Active Learning with Neural Processes for Epidemiology Modeling

Amogh Patankar
Halicioğlu Data Science Institute
University of California, San Diego
La Jolla, CA
apatankar@ucsd.edu

Abstract

Simulations in different fields such as epidemiology use models that can learn complex systems very accurately, with immense processing power and huge amounts of storage. One way to train a model that efficiently learns complex systems is by designing a model such that it can learn from different fidelities. Multi-fidelity models train neural processes on multi-fidelity datasets and sample a range of predictions while achieving cheaper simulation costs and better predictions. To improve on performance, this paper utilizes a technique called active learning. Active learning determines which training data to train on next via a reward function, further improving training convergence and stability. This paper explores the possibilities of extending active learning to multi-fidelity neural processes in order to improve accuracy, training time, and simulation cost.

1 Introduction

Computational models often characterize the relationship of physical systems in various scientific applications. In these systems, the input typically represents numerous different properties, while outputs of these systems represent the desired values and outcomes. In epidemiology, we utilize computational models to do forecasting of outbreaks of different viruses and outbreaks [1][2]. With regards to COVID-19, the inputs of these models [4] are including but not limited to, transmissibility, and contact rates. The outputs of these models describe the outcomes based on the aforementioned factors, such as predictions and forecasts.

Specifically, computational models in machine learning, are simulated in two different ways- low-fidelity and high-fidelity models. Low-fidelity models are cheaper with regards to cost, and pay the price with respect to [lower] accuracy. Alternatively, high-fidelity models produce outputs with higher accuracy, while also taking on higher costs.

Multi-fidelity models work with this trade off by combining outputs at various fidelity levels in order to speed up the time taken to learn. In doing so, we can gain accuracy predictions and insights that high-fidelity models would normally provide while having costs of low-fidelity models, improving the cost-accuracy tradeoff. In the simulation and multi-fidelity modeling space, there have been two types of processes utilized to tackle a plethora of problems- Gaussian processes and Neural Processes. Gaussian processes

Gaussian processes are models that define probability distributions over various functions; they are data efficient and flexible but they are computationally intensive [5]. Neural processes improve on Gaussian processes by adapting the priors to data, and thus improve model accuracy and reduce computation [6]. The downside of Gaussians is that they take a performance hit when it comes to high-dimensional data, which is a major issue when it comes to machine learning, specifically deep learning.

With the introduction of multi-fidelity hierarchical neural processes [8], neural processes can now be designed for and handle multi-fidelity data and outputs. This work also provides flexibility to combine data with varying dimensions, and specifically tailors to applications like climate modeling and epidemiology.

In this work, we combine active learning with the existing multi-fidelity hierarchical neural processes framework; the contributions are as follows:

- A variant of multi-fidelity hierarchical neural processes (MF-HNP), which uses acquisition functions, namely, maximum mean standard deviation (MEAN-STD) to improve performance.

- Real-world multi-fidelity application showing the use case for the utilization of active learning with regards to deep learning, specifically, for epidemiology modeling.

The code and data are available on https:/github.com/apatankar22/hier-neural-proc.

2 Related Work

Multi-fidelity modeling is a modeling strategy that has been used in various fields frequently, and works in the past have used gaussian processes at various different fidelity levels for a deep learning application [7]. Moreover, [8] includes disjoint data sets at low- and high-fidelity levels, which acts as a failure case for multi-fidelity Gaussian processes (NARGP), proposed in [9]. Moreover, deep gaussian processes introduced in [10] attempt to optimize parameters at fidelity levels jointly. With regards to different process types, [11] gives an alternative for gaussian processes by modeling using neural processes. But, the work done for that paper is unable to incorporate multi-fidelity data; the work done by [8] is able to do just that. [12] is an example where deep active learning has been used, but the application of this technique was image classification; the task at hand in this work is time series modeling. A similar technique is used in [13], where the task is natural language processing.

3 Background

3.1 Multi-Fidelity Modeling

With regards to modeling in a probabilistic setting, a model is defined as a function f; with an input domain $X \subseteq \mathbb{R}^{d_x}$ and output domain $\mathcal{Y} \subseteq \mathbb{R}^{d_y}$, then $f : X \to \mathcal{Y}$. When utilizing models like f, we are presented with computational costs that are greater than zero. These computational costs increase as we get to higher fidelities, and so, we are often limited with respect to high-fidelity data, especially as training data. When modeling in a multi-fidelity setting, we have functions $\{f_1, \cdots, f_K\}$ representing the costs of various f functions with their own individual accuracies and computational costs. Using multi-fidelity hierarchical modeling, we combine information from low-fidelity and high-fidelity models.

In this paper's work, we deal with epidemiology modeling, and we have data that corresponds to epidemic trajectories. In doing so, at a particular fidelity, we have x_k as our parameters; for a given scenario k, our data is as follows: $\mathcal{D}_k \equiv \left\{x_{k,i}, [y_{k,i}]_{s=1}^{S}\right\}_i$. In this given scenario, $[y_{k,i}]_{s=1}^{S}$ are S samples generated by $f_k\left(x_{k,i}\right)$ for scenario i. When applying this to epidemiology modeling in this paper, a scenario and corresponding data generates time series forecasting of daily infections using the model at hand.

The model used in this paper is a deep surrogate model approximating the data distribution at the highest fidelity level, given context sets at various fidelities and the corresponding input values. Having $\mathcal{D}_h$ as our high-fidelity data, and $\mathcal{D}_l$ as our low-fidelity data, we know that $\mathcal{D}_h \subset \mathcal{D}_l$, and because of this, the data domain X has a "nested structure". If $\mathcal{D}_h = \mathcal{D}_l$ we say the low- and high-fidelity data sets are "paired". The low and high fidelity data are split into context and target sets as follows:

low fidelity context set: $\mathcal{D}_l^c \equiv \left\{ x_{l,n}^c,\ \left[y_{l,n}^c \right]_{s=1}^S \right\}_{n=1}^{N_l}$

low fidelity target set: $\mathcal{D}_l^t \equiv \left\{ x_{l,m}^t, \left[y_{l,m}^t \right]_{s=1}^S \right\}_{m=1}^{M_l}$

high fidelity context set: $\mathcal{D}_h^c \equiv \left\{ x_{h,n}^c, \left[y_{h,n}^c \right]_{s=1}^S \right\}_{n=1}^{N_h}$

high fidelity target set: $\mathcal{D}_h^t \equiv \left\{ x_{h,m}^t, \left[y_{h,m}^t \right]_{s=1}^S \right\}_{m=1}^{M_h}$.

3.2 Neural Processes

Neural processes [6] are a family of conditional latent variable models for implicit stochastic processes, and produce somewhere between gaussian processes and neural networks. Neural processes are able to represent distributions over functions while also scale to higher dimensions, giving them an edge over gaussian processes. Using the Kolmogorov Extension Theorem [14], neural processes meet exchangeability and consistency conditions to define $\mathcal{SP}$ s.

Organically, neural processes' local latent variables z and θ are included and are trained using context sets and target sets. These sets are as follows:

CONTEXT: $\mathcal{D}^c \equiv \left\{ x_n^c, [y_n^c]_{s=1}^S \right\}_{n=1}^N$

TARGET: $\mathcal{D}^t \equiv \left\{ x_m^t, [y_m^t]_{s=1}^S \right\}_{m=1}^M$.

Neural processes use the evidence lower bound (ELBO) for training. The ELBO is calculated as follows:

$$\log p\left(y_{1:M}^t \mid x_{1:M}^t, \mathcal{D}^c, \theta\right) \geq$$
$$\mathbb{E}_{q_\phi(z|\mathcal{D}^c \cup \mathcal{D}^t)} \left[\sum_{m=1}^M \log p\left(y_m^t \mid z, x_m^t, \theta\right) + \log \frac{q_\phi\left(z \mid \mathcal{D}^c\right)}{q_\phi\left(z \mid \mathcal{D}^c \cup \mathcal{D}^t\right)} \right]$$

Neural Processes use neural networks, specifically encoder-decoder(s), to represent $q_\phi\left(z \mid \mathcal{D}^c\right)$, and $p\left(y_m^t \mid z, x_m^t, \theta\right)$.

$q_\phi()$ is referred as the encoder network (determined using ϕ), and $p(. \mid \theta)$ is referred as the decoder network (determined using θ). The encoder-decode architecture assumes that there is a Gaussian distribution that is followed by the latent variable(s) and outputs as well.

$$q_\phi\left(z \mid \mathcal{D}^c\right) = \mathcal{N}\left(z \mid \mu_z, \operatorname{diag}\left(\sigma_z^2\right)\right)$$
$$\mu_z = \operatorname{Enc}_{\mu_z,\phi}\left(\mathcal{D}^c\right), \quad \sigma_z^2 = \operatorname{Enc}_{\sigma_z^2,\phi}\left(\mathcal{D}^c\right)$$
$$p\left(y_m^t \mid z, x_m^t, \theta\right) = \mathcal{N}\left(y_m^t \mid \mu_y, \operatorname{diag}\left(\sigma_y^2\right)\right)$$
$$\mu_y = \operatorname{Dec}_{\mu_y,\theta}\left(z, x_m^t\right), \quad \sigma_y^2 = \operatorname{Dec}_{\sigma_y^2,\theta}\left(z, x_m^t\right)$$

4 Methodology

In this section, I introduce the acquisition functions used in conjuniction with the Multi-fidelity Hierarchical Neural Processes (MF-HNP) [8] framework.

4.1 Acquisition Functions

4.1.1 Low-level Mean

Using the mean of low-level mean of latent variables is an approach that is used to infer z_h. For every parameter θ, you generate a set of predictions $\{\hat{x}_{1:T}\}$ by sampling multiple $z_{1:T}$. So, for a given subset of T data, with D dimensions in each data point, we compute the mean of μ_{z_l} for a given time step (in our case, days) t and feature/attribute d. The calculation is as follows: $\overline{\mu} = \frac{1}{TD} * \sum_{t=1}^{T} * \sum_{d=1}^{D} * \sum \mu_{t,d}$ for each parameter θ, and the parameter θ with the highest $\overline{\mu}$ is selected.

4.1.2 Maximum Mean Standard Deviation

Maximum Mean Standard Deviation [12] is originally an approach that is used to estimate the model uncertainty, and to infer z_h, given μ_{z_l} and $\sigma^2_{z_l}$. For each parameter θ, you generate a set of predictions $\{\hat{x}_{1:T}\}$ by sampling multiple $z_{1:T}$. So, for a given subset of T data, with D dimensions in each data point, we compute the standard deviation of $\sigma_{t,d}$ for a given time step (in our case, days) t and feature/attribute d. Maximum Mean Standard Deviation (MEAN-STD) calculates $\overline{\sigma} = \frac{1}{TD} * \sum_{t=1}^{T} * \sum_{d=1}^{D} * \sum \sigma_{t,d}$ for each parameter θ, and the parameter θ with the highest $\overline{\sigma}$ is selected.

4.2 Multi-fidelity Hierarchical Neural Processes

Multi-fidelity Hierarchical Neural Processes is a framework introduced by Wu et al (2022) that utilizes Bayesian latent variable model's properties, simultaneously learning the joint distribution of multi-fidelity output.

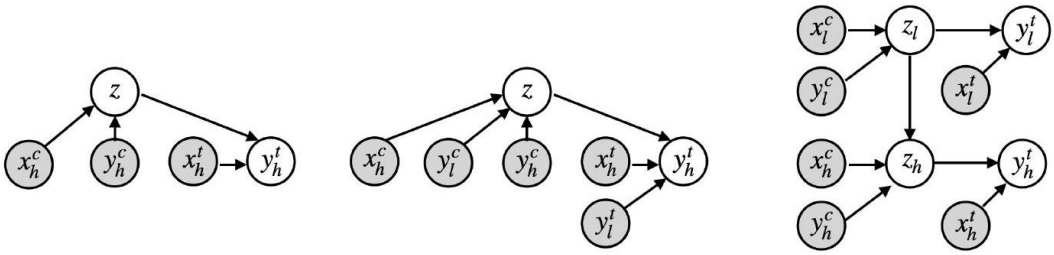

Figure 1: Graphical models for Single-Fidelity Neural Process (left), Multi-Fidelity Neural Process (middle), Multi-Fidelity Hierarchical Neural Process (right). Shaded circles denote observed variables and hollow circle represent latent variables. The directed edges represent conditional dependence.

As the figure shows, single fidelity neural processes (extreme left) work on the assumption that high-fidelity data is independent of the low-fidelity data. Multi-fidelity hierarchical neural processes (extreme right) assigns latent variables z_l and z_h at each respective fidelity level, with the prior of z_h is conditioned on z_l.

The MFHNP framework ensures that the model outputs at the respective fidelity levels are conditionally independent given the corresponding latent state [8]. Hence, correlations between fidelity levels are transformed to the latent space. Specifically, with vanilla multi-fidelity neural processes, $\hat{y}_h$ depends on (x_h, y_l) input pairs given z, while in MFHNP, $\hat{y}_h$ only depends on input x_h given z_h.

4.3 Scalable Training

As we use the same framework used by [8], the distributions are the same, and are as follows:

Neural Processes Family	Prior Distribution	Posterior Distribution	Generative Model
SF-NP [11]	$q\left(z_h \mid \mathcal{D}_h^c\right)$	$p\left(z \mid \mathcal{D}_h^c, \mathcal{D}_h^t\right)$	$p\left(y_h^t \mid x_h^t, z\right)$
MF-NP [42]	$q\left(z_h \mid \mathcal{D}_h^c\right)$	$p\left(z \mid \mathcal{D}_h^c, \mathcal{D}_h^t\right)$	$p\left(y_h^t \mid x_h^t, y_l^t, z\right)$
MF-HNP(As)	$q\left(z_h \mid z_l^{(s)}, \mathcal{D}_h^c\right)$	$p\left(z_h \mid z_l^{(s)}, \mathcal{D}_h^c, \mathcal{D}_h^t\right)$	$p\left(y_h^t \mid x_h^t, z_h\right)$
MF-HNP(MEAN)	$q\left(z_h \mid \mu_{z_l}, \mathcal{D}_h^c\right)$	$p\left(z_h \mid \mu_{z_l}, \mathcal{D}_h^c, \mathcal{D}_h^t\right)$	$p\left(y_h^t \mid x_h^t, z_h\right)$
MF-HNP(MEAN,STD)	$q\left(z_h \mid \mu_{z_l}, \sigma_{z_l}, \mathcal{D}_h^c\right)$	$p\left(z_h \mid \mu_{z_l}, \sigma_{z_l}, \mathcal{D}_h^c, \mathcal{D}_h^t\right)$	$p\left(y_h^t \mid x_h^t, z_h\right)$

Table 1: Comparison of different NP models at high-fidelity level.

Here the latent variables $z_l^{(k)}$ and $z_h^{(s)}$ are sampled by $q_{\phi_l}\left(z_l \mid \mathcal{D}_l^c\right)$ and $q_{\phi_h}\left(z_h \mid z_l^{(k)}, \mathcal{D}_h^c\right)$ respectively.

5 Experiments

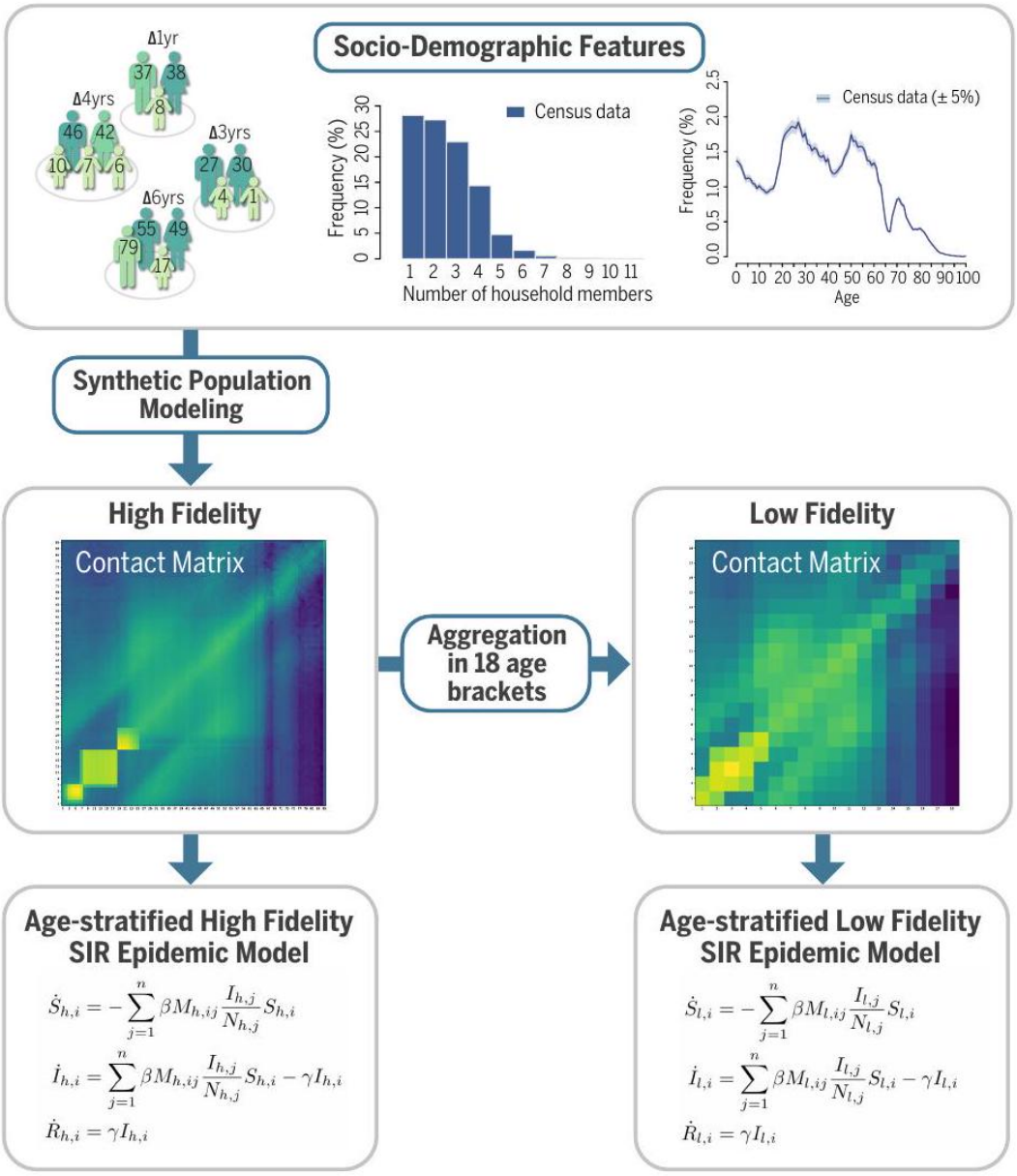

Figure 2: AS-SIR Modeling Framework: High-fidelity population-level contact matrices are generated using macro (census) and micro (survey) data [28]. Low-fidelity contact matrices are obtained by grouping individuals in fewer age brackets. Distinct age-stratified SIR models are used to simulate the epidemic at the two fidelity levels.

The performance of different active learning methods using the MFHNP framework are measured on stochastic epidemiology modeling (age-stratified). Figure 2 shows the modeling technique(s) that was also used in [8], as we use the same framework for this project.

5.1 Experiment Setup

For all experiments, the MF-HNP models with different acquisition functions (MEAN, MEAN-STD) are compared to the NARGP, SFGP, SFNP, and MFHNP baselines.

- The Gaussian Process (GP) baselines are the nonlinear auto-regressive multi-fidelity GP regression model (NARGP) [9] and single-fidelity Gaussian Processes (SFGP) model (under assumption of data independence at respective fidelity).
- The Neural Process (NP) baselines are the single-fidelity Neural Processes (SF-NP) from [15] and the multi-fidelity Hierarchical Neural Processes (MFHNP) [8].

For all neural process models, the context aggregation method used is mean context aggregation (MA), which generates latent variable z at each fidelity level. However, the NARGP baseline only works for nested data. Regarding model metrics, mean absolute error (MAE) is used for accuracy, and mean negative log likelihood (NLL) is used for uncertainty estimation. For age-stratified Susceptible-Infectious-Recovered (AS-SIR) experiment, we calculate NLL for AS-SIR experiment in the log space, and based on the Gaussian distribution. MAE is calculated in the original space, using the mean predictions and the truth.

5.2 Age-Stratified SIR Compartmental Model

We utilize the same age-stratified Susceptible-Infectious-Recovered (AS-SIR) epidemic model as [8]:

$$\dot{S}_i = -\lambda_i S_i, \quad \dot{I}_i = \lambda_i S_i - \gamma I_i, \quad \dot{R}_i = \gamma I_i$$

S_i, I_i, and R_i represent the number of susceptible, infected, and recovered age i individuals. Identical to [8], The age-specific "force" of infection is defined by λ_i:

$$\lambda_i = \beta \sum_j M_{i,j} \frac{I_j}{N_j},$$

β represents the transmissibility rate of the infection, N_j represents the total number of individuals of age j, and $M_{i,j}$ represents the overall age-stratified contact matrices describing the average number of contacts with individuals of age j for an individual of age i [8].

The active learning variants are built upon the same assumptions as the original MFHNP model: heterogeneous mixing of age groups to realistically capture the social mixing differences existing between various regions of the world [8].

Dataset. As per [8], there are exactly 109 scenarios from different locations in China, the United States, and Europe. The data in China, USA, and Europe is at the province, state, and country level, respectively. For each scenario, we generate 30 samples for 100 day's new infection prediction at low- and high-fidelity levels based on the corresponding initial conditions, R_0, age-stratified population, and the overall age-stratified contact matrices [8]. The high-fidelity data, as shown in Figure 2, has 85 age groups. As such, the age-stratified contact matrices $M_{h,ij}$ is of dimension 85×85. For low-fidelity data, the data obtained contains 18 age groups, resulting in a contact matrix $M_{l,ij}$ of dimension 18×18. All training, validation, and testing sets are conducted in the same manner as [8]; 31, 26, and 52 random scenarios at both fidelities, respectively.

Performance Analysis. Table 2 compares the prediction performance for both Gaussian Process (mean aggregation) methods and five MFHNP methods for daily infection forecasting over 100 days,

with performance is reported in MAE and NLL. MF-HNP(MEAN-STD) has the best performance with respect to MAE for nested data structure, meaning that using MEAN-STD as an acquisition function for active learning is fairly viable. As expected, both NARGP and SFGP underperform.

Process (Neural/Gaussian)	Mean Absolute Error (MAE)	Negative Log Likelihood (NLL)
Single Fidelity GP	342.989	1.715
Non-linear Autoregressive Multi-fidelity GP	342.737	1.746
Single Fidelity NP	293.128	3.193
Multi-fidelity NP	274.675	8.882
Multi-fidelity Hierarchical NP	258.128	6.233
Multi-fidelity Hierarchical NP (Mean)	238.987	1.986
Multi-fidelity Hierarchical NP (Mean-STD)	237.302	1.431

Table 2: Comparison of average performance of various processes on Age-Stratified SIR data sets.

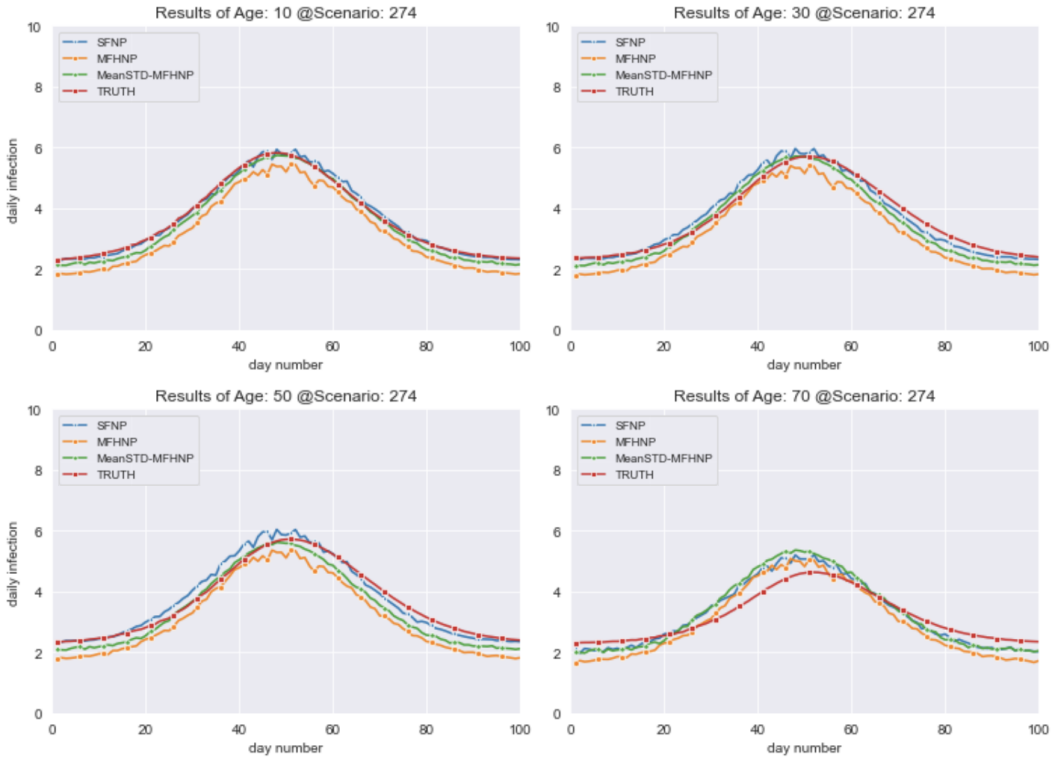

Figure 3: 100 days ahead infectious incidence compartment forecasting of randomly selected scenario at each row, analyzed in 4 age groups. Natural $\log$ scale for y axis.

Figure 3 shows the predictions of a randomly selected scenario in the nested dataset. It shows the truth, the SFNP and MFHNP predictions, as well as the MEAN-STD-MFHNP together with four age groups $(10, 30, 50, 70)$. In this experiment, the best neural process is MEAN-STD-MFHNP; while MEAN-STD-MFHNP has the best NLL score, studying its predictions across much more data reveals that the predictions made by the MEAN-STD acquisition function are rather conservative, and tend to fall in a larger band (i.e. larger confidence interval).

6 Conclusion and Limitation

In this paper, we show that using MEAN-STD [and to some extent, MEAN] is a viable approach to implementing active learning on top of the MFHNP framework. Applying these active learning methods to this framework is more accurate and efficient as compared to the standalone existing MFHNP framework as well as other neural and gaussian process approaches. Specifically, it utilizes these acquisition functions to score the data and invariably require less data to train on, while also supporting varying input and output dimensions at different fidelity levels (as per [8]). This report demonstrates the success of the aforementioned active learning methods on a large-scale time series application: age-stratified epidemiology modeling, an area that is highly prevalent nowadays.

Regarding future work, the next steps would be to utilize other acquisition functions, such as expected information gain (EIG), which is also known as Bayesian Active Learning by Disagreement, BALD, or maximum entropy. We could parameterize each feature using the mean and standard deviation in order to use EIG, or Regarding future work, the next steps would be to utilize other acquisition functions, such as expected information gain (EIG), which is also known as Bayesian Active Learning by Disagreement (BALD). We could parameterize each feature using the mean and standard deviation in order to use EIG. Another option would be to use maximum entropy as an acquisition function.

7 Acknowledgement

I acknowledge support from my mentors, Rose Yu, and Yian Ma for their continued support.

8 References

[1] Malchow, H., Petrovskii, S. V., Venturino, E. (2007). Spatiotemporal patterns in ecology and epidemiology: theory, models, and simulation. CRC Press.

[2] Nazia, N., Butt, Z. A., Bedard, M. L., Tang, W. C., Sehar, H., Law, J. (2022). Methods Used in the Spatial and Spatiotemporal Analysis of COVID-19 Epidemiology: A Systematic Review. International Journal of Environmental Research and Public Health, 19(14), 8267.

[3] Gandon, S., Day, T., Metcalf, C. J. E., Grenfell, B. T. (2016). Forecasting epidemiological and evolutionary dynamics of infectious diseases. Trends in ecology evolution, 31(10), 776-788.

[4] Jessica T Davis, Matteo Chinazzi, Nicola Perra, Kunpeng Mu, Ana Pastore y Piontti, Marco Ajelli, Natalie E Dean, Corrado Gioannini, Maria Litvinova, Stefano Merler, et al. 2021. Cryptic transmission of SARS-CoV-2 and the first COVID-19 wave. Nature 600, 7887 (2021), 127–132.

[5] Seeger, M. (2004). Gaussian processes for machine learning. International journal of neural systems, 14(02), 69-106.

[6] Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J Rezende, SM Eslami, and Yee Whye Teh. 2018. Neural processes. arXiv preprint arXiv:1807.01622 (2018).

[7] Raissi, M., Karniadakis, G. (2016). Deep multi-fidelity Gaussian processes. arXiv preprint arXiv:1604.07484.

[8] Wu, D., Chinazzi, M., Vespignani, A., Ma, Y. A., Yu, R. (2022). Multi-fidelity hierarchical neural processes. arXiv preprint arXiv:2206.04872.

[9] Paris Perdikaris, Maziar Raissi, Andreas Damianou, Neil D Lawrence, and George Em Karniadakis. 2017. Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling. Proceedings of the Royal Society A: Mathe- matical, Physical and Engineering Sciences 473, 2198 (2017), 20160751.

[10] Matteo Chinazzi, Jessica T Davis, Marco Ajelli, Corrado Gioannini, Maria Litvi- nova, Stefano Merler, Ana Pastore y Piontti, Kunpeng Mu, Luca Rossi, Kaiyuan Sun, et al. 2020. The effect of travel restrictions on the spread of the 2019 novel coronavirus (COVID-19) outbreak. Science (2020).

[11] Marta Garnelo, Dan Rosenbaum, Christopher Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo Rezende, and SM Ali Eslami. 2018. Conditional neural processes. In International Conference on Machine Learning. PMLR, 1704–1713.

[12] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep bayesian active learning with image data. In International Conference on Machine Learning, pp. 1183–1192. PMLR, 2017.

[13] Aditya Siddhant and Zachary C Lipton. Deep bayesian active learning for natural language processing: Results of a large-scale empirical study. arXiv preprint arXiv:1808.05697, 2018.

[14] Bernt Øksendal. 2003. Stochastic differential equations. In Stochastic differential equations. Springer, 65–84.

[15] Yating Wang and Guang Lin. 2020. MFPC-Net: Multi-fidelity Physics-Constrained Neural Process. arXiv preprint arXiv:2010.01378 (2020).

[16] Wu, D., Niu, R., Chinazzi, M., Vespignani, A., Ma, Y. A., Yu, R. (2021). Deep Bayesian Active Learning for Accelerating Stochastic Simulation. arXiv preprint arXiv:2106.02770.

A Experiment Details

For the Gaussian process baselines, we use RBF kernels. The optimal learning rate is $5e^{-2}$ for AS-SIR. We train 2000 epochs, with the patience with equal to 100 to ensure convergence. For NP baselines and our proposed MF-HNP model, the hyperparameters can be found in Table 4.

	LEARNING RATE	BATCH SIZE	PATIENCE	EPOCHS
AS-SIR	$1e^{-3}$	128	1000	2000

Table 4: Hyperparameters for all Neural Process models used are listed here.