Repackaging Temporal Evidence: A Unifying Interface for Temporal Prediction

Anonymous Authors[1]

Abstract

Temporal tabular prediction and irregular time-series prediction often use related longitudinal records, but they define different prediction protocols. Temporal tables expose a timestamped covariate row and test future-period generalization, whereas irregular time-series tasks expose entity-specific observations around a queried time. We study this difference as an evidence-packaging issue rather than a boundary between data modalities. Our unified protocol interface specifies the information available at query time before a predictor is applied. It identifies two failure modes in direct cross-domain transfer: entity collapse, where records from unrelated entities are treated as one sequence, and temporal relation mismatch, where a tabular row hides the source sequence needed by a time-series model. The same interface also yields protocol-level recoveries: context-based tabular predictors can be evaluated on irregular histories through entity contexts, and selected temporal tables can be recovered as entity histories or sequence-to-label examples.

1. Introduction

Forecasting systems in healthcare, sensor monitoring, and business operations must predict from records that are both heterogeneous and temporally structured: measurements arrive asynchronously, sensors report irregularly, and populations drift over calendar time (Rajkomar et al., 2018; Ganesan et al., 2004; Nederstigt et al., 2014). In these settings, performance depends not only on the model class, but also on the prediction protocol: what is the query, which past observations are admissible, and which covariates are known when the forecast is made? Two benchmark families answer these questions differently. Temporal tabular prediction represents each query as a timestamped covariate row and evaluates future-period generalization (Rubachev et al., 2025; Cai & Ye, 2025b;a). Irregular time-series prediction represents each query through observations in an entity history and evaluates interpolation or extrapolation at selected times (Rubanova et al., 2019; Schirmer et al., 2022; Shukla & Marlin, 2021). As shown in Figure 1, the distinction is therefore a difference in query-time evidence, not only a difference in architecture.

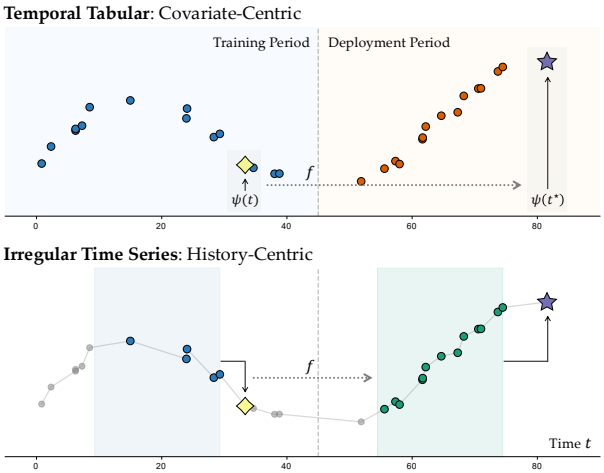

Figure 1. **Temporal tabular prediction** is covariate-centric: the query is a timestamped row evaluated on future periods. **Irregular time-series forecasting** is history-centric: the query is answered from timestamped observations in an entity history.

Our starting point is that this divide is induced by evidence packaging. The same longitudinal record can be exposed as independent rows, labeled candidate sets, entity histories, or sequence-to-label examples. These choices determine the query unit, admissible context, entity grouping, and target relation. Consequently, poor transfer between tabular and time-series methods can arise because the converted task no longer supplies the evidence expected by the model.

We formalize this point with a protocol interface that separates the query-time information set from the predictor applied to it. The interface exposes two concrete failures. *Entity collapse* occurs when records from different entities are ordered as one artificial stream. *Temporal relation mismatch* occurs when the predictive sequence lies inside a tabulated row rather than between rows. We then evaluate two protocol realignments. A tabular lens repackages irregular histories as temporally embedded, entity-specific

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

Table 1. **Comparison of default temporal prediction protocols.** A checkmark indicates that the default protocol explicitly exposes the structure to the learner; (✓) indicates that it is required by the standard formulation; × means that the structure is not explicitly exposed, even if it may exist in the raw data.

Settings	Multi-entity	Multivariate	Covariates	Look-back	Irregularity	Interpolation	Overlap
Regular Time Series	✓	✓	✓	(✓)	×	×	×
Irregular Time Series	✓	✓	✓	(✓)	✓	✓	×
I.i.d. Tabular	×	×	(✓)	×	×	(✓)	(✓)
Temporal Tabular	×	×	(✓)	×	✓	✓	✓
Protocol Interface	✓	✓	✓	✓	✓	✓	✓

contexts for a context-based tabular predictor. A time-series lens recovers station histories in Weather and pre-offer transaction sequences in Ecom-Offers, making selected temporal tables well specified for irregular time-series architectures.

2. Protocol Interface and Transfer Gap

2.1. A Unified Protocol Interface

Let q denote a prediction query, which may include an entity, a timestamp, a target channel, or a prediction horizon. A temporal prediction protocol $\mathcal{P}$ specifies an evidence-packaging operator $\mathcal{A}_{\mathcal{P}}$ that maps the available data $\mathcal{D}$ and query q to the information set

$$\mathcal{I}_{\mathcal{P}}(q) = \mathcal{A}_{\mathcal{P}}(\mathcal{D}, q), \qquad \hat{y}(q) = f\big(\mathcal{I}_{\mathcal{P}}(q), q\big). \quad (1)$$

Changing $\mathcal{A}_{\mathcal{P}}$ changes the task specification, even when the raw records and architecture are unchanged. We decompose the information set into a history/context component and a current or known-ahead component,

$$\mathcal{I}(t) \triangleq \big(\mathcal{I}_{\text{back}}(t), \mathcal{I}_{\text{fore}}(t)\big), \quad (2)$$

where $\mathcal{I}_{\text{back}}$ contains admissible observed evidence, such as a causal look-back window or interpolation context, and $\mathcal{I}_{\text{fore}}$ contains covariates known at query time and possibly over a horizon.

Temporal tabular prediction. A temporal table represents each example as $(t_i, \boldsymbol{x}_i, y_i)$. The default protocol has no explicit temporal context window and takes the query-time information to be the current covariates, optionally with time:

$$\mathcal{I}_{\text{back}}(t_i) = \emptyset, \qquad \mathcal{I}_{\text{fore}}(t_i) = (t_i, \boldsymbol{x}_i). \quad (3)$$

Generalization is time-based: training uses earlier periods and testing uses later periods. Fine-grained histories are usually absorbed into missing values or engineered summaries.

Irregular time-series prediction. For an entity u, observations arrive at non-uniform timestamps. An extrapolation protocol admits a causal history

$$\mathcal{I}_{\text{back}}(u, t) = \{(t_{u,j}, y_{u,j}, \boldsymbol{x}_{u,j}) : t - L < t_{u,j} < t\}, \quad (4)$$

Table 2. **Comparison of irregular time series methods on temporal tabular benchmark (Rubachev et al., 2025).** The reported metric is performance percentage change, *i.e.*, a positive percentage change indicates that the method outperforms static MLP.

Settings	**Methods**	**Avg. Imp.**
I.i.d. Tabular	MLP (Gorishniy et al., 2022)	0.00%
Irregular Time Series	GRU-D (Che et al., 2018)	–5.06%
	CRU (Schirmer et al., 2022)	–8.41%
	ACSSM (Park et al., 2025)	–5.12%
	Neural Flows (Bilos et al., 2021)	–3.35%
	mTAND (Shukla & Marlin, 2021)	–35.14%
Temporal Tabular	MLP (+e) (Cai & Ye, 2025b)	+1.31%
	MLP (+m) (Cai & Ye, 2025a)	+2.09%

and future known covariates, when available, form $\mathcal{I}_{\text{fore}}(u, t)$. Interpolation instead admits observed, unmasked measurements around the query time while excluding the queried value itself. Irregular protocols therefore expose variable-length temporal contexts, often with asynchrony across variables and timestamp overlap across entities.

2.2. Protocol Changes and Compatibility

Temporal embedding (+e) changes the query representation by adding a learned representation of time. Temporal modulation (+m) changes how time affects feature representations. Context construction (+c) changes $\mathcal{A}_{\mathcal{P}}$ by restricting the support or candidate set to protocol-admissible temporal evidence. Entity recovery changes event grouping, and sequence-to-label recasting changes the atomic query unit. Cross-domain transfer is meaningful only if the resulting $\mathcal{I}_{\mathcal{P}}(q)$ exposes the evidence expected by the model class. For sequence or continuous-time models, ordered events should correspond to a coherent entity or source sequence. For fixed-row tabular models, variable-length histories must be represented through admissible context or summarized features. Thus a cross-domain experiment must specify whether it changes only f, or also changes $\mathcal{A}_{\mathcal{P}}$.

2.3. Direct Transfer as a Stress Test

The interface makes protocols comparable, but it does not make arbitrary format conversions valid. We first use direct conversions as stress tests before applying protocol recoveries in Sections 3 and 4.

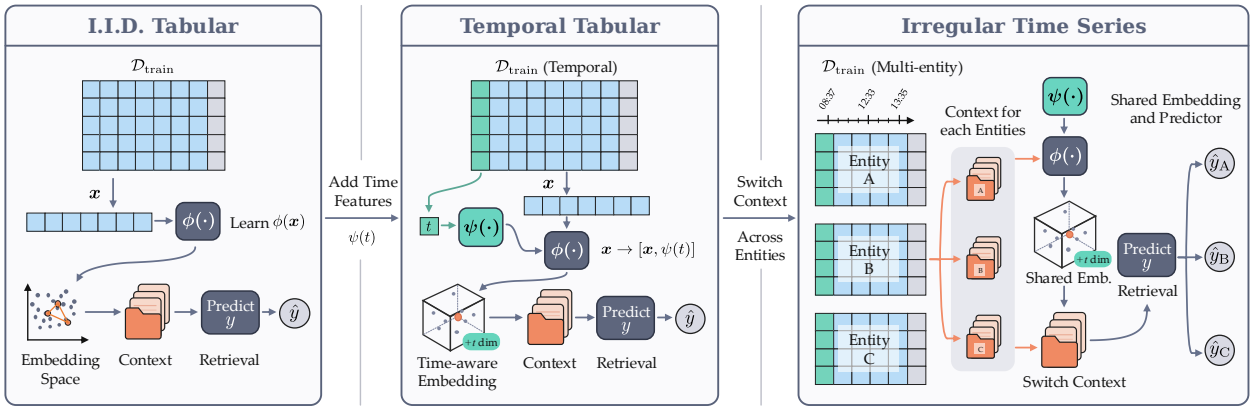

Figure 2. **A tabular lens on irregular time series.** Temporal embedding changes the query representation, while context construction changes the admissible evidence: the global candidate pool is repackaged into an entity-specific context defined by the ITS protocol.

Table 3. **Main results on the irregular time series datasets** Physionet (Silva et al., 2012) and USHCN (Menne et al., 2015). The metric is RMSE↓. The best performance is shown in **bold**, with the 2nd best result underlined.

Methods	Interpolation		Extrapolation		Avg. Rank
	Physionet↓	USHCN↓	Physionet↓	USHCN↓	
GRU-D	0.0968	0.0984	0.0942	0.1305	4.00
CRU	0.1068	0.0968	0.0997	0.1142	3.50
ACSSM	0.1075	0.0961	0.1046	0.1287	5.00
Latent ODE	0.1114	0.1266	**0.0920**	0.1283	4.50
Neural Flows	0.1043	0.1274	0.1029	0.1293	5.75
mTAND	**0.0810**	0.1009	0.0930	0.1193	2.75
ModernNCA (+ec)	0.0987	**0.0948**	0.1018	**0.1072**	**2.50**

Irregular models on temporal tables. We order timestamped table rows by time and feed them as event sequences, with row covariates treated as observed features. Table 2 reports average performance change on EO and WE relative to a static MLP baseline. The direct irregular time-series adaptations underperform this baseline, while simple temporal tabular variants, MLP (+e) and MLP (+m), improve over it. This does not show that irregular time-series architectures are intrinsically weak. Temporal tables can contain many records at the same coarse timestamp, often from different entities or independent events; under global row ordering, a recurrent or continuous-time model must explain transitions that may not correspond to any entity's dynamics. Section 4 tests whether recovering entity histories or source sequences reduces this mismatch.

Tabular models on irregular histories. The reverse transfer has a different obstacle. A fixed-row tabular predictor expects a covariate vector, possibly with time, whereas an irregular time-series query exposes a variable-length admissible context with entity-specific observation times and missingness. Padding, aggregation, or hand-crafted summaries can make this context tabular, but they decide which temporal relations remain visible. Section 3 instead treats the admissible history as a query-specific context for a context-based tabular predictor.

3. A Tabular Lens on Irregular Time Series

This section asks whether a context-based tabular predictor can be evaluated under irregular time-series protocols once $\mathcal{A}_\mathcal{P}$ exposes the proper evidence. Context-based tabular methods, including retrieval-based predictors such as TabR (Gorishniy et al., 2024) and ModernNCA (Ye et al., 2025), answer a query using a labeled support or candidate set. Let $q(t)$ be the query representation and $\mathcal{C}(t) = \{(q_j, y_j)\}_{j=1}^{n_t}$ be the context admitted by the protocol. A retrieval-style predictor embeds query and context items, computes similarity weights, and aggregates labels:

$$s_j(t) = -d(h_\theta(q(t)), h_\theta(q_j))/\tau,$$
$$\boldsymbol{w}_j(t) = \frac{\exp(s_j(t))}{\sum_{k=1}^{n_t}\exp(s_k(t))}, \quad \hat{y}(t) = \sum_{j=1}^{n_t} \boldsymbol{w}_j(t) y_j, \tag{5}$$

where h_θ is a learned representation, $d(\cdot,\cdot)$ is a distance, and $\tau > 0$ is a temperature. The exact readout may vary, but the dependence on an admissible context is protocol-level.

As shown in Figure 2, temporal tabular prediction first changes the query representation by attaching time,

$$q(t) = [\boldsymbol{x}_t, \psi(t)]. \tag{6}$$

This is the (+e) variant: it lets f adapt the covariate–target relationship across time while keeping $\mathcal{I}_{\text{back}}(t) = \emptyset$. Irregular time-series prediction further changes the admissible evidence. For an entity u and query time t, we replace the global candidate pool with the protocol-admitted entity context

$$\mathcal{C}_u^\mathcal{P}(t) = \{(\boldsymbol{x}_{u,j}, y_{u,j}, t_{u,j}) : j \in \mathcal{A}_\mathcal{P}(\mathcal{D}, u, t)\}. \tag{7}$$

For extrapolation, $\mathcal{A}_\mathcal{P}$ admits only causal observations before t; for interpolation, it admits observed unmasked measurements while excluding the masked target. This is the (+c) variant, and it recovers the history/context component of $\mathcal{I}_\mathcal{P}(q)$ without flattening a variable-length window into one row.

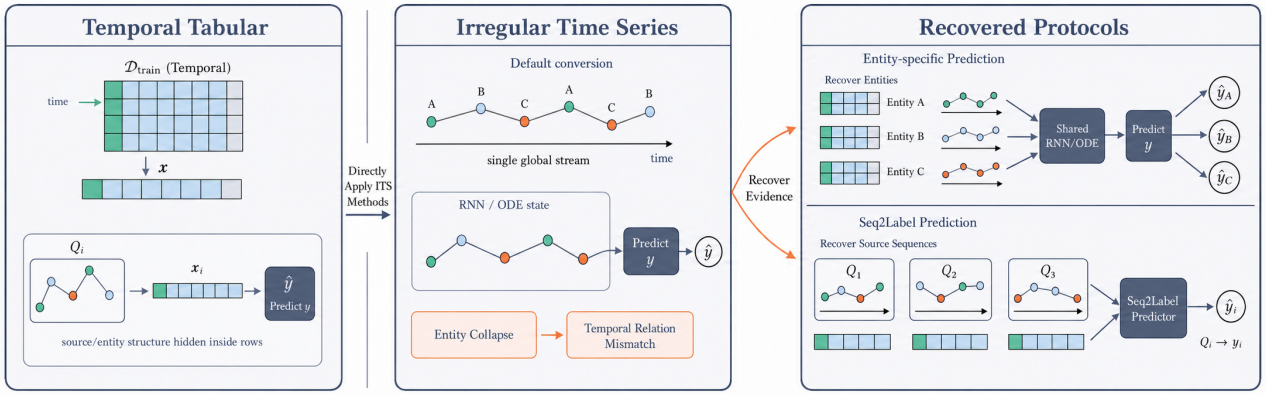

Figure 3. **Recovering time-series protocols from temporal tables.** Direct conversion can collapse entities or misplace temporal relations; protocol recovery restores entity histories or sequence-to-label examples.

We instantiate this protocol with ModernNCA, yielding ModernNCA (+ec) with temporal embedding (+e) and entity-specific context construction (+c). Table 3 shows that it is competitive on several irregular benchmarks, especially USHCN, although it does not uniformly dominate specialized time-series architectures. Component ablations are not well defined: without temporal embedding the query lacks temporal relations, and without context construction the candidate pool mixes entities unless one trains a separate model per entity. Additional PFN-style and running-time checks are in Section B.

4. Recovering Histories from Temporal Tables

Section 3 repackages irregular histories for a tabular retrieval interface. We now test the reverse direction on temporal tables whose raw records expose recoverable temporal structure. We do not claim that every temporal table is a time series; we ask whether the direct failures in Section 2.3 are reduced when $\mathcal{A}_{\mathcal{P}}$ restores histories hidden by tabulation.

Entity collapse occurs when rows from different entities are ordered as one stream, forcing the model state to jump across unrelated processes. For WE, whose rows are weather-station observations, we recover same-station histories while keeping the target, architecture, and station covariates unchanged. Station identity and location then support prediction instead of compensating for an incoherent global trajectory.

Temporal relation mismatch occurs when a row summarizes an underlying sequence, but a direct conversion asks the model to learn row-to-row dynamics. For EO, each row predicts coupon redemption from features that summarize the user's pre-offer transactions. We recast each coupon instance as a sequence-to-label example over those transactions; the label is unchanged, but the admissible evidence moves from offer-to-offer rows to the source sequence. Implementation details are in Section B.1.

Table 4 shows that these recoveries improve most irregular time-series runs on WE and all completed runs on EO. The best recovered EO runs slightly exceed the temporal tabular baseline, while WE still favors the best tabular model. Thus recovery reduces protocol mismatch without implying that time-series architectures universally dominate covariate-centric tabular modeling: EO benefits strongly from source-sequence recovery, whereas WE still leaves room for tabular models that exploit station covariates and static interactions. The result is diagnostic: cross-domain reuse becomes better specified when the recovered information set matches the model's assumptions. Our TabReD recovery analysis is limited to WE and EO because their raw data expose clear station-level histories and transaction-level sequences.

Table 4. **Protocol recovery on EO and WE.** Δ is Reframed–Default; positive is better for EO and negative is better for WE.

Data	Method	Def.	Refr.	Δ	MLP	+e	+m
EO↑	GRU-D	0.5732	0.5822	+0.0090			
	ACSSM	0.5631	0.5869	+0.0238			
	mTAND	0.4993	0.5885	+0.0892	0.5866	0.5877	0.5832
	CRU	0.5148	0.5731	+0.0583			
	Neural Flows	0.5779	**0.5914**	+0.0135			
WE↓	GRU-D	1.6533	**1.5941**	-0.0592			
	ACSSM	1.6286	1.5987	-0.0299			
	mTAND	2.3823	1.6274	-0.7549	1.5331	1.5363	1.5157
	CRU	1.6032	1.6045	+0.0013			
	Neural Flows	1.6131	1.6113	-0.0018			

5. Conclusion

We presented a protocol view of temporal prediction that makes query-time evidence explicit. It explains why direct transfer between temporal tables and irregular time series can fail through entity collapse or temporal relation mismatch, and why protocol recovery should precede model comparison. In both directions, the key intervention is to change the information set on which an architecture is evaluated. Our results therefore do not rank all temporal models; they show that forecasting comparisons should report the query unit, admissible histories, and entity structure alongside model class.

Impact Statement

This work aims to improve temporal prediction evaluation by making cross-domain comparisons better specified; risks arise in sensitive domains without validating distribution shift, privacy constraints, and subgroup performance.

References

Bilos, M., Sommer, J., Rangapuram, S. S., Januschowski, T., and Günnemann, S. Neural flows: Efficient alternative to neural odes. In *NeurIPS*, pp. 21325–21337, 2021.

Cai, H.-R. and Ye, H.-J. Feature-aware modulation for learning from temporal tabular data. *CoRR*, abs/2512.03678, 2025a.

Cai, H.-R. and Ye, H.-J. Understanding the limits of deep tabular methods with temporal shift. In *ICML*, pp. 6366–6386, 2025b.

Chang, C., Hwang, J., Shi, Y., Wang, H., Peng, W., Chen, T., and Wang, W. Time-imm: A dataset and benchmark for irregular multimodal multivariate time series. *CoRR*, abs/2506.10412, 2025.

Che, Z., Purushotham, S., Cho, K., Sontag, D., and Liu, Y. Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8:6085, 2018.

Ganesan, D., Ratnasamy, S., Wang, H., and Estrin, D. Coping with irregular spatio-temporal sampling in sensor networks. *Comput. Commun. Rev.*, 34(1):125–130, 2004.

Gorishniy, Y., Rubachev, I., and Babenko, A. On embeddings for numerical features in tabular deep learning. In *NeurIPS*, pp. 24991–25004, 2022.

Gorishniy, Y., Rubachev, I., Kartashev, N., Shlenskii, D., Kotelnikov, A., and Babenko, A. Tabr: Tabular deep learning meets nearest neighbors. In *ICLR*, 2024.

Hollmann, N., Müller, S., Purucker, L., Krishnakumar, A., Körfer, M., Hoo, S. B., Schirrmeister, R. T., and Hutter, F. Accurate predictions on small data with a tabular foundation model. *Nature*, 637:319–326, 2025.

Hoo, S. B., Muller, S., Salinas, D., and Hutter, F. From tables to time: Extending tabpfn-v2 to time series forecasting. *CoRR*, abs/2501.02945, 2025.

Liu, S.-Y., Cai, H.-R., Zhou, Q.-L., Yin, H.-H., Zhou, T., Jiang, J.-P., and Ye, H.-J. TALENT: A tabular analytics and learning toolbox. *J. Mach. Learn. Res.*, 26:226:1–226:16, 2025.

Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *ICLR*, 2019.

Menne, M., Williams Jr, C., Vose, R., and Files, D. Long-term daily climate records from stations across the contiguous united states, 2015.

Nederstigt, L. J., Aanen, S. S., Vandic, D., and Frasincar, F. FLOPPIES: A framework for large-scale ontology population of product information from tabular data in e-commerce stores. *Decis. Support Syst.*, 59:296–311, 2014.

Park, B., Lee, H., and Lee, J. Amortized control of continuous state space feynman-kac model for irregular time series. In *ICLR*, 2025.

Rajkomar, A., Oren, E., Chen, K., Dai, A. M., Hajaj, N., Hardt, M., Liu, P. J., Liu, X., Marcus, J., Sun, M., Sundberg, P., Yee, H., Zhang, K., Zhang, Y., Flores, G., Duggan, G. E., Irvine, J., Le, Q., Litsch, K., Mossin, A., Tansuwan, J., Wang, D., Wexler, J., Wilson, J., Ludwig, D., Volchenboum, S. L., Chou, K., Pearson, M., Madabushi, S., Shah, N. H., Butte, A. J., Howell, M. D., Cui, C., Corrado, G. S., and Dean, J. Scalable and accurate deep learning with electronic health records. *npj Digit. Medicine*, 1, 2018.

Rubachev, I., Kartashev, N., Gorishniy, Y., and Babenko, A. Tabred: A benchmark of tabular machine learning in-the-wild. In *ICLR*, 2025.

Rubanova, Y., Chen, T. Q., and Duvenaud, D. Latent ordinary differential equations for irregularly-sampled time series. In *NeurIPS*, pp. 5321–5331, 2019.

Schirmer, M., Eltayeb, M., Lessmann, S., and Rudolph, M. Modeling irregular time series with continuous recurrent units. In *ICML*, pp. 19388–19405, 2022.

Shukla, S. N. and Marlin, B. M. Multi-time attention networks for irregularly sampled time series. In *ICLR*, 2021.

Silva, I., Moody, G., Scott, D. J., Celi, L. A., and Mark, R. G. Predicting in-hospital mortality of icu patients: The physionet/computing in cardiology challenge 2012. In *2012 computing in cardiology*, pp. 245–248, 2012.

Ye, H.-J., Liu, S.-Y., Cai, H.-R., Zhou, Q.-L., and Zhan, D.-C. A closer look at deep learning methods on tabular datasets. *CoRR*, abs/2407.00956, 2024.

Ye, H.-J., Yin, H.-H., Zhan, D.-C., and Chao, W.-L. Revisiting nearest neighbor for tabular data: A deep tabular baseline two decades later. In *ICLR*, 2025.

A. Additional Experimental Setup

A.1. Training

All experiments are conducted on Linux using Python 3.11 and PyTorch 2.7.0. Experiments are run on machines equipped with NVIDIA RTX 4090 GPUs with 24 GB memory and four Intel Xeon Platinum 8352S CPUs. We use a unified batch size for all models on the irregular datasets and on TabReD: mTAND and GRU-D are trained with batch size 1 due to their high computational overhead and large GPU-memory footprint, while all other methods use batch size 128. We optimize all models with AdamW (Loshchilov & Hutter, 2019), apply early stopping with a patience of 16 epochs, and cap training at 200 epochs. Each method is trained with 15 different random seeds, and we report the average performance across runs.

For all models on the irregular datasets and TabReD, we adopt the same task-specific loss and evaluation metric. Following Chang et al. (2025), we first aggregate predictions and labels across all time steps and samples for each channel, and then average the resulting scores over channels; only mask-indicated valid entries are included. For classification, we use masked cross-entropy for training and masked AUC (higher is better) for evaluation. For regression, we use masked MSE for training and masked RMSE (lower is better) for evaluation. On TabReD, the target contains no missing values and there is a single prediction channel, so the masked objectives reduce to their standard counterparts.

A.2. Data Splitting and Evaluation Protocols

For USHCN (Menne et al., 2015) and PhysioNet (Silva et al., 2012), we follow the splitting protocol of Schirmer et al. (2022): we allocate 80% of sequences to training and 20% to testing, and further hold out 25% of the training split as a validation set for checkpoint selection. All final results are reported on the untouched test set. For TabReD, we use the same preprocessing, training, evaluation, and hyper-parameter tuning pipeline released by Cai & Ye (2025b); Ye et al. (2024); Liu et al. (2025).

We evaluate both interpolation and extrapolation on the same two datasets. **Interpolation:** given a subset of observations $\boldsymbol{x}_{\mathcal{S}}$ with $\mathcal{S} \subseteq \mathcal{T}$, we reconstruct the complete sequence $\boldsymbol{s}_{\mathcal{T}} = \boldsymbol{x}_{\mathcal{T}}$. We randomly mask a fixed 20% of time points in each sequence and infer the missing values from the remaining 80%. **Extrapolation:** we split the time axis into $\mathcal{T}_1 = \{t_0, \ldots, t_k\}$ and $\mathcal{T}_2 = \{t_{k+1}, \ldots, t_N\}$, and predict the full sequence on $\mathcal{T} = \mathcal{T}_1 \cup \mathcal{T}_2$ given only observations from $\mathcal{T}_1$. Following Schirmer et al. (2022), for PhysioNet we use the first 24 h of measurements to forecast the next 24 h; for USHCN we split each series into two equal halves and use the first half to predict the second.

B. Additional Results and Protocol Details

B.1. Protocol Details

Table 5 spells out the admissible evidence used in the main protocol transformations. The table makes explicit which observations are available at query time and which sources of information are excluded to preserve a valid prediction protocol. In particular, the recovered WE and EO settings differ from the direct TabReD conversion by restoring station-specific histories or pre-offer transaction sequences before applying time-series models.

Table 5. Admissible evidence in the main protocol transformations.

Experiment	Query	Admissible evidence	Excluded evidence
TabReD direct TS	table row	previous rows in global time order	future labels
WE recovered	station-time	same-station past rows	other stations, future rows
EO recovered	coupon offer	pre-offer transactions	post-offer transactions
ITS extrap.	entity-time	observed past segment	future observations
ITS interp.	masked point	unmasked observations	masked target value

Weather recovery. In WE, each row is a weather-station observation, and station identifiers or station attributes such as latitude and longitude are treated as ordinary covariates in the tabular benchmark. A direct time-series conversion orders all rows along the global timestamp axis, so observations from different stations become one artificial sequence. We recover the protocol by partitioning the table by station. For a query at station u and time t, the admissible input is the same-station history $\mathcal{H}_u(t) = \{(\boldsymbol{x}_{u,j}, y_{u,j}, t_{u,j}) : t_{u,j} < t\}$; the target and temporal architecture are unchanged.

Ecom-Offers recovery. EO predicts whether a user redeems an e-commerce coupon after receiving an offer. The tabular benchmark converts the user's variable-length transaction history into fixed-dimensional features and uses the coupon issue time as the timestamp. Directly sequencing coupon rows asks a time-series model to learn transitions between offer instances. We instead recover each user's transaction events before the offer time and use them as $\mathcal{Q}_i = \{(\boldsymbol{z}_{i,k}, \tau_{i,k})\}_{k=1}^{m_i}$, with transaction-derived event features and event times relative to the offer; the label remains coupon redemption.

B.2. PFN-Style Baselines

Tables 6 and 7 add PFN-style checks to the main results. These experiments are not intended to replace the main context-based instantiation; instead, they test whether the conclusions are specific to ModernNCA. TabPFN v2 (Hollmann et al., 2025) is strong on several temporal-table tasks but does not close the gap created by protocol mismatch, while TabPFN-TS (Hoo et al., 2025) remains below ModernNCA (+ec) on the irregular regression tasks considered here.

Table 6. PFN-style checks from the rebuttal experiments. TabPFN v2 is evaluated on TabReD, and TabPFN-TS is evaluated on irregular time-series regression tasks. These results are included to clarify that our main ModernNCA instantiation is not positioned as a universal best model.

Method	HI↑	EO↑	HD↑	SH↓	CT↓	DE↓	MR↓	WE↓
TabPFN v2	0.8215	0.5660	0.5000	0.2276	0.4858	0.5529	0.1673	1.7076
ModernNCA	0.9571	0.5712	0.8487	0.2526	0.4817	0.5523	0.1631	1.4977
ModernNCA (+c)	0.9575	0.6076	0.8488	0.2368	0.4810	0.5523	0.1628	1.4400

Table 7. TabPFN-TS on irregular time-series regression. P-I/U-I denote PhysioNet/USHCN interpolation, and P-E/U-E denote extrapolation.

Method	P-I↓	U-I↓	P-E↓	U-E↓
TabPFN-TS	0.1079	0.1091	0.1211	0.1163
ModernNCA (+ec)	0.0987	0.0948	0.1018	0.1072

B.3. Training Time

Table 8 reports training time relative to ModernNCA under the same experimental pipeline. The comparison complements the accuracy tables: the context-based tabular instantiation is not only competitive on selected irregular tasks, but also avoids the substantially higher training cost of several continuous-time and recurrent baselines.

Table 8. Average training time relative to ModernNCA.

Method	ModernNCA	Latent ODE	ACSSM	CRU	mTAND	Neural Flows
Relative time	100.00%	262.34%	144.43%	713.38%	365.96%	503.63%