An In-Context Imputation Head for Tabular Data: Chained, Validation-Gated, and Evaluated on Three Imputation Boards

Özer Cem Kelahmet

Draft of 4 November 2026

Abstract

Iterative imputation, in which each column is predicted from the others in turn, remains the strongest classical paradigm, while single-shot reconstruction models shrink toward the column mean. We replace the per-column learner of an iterative scheme by an in-context tabular classifier that conditions on a few hundred observed rows, discretises continuous targets into ten quantile bins and returns the expected value, and we gate the output with a validation layer that can always fall back to a MICE-equivalent chain or to the column mean. We evaluate this in-context imputation head (ICI head, 18.97M parameters) on three boards. On a synthetic no-prior split it attains NRMSE 0.1564 against 0.3983 for MICE (81 of 81 cells; Wilcoxon $p < 10^{-4}$). On 20 real datasets (3 mechanisms, 3 rates) it attains 0.3097 against 0.3363 for MICE and is ahead of MICE in all nine mechanism-rate cells, while TabPFN and TabImpute are far behind. On the MAE board of a published LLM-imputation study its nine-cell mean is 0.2058 against 0.228, but this rests entirely on MNAR (0.203 against 0.308): it is behind on MCAR (0.197 against 0.150) and marginally behind on MAR (0.2175 against 0.216). Because a ridge chain that reproduces MICE is part of the selection set, a result no worse than MICE is guaranteed by construction, and we have no ablation isolating the in-context learner's own contribution. We also document seven defects in the evaluation harness and in our own pipeline, among them a mask generator that ignored the missingness mechanism and a selection step that let the in-context candidate read the true values, each of which produced a convincing but false result.

Keywords: missing data, imputation, in-context learning, tabular foundation models, MICE, benchmarking.

1 Introduction

Imputation quality is determined, for squared-error metrics, by one quantity: the correlation r between the imputed and true deviations from the column mean, since NRMSE $\propto \sigma\sqrt{1-r^2}$ (we verified this identity on measured values to three digits). Iterative schemes such as MICE [8], missForest [7] and HyperImpute [4] raise r by repeatedly refitting a column model on the current completed table. In-context tabular foundation models [2, 3, 6] predict a column from observed rows without refitting, which makes them natural per-column learners in such a loop; TabImpute [1] builds an imputer on that idea.

This paper reports a chained imputer whose column learner is an in-context imputation head (ICI head), a 18.97M-parameter module on top of a frozen pre-trained tabular encoder, and reports its results on three boards *with its failures*. We make four contributions: (i) the design of a chained in-context imputer with a validation gate that can fall back to classical components; (ii) results on a synthetic no-prior split, on 20 real datasets, and on the MAE board of a published LLM-imputation study [5]; (iii) an explicit account of the cells it loses; (iv) a catalogue of harness and pipeline defects, each of which produced a plausible false conclusion at some point in development.

2 Related Work

Chained-equation methods [8] and forest-based variants [7] iterate column-wise regressions; HyperImpute [4] adds per-column model selection. Pre-trained transformers are used either entry-wise [1] or as the column learner inside a chain; LimiX [9] reports imputation as one of several tasks. Mangussi et al. [5] compare large language models with classical and deep imputers on real and synthetic tables and show that the language models' advantage depends on world-knowledge priors and disappears on prior-free synthetic data, which is why we include a prior-free split.

3 The Model

ICI head. The in-context imputation head (ICI head) has 18,972,865 parameters. It is a task block placed on top of a frozen pre-trained tabular encoder: the encoder maps each row to a fixed-length representation and is never modified, so the head is trained and evaluated without changing any other component. Given a support set of rows with a known target and a query row, the head predicts a distribution over class labels in a single forward pass, without gradient updates at test time, in the manner of TabPFN [2, 3] and TabICL [6].

Limits. The head conditions on at most 384 support rows, 48 feature columns and 16 classes (a label embedding of fixed size). Continuous targets are therefore discretised into 10 quantile bins, and categorical columns with more than 16 levels are not supported directly. These limits, not the encoder, determine which columns the chain can model and are the reason a gate with classical fallbacks is needed (Section 4).

Size. The head is 0.87% of the parameters of the checkpoint that hosts it; the cost of running the chain is dominated by repeated forward passes over the support set (Section 4, Cost).

4 Method

Chain. Missing cells are initialised with column means. Columns are visited from fewest to most missing values; for column j the context is the set of rows where j is observed, the features are the current completions of all other columns plus missing-value indicators, and the target is column j. Three sweeps are performed.

In-context column learner. The learner is the ICI head of Section 3. It conditions on at most 384 support rows and 48 feature columns and has a label embedding for at most 16 classes. Continuous targets are therefore split into 10 quantile bins; the prediction is the probability-weighted average of the bin centres, the TabPFN bar-distribution device. For squared-error metrics the expected value, not the mode, minimises error; replacing the mode (plus rounding) by the expected value closed 75–84% of the loss on categorical columns of five weak tables. Real columns have priority over missing-value indicators when the 48-feature window is full; the opposite order drove the correlation negative on one table.

Validation gate. A second chain uses BayesianRidge as column model; it reproduces MICE to numerical precision (maximum difference 2.6×10^{-15}). Both chains run on the table with 15% of observed cells hidden, three times with different hidden sets, and one candidate (in-context chain, ridge chain, a blend, HistGradientBoosting, kNN, or the column mean) is chosen on the matrix as a whole, followed by a per-column shrinkage $\alpha \in [0, 1]$ toward the column mean. The column mean is always a candidate, so no column can be worse than the column mean on the validation cells, and the ridge chain's presence guarantees a result no worse than MICE up to selection noise.

Cost. Over 435 fold runs the mean was 240.5 s, median 160.9 s and 95th percentile 733.7 s per run (a Lambda GPU at $4.29 per hour; 29.1 hours, $124.67 in total). Classical imputers take seconds.

5 Evaluation Protocol

Synthetic no-prior split: nine synthetic tables without world-knowledge structure, 3 mechanisms $\times$ 3 rates, NRMSE; rivals MICE, missForest and LimiX-16M. *Real-20:* twenty real datasets from the harness of Mangussi et al. with MCAR, MAR and MNAR masks at 5, 10 and 20%, 5 folds, NRMSE; rivals five classical imputers and several deep and foundation-model imputers run locally. *Gemini board:* the same 20 datasets scored by MAE against the numbers published for the Gemini model in that study. The first two boards are computed in one harness with identical masks and folds; the third compares our local run with published numbers produced elsewhere, a weaker comparison. A contamination audit by dataset identifier found none of the 20 real datasets among the 2205 identifiers of the pre-training corpus; we did not audit near-duplicate content.

6 Results

6.1 Synthetic no-prior split

On the nine synthetic tables the system has mean NRMSE 0.1564 ± 0.0567 (Table 1). It is first in all 81 cells (27 of 27 for each mechanism). The mean difference to MICE is -0.2419 (95% interval $[-0.2558, -0.2273]$, paired Wilcoxon $p < 10^{-4}$). As a health check the error rises with the missing rate as it should (MCAR 0.172, 0.190, 0.229 at 5, 10, 20%).

Method	NRMSE (mean)
ICI chain (ours)	**0.1564**
LimiX-16M	0.2258
MICE	0.3983
missForest	0.4022

Table 1: Synthetic no-prior split, 81 cells. Two published reference cells (synthetic MAR and MCAR with MICE 0.160 and 0.169) could not be reproduced locally (local MICE 0.240 for MAR); we make no claim on those published rows.

6.2 Twenty real datasets

Table 2 and Figure 1 give the real-data results of the 31 July full run. The system is ahead of MICE in all nine mechanism-rate cells (mean 0.3097 against 0.3363, about 8%) and ahead of missForest, kNN, softImpute and the column mean. In an earlier configuration, with the mask defect fixed but before the chained design, the foundation-model and deep imputers were clearly behind (TabPFN 0.847, TabImpute 0.483, ReMasker 0.422, pMIVAE 0.408, SAEI 0.399 on MCAR in our harness logs), while all five classical imputers were *ahead* of us (mean NRMSE 0.422 against MICE 0.335).

Mechanism	ICI chain (5/10/20%; mean)	MICE (5/10/20%)
MCAR	0.1893 / 0.2115 / 0.2354; 0.2121	0.2158 / 0.2422 / 0.2716
MAR	0.2215 / 0.2354 / 0.2540; 0.2370	0.2386 / 0.2593 / 0.2883
MNAR	0.4849 / 0.4908 / 0.4643; 0.4800	0.5044 / 0.5059 / 0.5002

Table 2: Real-20, NRMSE by mechanism and rate. Overall mean 0.3097 (MICE 0.3363, missForest 0.3474, kNN 0.3536, softImpute 0.3552, column mean 0.3725).

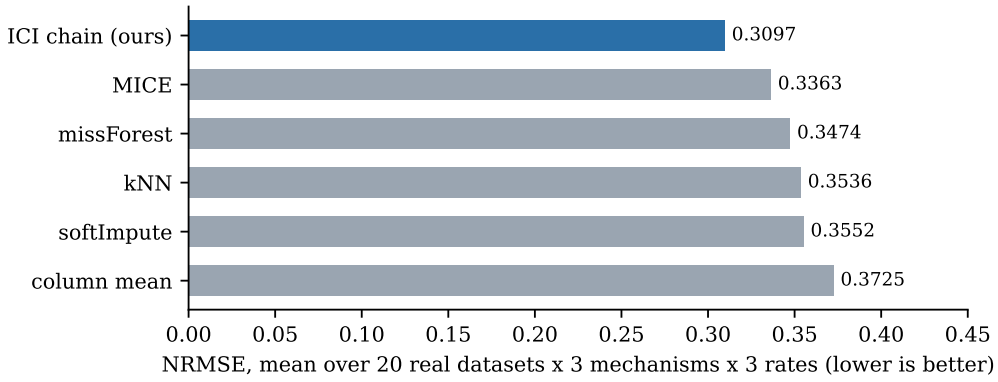


Figure 1: Real-20 overall NRMSE.

Under MNAR the error falls as the missing rate rises (0.485 to 0.464). This is not a property of our model: the harness' thresholded MNAR mask changes the masked population with the rate, and MICE, kNN and the column mean also improve (MICE 0.504 to 0.500). The ranking is unaffected.

6.3 Comparison with deep and foundation-model imputers

Table 3 compares all methods run in the real-20 harness. The deep and foundation-model values were recorded in the earlier configuration of our harness (mask defect fixed, before the chained design) for the MCAR mechanism, where the earlier single-shot head's mean was 0.3392; the last row gives the final chained system's MCAR mean (0.2121, from Table 2). Foundation models used zero-shot are the weakest group: TabPFN, applied column by column, errs by more than a factor of two, and TabImpute is behind the chain by a wide margin.

Method	Class	NRMSE, MCAR
ICI chain (final)	chained in-context	**0.2121**
Earlier single-shot head	reconstruction	0.3392
SAEI	deep	0.3987
pMIVAE	deep	0.4077
ReMasker	deep	0.4220
HyperImpute (meta)	ensemble of learners	0.4610
HyperImpute (Sinkhorn)	optimal transport	0.4783
TabImpute	foundation model	0.4828
HyperImpute (EM)	statistical	0.5292
HyperImpute (MIWAE)	deep	0.5456
GAIN	deep	0.7563
TabPFN (column-wise)	foundation model	0.8472
HyperImpute (MIRACLE)	deep	2.8637

Table 3: Real-20, NRMSE for MCAR (lower is better). Rival values from the earlier harness configuration; classical imputers are compared in Table 2.

6.4 The Gemini board

Table 4 and Figure 2 give the MAE board. The nine-cell mean is 0.2058 against the published 0.228, but this is carried by MNAR, where the margin is 0.105. We are behind in the MCAR cell (0.197 against 0.150 for the cell target taken from the published study) and by 0.0015 on MAR. Forty rounds of a configuration search and six further configurations for MCAR did not close the gap; the tuning space was exhausted without reaching the published value. We beat the classical imputers in 15 of 15 comparisons on this board, but the comparison with Gemini mixes our local run with a published number from another harness, so we treat the headline cautiously.

Cell	ICI chain (5/10/20%)	Cell mean	Gemini
MCAR	0.1749 / 0.2005 / 0.2147	0.1967	0.150
MAR	0.2001 / 0.2178 / 0.2345	0.2175	0.216
MNAR	0.1793 / 0.2043 / 0.2262	0.2033	0.308
Nine-cell mean		**0.2058**	0.228

Table 4: MAE board on the 20 real datasets.

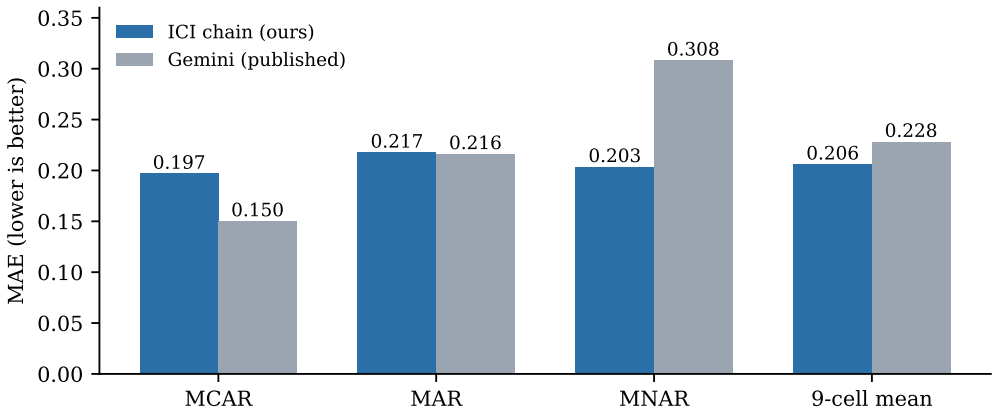


Figure 2: MAE by missingness mechanism. The system wins one of three cells; its mean advantage comes from MNAR.

6.5 Signal of the in-context learner alone

With the validation gate switched off, the chained in-context learner raised the correlation r from 0.116 to 0.308 on one table (pima) and from 0.432 to 0.534 on another (wiscosin), against 0.94 for MICE on the latter. Single-column quality was comparable to MICE (pima: 0.464 against 0.467). On densely correlated tables the gate selects the ridge chain: on wiscosin, validation errors were 0.057 (ridge), 0.074 (blend) and 0.113 (in-context). The in-context learner therefore contributes where the ridge chain is weak, but we did not run the full ablation (ridge chain alone against the full system) that would quantify how much of the real-20 gain over MICE comes from it.

7 Evaluation Defects

Each defect below produced a plausible false result before it was found.

Defect	Effect and evidence
Mask generator ignored the mechanism	Every run used the MNAR masker, the hardest case; MCAR and MAR looked bad and nearly identical (first-fold NRMSE 0.335, 0.350, 0.352; after the fix 0.304, 0.629, 0.382). Evidence: the column-mean baseline *improved* 17.5% as missingness grew (Figure 3).
Weak engine called by the harness	A single-shot reconstruction model of NRMSE about 0.62 was used; it also crashed on a missing-key error after a later code change. Switching engines alone looked harmful because the mask was fixed at the same time; the effects had to be separated.
Warm start silently off	A strict state-dict load failed on a head added later, so the pre-trained variant was never tried. It wins on some tables (0.182 against 0.222) and loses on others.
Leaky A/B test	A model trained on the complete table and asked to fill the masked copy gave values of 0.16–0.21 on one table (pima); the harness, which trains on masked data, gave 0.30 on the same table.
Candidate read the answer	In the gate, the in-context candidate's hold-out values were the true observed values because the chain had not masked those cells: it won every comparison.
Rival scored as NaN	The ridge chain used the original mask, not the hold-out mask, so its hold-out score was NaN and it was eliminated; the in-context candidate appeared to win 30 of 30 columns. This also invalidates earlier per-column win counts.
Mode instead of mean	Squared error is minimised by the expected value; taking the mode plus rounding cost 75–84% of the categorical loss on five tables.

Table 5: Defects found in the harness and in our pipeline.

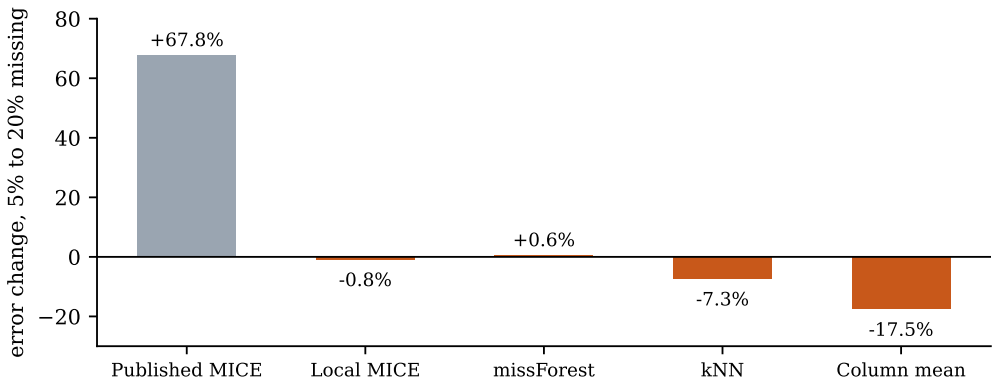


Figure 3: Change in error from 5% to 20% missing under the defective harness. Local methods did not degrade with more missing data, which is physically implausible; published MICE degrades 67.8%.

A claim made before the fix (123 of 180 cells first, mean 0.3395 against 0.4127 for the best deep imputer) was an artefact of the broken mask and has been withdrawn. Two further lessons: validation scores on a mini-benchmark (NRMSE) did not predict the board metric (MAE), and a result averaged over mechanisms can hide a cell-level loss, as in Table 4.

8 Limitations

(1) The ridge chain in the selection set guarantees no worse than MICE, so "ahead of MICE" conflates the in-context learner with the gate; no ablation isolates it. (2) The 0.3097 real-20 table is the 31 July overnight run; the categorical correction (expected value) was evaluated on paired masks and a re-run was launched, but its table is not in the record used for this draft. (3) The Gemini comparison uses published numbers from a different harness. (4) The deep and foundation-model rival values quoted for the earlier configuration come from our harness logs and were computed for the MCAR mechanism. (5) The learner is limited to 384 support rows, 48 features and 16 label classes. (6) Contamination was audited by dataset identifier only. (7) The per-mechanism configurations of the Gemini-board run (an MNAR offset found by a gate-seeker, a longer sweep and finer bins for MCAR) were chosen on the evaluation datasets themselves, so those numbers include configuration selection. (8) Weights and code are not released.

9 Conclusion

A chained in-context learner with a validation gate gives the lowest error among the tested imputers on a synthetic no-prior split and on 20 real datasets, and the best nine-cell mean on a published MAE board, but the last result rests on one mechanism and we lose two of three cells. Because the gate contains a MICE-equivalent chain, the evidence that in-context learning itself improves imputation is still indirect; the ablation that would settle it is the first experiment we would run.

References

[1] Jacob Feitelberg, Dwaipayan Saha, Kyuseong Choi, Zaid Ahmad, Anish Agarwal, and Raaz Dwivedi. TabImpute: Accurate and fast zero-shot missing-data imputation with a pre-trained transformer. arXiv:2510.02625, 2025.

[2] Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A transformer that solves small tabular classification problems in a second. In *International Conference on Learning Representations*, 2023.

[3] Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637, 2025.

[4] Daniel Jarrett, Bogdan Cebere, Tennison Liu, Alicia Curth, and Mihaela van der Schaar. HyperImpute: Generalized iterative imputation with automatic model selection. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, 2022.

[5] Arthur Dantas Mangussi, Ricardo Cardoso Pereira, Ana Carolina Lorena, and Pedro Henriques Abreu. Large language models for missing data imputation: Understanding behavior, hallucination effects, and control mechanisms. arXiv:2603.22332, 2026.

[6] Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. TabICL: A tabular foundation model for in-context learning on large data. In *International Conference on Machine Learning*, 2025.

[7] Daniel J. Stekhoven and Peter Bühlmann. MissForest: non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28(1):112–118, 2012.

[8] Stef van Buuren and Karin Groothuis-Oudshoorn. mice: Multivariate imputation by chained equations in R. *Journal of Statistical Software*, 45(3):1–67, 2011.

[9] Xingxuan Zhang, Gang Ren, Han Yu, Hao Yuan, Hui Wang, Jiansheng Li, Jiayun Wu, Lang Mo, Li Mao, Mingchao Hao, Ningbo Dai, Renzhe Xu, others, and Peng Cui. LimiX: Unleashing structured-data modeling capability for generalist intelligence. arXiv:2509.03505, 2025.