We thank all reviewers for the constructive comments.

To reviewer 1:

Q1: It is standard to compare log-likelihood. Examples include Xu et al., (2016, arXiv:1602.04511), Mei et al. (2017, arXiv:1612.09328) and almost all our cited papers on point process. This is because the prediction of next arrival is essentially predicting a distribution, and log-likelihood can give a better characterization of the predicted distribution compared to RMSE. Figure 1 shows a simple example. Generally speaking, RMSE only evaluates the mean, while likelihood evaluates mean and higher order moments (see consistency theorems in Wald (1949, The Annals of Mathematical Statistics)).

All baselines in our paper optimize log-likelihood. The comparison is fair. It is natural to first model event sequences as a probabilistic model, then estimate the model using MLE. Therefore, many existing methods adopt such a procedure.

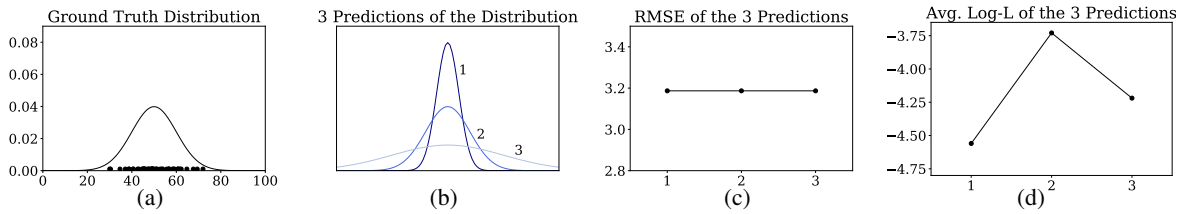

Figure 1: Simple illustration of why log-likelihood is a stronger metric than RMSE, where (a) shows the ground truth distribution, and (b) shows 3 predictions of (a). In (b), prediction 2 recovers the ground truth perfectly, while prediction 1 and 3 do not. (c) and (d) show the RMSE and log-likelihood estimated from 1000 realizations of the ground truth. RMSE cannot distinguish which prediction is better, while the log-likelihood for the best prediction is the largest.

Q2: We chose methods to be Hawkes process-based because we are targeting event sequences that have self-exciting property. For example, in the MathOverflow dataset, a user that just answered one question usually comments on several other answers. It is widely accepted that Hawkes process is suitable to model data with such property (Laub et al., 2015, arXiv:1507.02822, Rizoiu et al., 2017, arXiv:1708.06401).

RNN is not suitable for the short sequence data we are targeting, so we did not include it. There are two possible ways to incorporate RNN to perform the task:

1. The first one is to adopt an RNN (or LSTM, GRU) to directly fit $\tau^{(n+1)} = f(\tau^{(1)}, \cdots, \tau^{(n)})$. However, one critical drawback of this model is that it can only provide a point estimation, while the Hawkes process can predict a distribution. Such distribution is important in our case: (i) the next timestamp is a random variable in nature with a large variance; (ii) in real applications, the time interval that has larger probability density for next event may be of more interest than a single estimate. However, naive RNN models cannot give such information.

2. Another option is to adopt neural Hawkes process model in Mei et al. (2017, arXiv:1612.09328). Such a model uses RNN to parametrize the *evolving* intensity of the Hawkes process, which usually works well for *long* sequences. In our case, however, the short history cannot provide enough information to fit the complex evolution of the sequences. In our preliminary experiments, we observed that (i) the training is unstable, and (ii) the performance is not as good as the standard Hawkes process model (*MLE-Com*).

We will add clarification in the next version.

We want to remark that our HARMLESS framework can actually be complemented by RNN-based models. As we mentioned in the Discussion section (line 340), if longer sequences are targeted, HARMLESS can naturally extend to more flexible models by replacing the standard Hawkes process part to Mei et al. (2017, arXiv:1612.09328).

Q3: We use all other timestamps in training because HARMLESS builds different models for different subjects, and thus we need to evaluate each model individually. One should not use the model for another user to predict the future of this user. In addition, this is the common setting in unsupervised meta learning (Hsu et al., 2018, arXiv:1810.02334).

To reviewer 2: The biggest advantage of using MAML is its adaptivity (Grant et al., 2018, arXiv:1801.08930). We will add more explanation in the next version. We will also adjust the figures and add empirical guidelines for choosing meta-learning methods.

To reviewer 3: We will add more discussion regarding the performance of different models in the next version. In short, different models are suitable for different data. For example, Reptile is more suitable for larger datasets.

We will add discussions on the computation complexity in the next version. Roughly speaking, if the batch size is n, then the time complexity per iteration is $O(n^2)$.