AXIAL POINTER: GLOBAL-TO-LOCAL PREDICTION FOR MIXED-TYPE EVENT SEQUENCES

Anonymous authors
Paper under double-blind review

ABSTRACT

Real-world event sequences often record the next event as a complete set of interacting attributes, including irregular timing, numerical measurements, and categorical labels. Predicting that event is difficult because these output types behave differently, and because a categorical attribute may have thousands of possible values across a dataset even though only a small subset appears in one observed history. We introduce *Axial Pointer* (AP), which predicts all attributes from the same history and learns which earlier events provide useful evidence for each prediction. AP can reuse relevant categories or numerical values from that history while retaining the flexibility to predict when historical reuse is insufficient. In this way, AP uses a common source of historical evidence for numbers and categories without forcing these different outputs to be predicted in the same way. Across five heterogeneous transaction datasets and 23 predicted attributes, AP achieves the best or tied-best average performance on 21 attributes; it also performs best among the available learned comparison methods on all three evaluated categorical attributes with thousands of possible values.

1 INTRODUCTION

Many real-world sequences do not reduce an event to a timestamp and one label. A transaction, medical encounter, or online interaction may record irregular timing together with several numerical and categorical attributes (Osin et al., 2025). Next-event prediction is therefore a structured multi-output problem: the target is one complete heterogeneous record, not only a time and event label. Because its fields describe the same occurrence, patterns in one can provide useful historical context for another.

We call each fixed-schema field of that record an *event axis*; familiar examples are elapsed time, transaction amount, event type, and destination identity. Numerical axes admit distance, interpolation, and extrapolation, whereas categorical axes require choosing an identity. Irregular gaps affect which past events are relevant, dependencies run across time and axes, and categorical cardinalities can differ by orders of magnitude. A mixed-type model must share information across the record without treating all outputs as if they had the same geometry. Existing approaches address complementary parts of this problem. Neural temporal point processes model irregular timing and a discrete event label, or mark (Shchur et al., 2021; Bosser & Taieb, 2023), but do not by themselves specify a multi-field record. Sequential-tabular models preserve multiple fields, but that alone does not determine how unlike outputs should share historical evidence. One model must reconcile cross-field context, temporal dependence, and type-appropriate prediction.

A dataset may contain thousands of categories, but a person's recent history contains only a few. Conventional embedding tables and classifiers allocate parameters across the full vocabulary and become awkward as cardinality grows (Li et al., 2024). We call the values exposed by one observed prefix its *history-local support*, providing bias for events recurring. However, a category may be absent from the prefix, and a future number may lie outside its historical range. Purely history-restricted prediction is therefore also insufficient. Our organizing idea is to let each output axis identify relevant earlier events, then interpret that evidence according to its type. Axes make direct predictions when history alone is too restrictive. This is the organizing principle of *Axial Pointer* (AP). AP predicts all axes from the same history and couples fixed-width identity representation with prediction over the smaller set exposed by that history. The *global-to-local* principle revolves

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.

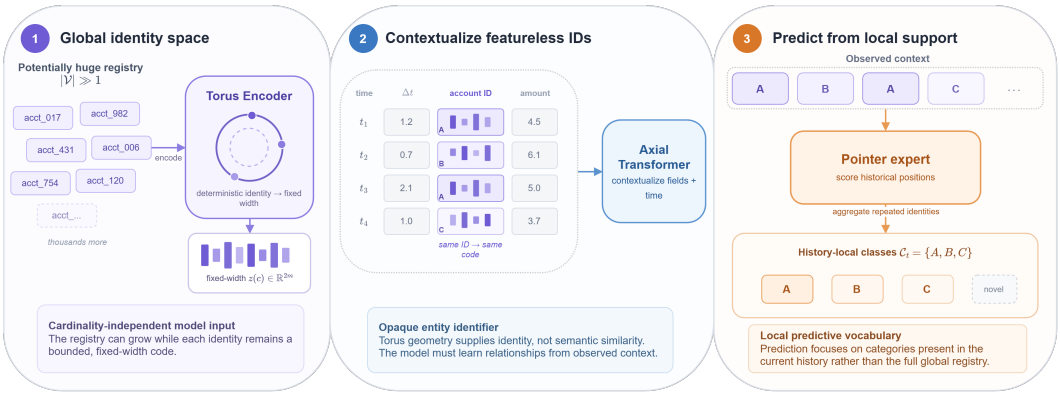

Figure 1: Global-to-local categorical prediction. Although an identity axis may draw from a large global set, one prefix exposes only a small candidate set. AP scores the historical events, combines evidence from repeated identities, and also represents the possibility that the next identity is absent from the prefix.

around representing identities in the full global category space while forming each prediction over the much smaller candidate set exposed by the current prefix. Figure 1 isolates the categorical case.

Contributions. Our main contributions are:

1. **Mixed-type prediction through shared historical evidence.** AP predicts one structured record from cross-field causal context while decoding numerical and categorical evidence in type-appropriate ways.
2. **A global-to-local treatment of large categorical spaces.** AP separates the large space needed to represent identities from the smaller history-local support useful for prediction, while retaining alternatives to historical reuse.
3. **Evidence across heterogeneous event schemas.** Across five transaction datasets and 23 predicted axes, AP has the best or tied-best mean on 21 axes; it also leads the available learned comparison on all three evaluated categorical axes with thousands of identities.

2 Related Work

Event-sequence models. Neural marked temporal point processes (MTPPs) model irregular event streams via the conditional distribution of the next event time and mark. RMTPP summarizes history with a recurrent state for timing and mark prediction, while the Neural Hawkes Process (NHP) uses a continuous-time LSTM to capture nonlinear excitation and inhibition among event types (Du et al., 2016; Mei & Eisner, 2017). Attention-based variants replace recurrent history compression with direct access to past events: SAHP introduces time-sensitive self-attention, THP targets long-range temporal dependence, and AttNHP reformulates NHP with a flatter attention architecture (Zhang et al., 2020; Zuo et al., 2020; Mei et al., 2021). IFTPP instead directly models the inter-event-time distribution rather than its intensity (Shchur et al., 2020). C-Diff performs joint multi-event forecasting through interacting diffusion processes over times and event types, whereas S2P2 uses continuous-time state-space dynamics with stochastic jumps and parallel scans (Zeng et al., 2024; Chang et al., 2026).

Pointers, copying, and recurrence. Pointer Networks reinterpret attention as an output distribution over input positions, yielding a variable-size output dictionary (Vinyals et al., 2015). Pointer-generator networks mix copying with ordinary vocabulary generation, allowing source tokens, including out-of-vocabulary words, to be reproduced while retaining novel generation (See et al., 2017). RepeatNet uses a repeat-explore mechanism that switches between selecting from session

history and predicting outside that repeated set (Ren et al., 2019). These works separate positional selection, copying, and repeat-versus-novel modeling as mechanisms for historical reuse.

Large and evolving categorical spaces. Large categorical vocabularies create distinct output and representation bottlenecks. Adaptive softmax reduces output cost by exploiting frequency imbalance and clustering classes to minimize expected computation (Joulin et al., 2017). DHE replaces embedding tables with deterministic multi-hash encodings followed by a learned network, supporting unseen identifiers without one learned vector per category (Kang et al., 2021). PHE targets evolving online vocabularies by treating hash embeddings probabilistically and applying Bayesian updates, yielding bounded parameterization, adaptation to new categories, and arrival-order invariance (Li et al., 2026). These approaches address complementary problems: output normalization, table-free representation, and online vocabulary growth.

Structured event vectors. Axial attention factorizes self-attention along tensor dimensions, reducing cost for multidimensional inputs while retaining broad receptive fields (Ho et al., 2019). Sequential tabular models exploit related structure across fields and time. TabFormer encodes fields within each row before modeling the row sequence, supporting BERT-style representation learning and GPT-style generation (Padhi et al., 2021). FATA-Trans separates static and dynamic fields and augments sequence order with inter-row time intervals (Zhang et al., 2023). Fieldy addresses coarse row-level hierarchy by combining row-wise and column-wise attention, allowing field-level interactions across rows (Azorin et al., 2024).

3 METHOD

3.1 MIXED EVENT-SEQUENCE PREDICTION

Let $\mathbf{x}_{1:T}$ be a chronologically ordered history with fixed-schema events $\mathbf{x}_t = (x_{t,1}, \ldots, x_{t,K})$ and sets of continuous and discrete axes $\mathcal{K}_{\text{cont}}$ and $\mathcal{K}_{\text{disc}}$. Given the complete prefix $\mathcal{H}_t = \mathbf{x}_{1:t}$, AP predicts every component of $\mathbf{x}_{t+1}$. All axes of the next event are predicted in parallel from this same causal prefix; none conditions on another, not-yet-predicted axis of $\mathbf{x}_{t+1}$.

AP supports two ways to turn that shared context into a prediction. The first scores positions in the prefix and reuses the categorical or numerical values found there. We call the learned axis-specific predictor the *pointer expert*, its complete computation route the *pointer path*, and the set of historical positions it scores the *pointer interface*. The second predicts directly from the contextual state without restricting its output to values in the prefix; we call it the *direct expert* and its route the *direct path*. An expert is a complete prediction mechanism, whereas a path is the route through that expert. When both experts are available, training retains both and validation ultimately deploys one. The mechanism that manages this competition is the *router*; it selects rather than mixes their predictions.

For a continuous axis k, preprocessing fits a standardized Yeo–Johnson transform g_k on training values only and maps raw values to $z_{t,k} = g_k(x_{t,k})$; for a discrete axis, an axis-local codec maps labels to namespaced identities. At cutoff t, the pointer interface is the candidate-position set $\mathcal{I}_t = \{1, \ldots, t\}$. The pointer path can retrieve a repeated discrete class or interpolate continuous values from these positions; the discrete mechanism also accounts for a target absent from the prefix.

Training uses every valid adjacent pair as a next-event example, contributing one loss per axis. All scoreable cutoffs share one causal forward pass; at inference, the last prefix state is queried and the selected experts are applied in parallel. Figure 2 summarizes this architecture, from mixed-type history encoding through axial causal context to parallel pointer and eligible direct experts, followed by validation-selected deployment.

3.2 MIXED TOKENIZATION AND AXIAL CAUSAL CONTEXT

For a discrete cell (t, k), a learned affine map projects a deterministic, fixed-width identity code to the model's hidden width D; the code and its registry are defined in Section 3.4. For a continuous cell, a shared two-layer network encodes $z_{t,k}$ while conditioning on a learned axis embedding.

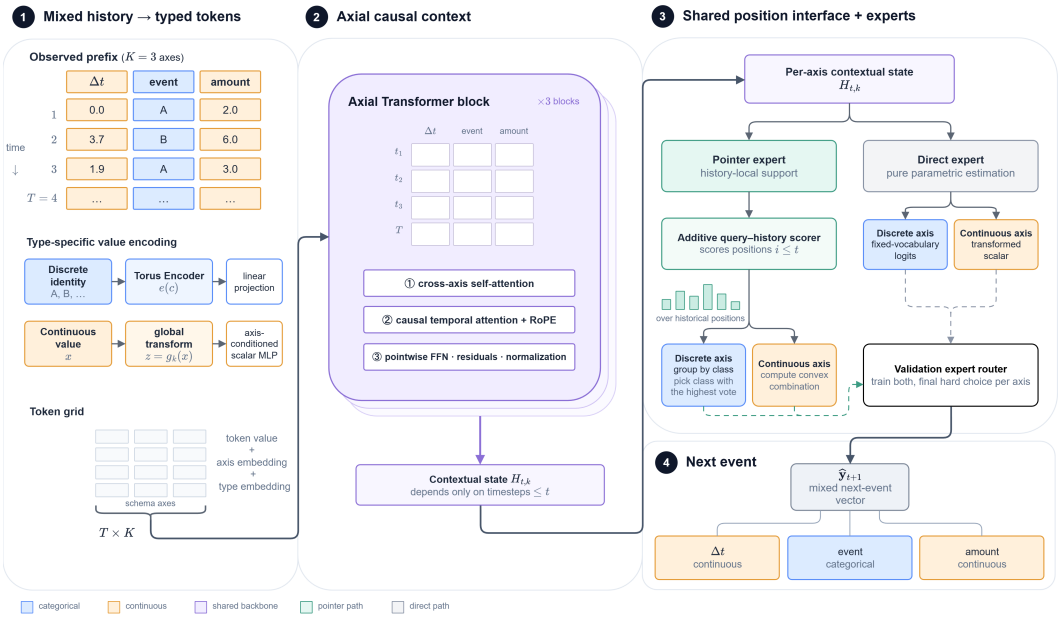

Figure 2: Overview of AP. Mixed-type histories become a typed token grid; axial causal blocks produce per-axis contextual states. Every axis feeds the shared historical-position interface of its pointer expert and, when eligible, a direct expert. Validation selects one expert per axis for deployment before the next event is decoded.

Adding learned axis and type embeddings gives an initial token $\mathbf{h}_{t,k}^{(0)} \in \mathbb{R}^D$; the resulting event-axis tokens form a time–axis grid.

These two grid dimensions explain the name *Axial Pointer*. *Axial* refers to axial attention, which factors attention across the event-axis and time dimensions rather than attending densely over the flattened grid. *Pointer* refers to the output interface formalized next: each axis scores positions in the observed history and decodes from those candidate positions. Thus the name identifies both the factored contextual backbone and the history-position prediction mechanism.

Each Transformer block (Vaswani et al., 2017) is an axial block. It first applies noncausal self-attention across all within-event token states. This permits cross-field interaction without imposing an arbitrary order on simultaneous fields. It then applies causal temporal self-attention, with rotary positions (Su et al., 2024), independently along each token stream, followed by one pointwise feed-forward network. Temporal attention permits information only from positions $i \leq t$, so the final state $\mathbf{h}_{t,k}$ depends on complete events $\mathbf{x}_{1:t}$ and not on future rows. Compared with dense attention over the flattened grid, one block has attention-score cost $O(TK^2 + KT^2)$, omitting batch and feature factors common to both expressions.

3.3 A SHARED HISTORICAL-POSITION POINTER INTERFACE

The pointer interface gives every axis the same variable-size set of candidates: positions in the observed history. Let $\mathbf{h}_{j,k}^{\mathrm{ptr}} \in \mathbb{R}^D$ denote the pointer expert's state for axis k at position j. For each axis k, an additive scorer compares the query state at cutoff t with every candidate state at position $i \leq t$:

$$s_{tki} = \mathbf{v}_k^\top \tanh\left(\mathbf{h}_{t,k}^{\mathrm{ptr}} \mathbf{W}_k^Q + \mathbf{h}_{i,k}^{\mathrm{ptr}} \mathbf{W}_k^C + \mathbf{b}_k\right). \tag{1}$$

Here $\mathbf{W}_k^Q, \mathbf{W}_k^C \in \mathbb{R}^{D \times P}$ and $\mathbf{v}_k, \mathbf{b}_k \in \mathbb{R}^P$ are learned axis-specific parameters, and P is the pointer hidden width. Future and padded positions are masked. Discrete axes combine these scores by observed class; continuous axes normalize them over positions.

3.3.1 DISCRETE PREDICTION FROM REPEATED CLASSES AND PREFIX ABSENCE

The same discrete class may occur at several historical positions. The pointer expert first combines the scores of every occurrence so that prediction chooses a class rather than an arbitrary duplicate position. For a discrete axis k, let $c_{i,k}$ be the class observed at position i, $\mathcal{C}_{t,k} = \{c_{i,k} : i \le t\}$ the set of prefix-observed classes, and $\mathcal{I}_{t,k}(a) = \{i \le t : c_{i,k} = a\}$ the positions at which class a occurs. Axial Pointer converts occurrence scores into the class score

$$u_{t,k,a} = \mathrm{logsumexp}_{i \in \mathcal{I}_{t,k}(a)} s_{tki} - \alpha_k \log |\mathcal{I}_{t,k}(a)|, \qquad \alpha_k = 2\,\mathrm{sigmoid}(\gamma_k) \in (0, 2). \tag{2}$$

This class marginalization uses every occurrence of a candidate class. Here γ_k is a learned axis-specific scalar, so the correction can retain, cancel, or reverse the log-count advantage from repeated occurrences. A separate learned affine map produces a score for the alternative that the target class is absent from the prefix. We call this alternative the *history-novel* outcome and denote its logit by $\eta_{t,k}$. A single softmax over the class and novelty scores defines probabilities for all prefix-observed classes and that outcome:

$$(\{\pi_{t,k,a}\}_{a \in \mathcal{C}_{t,k}}, \pi_{t,k,\mathrm{new}}) = \mathrm{softmax}(\{u_{t,k,a}\}_{a \in \mathcal{C}_{t,k}}, \eta_{t,k}). \tag{3}$$

Training applies class-level cross-entropy to the observed class or the novel outcome, so duplicate positions do not create ambiguous labels. At inference, the highest-probability joint outcome selects a prefix-observed class or signals that the target is absent from the prefix. History novelty concerns only this prefix: the class may occur elsewhere in the training data, so the outcome does not mean that it is globally unseen. The reserved unknown ID in the category codec is also separate: it is an encoding device for a raw label outside the training vocabulary, not a label whose prediction is correct by itself. Evaluation decodes categorical predictions to raw labels and compares them with the target labels using exact equality. Selecting novelty without identifying the held-out raw label therefore receives no accuracy credit.

3.3.2 CONTINUOUS PREDICTION BY WEIGHTED HISTORICAL VALUES

For a continuous axis, the pointer expert turns its position scores into nonnegative weights that sum to one, then predicts a weighted average of the historical transformed values. Formally,

$$p_{tki} = \frac{\exp s_{tki}}{\sum_{j \le t} \exp s_{tkj}}, \qquad \widehat{z}_{t+1,k} = \sum_{i \le t} p_{tki} z_{i,k}. \tag{4}$$

This weighted average is a convex, or *barycentric*, prediction. It cannot extrapolate beyond the transformed historical range; g_k^{-1} maps the prediction back to raw units. The continuous pointer expert is trained directly on its decoded value with transformed-space squared error,

$$\mathcal{L}_{t,k}^{\mathrm{ptr}} = \left(\widehat{z}_{t+1,k} - z_{t+1,k}\right)^2. \tag{5}$$

3.4 FIXED-WIDTH DETERMINISTIC IDENTITIES

The pointer path decodes from values in the observed history, but its discrete inputs must still identify categories drawn from the global space. A learned embedding table would add one input vector for every registered category. AP instead maintains a *registry*: an ordered mapping from each axis-namespaced category key c to a stable nonnegative index $n(c)$. The *Torus Encoder* deterministically converts that index into a bounded, fixed-width identity code, without asserting semantic similarity between nearby codes. For the first m primes $p_1, \ldots, p_m$, let $\omega_j = \sqrt{p_j} \bmod 1$ and $\theta_{n,j} = 2\pi(n\omega_j \bmod 1)$, and define

$$\mathbf{e}_n = \frac{1}{\sqrt{m}}(\cos\theta_{n,1}, \sin\theta_{n,1}, \ldots, \cos\theta_{n,m}, \sin\theta_{n,m}) \in \mathbb{R}^{2m}. \tag{6}$$

A learned map then projects this identity code to model width D.

Theorem 1 (Exact-arithmetic properties of the Torus Encoder)**.** *Fix $m \ge 1$ and assign registry indices in first-seen order, extending the registry only by appending new identities. In exact real arithmetic, $\mathbf{e}_n$ (i) always lies in the fixed space $\mathbb{R}^{2m}$, independent of registry cardinality; (ii) has unit norm; and (iii) is unchanged for every existing identity after registry extension. Moreover, (iv) if any ω_{j_*} is irrational, then $n \mapsto \mathbf{e}_n$ is injective on $\mathbb{N}$ (equivalently, the orbit has no nonzero finite period). The choice $\omega_j = \sqrt{p_j} \bmod 1$ satisfies this condition for every coordinate.*

Appendix A.1 proves the theorem and gives its numerical qualifications. In particular, it implies neither a positive minimum separation between codes nor collision-freedom or aperiodicity in finite precision. Operationally, append-only registry growth preserves existing codes and assigns each new identity the same fixed-width representation. Together with the history-local categorical decoder, this lets a *pointer-routed axis*—an axis deployed through its pointer expert—admit new registered identities without resizing learned per-class representations or pointer-path output structures. The ordered key–index registry still grows with cardinality; this is therefore a path-specific lifecycle property, not a whole-system efficiency guarantee. Appendix B.6 gives the complete accounting.

3.5 Adaptive pointer/direct routing

Because historical reuse is not always appropriate, every continuous axis and every discrete axis with at most 256 known training categories jointly trains its pointer and direct experts; larger categorical axes are *pointer-only*, meaning that they train and deploy only the pointer expert. We call an eligible axis that trains both a *dual-expert axis*. The continuous direct expert predicts an unconstrained scalar in transformed space, while the categorical direct expert predicts logits over the known training vocabulary and has no separate history-novel outcome. Validation compares the two experts per axis, and the router makes a final hard selection that deploys exactly one. This is selection rather than prediction mixing.

Both experts receive the same causal backbone state through private residual adapters. If $w_{k,e}$ is the router weight for expert $e \in \{\mathrm{ptr}, \mathrm{dir}\}$, define the gradient-routing operation by

$$\mathcal{R}_{w_{k,e}}(\mathbf{h}) = \mathrm{stopgrad}(\mathbf{h}) + w_{k,e}\big(\mathbf{h} - \mathrm{stopgrad}(\mathbf{h})\big) \tag{7}$$

where $\mathrm{stopgrad}$ is the identity on forward values and has zero derivative. Thus $\mathcal{R}_{w_{k,e}}$ preserves the forward state while scaling that expert's gradient into the shared backbone. Both native objectives are optimized on every eligible axis: transformed-space squared error for each continuous expert, or class-level pointer cross-entropy and fixed-vocabulary cross-entropy for the pointer and direct categorical experts, respectively. Private expert parameters remain trained, and router weights do not form a prediction mixture or remove either private loss.

Training takes one global mean over scoreable episode–cutoff and axis pairs: every axis contributes its native pointer loss, and each dual-expert axis also contributes its native direct loss, without renormalization by the number of active experts. For discrete pointers, the class-level weighting detailed in Appendix B.4 affects only the pointer loss and does not reweight the direct categorical loss. These weights affect optimization only; held-out expert selection uses unweighted raw-unit MAE or accuracy, and the reported categorical metric remains exact raw-label accuracy.

After distributed validation, the router records each expert's held-out error and smooths these metrics into per-axis evidence. For uncommitted axes, the relative evidence determines soft weights for gradient flow; these weights do not mix predictions. Validation compares both experts, while checkpoint selection and inference use a single hard per-axis deployment policy. A sustained winner can be committed, a sufficiently improved challenger can reopen the choice during a bounded window, and final locking selects the lower-error expert.

4 Experimental Setup

4.1 Task, datasets, and splits

Our task is conditional, teacher-forced mixed-type next-event prediction. For every scored cutoff t, each method receives the true prefix $\mathbf{x}_{1:t}$ and predicts every axis of $\mathbf{x}_{t+1}$ in parallel, without observing the target or future suffix. This asks how well a model predicts one next event given an observed history. It is distinct from long-horizon event-sequence forecasting, which evaluates multiple future events and may use free-running generation; method rankings need not agree between these tasks (Karpukhin et al., 2026). Autoregressive rollout is therefore a complementary evaluation for a different predictive question, not a prerequisite for evaluating the task studied here.

The evaluation suite spans five heterogeneous transaction schemas: IBM-CC and IBM-AML (Altman, 2022), MBD-Mini (Mollaev et al., 2025), The Complete Journey (Boehmke & Mortimer, 2025), and SynSEPA (Bajaj, 2026). A model event is one source transaction row in every dataset

except The Complete Journey, where product rows are aggregated into one household basket. Every schema predicts elapsed time together with its remaining continuous and categorical attributes. The complete mapping from datasets to entity histories and predicted axes is given in Appendix B.1, Table 2.

All episodes are fixed 50-event subsequences. Training, validation, and test are formed under dataset-specific split rules, with held-out targets evaluated after sufficient context. This length deliberately emphasizes a short-context, cold-start regime; its scope and implications for AP are discussed in Appendix B.2. Exact window construction, split membership, counts, and leakage controls are in Appendix B.1.

4.2 Methods and selection

The main comparison contains Axial Pointer (AP), SAHP (Zhang et al., 2020), RMTPP (Du et al., 2016), THP (Zuo et al., 2020), S2P2 (Chang et al., 2026), NHP (Mei & Eisner, 2017), AttNHP (Mei et al., 2021), IFTPP (Shchur et al., 2020), and C-Diff (Zeng et al., 2024). All receive the same episodes and predict the complete schema; necessary mixed-type adaptations are documented in Appendix B.3. AttNHP and C-Diff could not complete on SynSEPA because their vocabulary-sized categorical states exceeded A40 GPU VRAM for its 311,543-class `beneficiary_iban` vocabulary. Completed final runs use a fixed optimization budget and 5 random seeds, and hyperparameter selection is performed separately for each dataset–method pair. Complete optimization settings, seed assignments, checkpointing, HPO spaces, dataset-specific method coverage, and AP's routing schedule are deferred to Appendix B.4 and Appendix B.5; the routing mechanism is described in Section 3.

4.3 Metrics

We evaluate continuous axes with raw-unit MAE and discrete axes with exact raw-label accuracy. Continuous predictions are inverse-transformed before scoring so that MAE remains in each axis's declared physical unit; categorical predictions are decoded to raw labels and scored by exact equality. We report every completed method and axis separately as the mean and sample standard deviation over five independent final runs. Mean rank across equally weighted axes and datasets provides an aggregate cross-axis summary for AP and the learned baselines. Appendix B.2 gives the detailed metric rationale, including the relationship to likelihood, long-horizon rollout, class-balanced alternatives, and the causal prefix reference.

5 Results

5.1 Predictive comparison

The predictive comparison reports raw-unit mean absolute error for continuous axes and exact raw-label accuracy for discrete axes. Table 1 summarizes AP's rank and margin relative to the learned-method comparison across all five datasets. The complete per-axis results are reported in Appendix C.1. Appendix Tables 13 and 14 extend the rank comparison to all learned methods for continuous and discrete axes, respectively.

Table 1 separates continuous and discrete axes so that heterogeneous raw units are never averaged. A positive continuous margin is AP's relative MAE reduction from the strongest baseline, whereas a positive discrete margin is its accuracy gain in percentage points; negative margins identify axes on which another learned method is better.

AP has the best or tied-best seed-aggregated mean on 21 of 23 axes. It has the best mean rank among methods for both axis types on IBM-CC, IBM-AML, MBD-Mini, and SynSEPA. The only exceptions occur on The Complete Journey. On SynSEPA, AP's continuous margins range from $+0.1\%$ to $+0.6\%$, and its discrete margins range from approximate parity to $+5.9$ percentage points. On The Complete Journey (Boehmke & Mortimer, 2025), AP is strongest on elapsed time and number of items, but trails the best baseline on total sales by 2.3% relative MAE and on store identity by 3.1 accuracy points. Thus the full comparison supports competitiveness across mixed schemas.

Table 1: Compact results summary. Rank is computed for AP from each axis-seed pair among the learned methods in the comparison. Rank cells report the mean and sample SD over the axis-seed observations. Each margin compares AP with the strongest non-AP learned baseline on the same axis; continuous margins are relative MAE reductions and discrete margins are accuracy percentage-point gains. Overall rank and SD use equal-dataset weighting of the axis-seed observations, while Overall margin extrema pool axes.

	Continuous				Discrete			
Dataset	Axes	Rank ↓	Min margin ↑	Max margin ↑	Axes	Rank ↓	Min margin ↑	Max margin ↑
IBM-CC	2	**1.00 ± 0.00**	+1.8 %	+2.4 %	2	**1.00 ± 0.00**	+0.4 *pp*	+2.5 *pp*
IBM-AML	2	**1.00 ± 0.00**	+12.1 %	+31.9 %	3	**1.00 ± 0.00**	+0.3 *pp*	+20.2 *pp*
MBD-Mini	2	**1.00 ± 0.00**	≈ +0.0 %	+2.9 %	4	**1.00 ± 0.00**	+0.1 *pp*	+0.3 *pp*
The Complete Journey	3	2.73 ± 2.63	−2.3 %	+4.1 %	1	6.80 ± 0.45	−3.1 *pp*	−3.1 *pp*
SynSEPA	2	1.20 ± 0.42	+0.1 %	+0.6 %	2	2.00 ± 1.33	≈ +0.0 *pp*	+5.9 *pp*
Overall	**11**	**1.39 ± 1.35**	−2.3 %	+31.9 %	**12**	**2.36 ± 2.35**	−3.1 *pp*	+20.2 *pp*

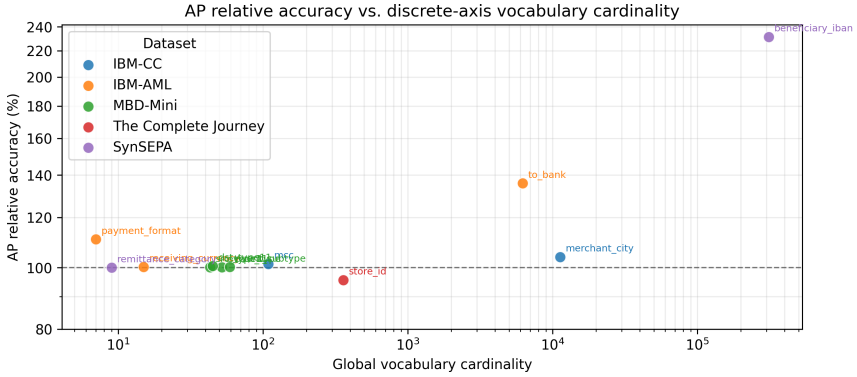

Figure 3: AP's relative accuracy across the 12 discrete axes from five datasets as a function of global training-vocabulary cardinality. Each point reports AP's seed-aggregated accuracy as a percentage of the strongest non-AP baseline on the same axis; 100% denotes parity, and values above 100% denote an AP advantage. Both axes use logarithmic scales. Colors identify datasets and labels identify axes.

The dataset-balanced summary gives AP overall ranks of 1.39 ± 1.35 on continuous axes and 2.36 ± 2.35 on discrete axes. The corresponding next-best overall ranks are 3.06 ± 1.24 for NHP on continuous axes and 3.00 ± 1.37 for NHP on discrete axes. Appendix C.2 reports the complete method comparison.

AP leads on high-cardinality axes. The clearest empirical evidence for AP's global-to-local design comes from three high-cardinality categorical axes: IBM-CC *merchant city*, IBM-AML *to bank*, and SynSEPA *beneficiary IBAN*. Their known training-label spaces contain approximately 11.3k, 6.2k, and 311.5k categories, respectively, while each held-out prediction is made from a prefix of at most 49 events. AP attains the best learned-method mean on all three (Appendix Tables 8, 9, and 12). Using the displayed means, its accuracy exceeds the strongest baseline by approximately 2.5, 20.2, and 5.9 percentage points, respectively. On these axes, the globally registered identity space is much larger than the support exposed by one finite history, and AP can predict over those history-local identities rather than score the full global vocabulary. Because the axes are pointer-only by design, the results support the practical suitability of the history-local path rather than a pointer-versus-direct routing comparison.

Figure 3 places that focused comparison within all 12 discrete axes. Most lower- and intermediate-cardinality axes cluster near parity, while The Complete Journey *store id* falls below parity. Within the highlighted trio, the advantage is largest for IBM-AML *to bank*, followed by SynSEPA *beneficiary IBAN*, and smaller for IBM-CC *merchant city*. Thus categorical prediction alone does not guarantee a pointer advantage.

AP deploys both expert types across axes. The deployed-expert annotations in Appendix Tables 8–12 show that every dataset deploys pointer and direct experts across its axes. Pointer deployments

appear on both continuous and categorical axes, while other axes select the direct expert. MBD-Mini deploys the pointer for amount but the direct expert on its other five axes; The Complete Journey uses the direct expert for elapsed time and pointers for its remaining axes. This variation is a practical motivation for retaining both geometries in one system, but the results also show that selecting a pointer does not ensure the strongest comparison mean: the pointer-routed store and total-sales axes in The Complete Journey are counterexamples. The predictive comparison evaluates the deployed system; the routing intervention is examined separately by the expert-competition ablation below.

Expert competition often beats single-expert ablations. Across matched axis-seed comparisons, regular AP has a win rate of 68.4% against direct-only AP and 67.6% against pointer-only AP. Both averages favor competition, but neither advantage holds on every dataset. Regular AP falls below a 50% win rate against both single-expert ablations only on The Complete Journey; SynSEPA favors regular AP in both comparisons. Appendix C.4 reports the complete per-axis results and discusses further.

Overall, the results support AP as a competitive mixed-type model while sharpening the limits of the empirical claim. IBM-CC, IBM-AML, MBD-Mini, and SynSEPA show uniformly first-ranked aggregate means, supplying the broad context for the focused intended-regime evidence above. Near parity on SynSEPA remittance category and reversals on The Complete Journey show that gains can shrink or reverse when target semantics and output geometry change. Appendix C.5 reports model-size and computational measurements separately from these predictive comparisons.

Limitations. AP deliberately relies on history-local support. On pointer-only categorical axes, it can select a prefix-observed label or detect that the target is novel, but it cannot identify an unseen external label; exact raw-label accuracy gives no credit for novelty detection alone. Pointer-based continuous predictions are confined to the transformed historical range. Direct experts relax these constraints on eligible axes by extrapolating continuous values or scoring a fixed training vocabulary, while larger categorical axes remain pointer-only. Theorem 1 assumes exact arithmetic and does not guarantee unlimited collision-free Torus codes in finite precision.

6 Conclusion

Axial Pointer separates the space needed to represent an identity from the support needed to predict its next value. AP operationalizes this global-to-local principle with an axial backbone and a shared interface over historical positions. Discrete pointers aggregate historical positions into class probabilities, continuous pointers interpolate observed values, and validation-selected direct experts preserve flexibility. This provides a common predictive coordinate system while preserving type-appropriate output geometries. Fixed-width Torus codes further remove learned per-class representations from categorical pointer paths.

Across five heterogeneous datasets, AP attains the best or tied-best mean on 21 of 23 axes, with the two remaining axes concentrated in The Complete Journey. Its best performance on all three high-cardinality, pointer-only categorical axes supports the motivating regime, while the dataset-specific method coverage qualifies the breadth of that evidence. Every dataset deploys both pointer and direct experts across its axes, and the fixed-expert ablations are also heterogeneous: competition is favored overall. Together, the results support history-local prediction as a strong option within a flexible mixed-type system rather than as a uniformly superior output rule.

These findings open several directions for making AP more efficient and expressive. Sparse retrieval could reduce pointer scoring costs over long histories, while compressed registry state and adaptive Torus representations could support larger, evolving categorical spaces. Hybrid continuous experts could combine history-local interpolation with extrapolation beyond observed values. Evaluating these extensions under controlled registry growth and longer sequences would connect AP's structural properties to deployment-scale efficiency. More broadly, extending the shared positional interface to new modalities, longer contexts, and open-ended registries offers a path toward event models that accommodate diverse output spaces without requiring a separate representational strategy.

AI USE STATEMENT

We used generative AI tools to support software and infrastructure development, develop proof sketches, assist with writing, editorial revision and verification, conduct literature surveys, and support general learning. We did not use generative AI tools for core design ideation, experiment design, citation or BibTeX creation, or qualitative or quantitative analysis. All AI-generated outputs were comprehensively checked and reviewed by the authors, who remain responsible for the final content. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The accompanying anonymous supplementary code and configuration release contains the Axial Pointer model, adapted baselines, preprocessing and evaluation pipelines, and experiment scripts used to produce the reported results. The main paper describes the task and evaluation protocol in Section 4, the model and objectives in Section 3, and the reported results in Section 5. Appendix B.1 documents dataset construction, preprocessing, splits, transforms, and persisted artifacts; Appendices B.3–B.4 describe the baseline adaptations, optimization, hyperparameter selection, and checkpoint selection; and Appendix B.5 specifies the routing protocol. The assumptions and complete proofs for the theoretical claims are given in Appendix A, while additional comparisons and efficiency measurements appear in Appendix C. Together, these materials provide the implementation, data-processing decisions, training and selection procedures, theoretical assumptions, and result-generation details needed to reproduce the study.

REFERENCES

Erik Altman. Synthesizing credit card transactions. In *Proceedings of the Second ACM International Conference on AI in Finance*, ICAIF '21, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450391481. doi: 10.1145/3490354.3494378. URL https://doi.org/10.1145/3490354.3494378.

Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, Geeta Chauhan, Anjali Chourdia, Will Constable, Alban Desmaison, Zachary DeVito, Elias Ellison, Will Feng, Jiong Gong, Michael Gschwind, Brian Hirsh, Sherlock Huang, Kshiteej Kalambarkar, Laurent Kirsch, Michael Lazos, Mario Lezcano, Yanbo Liang, Jason Liang, Yinghai Lu, CK Luk, Bert Maher, Yunjie Pan, Christian Puhrsch, Matthias Reso, Mark Saroufim, Marcos Yukio Siraichi, Helen Suk, Michael Suo, Phil Tillet, Eikan Wang, Xiaodong Wang, William Wen, Shunting Zhang, Xu Zhao, Keren Zhou, Richard Zou, Ajit Mathews, Gregory Chanan, Peng Wu, and Soumith Chintala. PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. In *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24)*. ACM, April 2024. doi: 10.1145/3620665.3640366. URL https://docs.pytorch.org/assets/pytorch2-2.pdf.

Raphael Azorin, Zied Ben Houidi, Massimo Gallo, Alessandro Finamore, and Pietro Michiardi. Fine-grained attention in hierarchical transformers for tabular time-series. *arXiv preprint arXiv:2406.15327*, 2024.

Gaurav Bajaj. Synsepa: A synthetic sepa instant payment dataset for app fraud detection research, 2026. URL https://huggingface.co/datasets/EpiphanyTech/SynSEPA. Generated as part of the SEPAGen project. Calibrated against EBA-ECB 2025 Payment Fraud Report.

James Bergstra, Brent Komer, Chris Eliasmith, Dan Yamins, and David D Cox. Hyperopt: a python library for model selection and hyperparameter optimization. *Computational Science & Discovery*, 8(1):014008, 2015.

Brad Boehmke and Steven M. Mortimer. *completejourney: Retail Shopping Data*, 2025. URL https://github.com/bradleyboehmke/completejourney. R package version 1.1.1.

Tanguy Bosser and Souhaib Ben Taieb. On the predictive accuracy of neural temporal point process models for continuous-time event data. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=3OSISBQPrM.

George EP Box and David R Cox. An analysis of transformations. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 26(2):211–243, 1964.

Yuxin Chang, Alex Boyd, Cao Danica Xiao, Taha Kass-Hout, Parminder Bhatia, Padhraic Smyth, et al. Deep continuous-time state-space models for marked event sequences. *Advances in neural information processing systems*, 38:162672–162713, 2026.

Nan Du, Hanjun Dai, Rakshit Trivedi, Utkarsh Upadhyay, Manuel Gomez-Rodriguez, and Le Song. Recurrent marked temporal point processes: Embedding event history to vector. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1555–1564, 2016.

Sylvain Gugger, Lysandre Debut, Thomas Wolf, Philipp Schmid, Zachary Mueller, Sourab Mangrulkar, Marc Sun, and Benjamin Bossan. Accelerate: Training and inference at scale made simple, efficient and adaptable. https://github.com/huggingface/accelerate, 2022.

Charles R Harris, K Jarrod Millman, Stéfan J Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, et al. Array programming with numpy. *nature*, 585(7825):357–362, 2020.

Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial attention in multidimensional transformers. *arXiv preprint arXiv:1912.12180*, 2019.

John D Hunter. Matplotlib: A 2d graphics environment. *Computing in science & engineering*, 9(3): 90–95, 2007.

Armand Joulin, Moustapha Cissé, David Grangier, Hervé Jégou, et al. Efficient softmax approximation for gpus. In *International conference on machine learning*, pp. 1302–1310. PMLR, 2017.

Wang-Cheng Kang, Derek Zhiyuan Cheng, Tiansheng Yao, Xinyang Yi, Ting Chen, Lichan Hong, and Ed H Chi. Learning to embed categorical features without embedding tables for recommendation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 840–850, 2021.

Ivan Karpukhin, Foma Shipilov, and Andrey Savchenko. Hotpp benchmark: Are we good at the long horizon events forecasting? *Neurocomputing*, pp. 132771, 2026.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International conference on learning representations*, 2015.

Aodong Li, Abishek Sankararaman, and Balakrishnan Murali Narayanaswamy. Probabilistic hash embeddings for online learning of categorical features. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 22841–22849, 2026.

Shiwei Li, Huifeng Guo, Xing Tang, Ruiming Tang, Lu Hou, Ruixuan Li, and Rui Zhang. Embedding compression in recommender systems: A survey. *ACM Computing Surveys*, 56(5):1–21, 2024.

Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=Skq89Scxx.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Wes McKinney et al. pandas: a foundational python library for data analysis and statistics. *Python for high performance and scientific computing*, 14(9):1–9, 2011.

Hongyuan Mei and Jason M Eisner. The neural hawkes process: A neurally self-modulating multivariate point process. *Advances in neural information processing systems*, 30, 2017.

Hongyuan Mei, Chenghao Yang, and Jason Eisner. Transformer embeddings of irregularly spaced events and their participants. In *International conference on learning representations*, 2021.

Dzhambulat Mollaev, Ivan Kireev, Mikhail Orlov, Alexander Kostin, Ivan Karpukhin, Maria Postnova, Gleb Gusev, and Andrey Savchenko. Multimodal banking dataset: Understanding client needs through event sequences. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pp. 6476–6480, 2025.

Dmitry Osin, Igor Udovichenko, Egor Shvetsov, Viktor Moskvoretskii, and Evgeny Burnaev. Ebes: Easy benchmarking for event sequences. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pp. 5730–5741, 2025.

Inkit Padhi, Yair Schiff, Igor Melnyk, Mattia Rigotti, Youssef Mroueh, Pierre Dognin, Jerret Ross, Ravi Nair, and Erik Altman. Tabular transformers for modeling multivariate time series. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3565–3569. IEEE, 2021.

Pengjie Ren, Zhumin Chen, Jing Li, Zhaochun Ren, Jun Ma, and Maarten De Rijke. Repeatnet: A repeat aware neural recommendation machine for session-based recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 4806–4813, 2019.

Abigail See, Peter J Liu, and Christopher D Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1073–1083, 2017.

Oleksandr Shchur, Marin Biloš, and Stephan Günnemann. Intensity-free learning of temporal point processes. *International Conference on Learning Representations (ICLR)*, 2020.

Oleksandr Shchur, Ali Caner Türkmen, Tim Januschowski, and Stephan Günnemann. Neural temporal point processes: A review. In Zhi-Hua Zhou (ed.), *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pp. 4585–4593. International Joint Conferences on Artificial Intelligence Organization, 8 2021. doi: 10.24963/ijcai.2021/623. URL https://doi.org/10.24963/ijcai.2021/623. Survey Track.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. *Advances in neural information processing systems*, 28, 2015.

Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272, 2020.

In-Kwon Yeo and Richard A Johnson. A new family of power transformations to improve normality or symmetry. *Biometrika*, 87(4):954–959, 2000.

Mai Zeng, Florence Regol, and Mark Coates. Interacting diffusion processes for event sequence forecasting. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24, 2024.

Dongyu Zhang, Liang Wang, Xin Dai, Shubham Jain, Junpeng Wang, Yujie Fan, Chin-Chia Michael Yeh, Yan Zheng, Zhongfang Zhuang, and Wei Zhang. Fata-trans: Field and time-aware transformer for sequential tabular data. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pp. 3247–3256, 2023.

Qiang Zhang, Aldo Lipani, Omer Kirnap, and Emine Yilmaz. Self-attentive hawkes process. In *International conference on machine learning*, pp. 11183–11193. PMLR, 2020.

Simiao Zuo, Haoming Jiang, Zichong Li, Tuo Zhao, and Hongyuan Zha. Transformer hawkes process. In *International conference on machine learning*, pp. 11692–11702. PMLR, 2020.

A Proofs of the Torus Encoder Properties

A.1 Torus Encoder Properties

Proof of Theorem 1. We prove the four claims in order and then verify the stated sufficient irrationality condition for the prime-square-root construction.

Fixed dimension. For each index n, the encoder emits exactly one cosine–sine pair for each of the m rotation coordinates. Hence $\mathbf{e}_n$ has exactly $2m$ real coordinates and belongs to $\mathbb{R}^{2m}$. Neither the definition of m nor the number of coordinates in $\mathbf{e}_n$ depends on the largest assigned index or on the number of keys currently stored in the registry. Appending further identities therefore enlarges the registry state but not the representation dimension.

Unit norm. Using the Pythagorean identity separately for each cosine–sine pair,

$$\begin{aligned}\|\mathbf{e}_n\|_2^2 &= \frac{1}{m}\sum_{j=1}^{m}\left(\cos^2\theta_{n,j} + \sin^2\theta_{n,j}\right)\\ &= \frac{1}{m}\sum_{j=1}^{m} 1 = 1.\end{aligned}$$

Since norms are nonnegative, $\|\mathbf{e}_n\|_2 = 1$ for every n. Thus the registry index changes only the phases of the m circular coordinates; it does not change the Euclidean scale of the representation.

Registry-extension stability. Let an existing ordered registry be $\mathcal{R} = (c_0, \ldots, c_{N-1})$, where first-seen assignment gives $n_{\mathcal{R}}(c_i) = i$. Let

$$\mathcal{R}' = (c_0, \ldots, c_{N-1}, d_0, \ldots, d_{M-1}) \tag{8}$$

be an extension obtained only by appending identities $d_0, \ldots, d_{M-1}$ that were not previously registered. By construction, no pre-existing entry is moved or re-indexed, so

$$n_{\mathcal{R}'}(c_i) = i = n_{\mathcal{R}}(c_i) \qquad \text{for every } i \in \{0, \ldots, N-1\}. \tag{9}$$

The vector assigned to a class is a deterministic function only of its integer index and the fixed frequencies $\boldsymbol{\omega}$. Therefore

$$\mathbf{e}_{n_{\mathcal{R}'}(c_i)} = \mathbf{e}_i = \mathbf{e}_{n_{\mathcal{R}}(c_i)} \qquad \text{for every previously registered identity } c_i. \tag{10}$$

Hence appending new keys does not alter any existing code. This property concerns extension of a fixed ordered registry; changing the encounter order of already registered semantic identities would in general change which index, and therefore which code, is assigned to them.

Injectivity from one irrational coordinate. Assume that ω_{j_*} is irrational for some $j_* \in \{1, \ldots, m\}$. Suppose, toward a contradiction, that two distinct indices $n, r \in \mathbb{N}$ satisfy $\mathbf{e}_n = \mathbf{e}_r$. Equality of the full vectors implies equality of the cosine–sine pair in coordinate j_*:

$$\left(\cos\theta_{n,j_*}, \sin\theta_{n,j_*}\right) = \left(\cos\theta_{r,j_*}, \sin\theta_{r,j_*}\right). \tag{11}$$

The map $t \mapsto (\cos 2\pi t, \sin 2\pi t)$ identifies two real numbers exactly when they differ by an integer. Consequently,

$$n\omega_{j_*} - r\omega_{j_*} = (n-r)\omega_{j_*} \in \mathbb{Z}. \tag{12}$$

Because $n \neq r$, the integer $d = n - r$ is nonzero. If $d\omega_{j_*} = k$ for some $k \in \mathbb{Z}$, then $\omega_{j_*} = k/d \in \mathbb{Q}$, contradicting the assumed irrationality of ω_{j_*}. Thus no two distinct integer indices can have the same code, and $n \mapsto \mathbf{e}_n$ is injective.

Table 2: Evaluation datasets. Elapsed time is itself a predicted continuous axis. Source-specific construction is given below.

Dataset	Entity history	Other predicted axes
`ibm-cc` (Altman, 2022)	user–card pair	amount; merchant category code; merchant city
`ibm-aml` (Altman, 2022)	sending account	destination bank; received amount; receiving currency; payment format
`mbd-mini` (Mollaev et al., 2025)	client	amount; event type; event subtype; source type; destination type
`the-complete-journey` (Boehmke & Mortimer, 2025)	household	store identity; basket sales; number of items
`synsepa` (Bajaj, 2026)	account	beneficiary IBAN; remittance category; signed amount

The same argument is equivalent to absence of a nonzero finite period. Indeed, if some integer $d \neq 0$ were a period, then $\mathbf{e}_{n+d} = \mathbf{e}_n$ wherever both indices are in the domain; applying the preceding coordinate argument would give $d\omega_{j_*} \in \mathbb{Z}$ and again force ω_{j_*} to be rational. Conversely, any equality $\mathbf{e}_n = \mathbf{e}_r$ with $n \neq r$ produces the nonzero difference $d = n - r$ satisfying the same phase-period condition. Hence injectivity over integer indices and absence of a nonzero integer period are equivalent here.

Prime-square-root frequencies. For the Torus Encoder used in Axial Pointer, $\omega_j = \sqrt{p_j} \bmod 1$, where p_j is prime. Since a prime is not a perfect square, $\sqrt{p_j}$ is irrational. Subtracting the integer $\lfloor \sqrt{p_j} \rfloor$ cannot turn an irrational number into a rational one, so its fractional part

$$\omega_j = \sqrt{p_j} - \lfloor \sqrt{p_j} \rfloor \tag{13}$$

is irrational as well. Therefore every coordinate satisfies the sufficient condition above; in particular, one coordinate alone would already suffice for injectivity. The stronger fact that the chosen square-root frequencies enjoy additional rational-independence properties is not needed for this proposition.

Finally, injectivity is only a statement that distinct integer indices map to distinct points in exact arithmetic. Because all codes lie on the unit sphere inside the finite-dimensional compact space $\mathbb{R}^{2m}$, the proposition asserts no uniform positive lower bound on pairwise distances as the registry grows. The implementation stores each frequency as an exact ratio of its finite-precision machine value and evaluates its phase by integer modular arithmetic before applying trigonometric functions. This mitigates large-angle phase loss, but the resulting finite-state orbit is eventually periodic and cannot provide unlimited numerical identity separation. The exact-arithmetic argument therefore does not establish machine-precision collision-freedom or aperiodicity. □

B Implementation Details

B.1 Datasets and Preprocessing

Evaluation dataset suite. The five datasets share the mixed-type next-event interface while retaining source-specific event schemas. Table 2 lists the entity history and non-time axes predicted for each dataset; elapsed time is also a predicted continuous axis.

Source-specific event definitions. Table 3 gives the source event and chronological identity used by each adapter. Sorting is stable and simultaneous logical events remain distinct, with elapsed time zero; the first row of every materialized episode also has elapsed time zero. The Complete Journey is the only dataset that aggregates source product rows, combining them within a household basket before episode construction.

For `ibm-cc`, all 24,386,900 rows are retained, signed amounts are preserved, and missing merchant codes or cities map to an explicit missing label. For `mbd-mini`, all 7,634,228 rows and signed amounts are retained; configured missing labels represent absent categorical codes.

Table 3: Source-event and ordering definitions. "Source order" denotes physical row order, or source-part name followed by row position for `mbd-mini`.

Dataset	Source event	Entity key	Within-entity chronology and ties
`ibm-cc` (Altman, 2022)	transaction CSV row	`User`, `Card`	parsed calendar date/time; source order breaks equal timestamps
`ibm-aml` (Altman, 2022)	transfer CSV row	sending `Account`	parsed timestamp; source order breaks ties
`mbd-mini` (Mollaev et al., 2025)	transaction Parquet row	`client_id`	`event_time`; sorted part name and row position break ties
The Complete Journey (Boehmke & Mortimer, 2025)	household–basket aggregate	`household_key`	parsed day/time; numeric basket ID, then lexical ID, breaks ties
SynSEPA (Bajaj, 2026)	transaction CSV row	`account_id`	naive source-local `timestamp`; source order breaks ties

Table 4: Exact emitted schemas and source-row policies. Elapsed-time units are explicit in the axis names.

Dataset	Emitted axes	Filtering and missingness
`ibm-cc`	`delta_time_hours`, `amount`, `mcc`, `merchant_city`	retain all rows and signed amounts; explicit missing category
`ibm-aml`	`delta_time_hours`, `to_bank`, `amount_received`, `receiving_currency`, `payment_format`	whole-source top-6,500 bank filter; remove affected account histories (disclosed look-ahead)
`mbd-mini`	`delta_time_days`, `amount`, `event_type`, `event_subtype`, `src_type11`, `dst_type11`	retain all rows and signed amounts; configured explicit missing labels
The Complete Journey	`delta_time_days`, `store_id`, `total_sales`, `number_of_items`	aggregate product rows by household and basket; reject inconsistent baskets
SynSEPA	`delta_time_days`, `beneficiary_iban`, `remittance_category`, `amount`	retain all rows and signed amounts; fail on invalid required values

For `ibm-aml`, the receiving account and laundering indicator are not loaded as model fields. Destination banks are ranked by whole-source frequency (label order breaks cutoff ties), the top 6,500 are retained, and an entire sending-account history is removed if it contains any other bank. This emits 29,803,316 of 31,251,483 rows. Because eligibility uses the full observation period, it is disclosed as distribution look-ahead; all later transforms and model-facing vocabularies remain training-fitted. Other adapters either map configured categorical missingness explicitly or fail on malformed, nonfinite, or missing required values.

For `the-complete-journey`, all 2,595,732 valid product rows are retained and aggregated into 276,484 household–basket events. Store and timestamp fields must be consistent within a basket; sales and item quantities are summed while preserving finite zero or negative totals. For `synsepa`, all 1,839,560 source transaction rows are retained as distinct events without filtering, deduplication, or aggregation. The adapter requires nonempty lexical account and category labels, finite

signed amounts, and valid timezone-naive timestamps in the source-local format. It preserves label spelling, case, whitespace, and leading zeros, and it neither validates IBAN checksums nor infers a currency. Rows are stably ordered by account and timestamp, with physical source order retaining simultaneous transactions as distinct zero-gap events.

Episode planning and split membership. The shared planner assigns an entity's final 50-event window to test and, when at least 100 events exist, the immediately preceding 50 to validation. Training requires at least 150 events and uses complete windows starting at zero with strides 250, 100, and 150 for `ibm-cc`, `ibm-aml`, and `mbd-mini`, respectively; all end before the 100 held-out events and no extra right-aligned window is added. These splits therefore concern future events of potentially known entities.

The Complete Journey and SynSEPA override this rule. Eligible households or accounts with at least 50 events are sorted lexicographically, and their final windows are assigned alternately to test and validation beginning with test. Fifty-event training windows use the prefix preceding that holdout, with strides 20 and 25 for The Complete Journey and SynSEPA, respectively. No entity belongs to both holdouts, although its earlier prefix may occur in training; entities with 50–99 events can contribute only a holdout.

For all datasets, validation and test candidates are independently sampled, with preprocessing seed 42, down to $\lfloor 0.25 N_{\text{train}} \rfloor$ when necessary. Each split is then independently shuffled with a split-specific deterministic stream and assigned round-robin to four shards, so shard counts differ by at most one. Training targets positions 21–50; validation and test target positions 41–50. Context-only rows remain available to causal attention but never enter an objective or metric.

Training-fitted transforms and categorical registries. Canonical Parquet artifacts retain continuous values in raw units. Each continuous axis is mapped to model space by a standardized Yeo–Johnson transform (Yeo & Johnson, 2000) fitted jointly to emitted training rows across all shards. The additional C-Diff elapsed-time Box–Cox state (Box & Cox, 1964) is fitted only to non-initial training intervals. Validation and test values do not affect either fit. Predictions are inverse-transformed before metrics are accumulated.

Each discrete axis has a separate vocabulary consisting of labels observed in training episodes. A reserved ID immediately after the known labels encodes validation/test-only labels; explicit source-missing labels are ordinary known categories when observed in training. This ID is an encoding device rather than an additional raw category: the raw validation/test label remains in the artifact, while an out-of-training label is represented internally by the reserved integer. AP's fixed Torus registry namespaces these IDs by axis and is prepopulated in metadata order, preventing episode or split order from changing an identity's coordinates. A history-novel pointer outcome is different again: it means that the target class is absent from the current prefix and does not name the raw label. Exact categorical accuracy compares decoded predictions with the raw target label, so detecting novelty or predicting the reserved ID alone receives no credit for a held-out label.

Artifact-level dataset summary. Table 5 reports the persisted setup state. Discrete cardinalities count known training labels and exclude the one reserved unknown ID per axis. All artifacts contain four nonempty shards.

Before clipping, the validation/test candidate counts were 5,072/5,488, 56,248/182,723, 12,024/14,413, 833/834, and 5,000/5,000 in table order. The resulting total episode counts are 107,405, 58,050, 64,315, 7,478, and 48,541. SynSEPA has 10,000 eligible accounts; all candidate holdouts are retained, and its four balanced shards contain 9,635 or 9,636 training episodes plus 1,250 episodes from each holdout. These are fixed data-construction quantities, not model results.

B.2 Experimental Scope and Metric Rationale

Short-context scope. Cold start is a central practical challenge in financial transaction forecasting: useful predictions must often be made before a long entity history has accumulated. We therefore fix episodes at 50 events and score only their later targets, deliberately studying learning and prediction from a relatively short observed history rather than asymptotic performance with abundant per-entity context. This choice does not give AP an unusually favorable setting. In particular, AP's pointer

Table 5: Persisted episode counts and training-vocabulary cardinalities.

Dataset	Train	Validation	Test	Discrete-axis cardinalities
ibm-cc	96,845	5,072	5,488	mcc: 109; merchant_city: 11,270
ibm-aml	38,700	9,675	9,675	to_bank: 6,208; receiving_currency: 15; payment_format: 7
mbd-mini	42,877	10,719	10,719	event_type: 52; event_subtype: 59; src_type11: 43; dst_type11: 45
the-complete-journey	5,811	833	834	store_id: 359
synsepa	38,541	5,000	5,000	beneficiary_iban: 311,543; remittance_category: 9

expert conditions its categorical prediction on positions in the available prefix, so a shorter prefix supplies fewer observations from which to estimate recurrence and fewer historical candidates on which to place pointer mass. Longer histories could expose more repeated identities and richer conditioning evidence. The reported setting should therefore be read as a cold-start-oriented scope choice and, if anything, a conservative one for the context-dependent pointer mechanism, not as a claim about performance at arbitrary sequence lengths.

Conditional next-event metrics and likelihood. The reported metrics directly evaluate the point predictions required by the task and apply uniformly across the heterogeneous model families, while separate per-axis reporting exposes time, other continuous attributes, and categorical marks rather than conflating them in a joint score (Bosser & Taieb, 2023). Likelihood instead evaluates a full predictive distribution; using it as a common metric would additionally require AP to specify normalized continuous predictive densities beyond its point-prediction mechanism. This is a difference in evaluation target, not a claim that likelihood is statistically invalid (Shchur et al., 2020).

Long-horizon rollout metrics. Long-horizon sequence metrics are not uniquely neutral replacements for these next-event metrics. For example, Optimal Transport Distance requires user-selected insertion and deletion costs, and changing those costs can materially alter method rankings (Zeng et al., 2024). We therefore do not treat one particular rollout-and-distance specification as definitive for the conditional next-event task. Autoregressive rollout remains a complementary evaluation for the distinct question of predicting multiple future events.

Raw-unit MAE for continuous next events. Elapsed times and transaction amounts are highly stochastic even conditional on the available prefix, and occasional very large values may be weakly predictable from a short history. MAE assigns error in direct proportion to the magnitude of each miss. It therefore preserves the real cost of a large error without allowing the quadratic penalty used by MSE or RMSE to make a small number of extreme, potentially irreducible deviations dominate the aggregate comparison. This is important here because the intended question is how accurately a method predicts continuous next-event values across the observed event distribution, rather than how aggressively it avoids the worst squared residuals. MAE is not insensitive to extremes, but its linear influence makes the reported score more representative of performance across the full test population.

We compute MAE after inverse transformation, in each axis's declared physical unit. This keeps an hour or unit of transaction amount directly interpretable and avoids evaluating methods in a preprocessing-dependent model space. Percentage errors would introduce a different and unstable convention for zero or near-zero targets and are ill-suited to axes that may be signed, while Huber-type metrics require selecting a scale-dependent transition threshold. Reporting every continuous axis separately avoids comparing or averaging raw errors across incompatible units. Thus

MAE supplies a simple common decision criterion without adding an outlier threshold, percentage denominator, or quadratic preference that is not part of the stated forecasting objective.

Exact accuracy for discrete next events. Each discrete axis is a single-label next-event problem: at every scored position exactly one categorical value is observed, and the prediction is correct precisely when that value has the highest predicted probability. Exact accuracy thus estimates the empirical probability of correctly identifying the next categorical event under the observed test distribution. This differs from rare-event detection, anomaly detection, or asymmetric binary classification, where precision and recall for a designated minority class may themselves be the target. In single-label multiclass evaluation, micro-averaged F1 is identical to accuracy. Macro-F1 instead gives every class equal weight irrespective of its empirical frequency and would therefore change the forecasting objective to class-balanced performance. Put differently, accuracy takes a utilitarian rather than class-egalitarian view: with one unit of value for each correctly forecast event, it rewards the total number of correct decisions under the deployment distribution. Macro-F1 instead prioritizes a form of fairness across classes, even when improving a very rare class would yield fewer correct event forecasts overall.

Rare occurrence alone is therefore not a basis for allowing a class to determine the aggregate metric. We do not assume that a rare transaction category is intrinsically as important as a category occurring orders of magnitude more often; such weighting is appropriate only when supplied by the application. Accuracy is therefore the direct, assumption-light metric for the forecasting objective studied here. Its familiar weakness under class imbalance is addressed through comparison with the separately reported causal prefix mean/mode reference (Appendix C.3). This training-free predictor is a statistically comparable baseline because it uses the same causal information and evaluation distribution while reducing its forecast to the empirical prefix mean or mode. It reveals when strong raw performance is largely attainable from simple statistics of the available history and thus supplies a meaningful baseline for gains beyond frequency and historical level. This preserves the empirical occurrence distribution as the evaluation target without silently introducing an additional class-weighting convention.

B.3 Mixed-Type Baseline Adaptations

Shared adaptation contract. Every adapter consumes the same fixed episodes. A target at row t can use only rows $[1, t-1]$; schema-local discrete IDs, metadata-transformed continuous inputs, valid masks, and split target masks are supplied by the common loader. Teacher-forced predictions cover the full schema, with continuous values decoded to raw units and categorical IDs decoded to raw labels before scoring. Adapters neither refit transforms nor change episode membership or evaluation vocabularies.

Marked temporal point-process baselines. The adaptations retain each model's principal temporal mechanism for RMTPP (Du et al., 2016), NHP (Mei & Eisner, 2017), SAHP (Zhang et al., 2020), THP (Zuo et al., 2020), AttNHP (Mei et al., 2021), S2P2 (Chang et al., 2026), and IFTPP (Shchur et al., 2020) while factorizing a single-mark interface over the mixed schema:

Each discrete axis receives an independent metadata-sized embedding and categorical classifier head rather than a Cartesian-product mark. A joint projection of all non-time transformed continuous axes augments the event input, and each such axis receives an unconstrained transformed-space scalar prediction head. Except for the intensity-free IFTPP law, the point-process models use one event-time law and one survival term per target, not one per categorical axis. Objectives combine the native time likelihood with the mean categorical cross-entropy over discrete axes and mean transformed error over auxiliary continuous axes; family weights preserve comparable scale as schema width changes. Training retains native Monte Carlo compensators where applicable, whereas held-out loss and time decoding are deterministic. All point decoders return elapsed time in the artifact's declared physical unit.

Sequential generative baseline. C-Diff (Zeng et al., 2024) retains a causal Transformer history encoder, multinomial and Gaussian forward processes, variational categorical and epsilon objectives, and synchronous reverse updates. The encoder and target decoder both cover the complete mixed event. Every discrete axis owns an axis-local multinomial diffusion; every continuous axis

Table 6: Preserved temporal mechanisms and deterministic held-out point decoders in the adapted temporal baselines.

Family	Preserved mechanism	Held-out time point
RMTPP	recurrent history and exponential conditional intensity	numerical conditional mean
NHP	continuous-time LSTM with between-event decay	finite adaptive density quadrature
SAHP	causal self-attention and decaying hidden state	finite density integration
THP	causal Transformer and softplus intensity	deterministic density integration
AttNHP	candidate-time attention and softplus intensity	finite adaptive density quadrature
S2P2	complex DPLR/HiPPO state-space dynamics and softplus intensity	finite adaptive density quadrature
IFTPP	GRU-conditioned log-normal-mixture time law, without an intensity	exact mixture expectation

owns a Gaussian diffusion, using the training-fitted C-Diff Box–Cox space for elapsed time and standardized Yeo–Johnson space otherwise. One sampled diffusion step jointly noised all fields during training, with losses averaged within axis families. Prediction jointly denoises one event from the causal prefix.

Causal-prefix mean/mode reference. For target row t, the training-free reference predicts each continuous axis by its raw-unit arithmetic mean over rows $[1, t-1]$ and each discrete axis by the prefix mode. Mode ties choose the smallest axis-local metadata ID. This method has no fitted state, optimizer, scheduler, or checkpoint. It is reported separately and supplies HPO validation-error denominators only; its normalized errors are used only for HPO and are not reported as evaluation metrics.

B.4 Optimization, HPO, and Checkpoint Selection

Computational resources. The training group was submitted as a Slurm array. Each array task was allocated one node with four A40 GPUs, 32 CPUs per task, and 172 GB of RAM. The Slurm wall-time limit was 24 hours. Multi-GPU training used one distributed process per GPU with bfloat16 mixed precision via the `accelerate` library (Gugger et al., 2022). The models and optimization are implemented in PyTorch (Ansel et al., 2024); NumPy (Harris et al., 2020) and pandas (McKinney et al., 2011) support preprocessing and result aggregation, SciPy (Virtanen et al., 2020) supplies statistical transforms, and Matplotlib (Hunter, 2007) generates figures.

Common training lifecycle. Each of the four distributed processes loads one metadata shard. The per-rank batch size is 64 (global batch 256), and local training loaders are resampled to a common length and reshuffled as they cycle. All methods run 50 stages of 50 optimizer updates, validating after each stage. The update count is therefore invariant to dataset size. Schedulers step after every optimizer update and the shared configuration uses zero warmup stages.

AP uses AdamW (Loshchilov & Hutter, 2019), cosine decay (Loshchilov & Hutter, 2017), and global gradient-norm clipping at 5. SAHP and RMTPP also use AdamW with cosine schedules; C-Diff, S2P2, NHP, AttNHP, and IFTPP use Adam (Kingma & Ba, 2015) with cosine schedules; THP uses Adam with its configured step decay. Baseline gradient norms are clipped at 1, except RMTPP at 0.5. Learning rates, widths, depths, and other method-native settings come from the selected dataset-specific configuration. Final group runs use seeds 1–5. The best validation checkpoint is loaded for one teacher-forced test evaluation; test data never select a model or route.

All reported SynSEPA learned runs use the same episode artifact, seeds, evaluation axes, sampled-prediction settings, and transaction configuration.

Table 7: Discrete HPO search spaces. The prespecified default is included in each listed set.

Method	Searched values
AP	width $\{256, 384, 512\}$; layers $\{3, 4, 5\}$; learning rate $\{3, 5, 8\} \times 10^{-4}$
SAHP	width $\{256, 384, 512\}$; layers $\{4, 6, 8\}$; learning rate $\{0.5, 1, 3\} \times 10^{-4}$
RMTPP	hidden width $\{256, 384, 512\}$; layers $\{2, 3, 4\}$; learning rate $\{0.5, 1, 3\} \times 10^{-4}$
THP	width/inner/key/value grouped as $\{(256, 512, 32, 32), (320, 640, 40, 40), (384, 768, 48, 48)\}$; layers $\{4, 6, 8\}$; learning rate $\{0.5, 1, 3\} \times 10^{-4}$
C-Diff	encoder/feed-forward width $\{(128, 512), (160, 640), (256, 784)\}$; encoder/decoder depth $\{(1, 1), (2, 2), (3, 3)\}$; learning rate $\{3 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}\}$
S2P2	hidden width $\{128, 256, 384, 512\}$; layers $\{2, 4, 6\}$; learning rate $\{3 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}\}$
NHP	hidden width $\{128, 256, 384, 512\}$; learning rate $\{10^{-4}, 3 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}, 5 \times 10^{-3}\}$
AttNHP	hidden width $\{64, 96, 128, 200, 256\}$; layers $\{2, 3, 4\}$; learning rate $\{3 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}\}$
IFTPP	hidden width $\{64, 128, 256, 512\}$; mixture components $\{2, 3, 5\}$; learning rate $\{3 \times 10^{-4}, 10^{-3}, 3 \times 10^{-3}\}$

AP categorical objective and expert aggregation. The final AP training runs weight these categorical pointer terms. Let $N_{k,a}$ be the number of occurrences of class a in the complete episode (not only the current prefix), let β be the class-balance exponent, and let λ_{new} be the relative history-novel weight. Before a final cap-aware normalization, a scoreable target receives

$$r_{t,k} = \begin{cases} N_{k,c_{t+1,k}}^{-\beta}, & c_{t+1,k} \in \mathcal{C}_{t,k}, \\ \lambda_{\text{new}}, & c_{t+1,k} \notin \mathcal{C}_{t,k}. \end{cases} \tag{14}$$

For observed targets, the inverse-frequency values are first divided by their mean over scoreable observed targets for that episode and axis; the novel value is relative to that observed-target mean. The resulting weights are then rescaled over all scoreable targets for that episode and axis to have mean one, with a hard maximum of five. The reported configuration uses $\beta = 0.5$ and $\lambda_{\text{new}} = 0.25$, so observed targets use tempered inverse-square-root full-episode frequency and history-novel targets receive a quarter of the pre-normalization observed-target mean. Thus the pointer categorical loss is $\ell_{t,k}^{\text{ptr,disc}} = w_{t,k}(-\log \pi_{t,k,y})$, where $w_{t,k}$ is the capped, mean-one version of $r_{t,k}$. These settings apply to both the final production AP configuration and the pointer-only AP ablation. Direct categorical classifier heads use plain fixed-vocabulary cross-entropy; they do not inherit the pointer weights. At every scoreable cutoff, AP sums the pointer term for every axis and adds the direct term for each eligible dual-expert axis, then takes one global mean over cutoff–axis pairs. Consequently, dual-expert axes contribute both native terms and pointer-only axes one; the objective does not renormalize by the number of active experts. The configured categorical weights are used for optimization, whereas validation, checkpoint selection, and routing use the unweighted raw metrics described below.

Dataset-specific hyperparameter selection. Each dataset–method study runs on shard 0 with one process and seed 42. A dedicated seeded permutation selects up to 512 local validation episodes. The constant reference is recomputed on exactly this subset. Each axis contributes the candidate error—MAE or $1 - \text{accuracy}$—divided by the corresponding constant error, and these ratios are averaged equally. The minimum objective over a trial's validation stages is its score.

Every executed study attempted 16 evaluations in total: forced default trial 0 followed by 15 Hyperopt candidates from TPE (Bergstra et al., 2015), with two startup jobs. Each trial has eight 200-update stages (at most 1,600 updates) and no warmup. After at least three stages, the trial stops after three consecutive stages without nonnegative relative improvement. Duplicate samples reuse their cached outcome; out-of-memory, non-finite, or otherwise invalid trials are marked failed and do not supply a winning configuration. The lowest successful score determines the single configuration used for all final seeds.

For the AP single-expert ablations, only AP's searched paths (width, depth, and learning rate) are transferred from the dataset-specific winning AP configuration; each ablation variant retains its own architecture, objective, and nonsearched defaults.

Validation checkpoint rule. Let b_k be the current selected checkpoint's metric and c_k the candidate's metric. A continuous axis contributes $(b_k - c_k)/\max(|b_k|, 10^{-8})$ and a discrete axis contributes $(c_k - b_k)/\max(|b_k|, 10^{-8})$. Every axis has equal weight. The first completed stage initial-

izes selection, and a later candidate replaces it when the sum is nonnegative, so exact aggregate ties are accepted. Native held-out losses are diagnostic and never enter this comparison.

The resumable "last" checkpoint restores model, optimizer, scheduler, stage, global update, router, and current best-selection state; the completed update must align with the configured stage length.

B.5 Adaptive Routing Protocol

Reported routing configuration. The reported AP configuration enables dual experts for every continuous axis and for discrete axes with at most 256 known training labels; larger axes remain pointer-only. Each private expert uses a width-128 residual adapter with a zero-initialized output projection.

Validation evidence and soft routing. After each 50-update stage, distributed full-validation metrics are computed separately for both experts. The router uses MAE for continuous axes and $1 - \text{accuracy}$ for discrete axes, with an EMA decay of 0.8. For uncommitted axes with finite evidence $E_{k,e}$,

$$w_{k,e} = \mathrm{softmax}_e \left(\frac{-\log(E_{k,e} + 10^{-8})}{\tau} \right), \tag{15}$$

where τ follows a cosine schedule (Loshchilov & Hutter, 2017) over a 30-assessment post-warmup horizon. The first five assessments keep weights at $(0.5, 0.5)$, validation becomes authoritative after three assessments, and the final route lock occurs at assessment 30.

Commitment, reopening, and final lock. Commitment can begin at assessment 15. An expert must hold weight strictly above 0.9 for three consecutive assessments. A committed route can reopen for ten assessments after commitment if the challenger EMA is more than 10% lower for two consecutive assessments. At assessment 30, every unresolved axis chooses the lower-error EMA, all committed routes are frozen, and router state is persisted in checkpoints. The prior best checkpoint is discarded and validation selection restarts under the fixed deployment policy.

B.6 Scalability Accounting

Trainable categorical parameterization. Let D be backbone width, $2m$ Torus width, P pointer hidden width, A the expert-adapter width, V_k^{known} the number of known training labels, $V_k = V_k^{\text{known}} + 1$ the model vocabulary size including the reserved unknown, K the number of event axes, and L the observed prefix length. The shared Torus projection has $D(2m + 1)$ parameters. For one pointer-routed categorical axis, the additive scorer, novelty head, and frequency correction contribute

$$P_{\text{cat-ptr},k} = (2DP + 2P) + (D + 1) + 1 = 2DP + 2P + D + 2. \tag{16}$$

Neither expression depends on V_k or L: the pointer categorical path has no learned per-class representation or output term. A dual-expert axis additionally allocates two residual adapters with $4AD+2A+6D$ parameters. For a dual-expert categorical axis, its direct expert adds the vocabulary-dependent classifier term $(D + 1)V_k^{\text{known}}$.

For a fixed schema and set of dual-expert axes, the complete AP count can therefore be written

$$P_{\text{AP}} = P_{\text{fixed}} + \sum_{k \in \mathcal{K}_{\text{disc,dual}}} (D + 1)V_k^{\text{known}}, \tag{17}$$

where P_{fixed} includes the backbone, Torus projection, pointer scorers, novelty heads, adapters, and continuous scalar prediction heads. The absence of a V_k term for a pointer-only categorical axis does not imply that this total is smaller than another architecture's instantiated total. For comparison, a learned input embedding of width E_k contributes $V_k E_k$, and a global classifier from width H_k contributes $(H_k + 1)V_k$.

History- and vocabulary-dependent computation. For all-cutoff training, pointer projections and compatibility scoring require $O(KLDP + KL^2P)$ arithmetic and materialize $O(KL^2)$ scores; a last-query prediction scores L candidates per axis and requires $O(KLDP + KLP)$ arithmetic. A direct categorical classifier head instead forms V_k^{known} logits with $O(DV_k^{\text{known}})$ arithmetic per

query. Independently, one axial backbone block has attention-score cost $O(LK^2 + KL^2)$, omitting batch and feature factors. Thus pointer-path parameterization can be vocabulary-independent while training and inference remain history-length-dependent; it establishes no general runtime advantage.

Persistent registry state. All experiments use $m = 4$, so each Torus code has width $2m = 8$. The fixed Torus registry stores m and an ordered mapping from axis-namespaced category keys to integer indices; it is prepopulated from each artifact's model vocabulary and serialized in AP's dataset cache. It contains no learned vector per class, but its key/index state grows with registered cardinality and is conceptually separate from learnable parameters. Metadata vocabularies are also persistent nonparameter state. Consequently, neither learned parameterization of the pointer path nor fixed Torus width makes total system state independent of registered cardinality.

C Additional Results

C.1 Complete per-axis results

Tables 8–12 report all 23 prediction axes for the learned methods. Continuous axes use raw-unit MAE and discrete axes use exact-label accuracy. Values are mean ± standard deviation over 5 completed seeds; best and second-best emphasis is based on unrounded seed-aggregated means. The AP annotations give the modal deployed pointer [P] or direct [D] expert over seeds. These tables are the auditable per-axis evidence underlying the compact summary.

Table 8: Raw-unit results for IBM-CC. Values are mean ± standard deviation over 5 completed seeds. Best and second-best unrounded means are bold and underlined, respectively. Standard deviations whose formatted numeric strings exceed 10 characters are shown as high. AP cells append [P] or [D] using the modal deployed expert over seeds.

Method	delta_time_hours (MAE ↓)	amount (MAE ↓)	mcc (Acc. ↑)	merchant_city (Acc. ↑)
AP	**23.2 ± 0.07** [D]	**37.3 ± 0.36** [P]	**0.302 ± 0.00** [D]	**0.642 ± 0.00** [P]
SAHP	43.9 ± 24.09	42.4 ± 1.26	0.232 ± 0.00	0.529 ± 0.01
RMTPP	25.2 ± 0.04	42.2 ± 0.16	0.272 ± 0.00	0.572 ± 0.00
THP	24.8 ± 0.15	38.2 ± 0.19	0.290 ± 0.00	0.601 ± 0.00
S2P2	25.5 ± 0.50	40.6 ± 0.74	0.243 ± 0.01	0.576 ± 0.01
NHP	$\underline{23.8 \pm 0.05}$	38.2 ± 0.21	0.296 ± 0.00	$\underline{0.617 \pm 0.00}$
AttNHP	24.0 ± 0.07	39.0 ± 0.17	0.283 ± 0.00	0.608 ± 0.00
IFTPP	$1.66 \times 10^{34} \pm high$	$\underline{38.0 \pm 0.13}$	$\underline{0.298 \pm 0.00}$	0.608 ± 0.01
C-Diff	54.0 ± 1.41	125 ± 3.48	0.120 ± 0.00	0.476 ± 0.00

Table 9: Raw-unit results for IBM-AML. Values are mean ± standard deviation over 5 completed seeds. Best and second-best unrounded means are bold and underlined, respectively. Standard deviations whose formatted numeric strings exceed 10 characters are shown as high. AP cells append [P] or [D] using the modal deployed expert over seeds.

Method	delta_time_hours (MAE ↓)	to_bank (Acc. ↑)	amount_received (MAE ↓)	receiving_currency (Acc. ↑)	payment_format (Acc. ↑)
AP	**2.87 ± 0.04** [P]	**0.765 ± 0.02** [P]	**$2.82 \times 10^6 \pm 230{,}972.24$** [P]	**0.993 ± 0.00** [P]	**0.682 ± 0.00** [D]
SAHP	4.47 ± 0.73	0.433 ± 0.01	$4.28 \times 10^6 \pm 38{,}991.61$	0.987 ± 0.00	0.489 ± 0.01
RMTPP	5.75 ± 0.72	0.0428 ± 0.00	$4.32 \times 10^6 \pm 282.97$	0.984 ± 0.00	0.512 ± 0.00
THP	3.66 ± 0.11	0.203 ± 0.03	$4.87 \times 10^6 \pm high$	0.989 ± 0.00	$\underline{0.615 \pm 0.01}$
S2P2	3.41 ± 0.04	0.153 ± 0.07	$4.29 \times 10^6 \pm 20{,}125.50$	0.973 ± 0.01	0.510 ± 0.01
NHP	$\underline{3.26 \pm 0.02}$	$\underline{0.563 \pm 0.00}$	$\underline{4.14 \times 10^6 \pm 141{,}585.60}$	$\underline{0.989 \pm 0.00}$	0.593 ± 0.00
AttNHP	3.41 ± 0.02	0.510 ± 0.09	$4.27 \times 10^6 \pm 19{,}577.47$	0.986 ± 0.00	0.528 ± 0.01
IFTPP	7.54 ± 4.36	0.166 ± 0.05	$4.29 \times 10^6 \pm 5{,}603.56$	0.987 ± 0.00	0.521 ± 0.00
C-Diff	$1.77 \times 10^3 \pm 331.81$	$4.57 \times 10^{-4} \pm 0.00$	$2.65 \times 10^{32} \pm high$	0.266 ± 0.13	0.265 ± 0.03

Table 10: Raw-unit results for MBD-Mini. Values are mean ± standard deviation over 5 completed seeds. Best and second-best unrounded means are bold and underlined, respectively. Standard deviations whose formatted numeric strings exceed 10 characters are shown as high. AP cells append [P] or [D] using the modal deployed expert over seeds.

Method	delta_time_days (MAE ↓)	amount (MAE ↓)	event_type (Acc. ↑)	event_subtype (Acc. ↑)	src_type11 (Acc. ↑)	dst_type11 (Acc. ↑)
AP	**3.42 ± 0.04** [D]	**$6.66\times10^{5} \pm 35.52$** [P]	**0.567 ± 0.00** [D]	**0.566 ± 0.00** [D]	**0.922 ± 0.00** [D]	**0.562 ± 0.00** [D]
SAHP	8.63 ± 2.73	$2.67\times10^{6} \pm high$	0.540 ± 0.00	0.538 ± 0.00	0.914 ± 0.00	0.530 ± 0.00
RMTPP	3.70 ± 0.03	$6.68\times10^{5} \pm 194.74$	0.559 ± 0.00	0.558 ± 0.00	0.918 ± 0.00	0.552 ± 0.00
THP	3.63 ± 0.03	$6.66\times10^{5} \pm 244.89$	0.563 ± 0.00	0.561 ± 0.00	0.919 ± 0.00	0.557 ± 0.00
S2P2	17.8 ± 13.61	$6.67\times10^{5} \pm 297.35$	0.538 ± 0.00	0.536 ± 0.00	0.914 ± 0.00	0.528 ± 0.00
NHP	3.56 ± 0.02	$6.66\times10^{5} \pm 177.40$	0.561 ± 0.00	0.559 ± 0.00	0.920 ± 0.00	0.551 ± 0.00
AttNHP	3.75 ± 0.02	$6.66\times10^{5} \pm 173.57$	0.559 ± 0.00	0.557 ± 0.00	0.918 ± 0.00	0.551 ± 0.00
IFTPP	$\underline{3.53 \pm 0.02}$	$\underline{6.66\times10^{5} \pm 47.22}$	$\underline{0.567 \pm 0.00}$	$\underline{0.564 \pm 0.00}$	$\underline{0.920 \pm 0.00}$	$\underline{0.559 \pm 0.00}$
C-Diff	5.91 ± 0.55	$4.09\times10^{7} \pm high$	0.333 ± 0.07	0.329 ± 0.07	0.777 ± 0.15	0.304 ± 0.06

Table 11: Raw-unit results for The Complete Journey. Values are mean ± standard deviation over 5 completed seeds. Best and second-best unrounded means are bold and underlined, respectively. Standard deviations whose formatted numeric strings exceed 10 characters are shown as high. AP cells append [P] or [D] using the modal deployed expert over seeds.

Method	delta_time_days (MAE ↓)	store_id (Acc. ↑)	total_sales (MAE ↓)	number_of_items (MAE ↓)
AP	**5.25 ± 0.05** [D]	0.676 ± 0.00 [P]	22.3 ± 0.21 [P]	**857 ± 4.20** [P]
SAHP	5.93 ± 0.40	0.664 ± 0.00	455 ± 665.29	886 ± 5.24
RMTPP	5.63 ± 0.01	**0.707 ± 0.00**	22.3 ± 0.09	879 ± 0.08
THP	$\underline{5.47 \pm 0.03}$	0.698 ± 0.00	22.0 ± 0.10	879 ± 0.08
S2P2	7.81 ± 5.11	0.691 ± 0.00	22.1 ± 0.16	879 ± 0.09
NHP	5.53 ± 0.13	$\underline{0.706 \pm 0.00}$	**21.8 ± 0.08**	$\underline{879 \pm 0.11}$
AttNHP	5.48 ± 0.03	0.682 ± 0.01	21.9 ± 0.05	879 ± 0.06
IFTPP	$9.05\times10^{13} \pm high$	0.704 ± 0.01	$\underline{21.9 \pm 0.15}$	879 ± 3.26
C-Diff	8.68 ± 0.41	0.539 ± 0.05	201 ± 75.20	$1.82\times10^{15} \pm high$

Table 12: Raw-unit results for SynSEPA. Values are mean ± standard deviation over 5 completed seeds. Best and second-best unrounded means are bold and underlined, respectively. Standard deviations whose formatted numeric strings exceed 10 characters are shown as high. AP cells append [P] or [D] using the modal deployed expert over seeds.

Method	delta_time_days (MAE ↓)	beneficiary_iban (Acc. ↑)	remittance_category (Acc. ↑)	amount (MAE ↓)
AP	**1.66 ± 0.00** [P]	**0.103 ± 0.00** [P]	**0.292 ± 0.00** [D]	**$1.77\times10^{3} \pm 2.99$** [P]
SAHP	1.71 ± 0.01	$6.89\times10^{-3} \pm 0.00$	0.277 ± 0.02	$1.79\times10^{3} \pm 13.40$
RMTPP	1.70 ± 0.00	$3.20\times10^{-5} \pm 0.00$	0.291 ± 0.00	$1.77\times10^{3} \pm 1.06$
THP	1.69 ± 0.01	$\underline{0.0445 \pm 0.00}$	0.287 ± 0.01	$\underline{1.77\times10^{3} \pm 2.13}$
S2P2	1.68 ± 0.01	$6.76\times10^{-4} \pm 0.00$	$\underline{0.292 \pm 0.00}$	$1.78\times10^{3} \pm 3.99$
NHP	1.67 ± 0.01	$5.44\times10^{-4} \pm 0.00$	0.291 ± 0.00	$1.78\times10^{3} \pm 1.98$
AttNHP	–	–	–	–
IFTPP	$\underline{1.67 \pm 0.01}$	0 ± 0.00	0.292 ± 0.00	$1.78\times10^{3} \pm 8.80$
C-Diff	–	–	–	–

C.2 MEAN-RANK SUMMARY

Tables 13 and 14 expand the AP-only rank columns in Table 1 to all learned methods. Methods are ranked within each usable axis–seed comparison, separately for continuous and discrete axes; lower ranks are better, and exact ties receive average ranks. The per-dataset cells average the resulting observations over that dataset's axes and seeds and report their sample standard deviation. Overall cells weight datasets equally, preventing schemas with more predicted axes from dominating the comparison.

This separation by axis type preserves the distinction made in the compact summary while exposing the full comparison behind AP's reported ranks. AP has the best overall mean rank for both continuous axes (1.39 ± 1.35, versus NHP at 3.06 ± 1.24) and discrete axes (2.36 ± 2.35, versus NHP

at 3.00 ± 1.37). The larger discrete variability reflects the wider spread across individual axis–seed comparisons, including AP's weaker rank on The Complete Journey store identity.

Table 13: Continuous mean-rank summary. Methods are ranked within every seed–axis comparison that has AP and at least one baseline, with lower ranks better and exact ties receiving average ranks. Cells report the mean rank $\pm$ sample standard deviation over the seed–axis observations for continuous axes. Overall cells use equal-dataset weighting of those observations.

Method	IBM-CC	IBM-AML	MBD-Mini	The Complete Journey	SynSEPA	Overall
AP	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{2.73 \pm 2.63}$	$\mathbf{1.20 \pm 0.42}$	$\mathbf{1.39 \pm 1.35}$
SAHP	7.70 ± 0.48	5.50 ± 1.78	7.60 ± 0.97	7.67 ± 0.82	6.50 ± 0.71	6.99 ± 1.34
RMTPP	6.90 ± 0.88	7.30 ± 0.48	6.20 ± 1.32	6.27 ± 0.70	4.50 ± 1.84	6.23 ± 1.47
THP	4.00 ± 1.15	5.00 ± 2.05	4.40 ± 0.97	3.60 ± 1.35	3.70 ± 1.95	4.14 ± 1.58
S2P2	6.20 ± 0.63	4.80 ± 1.32	7.00 ± 1.94	4.93 ± 1.39	4.90 ± 1.29	5.57 ± 1.60
NHP	2.80 ± 0.79	2.70 ± 1.89	3.20 ± 0.42	3.00 ± 1.41	3.60 ± 1.17	3.06 ± 1.24
AttNHP	4.40 ± 0.70	3.70 ± 0.82	5.40 ± 1.17	3.40 ± 1.24	–	4.23 ± 1.25
IFTPP	3.20 ± 2.10	6.00 ± 1.56	2.00 ± 0.00	4.93 ± 3.41	3.60 ± 1.71	3.95 ± 2.44
C-Diff	8.80 ± 0.63	9.00 ± 0.00	8.20 ± 0.92	8.47 ± 0.64	–	8.62 ± 0.69

Table 14: Discrete mean-rank summary. Methods are ranked within every seed–axis comparison that has AP and at least one baseline, with lower ranks better and exact ties receiving average ranks. Cells report the mean rank $\pm$ sample standard deviation over the seed–axis observations for discrete axes. Overall cells use equal-dataset weighting of those observations.

Method	IBM-CC	IBM-AML	MBD-Mini	The Complete Journey	SynSEPA	Overall
AP	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	$\mathbf{1.00 \pm 0.00}$	6.80 ± 0.45	$\mathbf{2.00 \pm 1.33}$	$\mathbf{2.36 \pm 2.35}$
SAHP	7.90 ± 0.32	5.33 ± 1.99	7.40 ± 0.50	8.00 ± 0.00	4.70 ± 1.89	6.67 ± 1.85
RMTPP	6.40 ± 0.52	7.27 ± 0.70	4.95 ± 0.51	2.20 ± 0.84	5.15 ± 1.63	5.19 ± 1.95
THP	4.30 ± 0.48	3.47 ± 1.68	3.25 ± 0.64	3.60 ± 0.55	3.50 ± 2.22	3.62 ± 1.32
S2P2	6.70 ± 0.67	6.80 ± 1.01	7.60 ± 0.50	5.00 ± 0.00	3.50 ± 1.58	5.92 ± 1.74
NHP	2.30 ± 0.48	2.53 ± 0.52	4.35 ± 1.04	$\mathbf{1.60 \pm 0.89}$	4.20 ± 1.14	3.00 ± 1.37
AttNHP	4.50 ± 0.71	4.33 ± 1.45	5.40 ± 0.99	6.20 ± 0.45	–	5.11 ± 1.22
IFTPP	2.90 ± 0.88	5.27 ± 0.80	2.05 ± 0.22	2.60 ± 1.34	4.95 ± 2.50	3.55 ± 1.86
C-Diff	9.00 ± 0.00	9.00 ± 0.00	9.00 ± 0.00	9.00 ± 0.00	–	9.00 ± 0.00

C.3 Causal Prefix Reference

The causal-prefix mean/mode reference is reported in Table 15. It is a deterministic, training-free point predictor applied independently to each episode and target axis. For target row t, it summarizes only the observed rows in the causal prefix $[1, t-1]$: each continuous axis is predicted by the arithmetic mean of its prefix values in raw units, while each discrete axis is predicted by its most frequent prefix label (the mode). If multiple labels have the same frequency, the reference chooses the smallest axis-local metadata ID. It therefore uses the same history available to a next-event predictor without fitting a predictive parameter or representation.

During teacher-forced evaluation, the summary is recomputed at every target position, so it expands as the ground-truth prefix grows. The resulting point forecast anchors continuous axes at their historical level and repeats the most common observed label on each categorical axis. In rollout, the same state is updated by folding each generated complete event into the running continuous sums and categorical counts before predicting the next event. The table reports the reference separately from the learned methods for the teacher-forced task.

Because this reference has no objective, optimizer, scheduler, or learnable parameters, it has no checkpoint-selection step. It is nevertheless used when selecting the learned models: on the HPO validation subset, each candidate's per-axis MAE or $1 - \mathrm{accuracy}$ is divided by the corresponding reference error, and these ratios are averaged. Keeping the reference outside the learned-method comparison and mean-rank population preserves those summaries for comparisons among learned methods while exposing improvement over a simple causal historical statistic.

Using the displayed means, AP is better than this reference on 20 of the 23 axes, tied on MBD-Mini elapsed time, and worse on The Complete Journey elapsed time and store identity. The Complete

Journey results show that a simple historical statistic can remain difficult to beat on some targets even when the learned system is stronger overall. This training-free reference remains outside the headline comparison and mean-rank population.

Table 15: Separate causal prefix mean/mode reference. This training-free method is excluded from best/second-best emphasis and mean rank. Values are mean ± standard deviation over 5 completed seeds.

Dataset	Axis	Metric	Mean ± SD
IBM-CC	delta_time_hours	MAE	26.1 ± 0.00
IBM-CC	amount	MAE	40.0 ± 0.00
IBM-CC	mcc	Accuracy	0.226 ± 0.00
IBM-CC	merchant_city	Accuracy	0.593 ± 0.00
IBM-AML	delta_time_hours	MAE	5.11 ± 0.00
IBM-AML	to_bank	Accuracy	0.428 ± 0.00
IBM-AML	amount_received	MAE	$5.13\times10^{6} \pm 0.00$
IBM-AML	receiving_currency	Accuracy	0.975 ± 0.00
IBM-AML	payment_format	Accuracy	0.373 ± 0.00
MBD-Mini	delta_time_days	MAE	3.42 ± 0.00
MBD-Mini	amount	MAE	$1.28\times10^{6} \pm 0.00$
MBD-Mini	event_type	Accuracy	0.555 ± 0.00
MBD-Mini	event_subtype	Accuracy	0.553 ± 0.00
MBD-Mini	src_type11	Accuracy	0.914 ± 0.00
MBD-Mini	dst_type11	Accuracy	0.548 ± 0.00
The Complete Journey	delta_time_days	MAE	5.22 ± 0.00
The Complete Journey	store_id	Accuracy	0.696 ± 0.00
The Complete Journey	total_sales	MAE	22.5 ± 0.00
The Complete Journey	number_of_items	MAE	$1.36\times10^{3} \pm 0.00$
SynSEPA	delta_time_days	MAE	1.75 ± 0.00
SynSEPA	beneficiary_iban	Accuracy	0.0497 ± 0.00
SynSEPA	remittance_category	Accuracy	0.276 ± 0.00
SynSEPA	amount	MAE	$1.93\times10^{3} \pm 0.00$

C.4 Pointer/direct expert competition ablation

The AP system allows the pointer and direct experts to compete during training on every eligible axis and uses validation evidence to select the expert deployed for that axis. The direct-only and pointer-only ablations fix one output geometry throughout while retaining AP's backbone and applicable optimization settings. They therefore intervene on expert competition rather than merely recording which expert the final model selected.

Table 16 gives the paired aggregate view. Regular AP's dataset-balanced win rates are 68.4% against direct-only AP and 67.6% against pointer-only AP. The aggregate favors competition in both comparisons, but the effect is heterogeneous. Against direct-only AP, the dataset rates range from 96.0% on IBM-AML to 25.0% on The Complete Journey and fall below half only on The Complete Journey. Against pointer-only AP, regular AP wins every matched comparison on MBD-Mini and reaches 70.0% on SynSEPA, but only 25.0% on The Complete Journey. Axes and seeds are equally weighted within a dataset and datasets are equally weighted overall, so schemas with more axes do not dominate the aggregate. Exact ties count as half a win.

Table 17: Pointer/direct expert competition ablation for Axial Pointer. The single-expert ablations use the direct or pointer expert throughout while sharing AP's backbone and applicable optimization settings. All prediction axes are shown. Values are mean ± standard error over 5 independent training seeds per method. Bold marks the best unrounded mean within each row. Standard errors whose formatted numeric strings exceed 10 characters are shown as high.

Dataset	Axis (metric)	AP	Direct-only AP	Pointer-only AP
IBM-CC	delta_time_hours (MAE ↓)	$\mathbf{23.2 \pm 0.03}$	23.4 ± 0.06	23.4 ± 0.01
	amount (MAE ↓)	37.3 ± 0.16	$\mathbf{37.2 \pm 0.04}$	37.5 ± 0.02
	mcc (Acc. ↑)	0.302 ± 0.00	$\mathbf{0.303 \pm 0.00}$	0.298 ± 0.00
	merchant_city (Acc. ↑)	0.642 ± 0.00	0.613 ± 0.00	$\mathbf{0.642 \pm 0.00}$
IBM-AML	delta_time_hours (MAE ↓)	2.87 ± 0.02	5.02 ± 0.87	$\mathbf{2.85 \pm 0.01}$
	to_bank (Acc. ↑)	$\mathbf{0.765 \pm 0.01}$	0.431 ± 0.04	0.741 ± 0.01
	amount_received (MAE ↓)	$\mathbf{2.82{\times}10^6 \pm 103{,}293.93}$	$4.32{\times}10^6 \pm 144{,}149.87$	$2.84{\times}10^6 \pm 28{,}061.80$
	receiving_currency (Acc. ↑)	$\mathbf{0.993 \pm 0.00}$	0.990 ± 0.00	0.992 ± 0.00
	payment_format (Acc. ↑)	$\mathbf{0.682 \pm 0.00}$	0.637 ± 0.01	0.670 ± 0.00
MBD-Mini	delta_time_days (MAE ↓)	$\mathbf{3.42 \pm 0.02}$	3.42 ± 0.01	3.49 ± 0.01
	amount (MAE ↓)	$\mathbf{6.66{\times}10^5 \pm 15.88}$	$6.67{\times}10^5 \pm 463.40$	$6.66{\times}10^5 \pm 105.68$
	event_type (Acc. ↑)	$\mathbf{0.567 \pm 0.00}$	0.567 ± 0.00	0.556 ± 0.00
	event_subtype (Acc. ↑)	$\mathbf{0.566 \pm 0.00}$	0.565 ± 0.00	0.554 ± 0.00
	src_type11 (Acc. ↑)	$\mathbf{0.922 \pm 0.00}$	0.922 ± 0.00	0.918 ± 0.00
	dst_type11 (Acc. ↑)	$\mathbf{0.562 \pm 0.00}$	0.561 ± 0.00	0.549 ± 0.00
The Complete Journey	delta_time_days (MAE ↓)	5.25 ± 0.02	$\mathbf{5.16 \pm 0.01}$	5.17 ± 0.01
	store_id (Acc. ↑)	0.676 ± 0.00	0.687 ± 0.00	$\mathbf{0.698 \pm 0.00}$
	total_sales (MAE ↓)	22.3 ± 0.09	21.5 ± 0.03	$\mathbf{21.5 \pm 0.03}$
	number_of_items (MAE ↓)	$\mathbf{857 \pm 1.88}$	$6.46{\times}10^{14} \pm high$	870 ± 1.78
SynSEPA	delta_time_days (MAE ↓)	$\mathbf{1.66 \pm 0.00}$	1.67 ± 0.00	1.66 ± 0.00
	beneficiary_iban (Acc. ↑)	0.103 ± 0.00	0.0399 ± 0.00	$\mathbf{0.103 \pm 0.00}$
	remittance_category (Acc. ↑)	0.292 ± 0.00	$\mathbf{0.292 \pm 0.00}$	0.271 ± 0.00
	amount (MAE ↓)	$\mathbf{1.77{\times}10^3 \pm 1.34}$	$1.78{\times}10^3 \pm 0.87$	$1.77{\times}10^3 \pm 1.95$

Table 16: Aggregate expert-competition ablation. Each entry is regular AP's pairwise win rate against each single-expert ablation. Rates compare unrounded per-seed metrics over matched prediction axis-seed pairs, with exact ties counting as half a win. Axes and seeds are equally weighted within each dataset; Overall equally weights datasets.

Dataset	AP Win Rate vs Direct-only AP (%)	AP Win Rate vs Pointer-only AP (%)
IBM-CC	62.5	75.0
IBM-AML	96.0	68.0
MBD-Mini	73.3	100.0
The Complete Journey	25.0	25.0
SynSEPA	85.0	70.0
Overall	68.4	67.6

The per-axis results in Table 17 clarify the aggregate. Regular AP has the best or tied-best unrounded mean on 14 of the 23 axes, including all six MBD-Mini axes, but the preferred policy is not uniform. Direct-only AP is strongest on such axes as IBM-CC *amount* and *mcc*, The Complete Journey elapsed time, and SynSEPA remittance category. Pointer-only AP is strongest on IBM-CC *merchant city*, The Complete Journey store identity and total sales, and SynSEPA beneficiary IBAN. This variation is the empirical premise for competition, while the Complete Journey block also demonstrates that adaptation can select a policy whose final mean is worse than a fixed-expert ablation.

The high-cardinality categorical axes expose the cost of forcing direct prediction. Relative to direct-only AP, regular AP improves accuracy from 0.613 to 0.642 on IBM-CC *merchant city*, from 0.431 to 0.765 on IBM-AML *to bank*, and from 0.0399 to 0.103 on SynSEPA *beneficiary IBAN*. These axes are pointer-only in regular AP by design, so the comparison is consistent with history-local output being useful when the global label space is large. The Complete Journey result prevents that observation from being generalized to all categorical axes: both fixed-expert ablations exceed AP on *store id*.

Because the single-expert ablations are retrained under fixed-expert interventions, these comparisons isolate the contribution of the competition policy within this experimental setup more directly than deployed-expert annotations alone. The result is an average benefit rather than a dataset-uniform

one. They report standard errors and paired win rates but no hypothesis test or confidence interval for the aggregate. They therefore support a qualified claim about the evaluated architecture and routing protocol, not a guarantee that adaptive AP improves every dataset or axis.

C.5 Efficiency and speed

For the five reported datasets, Table 18 reports the instantiated learnable parameter count together with mean wall-clock training time and mean teacher-forced test prediction time. Timing values aggregate completed non-resumed seeds. They are implementation-level measurements for the reported experiments, rather than a controlled hardware or asymptotic scaling benchmark.

Across these five reported datasets, AP is not the fastest method. Its training times on IBM-CC, IBM-AML, MBD-Mini, The Complete Journey, and SynSEPA are 4.37, 5.54, 7.72, 0.749, and 3.49 hours, respectively, compared with per-dataset fastest times of 0.619, 0.686, 0.742, 0.361, and 0.861 hours. The corresponding AP test prediction times are 276, 293, 472, 16.6, and 198 seconds, whereas the fastest methods take 2.81, 5.48, 6.04, 0.416, and 4.14 seconds. Thus AP's predictive strength in the reported comparison comes with substantial measured wall-clock cost under this setup: training is roughly two to ten times slower and inference is roughly forty to one hundred times slower than the fastest method on each dataset.

Parameter count does not have the same ranking as speed. AP has 10.6M parameters on IBM-CC, 22.9M on IBM-AML, 15.2M on MBD-Mini, 22.5M on The Complete Journey, and 10.6M on SynSEPA. It is the largest model on MBD-Mini and The Complete Journey, second-largest on IBM-AML, fourth-largest on IBM-CC, and the smallest completed model on SynSEPA. Conversely, C-Diff has only 6.51M parameters on IBM-CC but requires 96.2 seconds for test prediction, while IFTPP has 13.3M parameters and requires 2.81 seconds. These comparisons show that instantiated parameter count is not a sufficient proxy for end-to-end efficiency; method-specific operations and dataset workload also matter.

The broader lesson is to distinguish path-specific scalability from whole-system efficiency. The pointer-routed categorical path can remove learned per-class input and output terms as vocabulary grows, but AP still pays for its shared backbone, pointer scoring over the history, route and expert components, and persistent registry state; temporal and pointer computation can also depend on prefix length. The table therefore qualifies rather than contradicts the scalability argument: avoiding vocabulary-sized parameter terms is a structural property, not a promise of fewer total parameters, lower training time, or lower inference latency. Efficiency should be reported alongside predictive quality, and stronger systems conclusions would require controlled hardware, batch, sequence-length, and implementation conditions.

Table 18: Method efficiency and speed across the reported experiments. Number of parameters is the instantiated learnable parameter total. Training time is mean wall-clock optimization time in hours, and inference time is mean teacher-forced test prediction time in seconds, aggregated over completed non-resumed seeds. Best and second-best unrounded means within each dataset are bold and underlined, respectively.

Method	Dataset	Number of parameters	Training time (h)	Inference time (s)
AP	IBM-CC	10,607,091	4.37	276
SAHP	IBM-CC	37,674,359	0.835	9.33
RMTPP	IBM-CC	9,963,064	0.863	3.18
THP	IBM-CC	10,583,544	<u>0.700</u>	<u>2.97</u>
S2P2	IBM-CC	8,792,056	0.920	17.1
NHP	IBM-CC	15,340,664	0.874	11.2
AttNHP	IBM-CC	<u>8,285,560</u>	4.15	260
IFTPP	IBM-CC	13,251,717	**0.619**	**2.81**
C-Diff	IBM-CC	**6,511,737**	1.92	96.2
AP	IBM-AML	22,893,598	5.54	293
SAHP	IBM-AML	32,397,659	0.947	12.2
RMTPP	IBM-AML	4,115,612	<u>0.774</u>	<u>5.54</u>
THP	IBM-AML	7,409,244	0.863	6.60
S2P2	IBM-AML	**1,649,884**	1.27	27.6
NHP	IBM-AML	10,063,964	1.00	14.2
AttNHP	IBM-AML	7,359,436	3.21	182
IFTPP	IBM-AML	<u>3,597,417</u>	**0.686**	**5.48**
C-Diff	IBM-AML	10,600,558	5.39	331
AP	MBD-Mini	15,239,633	7.72	472
SAHP	MBD-Mini	6,620,493	0.921	10.1
RMTPP	MBD-Mini	636,302	<u>0.749</u>	**6.04**
THP	MBD-Mini	7,253,966	0.802	7.51
S2P2	MBD-Mini	**100,174**	1.13	32.6
NHP	MBD-Mini	<u>282,830</u>	0.938	13.0
AttNHP	MBD-Mini	692,942	3.62	197
IFTPP	MBD-Mini	504,027	**0.742**	<u>6.26</u>
C-Diff	MBD-Mini	8,305,649	6.94	425
AP	The Complete Journey	22,484,485	0.749	16.6
SAHP	The Complete Journey	5,122,027	0.562	0.860
RMTPP	The Complete Journey	727,660	0.427	**0.416**
THP	The Complete Journey	5,010,027	<u>0.410</u>	0.493
S2P2	The Complete Journey	**211,948**	0.418	1.61
NHP	The Complete Journey	<u>323,436</u>	0.492	0.996
AttNHP	The Complete Journey	414,444	0.721	15.2
IFTPP	The Complete Journey	1,956,217	**0.361**	<u>0.461</u>
C-Diff	The Complete Journey	6,687,884	1.02	24.8
AP	SynSEPA	**10,568,591**	3.49	198
SAHP	SynSEPA	166,343,556	1.35	6.32
RMTPP	SynSEPA	221,585,029	1.19	6.69
THP	SynSEPA	202,994,437	1.38	<u>4.56</u>
S2P2	SynSEPA	80,117,381	1.15	15.6
NHP	SynSEPA	80,300,037	<u>1.11</u>	7.10
AttNHP	SynSEPA	–	–	–
IFTPP	SynSEPA	<u>40,216,137</u>	**0.861**	**4.14**
C-Diff	SynSEPA	–	–	–