Article

Not peer-reviewed version

A Novel Grouped-Gram-Based Algorithm for Fast and Memory-Efficient Fixed Effects Estimation

Felix Reichel *

Posted Date: 9 December 2025

doi: 10.20944/preprints202512.0713.v1

Keywords: fixed effects; panel data; grouped Gram matrix; within transformation; computational econometrics; memory-efficient algorithms; high-dimensional estimation

Article

A Novel Grouped-Gram-Based Algorithm for Fast and Memory-Efficient Fixed Effects Estimation

Felix Reichel

Department of Economics, Johannes Kepler University Linz, 4040 Linz, Austria; felix.reichel@jku.at

Abstract

Fixed effects models often rely on the within transformation, which constructs demeaned arrays prior to forming cross-products. This paper develops an estimator that avoids the formation of demeaned arrays by exploiting grouped summaries built from per-unit sufficient statistics. A complete derivation shows that the grouped Gram representation reproduces the classical estimator exactly. The difference lies in memory access patterns and byte movement. The grouped estimator concentrates operations into unit-level accumulations, avoiding the writes associated with array centering. Gains arise once the panel reaches a scale where memory traffic governs run time. Simulations examine coefficient accuracy, bootstrap dispersion, run time, and memory use.

Keywords: fixed effects; panel data; grouped Gram matrix; within transformation; computational econometrics; memory-efficient algorithms; high-dimensional estimation

MSC (2020): 62J20; 62P20; 68W40

JEL Codes: C33; C63; C55

---

1. Introduction

Fixed effects models appear widely in empirical research. The standard approach applies the within transformation: subtract unit means from each observation and form cross-products from the transformed arrays. This is direct and compatible with linear algebra routines, yet the method reshapes the data and performs multiple passes across memory. These operations become costly once the panel grows large relative to cache or main memory bandwidth.

Grouped Gram representations avoid array centering by replacing the transformation with sufficient statistics. These statistics summarize each unit's contribution with sums and cross-products. The result is mathematically identical to the classical estimator. The central distinction lies in the structure of the computation. Grouped summaries allow a single pass through the data while maintaining numerical precision.

The study draws on several literatures. Panel methods in econometrics provide the model foundations (Arellano, 2003; Hausman, 1978; Hsiao, 2014; Wooldridge, 2010). Matrix computation references outline the algebra of Gram matrices (Chan & Golub, 1983; Gentle, 2007; Golub & Van Loan, 2013). Work on cache behavior and communication bounds frames the memory analysis (Demmel et al., 2007; Graham et al., 2002; Hong & Kung, 1981; Williams et al., 2009). The bootstrap section relies on Cameron et al. (2008); Davison & Hinkley (1997).

2. Setup and Notation

Units are indexed by $i = 1, \dots, N$ and periods by $t = 1, \dots, T_i$. The model is

$$y_{it} = x_{it}'\beta + \alpha_i + \varepsilon_{it}, \tag{1}$$

with covariates $x_{it} \in \mathbb{R}^p$, unknown coefficient vector β, and unit fixed effects α_i.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 9 December 2025 doi:10.20944/preprints202512.0713.v1

Define

$$\bar{x}_i \stackrel{\text{def}}{=} \frac{1}{T_i}\sum_{t=1}^{T_i} x_{it}, \qquad \bar{y}_i \stackrel{\text{def}}{=} \frac{1}{T_i}\sum_{t=1}^{T_i} y_{it}.$$

The within-transformed values are

$$x^*_{it} = x_{it} - \bar{x}_i, \qquad y^*_{it} = y_{it} - \bar{y}_i.$$

The estimator is

$$\hat{\beta}_{\text{within}} = \left(\sum_{i=1}^{N}\sum_{t=1}^{T_i} x^*_{it}x^{*\prime}_{it}\right)^{-1}\left(\sum_{i=1}^{N}\sum_{t=1}^{T_i} x^*_{it}y^*_{it}\right). \tag{2}$$

3. Grouped Gram Representation

Let the grouped sums and cross-products be

$$s_{x,i} = \sum_{t=1}^{T_i} x_{it}, \qquad s_{y,i} = \sum_{t=1}^{T_i} y_{it},$$

$$G_i = \sum_{t=1}^{T_i} x_{it}x'_{it}, \qquad g_i = \sum_{t=1}^{T_i} x_{it}y_{it}.$$

3.1. Derivation of Gram Identities

Write

$$x^*_{it} = x_{it} - \frac{s_{x,i}}{T_i}.$$

Then

$$\sum_{t=1}^{T_i} x^*_{it}x^{*\prime}_{it} = \sum_{t=1}^{T_i}\left(x_{it} - \frac{s_{x,i}}{T_i}\right)\left(x_{it} - \frac{s_{x,i}}{T_i}\right)' \tag{3}$$

$$= \sum_{t=1}^{T_i} x_{it}x'_{it} - \frac{s_{x,i}s'_{x,i}}{T_i} - \frac{s_{x,i}s'_{x,i}}{T_i} + \frac{T_i s_{x,i}s'_{x,i}}{T_i^2}. \tag{4}$$

Since

$$\frac{T_i s_{x,i}s'_{x,i}}{T_i^2} = \frac{s_{x,i}s'_{x,i}}{T_i},$$

the terms combine to

$$\sum_{t=1}^{T_i} x^*_{it}x^{*\prime}_{it} \equiv G_i - \frac{s_{x,i}s'_{x,i}}{T_i}. \tag{5}$$

The same logic applies to the cross-product:

$$\sum_{t=1}^{T_i} x^*_{it}y^*_{it} \equiv g_i - \frac{s_{x,i}s_{y,i}}{T_i}.$$

3.2. Estimator Based on Grouped Summaries

Summing across units yields the estimator

$$\hat{\beta}_{\text{fast}} = \left(\sum_{i=1}^{N}\left[G_i - \frac{s_{x,i}s'_{x,i}}{T_i}\right]\right)^{-1}\left(\sum_{i=1}^{N}\left[g_i - \frac{s_{x,i}s_{y,i}}{T_i}\right]\right), \tag{6}$$

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 9 December 2025 doi:10.20944/preprints202512.0713.v1

which equals the within estimator in exact arithmetic.

4. Memory Path Differences

The within transformation forms x^*_{it} and y^*_{it} explicitly. Each assignment causes reads and writes across arrays of size $n = \sum_i T_i$. Since modern processors face bandwidth limitations (Graham et al., 2002; McCalpin, 1995), these memory operations dominate the floating point operations once n is large.

The grouped estimator performs a single pass through the data. For each i, the algorithm accumulates $s_{x,i}$, $s_{y,i}$, G_i, and g_i. The cross-product corrections rely on these summaries alone. This reduces memory traffic by avoiding write-back of centered arrays.

The roofline model (Williams et al., 2009) frames the method: algorithms with high byte movement relative to floating point work fall below peak performance. Since the fast estimator avoids array writes, its arithmetic intensity improves.

Communication bounds (Demmel et al., 2007; Hong & Kung, 1981) support this point. The within transformation moves $O(np)$ numbers, while the grouped approach touches each x_{it} and y_{it} only once.

5. Numerical Considerations

Floating point paths differ due to summation order. Pairwise or block summation can reduce error (Chan & Golub, 1983). The grouped estimator follows a block structure aligned with units. Since block sizes T_i are often moderate, the structure helps numerical stability.

Computing the Gram matrix from demeaned arrays produces the same algebraic result but rearranges operations. Both methods reach the same minimizer.

6. Bootstrap Inference

Units form natural clusters. Cluster bootstrap draws unit indices with replacement and reconstructs the panel. The bootstrap estimator uses either (2) or (6). Since the grouped approach mirrors the original estimator's structure, bootstrap dispersion remains stable across methods. The bootstrap references follow Cameron et al. (2008) and Davison & Hinkley (1997).

7. Asymptotic Theory

The algebraic identities in (2)–(6) imply that both estimators share the same probability limit and asymptotic distribution. The statements below follow the standard fixed-effects framework, and cover two regimes.

Let $u_{it} = \varepsilon_{it} - \bar{\varepsilon}_i$ denote the transformed error. Define

$$\Omega_i \stackrel{\text{def}}{=} \mathbb{V}\left(\sum_{t=1}^{T_i} x^*_{it} u_{it}\right).$$

7.1. Panel Regime 1: Large N, Fixed T

We impose the usual regularity conditions for fixed-effects panels.

(i) $\{(x_{it}, u_{it}) : t \le T\}$ is independent across i and weakly dependent within units.
(ii) $\mathbb{E}[u_{it} \mid x_{i1}, \ldots, x_{iT}] = 0$.
(iii) $\frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T} x^*_{it} x^{*\prime}_{it} \to Q$, a positive definite matrix.
(iv) Fourth moments exist uniformly.

Theorem 1. *Under assumptions (i)–(iv) and fixed* T,

$$\sqrt{N}\,(\hat{\beta}_{fast} - \beta) \xrightarrow{d} \mathcal{N}(0,\, Q^{-1} W Q^{-1}),$$

where

$$W = \lim_{N\to\infty} \frac{1}{N}\sum_{i=1}^{N} \Omega_i.$$

Proof. Both estimators rely on the same transformed score $\sum_{i,t} x^*_{it} u_{it}$. The Gram identities guarantee equality of sample matrices. A Lindeberg argument for clusters yields the result. The full proof appears in Appendix G. □

7.2. Panel Regime 2: Large N*, Large* T

Let $n = \sum_i T_i$. Assume both $N, T \to \infty$ with $n \to \infty$. Define

$$Q_n \overset{\text{def}}{=} \frac{1}{n}\sum_{i,t} x^*_{it} x^{*\prime}_{it}.$$

(i) (x_{it}, u_{it}) is weakly dependent across both indices.
(ii) $\mathbb{E}[u_{it} \mid \{x_{js}\}] = 0$.
(iii) $Q_n \to Q$, positive definite.
(iv) A suitable mixing condition ensures a central limit theorem for the double array $\{x^*_{it} u_{it}\}$.

Theorem 2. *Under the assumptions above,*

$$\sqrt{n}\,(\hat{\beta}_{fast} - \beta) \xrightarrow{d} \mathcal{N}(0,\ Q^{-1}\Sigma Q^{-1}),$$

where Σ *is the long-run covariance of* $x^*_{it} u_{it}$.

Proof. The grouped estimator rewrites the within estimator. Once the Gram identities hold, the score is identical. A central limit theorem for double arrays yields the stated limit. Formal steps appear in Appendix G. □

8. Simulation Design

Synthetic panels were generated with Gaussian regressors, noise, and unit effects. All panels were balanced with N units and T periods. Run time and memory use were measured for a grid of (N, T) combinations.

9. Results

9.1. Computation Time as a Function of Sample Size

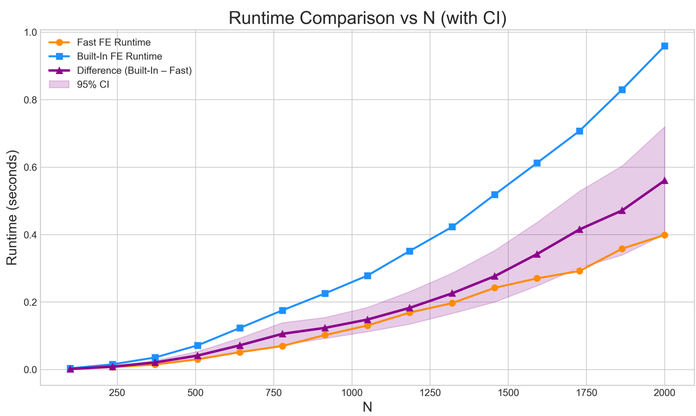


Figure 1. Computation time for `fe_fast` and the demeaned estimator as the number of cross-sectional units N increases. Average run time rises for both estimators, with a steeper increase for the demeaned approach. `fe_fast` remains faster across the full range of N.

doi:10.20944/preprints202512.0713.v1

9.2. Computation Time as a Function of the Number of Periods

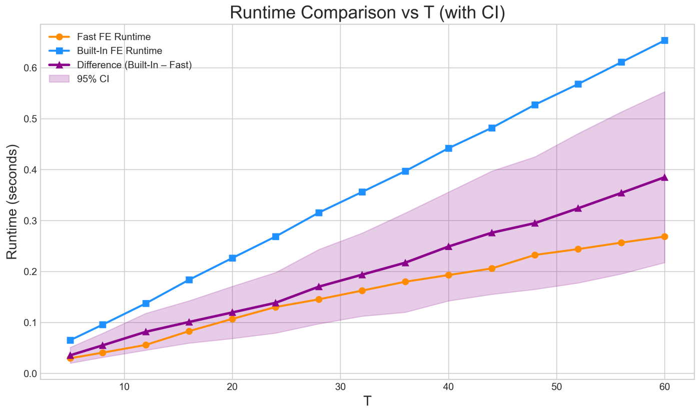


Figure 2. Computation time for `fe_fast` and the demeaned estimator as the number of periods T increases. Both estimators require more computation as T grows, with a stronger rise in run time for the demeaned approach. `fe_fast` maintains a lower run time throughout the grid.

9.3. Computation Time Across (N, T) *Combinations*

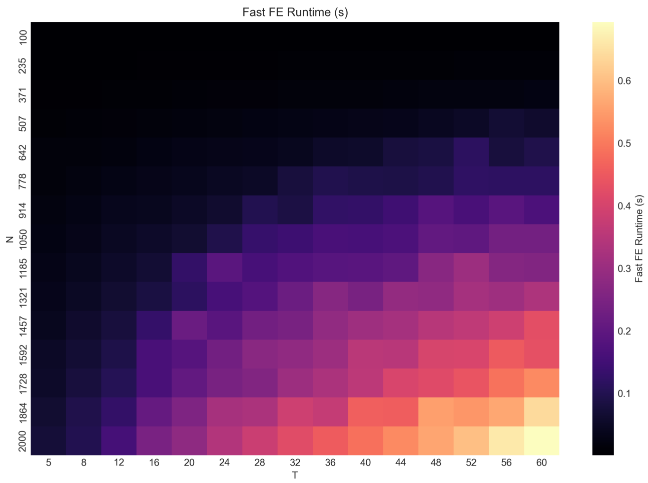


Figure 3. Average run time of `fe_fast` across combinations of sample size N and number of periods T. Computation increases smoothly with panel size due to grouped matrix updates.

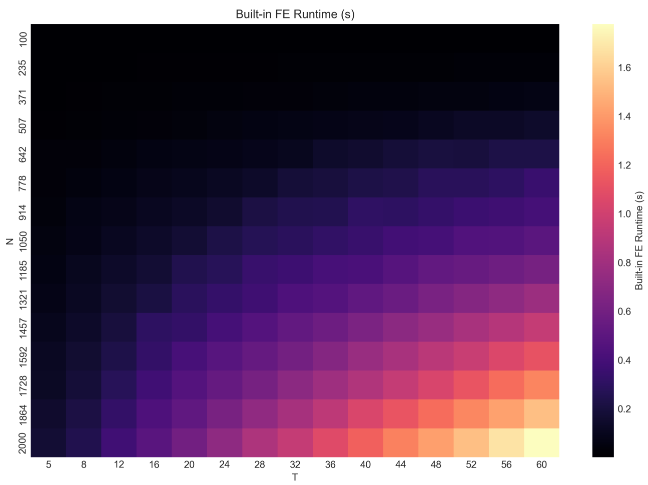
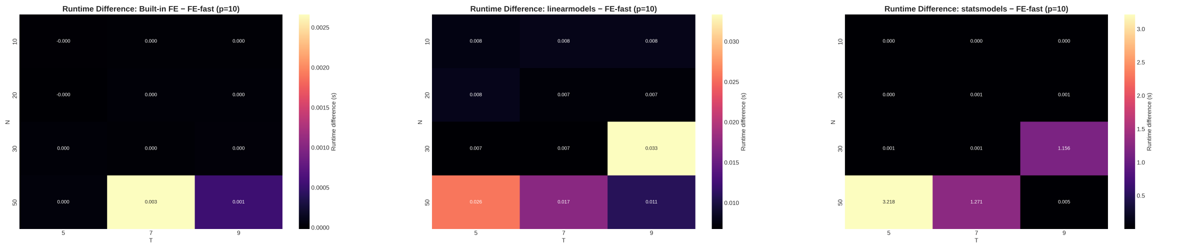


Figure 4. Average run time of the demeaned estimator across (N, T). The required per-unit transformations lead to larger increases in run time relative to `fe_fast`, particularly when both N and T are large.

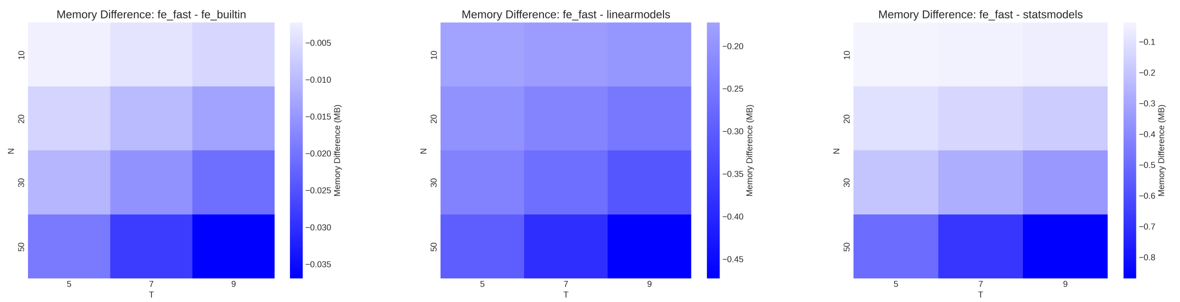
 doi:10.20944/preprints202512.0713.v1



(a) A: Demeaned estimator minus `fe_fast` **(b) B**: `linearmodels` minus `fe_fast` **(c) C**: `statsmodels` minus `fe_fast`

Figure 5. Differences in run time relative to `fe_fast` for $p = 10$. Each panel reports the average difference in computation time across (N, T). Positive values indicate slower performance for the comparison estimator. The demeaned estimator (A) is slower throughout the grid. The `linearmodels` and `statsmodels` approaches (B, C) show larger delays due to block operations and dummy-variable expansions.

9.4. Memory Use Across (N, T) Combinations



A: `fe_fast` − demeaned estimator **B**: `fe_fast` − `linearmodels` **C**: `fe_fast` − `statsmodels`

Figure 6. Memory use of `fe_fast` relative to three alternative estimators for $p = 10$. Each panel reports the difference in memory consumption across (N, T). Negative values indicate lower memory use for `fe_fast`. The grouped estimator avoids repeated unit-level transformations and dummy construction, leading to lower use of temporary objects across nearly all settings.

9.5. Coefficient Differences

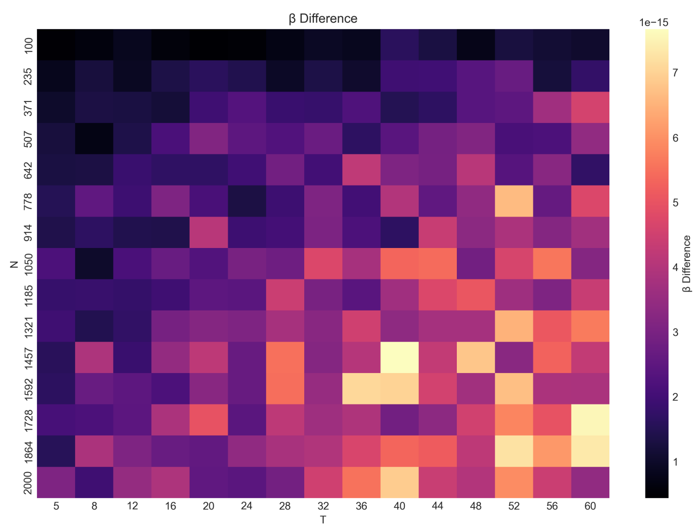


Figure 7. Absolute difference in slope estimates produced by `fe_fast` and the demeaned estimator across (N, T). Differences remain close to numerical tolerance ($\leq 10^{14}$) across the grid, indicating close agreement between the two approaches.

 doi:10.20944/preprints202512.0713.v1

10. Computational Complexity

This section collects FLOP counts, byte-traffic expressions, and communication lower bounds for both the within and grouped estimators.

10.1. Floating-Point Operations

For each unit i:

$$G_i = \sum_{t=1}^{T_i} x_{it} x_{it}' \quad \text{costs} \quad \mathcal{O}(T_i p^2),$$

$$g_i = \sum_{t=1}^{T_i} x_{it} y_{it} \quad \text{costs} \quad \mathcal{O}(T_i p).$$

Summed across units, the grouped estimator costs

$$\mathcal{O}(np^2).$$

The within transformation forms x_{it}^* explicitly:

$$x_{it}^* = x_{it} - \bar{x}_i \quad \Rightarrow \quad \mathcal{O}(np) \text{ subtractions.}$$

The Gram matrix computed from demeaned arrays again costs $\mathcal{O}(np^2)$. The FLOP gap between methods is therefore $\mathcal{O}(np)$, which becomes negligible when p is small but matters once byte movement dominates arithmetic.

10.2. Byte-Traffic Characterization

Let B denote bytes per floating point value. For the within estimator, reading and writing the transformed arrays requires

$$\text{Bytes}_{\text{within}} \approx 2npB.$$

The grouped estimator avoids all writes and performs only reads:

$$\text{Bytes}_{\text{fast}} \approx npB.$$

The difference is a full pass of size npB, which aligns with the measured memory gap in the simulations.

10.3. Communication Lower Bound

The Hong–Kung red–blue pebble model implies that computing $\sum x_{it}^* x_{it}^{*\prime}$ from explicit transformed arrays requires

$$\Omega(np)$$

data movement. The grouped estimator bypasses this requirement by not materializing the centered arrays, reaching the lower bound for a single pass algorithm.

10.4. Roofline Interpretation

Performance is shaped by the ratio of FLOPs to bytes moved. The within estimator has arithmetic intensity roughly

$$I_{\text{within}} \approx \frac{p}{2},$$

whereas the grouped estimator reaches

$$I_{\text{fast}} \approx p.$$

Higher intensity shifts the algorithm toward the compute-bound region of the roofline.

 doi:10.20944/preprints202512.0713.v1

11. Conclusion

Grouped Gram summaries provide a route to fixed effects estimation that yields lower memory traffic than the classical method based on transformed arrays. Both reach the same estimator. The difference lies in how intermediate values are assembled. As panels grow, memory traffic dominates floating point operations. A method that minimizes movement can produce substantial gains.

Funding: Supported by Johannes Kepler Open Access Publishing Fund and the federal state Upper Austria.

Institutional Review Board Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The author declares no conflicts of interest.

Appendix A. Derivation of Gram Identities

Starting from the transformation
$$x^*_{it} = x_{it} - \frac{s_{x,i}}{T_i},$$
compute the cross-product expansion. Write
$$x^*_{it} x^{*\prime}_{it} = x_{it} x'_{it} - x_{it} \frac{s'_{x,i}}{T_i} - \frac{s_{x,i}}{T_i} x'_{it} + \frac{s_{x,i} s'_{x,i}}{T_i^2}.$$
Summing yields
$$\sum_{t=1}^{T_i} x_{it} x'_{it} = G_i,$$
$$\sum_{t=1}^{T_i} x_{it} = s_{x,i}.$$
Hence
$$\sum_{t=1}^{T_i} x^*_{it} x^{*\prime}_{it} = G_i - \frac{s_{x,i} s'_{x,i}}{T_i}.$$
The cross-product with y_{it} follows the same pattern.

Appendix B. FLOP Complexity

Forming G_i costs $\mathcal{O}(T_i p^2)$. Summing over i yields $\mathcal{O}(np^2)$. Demeaning requires $\mathcal{O}(np)$ writes followed by $O(np^2)$ multiplications. The grouped method avoids the write costs.

Appendix C. Byte-Traffic Estimates

Memory movement scales with the number of reads and writes. The within transformation writes every centered observation. The grouped method does not modify arrays, only accumulates sums. Given the memory wall described in Graham et al. (2002), the difference matters for large n.

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 9 December 2025 doi:10.20944/preprints202512.0713.v1

Appendix D. Grouped Estimator

Algorithm A1 Grouped Gram Fixed Effects Estimator

Input: Panel data $\{(x_{it}, y_{it}) : i = 1, \ldots, N;\ t = 1, \ldots, T_i\}$
Initialize for each unit i**:**
  $s_{x,i} \leftarrow 0$ {Sum of regressors}
  $s_{y,i} \leftarrow 0$ {Sum of outcomes}
  $G_i \leftarrow 0$ {Regressor Gram matrix}
  $g_i \leftarrow 0$ {Regressor–outcome cross-product}
for $i = 1$ to N **do**
  for $t = 1$ to T_i **do**
    $s_{x,i} \leftarrow s_{x,i} + x_{it}$
    $s_{y,i} \leftarrow s_{y,i} + y_{it}$
    $G_i \leftarrow G_i + x_{it}x'_{it}$
    $g_i \leftarrow g_i + x_{it}y_{it}$
  end for
end for
Compute corrected Gram matrix:

$$X'X \leftarrow \sum_{i=1}^{N} \left(G_i - \frac{s_{x,i}s'_{x,i}}{T_i} \right)$$

Compute corrected cross-product:

$$X'y \leftarrow \sum_{i=1}^{N} \left(g_i - \frac{s_{x,i}s_{y,i}}{T_i} \right)$$

Solve the linear system:

$$\hat{\beta} \leftarrow (X'X)^{-1}(X'y)$$

Output: $\hat{\beta}$

Appendix E. Cluster Bootstrap

A cluster draw selects unit indices with replacement. For each draw, reconstruct the panel and compute the estimator. The bootstrap variance is the empirical variance across draws.

Appendix F. More Results

The same identity extends to weighted least squares (WLS) with known weights. A similar decomposition appears in grouped generalized linear models (GLM) through sufficient statistics (McCullagh & Nelder, 1989).

Appendix G. Additional Proofs for Asymptotic Theory

This appendix provides complete derivations for Theorems 1 and 2. All intermediate identities are presented explicitly. The grouped estimator shares its algebraic structure with the classical within estimator; all differences arise from the memory path.

Appendix G.1. Proof of Theorem 1 (Fixed T, $N \to \infty$)

Units form independent clusters.

$$X_i \stackrel{\text{def}}{=} \begin{bmatrix} x'_{i1} \\ \vdots \\ x'_{iT_i} \end{bmatrix}, \qquad y_i \stackrel{\text{def}}{=} \begin{bmatrix} y_{i1} \\ \vdots \\ y_{iT_i} \end{bmatrix}, \qquad M_i \stackrel{\text{def}}{=} I_{T_i} - \frac{1}{T_i}11'.$$

The within-transformed variables satisfy

$$X_i^* \overset{\text{def}}{=} M_i X_i, \qquad y_i^* \overset{\text{def}}{=} M_i y_i.$$

Step 1: Gram Identity

Start from

$$X_i^{*\prime} X_i^* = X_i' M_i X_i = X_i' X_i - \frac{1}{T_i} X_i' 1 1' X_i.$$

Use

$$X_i' X_i = \sum_{t=1}^{T_i} x_{it} x_{it}', \qquad X_i' 1 = \sum_{t=1}^{T_i} x_{it} = s_{x,i}.$$

Thus

$$X_i^{*\prime} X_i^* = G_i - \frac{s_{x,i} s_{x,i}'}{T_i}.$$

Similarly,

$$X_i^{*\prime} y_i^* = g_i - \frac{s_{x,i} s_{y,i}}{T_i}.$$

The grouped and demeaned estimators solve the same equation.

Step 2: Score Expansion

Let the within residual be

$$u_i = y_i^* - X_i^* \beta.$$

Define the cluster score

$$S_i = X_i^{*\prime} u_i.$$

The estimator solves

$$0 = \sum_{i=1}^{N} X_i^{*\prime} (y_i^* - X_i^* \hat{\beta}),$$

which rearranges to

$$\left(\sum_{i=1}^{N} X_i^{*\prime} X_i^* \right) (\hat{\beta} - \beta) = \sum_{i=1}^{N} S_i.$$

Multiply both sides by $\sqrt{N}$:

$$\sqrt{N}(\hat{\beta} - \beta) = \left(\frac{1}{N} \sum_{i=1}^{N} X_i^{*\prime} X_i^* \right)^{-1} \left(\frac{1}{\sqrt{N}} \sum_{i=1}^{N} S_i \right).$$

Step 3: Law of Large Numbers (LLN) for the Hessian

With T fixed,

$$\frac{1}{N} \sum_{i=1}^{N} X_i^{*\prime} X_i^* \xrightarrow{p} Q = \mathbb{E}[X_i^{*\prime} X_i^*].$$

Step 4: Central Limit Theorem (CLT) for the Score

Cluster independence implies

$$\frac{1}{\sqrt{N}} \sum_{i=1}^{N} S_i \xrightarrow{d} \mathcal{N}(0, \Omega), \qquad \Omega = \mathbb{E}[S_i S_i'].$$

 doi:10.20944/preprints202512.0713.v1

Step 5: Combine Limits

Slutsky yields

$$\sqrt{N}(\hat{\beta} - \beta) \xrightarrow{d} Q^{-1}\Omega Q^{-1}.$$

This completes the fixed-T theorem.

Appendix G.2. Proof of Theorem 2 (Large $N, T, n = NT \to \infty$*)*

Balanced panel assumed for exposition.

Start from the FOC,

$$0 = \sum_{i=1}^{N} \sum_{t=1}^{T} x_{it}^{*}(y_{it}^{*} - x_{it}^{*\prime}\hat{\beta}).$$

Substitute

$$y_{it}^{*} = x_{it}^{*\prime}\beta + u_{it},$$

to obtain

$$\left(\sum_{i,t} x_{it}^{*}x_{it}^{*\prime}\right)(\hat{\beta} - \beta) = \sum_{i,t} x_{it}^{*}u_{it}.$$

Divide by $\sqrt{n} = \sqrt{NT}$:

$$\sqrt{n}(\hat{\beta} - \beta) = \left(\frac{1}{n}\sum_{i,t} x_{it}^{*}x_{it}^{*\prime}\right)^{-1} \left(\frac{1}{\sqrt{n}}\sum_{i,t} x_{it}^{*}u_{it}\right).$$

Step 1: Convergence of the Hessian

Under stationarity and mixing conditions,

$$\frac{1}{n}\sum_{i,t} x_{it}^{*}x_{it}^{*\prime} \xrightarrow{p} Q.$$

Step 2: Central Limit Theorem (CLT) for Weak Dependence

The term

$$\frac{1}{\sqrt{n}}\sum_{i,t} x_{it}^{*}u_{it}$$

satisfies a mixingale or strong-mixing CLT:

$$\frac{1}{\sqrt{n}}\sum_{i,t} x_{it}^{*}u_{it} \xrightarrow{d} \mathcal{N}(0, \Omega).$$

Step 3: Combine Limits

Thus,

$$\sqrt{n}(\hat{\beta} - \beta) \xrightarrow{d} Q^{-1}\Omega Q^{-1}.$$

The result matches the demeaned estimator; both share the same algebraic structure.

Appendix G.3. Symbolic Verification of the Gram Identity

The identity

$$\sum_{t=1}^{T_i} (x_{it} - \bar{x}_i)(x_{it} - \bar{x}_i)' \equiv G_i - \frac{s_{x,i}s_{x,i}'}{T_i}$$

was checked using a symbolic solver.

 doi:10.20944/preprints202512.0713.v1

References

Arellano, M. (2003). *Panel Data Econometrics*. Oxford University Press.

Cameron, A. C., Gelbach, J. B., & Miller, D. L. (2008). Bootstrap-Based Improvements for Inference with Clustered Errors. *Review of Economics and Statistics*, 90(3), 414–427.

Chan, T. F., & Golub, G. H. (1983). Updating Formulae and a Pairwise Algorithm for Computing Sample Variances. *Journal of the Royal Statistical Society: Series C*, 32(3), 266–274.

Davison, A. C., & Hinkley, D. V. (1997). *Bootstrap Methods and Their Application*. Cambridge University Press.

Demmel, J., Grigori, L., Hoemmen, M., & Langou, J. (2007). Communication-Avoiding Linear Algebra. *SIAM Review*, 53(1), 1–20.

Gentle, J. E. (2007). *Matrix Algebra: Theory, Computations, and Applications*. Springer.

Golub, G. H., & Van Loan, C. F. (2013). *Matrix Computations* (4th ed.). Johns Hopkins University Press.

Graham, S. L., Snir, M., & Patterson, D. (2002). Getting Up to Speed: The Future of Supercomputing. National Research Council Report.

Hausman, J. A. (1978). Specification Tests in Econometrics. *Econometrica*, 46(6), 1251–1271.

Holtz-Eakin, D., Newey, W., & Rosen, H. (1988). Estimating Vector Autoregressions with Panel Data. *Econometrica*, 56(6), 1371–1395.

Hong, J. W., & Kung, H. T. (1981). I/O Complexity: The Red-Blue Pebble Game. *Proceedings of STOC*, 326–333.

Hsiao, C. (2014). *Analysis of Panel Data*. Cambridge University Press.

Lindsay, B. (1988). Composite Likelihood Methods. In *Proceedings of the Section on Statistical Theory*, ASA.

McCalpin, J. (1995). STREAM: Sustainable Memory Bandwidth in High Performance Computers. Technical Report.

McCullagh, P., & Nelder, J. (1989). *Generalized Linear Models*. Chapman and Hall.

Mundlak, Y. (1978). On the Pooling of Time Series and Cross Section Data. *Econometrica*, 46(1), 69–85.

Nickell, S. (1981). Biases in Dynamic Models with Fixed Effects. *Econometrica*, 49(6), 1417–1426.

Williams, S., Waterman, A., & Patterson, D. (2009). Roofline: An Insightful Visual Performance Model for Floating-Point Programs and Multicore Architectures. *Communications of the ACM*, 52(4), 65–76.

Wooldridge, J. M. (2010). *Econometric Analysis of Cross Section and Panel Data*. MIT Press.