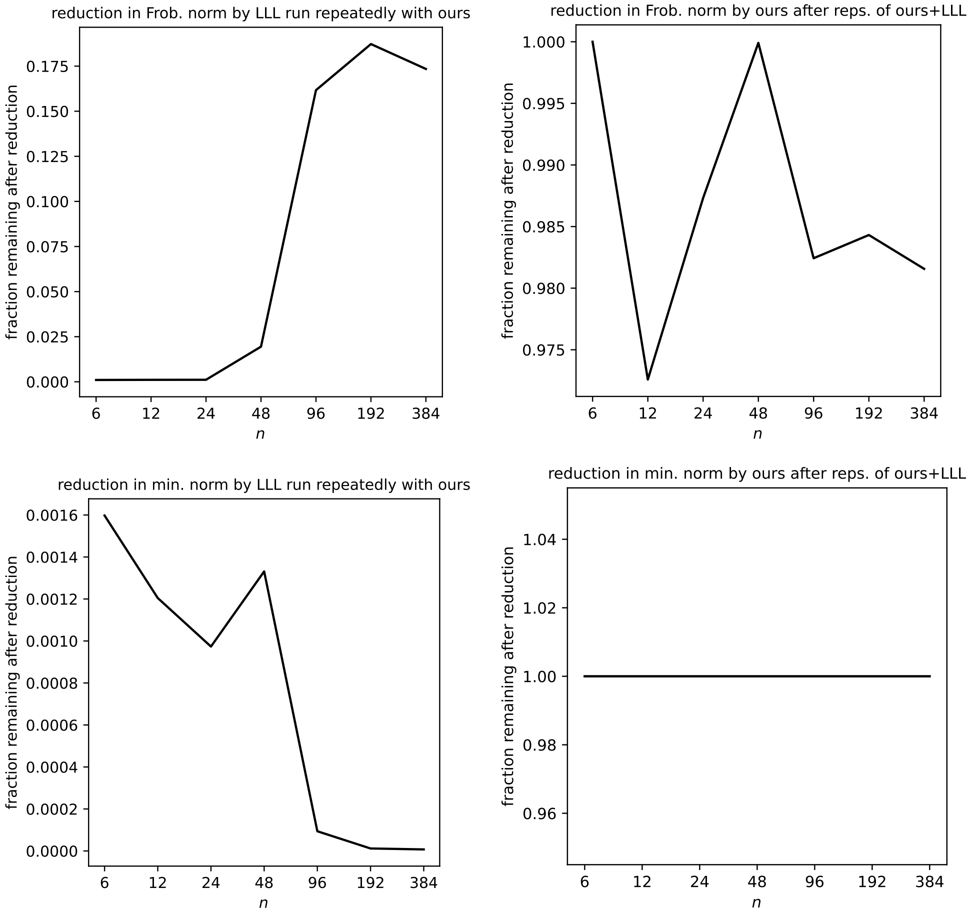

Fig. SM5. $\delta = 1 - 10^{-1}$, $p = 2$, $q = 2^{31} - 1$

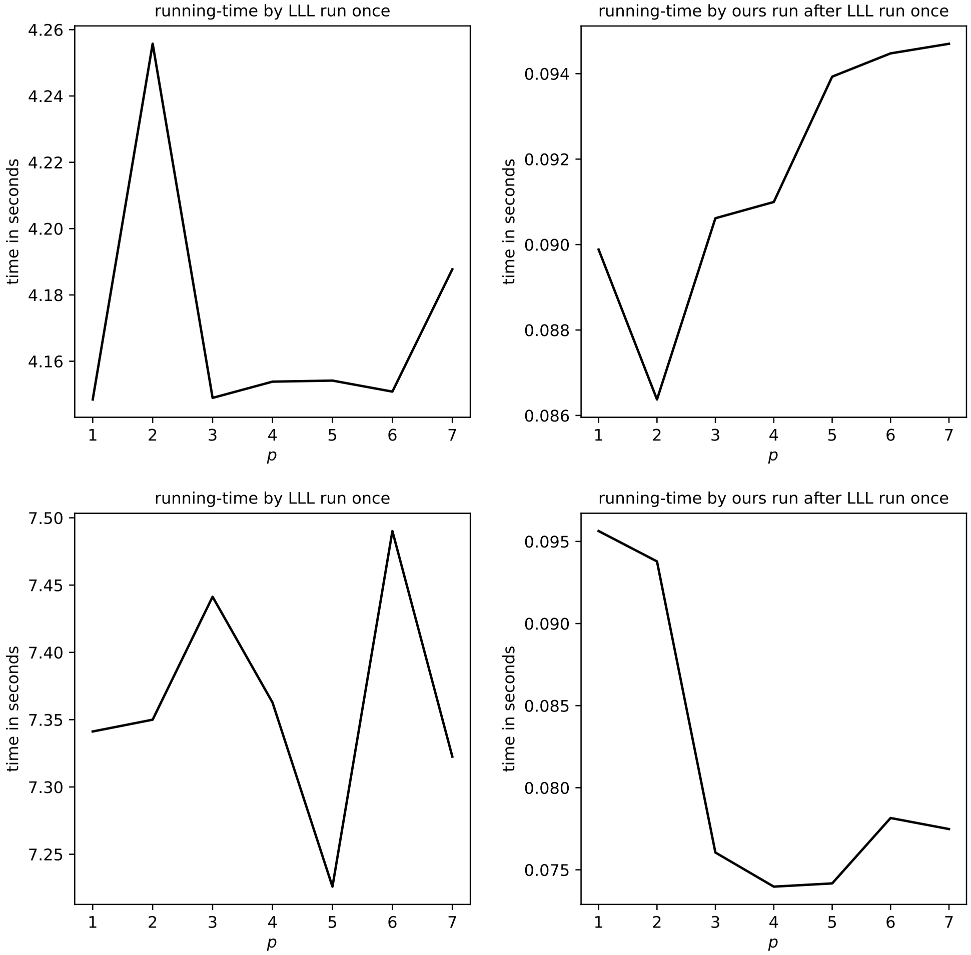

FIG. SM6. $\delta = 1 - 10^{-15}$, $n = 192$; *the upper plots are for* $q = 2^{13} - 1$, *the lower plots are for* $q = 2^{31} - 1$ . . . *the vertical ranges of the plots on the left are very small, with the vertical variations displayed being statistically insignificant, wholly attributable to randomness in the computational environment.*

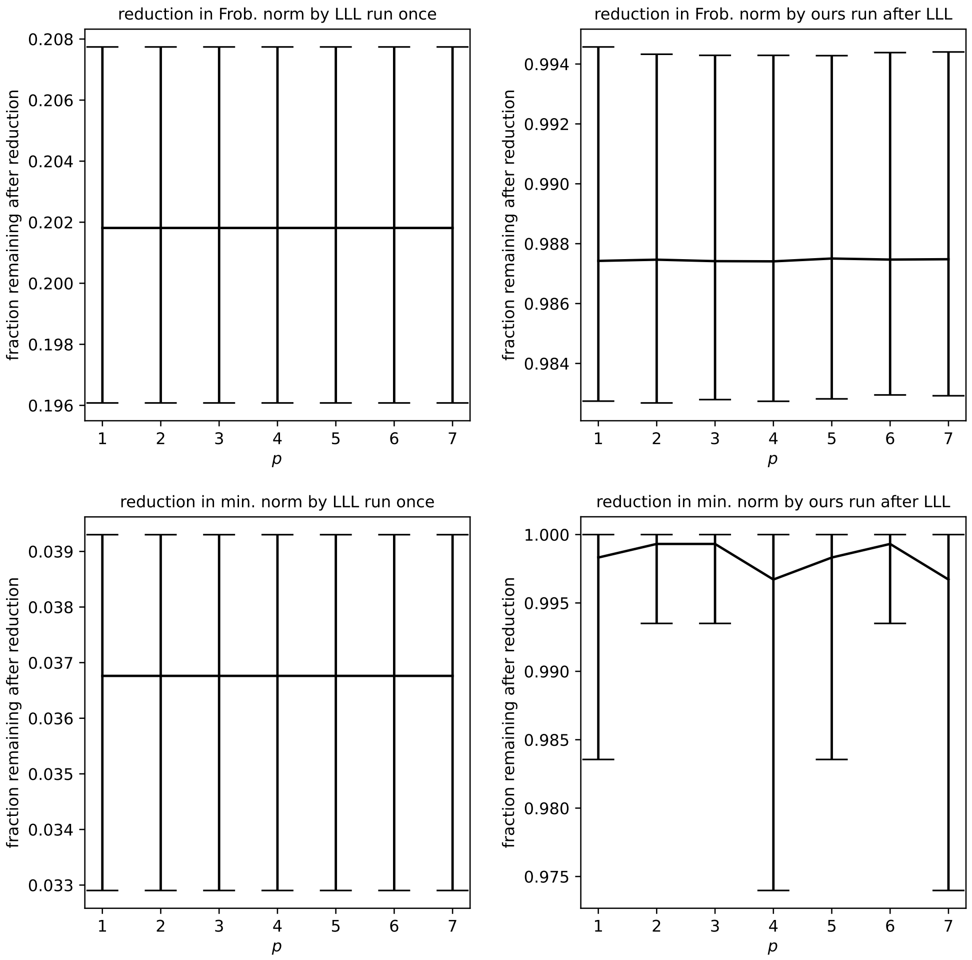

FIG. SM7. $\delta = 1 - 10^{-15}$, $n = 192$, $q = 2^{13} - 1$

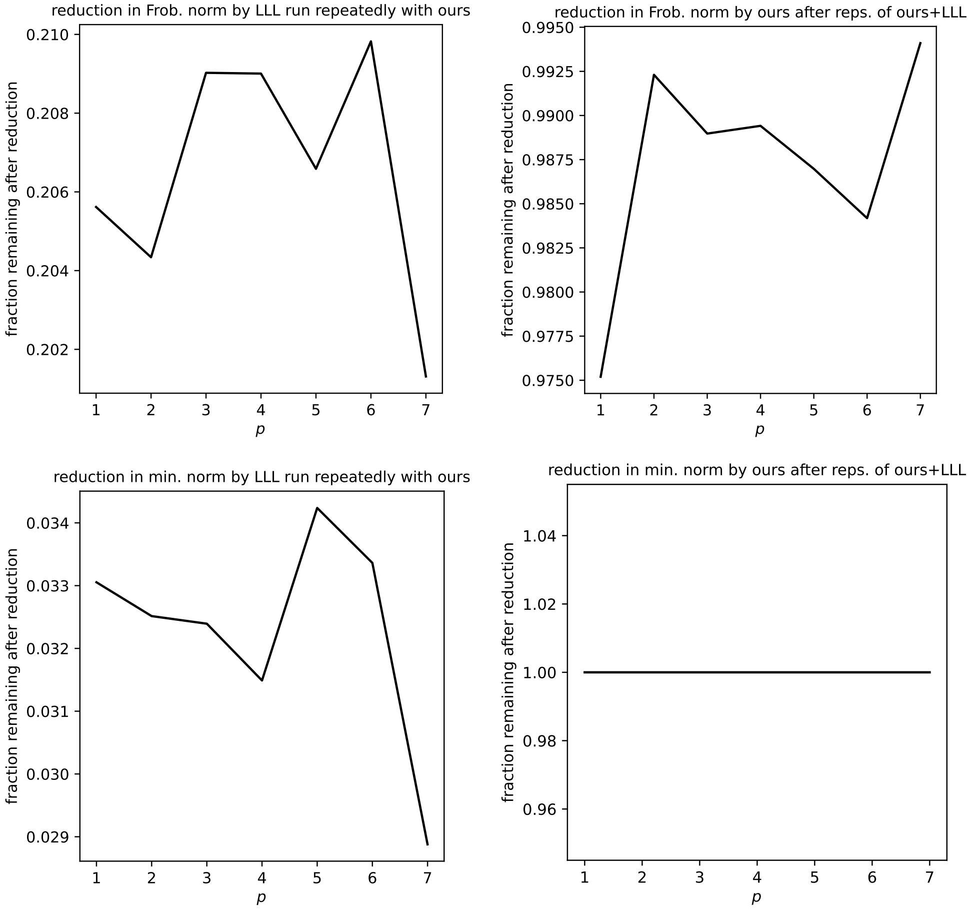

FIG. SM8. $\delta = 1 - 10^{-15}$, $n = 192$, $q = 2^{13} - 1$

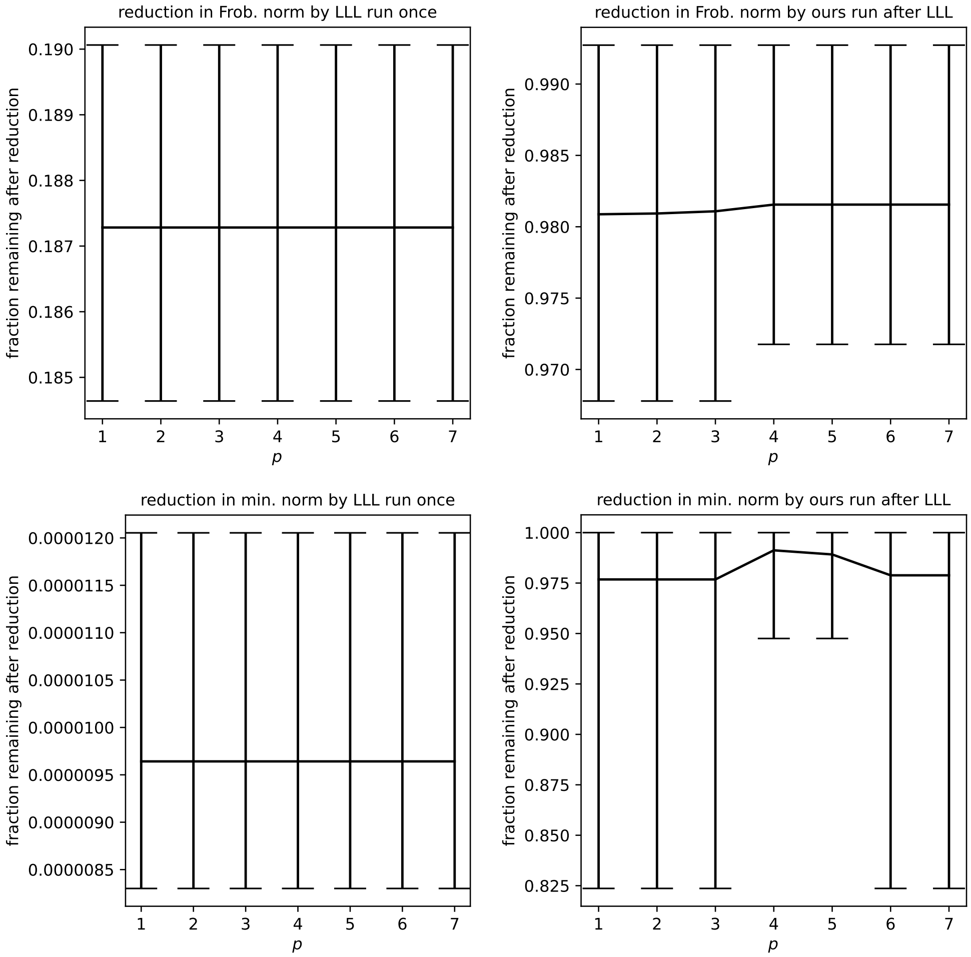

FIG. SM9. $\delta = 1 - 10^{-15}$, $n = 192$, $q = 2^{31} - 1$

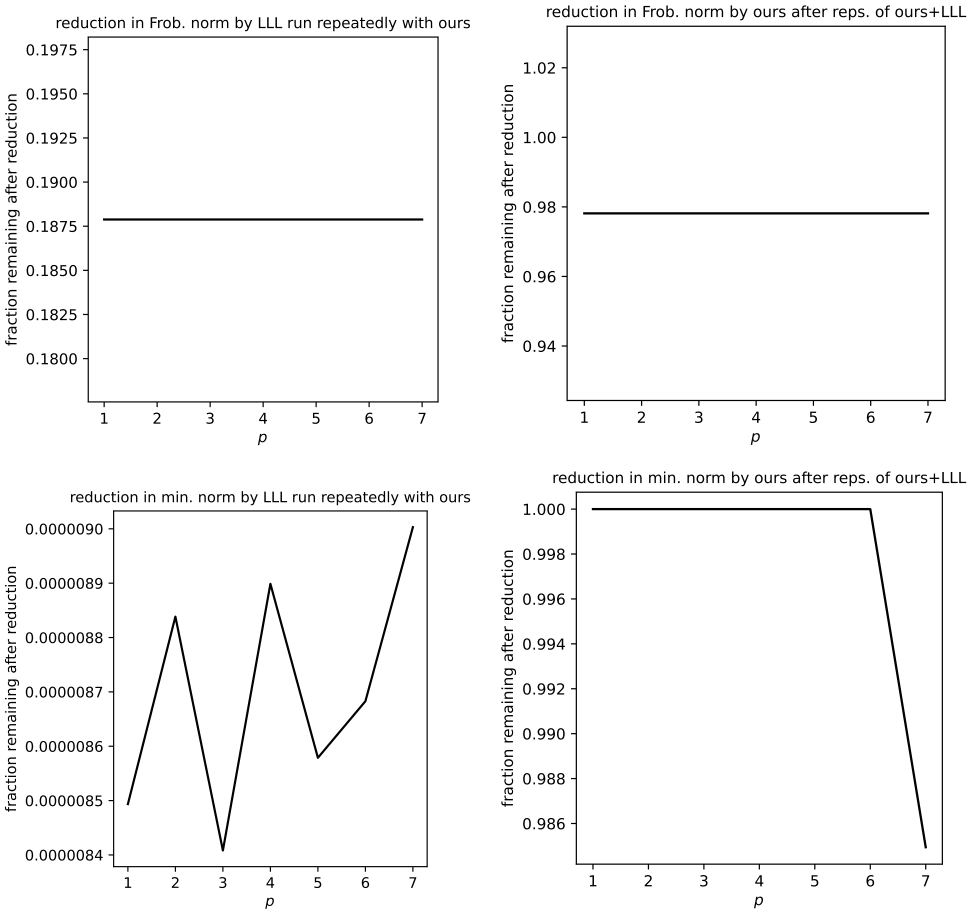

FIG. SM10. $\delta = 1 - 10^{-15}$, $n = 192$, $q = 2^{31} - 1$

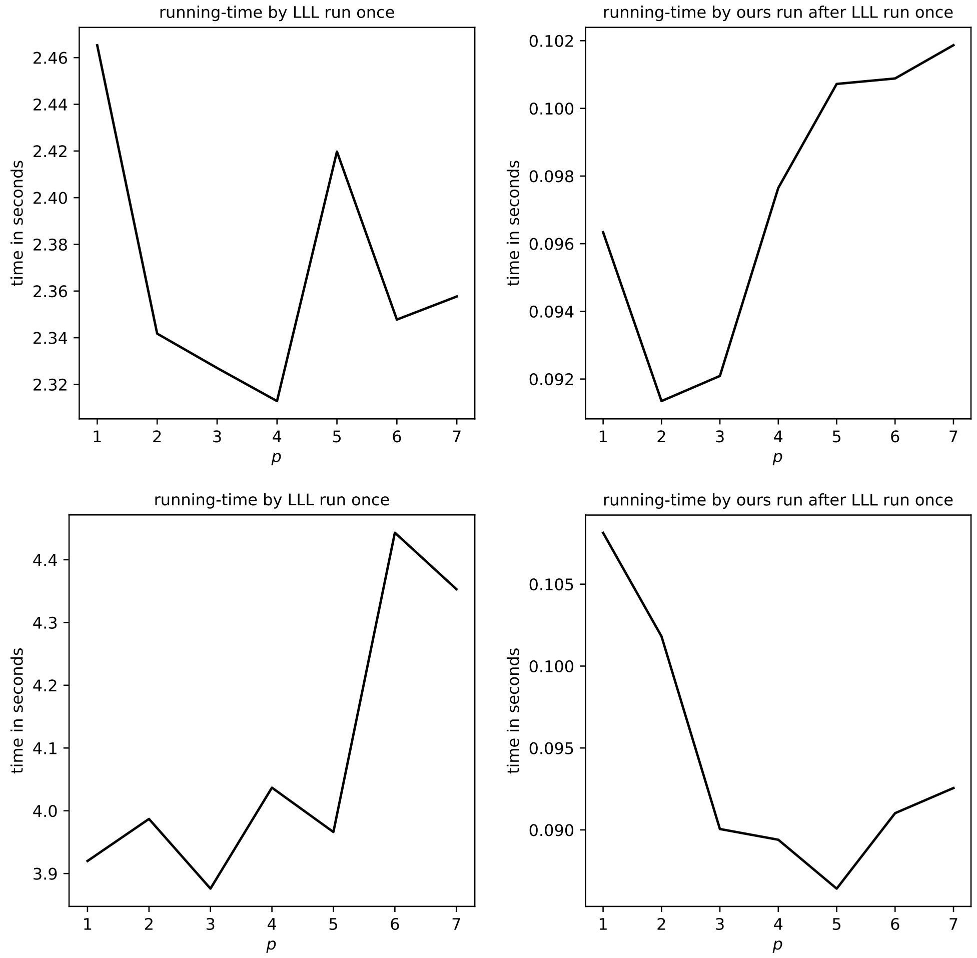

FIG. SM11. $\delta = 1 - 10^{-1}$, $n = 192$*; the upper plots are for* $q = 2^{13} - 1$*, the lower plots are for* $q = 2^{31} - 1$ *. . . the vertical ranges of the plots on the left are very small, with the vertical variations displayed being statistically insignificant, wholly attributable to randomness in the computational environment.*

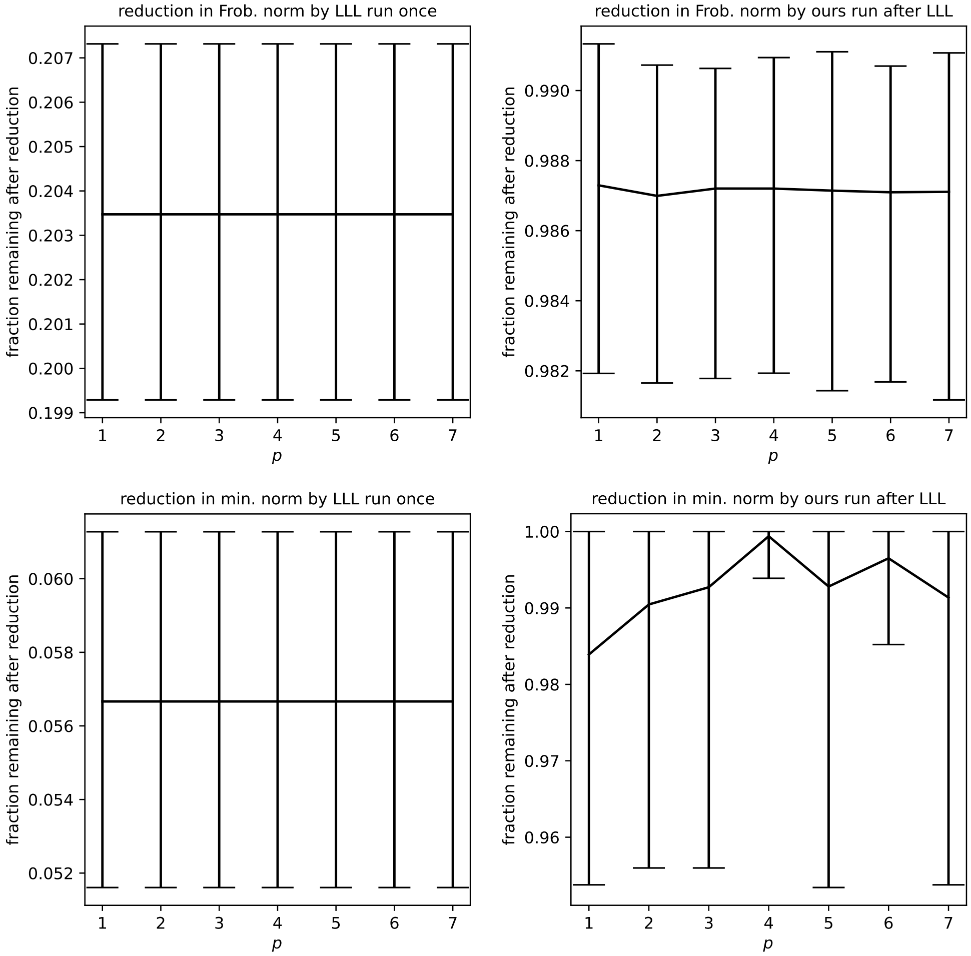

Fig. SM12. $\delta = 1 - 10^{-1}$, $n = 192$, $q = 2^{13} - 1$

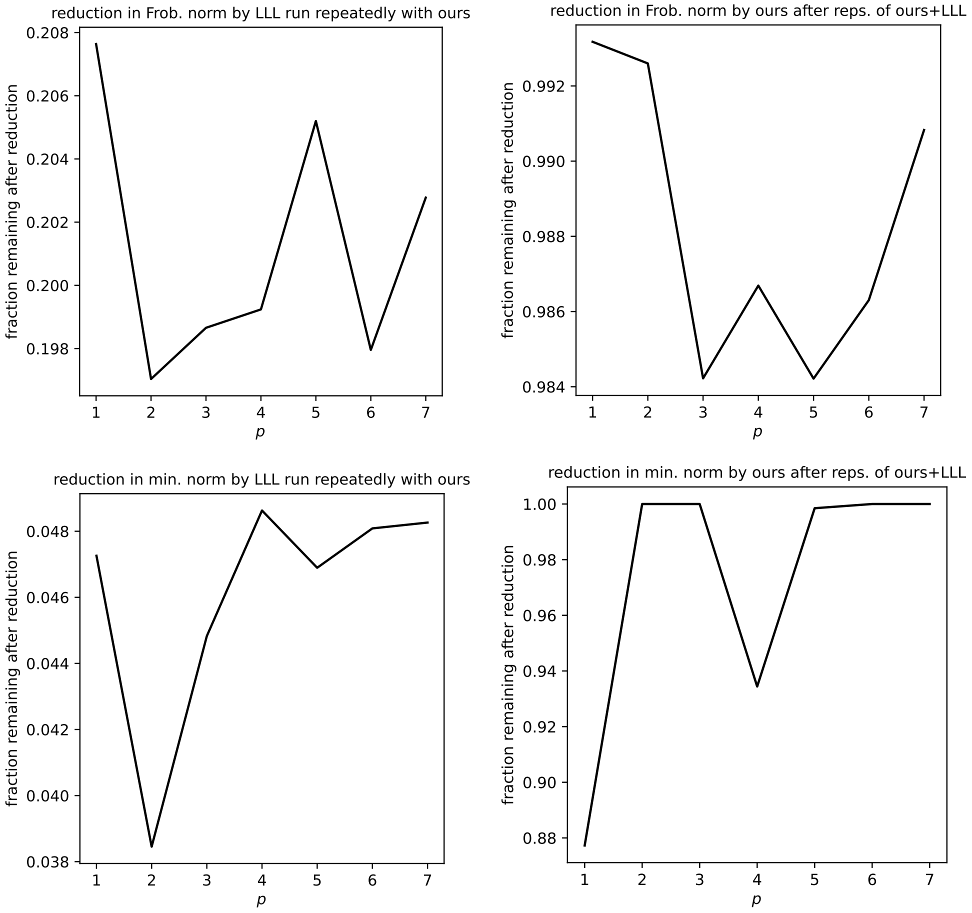

FIG. SM13. $\delta = 1 - 10^{-1}$, $n = 192$, $q = 2^{13} - 1$

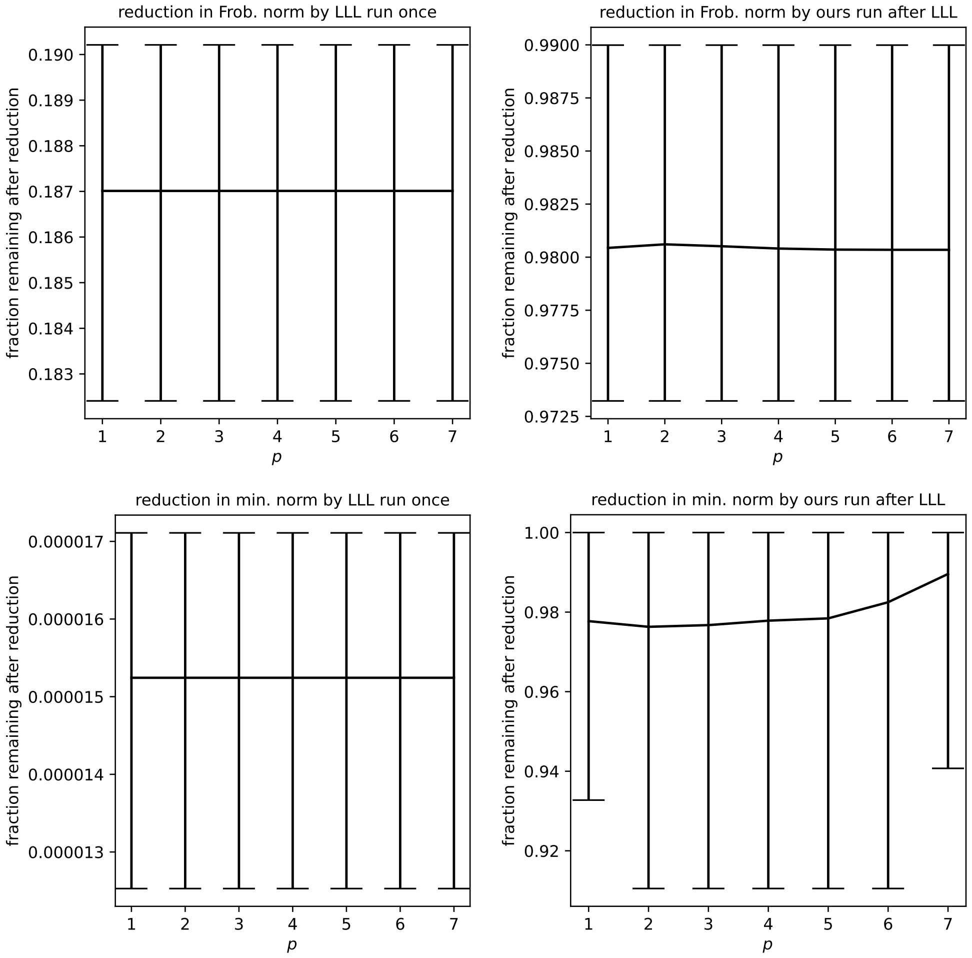

FIG. SM14. $\delta = 1 - 10^{-1}$, $n = 192$, $q = 2^{31} - 1$

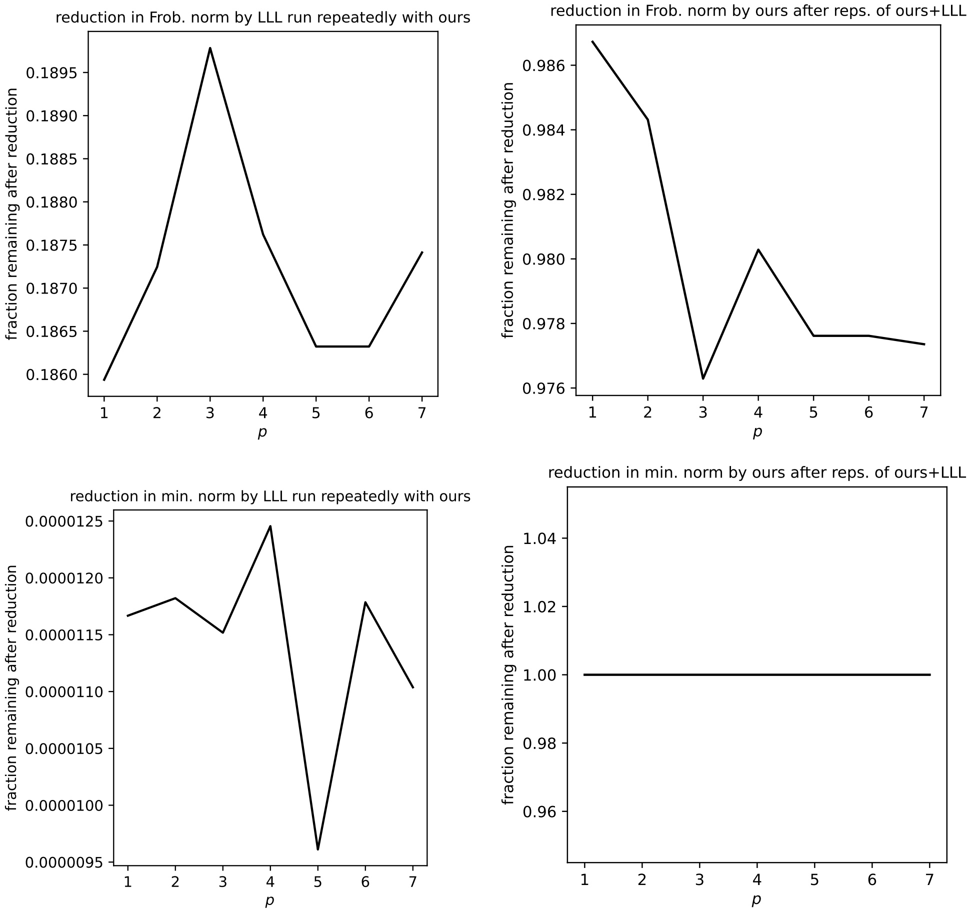

FIG. SM15. $\delta = 1 - 10^{-1}$, $n = 192$, $q = 2^{31} - 1$

REFERENCES

[SM1] F. Fontein, M. Schneider, and U. Wagner, *PotLLL: a polynomial time version of LLL with deep insertions*, Des. Codes Cryptogr., 73 (2014), pp. 355–368.

[SM2] A. K. Lenstra, H. W. Lenstra, and L. Lovász, *Factoring polynomials with rational coefficients*, Math. Ann., 261 (1982), pp. 515–534.

[SM3] P. Q. Nguyen and B. Vallée, eds., *The LLL Algorithm: Survey and Applications*, Information Security and Cryptography, Springer, 2010.

[SM4] C. P. Schnorr and M. Euchner, *Lattice basis reduction: improved practical algorithms and solving subset sum problems*, Math. Program., 66 (1994), pp. 181–199.

[SM5] M. Yasuda and J. Yamaguchi, *A new polynomial-time variant of LLL with deep insertions for decreasing the squared-sum of Gram-Schmidt lengths*, Des. Codes Cryptogr., 87 (2019), pp. 2489–2505.