NEURAL PROCESSES FOR SHORT-TERM FORECASTING OF WEATHER ATTRIBUTES

Benedetta L. Mussati
Department of Engineering Science
University of Oxford
blmussati@robots.ox.ac.uk

Helen McKay
Mind Foundry Ltd.

Stephen Roberts
Department of Engineering Science
University of Oxford
Mind Foundry Ltd.

ABSTRACT

Traditional weather prediction models rely on solving complex physical equations, with long computation time. Machine learning models can process large amount of data more quickly. We propose to use neural processes (NPs) for short-term weather attributes forecasting. This is a novel avenue of research, as previous work has focused on NPs for long-term forecasting. We compare a multi-task neural process (MTNP) to an ensemble of independent single-task NPs (STNP) and to an ensemble of Gaussian processes (GPs). We use time series data for multiple weather attributes from Chichester Harbour over a one-week period. We evaluate performance in terms of NLL and MSE with 2-hours and 6-hours time horizons. When limited context information is provided, the MTNP leverages inter-task knowledge and outperforms the STNP. The STNP outperforms both the MTNP and the GPs ensemble when a sufficient, but not exceeding, amount of context information is provided.

1 INTRODUCTION

Forecasting from time series data is of key relevance in the field of climate science. Data from multiple sensors records various correlated weather attributes. Modelling short-term dynamics is of particular interest to detect unexpected weather conditions, potentially precursors of extreme events and disasters such as flooding and high winds. Traditional methods for weather forecasting include classical statistical forecast methods and numerical weather prediction models. These approaches rely on complex physical models and generating forecasts is expensive both in terms of time and computation power.

In recent years there has been a surge in the use of machine learning models in climate sciences (Rolnick et al. (2019)), as they offer a more data-driven approach, identifying patterns in historical climate data to aid future predictions. Efficient models based on deep neural network architectures are a promising alternative for weather modelling (Espeholt et al. (2021)). These models can produce more timely weather predictions than traditional weather forecasting models and they can be used for short-term forecasting. Some of the machine learning models proposed for precipitation forecasting include using a U-Net with radar images (Agrawal et al. (2019)) and treating forecasting as an image-to-image translation problem, MetNet (Sønderby et al. (2020)), which uses input radar and satellite data to produce a probabilistic precipitation map, and MetNet-2 (Espeholt et al. (2021)). These models use images to forecast precipitations, whereas in our work we use time series weather attributes data.

We investigate how neural processes (NPs) (Garnelo et al. (2018a;b)) can be used for short-term forecasting, providing both expectation and uncertainty estimates over a range of weather-related variables. It was shown in Kim et al. (2022) that simultaneous learning a set of correlated tasks can help to leverage information from their joint distribution. This is useful when data is insufficient to

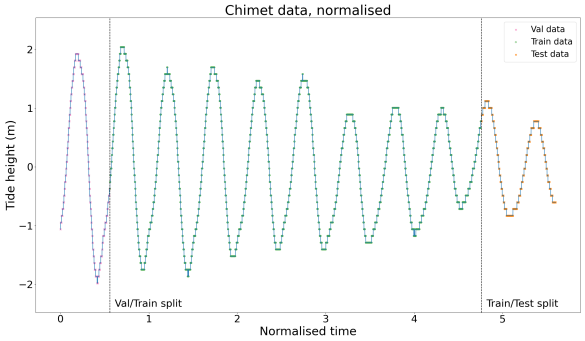


Figure 1: *Tide height (m)* weather attribute time series, with the train-validation-test splits. The training points are in green, the validation points in purple and the test points in orange. The timestamps have been converted to float values in range $[0, 6]$ and the datapoints have been standardised. The time series displays daily seasonality and periodic sea level changes.

reliable forecast one weather attribute, but we have information from another related attribute. We compare a multi-task neural process (MTNP), which learns a joint distribution of multiple weather attributes, to an ensemble of single-task neural processes (STNPs) and to an ensemble of GPs. Kim et al. (2022) have used MTNPs for long-term weather forecasts at multiple locations, while our work focuses on short-term forecasting at a single location, from a week of time series data collected by a set of weather, sea state and environment sensors.

We present preliminary results on using NPs for short-term forecasting multiple weather attributes. We observe that when the context information is insufficient, the MTNP leverages knowledge from related attributes to predict the dynamics. The STNP outperforms both the MTNP and the ensemble of GPs when limited but sufficient context information is provided.

2 METHODOLOGY

Data The *chimet* dataset was collected from the Chichester Harbour in the United Kingdom, between the 17th and the 23rd of August 2007. The weather attributes recorded at the location are: *air temperature (C)*, *max wave height (m)*, *mean wave height (m)*, *sea temperature (C)*, *tide height (m)*, *wind gust speed (kn)* and *wind speed (kn)*. As the readings are of various physical properties, they are naturally co-related by the physics of weather systems. Figures of the time series are in Appendix A. The readings from the multiple sensors are taken at roughly 1 min intervals, but the sampling is not exact. We cast time series forecasting as a multi-task meta-learning problem, where each weather attribute corresponds to a task.

Data preprocessing We convert the datetime indices into a fixed numerical range $[0, 6]$ and we standardise the raw measurements of each weather attribute to $\mathcal{N}(0, 1)$. The first 10% of the data is used for validation, the middle 75% is used for training and the final 15% is for evaluation, as shown in Fig. 1.

To create the NPs' training, validation and evaluation datasets, we take fixed-length (ie. the time horizon) slices of the original data with a fixed time lag between the start of each slice. These slices are the functions that make up the datasets and they correspond to samples from a task in the meta-learning settings. The MTNP and STNP are not translation equivariant and they cannot make predictions at input locations outside of the training range. Thus, we convert each slice of data to the x-range $[0, 1]$ before giving it as input to the models, and the sample windows have the same fixed length for training, validation and evaluation.

Models Neural processes (NPs) (Garnelo et al. (2018a;b)) are a family of meta-learning algorithms, which were developed as a neural network approximation to Gaussian processes (GPs) (Rasmussen & Williams (2006)). They are computationally efficient at training and evaluation and they can estimate the uncertainty in their predictions (Garnelo et al. (2018b)). Consider a dataset $D = \{C, T\}$ comprising of a labelled context set $C = (X_C, Y_C) = \{(x_i, y_i)\}_{i=1}^{I}$ and an unlabelled target set

$T = X_T = \{x_i\}_{i=I+1}^{I+M}$, where $x_i \in \mathcal{X}$ are inputs and $y_i \in \mathcal{Y}$ are outputs. We want to predict the outputs at T by learning the mapping $f : \mathcal{X} \to \mathcal{Y}$ that generated C. f is sampled from a distribution over functions P, which defines the conditional distribution $p(f(T)|C, T)$. The objective is to learn a model whose parameters maximise the likelihood of P. NPs use an encoder-decoder architecture to encode the context information and condition on it the decoder's predictions at target locations X_T. The encoder ϕ and the decoder ρ are learnable functions whose parameters are optimised during training. In this work, we consider the single-task neural process (STNP) and the multi-task neural process (MTNP) (Kim et al. (2022)), using the provided code[1] for our experiments. The STNP is an ensemble of N single-task attentive neural processes (Kim et al. (2019)), where N is the number of tasks. The latent variables $v^1, \ldots, v^N$ are independent, and the STNP can only model the marginal distribution for each task. The MTNP (Kim et al. (2022)) is a hierarchical model, with a single global latent variable z that captures knowledge about all tasks and task-specific latent variables $v^1, \ldots, v^N$ that capture task-specific knowledge. The inter-task correlation is expressed by conditioning the per-task latent variables on the global latent variable (Kim et al. (2022)).

To construct the GPs' ensemble, we train various GPs with different kernels and for each weather attribute we select the best performing GP on the validation set. At evaluation, we condition the fitted GPs on the same context points as the STNP and the MTNP. More details on the GPs' training and evaluation procedures are in Appendix B.

3 Results and analysis

As in Kim et al. (2022), we report the cumulative negative log-likelihood (NLL) and the cumulative mean squared error (MSE) to evaluate predictive performance. In the first set of experiments, we compare the MTNP and the STNP using datasets of 2-hours long slices with a 5 minutes lag between the start of each slice. Further details on the experimental settings are in Appendix C.

We experiment with missing rates $\gamma_{train}, \gamma_{eval} \in \{0.2, 0.5, 0.8\}$ and fixing the evaluation context set to size $cs_{eval} = 10$. The results in Table 1 suggest that the STNP usually performs better than the MTNP both in terms of NLL and MSE, when provided with sufficient context information. When $\gamma_{train}, \gamma_{eval} = 0.8$, the models receive less context information and the MTNP attains a lower NLL than the STNP. We hypothesise that the MTNP's performance is better than the STNP's because it can leverage the inter-task correlation and we perform an experiment to test this hypothesis, which is reported in Appendix C.

Missing rate γ_{train}	Missing rate γ_{eval}	Model	NLL	MSE
0.2	0.2	STNP	**−0.5913**	**0.0343**
		MTNP	−0.4438	0.0390
	0.5	STNP	**−0.4749**	**0.0435**
		MTNP	−0.3274	0.0458
	0.8	STNP	**0.0522**	0.0760
		MTNP	0.3661	**0.0756**
0.5	0.2	STNP	**−0.5890**	**0.0351**
		MTNP	−0.4341	0.0410
	0.5	STNP	**−0.4467**	**0.0443**
		MTNP	−0.2967	0.0484
	0.8	STNP	**0.0550**	0.0744
		MTNP	0.1676	**0.0735**
0.8	0.2	STNP	**−0.5478**	**0.0369**
		MTNP	−0.3158	0.0606
	0.5	STNP	**−0.4348**	**0.0446**
		MTNP	−0.2588	0.0660
	0.8	STNP	0.1910	**0.0724**
		MTNP	**0.1089**	0.0870

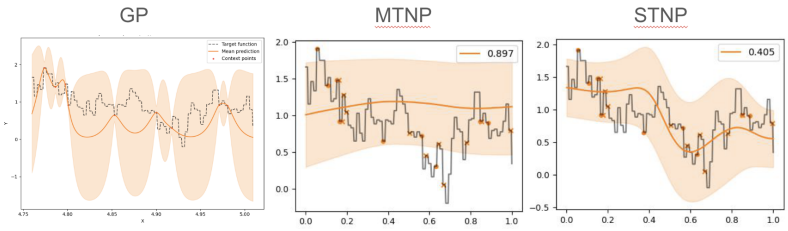
Table 1: Comparison of cumulative NLL and MSE on the 2-hours horizon, with varying missing rates $\gamma_{train}, \gamma_{eval}$ and fixed evaluation context set's size $cs_{eval} = 10$. The lowest NLL and MSE are highlighted for each combination of $\gamma_{train}, \gamma_{eval}$.

cs_{eval}	Model	NLL	MSE
10	MTNP	0.5542	0.1740
	STNP	**0.1449**	**0.0933**
	GP	0.4252	0.2955
20	MTNP	0.3058	0.1139
	STNP	**-0.0911**	**0.0725**
	GP	0.0348	0.1667
50	MTNP	0.2561	0.1086
	STNP	-0.2281	0.0632
	GP	**-0.5705**	**0.0475**
100	MTNP	0.2493	0.1069
	STNP	-0.2649	0.0616
	GP	**-0.9895**	**0.0203**
200	MTNP	0.2188	0.1044
	STNP	-0.2660	0.0613
	GP	**-1.3624**	**0.0080**

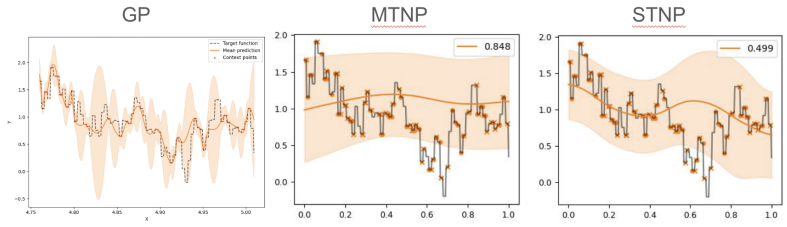
Table 2: Comparison of cumulative NLL and MSE on the 6-hours horizon, with varying evaluation context set's size cs_{eval}. The missing rates of the NPs are fixed $\gamma_{train}, \gamma_{eval} = 0.5$. For each cs_{eval}, bold represents the lowest NLL or MSE attained.

In the second set of experiments we use datasets of 6-hours long slices with a 30 minutes lag, to assess if inter-task correlation may help with a longer time horizon. We compare the performance at test time of an MTNP and a STNP with $\gamma_{train}, \gamma_{eval} = 0.5$ and of the GPs' ensemble comprising

[1] https://github.com/GitGyun/multi_task_neural_processes



(a) $cs_{eval} = 20$



(b) $cs_{eval} = 100$

Figure 2: Comparison of GP's, MTNP's and STNP's predictions on the same evaluation function of the *Wind speed (kn)* weather attribute on the 6-hours horizon. For the MTNP and STNP, the true target function is in black, the context points are in orange and the orange line with the confidence intervals is the $\mu \pm 2\sigma$ of the model's prediction. $\gamma_{train}, \gamma_{eval} = 0.5$ for both models. For the GP, the true target function is dashed and black, the context points are in red and the orange line with the confidence intervals is the $\mu \pm 2\sigma$ of the model's prediction.

the best-performing GPs at validation (see Appendix B). We compare the models with $cs_{eval} \in [10, 20, 50, 100, 200]$ and report in Table 2 the cumulative NLL and MSE. With smaller cs_{eval} such as 10 or 20, the STNP performs better than both the MTNP and the GPs' ensemble, whereas for $cs_{eval} \in \{50, 100, 200\}$ the GPs' ensemble outperforms both the MTNP and the STNP. Fig. 2 shows the three models' predictions for the weather attribute *Wind speed (kn)* with $cs_{eval} = 20$ and $cs_{eval} = 100$. The STNP is always better than the MTNP at approximating the general structure of the function. Too much context information may hinder performance, since the STNP's predictive distribution when $cs_{eval} = 100$ is slightly worse than when $cs_{eval} = 20$. Further research on the appropriate context set's size for NPs should be carried out. The GP's confidence intervals are larger than those of the MTNP and of the STNP when $cs_{eval} = 20$, which may imply that the GP is less certain of its prediction. Indeed, the predictive mean does not seem to approximate the true function as well as the STNP in the areas with no context points. When $cs_{eval} = 100$, the GP's predictive mean closely approximates the target function. A similar image for the weather attribute *Tide height (m)* is reported in Appendix D.

4 Conclusion and Future Work

In conclusion, this work shows preliminary results on how neural processes can be used as data-driven approach to learn short-term dynamics for forecasting purposes. We evaluated both the STNP and the MTNP on forecasting multiple weather attributes at a 2-hours time horizon and at a 6-hours time horizon. The STNP performs better than the MTNP in most evaluation settings. Since the MTNP and the STNP are not translation equivariant, we convert each slice of data to $[0, 1]$, but we lose the notion of different timestamps. The MTNP loses information that a pattern of one weather attribute can be observed with various patterns of another weather attribute. This may result in the MTNP not learning the proper correlation between the weather attributes, which could explain why the MTNP performs worse than the STNP. We will investigate introducing translation equivariance in the MTNP, starting by adapting the approach in the ConvCNP (Gordon et al. (2020)) and in the ConvNP (Foong et al. (2020)). When forecasting at the 6-hours time horizon, the STNP performs better than the ensemble of GPs when the context set is small, but it is beaten as context set's size increases. One of the next steps in our research is to estimate the optimal amount of context points for an NP, as too many context points may hinder its performance. It would be useful to re-run the experiment on a one

month time series, to have more datapoints and to generate longer slices or slices with a longer lag in between. This may be suitable to better capture the seasonality of each weather attributes, which may be longer than daily for some attributes.

ACKNOWLEDGMENTS AND DISCLOSURE OF FUNDING

Benedetta L. Mussati acknowledges funding and support from Mind Foundry Ltd. and the EPSRC Centre for Doctoral Training in Autonomous Intelligent Machines and Systems (Grant No. EP/S024050/1).

REFERENCES

Shreya Agrawal, Luke Barrington, Carla Bromberg, John Burge, Cenk Gazen, and Jason Hickey. Machine learning for precipitation nowcasting from radar images. *CoRR*, abs/1912.12132, 2019. URL http://arxiv.org/abs/1912.12132.

Lasse Espeholt, Shreya Agrawal, Casper Kaae Sønderby, Manoj Kumar, Jonathan Heek, Carla Bromberg, Cenk Gazen, Jason Hickey, Aaron Bell, and Nal Kalchbrenner. Skillful twelve hour precipitation forecasts using large context neural networks. *CoRR*, abs/2111.07470, 2021. URL https://arxiv.org/abs/2111.07470.

Andrew Y. K. Foong, Wessel P. Bruinsma, Jonathan Gordon, Yann Dubois, James Requeima, and Richard E. Turner. Meta-learning stationary stochastic process prediction with convolutional neural processes. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/5df0385cba256a135be596dbe28fa7aa-Abstract.html.

Marta Garnelo, Dan Rosenbaum, Christopher Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo Jimenez Rezende, and S. M. Ali Eslami. Conditional neural processes. In Jennifer G. Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1690–1699. PMLR, 2018a. URL http://proceedings.mlr.press/v80/garnelo18a.html.

Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J. Rezende, S. M. Ali Eslami, and Yee Whye Teh. Neural processes. *CoRR*, abs/1807.01622, 2018b. URL http://arxiv.org/abs/1807.01622.

Jonathan Gordon, Wessel P. Bruinsma, Andrew Y. K. Foong, James Requeima, Yann Dubois, and Richard E. Turner. Convolutional conditional neural processes. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=Skey4eBYPS.

Donggyun Kim, Seongwoong Cho, Wonkwang Lee, and Seunghoon Hong. Multi-task neural processes. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*, 2022. URL https://arxiv.org/abs/2110.14953.

Hyunjik Kim, Andriy Mnih, Jonathan Schwarz, Marta Garnelo, S. M. Ali Eslami, Dan Rosenbaum, Oriol Vinyals, and Yee Whye Teh. Attentive neural processes. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*, 2019. URL http://arxiv.org/abs/1901.05761.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, 2006. ISBN 026218253X. URL https://www.worldcat.org/oclc/61285753.

David Rolnick, Priya L. Donti, Lynn H. Kaack, Kelly Kochanski, Alexandre Lacoste, Kris Sankaran, Andrew Slavin Ross, Nikola Milojevic-Dupont, Natasha Jaques, Anna Waldman-Brown, Alexandra

Luccioni, Tegan Maharaj, Evan D. Sherwin, S. Karthik Mukkavilli, Konrad P. Körding, Carla P. Gomes, Andrew Y. Ng, Demis Hassabis, John C. Platt, Felix Creutzig, Jennifer T. Chayes, and Yoshua Bengio. Tackling climate change with machine learning. *CoRR*, abs/1906.05433, 2019. URL http://arxiv.org/abs/1906.05433.

Casper Kaae Sønderby, Lasse Espeholt, Jonathan Heek, Mostafa Dehghani, Avital Oliver, Tim Salimans, Shreya Agrawal, Jason Hickey, and Nal Kalchbrenner. Metnet: A neural weather model for precipitation forecasting. *CoRR*, abs/2003.12140, 2020. URL https://arxiv.org/abs/2003.12140.

A TIME SERIES OF THE WEATHER ATTRIBUTES

The time series of the weather attributes used in the experiments are shown in Fig. 3. Our goal is to learn their short-term dynamics. We evaluate if learning across weather attributes can enhance the accuracy of attributes forecasts.

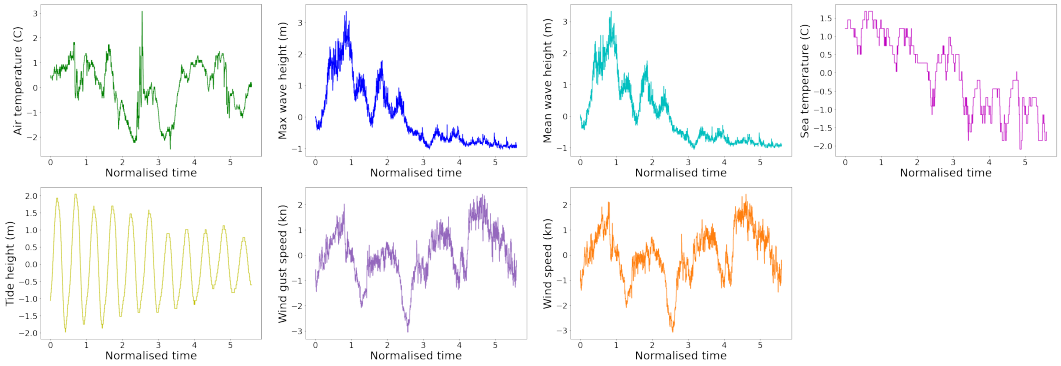


Figure 3: Time series of the weather attributes used in the experiments: *Air temperature*, *Max wave height*, *Mean wave height*, *Tide height*, *Sea temperature*, *Wind gust speed* and *Wind speed*. For each weather attribute the raw measurements are standardised and on the x-axis there are the datetime indices converted to the numerical range $[0, 6]$.

B CONSTRUCTING THE GPS' ENSEMBLE

To construct the GPs' ensemble, we train various GPs with different kernels. We select the best performing GP on the validation set for each weather attribute. The validation results are reported in Table 3. Although the GP with *Exponential* kernel attains lowest NLL in almost all tasks, it is a non-smooth kernel. As shown in Fig. 4, its predictions are non-smooth and such kernel is not an appropriate choice for fitting to the data in these experiments. Instead, we select the GP with a smooth kernel that attains the lowest NLL for each weather attribute.

Since the GPs are translation equivariant, the creation of the datasets is different from that required by the NPs. The x-range of the functions is not converted to $[0, 1]$. To ensure a fair comparison between models, the NPs and the GPs receive the same context points at training, validation and evaluation. At training we use also the target points to fit the GPs, since they are used in the backprop phase of the NPs' training. Contrary to the NPs, which are trained on slices of data over multiple epochs, a GP is fit to the selected points of all training functions in a single step, and we use the *L-BFGS-B* optimisation algorithm to optimise the kernel's hyperparameters on the training data. At validation and evaluation, we condition the fitted GPs to the context points and predict at all points in the function. Contrary to training, at validation and evaluation the GP's performance is computed on each function and then the average performance on the dataset is reported.

Kernel	Air Temperature	Max Wave Height	Mean Wave Height	Sea Temperature	Tide Height	Wind Gust Speed	Wind Speed
RBF	0.5688	1.2260	1.2330	1.1070	**−0.6307**	0.6630	0.6584
Periodic	0.9693	1.1560	1.1690	1.0390	0.0358	1.0380	1.0390
Exponential	**−0.4101**	**0.0301**	**0.0472**	**−0.7191**	0.3292	**0.1127**	**0.1237**
Matern32	0.2862	0.7377	0.7001	0.2463	−0.0408	0.5301	0.5156
Matern52	0.4381	0.9497	0.9251	0.6458	0.5700	0.5924	0.5796

Table 3: Comparison of GPs with different kernels. The normalised NLL for each task on the validation set for the 6-hours time horizon is reported. For each task, the lowest NLL is highlighted.

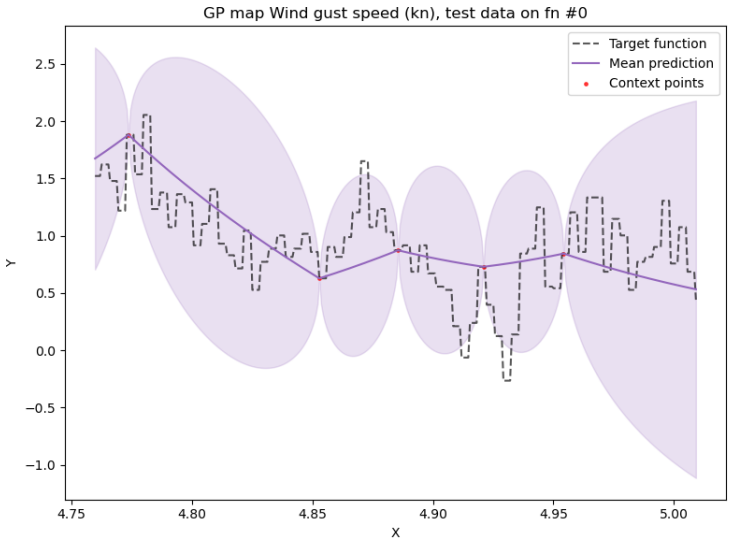


Figure 4: The black dashed line is the true target function, the red points are the context set that the GP is conditioned on and the purple line with the confidence intervals is $\mu \pm 2\sigma$ of the GP's predicted $p(f(T)|T, C)$. The GP has a non-smooth exponential kernel and its prediction is also non-smooth, as it displays abrupt, discontinuous changes in the graph.

C 2-HOURS TIME HORIZON EXPERIMENTS

When reporting the cumulative NLL and MSE across all $N = 7$ tasks, we compute them as:

$$\text{NLL} = \frac{\sum_{n=1}^{N} \text{NLL}^n}{N} = -\frac{\sum_{n=1}^{N}\sum_{m=1}^{M} \log p(\hat{y}_m^n | x_m^n, C)}{N \times M}$$

$$\text{MSE} = \frac{\sum_{n=1}^{N} \text{MSE}^n}{N} = \frac{\sum_{n=1}^{N}\sum_{m=1}^{M} (\hat{y}_m^n - y_m^n)^2}{N \times M}$$

where M is the total number of target points and $\hat{y}_m^n$ is the ground truth at target location x_m for task n.

SETTINGS SHARED BY THE 2-HOURS TIME HORIZON EXPERIMENTS AND THE 6-HOURS TIME HORIZON EXPERIMENTS

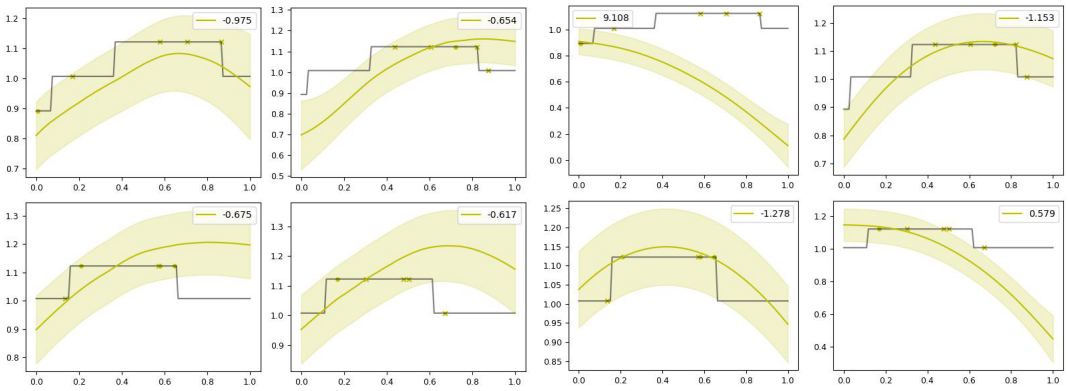
The training routine is $50,000$ epochs and the model is validated every $1,000$ epochs. At every training epoch, the size of the context set is chosen at random from the range $[10, 30]$. As in Kim et al. (2022), the incomplete context data is constructed by selecting a complete subset of context points and then randomly dropping the output points independently according to the *missing rate* $\gamma \in [0, 1]$, where $\gamma = 0$ means complete data. In all the experiments, we select a validation context set of size $cs_{valid} = 20$ and we always use $\gamma_{valid} = \gamma_{train}$. $C \subseteq T$ in all training and evaluation phases. All available points in the validation dataset are used as targets. At evaluation, we use the model which has achieved lowest cumulative NLL on the validation set and we predict at all available points in the evaluation set.

EXPERIMENT ON LEVERAGING INTER-TASK CORRELATION WHEN CONTEXT INFORMATION IS INSUFFICIENT

To test the hypothesis that the MTNP leverages inter-task correlation when insufficient context information is provided, we fix $\gamma_{train}, \gamma_{eval} = 0.8$ and we compare the MTNP and the STNP on evaluation context set's sizes $cs_{eval} \in [5, 10, 20]$. From the results reported in Table 4, we observe that with small context set's size such as $cs_{eval} \in \{5, 10\}$, the MTNP performs better in terms of

cs_{eval}	Model	NLL	MSE
5	STNP	0.9839	**0.1455**
	MTNP	**0.5502**	0.1515
10	STNP	0.1910	**0.0724**
	MTNP	**0.1089**	0.0870
20	STNP	**−0.3140**	**0.0505**
	MTNP	−0.1668	0.0694

Table 4: Comparison of cumulative NLL and MSE on the 2-hours horizon, with fixed missing rates $\gamma_{train}, \gamma_{eval} = 0.8$ and varying evaluation context set's size cs_{eval}. The lowest NLL and MSE are highlighted for every cs_{eval}.



(a) MTNP (b) STNP

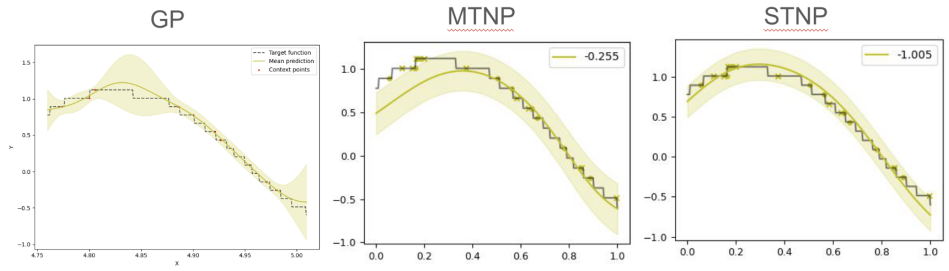
Figure 5: Comparison of STNP's and MTNP's predictions on evaluation functions of the *Tide height (m)* weather attribute on the 2-hours horizon. For each image, the true target function is in black, the context points are in green and the green line with the confidence intervals is the $\mu \pm 2\sigma$ of the model's prediction. The experiment settings use $cs_{eval} = 5$ and $\gamma_{eval} = 0.8$. The legend reports the NLL for the function.

NLL. Jointly inferring multiple tasks enables the MTNP to reason about a task's expected behaviour, based on how the other tasks are behaving at that time. This helps maximising the likelihood of $p(f(T^n)|C^n, T^n)$ for task n when C^n is too small. As the size of the context set increases, more information becomes available, and the NLL and MSE of both models decrease. Further experiments will be done to assess if and how inter-task knowledge hinder the model's performance when C is sufficiently large. Fig. 5 compares the MTNP and the STNP predictions for the weather attribute *Tide height* when using $cs_{eval} = 5$. We can observe that the MTNP's predictions are reasonable for all samples, whereas the STNP erroneously predicts the first function.

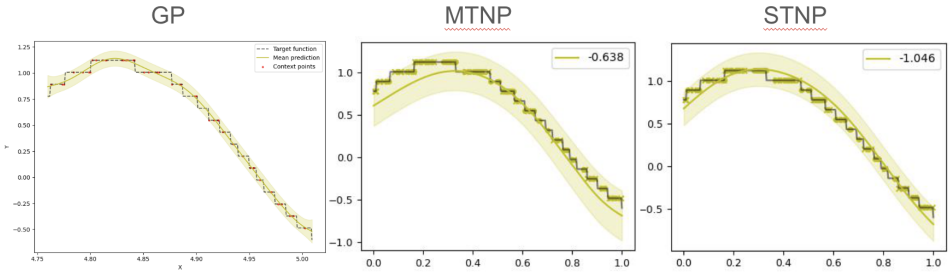
D 6-HOURS TIME HORIZON EXPERIMENTS

ADDITIONAL IMAGES FOR THE 6-HOURS TIME HORIZON EXPERIMENTS

Fig. 6 shows the three models' predictions for the weather attribute *Tide height (m)* with either $cs_{eval} = 20$ or $cs_{eval} = 100$. For this task, all models output reasonable predictive mean and confidence intervals. Arguably, the MTNP and STNP's predictive distributions follow the general trend better than that of the GP's when $cs_{eval} = 20$.



(a) $cs_{eval} = 20$



(b) $cs_{eval} = 100$

Figure 6: Comparison of GP's, MTNP's and STNP's predictions on the same evaluation function of the *Tide height (m)* weather attribute on the 6-hours horizon. For the MTNP and STNP, the true target function is in black, the context points are in green and the green line with the confidence intervals is the $\mu \pm 2\sigma$ of the model's prediction. For both models, $\gamma_{train}, \gamma_{eval} = 0.5$ and the legend reports the NLL for the function. For the GP, the true target function is dashed and black, the context points are in red and the green line with the confidence intervals is the $\mu \pm 2\sigma$ of the model's prediction.