We thank reviewers for the relevant comments. We first address general questions and then give brief individual answers.

On the necessity of GRU-Bayes and link to filtering methods. GRU-ODE and GRU-Bayes have complementary roles and should be used together. GRU-ODE integrates the dynamics of the hidden process $h(t)$ in time. It thus provides the future estimated distribution of the observations. It computes our **belief** about the unobserved future time series observations. Those projected distributions vary smoothly as they are driven by an ODE. But, as soon as any new observation is available, our belief about the state of the process should be updated **instantaneously**. GRU-Bayes is responsible for this and compares the prediction from GRU-ODE and the actual observation. This update necessarily has to be discrete to accommodate for the new information arriving in packets (or sporadically). See figures 1 and 2 for illustration of this dynamic. This also motivates why sporadic measurements cannot be fed directly to the ODE : observing a sample should make us update our belief immediately. A useful analogy is the Kalman filter, which consists in a **prediction** and an **update** phase. In prediction phase, it propagates in time the predictions about the distribution of the state (analog to GRU-ODE). In update phase, it computes a new estimate for the state probability distribution conditioned on the new information (analog to GRU-Bayes). Yet, in contrast to the (extended) Kalman filter, our approach is able to learn complex non-linear dynamics for the hidden process and is computationally cheaper.

More related works context. Other recent works have previously investigated the relationship between deep neural nets and differential equations. They mainly focused on deriving better deep architectures motivated by the stability of the corresponding partial differential equations (Ruthotto and Haber, *arXiv* 2018; Chang et al., *ICLR* 2019). Despite their ODE motivation, those approaches aim at designing new **discrete** architectures and don't explore neural networks parametrized ODEs as such.

Point processes (Mei and Eisner, *NeurIPS* 2017; Gunawardana et al., *NeurIPS* 2011) are intrinsically continuous as they focus on **time-to-event** modelling. Continuous-time Bayesian networks (Nodelman et al., *UAI* 2002) address a related problem where they frame events as state transitions. In contrast, our work aims at modelling continuous-time **real-valued** measurements, not only events. Yet, our continuous modelling of the latent process $h(t)$ allows to easily **jointly** model a continuous intensity function (*e.g.* by modulating the intensity function of a Poisson process with $h(t)$). This joint modeling of continuous measurements and events was left for future work.

Reviewer 2. Some assumptions have to be made about the conditional distribution of the observations. However, this is not very restrictive as a broad range of distributions have a tractable KL (*e.g.* Poisson, Exponential, ...).

The main difference with GRU-Discretized-Bayes resides in the continuity of the latent process (will be detailed in the Appendix). We then considered this experiment as a continuity ablation study. Note that the continuity prior can be tuned by rescaling time accordingly. Elongating time by a factor 2 would lead to a Lipschitz-4 prior.

Unlike sporadic measurements, the continuous measurements can indeed be directly fed to the GRU-ODE as suggested at line 89 and would considerably improve the prediction of the model as vital signs are very strong predictors in critical care. In this work we didn't include them (1) to compare equally with the other baselines which are not capable of handling continuous inputs, (2) we re-use the same variable subsets as used in (Che et al. *Scientific Reports* 2018). Still, we warmly welcome this suggestion from the reviewer as this would further highlight the capabilities of our model.

For fair comparison, all compared models were *made* probabilistic if not already (*i.e.* they all output log-variance).

Reviewer 4. We added the following (at l.264) : "We assessed the performance of the models by creating 5 different folds, each consisting of training (70%), validation (20%) and a left out test set (10%). Those folds are **reused** across compared architectures. For each fold, the models were trained with various hyper-parameters on the train set then evaluated on the validation set to select the best ones. We then retrain the model on the train data and assess performance on the test set. We do this 5 times and report the mean and stdev of our test set performances." Test samples were **never** used in training nor in tuning hyper-parameters, but for performance reporting **only**. The same train, val and test indices were used to compare all models and we used the **same number of hyper-parameters combinations** for all methods.

Minibatch is performed by jointly integrating all time series in the batch between unique observation times over the whole batch. At each unique time, we only update (with GRU-Bayes) the time series actually having an observation at that time and leave the other untouched. We then proceed with joint time integration until the next unique time.

Reviewer 5. A basic MLP does not enjoy the properties stated in sections 2.2 and in first paragraph of section 2.3. Still we investigated an other ODE parametrization : *GRU-minimal*, as described in Appendix G.

A important strength of our approach is to be able to predict future observations at **any** point in time. Yet it can also be easily used for time series classification or regression (e.g. using the last latent $h(T)$ to feed a classifier). If those labels are widely used in the literature, we are convinced vitals forecasting is just as important in medical practice as the health status of a patient is complex and cannot usually be fully captured by discrete labels.