TPP-SD: Accelerating Transformer Point Process Sampling with Speculative Decoding

Shukai Gong[1*], Yiyang Fu[2*], Fengyuan Ran[3*], Quyu Kong[4], Feng Zhou[1†]

[1]Center for Applied Statistics and School of Statistics, Renmin University of China
[2]School of Information, Renmin University of China
[3]School of Cyber Science and Engineering, Wuhan University
[4]Alibaba Group

*Equal contribution.
†Corresponding author.

Background

Sampling from a learned TPP is essential for synthesizing event sequence data, discovering complex process dynamics, etc.
However, most current Transformer TPPs' researches pay limited attention to improving sampling efficiency.

- **Thinning algorithm:** A rejection-sampling-based method whose single forward pass costs $O(N^2)$ and may still fail to produce any event.
- **Autoregressive sampling:** Intrinsically non-parallelizable. Demands $O(N^2)$ complexity per forward pass.

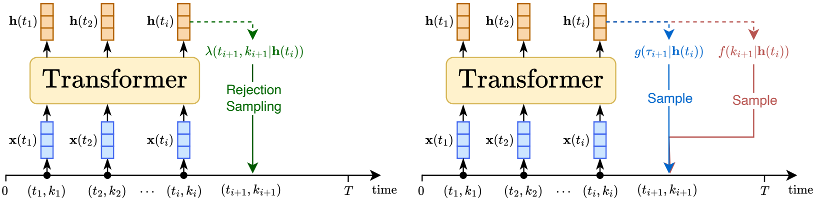


Figure 1: Thinning Algorithm (Left) and Autoregressive Sampling (Right)

Background: Key observation

- Speculative decoding (SD) methods for accelerating LLMs can significantly speed up autoregressive sampling without affecting model performance.

$$(\widehat{\text{token}}_{m+1}, q_{m+1}), \cdots, (\widehat{\text{token}}_{m+\gamma}, q_{m+\gamma}) \sim \text{LLM}_{\text{draft}}(\text{token}_{1:m})$$

$$p_{m+1}, \cdots, p_{m+\gamma} \sim \text{LLM}_{\text{target}}(\widehat{\text{token}}_{m:m+\gamma}; \text{token}_{1:m})$$

$$\text{Accept } \widehat{\text{token}}_{m+i} \text{ if } \epsilon_i < \frac{p_{m+i}(\widehat{\text{token}}_{m+i})}{q_{m+i}(\widehat{\text{token}}_{m+i})} \quad (i = 1, \cdots, \gamma) \text{ Until first rejection.}$$

- Thinning algorithm for TPP

$$\tilde{t}_{i+1} \sim \text{PoiP}(\mathcal{H}_{t_i}).$$

$$\lambda^*(\tilde{t}_{i+1}) = \text{OriP}(\tilde{t}_{i+1}; \mathcal{H}_{t_i}).$$

$$\text{Accept } \tilde{t}_{i+1} \text{ if } \epsilon < \frac{\lambda^*(\tilde{t}_{i+1})}{\bar{\lambda}}, \quad \epsilon \sim \text{Uniform}[0, 1].$$

is highly similar to SD for LLMs $\Rightarrow$ use SD for acceleration.

1. Background
2. Methodology
3. Experiments
4. Conclusion
5. Reference

CDF-based Transformer TPPs: Encoder

Encoder: For the observed event sequence $S = \{(t_i, k_i)\}_{i=1}^{N}$

- The timestamp t_i is encoded as a vector $\mathbf{z}(t_i) \in \mathbb{R}^D$
  $\Rightarrow \mathbf{Z} = [\mathbf{z}_1, \ldots, \mathbf{z}_N]^\top \in \mathbb{R}^{N\times D}$
- The event type k_i is transformed into a one-hot vector $\mathbf{k}_i \in \mathbb{R}^K$
  $\Rightarrow \mathbf{K} = [\mathbf{k}_1, \ldots, \mathbf{k}_N]^\top \in \mathbb{R}^{N\times K}$
- Aggregate historical information:

$$\mathbf{H} = T_\theta(\mathbf{X}) = T_\theta(f(\mathbf{KW}, \mathbf{Z})) \in \mathbb{R}^{N\times D}.$$

where the embedding matrix $\mathbf{W} \in \mathbb{R}^{K\times D}$, $\mathbf{h}^\top(t_i) = \mathbf{H}(i, :) \in \mathbb{R}^D$ is the historical information up to the event (t_i, k_i). T_θ can be any Transformer Backbone.

CDF-based Transformer TPPs: Mixture-of-Log-Normal Decoder

Decoder: Based on the historical information vector $\mathbf{h}(t_i)$, we parameterize the conditional distributions of the time interval $\tau_{i+1} = t_{i+1} - t_i$ and event type k_{i+1} as follows:

$$g_\theta(\tau_{i+1}|\mathbf{h}(t_i)) = \sum_{m=1}^{M} w_{im} \frac{1}{\tau\sqrt{2\pi}\sigma_{im}} \exp\left(-\frac{(\log \tau_{i+1} - \mu_{im})^2}{2\sigma_{im}^2}\right),$$

$$f_\theta(k_{i+1}|\mathbf{h}(t_i)) = \mathrm{softmax}\left(\mathbf{V}_\mathbf{k}^{(2)} \tanh(\mathbf{V}_\mathbf{k}^{(1)}\mathbf{h}(t_i) + \mathbf{b}_\mathbf{k}^{(1)}) + \mathbf{b}_\mathbf{k}^{(2)}\right).$$

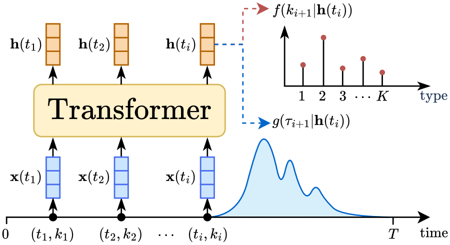


Figure 2: CDF-based Transformer TPP

TPP-SD Algorithm

Event Sampling: Use a parameter-efficient Draft model M_D to approximate the Target model M_T.

Drafting. Autoregressively sample γ candidate events $\{(\hat{t}_{i+1}, \hat{k}_{i+1}), \dots, (\hat{t}_{i+\gamma}, \hat{k}_{i+\gamma})\}$ from M_D and record the $g_D(\hat{\tau}_{i+l}|\cdot)$ and $f_D(\hat{k}_{i+l}|\cdot)$ for all candidate events.

Verification. Run M_T in parallel to compute $g_T(\hat{\tau}_{i+l}|\cdot)$ and $f_T(\hat{k}_{i+l}|\cdot)$ for all candidate events. Compute the acceptance rates for all candidate events:

$$\frac{g_T(\hat{\tau}_{i+l}|\cdot)}{g_D(\hat{\tau}_{i+l}|\cdot)} \quad \text{and} \quad \frac{f_T(\hat{k}_{i+l}|\cdot)}{f_D(\hat{k}_{i+l}|\cdot)}$$

If $\epsilon_\tau < \frac{g_T(\hat{\tau}_{i+l}|\cdot)}{g_D(\hat{\tau}_{i+l}|\cdot)}$ and $\epsilon_k < \frac{f_T(\hat{k}_{i+l}|\cdot)}{f_D(\hat{k}_{i+l}|\cdot)}$, where $\epsilon_\tau, \epsilon_k \sim \text{Uniform}[0, 1]$, then accept the candidate event $(\hat{\tau}_{i+l}, \hat{k}_{i+l})$

TPP-SD Algorithm

Once an event $(\hat{\tau}_{i+l}, \hat{k}_{i+l})$ is rejected, all subsequent candidate events will be automatically discarded, and the replacements $\hat{\tau}_{i+l}$ or $\hat{k}_{i+l}$ will be sampled from the following defined adjusted distributions:

$$g'(\tau_{i+l}|\cdot) = \text{norm}\left(\max(0, g_T(\tau_{i+l}|\cdot) - g_D(\tau_{i+l}|\cdot))\right),$$

$$f'(\hat{k}_{i+l}|\cdot) = \text{norm}\left(\max(0, f_T(\hat{k}_{i+l}|\cdot) - f_D(\hat{k}_{i+l}|\cdot))\right),$$

where norm$(\cdot)$ denotes the normalization operation. $g'(\tau_{i+l}|\cdot)$ is a continuous distribution, and normalization is more difficult because we need to compute the normalization constant:

$$R = \int \max(0, g_T(\tau_{i+l}|\cdot) - g_D(\tau_{i+l}|\cdot))d\tau_{i+l}.$$

This is the main difference between TPP-SD and LLM-SD, as continuous distributions are not involved in LLM applications. To solve this, we leverage **rejection sampling** to simulate sampling from $g'(\tau_{i+l}|\cdot)$.

For $\hat{\tau}_{i+l} \sim g_T(\tau_{i+l}|\cdot)$, we compute the acceptance threshold:

$$\alpha = \frac{\max(0, g_T(\hat{\tau}_{i+l}|\cdot) - g_D(\hat{\tau}_{i+l}|\cdot))}{g_T(\hat{\tau}_{i+l}|\cdot)}.$$

If $\epsilon < \alpha$, $\hat{\tau}_{i+l}$ is accepted where $\epsilon \sim \text{Uniform}(0,1)$. This rejection sampling process generates samples from the adjusted distribution $g'(\tau_{i+l}|\cdot)$.

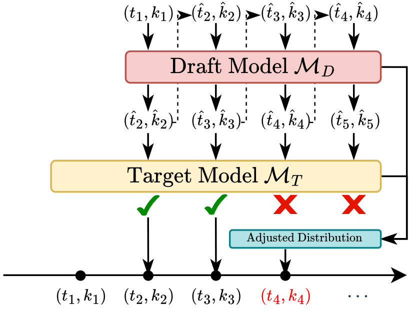


Figure 3: TPP-SD Sampling Process

Datasets

- Synthetic datasets: Inhomogeneous Poisson, Univariate Hawkes, and Multivariate Hawkes processes.
- Real-world datasets: Taobao, Amazon, Taxi, and StackOverflow.

Metrics

- **Likelihood Discrepancy (Synthetic and Real).** On synthetic data, compute $\Delta\mathcal{L}_{\text{ar}}^{\text{syn}} = |\mathcal{L}_{\text{gt}} - \mathcal{L}_{\text{ar}}|$ for AR sampling and $\Delta\mathcal{L}_{\text{sd}}^{\text{syn}} = |\mathcal{L}_{\text{gt}} - \mathcal{L}_{\text{sd}}|$ for TPP-SD. On real-world data, compute $\Delta\mathcal{L}^{\text{real}} = |\mathcal{L}_{\text{ar}} - \mathcal{L}_{\text{sd}}|$
- **Kolmogorov-Smirnov Statistic (Synthetic Only).** First convert event times $\{t_i\}_{i=1}^{n}$ into $\{z_i\}_{i=1}^{n-1}$ using the ground-truth CIF $\lambda^*(t)$, where $z_i = \int_{t_{i-1}}^{t_i} \lambda^*(\tau)\, d\tau$. Then compute the KS statistic D_{KS} between the ECDF of $\{z_i\}$ and the CDF of Exponential(1).
- **Wasserstein Distance (Real Only).** Using the first M events as history, perform N independent repetitions of sampling $(M+1)$-th event, yielding $\{(t_i^{\text{AR}}, k_i^{\text{AR}})\}_{i=1}^{N}$ from AR sampling and $\{(t_i^{\text{SD}}, k_i^{\text{SD}})\}_{i=1}^{N}$ from TPP-SD. Then compute the 1-Wasserstein distance D_{WS}^{t} between the ECDF of $\{t_i^{\text{AR}}\}_{i=1}^{N}$ and $\{t_i^{\text{SD}}\}_{i=1}^{N}$, and the earth mover's distance D_{WS}^{k} between the ECDF of $\{k_i^{\text{AR}}\}_{i=1}^{N}$ and $\{k_i^{\text{SD}}\}_{i=1}^{N}$.
- **Speedup Ratio (Synthetic and Real).** $S_{\text{AR/SD}} = \frac{T_{\text{AR}}}{T_{\text{SD}}}$, where T_{AR} and T_{SD} denote the execution wall times of AR sampling and TPP-SD, respectively.

Experiments Results on Synthetic Datasets

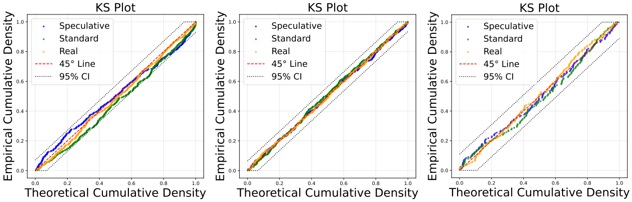
We experiment on three synthetic datasets across three Transformer Backbones: THP, SAHP, and AttNHP.

- The point distribution sampled by TPP-SD closely matches that of autoregressive sampling $\Rightarrow$ preserves sampling quality
- Achieves a speed-up ratio of 2 to $6\times$ $\Rightarrow$ greatly improves sampling efficiency
- Acceleration: AttNHP$>$THP$>$SAHP, Runtime: SAHP$<$THP$<$AttNHP.

Dataset		Poisson			Hawkes			Multi-Hawkes		
Encoder Type		THP	SAHP	AttNHP	THP	SAHP	AttNHP	THP	SAHP	AttNHP
$\Delta\mathcal{L}^{\text{syn}}$ ($\downarrow$)	AR Sampling	0.542	**0.012**	**1.879**	0.753	0.884	**0.220**	**0.022**	0.146	0.334
	TPP-SD	**0.349**	0.204	1.952	**0.276**	**0.630**	0.722	0.321	**0.070**	**0.199**
D_{KS} ($\downarrow$)	AR Sampling	0.038	**0.033**	0.076	0.044	0.031	0.029	0.069	**0.055**	0.065
	TPP-SD	**0.036**	0.050	**0.068**	**0.043**	**0.028**	**0.027**	**0.053**	0.080	**0.045**
Wall-time T ($\downarrow$)	AR Sampling	3.477	2.680	12.103	5.147	2.747	20.503	4.007	2.490	12.403
	TPP-SD	**1.647**	**2.077**	**4.063**	**2.547**	**1.863**	**3.567**	**1.893**	**1.647**	**2.770**
Speedup Ratio $S_{\text{AR/SD}}$ ($\uparrow$)		2.110	1.290	2.967	2.113	1.513	5.743	2.117	1.277	4.467

Table 1: Performance of TPP-SD with draft length $\gamma = 10$ against AR sampling across synthetic datasets and Transformer encoders.

Experiments Results on Synthetic Datasets



(a) Poisson (b) Hawkes (c) Multi-Hawkes

Figure 4: KS plots for (a) Poisson, (b) Hawkes, and (c) Multi-Hawkes datasets. We use AttNHP as encoder, and blue, green, and orange points represent samples from TPP-SD ($\gamma = 10$), AR sampling, and ground truth, respectively. Black dotted lines show 95% KS confidence bands.

Experiments Results on Real-world Datasets

We experiment on four real-world datasets across three Transformer Backbones: THP, SAHP, and AttNHP.

- The speedup inversely correlates with event type cardinality. A larger number of event types increases the probability of divergence between the draft and target models, leading to more rejections during SD.

Dataset		Taobao			Amazon			Taxi			StackOverflow		
Encoder Type		THP	SAHP	AttNHP	THP	SAHP	AttNHP	THP	SAHP	AttNHP	THP	SAHP	AttNHP
$\Delta\mathcal{L}^{\text{real}}$ (↓)	AR Sampling	0.446	**0.148**	**0.629**	**0.056**	0.099	**0.118**	1.4411	0.563	0.859	0.587	**0.340**	0.985
	TPP-SD	**0.033**	0.746	0.860	0.129	**0.035**	0.197	**0.065**	**0.093**	**0.506**	**0.231**	0.602	**0.020**
D^t_{WS} (↓)	AR Sampling	0.236	**0.328**	0.187	0.189	**0.019**	0.975	0.201	0.236	**0.249**	0.470	**0.378**	0.677
	TPP-SD	**0.076**	0.493	**0.116**	**0.078**	0.146	**0.464**	**0.082**	**0.036**	0.331	**0.391**	0.518	**0.614**
D^k_{WS} (↓)	AR Sampling	**0.267**	0.414	**0.193**	**0.184**	**0.459**	**0.252**	**0.055**	0.778	**0.094**	0.376	0.381	**0.218**
	TPP-SD	0.751	**0.368**	0.206	0.418	1.409	0.327	0.655	**0.744**	0.134	**0.375**	**0.199**	0.507
Wall-time T (↓)	AR Sampling	5.890	2.460	16.256	1.023	0.900	7.657	1.157	1.183	2.573	1.353	1.423	3.217
	TPP-SD	**3.460**	**1.643**	**5.180**	**0.290**	**0.317**	**1.353**	**0.453**	**0.347**	**0.650**	**0.700**	**0.663**	**0.783**
Speedup Ratio $S_{\text{AR/SD}}$ (↑)		1.597	1.553	3.183	3.550	2.847	5.849	2.553	3.637	4.310	1.930	2.153	4.290

Table 2: Performance of TPP-SD with draft length $\gamma = 10$ against AR sampling across real datasets and Transformer encoders.

Ablation Studies

We analyze the sensitivity of two critical hyperparameters, draft length γ and draft model size.

- A single-layer, single-head Draft model generating $\gamma = 5$–15 candidate events at each step $\Rightarrow$ maintains sampling quality while attaining the highest acceleration

Dataset	Encoder Type	Draft Model head	Draft Model layer	$\Delta\mathcal{L}$	Distance D_{KS} (↓)	Distance D^t_{WS} (↓)	Distance D^k_{WS} (↓)	α (↑)	Wall-time T_{AR} (↓)	Wall-time T_{SD} (↓)	Speedup Ratio $S_{\mathrm{AR/SD}}$ (↑)
Multi-Hawkes	AttNHP	1	1	0.098	**0.011**	-	-	0.600	12.403	**2.650**	**4.680**
		2	4	<u>0.139</u>	<u>0.009</u>	-	-	<u>0.710</u>	12.403	<u>3.003</u>	<u>4.130</u>
		4	6	**0.227**	0.004	-	-	**0.740**	12.403	5.176	2.676
Taobao	AttNHP	1	1	<u>0.276</u>	-	**0.080**	<u>0.197</u>	0.220	16.256	**5.727**	**2.838**
		2	4	**0.174**	-	<u>0.129</u>	0.200	<u>0.300</u>	16.256	<u>6.513</u>	<u>2.496</u>
		4	6	0.371	-	0.131	**0.190**	**0.35**	16.256	8.81	1.845

Table 3: Performance of TPP-SD with draft length $\gamma = 10$ under different size of draft model. The distance metrics D_{KS} is used for synthetic datasets, while D^t_{WS} and D^k_{WS} are used for real datasets.

Conclusion

- By identifying structural similarities between the thinning algorithm in TPPs and speculative decoding in LLMs, we develop an efficient framework, **TPP-SD**, that employs a lightweight draft model to propose candidate events for verification by the target model.
- TPP-SD significantly improves sampling efficiency by 2–6$\times$ while preserving distributional consistency with AR sampling.

Reference

[1] Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In International Conference on Machine Learning, pages 19274–19286. PMLR, 2023.

[2] Aristeidis Panos. Decomposable transformer point processes. In Annual Conference on Neural Information Processing Systems, 2024.

[3] Oleksandr Shchur, Marin Bilos, and Stephan Günnemann. Intensity-free learning of temporal point processes. In 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020. OpenReview.net, 2020.

[4] Zili Wang, Robert Zhang, Kun Ding, Qi Yang, Fei Li, and Shiming Xiang. Continuous speculative decoding for autoregressive image generation, 2024.