NeurIPS 2023

NEURAL INFORMATION PROCESSING SYSTEMS

Prompt-augmented Temporal Point Process for Streaming Event Sequence

Siqiao Xue*, Yan Wang*, Zhixuan Chu, Xiaoming Shi, Caigao Jiang, Hongyan Hao, Gangwei Jiang, Xiaoyun Feng, James Y Zhang, Jun Zhou

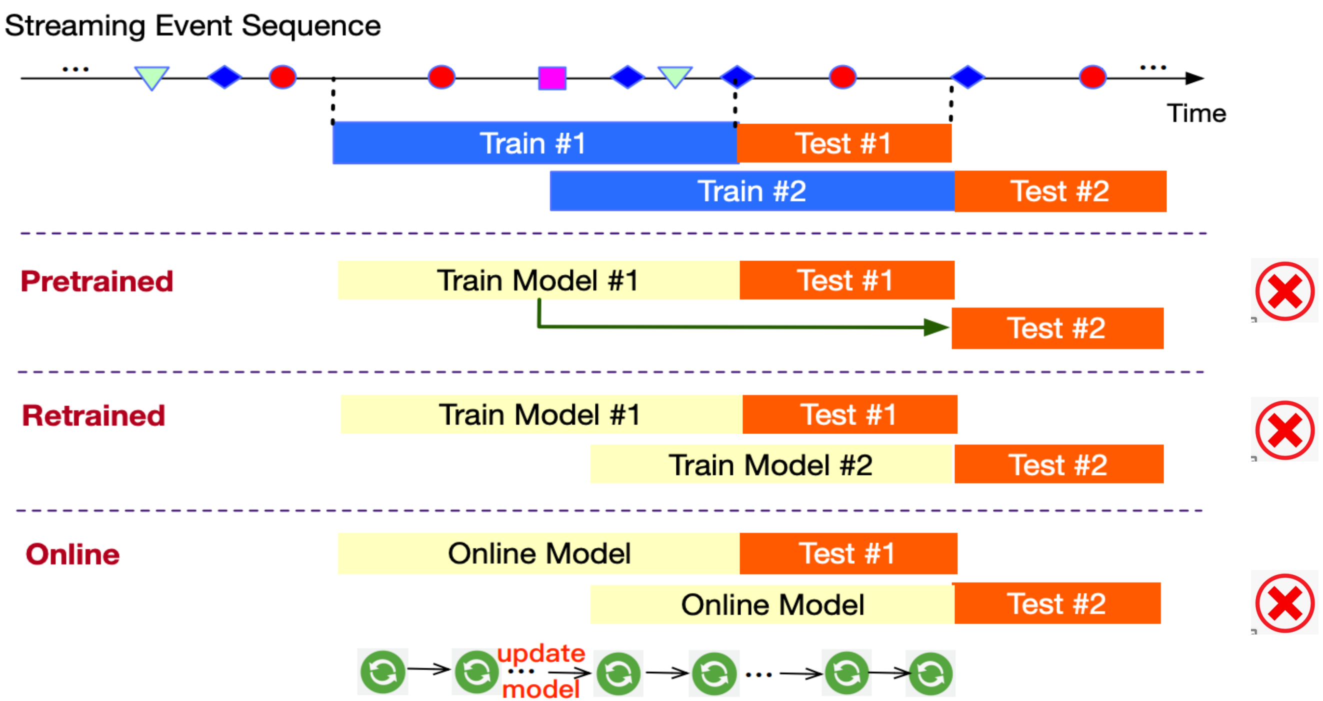
Ant Group

2023.10

ANT GROUP

Our Problem

Event data typically comes in ***streams,*** how to learn event streams continuously?

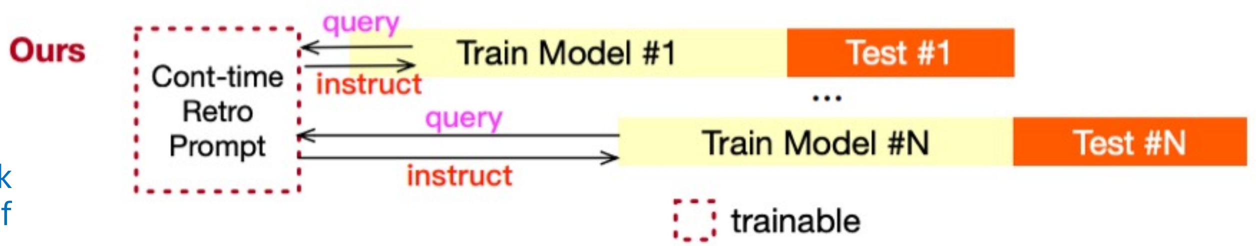


Failed to handle the data with distribution shift

Exposed to catastrophic forgetting

Hard to monitor, also exposed to catastrophic forgetting

Our Key Idea: Using Prompt Pool to Instruct the Learning



Define the task by a window of sequence

Construct a pool of prompts to encode knowledge of tasks

Using a retrieval mechanism to dynamically lookup a subset of task-relevant prompts

Prompts are jointly optimized with TPP base model

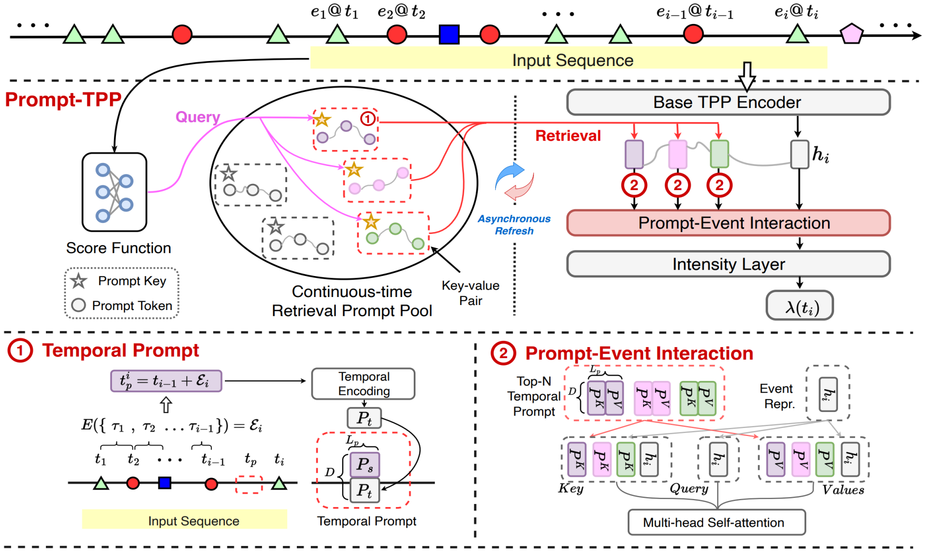
Our Model: Prompt-augmented Temporal Point Process

$$\boldsymbol{P} = [\boldsymbol{P}_1, ..., \boldsymbol{P}_M] \quad \boldsymbol{P}_i \in \mathbb{R}^{L_p \times D}$$

From Prompt to Prompt Pool

$$\mathrm{K}_{top-N} = \underset{\{r_j\}_{j=1}^{N}}{\operatorname{argmin}} \sum_{i=1}^{N} \varphi(\boldsymbol{h}_i, \boldsymbol{k}_{r_j})$$

Retrieval Mechanism



$P = [P_s; P_t]$ encodes the structural and temporal knowledge of the event sequence.

Prompt-Event Interaction combines retrieval prompts with the encoded event states.

Model Training: Joint Optimization with Prompts and TPP

$$\min_{\boldsymbol{P}, \phi_{enc}, \phi_{dec}, \mathcal{K}} \mathcal{L}_{nll}(\boldsymbol{P}, f_{\phi_{enc}}, f_{\phi_{dec}}) + \alpha \sum_{i} \sum_{\mathrm{K}_{top-N}} \varphi(f_{\phi_{enc}}(e_i @ t_i), \boldsymbol{k}_{r_j}),$$

Negative loglikelihood of event sequence

a surrogate loss to pull selected keys closer to corresponding query in the retrieval process

Model Inference: Thinning Sampling with retrieved prompts

Algorithm 2 PromptTPP at test time of the $\mathcal{T}$-th task.

Input: An event sequence $s_{[0,T]} = \{e_i@t_i\}_{i=1}^{I}$. Trained base model with a encoder $f_{\phi_{enc}}$ and a decoder $f_{\phi_{dec}}$; trained CtRetroPromptPool $(\mathcal{K}, \mathcal{V}) = \{(\boldsymbol{k}_i, \boldsymbol{P}_i)\}_{i=1}^{M}$ and the score function φ.

Output: Sampled next event $\widehat{e}_{I+1}@\widehat{t}_{I+1}$.

procedure DrawNextEvent($s_{[0,T]}, f_{\phi_{enc}}, f_{\phi_{dec}}$)
    $t_0 \leftarrow T$; $\mathcal{H} \leftarrow s_{[0,T]}$
    ▷ *Compute sampling intensity*
    $\{\lambda_e(t_j \mid \mathcal{H})\}_{j=1}^{N} \leftarrow$ SampleIntensity($s_{[0,T]}, f_{\phi_{enc}}, f_{\phi_{dec}}, \{(\boldsymbol{k}_i, \boldsymbol{P}_i)\}_{i=1}^{M}$) for all $t_j \in (t_0, \infty)$
    ▷ *Compute the upper bound λ^*.*
    ▷ *Technical details can be found in Mei & Eisner (2017)*
    find upper bound $\lambda^* \geq \sum_{e=1}^{E} \lambda_e(t_j \mid \mathcal{H})$ for all $t_j \in (t_0, \infty)$
    repeat
        draw $\Delta \sim \mathrm{Exp}(\lambda^*)$; $t_0 \mathrel{+}= \Delta$ ▷ *time of next proposed event $\widehat{t}_{I+1}$*
        $u \sim \mathrm{Unif}(0, 1)$
    until $u\lambda^* \leq \sum_{e=1}^{E} \lambda_e(t_0 \mid \mathcal{H})$
    draw $\widehat{e}_{I+1} \in \{1, \ldots, E\}$ where probability of e is $\propto \lambda_e(t_0 \mid \mathcal{H})$
    return $\widehat{e}_{I+1}@\widehat{t}_{I+1}$
procedure SampleIntensity($s_{[0,T]}, f_{\phi_{enc}}, f_{\phi_{dec}}, \{(\boldsymbol{k}_i, \boldsymbol{P}_i)\}_{i=1}^{M}$)
    Assume the last event in $s_{[0,T]}$ is $e@t$
    Generate a list of sample times $\{t_j\}_{j=1}^{N}, t_j \geq T$.
    Compute the intensity at sample times $\lambda_e t_j \leftarrow$ CalcIntensity($s_{[0,t]}, e@t, f_{\phi_{enc}}, f_{\phi_{dec}}, \{(\boldsymbol{k}_i, \boldsymbol{P}_i)\}_{i=1}^{M}$)
    return $\{\lambda_e(t_j \mid \mathcal{H})\}_{j=1}^{N}$

Take retro prompts as input into the calculation of the intensities

Please come to our **poster** for

Model details !

Training details !

Work well ? Very well !

Please download our paper at