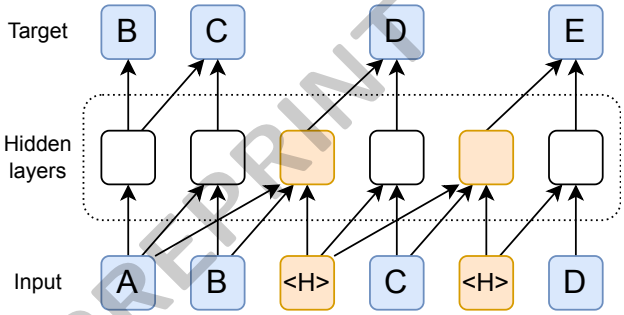
HT-Transformer: Event Sequences Classification by Accumulating Prefix Information with History Tokens

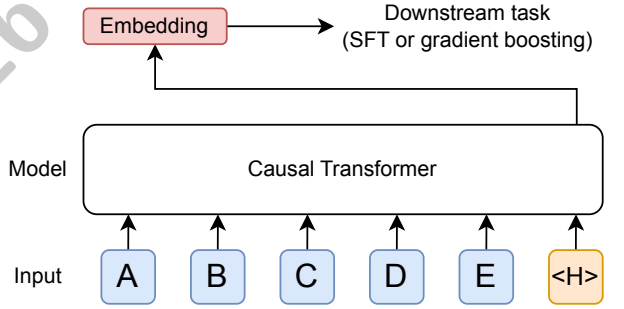
Ivan Karpukhin[1] , **Maksim Polesskii**[1,2] , **Andrey Savchenko**[1,3]
[1] Sber AI Lab, Russia
[2] Lomonosov Moscow State University, Russia
[3] HSE University, Laboratory for Theoretical Foundations of AI Models, Russia
karpuhini@yandex.ru, avsavchenko@hse.ru

Abstract

Deep learning has achieved strong results in modeling sequential data, including event sequences, temporal point processes, and irregular time series. Recently, transformers have largely replaced recurrent networks in these tasks. However, transformers often underperform recurrent networks in classification tasks that aim to predict future targets, such as churn, user reactions, or treatment response. The reason behind this performance gap remains largely underexplored. In this paper, we identify a key limitation of transformers: the lack of a single vector representation that compactly summarizes the evolving state of a sequence. We further show that commonly used contrastive embeddings are poorly suited to capturing the local context needed for accurate forward-looking prediction. To address these challenges, we introduce history tokens, a novel concept that enables the accumulation of historical information during next-token prediction pretraining. Our approach significantly improves transformer-based models, achieving impressive results in finance, e-commerce, and healthcare tasks. The code is publicly available: https://github.com/ivan-chai/pretpp.



(a) History Token serves as a bottleneck during pretraining.



(b) The embedding of History Token is used in downstream tasks.

Figure 1: History tokens accumulate prefix information during pretraining via next-token-prediction. The embedding of the History Token is later used in downstream tasks.

1 Introduction

Many real-world problems involve predicting future events from historical observations. In generative tasks, the goal is to forecast events similar to those previously observed [Xue *et al.*, 2024]. However, many practical applications require anticipating events that do not explicitly appear in the training history. Examples include high-impact tasks in fintech, e-commerce, and medical analytics, such as predicting loan default, customer churn, and disease onset. These scenarios are typically addressed using classical machine learning models, such as logistic regression or gradient boosting, applied to handcrafted features or unsupervised model-based embeddings derived from historical data [Osin *et al.*, 2025; Synerise, 2025].

Deep learning has achieved significant success in modeling sequential structures, including event sequences, temporal point processes, and time series. A prominent trend is the adoption of pretrained Transformer architectures due to their ability to capture long-range dependencies and complex temporal patterns [Padhi *et al.*, 2021; Zuo *et al.*, 2020]. Unlike recurrent neural networks, however, Transformers lack a canonical mechanism for extracting a fixed-size embedding from a sequence, as information is distributed across all token activations. This issue is commonly mitigated through auxiliary objectives during pretraining, such as contrastive learning [BehnamGhader *et al.*, 2024], sentence order prediction [Lan *et al.*, 2020], or next-sentence prediction [Devlin *et al.*, 2019]. However, each of these approaches introduces limitations. For instance, it is well documented that contrastive pretraining may overemphasize “easy features”,

compromising downstream quality [Robinson *et al.*, 2021]. Consequently, the problem of learning robust and informative sequence embeddings, including methods based on the next-token prediction (NTP) objective [Yenduri *et al.*, 2024], remains an open research question.

In this work, we propose a novel approach to pretraining Transformer-based embeddings without relying on auxiliary tasks. Our method draws inspiration from recurrent architectures and leverages sparse attention masks to guide the accumulation of historical information [Bulatov *et al.*, 2022]. Specifically, we introduce *history tokens* that gather and summarize contextual information during training via a standard next-token prediction objective, as illustrated in Figure 1. We empirically evaluate the resulting embeddings across multiple domains, including finance, e-commerce, and healthcare, and show that they offer strong predictive performance, especially for tasks focused on forecasting future events rather than global sequence classification.

The contributions of this paper are as follows:

1. We propose a novel HT-Transformer architecture that employs history tokens to accumulate past information during pretraining using only the next-token prediction objective.
2. We develop advanced strategies for selecting history token positions and applying attention masking to improve downstream performance.
3. We introduce strong baselines for embedding extraction by utilizing adapted variants of the Recurrent Memory Transformer and Longformer.
4. We demonstrate that the proposed HT-Transformer is especially effective for forecasting future events, consistently outperforming widely used transformer models and achieving strong results across multiple benchmarks in finance, e-commerce, and healthcare.

2 Preliminaries on Event Sequences

Consider sequences of discrete events $S = \{s_i\}_{i=1}^{N}$, where each event s_i is represented by a collection of fields, including a timestamp t_i, optional numerical attributes, and categorical variables. Each sequence typically corresponds to a single entity, such as a user or client, and the events are ordered chronologically by their timestamps: $t_1 \leq t_2 \leq \cdots \leq t_N$. An illustration of event sequences is provided in Figure 2.

Data preprocessing. Before feeding data into a deep model, each event in the sequence must be transformed into an embedding in a latent space. In a typical pipeline, each data field is encoded independently, and the resulting embeddings are concatenated to form a single event representation [Gorishniy *et al.*, 2021]. Categorical features are transformed by assigning a trainable embedding vector to each possible value. Numerical features are incorporated directly into the event embedding without additional preprocessing.

We apply time-based positional encoding when using Transformer models, following the approach proposed in prior work [Yang *et al.*, 2022]. Specifically, for each timestamp t, we compute a positional embedding $\mathrm{PE}_i(t)$ of dimension d as:

$$\mathrm{PE}_i(t) = \begin{cases} \sin\left(t/(m * (\frac{5M}{m})^{\frac{i}{d}})\right), & \text{if } i \text{ is even} \\ \cos\left(t/(m * (\frac{5M}{m})^{\frac{i-1}{d}})\right), & \text{otherwise} \end{cases} \quad (1)$$

where m and M are constants determined from the distribution of timestamp values. The precise hyperparameters are listed in Appendix D in the supplementary materials.

Pretraining. We consider sequence-level classification tasks, where each sequence S is associated with a single target label l. Because labels are assigned at the sequence level rather than the event level, the number of labeled examples is often much smaller than the total number of events. This imbalance motivates the development of unsupervised pretraining algorithms that can leverage the abundance of unlabeled sequential data to improve downstream performance [Osin *et al.*, 2025].

Unsupervised pretraining on event sequences is typically based on either generative or contrastive learning objectives.

In the *generative* approach, the model is trained to predict the next event s_{i+1} given the historical context $s_1, \ldots, s_i$, thereby encouraging it to capture temporal dependencies and sequential structure. A typical NTP objective is formulated as a weighted sum of individual losses over each data field [Shchur *et al.*, 2020; Padhi *et al.*, 2021; McDermott *et al.*, 2023]:

$$\mathcal{L}_{\mathrm{NTP}}(\hat{s}, s) = \sum_{i=1}^{F} \alpha_i \mathcal{L}_i(\hat{s}^i, s^i), \quad (2)$$

where $\hat{s} = (\hat{s}^1, \ldots, \hat{s}^F)$ denotes the predicted event with F fields, $s = (s^1, \ldots, s^F)$ is the ground-truth event, and $\mathcal{L}_i$ is a field-specific loss determined by the type of the i-th field. The individual losses are combined using weights α_i.

Timestamps can be predicted using standard regression losses such as Mean Absolute Error (MAE) or Mean Squared Error (MSE), or through more expressive temporal point process models based on intensity functions [Rizoiu *et al.*, 2017; Zuo *et al.*, 2020]. In the current work, we use MAE to predict inter-event times:

$$\mathcal{L}_{\mathrm{MAE}}(\Delta\hat{t}, \Delta t) = |\Delta\hat{t} - \Delta t|, \quad (3)$$

where $\Delta\hat{t}$ is the predicted inter-event time and Δt is the ground truth. We also apply MAE to other numerical fields, such as transaction amount. For categorical fields, we use cross-entropy:

$$\mathcal{L}_{\mathrm{CE}}(\hat{p}(\cdot), l) = -\log \hat{p}(l), \quad (4)$$

where $\hat{p}(\cdot)$ is the predicted label distribution and l is the ground-truth category. The exact combination of MAE and cross-entropy terms depends on the dataset. The total number of fields F for each dataset is reported in Table 1.

An alternative to generative modeling is *contrastive learning* [Babaev *et al.*, 2022], which aims to learn sequence representations by maximizing agreement between different augmented views of the same sequence and pushing apart views from different sequences. Typically, each sequence is divided into multiple, possibly overlapping, subsequences $R_k \subset S$

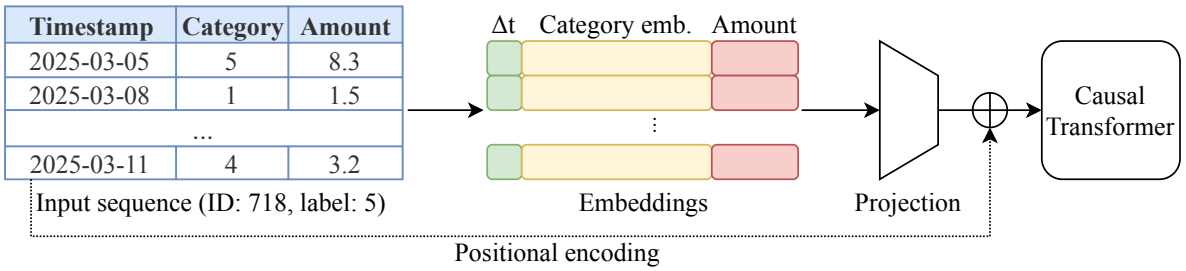


Figure 2: Event sequences preprocessing.

for $k = 1, \ldots, K$. Let $\mathrm{ID}(R)$ denote the index of the original sequence from which a chunk R was derived. Then the contrastive loss [Chopra *et al.*, 2005] is defined as:

$$\mathcal{L}_{\mathrm{CoLES}}(R_i, R_j) = \begin{cases} \|f(R_i) - f(R_j)\|^2, & \mathrm{ID}(R_i) = \mathrm{ID}(R_j) \\ \max\left(0, \epsilon - \|f(R_i) - f(R_j)\|\right)^2, & \text{otherwise} \end{cases} \tag{5}$$

where $f(R) \in \mathbb{R}^d$ is the embedding of a subsequence R produced by the model. Following prior works [Babaev *et al.*, 2022; Sakhno *et al.*, 2025], we use $\epsilon = 0.5$ and $K = 5$ subsequences per sequence.

Both generative and contrastive paradigms have been successfully adapted to neural architectures such as recurrent neural networks (RNNs) and Transformers. However, while effective for specific tasks, these approaches exhibit notable limitations, especially when the goal is to anticipate future events rather than to summarize past behavior. Overcoming these limitations is a key motivation behind the approach proposed in this work.

Downstream tasks. Embeddings extracted from pretrained models are widely used for downstream classification and regression tasks, either via a fully connected prediction head [Synerise, 2025] or in combination with gradient boosting methods [Babaev *et al.*, 2022]. Following prior work on event sequences [Babaev *et al.*, 2022; Osin *et al.*, 2025], we apply gradient boosting on top of pretrained embeddings for downstream classification. We discuss the motivation for this choice and compare gradient boosting with an MLP head in Appendix E.

3 Proposed Method

The core idea of the proposed method is to introduce special *history tokens* into Transformer models. These tokens are designed to accumulate information from preceding tokens in the sequence. A carefully constructed attention mask ensures that these tokens act as an information bottleneck, similar in function to the hidden states in RNNs. In the following, we describe the training procedure and the application of history tokens to downstream classification tasks.

3.1 Unsupervised Pretraining with History Tokens

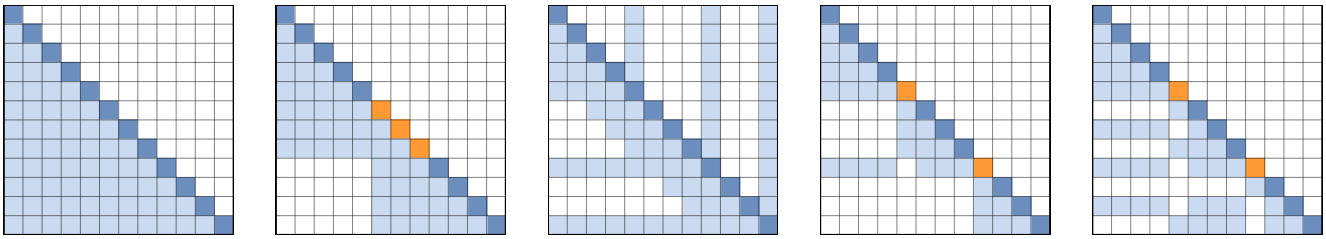
The proposed approach is compatible with any Transformer architecture that employs a causal attention mask (Fig. 3a), where each token attends only to preceding tokens. Transformers for event sequences typically have three primary components: an event embedder, a backbone, and a prediction head. History tokens are inserted into the backbone input after event embeddings are computed, as shown in Figure 1a. Each history token is assigned the timestamp of a preceding event. These timestamps are then used for positional encoding.

To enable history tokens to serve as memory units, we modify the backbone's attention mask. Each history token can attend to all preceding event tokens (except other history tokens), thereby accumulating prefix information. In contrast, event tokens can attend only to history tokens and to events between the current position and the most recent history token. This attention pattern is illustrated in Figure 3d. When multiple history tokens are present, we introduce two attention strategies. In the *Last* strategy (Figure 3d), each event token is restricted to attend only to the most recent preceding history token. In the *Random* strategy, illustrated in Figure 3e, each event token randomly selects one of the prior history tokens during attention computation. As demonstrated in our experiments, the Random strategy consistently yields better performance across various tasks.

The proposed method provides considerable flexibility in selecting both the number and the placement of history tokens. For a given sequence of length L, the number of history tokens is computed as $\max(1, \lfloor fL \rceil)$, where $\lfloor \cdot \rceil$ is a rounding operator and f is a *frequency* hyperparameter.

We also implement two strategies for inserting history tokens into the sequence. The *Uniform* strategy inserts history tokens at uniformly sampled positions. However, this approach can lead to a discrepancy between training and inference, as history tokens are typically positioned near the end of the sequence during evaluation. To address this issue, we introduce the *Bias-End* strategy, which places history tokens closer to the end of the sequence. Specifically, it samples positions uniformly within the range $[\mu/2, L]$, where μ is the mean sequence length in the batch, and L is the maximum sequence length. Our experiments demonstrate that the Bias-End strategy improves downstream performance in the considered tasks.

Finally, to align pretraining and inference, we introduce one more modification. At inference time, a single history token is appended to the end of the sequence and attends to the entire prefix through the causal masking scheme. The hidden representation of this token is then used as the sequence



(a) Causal. (b) Rec. Memory. (c) Longformer. (d) HT Last. (e) HT Random.

Figure 3: Comparison of attention masks. Special tokens are orange-colored.

embedding for downstream classification. This creates a bottleneck analogous to the pretraining setup, although inference uses only one history token instead of multiple intermediate tokens. To reduce the mismatch between training and inference, we apply history tokens only to a subset of pretraining batches using *application probability* $p = 0.5$.

3.2 Classification

During embedding extraction, a single history token is appended to the end of the input sequence, and the average of its hidden activations across Transformer layers is used as the sequence-level embedding (Figure 1b). This embedding then serves as an input feature for a downstream gradient boosting classifier.

4 Related Work

Transformer models have been successfully applied to sequence classification tasks, especially in natural language processing (NLP) [Vaswani *et al.*, 2017]. One of the main challenges in this setting is the limited availability of labeled data, which has driven the development of effective unsupervised pretraining strategies [Muennighoff *et al.*, 2023]. A notable early approach is BERT [Devlin *et al.*, 2019], which introduced a masked language modeling (MLM) objective alongside a next sentence prediction (NSP) task to enable powerful representation learning. Fine-tuned BERT models have been widely adopted for various classification tasks such as natural language understanding (NLU) [Wang *et al.*, 2019].

A central issue in applying Transformers to NLU is extracting a compact, semantically meaningful representation of an entire sequence. In BERT, this was addressed by introducing a special classification token trained via the NSP objective. However, subsequent work such as RoBERTa [Liu *et al.*, 2019] challenged the necessity of the NSP task, demonstrating that it could be omitted without degrading performance.

Beyond BERT-style objectives, other works have explored contrastive pretraining techniques, such as LLM2Vec [BehnamGhader *et al.*, 2024], which aim to bring semantically similar sequences closer in embedding space. Although contrastive learning can perform well across a range of tasks, it has notable limitations. In particular, models can rely on "easy" features, such as surface-level similarities, to distinguish positive pairs, bypassing the need for deeper semantic understanding. Furthermore, contrastive pretraining tends to bias models toward capturing global sequence properties at the expense of local or up-to-date information. This bias poses a particular challenge in event sequence modeling, where the most recent context is often essential for accurate prediction, in contrast to many natural language processing tasks emphasizing global semantics.

Alternative methods for sequence embedding extraction include averaging token activations across specific layers or using the final token's activation [Stankevičius and Lukoševičius, 2024]. However, these approaches generally underperform specialized embedding pretraining techniques, particularly for tasks that require nuanced or fine-grained representations.

In contrast, RNNs naturally summarize sequences through the hidden state at the final timestep, which compactly encodes the information needed for future prediction. This property has inspired the integration of recurrent principles into Transformer architectures, particularly for modeling long sequences. For instance, *Recurrent-Memory Transformers* (RMT) recursively apply a Transformer model to chunks of the input sequence, using special tokens to pass information between successive segments [Bulatov *et al.*, 2022]. The corresponding attention mask is shown in Figure 3b. Similar ideas are adopted in architectures such as *Longformer* [Beltagy *et al.*, 2020], where global tokens aggregate and propagate information across extended contexts (Figure 3c). More recently, recurrent-style Transformers have been combined with contrastive learning objectives to attain strong results on natural language understanding (NLU) tasks, while retaining the generative abilities of causal models [Zhang *et al.*, 2025].

Our work extends recurrent-style Transformer architectures toward representation learning for event-sequence classification. While prior approaches such as RMT and Longformer primarily target efficient long-context modeling, HT-Transformer uses history tokens as predictive bottlenecks during standard next-token prediction pretraining. Unlike RMT, the HT-Transformer processes a sequence in a single pass while retaining activations from all layers, rather than only the last, for subsequent token processing. In contrast to Longformer, HT-Transformer employs special tokens instead of solely modifying the attention matrix. Moreover, it restricts attention to future tokens, forcing history tokens to accumulate local information.

A detailed comparison of the proposed history tokens with other special tokens, such as `[CLS]`, `[SEP]`, `[SOS]`, and `[EOS]`, is provided in Appendix B.

Dataset	# Seq.	Events	Fields	Mean length	Mean duration	Time unit	Downstream Target	Downstream Classes	Downstream Metric
Churn	10217	1M	6	99.3	80.5	Day	Churn	2	ROC AUC
AgePred	50000	44M	3	875	718	Day	Age group	4	Accuracy
Alfabattle	1466527	343M	15	234	275	Day	Default	2	ROC AUC
MIMIC-III	52103	23M	3	407	108	Day	Mortality	2	ROC AUC
Taobao	9904	5M	3	527	12.9	Day	Activity	2	ROC AUC
MBD mini	98721	37M	13	372	148	4 Days	Credit	2	ROC AUC

Table 1: Datasets statistics

5 Experimental Results

Setup. We conduct experiments on datasets spanning multiple domains. The Churn[1], AgePred[2], and Alfabattle[3] datasets represent a range of downstream tasks in the financial domain. The Taobao dataset[4] represents user interactions in e-commerce, and MIMIC-III [Johnson *et al.*, 2016] is a widely used collection of medical records. Summary statistics for these datasets are provided in Table 1.

We evaluate four primary baseline approaches: supervised learning, NTP pretraining [Radford *et al.*, 2018], BERT [Devlin *et al.*, 2019], and contrastive learning (CoLES) [Babaev *et al.*, 2022]. Each method is applied to two backbone architectures. For RNN-based models, we use a GRU backbone [Cho *et al.*, 2014], while Transformer-based models employ a decoder-only architecture [Radford *et al.*, 2018].

All models are trained using the Adam optimizer [Kingma and Ba, 2015] with a fixed learning rate of 0.001. The maximum number of training epochs varies by dataset and ranges from 60 to 120 (Appendix D). Early stopping is applied based on validation performance to prevent overfitting. Experiments were conducted on NVIDIA A100 GPUs. All datasets were trained on a single GPU, except Alfabattle, for which we used two GPUs in some experiments.

Hyperparameters, including the loss weights for the NTP objective and model size (Appendix D), are optimized using a Bayesian optimizer [Snoek *et al.*, 2012] applied to the NTP RNN baseline. The resulting configurations are reused across all other RNN settings. For Transformer models, we separately tune the number of layers and the hidden dimension using the NTP configuration and apply these settings consistently across all Transformer-based variants.

We report the mean and standard deviation across five seeds for all methods, except Alfabattle, for which we use three seeds due to computational constraints.

To assess the quality of extracted embeddings, we train a gradient boosting classifier for each downstream task using the LightGBM library [Ke *et al.*, 2017]. The classifier is trained on frozen embeddings and uses the same hyperparameters as in the CoLES baseline [Babaev *et al.*, 2022].

Appendix A contains further reproducibility details.

Baselines implementation details. Two of our baselines, Recurrent Memory Transformer (RMT) [Synerise, 2025] and Longformer [Beltagy *et al.*, 2020], were initially proposed to address the limited scalability of standard Transformers, whose complexity grows quadratically with input length. These models were not explicitly designed for embedding extraction and therefore require additional modifications when applied to classification tasks.

RMT was originally applied to fixed-length chunks of the input sequence. We follow the same procedure during pretraining but append memory tokens to the end of each chunk for embedding extraction. The final embedding is obtained by averaging the memory token activations.

The Longformer architecture was introduced for bidirectional Transformers, which are incompatible with next-token prediction pretraining. To adapt it, we modify the attention mask to enforce causality for regular tokens while allowing global tokens to attend to all tokens, including future ones. Conversely, all tokens can attend to global tokens regardless of relative position. An illustration of our modified Longformer mask is provided in Figure 3c. In addition, we observed that convolutional attention leads to suboptimal performance. Instead, we integrate global tokens directly into the causal attention mask. We further found that using a single global token at the end of each sequence consistently outperforms configurations with multiple tokens placed at regular or random positions. As a result, our final Longformer variant employs a single global token without convolutional masking, and the sequence embedding is obtained by averaging the activations of the last token.

To the best of our knowledge, we are applying RMT and Longformer to extract embeddings for event sequence classification tasks for the first time.

5.1 Classification of Event Sequences

Classification results are presented in Table 2. Among baselines in the unsupervised setting, the standard NTP Transformer significantly outperforms its RNN counterpart only on the MIMIC-III dataset, while performing worse on Churn, AgePred, and Alfabattle. This highlights the limitations of traditional Transformer architectures in learning compact and informative sequence representations for downstream tasks.

The proposed HT-Transformer effectively addresses these limitations. It consistently outperforms the NTP Transformer in all comparisons. At the same time, contrastively pretrained RNN (CoLES) surpasses HT-Transformer on AgePred and Taobao datasets. However, on Taobao, the difference is not statistically significant. The same is true for BERT, which uses CoLES for `[CLS]` pretraining (Appendix C).

[1]https://boosters.pro/championship/rosbank1/

[2]https://ods.ai/competitions/sberbank-sirius-lesson

[3]https://boosters.pro/championship/alfabattle2/overview

[4]https://tianchi.aliyun.com/dataset/46

Method	Churn	AgePred	Alfabattle	MIMIC-III	Taobao	MBD mini	AVG
Supervised RNN	79.10± 0.80	61.18± 0.49	76.47± 1.13	91.46± 0.10	84.91± 1.17	82.17± 1.01	79.21
Supervised Transformer	78.88± 0.90	54.88± 2.37	74.90± 0.08	77.48± 1.22	79.71± 1.68	79.11± 1.34	74.16
NTP RNN	81.56± 0.59	60.05± 0.29	79.83± 0.05	90.68± 0.07	83.28± 1.42	**87.78± 0.50**	80.53
NTP Transformer	80.92± 0.66	56.16± 0.51	78.63± 0.12	91.28± 0.10	83.39± 1.43	86.93± 0.21	79.55
NTP Rec. Mem. Transf.	80.23± 0.21	60.15± 0.55	80.25± 0.05	91.58± 0.11	84.48± 1.22	87.04± 0.15	80.62
NTP Longformer	82.60± 0.26	48.29± 0.51	65.91± 0.34	86.41± 0.18	82.91± 1.04	86.10± 0.25	75.37
BERT (BiRNN)	82.00± 0.44	**62.49± 0.33**	79.38± 0.14	90.26± 0.05	84.98± 0.71	79.43± 2.12	79.76
BERT	81.55± 1.03	59.39± 0.40	77.27± 0.03	73.37± 3.01	**86.04± 2.18**	83.57± 0.58	76.87
CoLES RNN	82.82± 0.28	62.42± 0.33	79.30± 0.08	87.44± 0.20	85.56± 1.14	83.97± 0.16	80.25
CoLES Transformer	78.92± 0.49	59.92± 0.30	78.40± 0.00	87.06± 0.38	82.03± 0.98	83.90± 0.65	78.37
HT-Transformer	**83.34± 0.42**	60.10± 0.39	**80.42± 0.12**	**92.00± 0.09**	84.65± 1.07	87.44± 0.09	**81.32**
Impr. over NTP Transf.	+2.42	+3.94	+1.79	+0.72	+1.26	+0.51	+1.77

Table 2: Pretrained models classification results.

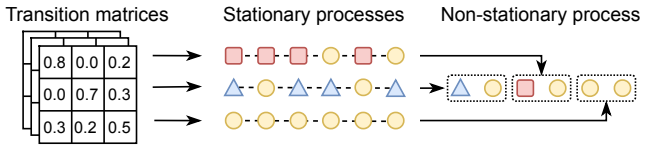


Figure 4: Markovian generative process for the toy dataset.

Method	Local (Last part)	Global (Num. parts)
Supervised	0.71	1.00
NTP Last	0.53	0.73
NTP Avg	0.40	0.88
CoLES	0.33	**0.94**
NTP HT	**0.55**	0.85

Table 3: Toy dataset classification accuracy.

The AgePred task differs from the others because it requires predicting a global property, specifically the client's age group, using historical event data. As discussed in the following section, history tokens are designed to capture recent and predictive information, which is beneficial for forecasting future events [Karpukhin and Savchenko, 2026] but less effective for tasks that require encoding global sequence properties. As a result, embeddings extracted from HT-Transformer underperform CoLES RNN.

5.2 Global Classification and Future-Oriented Tasks

While the concept of history tokens is broadly applicable, we observe certain limitations when combined with next token prediction during pretraining. The next token prediction objective encourages the model to focus on extracting recent, directly relevant information for forecasting upcoming events. In contrast, downstream tasks involving classification based on global or persistent properties, such as long-term user characteristics, may benefit more from contrastive pretraining or simpler aggregation strategies, such as averaging Transformer outputs across the sequence.

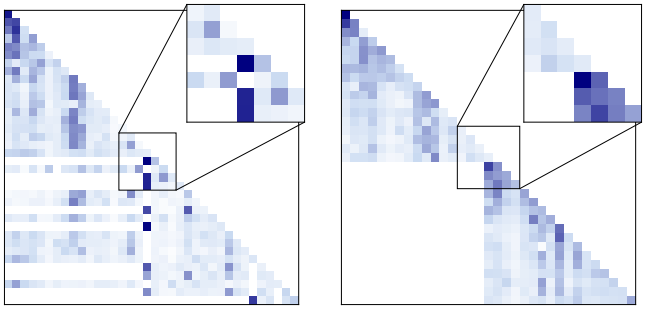
To investigate this effect, we conduct experiments on a synthetic dataset specifically designed to evaluate the suitability of different representation learning methods for local versus global tasks. In this dataset, we sample ten distinct transition matrices, each defining a Markov process by specifying the probability of transitioning from one label to another. We then construct nonstationary sequences by concatenating multiple segments, each generated using a different transition matrix, as illustrated in Figure 4. Note that we use regular timestamps here for simplicity.

We introduce two classification tasks for our synthetic dataset. The global classification task is to predict the total number of transition matrices used in a sequence, ranging from 1 to 5. This task requires the model to capture information across the entire sequence. In contrast, the local classification task involves identifying the index of the transition matrix used in the final segment, which depends only on the most recent data.

The results of classification experiments with Transformer-based models are reported in Table 3. The NTP Last and NTP Avg baselines correspond to Transformer models with different embedding extraction strategies. NTP Last uses the activations of the final token, whereas NTP Avg computes the average activations across all input tokens. The results show that the Last aggregation and HT-Transformer achieve superior accuracy on the local task, while average aggregation and contrastive pretraining yield the highest accuracy on global classification. These findings suggest that history tokens are particularly effective for tasks that depend on recent context, whereas contrastive pretraining and embedding averaging are more suitable for global classification tasks that require holistic sequence understanding.

5.3 Qualitative Analysis of History Tokens

To verify that history tokens are used to predict future tokens and to assess their relative contribution, we analyze attention patterns from the fifth layer of an HT-Transformer trained on MIMIC-III. Figure 5.a shows that the history token receives substantially higher attention weights than standard event tokens. This suggests that the history token serves as an accumulator of prefix information and actively influences the model's next-token predictions.



(a) HT-Transformer (b) NTP Transformer.

Figure 5: Attention matrices for (a) the HT-Transformer and (b) a standard Transformer under an equivalent prefix masking scheme.

Method	Churn	MIMIC	Taobao	AVG
Uniform placement	83.23	91.92	83.72	86.29
Last selection	82.92	91.90	83.78	86.20
Bias-End + Random	**83.34**	**92.00**	**84.65**	**86.66**

Table 4: Comparison of history token placement and selection strategies.

However, the increased attention to the history token could be a side effect of the masking scheme that restricts access to earlier tokens in the HT Transformer. To test this alternative, we analyze a standard NTP Transformer under an analogous masking regime, in which prefix tokens are manually excluded from attention. In this setting, shown in Figure 5.b, the last unmasked token does not receive comparably high attention. This contrast indicates that the prefix-accumulation behavior of the history token is not induced solely by masking, but instead emerges during pretraining.

5.4 Ablation Studies

In the introduction of HT-Transformer, we outlined alternative strategies for history token placement and selection. Table 4 compares these alternatives with the default HT-Transformer configuration, which employs the Bias-End placement strategy and Last selection of history tokens. The results indicate that the default configuration yields superior downstream performance on the considered datasets.

Additional ablation studies in Appendix D demonstrate that the proposed method is largely insensitive to the choice of p and f. Notably, the model trains successfully even with a single history token per sequence. Performance is best when the application probability p is between 0.25 and 0.75.

6 Limitations and Future Work

In this paper, we demonstrate the effectiveness of using history tokens for event sequence classification. However, several aspects of the method remain open for further exploration. First, as shown in our ablation studies, the placement of history tokens affects downstream performance. We compared two strategies: uniform placement and the Bias-End approach. Future work could pursue a more detailed analysis of token positioning and sampling policies.

Second, our current implementation relies on the standard PyTorch multi-head attention module. This component may not be optimal for working with the custom attention masks required by the HT-Transformer. Future technical improvements could focus on optimizing the attention mechanism, particularly by exploiting the sparsity of the mask. Although HT-Transformer preserves the same asymptotic complexity as standard causal Transformers in the current implementation, the sparse masking structure could enable future optimized implementations with reduced memory and computational cost.

Our experiments demonstrate that HT-Transformer may be suboptimal for global classification tasks. Future research could explore the design of architectures that effectively address both global and local tasks within a unified framework. One promising direction is to combine embeddings extracted from the model using different algorithms, thereby leveraging their complementary strengths.

Overall, the proposed architecture offers a promising direction for highly efficient and accurate modeling of event sequences. Future work can continue to improve the method's predictive performance and computational scalability.

7 Conclusion

This paper introduces HT-Transformer, a novel architecture that enhances Transformer-based models for event sequence classification by explicitly accumulating historical information via learnable history tokens. We identified the inherent limitations of standard Transformers in tasks requiring future event prediction, specifically the lack of a unified representation that effectively captures sequential context. To address this limitation, we proposed a simple yet effective mechanism where history tokens act as information bottlenecks during NTP pretraining, analogous to hidden states in recurrent neural networks.

Our method eliminates the need for auxiliary objectives, such as contrastive learning, and instead leverages sparse attention patterns to ensure efficient information aggregation. Extensive empirical evaluations across real-world financial, healthcare, and e-commerce datasets demonstrate that HT-Transformer consistently outperforms conventional Transformer baselines. Moreover, the proposed method achieves the highest accuracy on three datasets, surpassing all competing methods.

Overall, HT-Transformer represents a significant step toward bridging the performance gap between recurrent and Transformer-based models for future-oriented sequence modeling, combining high-quality embeddings with the long-context modeling capabilities of Transformer architectures.

Acknowledgments

The work of A. Savchenko was partially prepared within the framework of the Basic Research Program at the National Research University Higher School of Economics (HSE).

References

[Babaev *et al.*, 2022] Dmitrii Babaev, Nikita Ovsov, Ivan Kireev, Maria Ivanova, Gleb Gusev, Ivan Nazarov, and

Alexander Tuzhilin. ColLES: Contrastive learning for event sequences with self-supervision. In *Proceedings of the 2022 International Conference on Management of Data*, pages 1190–1199, 2022.

[BehnamGhader *et al.*, 2024] Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. LLM2Vec: Large language models are secretly powerful text encoders. In *First Conference on Language Modeling*, 2024.

[Beltagy *et al.*, 2020] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

[Bulatov *et al.*, 2022] Aydar Bulatov, Yury Kuratov, and Mikhail Burtsev. Recurrent memory transformer. *Advances in Neural Information Processing Systems*, 35:11079–11091, 2022.

[Cho *et al.*, 2014] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, 2014.

[Chopra *et al.*, 2005] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 539–546. IEEE, 2005.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

[Gorishniy *et al.*, 2021] Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. *Advances in neural information processing systems*, 34:18932–18943, 2021.

[Johnson *et al.*, 2016] Alistair EW Johnson, Tom J Pollard, Lu Shen, Li-wei H Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.

[Karpukhin and Savchenko, 2026] Ivan Karpukhin and Andrey Savchenko. Detecting the future: All-at-once event sequence forecasting with horizon matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 22536–22544, 2026.

[Ke *et al.*, 2017] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

[Kingma and Ba, 2015] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *The Twelfth International Conference on Learning Representations*, 2015.

[Lan *et al.*, 2020] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*, 2020.

[Liu *et al.*, 2019] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[McDermott *et al.*, 2023] Matthew McDermott, Bret Nestor, Peniel Argaw, and Isaac S Kohane. Event stream gpt: a data pre-processing and modeling library for generative, pre-trained transformers over continuous-time sequences of complex events. *Advances in Neural Information Processing Systems*, 36:24322–24334, 2023.

[Muennighoff *et al.*, 2023] Niklas Muennighoff, Nouamane Tazi, Loic Magne, and Nils Reimers. Mteb: Massive text embedding benchmark. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 2014–2037, 2023.

[Osin *et al.*, 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*, pages 5730–5741, 2025.

[Padhi *et al.*, 2021] 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)*, pages 3565–3569. IEEE, 2021.

[Radford *et al.*, 2018] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[Rizoiu *et al.*, 2017] Marian-Andrei Rizoiu, Young Lee, Swapnil Mishra, and Lexing Xie. Hawkes processes for events in social media. In *Frontiers of multimedia research*, pages 191–218. 2017.

[Robinson *et al.*, 2021] Joshua Robinson, Li Sun, Ke Yu, Kayhan Batmanghelich, Stefanie Jegelka, and Suvrit Sra. Can contrastive learning avoid shortcut solutions? *Advances in neural information processing systems*, 34:4974–4986, 2021.

[Sakhno *et al.*, 2025] Artem Sakhno, Ivan Kireev, Dmitrii Babaev, Maxim Savchenko, Gleb Gusev, and Andrey Savchenko. PyTorch-Lifestream: Learning embeddings on discrete event sequences. In James Kwok, editor, *Proceedings of the Thirty-Fourth International Joint Confer-*

ence on Artificial Intelligence, IJCAI-25, pages 11104–11108. International Joint Conferences on Artificial Intelligence Organization, 8 2025. Demo Track.

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

[Snoek *et al.*, 2012] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25, 2012.

[Stankevičius and Lukoševičius, 2024] Lukas Stankevičius and Mantas Lukoševičius. Extracting sentence embeddings from pretrained transformer models. *Applied Sciences*, 14(19):8887, 2024.

[Synerise, 2025] Synerise. Recsys challenge 2025. https://recsys.synerise.com/, 2025. Accessed: 2025-07-25.

[Vaswani *et al.*, 2017] 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.

[Wang *et al.*, 2019] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2019.

[Xue *et al.*, 2024] Siqiao Xue, Xiaoming Shi, Zhixuan Chu, Yan Wang, Hongyan Hao, Fan Zhou, Caigao JIANG, Chen Pan, James Y Zhang, Qingsong Wen, et al. Easytpp: Towards open benchmarking temporal point processes. In *The Twelfth International Conference on Learning Representations*, 2024.

[Yang *et al.*, 2022] Chenghao Yang, Hongyuan Mei, and Jason Eisner. Transformer embeddings of irregularly spaced events and their participants. In *Proceedings of the Tenth International Conference on Learning Representations (ICLR)*, 2022.

[Yenduri *et al.*, 2024] Gokul Yenduri, Muthukumaran Ramalingam, G Chemmalar Selvi, Y Supriya, Gautam Srivastava, Praveen Kumar Reddy Maddikunta, G Deepti Raj, Rutvij H Jhaveri, B Prabadevi, Weizheng Wang, et al. Gpt (generative pre-trained transformer)—a comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions. *IEEE access*, 12:54608–54649, 2024.

[Zhang *et al.*, 2025] Caojin Zhang, Qiang Zhang, Ke Li, Sai Vidyaranya Nuthalapati, Benyu Zhang, Jason Liu, Serena Li, Lizhu Zhang, and Xiangjun Fan. Gem: Empowering llm for both embedding generation and language understanding. *arXiv preprint arXiv:2506.04344*, 2025.

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