BEYOND IMPUTATION: RETHINKING MATRIX MULTIPLICATION UNDER MISSING DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

Matrix multiplication is a fundamental operation behind machine learning, yet the usual operator becomes undefined when either operand contains missing entries. Existing solutions typically rely on task-specific adaptations or imputation-based pipelines, both of which can introduce additional complexity, computational costs, and task-specific development efforts. To address these limitations, we introduce Missing-aware Matrix Multiplication, denoted M^3, which estimates unavailable multiplicative contributions directly through conditional moments. Under a pair-local Gaussian working model, four observation patterns admit closed-form contributions, while marginal moments exploit all available observations and estimate dependence only from overlapping entries. We prove pair-local oracle optimality and show that exploiting cross-factor dependence strictly reduces mean-squared error relative to mean imputation. Across direct product recovery, Gram and PCA recovery, cross-view spectral estimation, and recommendation tasks, our method generally improves effectiveness while remaining computationally efficient. These findings suggest that M^3 provides a promising and efficient approach to matrix multiplication in the presence of missing data.

1 INTRODUCTION

Matrix multiplication is a primitive operation underlying machine learning. Gram and covariance matrices, similarity calculations, linear projections, principal-component methods, and cross-view models all reduce to products of the form:

$$\mathbf{C} = \mathbf{AB}, \qquad C_{ij} = \sum_{k=1}^{d} A_{ik} B_{kj}. \tag{1}$$

Here, $\mathbf{A} \in \mathbb{R}^{r \times d}$ and $\mathbf{B} \in \mathbb{R}^{d \times s}$. When either operand is incomplete, however, the usual operator cannot be applied without first deciding how missing factors should contribute to the sum. Existing approaches largely follow two directions:

Impute then compute. This conventional pipeline reconstructs the missing entries and applies the ordinary product. It covers a wide range of imputation methods, from simple approaches (Acuna & Rodriguez, 2004; Zhang, 2012) to more sophisticated methods (Deng & Lumley, 2024; Vo et al., 2026c; Yoon et al., 2018; Mattei & Frellsen, 2019a). This strategy can be effective for a single incomplete matrix, especially when a suitable imputation method is readily available, and the reconstructed data can be reused for subsequent analyses. However, imputation can introduce substantial computational overhead, particularly for large matrices or complex imputation models (Jäger et al., 2021). In addition, imputation inaccuracies can propagate to subsequent computations and affect downstream results (Oh et al., 2011; Shadbahr et al., 2023; Verma et al., 2025). Moreover, downstream performance can be highly sensitive to the choice of imputation method, making it difficult to identify an appropriate imputer for a given task and dataset (Jäger et al., 2021; Deforth et al., 2024).

Develop task-specific methods. This second line of work develops methods tailored to particular downstream tasks under missing data, rather than reconstructing the incomplete data first. For example, one can directly estimate the covariance matrix of incomplete data (Nguyen et al., 2022; Vo et al., 2026d; Yu et al., 2025b). Such approaches can be effective in their native settings because they are specifically designed around the structure and objective of the target problem. However,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

this specialization also limits their generality. Because missing data is common in many analytical problems, this creates a fragmented landscape in which solutions do not easily transfer from one task to another.

The product itself suggests a simpler target. For the kth term in Equation (1), the missing object is the contribution $A_{ik}B_{kj}$, not necessarily the two factors separately. If one factor is observed and the other is missing, the observed value can still inform the conditional expectation of their product. If both are missing, the relevant quantity is their cross-moment. Instead of reconstructing complete matrices and then multiplying them, we aim to answer a more direct research question: *How should each missing multiplicative contribution be estimated from the information available at that exact location?*. This leads to our proposed method, Missing-aware Matrix Multiplication (M^3), a training-free operator that computes matrix products from partially observed inputs by exploiting pair-local observed information.

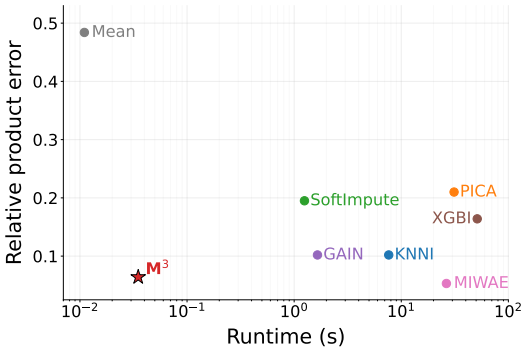

Figure 1: Error-runtime comparison for direct multiplication of incomplete matrices at $(r, d, s) = (128, 2048, 128)$, under 50% MCAR missingness. Lower and further left are better. Full results are reported in Appendix D. M^3 remains competitive with deep learning methods while retaining the same leading-order runtime as mean imputation.

For each output entry, M^3 uses only the corresponding row-column pair and replaces an unavailable product with its conditional expectation given the locally observed factors. This training-free and simply reusable operator can handle distinct and independently masked operands, and have the same $O(rds)$ order complexity as dense multiplication (see Figure 1). To the best of our knowledge, M^3 is the first operator designed to directly estimate a matrix product $\mathbf{AB}$ when both operands may contain missing entries.

The main contributions and structure of this work can be summarized as follows:

- We formulate incomplete matrix multiplication as direct estimation of missing multiplicative contributions and derive M^3, a training-free product operator with $O(rds)$ complexity (Section 3).
- We establish pair-local oracle optimality and quantify the exact MSE reduction obtained by exploiting cross-factor dependence (Section 4).
- We evaluate M^3 on direct matrix multiplication, Gram recovery, principal component analysis (PCA), partial least squares (PLS), and recommender systems, demonstrating strong performance against imputation and task-specific baselines (Section 5).

2 Related Works

We focus on methods most directly related to computation with incomplete matrices. Broader topics such as missingness mechanism modeling and domain-specific applications of missing-data methods fall outside the scope of this work.

Imputation methods. This strategy remains an active line of research, as the suitability of a method can depend strongly on the data type, downstream task, and application setting (Le Morvan & Varoquaux, 2025; Vo et al., 2026a). This has led to a broad range of approaches, from simple statistical and neighborhood-based methods, such as mean imputation (Acuna & Rodriguez, 2004) and KNNI (Zhang, 2012), to low-rank methods such as SoftImpute (Mazumder et al., 2010; Vaknin Laufer & Nadler, 2026). Next, we can name multiple and tree-based approaches such as XGBI (Deng & Lumley, 2024), covariance-aware methods such as PICA (Vo et al., 2026c), and deep generative models such as GAIN (Yoon et al., 2018) and MIWAE (Mattei & Frellsen, 2019a). In matrix multiplication, these methods reconstruct the incomplete data and then compute the matrix product as usual, whereas M^3 directly estimates the final product.

Task-specific estimation under missing data. Another line of work avoids full reconstruction by directly estimating quantities required for a particular downstream task (Vo et al., 2024). In covariance estimation, DPER was shown empirically to outperform conventional impute-then-compute pipelines for continuous data (Nguyen et al., 2022), while DPERC extends this idea to mixed-type data (Vo et al., 2026d). Related work further considers positive-semidefinite covariance recovery (Sood & Hastie, 2025), PCA under incomplete observations (Yan et al., 2024; Do et al., 2024), and probabilistic formulations such as PPCA that naturally accommodate missing entries (Palummo et al., 2024). Most closely related to our setting, EC-IID (Yu et al., 2025b) directly estimates inner products from incomplete observations and is therefore relevant to self-product problems such as Gram and PCA eigenvalue recovery. However, inner-product estimation addresses a more specific setting than the general multiplication of two incomplete matrices. Beyond second-order estimation, recent work has also developed missing-data-specific procedures for predictive modeling (Stempfle et al., 2025) and score matching (Givens et al., 2025). While these methods can be effective in their intended settings, M^3 targets the matrix product itself, estimating $\mathbf{AB}$ directly while explicitly accounting for all four observation patterns of each multiplicative contribution.

3 The Proposed Method: M^3 Operator

This section introduces the M^3 operator, beginning by formulating the problem in Section 3.1. Next, in Section 3.2, we derive the Gaussian conditional-product estimator and its parameter estimates. Section 3.3 then presents the resulting M^3 operator, implementation details, and computational complexity. All proofs for propositions are deferred to Appendix A.

3.1 Problem Formulation

Let $\mathbf{A} \in \mathbb{R}^{r\times d}$ and $\mathbf{B} \in \mathbb{R}^{d\times s}$ be incomplete matrices with observation masks $\mathbf{M}^A$ and $\mathbf{M}^B$, respectively, where mask value 1 indicates an observed entry. We use A_{ik} and B_{kj} to denote the underlying values, whether observed or missing. For a fixed (i, j), let X_k and Y_k denote the kth entries of row i of $\mathbf{A}$ and column j of $\mathbf{B}$, respectively:

$$X_k = A_{ik}, \qquad Y_k = B_{kj}, \qquad k = 1, \dots, d. \tag{2}$$

The observed information of the kth product term is $O_k = (M^A_{ik}, M^B_{kj}, M^A_{ik}X_k, M^B_{kj}Y_k)$. The coordinates can be partitioned into four observation patterns:

$$\begin{aligned} \mathcal{I}^{11}_{ij} &= \{k : M^A_{ik} = 1,\ M^B_{kj} = 1\}, & \mathcal{I}^{10}_{ij} &= \{k : M^A_{ik} = 1,\ M^B_{kj} = 0\}, \\ \mathcal{I}^{01}_{ij} &= \{k : M^A_{ik} = 0,\ M^B_{kj} = 1\}, & \mathcal{I}^{00}_{ij} &= \{k : M^A_{ik} = 0,\ M^B_{kj} = 0\}. \end{aligned} \tag{3}$$

A complete-pair estimator discards the one-sided observations in $\mathcal{I}^{10}_{ij}$ and $\mathcal{I}^{01}_{ij}$. Instead, for any joint distribution with finite second moments, the ideal contribution of coordinate k is $\mathbb{E}[X_kY_k \mid O_k]$. The corresponding oracle product estimator is $\widetilde{C}_{ij} = \sum_{k=1}^{d} \mathbb{E}[X_kY_k \mid O_k]$. This captures the main idea of M^3: estimating each product contribution from the information available at the point needed. We next develop a lightweight Gaussian realization of this principle.

3.2 Gaussian Conditional-Product Estimator

Assumption 3.1. For each output pair (i, j),

$$\begin{pmatrix} X_k \\ Y_k \end{pmatrix} \overset{\text{i.i.d.}}{\sim} \mathcal{N}_2\left(\begin{pmatrix} \mu^A_i \\ \mu^B_j \end{pmatrix}, \begin{pmatrix} v^A_i & \tau_{ij} \\ \tau_{ij} & v^B_j \end{pmatrix}\right), \qquad k = 1, \dots, d, \tag{4}$$

where $v^A_i > 0$, $v^B_j > 0$, and $\tau^2_{ij} < v^A_i v^B_j$. Moreover, the observation masks are supposed to be independent of the underlying values (X_k, Y_k).

This assumption is local to each output pair (i, j) and does not impose a global multivariate Gaussian model on $\mathbf{A}$ or $\mathbf{B}$. It is most natural when the inner dimension indexes observations drawn from a common distribution; otherwise, it should be interpreted as a working approximation.

Proposition 3.2. *Under Assumption 3.1, then*

$$\mathbb{E}[X_k Y_k \mid O_k] = \begin{cases} X_k Y_k, & k \in \mathcal{I}_{ij}^{11}, \\ X_k \left[\mu_j^B + \dfrac{\tau_{ij}}{v_i^A}(X_k - \mu_i^A)\right], & k \in \mathcal{I}_{ij}^{10}, \\ Y_k \left[\mu_i^A + \dfrac{\tau_{ij}}{v_j^B}(Y_k - \mu_j^B)\right], & k \in \mathcal{I}_{ij}^{01}, \\ \mu_i^A \mu_j^B + \tau_{ij}, & k \in \mathcal{I}_{ij}^{00}. \end{cases} \tag{5}$$

This proposition shows that each product contribution is determined by the local first- and second-order moments $\theta_{ij} = \left(\mu_i^A, \mu_j^B, v_i^A, v_j^B, \tau_{ij}\right)$. These quantities are unknown in practice and must therefore be estimated from the observed entries. Importantly, the marginal moments can exploit all individually observed coordinates, whereas the dependence term τ_{ij} requires jointly observed pairs. To estimate these quantities, define the observed-coordinate sets $\mathcal{O}_i^A = \{k : M_{ik}^A = 1\}$ and $\mathcal{O}_j^B = \{k : M_{kj}^B = 1\}$, and let $m_{ij} = |\mathcal{I}_{ij}^{11}|$. The following proposition characterizes the resulting plug-in estimators and reduces estimation of the pair-specific dependence parameter to a polynomial equation.

Proposition 3.3. *Under Assumption 3.1, the marginal Gaussian maximum-likelihood estimators are*

$$\begin{aligned} \widehat{\mu}_i^A &= \frac{1}{|\mathcal{O}_i^A|} \sum_{k \in \mathcal{O}_i^A} A_{ik}, \quad \widehat{v}_i^A = \frac{1}{|\mathcal{O}_i^A|} \sum_{k \in \mathcal{O}_i^A} (A_{ik} - \widehat{\mu}_i^A)^2, \\ \widehat{\mu}_j^B &= \frac{1}{|\mathcal{O}_j^B|} \sum_{k \in \mathcal{O}_j^B} B_{kj}, \quad \widehat{v}_j^B = \frac{1}{|\mathcal{O}_j^B|} \sum_{k \in \mathcal{O}_j^B} (B_{kj} - \widehat{\mu}_j^B)^2. \end{aligned} \tag{6}$$

Using the jointly observed coordinates, define

$$S_{ij}^A = \sum_{k \in \mathcal{I}_{ij}^{11}} (A_{ik} - \widehat{\mu}_i^A)^2, S_{ij}^B = \sum_{k \in \mathcal{I}_{ij}^{11}} (B_{kj} - \widehat{\mu}_j^B)^2, S_{ij}^{AB} = \sum_{k \in \mathcal{I}_{ij}^{11}} (A_{ik} - \widehat{\mu}_i^A)(B_{kj} - \widehat{\mu}_j^B). \tag{7}$$

For $|t| \le (1-\epsilon)\sqrt{\widehat{v}_i^A \widehat{v}_j^B}$, any interior maximizer of the plug-in log-likelihood for τ_{ij} must be a real root of

$$p_{ij}(t) = m_{ij} t^3 - S_{ij}^{AB} t^2 + \left(\widehat{v}_j^B S_{ij}^A + \widehat{v}_i^A S_{ij}^B - m_{ij} \widehat{v}_i^A \widehat{v}_j^B\right) t - \widehat{v}_i^A \widehat{v}_j^B S_{ij}^{AB}. \tag{8}$$

3.3 M^3 Operator, Implementation, and Complexity

Operator. The practical M^3 operator is obtained by substituting the estimated moments into the conditional-product rule of Proposition 3.2. For each output entry,

$$\begin{aligned} \widehat{C}_{ij} = &\sum_{k \in \mathcal{I}_{ij}^{11}} A_{ik} B_{kj} + \sum_{k \in \mathcal{I}_{ij}^{10}} A_{ik} \left[\widehat{\mu}_j^B + \frac{\widehat{\tau}_{ij}}{\widehat{v}_i^A}(A_{ik} - \widehat{\mu}_i^A)\right] \\ &+ \sum_{k \in \mathcal{I}_{ij}^{01}} B_{kj} \left[\widehat{\mu}_i^A + \frac{\widehat{\tau}_{ij}}{\widehat{v}_j^B}(B_{kj} - \widehat{\mu}_j^B)\right] + |\mathcal{I}_{ij}^{00}| \left(\widehat{\mu}_i^A \widehat{\mu}_j^B + \widehat{\tau}_{ij}\right). \end{aligned} \tag{9}$$

The resulting matrix estimate is $\widehat{\mathbf{C}} = [\widehat{C}_{ij}]_{i=1,j=1}^{r,s}$.

Implementation notes. For fixed marginal estimates, the plug-in log-likelihood used to select the covariance is:

$$\ell_{ij}(t) = -\frac{m_{ij}}{2} \log\left(\widehat{v}_i^A \widehat{v}_j^B - t^2\right) - \frac{\widehat{v}_j^B S_{ij}^A + \widehat{v}_i^A S_{ij}^B - 2t S_{ij}^{AB}}{2\left(\widehat{v}_i^A \widehat{v}_j^B - t^2\right)}. \tag{10}$$

When Equation (8) has multiple feasible real roots, we evaluate ℓ_{ij} at these roots and at the two stability boundaries and retain the maximizing candidate, denoted by $\widehat{\tau}_{ij}^{\mathrm{PL}}$. If $m_{ij} = 0$, no pair-local information is available to estimate dependence and we set $\widehat{\tau}_{ij}^{\mathrm{PL}} = 0$.

Algorithm 1 M^3: Missing-aware matrix multiplication

Require: Incomplete matrices $\mathbf{A}, \mathbf{B}$; masks $\mathbf{M}^A, \mathbf{M}^B$; shrinkage λ_τ; stability constant ϵ.
Ensure: Product estimate $\widehat{\mathbf{C}}$.
Estimate row moments $\{\widehat{\mu}_i^A, \widehat{v}_i^A\}_{i=1}^r$ and column moments $\{\widehat{\mu}_j^B, \widehat{v}_j^B\}_{j=1}^s$.
for $i = 1, \ldots, r$ **do**
  for $j = 1, \ldots, s$ **do**
    Determine the four observation patterns in Equation (3).
    Compute S_{ij}^A, S_{ij}^B, S_{ij}^{AB}, and m_{ij}.
    Estimate $\widehat{\tau}_{ij}^{\mathrm{PL}}$ from Equation (8) and ℓ_{ij}.
    Shrink and clip $\widehat{\tau}_{ij}$ using Equations (11)–(12).
    Compute $\widehat{C}_{ij}$ using Equation (9).
  end for
end for
return $\widehat{\mathbf{C}}$.

To reduce instability when the overlap is small, the covariance estimate is shrunk according to

$$\widetilde{\tau}_{ij} = \frac{m_{ij}}{m_{ij} + \lambda_\tau} \widehat{\tau}_{ij}^{\mathrm{PL}}, \tag{11}$$

and clipped to the valid Gaussian covariance range:

$$\widehat{\tau}_{ij} = \mathrm{clip}\left(\widetilde{\tau}_{ij}, -(1-\epsilon)\sqrt{\widehat{v}_i^A \widehat{v}_j^B}, (1-\epsilon)\sqrt{\widehat{v}_i^A \widehat{v}_j^B}\right). \tag{12}$$

Thus, sparse overlap pulls the estimated dependence toward zero, whereas sufficiently large overlap leaves the likelihood estimate nearly unchanged. Combining the moment estimates, covariance stabilization, and conditional-product rule yields the complete M^3 procedure summarized in Algorithm 1.

Complexity. Estimating all marginal moments costs $O((r+s)d)$. For each output pair (i, j), accumulating the sufficient statistics and evaluating Equation (9) costs $O(d)$, while solving the cubic and selecting among its finite candidate set costs $O(1)$. Hence, the overall time complexity is $O(rds)$, with memory complexity $O(rs + r + s)$ including the output. Thus, M^3 has the same leading-order arithmetic complexity as dense matrix multiplication while avoiding constructing and storing completed versions of the input matrices. If $\widehat{\tau}_{ij} = 0$ for all (i, j), Equation (9) reduces to multiplying matrices completed by observed row and column means. Hence, the zero-dependence version of M^3 coincides with the mean-imputation baseline.

4 Theoretical Guarantees

In this section, we address two questions about the pair-local oracle underlying M^3. First, among estimators using the same pair-local observed information, is the conditional-product oracle optimal? Second, how much can be gained, in terms of squared error, by exploiting cross-factor dependence rather than ignoring it? Theorem 4.2 answers the first question by establishing pair-local oracle optimality, while Theorem 4.3 answers the second by quantifying the exact squared-error reduction. The proofs are deferred to Appendix A.

Assumption 4.1. For a fixed output pair (i, j), the tuples $(X_k, Y_k, M_{ik}^A, M_{kj}^B)$ are independent across $k = 1, \ldots, d$, with $\mathbb{E}[X_k^2 Y_k^2] < \infty$. Let $\mathcal{F}_{ij} = \sigma(O_1, \ldots, O_d)$ denote the pair-local observed information.

Theorem 4.2. *Under Assumption 4.1, then*

$$\widetilde{C}_{ij} = \sum_{k=1}^{d} \mathbb{E}[X_k Y_k \mid O_k] = \mathbb{E}[C_{ij} \mid \mathcal{F}_{ij}]. \tag{13}$$

Moreover, for every square-integrable $\mathcal{F}_{ij}$-measurable estimator T,

$$\mathbb{E}[(C_{ij} - \widetilde{C}_{ij})^2] \le \mathbb{E}[(C_{ij} - T)^2]. \tag{14}$$

Consequently, $\widetilde{C}_{ij}$ uniquely minimizes the mean-squared product error, up to almost-sure equality, among all estimators based on the same pair-local information.

Note that Theorem 4.2 establishes pair-local rather than global optimality: methods using other rows or columns may exploit additional information outside $\mathcal{F}_{ij}$.

Theorem 4.3. *Under Assumption 3.1, let $\psi^\star$ denote the conditional-product rule in Equation (5), and let ψ_0 denote the same rule with $\tau_{ij} = 0$. For each observation pattern $(a, b) \in \{0, 1\}^2$, let $\mathrm{MSE}_{ab}(\psi)$ denote the mean-squared error of a rule ψ for estimating the product contribution $X_k Y_k$ within that pattern $\mathrm{MSE}_{ab}(\psi) = \mathbb{E}\big[(X_k Y_k - \psi(O_k))^2 \mid k \in \mathcal{I}_{ij}^{ab}\big]$. Then*

$$\mathrm{MSE}_{ab}(\psi_0) - \mathrm{MSE}_{ab}(\psi^\star) = \begin{cases} 0, & (a,b) = (1,1), \\ \tau_{ij}^2 \left(3 + \dfrac{(\mu_i^A)^2}{v_i^A}\right), & (a,b) = (1,0), \\ \tau_{ij}^2 \left(3 + \dfrac{(\mu_j^B)^2}{v_j^B}\right), & (a,b) = (0,1), \\ \tau_{ij}^2, & (a,b) = (0,0). \end{cases} \tag{15}$$

Thus, exploiting dependence strictly reduces the mean-squared error for every incomplete pattern whenever $\tau_{ij} \neq 0$.

In Theorem 4.3, for centered variables, M^3 reduces the MSE relative to mean imputation by $3\tau_{ij}^2$, $3\tau_{ij}^2$, and τ_{ij}^2 for the three incomplete patterns.

These results characterize the oracle conditional-product rule underlying M^3; the practical operator in Section 3.3 replaces its unknown moments with the corresponding plug-in estimates. Under standard regularity conditions, the moment estimators are consistent as the effective number of observed coordinates grows (Van der Vaart, 2000); with fixed λ_τ, the shrinkage vanishes asymptotically, while clipping is inactive when the true covariance lies in its interior. By continuity of the conditional-product rule, the resulting plug-in contributions therefore converge to their oracle counterparts, so Theorem 4.2 provides the natural asymptotic benchmark for practical M^3.

5 Experiments and Results

This section presents experiments to benchmark M^3. We first outline the experimental setup in Section 5.1. Then, in Section 5.2, we compare our method against impute-then-compute pipelines in different controlled data distributions. Next, we evaluate whether direct estimation of incomplete matrix products remains effective in different settings, from self-products to cross-view spectral quantities, and downstream recommendation in Sections 5.3, 5.4, and 5.5 respectively.

5.1 Experimental setup

Datasets. We synthesize data for direct matrix multiplication, allowing controlled evaluation across different distributions. Self-product experiments use ANES96, Fair, RandHIE, Economics, and Texas Housing, while cross-view experiments use Diamonds, Breast Cancer, Gapminder, and Seals (Seabold & Perktold, 2010; Wickham, 2016; Pedregosa et al., 2011; Bryan, 2025). For the recommendation experiment, we use the Jester explicit-rating dataset (Goldberg et al., 2001).

Missingness. For the direct-multiplication, self-product, and cross-view benchmarks, missing values are generated under the MCAR mechanism at rates of 20%, 30%, 50%, and 70%, following the masking procedure of Vo et al. (2026c). In the recommendation experiment, 40% of each user's ratings are first held out for evaluation, after which 50% of the remaining training ratings are independently masked under MCAR.

Compared methods. We compare M^3 with two groups of competitors. For each method, including M^3, we use a fixed hyperparameter configuration across all applicable experiments; implementation details are provided in Appendices B.2 and B.3.

- **Imputation methods.** We include seven representative methods spanning a broad range of techniques, from simple statistical and neighborhood-based approaches to low-rank, tree-based, and deep generative methods: Mean, KNNI (Zhang, 2012), SoftImpute (Mazumder

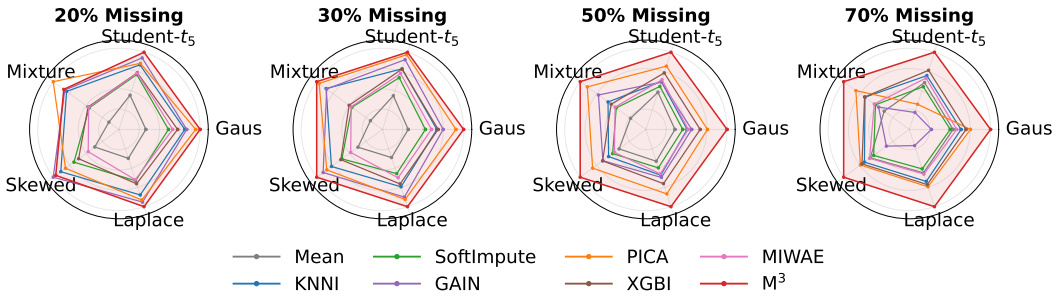

Figure 2: Direct matrix multiplication across five data-generating distributions. Each panel corresponds to one missing rate. The radial score is the best relative Frobenius error at that distribution divided by the method's error; higher is better and 1 is best. Details are in Appendix C.

et al., 2010), GAIN (Yoon et al., 2018), PICA (Vo et al., 2026c), XGBI (Deng & Lumley, 2024), and MIWAE (Mattei & Frellsen, 2019a).

- **Task-specific methods.** We additionally include task-specific competitors when applicable: Pairwise-PSD (Sood & Hastie, 2025), HeteroPCA (Yan et al., 2024), and EC-IID (Yu et al., 2025b) are used for self-products. Gram recovery additionally includes DPER (Nguyen et al., 2022), while PCA includes PPCA-EM (Palummo et al., 2024). For recommendation, we include CoRated-ItemKNN (Sarwar et al., 2001) and Shrunk-ItemKNN (Koren et al., 2021) as task-specific item-based collaborative-filtering baselines.

Metrics. Unless otherwise stated, recovery quality is measured by the relative error

$$E_{\mathrm{rel}}(\widehat{\mathbf{T}}, \mathbf{T}) = \frac{\|\widehat{\mathbf{T}} - \mathbf{T}\|}{\|\mathbf{T}\|}, \tag{16}$$

where $\mathbf{T}$ denotes the complete data, and lower is better. We use the Frobenius norm for matrices and the ℓ_2 norm for vectors. All results are averaged over 10 independent runs. For visual clarity, the main figures report only the mean performance, while the corresponding numerical results, including standard deviations across runs, are provided in Appendices C and D.

5.2 Simulation: Direct Matrix Multiplication

We first evaluate the matrix product directly, without any downstream transformation. We generate $\mathbf{A} \in \mathbb{R}^{16\times 256}$ and $\mathbf{B} \in \mathbb{R}^{256\times 16}$ under Gaussian, Student-t_5, shifted Gaussian-mixture, skewed, and Laplace distributions. Product recovery is evaluated using the relative Frobenius error in Equation (16). For visualization, Figure 2 reports the normalized score $\min_{m'} E^{(m')}_{\mathrm{prod}} / E^{(m)}_{\mathrm{prod}}$, so that 1 denotes the best-performing method. M^3 achieves the best product recovery across all five distributions and missing rates, with the only exceptions occurring at 20% missingness for the mixture and skewed distributions, where PICA and GAIN perform best, respectively. A plausible explanation is that, at low missingness, these more flexible models can exploit the abundant observed entries to better capture the underlying non-Gaussian structure. The advantage of M^3 becomes more pronounced as the missing rate increases, while the others' performance varies substantially across distributions. For example, GAIN performs competitively at 20% and 30% missingness, but drops to the bottom of the benchmark at 70%. This may reflect GAIN's greater dependence on sufficient observed data for learning the underlying distribution. Notably, our method remains strong under Student-t_5, mixture, skewed, and Laplace data, indicating that the Gaussian working approximation is robust to substantial distributional departures.

Scaling behavior. We next isolate scaling with the inner and outer dimensions under Gaussian data and 50% MCAR missingness. We first fix $r = s = 16$ and increase d from 256 to 4096, and then fix $d = 1024$ and increase $r = s = p$ from 16 to 128. Figure 3 shows that M^3 maintains low product error in both regimes while remaining substantially faster than computationally intensive impute-then-compute alternatives. These results empirically complement its $O(rds)$ complexity. Joint-scaling results under Gaussian and Student-t_5 data are provided in Appendix D.

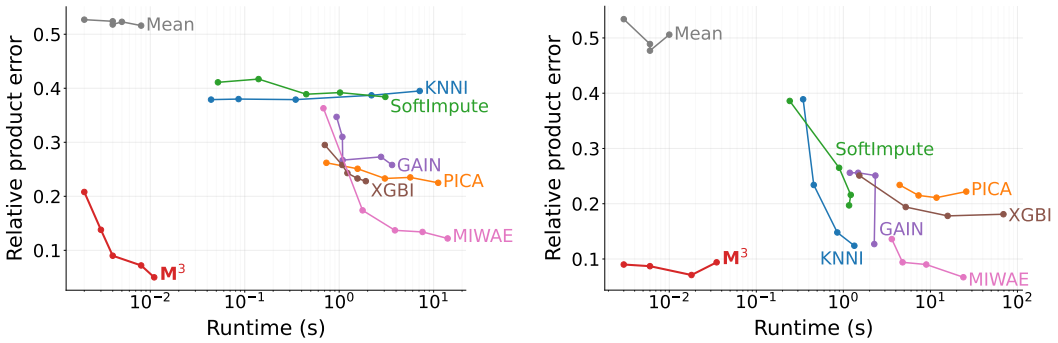

Figure 3: Controlled scaling experiments under 50% MCAR missingness and Gaussian data. **(Left)** inner-dimension scaling with $r = s = 16$ and increasing $d \in \{256, 512, 1024, 2048, 4096\}$. **(Right)** outer-dimension scaling with $d = 1024$ and increasing $r = s = p \in \{16, 32, 64, 128\}$. Full numerical results are reported in Appendix D.



Figure 4: Self-product recovery on five real-world datasets. Top: relative error of the centered Gram matrix. Bottom: relative error of the two leading PCA eigenvalues. Details are in Appendix C.

5.3 Self-products: Gram and PCA eigenvalues recovery

We next consider self-products, where the product encodes second-order structure. On five real-world datasets, we evaluate two related targets: the centered Gram matrix $\mathbf{X}_c^\top \mathbf{X}_c$ and the two leading eigenvalues derived from it. The former measures direct recovery of second-order structure, whereas the latter tests whether product accuracy is preserved in a downstream spectral task. Imputation methods first reconstruct the incomplete data, while the task-specific baselines, such as Pairwise-PSD, EC-IID, HeteroPCA, DPER, and PPCA-EM, operate directly on the incomplete observations or their second-order structure. As shown in Figure 4, imputation-based methods exhibit progressively larger errors as the missing rate increases, whereas M^3 more effectively preserves the underlying second-order structure. This advantage also holds for recovering the leading PCA spectrum. This indicates that the benefit of direct product estimation is not limited to the matrix product itself, but also persists after spectral decomposition.

5.4 Cross-view products: PLS spectral recovery

We move beyond self-products and evaluate cross-view structure recovery. For paired standardized views $\mathbf{X} \in \mathbb{R}^{n \times p}$ and $\mathbf{Y} \in \mathbb{R}^{n \times q}$, the target is the centered cross-product $\mathbf{S}_{XY} = n^{-1}\mathbf{X}_c^\top \mathbf{Y}_c$. Since PLS-SVD depends on the singular structure of this matrix (Mergny & Zdeborová, 2026), we evaluate the recovery of its two leading singular values on the Diamonds, Breast Cancer, Gapminder, and Seals datasets. The two views are masked independently. The fixed feature partitions are given in Appendix B. Fig. 5 shows that M^3 generally achieves the lowest singular-value error across settings, indicating that its advantage extends from self-products to genuine cross-view products.

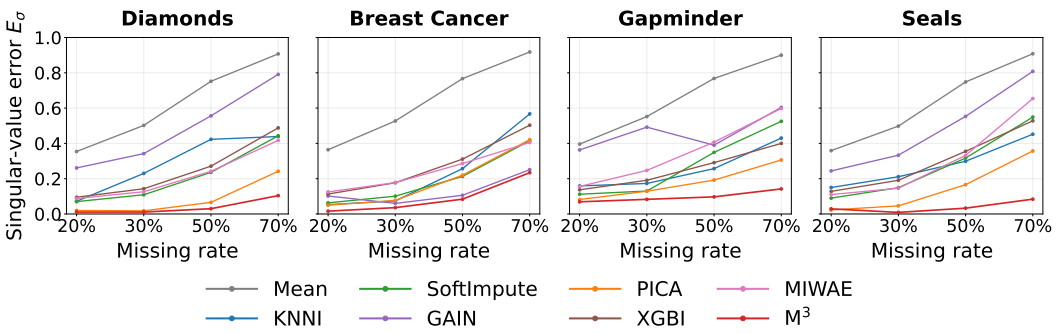

Figure 5: Cross-view PLS spectral recovery on four real-world datasets. Relative error of the two leading singular values. Details are in Appendix C.

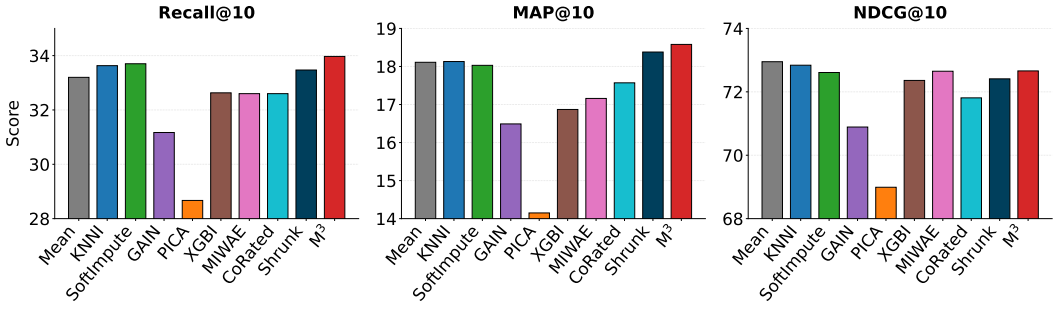

Figure 6: Recommendation performance on Jester dataset with 50% missing training entries. Score results are reported in percentages; higher values are better. Details are in Appendix C.

5.5 Application to Recommender System

We finally evaluate whether product recovery translates into downstream recommendation quality using a 1000×100 subset of the Jester rating dataset (Goldberg et al., 2001), whose relatively dense explicit ratings support controlled evaluation of item–item products under missingness. For each user, 40% of the items are held out, and 50% of the remaining training ratings are masked. Imputation methods reconstruct the rating matrix before forming the item–item product, whereas M^3 estimates it directly. We additionally include CoRated-ItemKNN (Sarwar et al., 2001) and Shrunk-ItemKNN (Koren et al., 2021) as task-specific baselines. All methods use the same downstream ranking procedure. As shown in Figure 6, M^3 achieves the best Recall@10 and MAP@10 while remaining competitive on NDCG@10. This suggests that direct product estimation effectively preserves the item-neighborhood structure needed to retrieve relevant items, while offering less advantage for the finer-grained ordering captured by NDCG. Thus, while different baselines are competitive for different evaluation criteria, M^3 delivers consistently strong performance across all three metrics. These results indicate that accurate direct estimation of the item-item product can translate effectively into downstream quality.

6 Conclusion

Our study of M^3 reveals that matrix multiplication under missing data can be treated as a direct estimation problem rather than a preprocessing problem. By estimating multiplicative contributions locally, the method avoids full imputation while retaining strong accuracy, theoretical guarantees, and practical efficiency. Across direct product recovery, second-order structure estimation, and downstream recommendation, the results show that this perspective remains effective across different computational settings. More broadly, the results suggest that some missing-data problems are better solved by redesigning the operator itself. This viewpoint may also motivate missing-aware formulations of other fundamental operations beyond matrix multiplication.

AI USE STATEMENT

Generative AI tools were used to assist with language editing, literature search, code debugging, and to provide feedback on experimental design and presentation. All AI-assisted work was independently reviewed and verified by the authors. The authors take full responsibility for the correctness and content of the paper.

ETHICS STATEMENT

This work develops a methodological framework for matrix multiplication under missing data and does not involve the collection of new human-subject data. The empirical evaluation uses existing publicly available datasets solely for methodological benchmarking. We do not release any additional personal or identifying information. To the best of our knowledge, the proposed methodology does not introduce specific ethical risks beyond those generally associated with statistical learning and data analysis.

REPRODUCIBILITY STATEMENT

We have taken several steps to support reproducibility. The assumptions and theoretical results underlying M^3 are stated in the main paper, with complete proofs provided in the appendix. Detailed experimental settings, dataset descriptions, preprocessing procedures, missingness mechanisms, evaluation metrics, and baseline configurations are also documented in the appendix. The supplementary material includes an implementation of M^3 together with code for reproducing the experiment.

REFERENCES

Edgar Acuna and Caroline Rodriguez. The treatment of missing values and its effect on classifier accuracy. In *Classification, Clustering, and Data Mining Applications: Proceedings of the Meeting of the International Federation of Classification Societies (IFCS), Illinois Institute of Technology, Chicago, 15–18 July 2004*, pp. 639–647. Springer, 2004.

Jennifer Bryan. *gapminder: Data from Gapminder*, 2025. R package version 1.0.1.

Manja Deforth, Georg Heinze, and Ulrike Held. The performance of prognostic models depended on the choice of missing value imputation algorithm: a simulation study. *Journal of Clinical Epidemiology*, 176:111539, 2024.

Yongshi Deng and Thomas Lumley. Multiple imputation through xgboost. *Journal of Computational and Graphical Statistics*, 33(2):352–363, 2024.

Tu T Do, Anh M Vu, Tuan L Vo, Hoang Thien Ly, Thu Nguyen, Steven Hicks, Pål Halvorsen, Michael Riegler, and Binh T Nguyen. Blockwise principal component analysis for monotone missing data imputation and dimensionality reduction. In *2024 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2024.

Zhiyu Fu. Heteropca.jl: Heteroskedastic pca with missing-data support. `https://github.com/FuZhiyu/HeteroPCA.jl`, 2025. Julia implementation of HeteroPCA.

Josh Givens, Song Liu, and Henry Reeve. Score matching with missing data. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 19523–19561. PMLR, 13–19 Jul 2025.

Ken Goldberg, Theresa Roeder, Dhruv Gupta, and Chris Perkins. Eigentaste: A constant time collaborative filtering algorithm. *information retrieval*, 4(2):133–151, 2001.

Sebastian Jäger, Arndt Allhorn, and Felix Bießmann. A benchmark for data imputation methods. *Frontiers in big Data*, 4:693674, 2021.

Yehuda Koren, Steffen Rendle, and Robert Bell. Advances in collaborative filtering. *Recommender systems handbook*, pp. 91–142, 2021.

Marine Le Morvan and Gaël Varoquaux. Imputation for prediction: beware of diminishing returns. In *International Conference on Learning Representations*, volume 2025, pp. 41174–41213, 2025.

Pierre-Alexandre Mattei and Jes Frellsen. Miwae: Deep generative modelling and imputation of incomplete data sets. In *International conference on machine learning*, pp. 4413–4423. PMLR, 2019a.

Pierre-Alexandre Mattei and Jes Frellsen. Miwae: Deep generative modelling and imputation of incomplete data. `https://github.com/pamattei/miwae`, 2019b. Official implementation.

Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *The Journal of Machine Learning Research*, 11:2287–2322, 2010.

Pierre Mergny and Lenka Zdeborová. Spectral thresholds in correlated spiked models and fundamental limits of partial least squares. In *Proceedings of The 29th International Conference on Artificial Intelligence and Statistics*, volume 300 of *Proceedings of Machine Learning Research*, pp. 3925–3933. PMLR, 02–05 May 2026.

Thu Nguyen, Khoi Minh Nguyen-Duy, Duy Ho Minh Nguyen, Binh T. Nguyen, and Bruce Alan Wade. Dper: Efficient parameter estimation for randomly missing data. `https://github.com/thunguyen177/DPER`, 2021. Official implementation.

Thu Nguyen, Khoi Minh Nguyen-Duy, Duy Ho Minh Nguyen, Binh T Nguyen, and Bruce Alan Wade. Dper: Direct parameter estimation for randomly missing data. *Knowledge-Based Systems*, 240:108082, 2022.

Sunghee Oh, Dongwan D Kang, Guy N Brock, and George C Tseng. Biological impact of missing-value imputation on downstream analyses of gene expression profiles. *Bioinformatics*, 27(1):78–86, 2011.

Alessandro Palummo, Eleonora Arnone, Luca Formaggia, and Laura M Sangalli. Functional principal component analysis for incomplete space–time data. *Environmental and Ecological Statistics*, 31 (2):555–582, 2024.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Alex Rubinsteyn and Sergey Feldman. fancyimpute: An imputation library for python. URL `https://github.com/iskandr/fancyimpute`.

Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pp. 285–295, 2001.

Skipper Seabold and Josef Perktold. Statsmodels: Econometric and statistical modeling with python. In *Proceedings of the 9th Python in Science Conference*, 2010.

Tolou Shadbahr, Michael Roberts, Jan Stanczuk, Julian Gilbey, Philip Teare, Sören Dittmer, Matthew Thorpe, Ramon Viñas Torné, Evis Sala, Pietro Lió, et al. The impact of imputation quality on machine learning classifiers for datasets with missing values. *Communications medicine*, 3(1):139, 2023.

Anav Sood and Trevor Hastie. A statistical view of column subset selection. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 87(5):1382–1403, 2025.

Lena Stempfle, Anton Matsson, Newton Mwai, and Fredrik D. Johansson. Prediction models that learn to avoid missing values. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 56899–56919. PMLR, 13–19 Jul 2025.

Michael E Tipping and Christopher M Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 61(3):611–622, 1999.

Eilon Vaknin Laufer and Boaz Nadler. Rgnmr: A gauss-newton method for robust matrix completion with theoretical guarantees. *Advances in Neural Information Processing Systems*, 38:60146–60188, 2026.

Aad W Van der Vaart. *Asymptotic statistics*, volume 3. Cambridge university press, 2000.

Yogesh Verma, Ayush Bharti, and Vikas Garg. Robust simulation-based inference under missing data via neural processes. In *International Conference on Learning Representations*, volume 2025, pp. 24995–25017, 2025.

Long Tuan Vo, Van Vinh Le, Quoc Toan Ha, and Anh Quoc Nguyen. scziva: imputation method for single-cell rna-seq data with zero-inflated variational autoencoder. *BMC bioinformatics*, 27(1):92, 2026a.

Tuan L Vo, Quan Huu Do, Thu Nguyen, Pål Halvorsen, Michael A Riegler, and Binh T Nguyen. The effects of data imputation on covariance and inverse covariance matrix estimation. *IEEE Access*, 12:134688–134701, 2024.

Tuan L. Vo, Uyen Dang, Van Hua, Xuan-Hoang Nguyen, Thu Nguyen, and Bao Huynh. Dpeimputation: Parameter-based imputation via cycle-ensemble averaging for randomly missing data. `https://github.com/DangLeUyen/DPEImputation`, 2026b. Implementation of PICA.

Tuan L Vo, Uyen Dang, Van Hua, Xuan Hoang Nguyen, Thu Nguyen, and Bao Huynh. Pica: Interpretable imputation for randomly missing data. *Information Sciences*, pp. 123532, 2026c.

Tuan L Vo, Uyen Dang, Thu Nguyen, Pål Halvorsen, Michael A Riegler, and Binh T Nguyen. Dperc: Direct parameter estimation for mixed data with random missingness. In *International Conference on Multimedia Modeling*, pp. 661–675. Springer, 2026d.

Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016.

Yuling Yan, Yuxin Chen, and Jianqing Fan. Inference for heteroskedastic pca with missing data. *The Annals of Statistics*, 52(2):729–756, 2024.

Jinsung Yoon. Gain: Missing data imputation using generative adversarial nets. `https://github.com/jsyoon0823/GAIN`, 2018. Official implementation.

Jinsung Yoon, James Jordon, and Mihaela Schaar. Gain: Missing data imputation using generative adversarial nets. In *International conference on machine learning*, pp. 5689–5698. PMLR, 2018.

Fangchen Yu, Yicheng Zeng, Jianfeng Mao, and Wenye Li. Inner-product-estimation. `https://github.com/SciYu/Inner-Product-Estimation`, 2025a. Official implementation of "A Theory-Driven Approach to Inner Product Matrix Estimation for Incomplete Data: An Eigenvalue Perspective".

Fangchen Yu, Yicheng Zeng, Jianfeng Mao, and Wenye Li. A theory-driven approach to inner product matrix estimation for incomplete data: An eigenvalue perspective. In *Proceedings of the ACM on Web Conference 2025*, pp. 4077–4088, 2025b.

Shichao Zhang. Nearest neighbor selection for iteratively knn imputation. *Journal of Systems and Software*, 85(11):2541–2552, 2012.

Supplementary Material

BEYOND IMPUTATION: RETHINKING MATRIX MULTIPLICATION UNDER MISSING DATA

A PROOFS

A.1 PROOF OF PROPOSITION 3.2

Proof. Fix an output pair (i, j) and suppress the indices by writing $\mu_X = \mu_i^A$, $\mu_Y = \mu_j^B$, $v_X = v_i^A$, $v_Y = v_j^B$, and $\tau = \tau_{ij}$. Under Assumption 3.1,

$$(X_k, Y_k)^\top \sim \mathcal{N}_2\left((\mu_X, \mu_Y)^\top, \begin{pmatrix} v_X & \tau \\ \tau & v_Y \end{pmatrix}\right). \tag{17}$$

Because the observation masks are independent of (X_k, Y_k), conditioning on a missingness pattern does not alter the distribution of the underlying variables. We consider the four possible patterns separately.

If $k \in \mathcal{I}_{ij}^{11}$, both X_k and Y_k are observed and hence $X_k Y_k$ is $\sigma(O_k)$-measurable. Therefore,

$$\mathbb{E}[X_k Y_k \mid O_k] = X_k Y_k. \tag{18}$$

If $k \in \mathcal{I}_{ij}^{10}$, X_k is observed whereas Y_k is missing. The conditional distribution of a component of a bivariate Gaussian gives

$$Y_k \mid X_k \sim \mathcal{N}\left(\mu_Y + \frac{\tau}{v_X}(X_k - \mu_X),\, v_Y - \frac{\tau^2}{v_X}\right). \tag{19}$$

Since X_k is observed,

$$\mathbb{E}[X_k Y_k \mid O_k] = X_k \mathbb{E}[Y_k \mid X_k] = X_k\left[\mu_Y + \frac{\tau}{v_X}(X_k - \mu_X)\right]. \tag{20}$$

Similarly, if $k \in \mathcal{I}_{ij}^{01}$, then

$$X_k \mid Y_k \sim \mathcal{N}\left(\mu_X + \frac{\tau}{v_Y}(Y_k - \mu_Y),\, v_X - \frac{\tau^2}{v_Y}\right), \tag{21}$$

and consequently

$$\mathbb{E}[X_k Y_k \mid O_k] = Y_k \mathbb{E}[X_k \mid Y_k] = Y_k\left[\mu_X + \frac{\tau}{v_Y}(Y_k - \mu_Y)\right]. \tag{22}$$

Finally, if $k \in \mathcal{I}_{ij}^{00}$, neither variable is observed. Independence between the mask and the underlying values implies that conditioning on this pattern provides no information about (X_k, Y_k). Thus,

$$\mathbb{E}[X_k Y_k \mid O_k] = \mathbb{E}[X_k Y_k] = \mathbb{E}[X_k]\mathbb{E}[Y_k] + \mathrm{Cov}(X_k, Y_k) = \mu_X \mu_Y + \tau. \tag{23}$$

Restoring the original notation yields exactly Equation (5). □

A.2 PROOF OF PROPOSITION 3.3

Proof. Fix (i, j) and let $n_i^A = |\mathcal{O}_i^A|$ and $n_j^B = |\mathcal{O}_j^B|$. Since the observation masks are independent of the underlying values, conditional on the observed index sets, $\{A_{ik} : k \in \mathcal{O}_i^A\}$ are observations from $\mathcal{N}(\mu_i^A, v_i^A)$ and $\{B_{kj} : k \in \mathcal{O}_j^B\}$ are observations from $\mathcal{N}(\mu_j^B, v_j^B)$. The marginal log-likelihood for the A-entries, up to an additive constant, is

$$\ell_A(\mu, v) = -\frac{n_i^A}{2}\log v - \frac{1}{2v}\sum_{k \in \mathcal{O}_i^A}(A_{ik} - \mu)^2. \tag{24}$$

Differentiation with respect to μ gives

$$\frac{\partial \ell_A}{\partial \mu} = \frac{1}{v} \sum_{k \in \mathcal{O}_i^A} (A_{ik} - \mu), \tag{25}$$

so the stationary point satisfies

$$\widehat{\mu}_i^A = \frac{1}{n_i^A} \sum_{k \in \mathcal{O}_i^A} A_{ik}. \tag{26}$$

Substituting $\widehat{\mu}_i^A$ and differentiating with respect to v gives

$$\frac{\partial \ell_A}{\partial v} = -\frac{n_i^A}{2v} + \frac{1}{2v^2} \sum_{k \in \mathcal{O}_i^A} (A_{ik} - \widehat{\mu}_i^A)^2, \tag{27}$$

and hence

$$\widehat{v}_i^A = \frac{1}{n_i^A} \sum_{k \in \mathcal{O}_i^A} (A_{ik} - \widehat{\mu}_i^A)^2. \tag{28}$$

The same calculation applied to the observed entries of B yields

$$\widehat{\mu}_j^B = \frac{1}{n_j^B} \sum_{k \in \mathcal{O}_j^B} B_{kj}, \qquad \widehat{v}_j^B = \frac{1}{n_j^B} \sum_{k \in \mathcal{O}_j^B} (B_{kj} - \widehat{\mu}_j^B)^2. \tag{29}$$

It remains to characterize the plug-in estimator of τ_{ij}. For compactness, write $a = \widehat{v}_i^A$, $b = \widehat{v}_j^B$, $S_A = S_{ij}^A$, $S_B = S_{ij}^B$, $S_{AB} = S_{ij}^{AB}$, $m = m_{ij}$, and let t denote a candidate value of τ_{ij}. For each jointly observed pair, the covariance matrix is

$$\Sigma(t) = \begin{pmatrix} a & t \\ t & b \end{pmatrix}, \qquad |\Sigma(t)| = ab - t^2. \tag{30}$$

After fixing the marginal estimates, the joint Gaussian log-likelihood of the m complete pairs, up to terms independent of t, is therefore

$$\ell(t) = -\frac{m}{2} \log(ab - t^2) - \frac{bS_A + aS_B - 2tS_{AB}}{2(ab - t^2)}. \tag{31}$$

Let $D(t) = ab - t^2$. Direct differentiation gives

$$\ell'(t) = -\frac{mt^3 - S_{AB}t^2 + (bS_A + aS_B - mab)t - abS_{AB}}{(ab - t^2)^2}. \tag{32}$$

For every interior feasible point $|t| < (1-\epsilon)\sqrt{ab}$, we have $ab - t^2 > 0$. Hence an interior maximizer must satisfy $\ell'(t) = 0$, which is equivalent to

$$mt^3 - S_{AB}t^2 + (bS_A + aS_B - mab)t - abS_{AB} = 0. \tag{33}$$

Restoring the original notation gives

$$m_{ij}t^3 - S_{ij}^{AB}t^2 + \left(\widehat{v}_j^B S_{ij}^A + \widehat{v}_i^A S_{ij}^B - m_{ij}\widehat{v}_i^A \widehat{v}_j^B\right) t - \widehat{v}_i^A \widehat{v}_j^B S_{ij}^{AB} = 0, \tag{34}$$

which is precisely Equation (8). Therefore every interior maximizer of the plug-in likelihood must be a feasible real root of $p_{ij}(t)$. □

A.3 Proof of Theorem 4.2

Proof. Fix (i, j) and define $Z_k = X_k Y_k$, so that $C_{ij} = \sum_{k=1}^d Z_k$. Since $\mathbb{E}[Z_k^2] = \mathbb{E}[X_k^2 Y_k^2] < \infty$, we have

$$\mathbb{E}[C_{ij}^2] = \mathbb{E}\left[\left(\sum_{k=1}^d Z_k\right)^2\right] \leq d \sum_{k=1}^d \mathbb{E}[Z_k^2] < \infty, \tag{35}$$

and thus $C_{ij} \in L^2$.

Let $\mathcal{F}_{ij} = \sigma(O_1, \ldots, O_d)$. By Assumption 4.1, the tuples $(X_k, Y_k, M^A_{ik}, M^B_{kj})$ are independent across k. Consequently, (Z_k, O_k) is independent of $\{O_\ell : \ell \neq k\}$. It follows that conditioning additionally on the observations from other coordinates does not change the conditional expectation of Z_k:

$$\mathbb{E}[Z_k \mid \mathcal{F}_{ij}] = \mathbb{E}[Z_k \mid O_1, \ldots, O_d] = \mathbb{E}[Z_k \mid O_k]. \tag{36}$$

Using linearity of conditional expectation,

$$\mathbb{E}[C_{ij} \mid \mathcal{F}_{ij}] = \sum_{k=1}^{d} \mathbb{E}[Z_k \mid \mathcal{F}_{ij}] = \sum_{k=1}^{d} \mathbb{E}[Z_k \mid O_k] = \widetilde{C}_{ij}. \tag{37}$$

This proves Equation (13).

To establish optimality, let T be any square-integrable $\mathcal{F}_{ij}$-measurable estimator. Write

$$C_{ij} - T = (C_{ij} - \widetilde{C}_{ij}) + (\widetilde{C}_{ij} - T). \tag{38}$$

Squaring and taking expectations gives

$$\mathbb{E}[(C_{ij} - T)^2] = \mathbb{E}[(C_{ij} - \widetilde{C}_{ij})^2] + \mathbb{E}[(\widetilde{C}_{ij} - T)^2] + 2\mathbb{E}[(C_{ij} - \widetilde{C}_{ij})(\widetilde{C}_{ij} - T)]. \tag{39}$$

Since both $\widetilde{C}_{ij}$ and T are $\mathcal{F}_{ij}$-measurable and $\widetilde{C}_{ij} = \mathbb{E}[C_{ij} \mid \mathcal{F}_{ij}]$, the cross term vanishes:

$$\mathbb{E}[(C_{ij} - \widetilde{C}_{ij})(\widetilde{C}_{ij} - T)] = \mathbb{E}\left[\mathbb{E}[(C_{ij} - \widetilde{C}_{ij})(\widetilde{C}_{ij} - T) \mid \mathcal{F}_{ij}]\right] \tag{40}$$

$$= \mathbb{E}\left[(\widetilde{C}_{ij} - T)\mathbb{E}[C_{ij} - \widetilde{C}_{ij} \mid \mathcal{F}_{ij}]\right] = 0. \tag{41}$$

Therefore,

$$\mathbb{E}[(C_{ij} - T)^2] = \mathbb{E}[(C_{ij} - \widetilde{C}_{ij})^2] + \mathbb{E}[(\widetilde{C}_{ij} - T)^2] \geq \mathbb{E}[(C_{ij} - \widetilde{C}_{ij})^2], \tag{42}$$

which proves Equation (14). Equality holds if and only if $\mathbb{E}[(\widetilde{C}_{ij} - T)^2] = 0$, equivalently $T = \widetilde{C}_{ij}$ almost surely. Hence $\widetilde{C}_{ij}$ is the unique, up to almost-sure equality, minimum-MSE estimator among all square-integrable estimators measurable with respect to $\mathcal{F}_{ij}$. □

A.4 Proof of Theorem 4.3

Proof. Fix a coordinate k and write $Z = X_k Y_k$, $\mu_X = \mu^A_i$, $\mu_Y = \mu^B_j$, $v_X = v^A_i$, $v_Y = v^B_j$, and $\tau = \tau_{ij}$. For a pattern $(a, b) \in \{0, 1\}^2$, let $E_{ab} = \{M^A_{ik} = a, M^B_{kj} = b\}$. By Assumption 3.1, the masks are independent of (X_k, Y_k), so conditioning on E_{ab} does not alter their Gaussian distribution.

Within each pattern, $\psi^\star(O_k) = \mathbb{E}[Z \mid O_k]$. Hence, for any $\sigma(O_k)$-measurable rule ψ, the conditional-expectation orthogonality identity gives

$$\mathbb{E}[(Z - \psi)^2 \mid E_{ab}] = \mathbb{E}[(Z - \psi^\star)^2 \mid E_{ab}] + \mathbb{E}[(\psi^\star - \psi)^2 \mid E_{ab}]. \tag{43}$$

Indeed, the cross term vanishes because

$$\mathbb{E}[(Z - \psi^\star)(\psi^\star - \psi) \mid E_{ab}] = \mathbb{E}\left[(\psi^\star - \psi)\mathbb{E}[Z - \psi^\star \mid O_k] \mid E_{ab}\right] = 0. \tag{44}$$

Taking $\psi = \psi_0$ therefore yields

$$\mathrm{MSE}_{ab}(\psi_0) - \mathrm{MSE}_{ab}(\psi^\star) = \mathbb{E}[(\psi^\star - \psi_0)^2 \mid E_{ab}]. \tag{45}$$

For $(a, b) = (1, 1)$, both rules equal the exactly observed product XY, and therefore the difference is zero.

For $(a, b) = (1, 0)$, Proposition 3.2 gives

$$\psi^\star = X\left[\mu_Y + \frac{\tau}{v_X}(X - \mu_X)\right], \qquad \psi_0 = X\mu_Y, \tag{46}$$

so

$$\psi^\star - \psi_0 = \frac{\tau}{v_X} X(X - \mu_X). \tag{47}$$

Let $U = X - \mu_X$. Since $U \sim \mathcal{N}(0, v_X)$, its Gaussian moments satisfy $\mathbb{E}[U^2] = v_X$, $\mathbb{E}[U^3] = 0$, and $\mathbb{E}[U^4] = 3v_X^2$. Consequently,

$$\mathbb{E}[X^2(X-\mu_X)^2] = \mathbb{E}[(\mu_X + U)^2 U^2] \tag{48}$$

$$= \mu_X^2 \mathbb{E}[U^2] + 2\mu_X \mathbb{E}[U^3] + \mathbb{E}[U^4] = \mu_X^2 v_X + 3v_X^2. \tag{49}$$

Using Equation (45),

$$\mathrm{MSE}_{10}(\psi_0) - \mathrm{MSE}_{10}(\psi^\star) = \frac{\tau^2}{v_X^2}\left(\mu_X^2 v_X + 3v_X^2\right) = \tau^2\left(3 + \frac{\mu_X^2}{v_X}\right). \tag{50}$$

For $(a, b) = (0, 1)$, the same argument with X and Y interchanged gives

$$\psi^\star - \psi_0 = \frac{\tau}{v_Y} Y(Y - \mu_Y). \tag{51}$$

Writing $V = Y - \mu_Y \sim \mathcal{N}(0, v_Y)$ and using $\mathbb{E}[V^2] = v_Y$, $\mathbb{E}[V^3] = 0$, and $\mathbb{E}[V^4] = 3v_Y^2$ yields

$$\mathrm{MSE}_{01}(\psi_0) - \mathrm{MSE}_{01}(\psi^\star) = \tau^2\left(3 + \frac{\mu_Y^2}{v_Y}\right). \tag{52}$$

Finally, for $(a, b) = (0, 0)$,

$$\psi^\star = \mu_X \mu_Y + \tau, \qquad \psi_0 = \mu_X \mu_Y, \tag{53}$$

and therefore

$$\mathrm{MSE}_{00}(\psi_0) - \mathrm{MSE}_{00}(\psi^\star) = \mathbb{E}[(\psi^\star - \psi_0)^2 \mid E_{00}] = \tau^2. \tag{54}$$

Restoring the original parameters $\mu_i^A, \mu_j^B, v_i^A, v_j^B, \tau_{ij}$ proves Equation (15). Since $v_i^A > 0$ and $v_j^B > 0$, every gain associated with an incomplete pattern is strictly positive whenever $\tau_{ij} \neq 0$. In the centered case $\mu_i^A = \mu_j^B = 0$, the three gains reduce to $3\tau_{ij}^2$, $3\tau_{ij}^2$, and τ_{ij}^2, respectively. □

B Experimental Details

B.1 Common Experimental Protocol

Dataset preparation. All experiments start from complete reference data and subsequently introduce artificial missingness under the missing-completely-at-random (MCAR) mechanism (Vo et al., 2026c). For each dataset, missing rate, and random seed, the missingness mask is generated once and shared across all competing methods. Thus, all methods are evaluated on exactly the same observed entries within each run. For the direct-multiplication, self-product, and cross-view benchmarks, we consider missing rates of 20%, 30%, 50%, and 70%. The controlled scaling experiments use 50% MCAR missingness. The recommendation setting is described separately in Appendix B.7.

Evaluated methods. Imputation methods first reconstruct the incomplete operand or data matrix and then apply the ordinary matrix operation required by the task. In contrast, M^3 operates directly on the incomplete operands and returns the required matrix product without constructing completed versions of the inputs. Task-specific baselines are applied directly according to their original formulations.

B.2 Implementation of M^3

The same M^3 configuration is used across all applicable experiments, datasets, missing rates, and random seeds. The implementation follows Algorithm 1. Marginal means and variances are estimated using all individually observed entries in the corresponding row of $\mathbf{A}$ or column of $\mathbf{B}$, whereas the dependence parameter τ_{ij} is estimated only from jointly observed coordinates.

For each output pair (i, j), the feasible real roots of the cubic equation in Equation (8) are first identified. The plug-in log-likelihood in Equation (10) is then evaluated at all feasible interior roots and at the two covariance boundaries. The candidate attaining the largest likelihood is retained as $\widehat{\tau}_{ij}^{\mathrm{PL}}$. When no jointly observed coordinate is available, i.e., $m_{ij} = 0$, we set $\widehat{\tau}_{ij}^{\mathrm{PL}} = 0$.

The resulting dependence estimate is shrunk according to Equation (11) and clipped according to Equation (12). Throughout all experiments, we use the fixed configuration $\lambda_\tau = 10, \epsilon = 10^{-5}$. No training stage, validation set, or dataset-specific hyperparameter search is used for M^3.

Table 1: Fixed configurations used for the competing methods.

Method	Configuration
Mean	Mean imputation using all observed entries of the corresponding marginal vector.
KNNI	5 neighbors with distance weighting and the default `nan_euclidean` distance. Implemented using scikit-learn's `KNNImputer` (Pedregosa et al., 2011).
SoftImpute	Maximum 100 iterations and convergence tolerance 10^{-5}. Implemented via fancyimpute Rubinsteyn & Feldman.
GAIN	Hyperparameters follow the authors' official implementation (Yoon, 2018).
PICA	The window size is suggested by authors $\min(p, 7)$ (Vo et al., 2026c). Implemented via the authors' official implementation (Vo et al., 2026b).
XGBI	Chained `IterativeImputer` with an XGBoost regressor using 100 trees, maximum depth 3, learning rate 0.05, subsampling rate 0.9, column-subsampling rate 0.9, and 5 imputation rounds (Deng & Lumley, 2024).
MIWAE	Hyperparameters follow the configuration of the authors' official implementation (Mattei & Frellsen, 2019b).
Pairwise-PSD	Pairwise covariance estimation from jointly observed entries followed by projection onto the positive-semidefinite cone; no tuned hyperparameters.
DPER	No tuned hyperparameters. Implemented via the authors' official implementation (Nguyen et al., 2021).
EC-IID	No tuned hyperparameters. Implemented via the authors' official implementation (Yu et al., 2025a).
HeteroPCA	Hyperparameters follow the configuration of the HeteroPCA.jl implementation (Fu, 2025).
PPCA-EM	Our implementation of probabilistic PCA with expectation–maximization (Tipping & Bishop, 1999), using at most 100 EM iterations, convergence tolerance 10^{-5}, and variance floor 10^{-6}.
CoRated-ItemKNN	Co-rated cosine similarity on bias-corrected residuals, with no similarity shrinkage and the top 10 neighbors retained by absolute similarity (Sarwar et al., 2001).
Shrunk-ItemKNN	Co-rated cosine similarity on bias-corrected residuals with overlap shrinkage $\frac{n_{ij}}{n_{ij}+50}$, retaining the top 10 neighbors by absolute similarity (Koren et al., 2021).
M^3	$\lambda_\tau = 10, \epsilon = 10^{-5}$.

B.3 Baseline Configurations

All competing methods use a single fixed configuration across all applicable datasets, missing rates, and random seeds. No configuration is selected separately for individual datasets or missingness levels. Table 1 summarizes the implementations used in the experiments.

B.4 Synthetic Data Generation

For each random seed, row- and column-specific loadings are sampled as

$$a_i \sim \mathrm{Unif}(0.4, 1)\sqrt{\rho}, \qquad b_j \sim \mathrm{Unif}(0.4, 1)\sqrt{\rho}, \tag{55}$$

with $\rho = 0.75$. Row and column means are independently sampled as $\mu_i^A, \mu_j^B \sim \mathcal{N}(0, 0.7^2)$. The same latent coordinate is shared between the corresponding rows of $\mathbf{A}$ and columns of $\mathbf{B}$, inducing nonzero cross-factor dependence. After generating centered variables $\widetilde{A}_{ik}$ and $\widetilde{B}_{kj}$ as described below, the final operands are

$$A_{ik} = \mu_i^A + \widetilde{A}_{ik}, \qquad B_{kj} = \mu_j^B + \widetilde{B}_{kj}. \tag{56}$$

The benchmark considers five data-generating distributions.

1. **Gaussian.** For each shared coordinate k, let $Z_k, \varepsilon_{ik}^A, \varepsilon_{kj}^B \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1)$. We generate

$$\widetilde{A}_{ik} = a_i Z_k + \sqrt{1 - a_i^2}\, \varepsilon_{ik}^A, \qquad \widetilde{B}_{kj} = b_j Z_k + \sqrt{1 - b_j^2}\, \varepsilon_{kj}^B. \tag{57}$$

   Thus, each centered marginal has unit variance and $\mathrm{Cov}(\widetilde{A}_{ik}, \widetilde{B}_{kj}) = a_i b_j$.

2. **Student-t_5.** We first construct the same correlated Gaussian variables as above and then apply a common random scale

$$S_k = \sqrt{\frac{3}{Q_k}}, \qquad Q_k \sim \chi^2_5, \tag{58}$$

giving

$$\widetilde{A}_{ik} = S_k \left(a_i Z_k + \sqrt{1 - a_i^2}\, \varepsilon^A_{ik} \right), \qquad \widetilde{B}_{kj} = S_k \left(b_j Z_k + \sqrt{1 - b_j^2}\, \varepsilon^B_{kj} \right). \tag{59}$$

The factor $3 = 5 - 2$ normalizes the marginal variances to one.

3. **Shifted Gaussian mixture.** For each coordinate, we sample $C_k \in \{-1, 1\}$ with $\mathbb{P}(C_k = -1) = 0.55$ and $\mathbb{P}(C_k = 1) = 0.45$, and define

$$Z^{\mathrm{mix}}_k = Z_k + 1.5 C_k. \tag{60}$$

The preliminary variables are then generated as

$$A^{(0)}_{ik} = a_i Z^{\mathrm{mix}}_k + \sqrt{1 - a_i^2}\, \varepsilon^A_{ik}, \qquad B^{(0)}_{kj} = b_j Z^{\mathrm{mix}}_k + \sqrt{1 - b_j^2}\, \varepsilon^B_{kj}. \tag{61}$$

Each generated row of $\mathbf{A}$ and column of $\mathbf{B}$ is subsequently standardized over the shared coordinate dimension to zero empirical mean and unit empirical variance before adding μ^A_i and μ^B_j.

4. **Skewed.** Starting from $Z_k \sim \mathcal{N}(0, 1)$, we construct a positively skewed latent variable by

$$Z^{\mathrm{skew}}_k = \mathrm{Std}[\exp(0.7 Z_k)], \tag{62}$$

where $\mathrm{Std}[\cdot]$ denotes empirical standardization to zero mean and unit variance over the d shared coordinates. We then generate

$$\widetilde{A}_{ik} = a_i Z^{\mathrm{skew}}_k + \sqrt{1 - a_i^2}\, \varepsilon^A_{ik}, \qquad \widetilde{B}_{kj} = b_j Z^{\mathrm{skew}}_k + \sqrt{1 - b_j^2}\, \varepsilon^B_{kj}. \tag{63}$$

5. **Laplace.** The shared latent variable and independent noise terms are sampled from centered Laplace distributions and rescaled to unit variance:

$$L_k, \eta^A_{ik}, \eta^B_{kj} \overset{\text{i.i.d.}}{\sim} \frac{1}{\sqrt{2}} \mathrm{Laplace}(0, 1). \tag{64}$$

The centered variables are

$$\widetilde{A}_{ik} = a_i L_k + \sqrt{1 - a_i^2}\, \eta^A_{ik}, \qquad \widetilde{B}_{kj} = b_j L_k + \sqrt{1 - b_j^2}\, \eta^B_{kj}. \tag{65}$$

For each generator, the complete matrices are generated first, and the two operands are then masked independently under MCAR at rates of 20%, 30%, 50%, and 70%. Within each random seed and missing rate, all competing methods receive the same incomplete operands.

B.5 Self-Product Tasks

For each dataset, the complete numerical matrix is denoted by $\mathbf{X} \in \mathbb{R}^{n \times p}$. The centered matrix is $\mathbf{X}_c = \mathbf{X} - \mathbf{1}_n \bar{\mathbf{x}}^\top$, and the oracle second-order target is the centered Gram matrix $\mathbf{G} = \mathbf{X}_c^\top \mathbf{X}_c$.

For Gram recovery, performance is measured by

$$E_{\mathrm{Gram}} = \frac{\|\widehat{\mathbf{G}} - \mathbf{G}\|_F}{\|\mathbf{G}\|_F}. \tag{66}$$

For PCA, let $\lambda_1 \geq \lambda_2$ denote the two leading eigenvalues obtained from the complete-data second-order matrix, and let $\widehat{\lambda}_1, \widehat{\lambda}_2$ denote their estimates. We report

$$E_{\mathrm{PCA}} = \frac{\|(\widehat{\lambda}_1, \widehat{\lambda}_2)^\top - (\lambda_1, \lambda_2)^\top\|_2}{\|(\lambda_1, \lambda_2)^\top\|_2}. \tag{67}$$

B.6 Cross-View PLS Setup

The PLS benchmark evaluates recovery of a genuine cross-view product. For paired views $\mathbf{X} \in \mathbb{R}^{n\times p}$ and $\mathbf{Y} \in \mathbb{R}^{n\times q}$, the complete-data target is

$$\mathbf{S}_{XY} = \frac{1}{n}\mathbf{X}_c^\top \mathbf{Y}_c. \tag{68}$$

The benchmark uses fixed two-view feature partitions. For Diamonds, $\mathbf{X}$ contains carat and the physical dimensions (x, y, z), whereas $\mathbf{Y}$ contains depth, table, and price. For Breast Cancer, $\mathbf{X}$ contains the ten "mean" diagnostic measurements and $\mathbf{Y}$ contains the ten "worst" diagnostic measurements. Gapminder uses year and population as $\mathbf{X}$, and life expectancy and GDP per capita as $\mathbf{Y}$. Seals uses latitude and longitude as $\mathbf{X}$, and longitudinal and latitudinal displacement as $\mathbf{Y}$.

The features in each view are standardized before evaluating the cross-view product. The two views are then independently masked at the same missing rate, and identical masks are used across competing methods within each random seed.

Imputation methods reconstruct the concatenated incomplete matrix $[\mathbf{X}, \mathbf{Y}]$ before extracting the two views and forming $\widehat{\mathbf{S}}_{XY}$. In contrast, M^3 directly estimates $\mathbf{X}_c^\top \mathbf{Y}_c$ from the incomplete views, which is then divided by n to obtain $\widehat{\mathbf{S}}_{XY}$.

Let $\sigma_1 \geq \sigma_2$ denote the two leading singular values of the complete cross-product and $\widehat{\sigma}_1, \widehat{\sigma}_2$ their estimates. PLS spectral recovery is measured by

$$E_{\mathrm{PLS}} = \frac{\|(\widehat{\sigma}_1, \widehat{\sigma}_2)^\top - (\sigma_1, \sigma_2)^\top\|_2}{\|(\sigma_1, \sigma_2)^\top\|_2}. \tag{69}$$

B.7 Recommendation Setup

We evaluate the use of the estimated product in a downstream item-ranking problem using the Jester explicit-rating dataset. We select a 1000×100 user–item rating matrix. For each user, 40% of the available ratings are held out for evaluation. Among the remaining training ratings, 50% are independently masked to create the incomplete input supplied to the competing methods.

Imputation methods first reconstruct the incomplete user–item matrix and then form the item–item product. In contrast, M^3 estimates the corresponding item–item product directly from the incomplete training matrix. CoRated-ItemKNN and Shrunk-ItemKNN are included as task-specific item-based collaborative-filtering baselines.

All methods subsequently use exactly the same bias-corrected ItemKNN ranking procedure. Let μ, b_u, and b_j denote the global, user, and item biases estimated from the observed training ratings, and let $z_{u\ell} = R_{u\ell} - (\mu + b_u + b_\ell)$ be the corresponding observed residual. The estimated item–item product is converted to cosine similarities, and for each candidate item j only the 10 neighbors with largest absolute similarity are retained. The ranking score is

$$s_{uj} = \mu + b_u + b_j + \frac{\sum_{\ell \in O_u \cap \mathcal{N}_j} z_{u\ell}\, \widehat{w}_{j\ell}}{\sum_{\ell \in O_u \cap \mathcal{N}_j} |\widehat{w}_{j\ell}|}, \tag{70}$$

where O_u is the set of observed training items for user u, $\mathcal{N}_j$ is the retained neighbor set of item j, and $\widehat{w}_{j\ell}$ is the estimated cosine similarity between items j and ℓ. The adjustment term is set to zero when its denominator is zero.

Ranking is evaluated over the held-out items of each user. For Recall and MAP, the top 25% highest-rated held-out items for that user are treated as relevant. For NDCG, we retain graded relevance by rescaling each held-out Jester rating $r_{uj} \in [-10, 10]$ to $(r_{uj} + 10)/20$. We report NDCG@10, Recall@10, and MAP@10 using the same held-out entries and candidate sets for every method. Higher values indicate better recommendation performance.

B.8 Runtime Measurement

Runtime is measured as the end-to-end wall-clock time required to obtain the product estimate from the already-masked input matrices. For imputation-based methods, the reported time includes

Table 2: Direct matrix multiplication under five data distributions and MCAR missingness rates. Relative Frobenius error is reported; lower is better. Here, mr denotes the missing rate, and standard deviations are shown as subscripts with leading zeros omitted. Bold indicates the lowest mean error.

Distribution	mr	Mean	KNNI	SoftImpute	GAIN	PICA	XGBI	MIWAE	M^3
Gaussian	20%	$0.259_{.039}$	$0.106_{.019}$	$0.141_{.014}$	$0.103_{.015}$	$0.091_{.016}$	$0.119_{.013}$	$0.130_{.017}$	$\mathbf{0.086}_{.014}$
	30%	$0.365_{.047}$	$0.173_{.020}$	$0.219_{.015}$	$0.155_{.017}$	$0.128_{.014}$	$0.169_{.005}$	$0.192_{.017}$	$\mathbf{0.116}_{.011}$
	50%	$0.525_{.066}$	$0.370_{.044}$	$0.414_{.016}$	$0.333_{.049}$	$0.248_{.029}$	$0.281_{.023}$	$0.361_{.022}$	$\mathbf{0.188}_{.017}$
	70%	$0.642_{.082}$	$0.543_{.074}$	$0.688_{.026}$	$1.273_{.180}$	$0.460_{.041}$	$0.496_{.022}$	$0.593_{.030}$	$\mathbf{0.346}_{.047}$
Student-t_5	20%	$0.255_{.033}$	$0.135_{.022}$	$0.159_{.021}$	$0.122_{.040}$	$0.132_{.034}$	$0.154_{.030}$	$0.155_{.032}$	$\mathbf{0.113}_{.032}$
	30%	$0.343_{.059}$	$0.191_{.037}$	$0.224_{.024}$	$0.166_{.058}$	$0.155_{.036}$	$0.192_{.044}$	$0.207_{.056}$	$\mathbf{0.150}_{.038}$
	50%	$0.511_{.086}$	$0.397_{.067}$	$0.441_{.016}$	$0.401_{.086}$	$0.300_{.060}$	$0.336_{.069}$	$0.382_{.086}$	$\mathbf{0.246}_{.054}$
	70%	$0.615_{.099}$	$0.535_{.086}$	$0.674_{.046}$	$1.695_{.339}$	$1.142_{1.073}$	$0.487_{.076}$	$0.561_{.060}$	$\mathbf{0.373}_{.074}$
Mixture	20%	$0.308_{.027}$	$0.059_{.010}$	$0.104_{.011}$	$0.057_{.007}$	$\mathbf{0.047}_{.006}$	$0.100_{.003}$	$0.102_{.010}$	$0.056_{.008}$
	30%	$0.428_{.032}$	$0.091_{.009}$	$0.168_{.013}$	$0.092_{.007}$	$0.081_{.010}$	$0.154_{.007}$	$0.164_{.007}$	$\mathbf{0.078}_{.007}$
	50%	$0.635_{.058}$	$0.257_{.042}$	$0.328_{.021}$	$0.205_{.037}$	$0.166_{.028}$	$0.280_{.016}$	$0.316_{.027}$	$\mathbf{0.148}_{.018}$
	70%	$0.765_{.069}$	$0.434_{.050}$	$0.568_{.029}$	$0.626_{.120}$	$0.360_{.066}$	$0.430_{.051}$	$0.549_{.093}$	$\mathbf{0.293}_{.030}$
Skewed	20%	$0.272_{.054}$	$0.113_{.028}$	$0.146_{.015}$	$\mathbf{0.100}_{.023}$	$0.123_{.047}$	$0.163_{.055}$	$0.214_{.068}$	$0.104_{.030}$
	30%	$0.371_{.064}$	$0.178_{.038}$	$0.224_{.024}$	$0.153_{.027}$	$0.163_{.057}$	$0.218_{.062}$	$0.286_{.081}$	$\mathbf{0.138}_{.030}$
	50%	$0.535_{.085}$	$0.384_{.067}$	$0.431_{.024}$	$0.333_{.072}$	$0.269_{.059}$	$0.331_{.055}$	$0.455_{.069}$	$\mathbf{0.218}_{.040}$
	70%	$0.648_{.097}$	$0.563_{.094}$	$0.708_{.030}$	$1.101_{.113}$	$0.517_{.114}$	$0.534_{.073}$	$0.636_{.067}$	$\mathbf{0.385}_{.069}$
Laplace	20%	$0.254_{.040}$	$0.112_{.021}$	$0.143_{.018}$	$0.101_{.016}$	$0.104_{.029}$	$0.135_{.022}$	$0.146_{.022}$	$\mathbf{0.095}_{.014}$
	30%	$0.349_{.046}$	$0.171_{.029}$	$0.222_{.012}$	$0.145_{.025}$	$0.140_{.022}$	$0.180_{.028}$	$0.203_{.027}$	$\mathbf{0.127}_{.018}$
	50%	$0.515_{.059}$	$0.358_{.036}$	$0.426_{.029}$	$0.341_{.057}$	$0.254_{.045}$	$0.302_{.023}$	$0.368_{.023}$	$\mathbf{0.212}_{.029}$
	70%	$0.626_{.080}$	$0.523_{.073}$	$0.689_{.036}$	$1.683_{.356}$	$0.474_{.064}$	$0.493_{.063}$	$0.608_{.057}$	$\mathbf{0.352}_{.054}$

imputation of both operands followed by ordinary matrix multiplication. For M^3, it includes marginal-moment estimation, pair-local dependence estimation, covariance stabilization, and construction of the final product estimate.

Data generation, missing-mask construction, complete-data target computation, evaluation metrics, and result serialization are excluded from the timing. Wall-clock time is measured using Python's `time.perf_counter()`.

C Complete Numerical Results

C.1 Table 2: Direct Matrix Multiplication

Across the 20 distribution–missing-rate settings in Table 2, M^3 achieves the lowest mean relative error in 18 cases. The only exceptions occur at the lowest missing rate (20%): under the mixture distribution, PICA obtains 0.047 compared with 0.056 for M^3, while under the skewed distribution, GAIN obtains 0.100 compared with 0.104 for M^3. As the missing rate increases, the advantage of direct product estimation becomes more pronounced. At 70% missingness, for example, M^3 achieves errors of 0.346, 0.373, 0.293, 0.385, and 0.352 for Gaussian, Student-t_5, mixture, skewed, and Laplace data, respectively, corresponding to approximately 19–26% lower error than the strongest competing method in each setting. Overall, M^3 attains the lowest error for every distribution at 30%, 50%, and 70% missingness, including all non-Gaussian settings.

C.2 Table 3: Gram Recovery

Across the 20 dataset–missing-rate settings in Table 3, M^3 achieves the lowest or tied-lowest mean error in 15 cases, including all settings on ANES96 and Fair. The advantage over imputation-based methods is particularly pronounced at high missing rates: at 70% missingness, M^3 obtains errors of 0.157 on ANES96 and 0.199 on Fair, compared with 0.446 and 0.512, respectively, for the strongest imputation-based competitors. Among the task-specific methods, DPER is the strongest competitor and performs particularly well on Economics, where it attains errors of 0.039, 0.069, and 0.087 at 30%, 50%, and 70% missingness, compared with 0.044, 0.082, and 0.129 for M^3. EC-IID also slightly outperforms M^3 on RandHIE at 70% missingness (0.274 versus 0.280). Overall,

Table 3: Centered-Gram recovery on five real datasets under MCAR missingness. Relative Frobenius error is reported; lower is better. Here, mr denotes the missing rate, and standard deviations are shown as subscripts with leading zeros omitted. Bold indicates the lowest mean error.

Dataset	mr	Mean	KNNI	SoftImpute	GAIN	PICA	XGBI	MIWAE	Pairwise-PSD	EC-IID	HeteroPCA	DPER	M^3
ANES96	20%	$0.227_{.008}$	$0.135_{.012}$	$0.205_{.007}$	$0.192_{.013}$	$0.189_{.005}$	$0.191_{.007}$	$0.210_{.022}$	$0.054_{.012}$	$0.073_{.019}$	$0.743_{.012}$	$\mathbf{0.053}_{.012}$	$\mathbf{0.053}_{.013}$
	30%	$0.324_{.013}$	$0.213_{.020}$	$0.299_{.013}$	$0.287_{.020}$	$0.274_{.019}$	$0.267_{.014}$	$0.310_{.007}$	$0.079_{.012}$	$0.099_{.016}$	$0.739_{.028}$	$0.077_{.011}$	$\mathbf{0.076}_{.010}$
	50%	$0.486_{.015}$	$0.329_{.021}$	$0.453_{.015}$	$0.443_{.019}$	$0.438_{.023}$	$0.401_{.016}$	$0.486_{.020}$	$0.117_{.033}$	$0.156_{.019}$	$0.753_{.028}$	$0.115_{.031}$	$\mathbf{0.111}_{.031}$
	70%	$0.593_{.009}$	$0.381_{.028}$	$0.559_{.016}$	$0.546_{.023}$	$0.521_{.009}$	$0.446_{.013}$	$0.581_{.007}$	$0.169_{.044}$	$0.207_{.015}$	$0.769_{.041}$	$0.167_{.042}$	$\mathbf{0.157}_{.033}$
Fair	20%	$0.276_{.008}$	$0.133_{.010}$	$0.139_{.007}$	$0.207_{.016}$	$0.133_{.011}$	$0.148_{.009}$	$0.163_{.016}$	$0.051_{.008}$	$0.131_{.010}$	$0.533_{.012}$	$\mathbf{0.048}_{.006}$	$\mathbf{0.048}_{.006}$
	30%	$0.391_{.012}$	$0.237_{.019}$	$0.208_{.011}$	$0.308_{.022}$	$0.199_{.010}$	$0.213_{.007}$	$0.243_{.009}$	$0.073_{.015}$	$0.185_{.016}$	$0.533_{.013}$	$0.069_{.013}$	$\mathbf{0.068}_{.013}$
	50%	$0.602_{.010}$	$0.434_{.016}$	$0.369_{.012}$	$0.518_{.032}$	$0.340_{.014}$	$0.347_{.019}$	$0.423_{.018}$	$0.114_{.019}$	$0.313_{.013}$	$0.542_{.018}$	$0.108_{.019}$	$\mathbf{0.107}_{.018}$
	70%	$0.762_{.011}$	$0.564_{.030}$	$0.555_{.018}$	$0.698_{.050}$	$0.520_{.024}$	$0.512_{.013}$	$0.653_{.064}$	$0.216_{.027}$	$0.433_{.022}$	$0.587_{.023}$	$0.209_{.023}$	$\mathbf{0.199}_{.026}$
RandHIE	20%	$0.242_{.022}$	$0.153_{.026}$	$0.172_{.024}$	$0.212_{.018}$	$0.158_{.011}$	$0.141_{.016}$	$0.191_{.019}$	$0.082_{.019}$	$0.110_{.013}$	$0.716_{.015}$	$\mathbf{0.081}_{.019}$	$\mathbf{0.081}_{.019}$
	30%	$0.350_{.008}$	$0.246_{.013}$	$0.231_{.011}$	$0.309_{.020}$	$0.266_{.010}$	$0.233_{.008}$	$0.308_{.026}$	$0.112_{.017}$	$0.149_{.010}$	$0.718_{.012}$	$0.108_{.014}$	$\mathbf{0.107}_{.014}$
	50%	$0.558_{.014}$	$0.423_{.014}$	$0.379_{.020}$	$0.503_{.024}$	$0.452_{.027}$	$0.359_{.037}$	$0.511_{.011}$	$0.171_{.017}$	$0.231_{.021}$	$0.722_{.017}$	$0.163_{.022}$	$\mathbf{0.159}_{.022}$
	70%	$0.729_{.014}$	$0.559_{.024}$	$0.576_{.023}$	$0.679_{.023}$	$0.625_{.013}$	$0.610_{.132}$	$0.685_{.030}$	$\mathbf{0.274}_{.041}$	$0.320_{.020}$	$0.760_{.039}$	$0.309_{.064}$	$0.280_{.057}$
Economics	20%	$0.317_{.029}$	$0.047_{.012}$	$0.098_{.015}$	$0.139_{.021}$	$0.044_{.010}$	$0.104_{.014}$	$0.160_{.026}$	$0.050_{.013}$	$0.170_{.018}$	$0.106_{.025}$	$\mathbf{0.035}_{.010}$	$\mathbf{0.035}_{.009}$
	30%	$0.456_{.016}$	$0.088_{.014}$	$0.148_{.012}$	$0.211_{.031}$	$0.047_{.009}$	$0.146_{.013}$	$0.243_{.026}$	$0.057_{.008}$	$0.259_{.013}$	$0.120_{.032}$	$\mathbf{0.039}_{.011}$	$0.044_{.012}$
	50%	$0.654_{.023}$	$0.174_{.020}$	$0.256_{.025}$	$0.363_{.066}$	$0.113_{.034}$	$0.254_{.015}$	$0.429_{.006}$	$0.112_{.019}$	$0.408_{.028}$	$0.180_{.055}$	$\mathbf{0.069}_{.016}$	$0.082_{.023}$
	70%	$0.774_{.009}$	$0.211_{.024}$	$0.327_{.016}$	$0.506_{.098}$	$0.128_{.015}$	$0.293_{.018}$	$0.553_{.045}$	$0.149_{.046}$	$0.570_{.013}$	$0.317_{.056}$	$\mathbf{0.087}_{.022}$	$0.129_{.019}$
Texas Housing	20%	$0.311_{.029}$	$0.192_{.035}$	$0.148_{.008}$	$0.293_{.028}$	$0.100_{.005}$	$0.146_{.019}$	$0.262_{.039}$	$0.080_{.034}$	$0.169_{.035}$	$0.442_{.018}$	$0.060_{.021}$	$\mathbf{0.057}_{.015}$
	30%	$0.419_{.031}$	$0.302_{.035}$	$0.213_{.019}$	$0.391_{.031}$	$0.148_{.013}$	$0.204_{.023}$	$0.345_{.026}$	$0.102_{.034}$	$0.220_{.034}$	$0.453_{.011}$	$0.074_{.023}$	$\mathbf{0.071}_{.020}$
	50%	$0.637_{.019}$	$0.487_{.020}$	$0.377_{.016}$	$0.600_{.021}$	$0.268_{.041}$	$0.378_{.008}$	$0.559_{.026}$	$0.151_{.041}$	$0.367_{.033}$	$0.476_{.033}$	$\mathbf{0.118}_{.027}$	$\mathbf{0.118}_{.026}$
	70%	$0.786_{.010}$	$0.582_{.021}$	$0.531_{.021}$	$0.764_{.010}$	$0.401_{.032}$	$0.550_{.016}$	$0.713_{.030}$	$0.242_{.070}$	$0.510_{.033}$	$0.584_{.054}$	$\mathbf{0.202}_{.050}$	$0.204_{.047}$

Table 4: PCA eigenvalue recovery on five real datasets under MCAR missingness. Relative error of the two leading eigenvalues is reported; lower is better. Here, mr denotes the missing rate, and standard deviations are shown as subscripts with leading zeros omitted. Bold indicates the lowest mean error.

Dataset	mr	Mean	KNNI	SoftImpute	GAIN	PICA	XGBI	MIWAE	Pairwise-PSD	EC-IID	HeteroPCA	PPCA-EM	M^3
ANES96	20%	$0.247_{.015}$	$0.127_{.020}$	$0.201_{.017}$	$0.176_{.021}$	$0.172_{.018}$	$0.185_{.015}$	$0.210_{.032}$	$0.027_{.009}$	$0.055_{.016}$	$0.536_{.027}$	$0.162_{.018}$	$\mathbf{0.026}_{.009}$
	30%	$0.350_{.016}$	$0.197_{.028}$	$0.290_{.020}$	$0.258_{.042}$	$0.228_{.025}$	$0.233_{.017}$	$0.291_{.006}$	$0.038_{.021}$	$0.079_{.025}$	$0.512_{.065}$	$0.240_{.026}$	$\mathbf{0.034}_{.021}$
	50%	$0.520_{.017}$	$0.330_{.019}$	$0.439_{.021}$	$0.395_{.042}$	$0.375_{.024}$	$0.317_{.022}$	$0.450_{.031}$	$0.051_{.027}$	$0.126_{.022}$	$0.540_{.078}$	$0.363_{.025}$	$\mathbf{0.044}_{.025}$
	70%	$0.630_{.013}$	$0.352_{.048}$	$0.543_{.034}$	$0.483_{.058}$	$0.435_{.017}$	$0.337_{.033}$	$0.566_{.026}$	$0.082_{.044}$	$0.179_{.021}$	$0.551_{.094}$	$0.457_{.052}$	$\mathbf{0.080}_{.033}$
Fair	20%	$0.293_{.010}$	$0.122_{.012}$	$0.097_{.012}$	$0.197_{.023}$	$0.085_{.014}$	$0.119_{.011}$	$0.144_{.024}$	$0.017_{.008}$	$0.118_{.012}$	$0.213_{.023}$	$0.112_{.013}$	$\mathbf{0.013}_{.008}$
	30%	$0.415_{.015}$	$0.232_{.023}$	$0.149_{.016}$	$0.296_{.033}$	$0.130_{.015}$	$0.169_{.006}$	$0.213_{.017}$	$0.025_{.018}$	$0.166_{.019}$	$0.204_{.036}$	$0.175_{.017}$	$\mathbf{0.019}_{.007}$
	50%	$0.636_{.012}$	$0.445_{.020}$	$0.311_{.017}$	$0.513_{.054}$	$0.257_{.026}$	$0.289_{.030}$	$0.378_{.035}$	$0.039_{.016}$	$0.287_{.014}$	$0.221_{.036}$	$0.331_{.031}$	$\mathbf{0.034}_{.012}$
	70%	$0.795_{.013}$	$0.569_{.038}$	$0.507_{.023}$	$0.687_{.061}$	$0.446_{.042}$	$0.458_{.029}$	$0.599_{.069}$	$0.075_{.048}$	$0.398_{.025}$	$0.276_{.093}$	$0.491_{.032}$	$\mathbf{0.062}_{.038}$
RandHIE	20%	$0.260_{.026}$	$0.128_{.034}$	$0.033_{.019}$	$0.189_{.030}$	$0.104_{.025}$	$0.096_{.024}$	$0.159_{.039}$	$0.029_{.017}$	$0.084_{.015}$	$0.311_{.033}$	$0.129_{.024}$	$\mathbf{0.026}_{.018}$
	30%	$0.380_{.012}$	$0.229_{.033}$	$0.066_{.015}$	$0.276_{.032}$	$0.212_{.009}$	$0.175_{.018}$	$0.284_{.044}$	$0.032_{.021}$	$0.120_{.011}$	$0.278_{.080}$	$0.205_{.019}$	$\mathbf{0.026}_{.017}$
	50%	$0.599_{.014}$	$0.427_{.030}$	$0.248_{.031}$	$0.458_{.043}$	$0.400_{.026}$	$0.284_{.025}$	$0.469_{.030}$	$0.067_{.039}$	$0.207_{.024}$	$0.248_{.071}$	$0.382_{.025}$	$\mathbf{0.047}_{.022}$
	70%	$0.762_{.007}$	$0.554_{.031}$	$0.481_{.031}$	$0.613_{.039}$	$0.542_{.030}$	$0.404_{.069}$	$0.601_{.056}$	$0.081_{.038}$	$0.297_{.014}$	$0.246_{.078}$	$0.517_{.019}$	$\mathbf{0.065}_{.049}$
Economics	20%	$0.310_{.028}$	$0.040_{.014}$	$0.094_{.014}$	$0.132_{.022}$	$0.036_{.009}$	$0.103_{.014}$	$0.155_{.024}$	$0.021_{.012}$	$0.144_{.018}$	$0.041_{.021}$	$0.019_{.010}$	$\mathbf{0.017}_{.010}$
	30%	$0.447_{.015}$	$0.077_{.015}$	$0.145_{.012}$	$0.201_{.035}$	$0.039_{.008}$	$0.146_{.013}$	$0.235_{.021}$	$0.025_{.018}$	$0.223_{.012}$	$0.055_{.027}$	$0.031_{.009}$	$\mathbf{0.023}_{.014}$
	50%	$0.645_{.022}$	$0.160_{.021}$	$0.252_{.022}$	$0.351_{.073}$	$0.099_{.031}$	$0.251_{.013}$	$0.416_{.006}$	$0.056_{.023}$	$0.352_{.025}$	$0.092_{.063}$	$0.071_{.021}$	$\mathbf{0.049}_{.025}$
	70%	$0.766_{.009}$	$0.191_{.025}$	$0.322_{.016}$	$0.484_{.112}$	$0.104_{.006}$	$0.291_{.019}$	$0.543_{.041}$	$\mathbf{0.068}_{.035}$	$0.493_{.014}$	$0.238_{.038}$	$0.073_{.023}$	$0.086_{.018}$
Texas Housing	20%	$0.315_{.029}$	$0.182_{.033}$	$0.128_{.011}$	$0.295_{.028}$	$0.059_{.010}$	$0.132_{.023}$	$0.254_{.036}$	$0.035_{.035}$	$0.146_{.033}$	$0.130_{.045}$	$0.115_{.019}$	$\mathbf{0.022}_{.017}$
	30%	$0.427_{.033}$	$0.276_{.034}$	$0.187_{.021}$	$0.394_{.033}$	$0.092_{.022}$	$0.188_{.028}$	$0.345_{.027}$	$0.051_{.033}$	$0.183_{.043}$	$0.107_{.045}$	$0.151_{.034}$	$\mathbf{0.025}_{.024}$
	50%	$0.651_{.021}$	$0.461_{.035}$	$0.339_{.022}$	$0.604_{.025}$	$0.165_{.054}$	$0.344_{.013}$	$0.547_{.031}$	$0.057_{.039}$	$0.319_{.034}$	$0.126_{.071}$	$0.295_{.037}$	$\mathbf{0.036}_{.021}$
	70%	$0.800_{.010}$	$0.546_{.039}$	$0.500_{.025}$	$0.770_{.013}$	$0.304_{.050}$	$0.519_{.015}$	$0.700_{.043}$	$0.112_{.061}$	$0.451_{.031}$	$0.227_{.132}$	$0.385_{.034}$	$\mathbf{0.084}_{.030}$

M^3 substantially improves over conventional imputation-based recovery while remaining highly competitive with methods designed specifically for Gram-matrix estimation.

C.3 Table 4: PCA Eigenvalue Recovery

Across the 20 dataset–missing-rate settings in Table 4, M^3 achieves the lowest mean error in 19 cases. Notably, its relative error remains below 0.1 across all datasets and missing rates, even at 70% missingness, where it obtains 0.080, 0.062, 0.065, 0.086, and 0.084 on ANES96, Fair, RandHIE, Economics, and Texas Housing, respectively. The only exception is Economics at 70% missingness, where Pairwise-PSD achieves 0.068 compared with 0.086 for M^3. On the remaining four datasets at this missing rate, M^3 matches or improves upon the strongest task-specific competitors while substantially outperforming the imputation-based methods. These results indicate that the improved recovery of the underlying second-order structure largely carries over to the leading PCA spectrum.

C.4 Table 5: PLS Spectral Recovery

Across the 16 dataset–missing-rate settings in Table 5, M^3 achieves the lowest mean error in 15 cases. The only exception is Seals at 20% missingness, where PICA obtains an error of 0.023 compared with 0.029 for M^3. The advantage of direct product estimation is particularly evident at high missing

Table 5: PLS singular-value recovery on four real datasets under MCAR missingness. Relative error of the two leading singular values is reported; lower is better. Here, mr denotes the missing rate, and standard deviations are shown as subscripts with leading zeros omitted. Bold indicates the lowest mean error.

Dataset	mr	Mean	KNNI	SoftImpute	GAIN	PICA	XGBI	MIWAE	M^3
Diamonds	20%	$0.354_{.021}$	$0.072_{.011}$	$0.070_{.015}$	$0.261_{.098}$	$0.020_{.010}$	$0.095_{.014}$	$0.088_{.016}$	$\mathbf{0.011}_{.006}$
	30%	$0.502_{.024}$	$0.230_{.016}$	$0.109_{.016}$	$0.342_{.113}$	$0.018_{.015}$	$0.143_{.017}$	$0.126_{.012}$	$\mathbf{0.011}_{.008}$
	50%	$0.752_{.015}$	$0.423_{.029}$	$0.236_{.030}$	$0.556_{.125}$	$0.066_{.034}$	$0.271_{.024}$	$0.242_{.030}$	$\mathbf{0.030}_{.004}$
	70%	$0.907_{.015}$	$0.439_{.076}$	$0.443_{.074}$	$0.791_{.100}$	$0.242_{.105}$	$0.488_{.075}$	$0.417_{.058}$	$\mathbf{0.104}_{.035}$
Breast Cancer	20%	$0.364_{.005}$	$0.053_{.009}$	$0.063_{.006}$	$0.102_{.009}$	$0.050_{.008}$	$0.113_{.005}$	$0.125_{.009}$	$\mathbf{0.016}_{.006}$
	30%	$0.527_{.008}$	$0.074_{.005}$	$0.101_{.006}$	$0.060_{.008}$	$0.077_{.010}$	$0.177_{.007}$	$0.178_{.017}$	$\mathbf{0.036}_{.007}$
	50%	$0.767_{.010}$	$0.257_{.022}$	$0.213_{.014}$	$0.106_{.039}$	$0.220_{.024}$	$0.311_{.022}$	$0.286_{.037}$	$\mathbf{0.084}_{.038}$
	70%	$0.918_{.002}$	$0.567_{.046}$	$0.415_{.024}$	$0.251_{.028}$	$0.421_{.148}$	$0.503_{.035}$	$0.405_{.054}$	$\mathbf{0.234}_{.034}$
Gapminder	20%	$0.396_{.054}$	$0.159_{.021}$	$0.112_{.028}$	$0.363_{.031}$	$0.082_{.033}$	$0.137_{.027}$	$0.156_{.110}$	$\mathbf{0.069}_{.055}$
	30%	$0.552_{.068}$	$0.173_{.025}$	$0.130_{.022}$	$0.492_{.070}$	$0.129_{.062}$	$0.191_{.061}$	$0.247_{.225}$	$\mathbf{0.083}_{.072}$
	50%	$0.768_{.027}$	$0.257_{.049}$	$0.349_{.056}$	$0.390_{.234}$	$0.192_{.100}$	$0.291_{.151}$	$0.407_{.280}$	$\mathbf{0.097}_{.010}$
	70%	$0.900_{.015}$	$0.430_{.079}$	$0.525_{.048}$	$0.604_{.126}$	$0.306_{.139}$	$0.400_{.105}$	$0.598_{.280}$	$\mathbf{0.142}_{.113}$
Seals	20%	$0.359_{.033}$	$0.150_{.024}$	$0.090_{.015}$	$0.244_{.031}$	$\mathbf{0.023}_{.012}$	$0.128_{.014}$	$0.110_{.022}$	$0.029_{.007}$
	30%	$0.498_{.016}$	$0.211_{.024}$	$0.149_{.009}$	$0.333_{.031}$	$0.046_{.000}$	$0.191_{.018}$	$0.146_{.050}$	$\mathbf{0.009}_{.004}$
	50%	$0.748_{.004}$	$0.299_{.030}$	$0.317_{.019}$	$0.553_{.055}$	$0.166_{.026}$	$0.355_{.015}$	$0.332_{.073}$	$\mathbf{0.033}_{.022}$
	70%	$0.908_{.015}$	$0.452_{.010}$	$0.549_{.013}$	$0.808_{.029}$	$0.357_{.010}$	$0.528_{.008}$	$0.654_{.095}$	$\mathbf{0.084}_{.008}$

Table 6: Top-K recommendation performance at $K = 10$ on the Jester dataset with 50% missing training entries. NDCG, Recall, and MAP are reported in percentages; higher is better. Standard deviations are shown as subscripts with leading zeros omitted. Bold indicates the highest mean value for each metric.

Metric	Mean	KNNI	SoftImpute	GAIN	PICA	XGBI	MIWAE	CoRated	Shrunk	M^3
NDCG@10	$\mathbf{72.95}_{1.05}$	$72.84_{.69}$	$72.61_{.80}$	$70.89_{1.11}$	$68.99_{.76}$	$72.36_{.79}$	$72.65_{.57}$	$71.81_{.53}$	$72.41_{.34}$	$72.66_{.38}$
Recall@10	$33.20_{1.25}$	$33.63_{1.59}$	$33.70_{1.04}$	$31.17_{1.25}$	$28.67_{2.62}$	$32.63_{1.76}$	$32.60_{.75}$	$32.60_{.60}$	$33.47_{.98}$	$\mathbf{33.97}_{1.65}$
MAP@10	$18.11_{.47}$	$18.13_{1.49}$	$18.03_{.94}$	$16.49_{1.16}$	$14.15_{1.65}$	$16.87_{.73}$	$17.16_{.60}$	$17.57_{.53}$	$18.38_{.55}$	$\mathbf{18.58}_{.78}$

rates: at 70% missingness, M^3 obtains errors of 0.104, 0.234, 0.142, and 0.084 on Diamonds, Breast Cancer, Gapminder, and Seals, respectively, compared with 0.242, 0.251, 0.306, and 0.357 for the strongest competing method in each case. Overall, the results show that accurate recovery of the cross-view product translates effectively to the leading PLS singular spectrum, including under severe missingness.

C.5 Table 6: Recommendation system

As shown in Table 6, the strongest competing method varies across evaluation metrics, whereas M^3 maintains consistently strong performance across all three. Specifically, M^3 achieves the highest Recall@10 (33.97) and MAP@10 (18.58), while attaining an NDCG@10 of 72.66, only 0.29 percentage points below the best result of 72.95 obtained by Mean. In comparison, SoftImpute is competitive on Recall@10 (33.70), while Shrunk-ItemKNN is the strongest competing method on MAP@10 (18.38). These results suggest that direct estimation of the item–item product provides a favorable balance across ranking quality, retrieval coverage, and average precision rather than specializing in a single recommendation metric.

D Additional Scaling Results

D.1 Table 7: Joint Scaling under Gaussian Data

As shown in Table 7, M^3 maintains favorable accuracy as all three matrix dimensions increase jointly. It achieves the lowest mean error in S1–S3, with errors decreasing from 0.208 in S1 to 0.110 in S2 and 0.072 in S3, approximately 18–21% lower than the strongest competing method in these settings. At the largest scale S4, MIWAE obtains the lowest error (0.053), while M^3 remains competitive

Table 7: Joint scaling under Gaussian data and 50% MCAR missingness. We consider S1 $(r, d, s) = (16, 256, 16)$, S2 $(32, 512, 32)$, S3 $(64, 1024, 64)$, and S4 $(128, 2048, 128)$. For each setting, the first row reports relative Frobenius error and the second reports runtime in seconds. Standard deviations are shown as subscripts with leading zeros omitted. Lower is better for both metrics; bold indicates the lowest mean error.

Scale	Metric	Mean	KNNI	SoftImpute	GAIN	PICA	XGBI	MIWAE	M^3
S1	Error	$0.527_{.012}$	$0.379_{.048}$	$0.411_{.042}$	$0.347_{.016}$	$0.262_{.045}$	$0.295_{.030}$	$0.363_{.050}$	$\mathbf{0.208}_{.043}$
	Runtime	$0.001_{.000}$	$0.020_{.002}$	$0.035_{.003}$	$0.619_{.016}$	$0.586_{.002}$	$0.497_{.015}$	$0.533_{.034}$	$0.001_{.000}$
S2	Error	$0.501_{.013}$	$0.248_{.005}$	$0.287_{.004}$	$0.299_{.027}$	$0.233_{.007}$	$0.219_{.010}$	$0.134_{.005}$	$\mathbf{0.110}_{.002}$
	Runtime	$0.002_{.000}$	$0.105_{.005}$	$0.183_{.005}$	$0.741_{.016}$	$2.144_{.018}$	$2.684_{.070}$	$1.732_{.030}$	$0.002_{.000}$
S3	Error	$0.471_{.040}$	$0.151_{.014}$	$0.218_{.005}$	$0.257_{.012}$	$0.209_{.020}$	$0.176_{.015}$	$0.090_{.007}$	$\mathbf{0.072}_{.006}$
	Runtime	$0.003_{.000}$	$0.838_{.018}$	$0.830_{.062}$	$1.256_{.027}$	$8.108_{.184}$	$11.843_{.275}$	$7.722_{.099}$	$0.008_{.000}$
S4	Error	$0.484_{.041}$	$0.102_{.007}$	$0.195_{.002}$	$0.102_{.005}$	$0.210_{.017}$	$0.164_{.014}$	$\mathbf{0.053}_{.003}$	$0.064_{.004}$
	Runtime	$0.011_{.001}$	$7.636_{.278}$	$1.248_{.040}$	$1.651_{.094}$	$31.116_{.207}$	$51.197_{.189}$	$26.394_{.206}$	$0.035_{.001}$

Table 8: Inner-dimension scaling under Gaussian data and 50% MCAR missingness. We fix $r = s = 16$ and vary $d \in \{256, 512, 1024, 2048, 4096\}$. For each value of d, the first row reports relative Frobenius error and the second reports runtime in seconds. Standard deviations are shown as subscripts with leading zeros omitted. Lower is better for both metrics; bold indicates the lowest mean error.

d	Metric	Mean	KNNI	SoftImpute	GAIN	PICA	XGBI	MIWAE	M^3
256	Error	$0.527_{.012}$	$0.379_{.048}$	$0.411_{.042}$	$0.347_{.016}$	$0.262_{.045}$	$0.295_{.030}$	$0.363_{.050}$	$\mathbf{0.208}_{.043}$
	Runtime	$0.002_{.002}$	$0.044_{.015}$	$0.052_{.008}$	$0.938_{.016}$	$0.732_{.010}$	$0.704_{.028}$	$0.681_{.023}$	$0.002_{.001}$
512	Error	$0.524_{.032}$	$0.380_{.039}$	$0.417_{.031}$	$0.310_{.014}$	$0.251_{.028}$	$0.258_{.030}$	$0.174_{.016}$	$\mathbf{0.138}_{.018}$
	Runtime	$0.004_{.003}$	$0.086_{.022}$	$0.140_{.012}$	$1.081_{.189}$	$1.566_{.402}$	$1.075_{.109}$	$1.769_{.560}$	$0.003_{.001}$
1024	Error	$0.518_{.049}$	$0.379_{.038}$	$0.389_{.017}$	$0.267_{.047}$	$0.233_{.022}$	$0.243_{.022}$	$0.137_{.013}$	$\mathbf{0.090}_{.006}$
	Runtime	$0.004_{.002}$	$0.346_{.030}$	$0.448_{.035}$	$1.089_{.014}$	$3.046_{.055}$	$1.222_{.024}$	$3.893_{1.486}$	$0.004_{.001}$
2048	Error	$0.523_{.056}$	$0.387_{.038}$	$0.392_{.012}$	$0.273_{.029}$	$0.235_{.020}$	$0.233_{.019}$	$0.134_{.010}$	$\mathbf{0.072}_{.009}$
	Runtime	$0.005_{.000}$	$2.198_{1.457}$	$1.021_{.334}$	$2.773_{1.545}$	$5.659_{.444}$	$1.562_{.356}$	$7.625_{1.916}$	$0.008_{.003}$
4096	Error	$0.516_{.055}$	$0.395_{.040}$	$0.384_{.019}$	$0.258_{.036}$	$0.225_{.023}$	$0.228_{.027}$	$0.122_{.018}$	$\mathbf{0.050}_{.008}$
	Runtime	$0.008_{.002}$	$7.122_{1.065}$	$3.080_{2.157}$	$3.630_{.480}$	$11.171_{.344}$	$1.914_{.119}$	$14.069_{.238}$	$0.011_{.002}$

at 0.064. The computational difference becomes increasingly pronounced with scale: the runtime of M^3 grows from 0.001 s in S1 to only 0.035 s in S4, compared with 7.636 s for KNNI, 26.394 s for MIWAE, 31.116 s for PICA, and 51.197 s for XGBI at S4. Thus, M^3 preserves strong recovery accuracy while scaling substantially more efficiently than the competing estimation pipelines.

D.2 Table 8: Inner-Dimension Scaling

As shown in Table 8, M^3 achieves the lowest mean error at every inner dimension considered. As d increases from 256 to 4096 while keeping $r = s = 16$ fixed, its error decreases from 0.208 to 0.050, whereas the strongest competing error decreases from 0.262 to 0.122. The relative advantage of M^3 therefore becomes larger with increasing d: its error is approximately 21% lower than the strongest competitor at $d = 256$ and about 59% lower at $d = 4096$. This improvement is obtained with only a modest increase in runtime, from 0.002 s to 0.011 s, while several competing methods require seconds at the largest dimension, including 7.122 s for KNNI, 11.171 s for PICA, and 14.069 s for MIWAE. These results indicate that M^3 benefits from increasing inner dimension while retaining highly favorable computational scaling.

D.3 Table 9: Outer-Dimension Scaling

As shown in Table 9, M^3 achieves the lowest mean error for $p = 16$, 32, and 64, with errors of 0.090, 0.087, and 0.071, respectively. Relative to the strongest competing method, these correspond to reductions of approximately 34%, 7%, and 21%. At the largest outer dimension, $p = 128$, MIWAE

Table 9: Outer-dimension scaling under Gaussian data and 50% MCAR missingness. We fix $d = 1024$ and vary $r = s = p \in \{16, 32, 64, 128\}$, so that the output matrix grows from 16×16 to 128×128 while the shared inner dimension remains fixed. For each value of p, the first row reports relative Frobenius error and the second reports runtime in seconds. Standard deviations are shown as subscripts with leading zeros omitted. Lower is better for both metrics; bold indicates the lowest mean error.

p	Metric	Mean	KNNI	SoftImpute	GAIN	PICA	XGBI	MIWAE	M^3
16	Error	$0.534_{.056}$	$0.389_{.048}$	$0.386_{.022}$	$0.256_{.060}$	$0.234_{.031}$	$0.251_{.024}$	$0.136_{.018}$	$\mathbf{0.090}_{.009}$
	Runtime	$0.003_{.000}$	$0.345_{.003}$	$0.242_{.002}$	$1.186_{.014}$	$4.432_{1.498}$	$1.525_{.119}$	$3.600_{.483}$	$0.003_{.000}$
32	Error	$0.489_{.011}$	$0.234_{.017}$	$0.265_{.011}$	$0.256_{.007}$	$0.215_{.008}$	$0.194_{.008}$	$0.094_{.002}$	$\mathbf{0.087}_{.010}$
	Runtime	$0.006_{.002}$	$0.457_{.057}$	$0.894_{.318}$	$1.478_{.035}$	$7.313_{1.310}$	$5.206_{1.445}$	$4.789_{.328}$	$0.006_{.001}$
64	Error	$0.477_{.054}$	$0.148_{.019}$	$0.216_{.002}$	$0.251_{.009}$	$0.211_{.027}$	$0.178_{.021}$	$0.090_{.010}$	$\mathbf{0.071}_{.008}$
	Runtime	$0.006_{.000}$	$0.852_{.195}$	$1.216_{.187}$	$2.333_{.819}$	$11.756_{.695}$	$15.788_{.184}$	$8.959_{.201}$	$0.018_{.008}$
128	Error	$0.506_{.055}$	$0.124_{.004}$	$0.197_{.001}$	$0.127_{.007}$	$0.222_{.018}$	$0.181_{.017}$	$\mathbf{0.067}_{.002}$	$0.094_{.003}$
	Runtime	$0.010_{.000}$	$1.337_{.009}$	$1.163_{.332}$	$2.265_{.050}$	$25.800_{4.743}$	$69.038_{2.000}$	$23.920_{1.630}$	$0.035_{.005}$

Table 10: Joint scaling under Student-t_5 data and 50% MCAR missingness using the same four settings as the Gaussian joint-scaling experiment: S1 $(r, d, s) = (16, 256, 16)$, S2 $(32, 512, 32)$, S3 $(64, 1024, 64)$, and S4 $(128, 2048, 128)$. For each setting, the first row reports relative Frobenius error and the second reports runtime in seconds. Standard deviations are shown as subscripts with leading zeros omitted. Lower is better for both metrics; bold indicates the lowest mean error.

Scenario	Metric	Mean	KNNI	SoftImpute	GAIN	PICA	XGBI	MIWAE	M^3
S1	Error	$0.492_{.042}$	$0.383_{.056}$	$0.437_{.033}$	$0.426_{.099}$	$0.281_{.050}$	$0.312_{.052}$	$0.362_{.062}$	$\mathbf{0.209}_{.095}$
	Runtime	$0.002_{.000}$	$0.022_{.002}$	$0.034_{.003}$	$2.325_{.116}$	$0.625_{.016}$	$0.525_{.019}$	$0.568_{.024}$	$0.001_{.000}$
S2	Error	$0.508_{.037}$	$0.313_{.035}$	$0.318_{.026}$	$0.337_{.021}$	$0.253_{.040}$	$0.268_{.040}$	$0.180_{.030}$	$\mathbf{0.177}_{.034}$
	Runtime	$0.002_{.000}$	$0.128_{.011}$	$0.256_{.021}$	$2.342_{.111}$	$2.279_{.079}$	$2.715_{.121}$	$1.916_{.151}$	$0.002_{.000}$
S3	Error	$0.471_{.043}$	$0.215_{.027}$	$0.235_{.011}$	$0.259_{.046}$	$0.213_{.024}$	$0.208_{.026}$	$0.115_{.008}$	$\mathbf{0.110}_{.019}$
	Runtime	$0.004_{.000}$	$0.889_{.039}$	$0.812_{.338}$	$2.911_{.163}$	$8.832_{.193}$	$12.079_{.349}$	$8.751_{.960}$	$0.014_{.004}$
S4	Error	$0.492_{.047}$	$0.175_{.017}$	$0.200_{.001}$	$0.110_{.004}$	$0.215_{.018}$	$0.200_{.018}$	$\mathbf{0.083}_{.007}$	$0.090_{.003}$
	Runtime	$0.015_{.002}$	$7.602_{.084}$	$3.577_{1.536}$	$3.038_{.095}$	$33.155_{.576}$	$52.067_{.365}$	$29.487_{.558}$	$0.032_{.002}$

attains the lowest error (0.067), while M^3 remains competitive at 0.094. Despite the growing output size, the runtime of M^3 increases only from 0.003 s at $p = 16$ to 0.035 s at $p = 128$, compared with 1.337 s for KNNI, 23.920 s for MIWAE, 25.800 s for PICA, and 69.038 s for XGBI at the largest scale. Overall, M^3 retains strong recovery accuracy while scaling efficiently as the outer dimensions and output matrix size increase.

D.4 Table 10: Robustness under Student-t_5 Data

As shown in Table 10, M^3 remains effective under Student-t_5 data despite relying on a Gaussian working model. It achieves the lowest mean error in S1–S3, with errors of 0.209, 0.177, and 0.110, compared with 0.281, 0.180, and 0.115 for the strongest competing method in the corresponding settings. At the largest scale S4, MIWAE achieves the lowest error (0.083), while M^3 remains close at 0.090. Computationally, the runtime of M^3 increases only from 0.001 s in S1 to 0.032 s in S4; at S4, the competing methods require between 3.038 s and 52.067 s. These results suggest that the computational and statistical advantages of direct product estimation extend beyond Gaussian data to a heavier-tailed setting.

E Discussion

The results suggest a broader perspective on computation with missing data: when the downstream target is a matrix product, reconstructing the operands is not always necessary. By directly estimating missing multiplicative contributions, M^3 provides a unified operator that applies to direct products, self-products, and cross-view products. The experiments further indicate that the pair-local Gaussian

model can remain effective beyond Gaussian data, despite being introduced primarily as a lightweight working approximation.

Several directions are particularly promising for future work. One is to extend M^3 beyond the current setting in which observation masks are independent of the underlying values, for example, to MAR and MNAR mechanisms that explicitly model the missingness process. Another is to generalize the pair-local conditional-product framework beyond Gaussian working models to richer dependence structures and non-Gaussian distributions. A further direction is to investigate estimators that exploit information beyond individual row–column pairs, potentially improving statistical efficiency while retaining the computational advantages of direct product estimation.