We thank the reviewers for their time and effort.

We begin by clarifying the scope and novelty of our contributions. Broadly, our work is a general extension of RNN-based MTPP models (such as the RMTPP, but also the Neural Hawkes Process and others). More specifically: (1) our work is the first to address the problem of personalizing neural MTPP models; (2) we combine VAE and neural MTPP approaches in a non-trivial fashion (e.g., training with cyclical annealing); (3) we provide extensive experimental results on multiple real-world datasets that show consistent and significant performance improvements, and (4) we provide a detailed breakdown of where personalization helps in prediction (i.e., particularly at the beginning of sequences). (In our paper we removed a list of contributions from the work to save space, but will add this back in the revised version).

We appreciate all of your comments and critiques about our work's clarity (as mentioned by Reviewers 1, 2, and 4), citations and missing related works (Reviewers 1 and 3), and typos / terminology misuse (Reviewers 1 and 2). We will be sure to incorporate this feedback into the camera-ready version. Listed below are our specific responses to all other comments made. For brevity, many of them are paraphrased.

Reviewer 1 *How are the three different information sources (on line 73) a hierarchy?* We realized that our language here is not precise. We were trying to say that the data is organized first as users, each of which have multiple event sequences, with each sequence having a prefix and a future trajectory. We will refine these comments in the revised version to avoid possible confusion.

Are user features available? Great question! For some of the datasets, user features were available to some degree (e.g. a username, user-entered bio, etc.), but were not used for uniformity. In practice, user features would be a useful addition to our approach and should be straightforward to add, e.g., by embedding this information and concatenating with our user embeddings.

In Eq. 2, why concatenate z^u to the mark embedding and not the timing? Another great question! We wanted our proposed framework to be applicable regardless of base neural MTPP model. As such, all neural MTPPs represent marks via embeddings which would allow us to concatenate the user embedding to it without problems. On the other hand, different models incorporate the event timings differently where some embed it and others require using a scalar. As such, there was no way to feasibly incorporate the user embedding to the timings in a universal manner.

Dimensionality of z^u? Interpretation? z^u is a real-valued vector, and the dimensionality ranges from 32 to 64 depending on the dataset (see supplement for more information). This vector can be interpreted as the sequence and user-specific dynamics for a single history of events. $p(z^u)$ represent the various modes of dynamics for a given user u. For future work it would be interesting to investigate using z^u for downstream tasks, such as clustering users.

Clarify why a "single sample" is used in the loss term? We found using one (five) sample(s) for a Monte-Carlo estimate of the expected value in the loss term to be sufficient for training (testing). We did this for computational efficiency as each additional sample is tied to processing another event sequence.

Why is the predicted timing performance poor? We believe there may be a misunderstanding as the timing performance for the personalized model is not poor, but rather just not that different to the baseline model performance (see more on lines 255-259). As for why this is, we hypothesize that this could be because (i) the variation between users lies in the subset of marks that occur for them rather than the timing or (ii) the temporal information in the encoding steps is not being adequately captured which could be better enforced via regularization.

Reviewer 2 Please refer to our introductory statements as we believe this should hopefully address your concerns.

Reviewer 3 *Why is source identification meaningful / practical?* This is a practical problem in a number of applications, e.g., when trying to match clickstreams of non-logged-in users to known users, or for fraud detection in cybersecurity. While our experiments are not meant to replicate a real-world application, we nonetheless believe the experiments provide a useful way to evaluate and compare MTPP models.

Reviewer 4 *Report the reference sequence sizes?* The reference and target sequences have the same time window and follow the same distribution of number of events (see "Mean $|\mathcal{H}|$" column in Table 1).

Performance as a function of reference sequence size? This is an interesting point. As of now, the model expects reference sequences that span similar lengths of time that the target sequence does. This setup reflects how the framework would typically be used in practice; however, restricting the reference information would be a way to test generalization to longer sequences. Do note that we do analyze performance as a function of the *target* sequence size.

"The two baselines are not strong enough" We respectfully disagree. We believe that in conjunction with our strong and extensive experimental findings that these two powerful baselines are sufficient at establishing a reference point to observe how incorporating latent user embeddings improves modeling performance.

"Better to compare to other methods that can train user embeddings or use randomly generated user embeddings." To the former point, expanding on Lines 184-185, we are especially interested in scenarios with brand new users. That means that conventional means of having a set of explicitly learned user embeddings (as opposed to amortized embeddings) would not be applicable as there would be no associated embeddings at test time. For the latter suggestion, we are not quite sure how this implementation would be better from one without user embeddings at all as at best the model would learn to ignore the randomized embeddings.