Inference for Mark-Censored Temporal Point Processes: Supplementary Material

Alex Boyd[1] **Yuxin Chang**[2] **Stephan Mandt**[1,2] **Padhraic Smyth**[1,2]

[1]Department of Statistics, University of California, Irvine
[2]Department of Computer Science, University of California, Irvine

A BIAS AND VARIANCE ANALYSIS OF CENSORED INTENSITY ESTIMATOR

In practice, the numerator and denominator of Eq. (4) are estimated with Monte-Carlo samples, resulting in the following approximation:

$$\underline{\lambda}_k^*(t) \approx \frac{\frac{1}{M}\sum_{i=1}^{M} \lambda_k(t\,|\,\mathcal{H}^{(i)}(t)) \exp\left(-\int_0^t \sum_{k'\in\mathbb{O}} \lambda_{k'}(s\,|\,\mathcal{H}^{(i)}(s))ds\right)}{\frac{1}{M'}\sum_{j=1}^{M'} \exp\left(-\int_0^t \sum_{k'\in\mathbb{O}} \lambda_{k'}(s\,|\,\mathcal{H}^{(j)}(s))ds\right)}$$

where $\mathcal{H}^{(i)}(t), \mathcal{H}^{(j)}(t) \overset{\text{iid}}{\sim} q$ for $i = 1, \dots, M$ and $j = 1, \dots, M'$. For simplicity, we typically set $M = M'$. This estimator is what is typically referred to as a ratio estimator, and while it is consistent unfortunately for finite samples it is biased.

To see in what way this is biased, we will recast this form into a more general format. Consider random variables $X, \{X_i\}_{i=1}^{M}, \{X'_j\}_{j=1}^{M'} \overset{\text{iid}}{\sim} p_X$ with support $\mathcal{X}$, and functions $f : \mathcal{X} \to \mathbb{R}^{+,0}$ and $g : \mathcal{X} \to [0, 1]$. We assume the mean and variance of both $f(X)g(X)$ (μ_{fg} and σ^2_{fg} respectively) and $g(X)$ (μ_g and σ^2_g) exist and $\mu_g \in (0, 1)$. This implies that the quantity of interest $\frac{\mu_{fg}}{\mu_g} := \frac{\mathbb{E}[f(X)g(X)]}{\mathbb{E}[g(X)]}$ is well defined. We now can investigate the bias of a finite sample ratio estimator through a second-order Taylor series expansion around $\frac{\mu_{fg}}{\mu_g}$:

$$\begin{aligned}
\mathbb{E}\left[\frac{\frac{1}{M}\sum_{i=1}^{M} f(X_i)g(X_i)}{\frac{1}{M'}\sum_{j=1}^{M'} g(X'_j)}\right] &\approx \frac{\mathbb{E}\left[\frac{1}{M}\sum_{i=1}^{M} f(X_i)g(X_i)\right]}{\mathbb{E}\left[\frac{1}{M'}\sum_{j=1}^{M'} g(X'_j)\right]} - \frac{\text{Cov}\left(\frac{1}{M}\sum_{i=1}^{M} f(X_i)g(X_i), \frac{1}{M'}\sum_{j=1}^{M'} g(X'_j)\right)}{\mathbb{E}\left[\frac{1}{M'}\sum_{j=1}^{M'} g(X'_j)\right]^2} \\
&\quad + \frac{\text{Var}\left(\frac{1}{M'}\sum_{j=1}^{M'} g(X'_j)\right)\mathbb{E}\left[\frac{1}{M}\sum_{i=1}^{M} f(X_i)g(X_i)\right]}{\mathbb{E}\left[\frac{1}{M'}\sum_{j=1}^{M'} g(X'_j)\right]^3} \\
&= \frac{\mu_{fg}}{\mu_g} - \frac{\sum_{i=1}^{M}\sum_{j=1}^{M'} \text{Cov}\left(f(X_i)g(X_i), g(X'_j)\right)}{MM'\mu_g^2} + \frac{\text{Var}\left(g(X)\right)\mu_{fg}}{M'\mu_g^3} \\
&= \frac{\mu_{fg}}{\mu_g} + \frac{\sigma_g^2\mu_{fg}}{M'\mu_g^3} \text{ since } X_i \perp X'_j
\end{aligned}$$

Likewise, the variance of the ratio estimator can also be approximated with a second-order Taylor series expansion around $\frac{\mu_{fg}}{\mu_g}$:

$$\text{Var}\left(\frac{\frac{1}{M}\sum_{i=1}^{M} f(X_i)g(X_i)}{\frac{1}{M'}\sum_{j=1}^{M'} g(X'_j)}\right) \approx \frac{\text{Var}\left(\frac{1}{M}\sum_{i=1}^{M} f(X_i)g(X_i)\right)}{\mathbb{E}\left[\frac{1}{M'}\sum_{j=1}^{M'} g(X'_j)\right]^2}$$

Accepted for the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023).

$$
\begin{aligned}
&- \frac{2\text{Cov}\left(\frac{1}{M}\sum_{i=1}^{M} f(X_i)g(X_i), \frac{1}{M'}\sum_{j=1}^{M'} g(X'_j)\right)\mathbb{E}\left[\frac{1}{M}\sum_{i=1}^{M} f(X_i)g(X_i)\right]}{\mathbb{E}\left[\frac{1}{M'}\sum_{j=1}^{M'} g(X'_j)\right]^3} \\
&+ \frac{\text{Var}\left(\frac{1}{M'}\sum_{j=1}^{M'} g(X'_j)\right)\mathbb{E}\left[\frac{1}{M}\sum_{i=1}^{M} f(X_i)g(X_i)\right]^2}{\mathbb{E}\left[\frac{1}{M'}\sum_{j=1}^{M'} g(X'_j)\right]^4} \\
&= \frac{\sigma_{fg}^2}{M\mu_g^2} - \frac{2\mu_{fg}\sum_{i=1}^{M}\sum_{j=1}^{M'}\text{Cov}\left(f(X_i)g(X_i), g(X'_j)\right)}{MM'\mu_g^3} + \frac{\sigma_g^2\mu_{fg}^2}{M'\mu_g^4} \\
&= \frac{\sigma_{fg}^2}{M\mu_g^2} + \frac{\sigma_g^2\mu_{fg}^2}{M'\mu_g^4} \text{ since } X_i \perp X'_j.
\end{aligned}
$$

It can be tempting to consider reusing samples for both the numerator and the denominator (i.e., $M = M'$ and $X_i = X'_i$ for $i = 1, \ldots, M$) as this would save in the amount of computations needed for computing the ratio estimate. This would result in the following expected value and variance of the estimator:

$$
\begin{aligned}
\mathbb{E}\left[\frac{\frac{1}{M}\sum_{i=1}^{M} f(X_i)g(X_i)}{\frac{1}{M}\sum_{j=1}^{M} g(X_j)}\right] &\approx \frac{\mu_{fg}}{\mu_g} - \frac{\sum_{i=1}^{M}\sum_{j=1}^{M}\text{Cov}\left(f(X_i)g(X_i), g(X_j)\right)}{M^2\mu_g^2} + \frac{\sigma_g^2\mu_{fg}}{M\mu_g^3} \\
&= \frac{\mu_{fg}}{\mu_g} - \frac{\text{Cov}\left(f(X)g(X), g(X)\right)}{M\mu_g^2} + \frac{\sigma_g^2\mu_{fg}}{M\mu_g^3} \\
\text{Var}\left(\frac{\frac{1}{M}\sum_{i=1}^{M} f(X_i)g(X_i)}{\frac{1}{M}\sum_{j=1}^{M} g(X_j)}\right) &\approx \frac{\sigma_{fg}^2}{M\mu_g^2} - \frac{2\mu_{fg}\sum_{i=1}^{M}\sum_{j=1}^{M}\text{Cov}\left(f(X_i)g(X_i), g(X_j)\right)}{M^2\mu_g^3} + \frac{\sigma_g^2\mu_{fg}^2}{M\mu_g^4} \\
&= \frac{\sigma_{fg}^2}{M\mu_g^2} - \frac{2\mu_{fg}\text{Cov}\left(f(X)g(X), g(X)\right)}{M\mu_g^3} + \frac{\sigma_g^2\mu_{fg}^2}{M\mu_g^4}.
\end{aligned}
$$

Either forms of the expected values of the estimators can be used to help us correct for the bias by simply moving all terms on the right that are not $\frac{\mu_{fg}}{\mu_g}$ to the left. Interestingly, we can see that there is potential for reusing samples to not only save on computation, but to also reduce the variance of the estimator. Should $\text{Cov}(f(X)g(X), g(X)) > 0$, which is often the case in practice, then the variance will be reduced.

B FURTHER EXPERIMENTAL DETAILS AND RESULTS

B.1 DATASETS

The following are more in depth descriptions on the different real-world datasets used in experiments. All sequences used for both training and inference are preprocessed to only allow sequences with at least 5 events and at most 200. Summary statistics can be found in Table 1.

Taobao The Taobao user behavior dataset [Zhu et al., 2018] was originally intended for recommendations during online shopping sessions, which includes four different behaviors: page viewing, purchasing, adding items to the chart, and adding items to a wishlist. We focus on modeling the page viewing of users as events, and let the item category be the associated event mark. Modeling this information has various marketing implications such as click through rate of recommending some types of items. Due to the large scale of the dataset, we use a subset of 2,000,000 events on 8 consecutive calendar days inclusive (November 25th, 2017 - December 2nd, 2017), as well as the most frequent 1,000 marks (item categories). All user sequences have the same time length of $T = 192$ hours.

Reddit The Reddit comments dataset [Baumgartner et al., 2020] contains records of comments made by different users on various posts listed in the social media site `reddit.com`. One month's worth of data (October 2018) was used to extract user sequences, and the mark vocabulary was defined as the top 1000 communities (subreddits) determined by marginal comment volume. The month was divided into multiple week-long sequences for each user, with event times in units of hours ($T = 178$ hours).

Table 1: Summary Statistics for the Four Real-World Datasets

			Mean	# Sequences		
Dataset	T	M	$\lvert\mathcal{H}\rvert$	Train	Valid	Test
Taobao	8 Days	1000	62.6	13.3K	1.8K	2.7K
Reddit	1 Week	1000	65.2	343K	15K	34K
MemeTracker	1 Week	5000	23.4	271K	9K	21K
Email	28 Days	808	31.1	6.9K	1.5K	1.5K

Table 2: Model Hyperparameters for Real-World Datasets

Hyperparameter	Taobao	Reddit	MemeTracker	Email
# Training Epochs	300	50	50	300
Mark Embedding Size	64	64	64	32
Recurrent Hidden State Size	128	128	128	64

MemeTracker The MemeTracker dataset [Leskovec et al., 2009] tracks to common phrases (memes) as they appear on various websites. We compile these records into sequences, each pertaining to a single meme with events defined as the time of mention and the website they appeared on as the mark. Only the mentions in the top 5000 websites by marginal volume were considered. Sequences were defined as one-week-long chunks spanning August 2008 to April 2009, and event times were measured in hours ($T = 178$ hours).

Email Lastly, the Email dataset [Paranjape et al., 2017] contains the email records for a research organization over the course of 803 days. Sequences were defined as the collection of incoming emails for a given user where each mark was the address of the original sender. These sequences were defined over four week intervals and event times were measured in days ($T = 28$ days). After preprocessing the sequences, we were left with 808 different unique addresses.

B.2 MODEL & TRAINING DETAILS

For each of the real-world datasets, a neural Hawkes process model [Mei and Eisner, 2017] was trained on fully observed sequences for a given dataset. Each model was trained using the Adam stochastic gradient optimization algorithm [Kingma and Ba, 2015] with default hyperparameters, a learning rate of 0.001, and a linear warm-up learning rate schedule over the first 1% of training iterations. Each iteration optimized the parameters against the average log-likelihood for a batch of 128 training sequences. Gradients were clipped to have a maximum norm of 10^4 for stability. All models were trained for a fixed amount of epochs; however, each one was confirmed to have converged based on average held-out validation log-likelihood.

Models possessed different hyperparameters depending on the dataset due to differences in the amount of data and total possible marks. Details can be found in Table 2.

B.3 NEXT EVENT PREDICTION

Alongside likelihood, we are also interested in making predictions for next events in the presence of censored data. The following section details the prediction experiments conducted for both synthetic and real-world settings.

Setup We follow the same settings for the next event prediction as Du et al. [2016], Mei and Eisner [2017] on both event time and event mark. The predicted time is chosen to be the expected time of the next event occurrence, which is defined as

$$\hat{\tau}_i = \mathbb{E}\left[\tau_i \mid \mathcal{H}[0, \tau_{i-1}]\right] = \int_{\tau_{i-1}}^{\infty} t\lambda^*(t) \exp\left(-\int_{\tau_{i-1}}^{t} \lambda^*(s)ds\right) dt.$$

We measure predictive performance for this with the mean absolute error between predicted and true next event time. Without the knowledge of the event time τ_i, the predicted distribution of the next event type is defined to be

$$p(\hat{\kappa}_i = k) = \int_{\tau_{i-1}}^{\infty} \lambda_k^*(t) \exp\left(- \int_{\tau_{i-1}}^{t} \lambda^*(s) ds \right) dt,$$

and is evaluated via top-10 accuracy (i.e., the proportion of predictions in which true mark κ_i appears in the set of top-10 highest probability predicted marks). Both predictions can be achieved by approximating integrals numerically, for both the censored and baseline methods.

Similar to the likelihood ratio experiments, we evaluated these methods on sequences that have been artificially censored. For the synthetic experiments, we evaluate 1000 sequences $\mathcal{H}(T)$ sampled from their respective models and then randomly choose a subset of unique marks that appear in each sequence to be censored $\mathbb{C}$, the proportion of which is determined for each value $\gamma \in \{0.2, 0.4, 0.6, 0.8\}$. For real-world experiments, the same is done except the sequences originate from held-out sets and γ is also allowed to be 0.5.

We condition each method on the occluded sequence $\mathcal{H}_{\mathbb{O}}[0, \tau_{\lfloor \frac{n}{2} \rfloor}]$ where $|\mathcal{H}(T)| = n$ and have each produce predictions for the next time $\hat{\tau}_{\lfloor \frac{n}{2} \rfloor + 1}$ and the next mark $\hat{\kappa}_{\lfloor \frac{n}{2} \rfloor + 1}$.

Synthetic Results Figure 6 reports the results evaluated on three parametric point process models with 20 distinct marks. When predicting next time to event, both versions of Hawkes processes achieve less error under our framework compared to the baseline. The performance gap between methods widen as more information is censored. However, the baseline outperforms our method for self-correcting models, which may be due to the fact that the occurrence of an event has an inhibiting effect on future events. This results in the baseline always overestimating the intensity as it lacks the censored events to correct it. For this model, this leads to always underestimating the next time to event which is favorable as this will be bounded between $\tau_{\lfloor \frac{n}{2} \rfloor}$ and $\tau_{\lfloor \frac{n}{2} \rfloor + 1}$. This can be seen as a systematic bias inherent to the specific model parameterization.

As for the prediction of the next event type, both self-correcting processes and Hawkes processes with dense interaction between events have similar performances as random guesses that will have an accuracy of around 0.5 for top-10 accuracy. This is expected for both models, as there is not much imposed correlation between events of different types due to how the models were instantiated. However, the Hawkes processes with block-diagonal interactions better model the structure in sequential events, where the prediction accuracy is much higher than 0.5, which in general decreases as more marks are censored. It is clear that our method is less sensitive to the amount of censored information and significantly outperforms random guesses, as long as the model is able to capture the underlying structured dynamics of the event sequences.

Real-World Results Real-world datasets naturally have more meaningful structures and larger vocabulary sets compared to synthetic experiments. We evaluate the results on all four datasets that have different numbers of marks ranging from 808 to 5000. The prediction of the next event time of our method is on par with the baseline, while we see consistent improvements in the next event prediction evaluated by top-10 accuracy. Furthermore, the accuracy in general, regardless of method, tends to decrease with more information being censored which is expected.

B.4 MODEL MISSPECIFICATION

Recall in the synthetic experiments that we evaluated the log-likelihood for the mark-censored model p_{Cen} and the baseline method p_{Base} on censored sequences that were originally sampled from the same model p used in both methods. Under this setting, for a given mark-censoring scheme $\mathbb{C}$ and $\mathbb{O}$ and sampled sequences $\mathcal{H}_{\mathbb{O}}(t) \sim p$, it is guaranteed that

$$\mathbb{E}_{p(\mathcal{H}_{\mathbb{O}}(t))} \left[p_{\text{Cen}}(\mathcal{H}_{\mathbb{O}}) \right] \geq \mathbb{E}_{p(\mathcal{H}_{\mathbb{O}}(t))} \left[p_{\text{Base}}(\mathcal{H}_{\mathbb{O}}) \right]$$

with the inequality being strict so long as $p(\mathcal{H}_{\mathbb{C}}(t) = \emptyset \,|\, \mathcal{H}_{\mathbb{O}}(t)) > 0$. This is due to the fact that the mark-censored model is simply a marginalized version of the original model, thus resulting in no model misspecification for this setup.

That being said, we no long have this guarantee once we start considering sequences that are drawn from a different distribution from the model we are using. This is inherently the same scenario that was evaluated in the real-world data experiments, as all of the sequences used there came from some other source p_{data} whereas the models p were learned to best approximate this distribution. Naturally, the closer p is to p_{data} (i.e., the less model misspecification there is) the more we can start to trust that the censored method will produce superior results to the baseline.

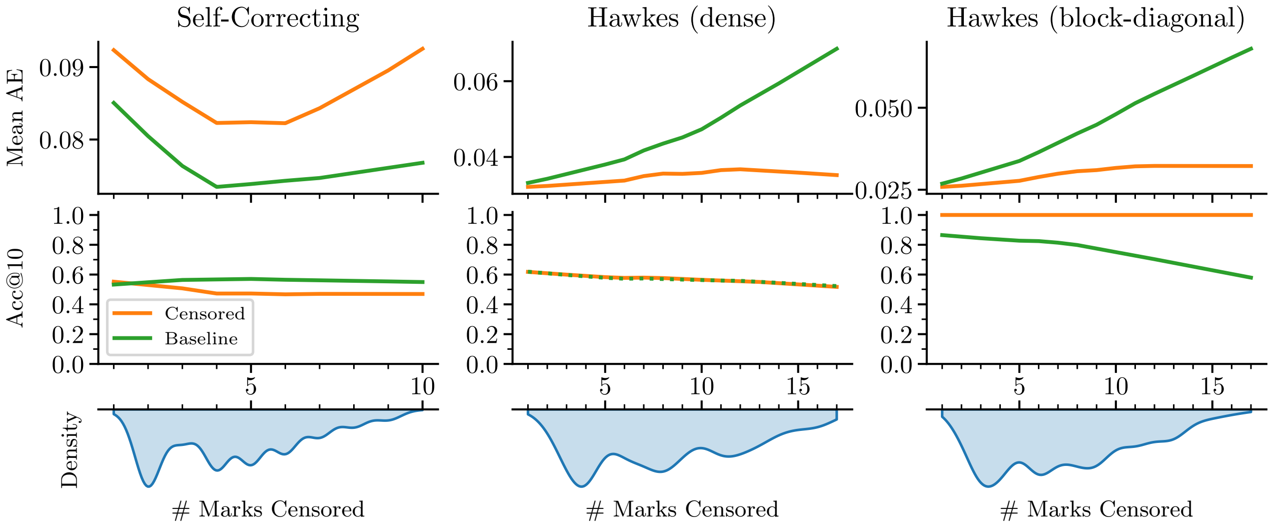

Figure 6: Next event prediction results for censored and baseline methods across the three different parametric MTPPs. Top plots indicate the mean absolute error in next time prediction, middle plots indicate top-10 accuracy in next mark prediction, and bottom plots show density of the number of marks censored across the sequences used for the experiments.

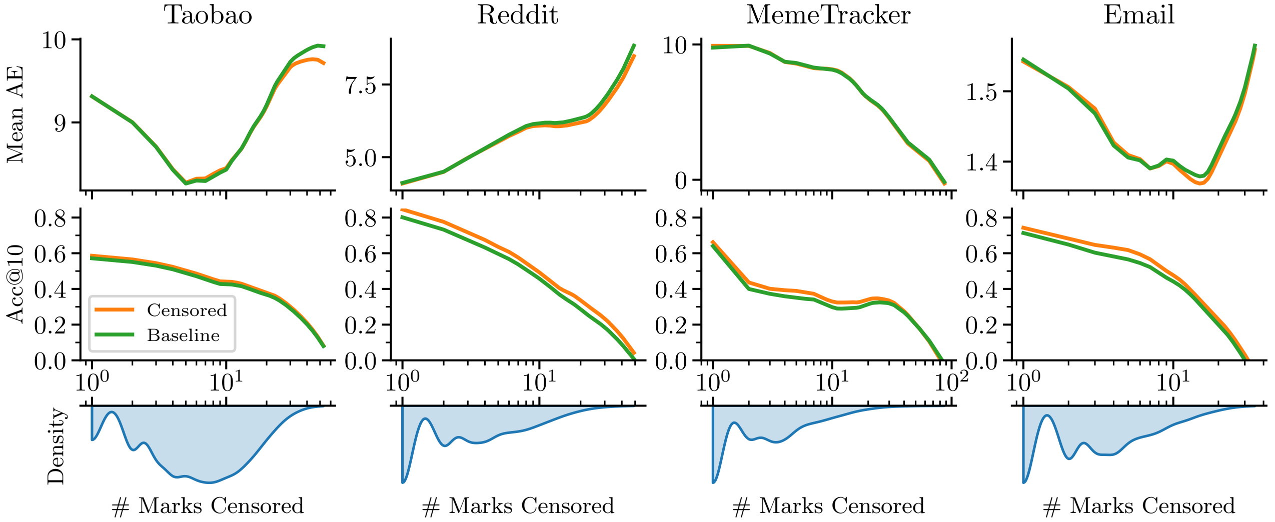

Figure 7: Same format as Fig. 6 except using held-out sequences from real-world datasets with respectively trained neural-based models.

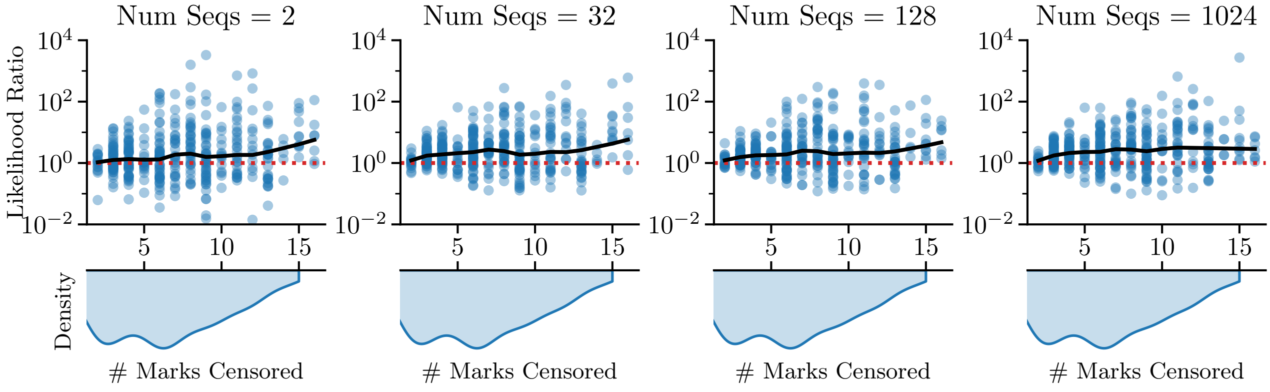

Figure 8: Distributions of likelihood ratios across number of marks censored for the duration of the sequences used for synthetic experiments. Integration points is fixed as 1024, with varying numbers of MC samples used for estimation. Values greater than 1 indicate higher likelihoods under the mark-censored model.

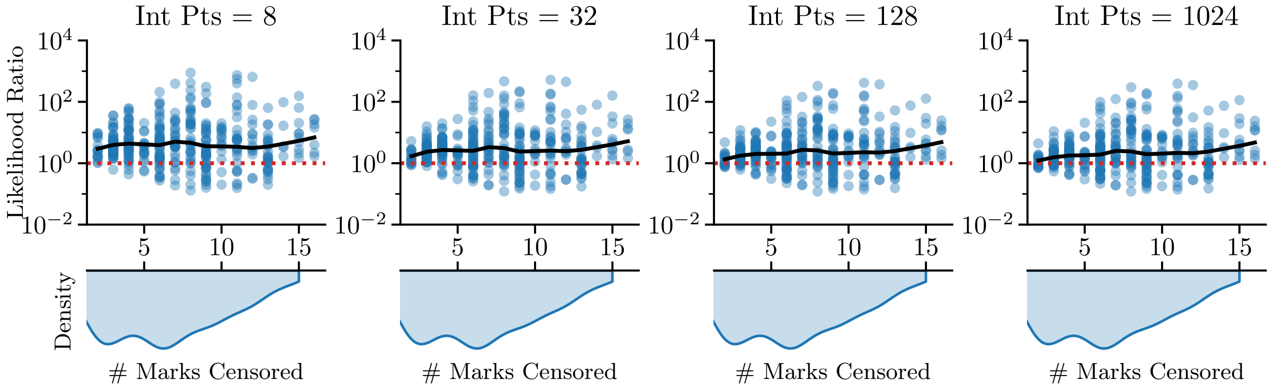

Figure 9: Same format as Fig. 8 except using 128 MC samples and different numbers of integration points for estimation.

B.5 SENSITIVITY ANALYSIS

We perform an ablation study for synthetic experiments using different numbers of samples and integration points. The parameters of the Hawkes process are drawn from the same distributions as described in Section 4.1, where we used 128 MC samples and 1024 integration points. Figure 8 shows the results of the same experiment but varies the number of Monte Carlo sampled sequences and keeps the number of integration points as 1024, while 9 shows the same results but varies the number of integration points while keeping the number of Monte Carlo samples fixed to 128. Aside from slight deviations on the lower end of the values tested (e.g., number of sampled sequences $= 2$ and number of integration points $= 8$), the results across the board are roughly consistent. This indicates that our method is fairly robust and does not necessitate prohibitive amounts of computing resources to employ.

That being said, we do recommend evaluating this on a case-by-case basis as each model and dataset are different. In critical applications, this concern can be taken care of by iteratively sampling sequences and monitoring the convergence of the resulting censored intensity function.

References

Jason Baumgartner, Savvas Zannettou, Brian Keegan, Megan Squire, and Jeremy Blackburn. The Pushshift Reddit dataset. In *Proceedings of the International AAAI Conference on Web and Social media*, volume 14, pages 830–839, 2020.

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*, pages 1555–1564, 2016.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015. URL `http://arxiv.org/abs/1412.6980`.

Jure Leskovec, Lars Backstrom, and Jon Kleinberg. Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 497–506, 2009.

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.

Ashwin Paranjape, Austin R Benson, and Jure Leskovec. Motifs in temporal networks. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 601–610, 2017.

Han Zhu, Xiang Li, Pengye Zhang, Guozheng Li, Jie He, Han Li, and Kun Gai. Learning tree-based deep model for recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1079–1088, 2018.