We thank the reviewers for valuable and timely comments. We'd like to first emphasize the challenges and contributions:

- Finding the most relevant auxiliary forecasting tasks for pre-training and knowledge transferring to a given primary clinical outcome prediction task is challenging in that it is by nature a combinatorial optimization problem. Our paper solves this problem by building a new connection between multitask learning and transfer learning within the framework of meta-learning in an end-to-end fashion, which is new and has not been addressed before.
- We develop a novel two-loop optimization framework where the learned representation from pre-training is guided by the generalization performance of the target task rather than being model agnostic. We present an efficient first-order learning algorithm which learns the task weights end-to-end among up to hundreds of auxiliary time-series tasks.
- Extensive experiments with thorough generalization analysis demonstrates the superior predictive performance of our method and its robustness, interpretability, and generalization capability to unseen target tasks.

[Reviewer #3] I. Self-supervised Training (ST) exploits the abundant training signals that can be easily obtained from the data itself as augmentation to human labels. On the same dataset, it is often that only a small portion of the data has human labels especially for medical data, so the model can be first trained against these cheap signals via ST to get familiar with the structure of the data itself and then fine-tuned against expensive human labels. In our case, 1% of the time-series have human labels, but ST is conducted using all the patient time-series.

II. The hyperparameter λ is normalized by a softmax layer preserving $\sum_i \lambda_i = 1$ and optimized together with the model parameters in a two-loop learning process. The outer loop directly optimizes λ by minimizing the validation loss from the target task via hyper-gradient. This effectively change the hypothesis space from which our model parameters are optimized against the training loss in the inner loop. Section 3.2 explains how to calculate this hyper-gradient of λ by following the Lagrangian formulation originally used to analyze the back-propagation algorithm (A Theoretical Framework for BackPropagation, LeCun, 1988), and widely adopted in the literature [14, 15, 35]. Because λ determines the hypothesis space of the model parameters, they implicitly depend on λ in Equation 8. Finally, Algorithm 1 is based on the first order approximation to the Jacobian and Hessian in Equation [10-13].

III. The sub-index notation is slightly overloaded to indicate the dependency of the loss function on the parameters and on the dataset in Equation 2 and 3 explicitly. We would like to further polish the notation to be more consistent. Figure 2 plots the AUC on the validation dataset. It shows the learned representation gradually improves the generalization of the target task during pretraining stage. The decay in the finetune stage is due to overfitting since the training loss continues to improve for all compared methods.

IV. We also present the suggested experiment where **predicting the creatinine levels for the next 48 hours is used for pretraining** and then thresholding is applied for predicting the target task (kidney failure). From 1%, 10%, and even 100% data, **this method achieves 0.735(0.009), 0.833(0.017), and 0.897(0.012)**, which are worse than our approach due to the fact that the target task can receive additional benefits from other correlated trajectory predictions.

[Reviewer #1 and #4] I. In addition to AUC, **Average Precision (AP)** is reported in the Appendix, providing a quantitative evaluation of the relation between precision and recall(sensitivity). **II.** Most recent multitask learning algorithms focus either on the adaptability of trained models into new multitask learning settings or finding a mixing strategy specific to a few NLP tasks (**up to 8**) in [25, 26, 27]. By contrast, we have up to 100 tasks, and we further compare to the two-stage task selection work of [25] on the mortality task. From 1% and 10% data, **[25] achieves 0.805(0.001), 0.855(0.012)**, which are worse than our approach due to the separation of the task selection and task weighting phases. To the best of our knowledge, we are the first to handle the automatic task selection problem in the time series domain end-to-end by the time of writing.

[Reviewer #2] I. In the auxiliary tasks, the model is trained only using the true values as the targets instead of the imputed values. Because the event sequence of each patient is restricted to a window of 48 hours into both the future and the past history, the bias towards longer stays can be alleviated. **II.** Across the 10 different training data splits, 15 selected tasks for mortality prediction consistently rank within top 20 of all the tasks, which are annotated in Figure 2b. Moreover, the generalization analysis is designed for verifying the robustness of the learned task weights λ. λ learned from the mortality target task can be directly used for shock and kidney failure prediction since mortality is a more general endpoint than specific dysfunctions shown in Table 3.

[Reviewer #4] I. 5,000 iterations in Figure 2a are selected for the pretraining phase as the loss (MSE) of the auxiliary tasks converges and the validation performance of the auxiliary tasks reaches the peak. **II.** MIMIC-III is the largest open medical dataset, and as shown in the generalization analysis, the learned task weights can be generalized to unseen similar tasks without being retrained every time. The full-pretraining approach learns a representation that is agnostic to the target task. The useful information from relevant auxiliary tasks could be overwhelmed by noisy signals from unrelated tasks. In contrast, for AutoSelect, the task weights are guided by the generalization performance of the target task so that the learned representation from the pretraining stage is task aware. In spirits, this process is similar to PCA where the selected auxiliary tasks can be treated as the 'principle components'. This is also verified by Table 3a where 'Pretrain (Top)' is much better than 'Pretrain (Down)'.