Improved Attentive Neural Processes

Anirudh Suresh
SEAS, Harvard University
anirudh_suresh@college.harvard.edu

Srivatsan Srinivasan
SEAS, Harvard University
srivatsansrinivasan@g.harvard.edu

Abstract

Attentive Neural Processes (ANPs) mitigate traditional Neural Processes'(NP) under-fitting of context data by incorporating multiple attention modules into training, creating a query-specific context representation for each input query instead of the mean aggregated context vector created in NPs. Thus, ANPs boast better prediction accuracy, lower training time, and better flexibility in terms of modeling a wider range of functions. In this work, we further the ability of ANPs to learn local latent structure in the data by (1) adding local latents for each context point and generating query-specific local latent representation via attention and (2) implementing self-attention in the decoder. We demonstrate the performance of these improvements in specifically constructed synthetic 1-D regression tasks and a few-shot regression MNIST image completion task.

1 Introduction

Neural processes (NPs) [1] are a method tailored to combine the computational efficiency of neural networks with the desirable properties of stochastic processes. NPs set up the regression problem as learning a distribution over regression functions given a set of input-output pairs as context, with each function modeling the distribution of the output given an input conditioned on the context (Figure 3(a)). However, the learning process in NPs depends on a single aggregate context embedding (usually mean) from the entire context set, which can serve as a bottleneck, since this representation is invariant to the query (point) the model decodes. To counter this, the Attentive Neural Process (ANP) [2] adds self-attention and cross-attention modules to the NP framework in order to account for differential relevance of context points toward particular target points and better learn the internal structure of context data. The resulting model is better equipped to produce crisper distributional predictions as opposed to collapsing to a mean prediction and outperforms NPs with regard to reconstruction of context information (Figure 3(b) in Appendix B).

While training with a query-specific context representation, the ANP still assumes a single global latent variable that is parameterized by an aggregation (mean) of output embeddings of the entire context data. [2] posits that this global latent variable is key toward recognizing any global structure in the task arising from dependencies between context points. A single global latent might be restrictive in datasets which have strongly varying local structures, which we aim to address in this work. Specifically, we retain the global latent variable and add an additional cross-attention mechanism over a set of local latents learned from the context embeddings in order to exploit the differential relevance of the particular local structure of context points with respect to target queries. Additionally, we replace the multilayer perceptron (MLP) in the decoder with a self-attention module, hence replacing all instances of MLPs in the ANP with attention. Our model, called ANP^{++} (Figure 1), performs at least as well as ANP, if not better, on a set of few-shot regression tasks–different 1-D synthetic regression tasks and a 2-D MNIST image completion task.

4th workshop on Bayesian Deep Learning (NeurIPS 2019), Vancouver, Canada.

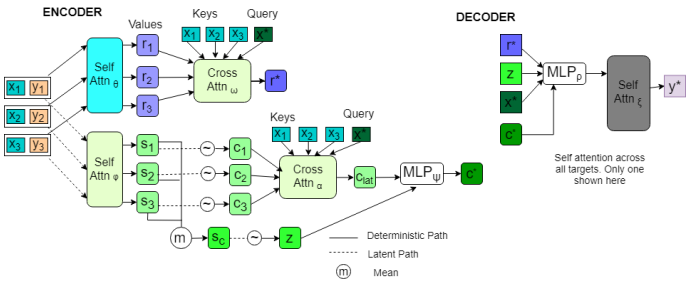


Figure 1: Schematic description of ANP^{++}

2 Methods

2.1 Model: ANP^{++}–ANP with cross-attention across the local latents

In the ANP model (Figure 3(b)), the global latent variable z models serves to enforce certain global properties of the task across all target predictions. Each single sample of z is meant to represent a single realization from the distribution of functions that explain the observed data. While having a global latent is necessary to preserve global structure of the distribution, it is still insufficient to learn functions which have varying local structures (e.g. periodic functions whose period depends on input points). Hence, in our ANP^{++}, along with a global latent, we also add a cross-attention module across the learned local latents which capture the local structural properties, to provide a query-specific latent representation of the context points in Figure 1. Individual s_is are still generated similar to ANP from the context input-output pairs via self-attention, and a mean aggregation is still used to get a single s_C from these s_is: s_C once again parameterizes a distribution from which the global latent z is drawn in order to preserve global structural distribution. Simultaneously, each s_i parameterizes a local latent distribution from which a c_i is drawn. Formally, we have $z \sim \mathcal{N}(\mu_z(s_C), \sigma_z(s_C)), c_i \sim \mathcal{N}(\mu_c(s_i), \sigma_c(s_i)) \, \forall \, i \in C$, where μ_z and σ_z are the learned mean and variance parameter functions, respectively, for the global latent and μ_c and σ_c are the learned mean and variance parameter functions, respectively, for the local latents.

Each local latent c_i (for the i^{th} context point (x_i, y_i)) can be thought of as a global latent if the context set were a singleton consisting of only the corresponding point. Intuitively then, the cross-attention scheme that attends to the c_is (with X_C serving as the keys and X_T as the query) determines the relevance of the target queries to these local latent samples that capture local structure of the data distribution. Cross-attention yields a query-specific c_{lat} for each query $x_j \in X_T$, and we pass this output through a network along with z to yield a final local latent path output c^*. We pass this output and the global latent path output z to the decoder. We believe this model is useful when decoding a target query depends on local structure around the context points, as the query-specific local latent representation c^* captures this particular dependency while z preserves the global structure. Using $\mathcal{A}(k, q, v)$ to indicate an attention module and NN to refer to a feed-forward net, we have

$$c_j^* = \text{NN}\big(\, \mathcal{A}(X_C,\, x_j^*,\, \{c_i\}_{i=1}^{|C|}),\, z\big),\; r_j^* = \mathcal{A}(X_C,\, x_j^*,\, \{r_i\}_{i=1}^{|C|})$$

The ANP^{++} decoder is also endowed with self-attention, amending the MLP-only ANP decoder. This change is motivated by the idea that the self-attention module can force the decoder to adjust to the the decoded targets' internal dependencies with respect to each other [3]. The self-attention mechanism takes as its input a representation of the concatenation of $x_j \in X_T, r^*, c^*$, and z. A schematic description of the workings of the ANP^{++} model can be seen in Figure 1. The ELBO loss definition, latent variable parameterization, and inference details can be found in the Appendix A.

3 Experiments

We run three different few-shot regression experiments to benchmark ANP^{++}'s results against the ANP. Table 1 compares the target negative log-likelihood for different data-generating sources in few-shot regression settings. Using Gaussian Processes with Squared Exponential Kernel (**GP-SE**) and Periodic Kernel (**GP-PER**) as data-generating stochastic processes, we test the models' ability to

Experiment	ANP (Target NLL)	ANP^{++} (Target NLL)
GP-SE	0.72	**0.55**
GP-PER	0.48	**0.37**
Non-stationary Sines	0.81	**0.51**
MNIST	1.1	**0.84**

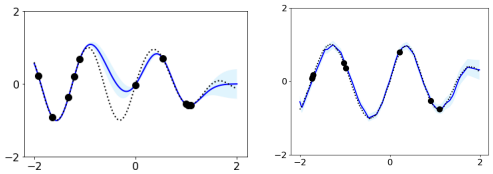
Table 1: Target negative log-likelihoods of the models across different experiments

learn a wider class of functions. The experiment's setup is similar to that of [2], with varying kernel parameters for each sample, and the number of context points is randomly chosen between 3 and 50 with 400 target points in total. We assert that ANP^{++} achieves better target NLL whilst retaining the ability to reconstruct the context points effectively (illustrated through plots in Appendix C).

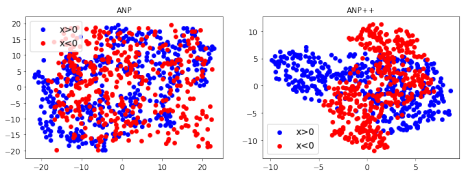
Next, we turn our attention to another experiment, **Non-stationary Sines**, with a synthetic 2-D dataset that has local structure that varies with position; we expect ANP^{++} to capture this local structure better than ANP. Specifically, the data is sampled from a non-stationary linear combination of sine functions: $y(x_i) = \sum_{k=1}^{K} w_k(x_i) \sin(\gamma_k \cdot \pi \cdot x_k)$, with $\sum_{k=1}^{K} w_k(x_i) = 1$. In this specific formulation, the weights of the linear combination are a function of the position x, and thus, we hypothesize that learning global statistics with a global latent vector z alone may not sufficiently capture this position-specific linear combination. Along with similar settings as the previous GP experiment, we choose a randomly varying K between 2 and 10 for each experiment.

Along with better target NLL observed in Table 1 and plots in Appendix C, Figure 2(a) demonstrates an example where ANP underperforms for non-stationary sines with position-specific periodic structure compared to ANP^{++}. Figure 2(b) shows a low-dimensional projection of the latent path vector used while decoding for a specific example with $K = 2$ and $w(x)$ being a step function at 0, in which we see that ANP^{++} learns a clearly distinguishable latent structure compared to ANP for the two classes of points ($x > 0, x < 0$) which have varying periodic structure in the data. For the non-stationary sines, we have also included a couple of sample predictions of the models as examples in Appendix D.

Finally, we experiment with a non-synthetic few-shot regression experiment (image completion) using MNIST digits (**MNIST**). Apart from the superior target NLL performance of ANP^{++}, the results in Appendix E show that the model is able to learn accurate and reliable realizations of digits using very few context points while still producing diverse samples through different realizations of the latent variable.



(a) ANP (left) vs ANP^{++} (right) on non-stationary sines. Dotted lines represent ground truth.



(b) ANP^{++} learns distinguishable latents for points $x < 0$ and $x > 0$.

Figure 2: ANP vs ANP^{++} on non-stationary sines. ANP^{++} captures local structure better.

We supplement our results with an exhaustive discussion on the time complexity, comparison with GPs, the marginal impact of different components, and possible extensions of this model in Appendix F. In a nutshell, across different experiments, we find that ANP^{++} better captures intricate local structure because it employs query-specific attention across local latents to produce both better context reconstruction and target prediction across all experiments, while training marginally faster (in terms of wall-clock time to convergence) than ANP (although ANP^{++} takes more time per epoch of training, see Appendix F for more details). In future work, one might focus on thorough ablation studies of this model, as well as training on larger image completion datasets such as CelebA and tasks where the contexts are chosen in a pre-determined fashion (e.g. first half of time series, one half of image etc.). This work can also be extended to other applications such as natural language (sentence completion) and reinforcement learning (transfer learning, exploration in bandits, etc.).

4 Conclusion

We introduce ANP^{++}, an iteration on the vanilla ANP that (1) further incorporates an attention module into the latent path in order to better capture local structure of the tasks and (2) improves the expressiveness of the decoder with self-attention. We show that ANP^{++} performs at least as well as vanilla ANP on various synthetic tasks, and we demonstrate specific experiments in which ANP^{++} outperforms ANP as a result of better accommodating local structure in the data. In a nutshell, ANP^{++} shows faster learning (in terms of number of iterations) and better context reconstruction and target prediction likelihood compared to ANP.

Acknowledgements

We thank David Belanger and Jasper Snoek for their valuable inputs and discussions with respect to the experiments and model design in this work.

References

[1] Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J. Rezende, S. M. Ali Eslami, and Yee Whye Teh. Neural Processes. *arXiv e-prints*, page arXiv:1807.01622, Jul 2018.

[2] Hyunjik Kim, Andriy Mnih, Jonathan Schwarz, Marta Garnelo, Ali Eslami, Dan Rosenbaum, Oriol Vinyals, and Yee Whye Teh. Attentive Neural Processes. *arXiv e-prints*, page arXiv:1901.05761, Jan 2019.

[3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *arXiv e-prints*, Dec 2017.

[4] Diederik P Kingma and Max Welling. Auto-Encoding Variational Bayes. *arXiv e-prints*, page arXiv:1312.6114, Dec 2013.

[5] Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight Uncertainty in Neural Networks. *arXiv e-prints*, May 2015.

[6] Chao Ma, Yingzhen Li, and José Miguel Hernández-Lobato. Variational Implicit Processes. *arXiv e-prints*, page arXiv:1806.02390, Jun 2018.

[7] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching Networks for One Shot Learning. *arXiv e-prints*, page arXiv:1606.04080, Jun 2016.

[8] Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P. Xing. Deep Kernel Learning. *arXiv e-prints*, page arXiv:1511.02222, Nov 2015.

[9] Scott Reed, Yutian Chen, Thomas Paine, Aäron van den Oord, S. M. Ali Eslami, Danilo Rezende, Oriol Vinyals, and Nando de Freitas. Few-shot Autoregressive Density Estimation: Towards Learning to Learn Distributions. *arXiv e-prints*, page arXiv:1710.10304, Oct 2017.

[10] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A Closer Look at Few-shot Classification. *arXiv e-prints*, page arXiv:1904.04232, Apr 2019.

[11] Danilo Jimenez Rezende, Shakir Mohamed, Ivo Danihelka, Karol Gregor, and Daan Wierstra. One-Shot Generalization in Deep Generative Models. *arXiv e-prints*, page arXiv:1603.05106, Mar 2016.

[12] Philip Bachman, Riashat Islam, Alessandro Sordoni, and Zafarali Ahmed. VFunc: a Deep Generative Model for Functions. *arXiv e-prints*, page arXiv:1807.04106, Jul 2018.

[13] Harrison Edwards and Amos Storkey. Towards a Neural Statistician. *arXiv e-prints*, page arXiv:1606.02185, Jun 2016.

[14] Luke B. Hewitt, Maxwell I. Nye, Andreea Gane, Tommi Jaakkola, and Joshua B. Tenenbaum. The Variational Homoencoder: Learning to learn high capacity generative models from few examples. *arXiv e-prints*, page arXiv:1807.08919, Jul 2018.

[15] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv e-prints*, page arXiv:1409.0473, Sep 2014.

[16] Yoon Kim, Carl Denton, Luong Hoang, and Alexander M. Rush. Structured Attention Networks. *arXiv e-prints*, page arXiv:1702.00887, Feb 2017.

[17] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. *arXiv e-prints*, page arXiv:1502.03044, Feb 2015.

[18] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Łukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image Transformer. *arXiv e-prints*, page arXiv:1802.05751, Feb 2018.

Appendix

A Model Details–Loss and Inference

Using variational approximations, the ELBO loss function for the ANP^{++} model can easily be constituted by the reconstruction loss and the KL-terms of the global and local latents.

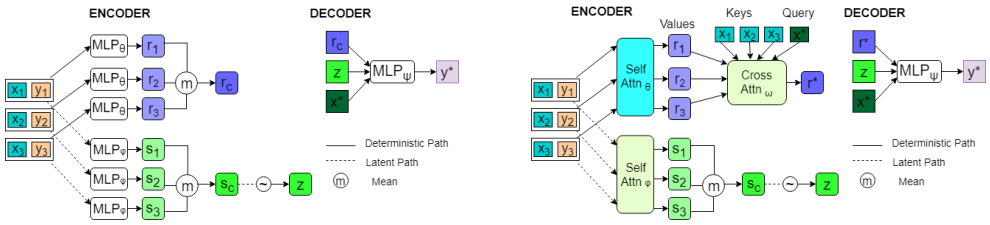
$$\underbrace{\mathcal{L}}_{\text{ELBO}} = \underbrace{R}_{\text{(likelihood)}} + \underbrace{G}_{\text{(global KL)}} + \underbrace{L}_{\text{(local KL)}} \qquad R = \mathbb{E}_{q(z,\{c_i\}_{i=1}^{|C|} \mid s_T)}\left[\sum_{j=1}^{|T|} \log p(y_j|x_j, r_j^*, c_j^*, z; \theta)\right]$$

$$G = D_{KL}\left(\, q(z|s_T;\phi) \,||\, q(z|s_C;\phi)\,\right) \qquad L = \sum_{i=1}^{|C|} D_{KL}\big(\, q(c_i|s_T;\lambda) \,||\, \underbrace{q(c_i|s_i;\lambda)}_{\substack{\text{local latent for} \\ \text{the } i^{th} \text{ context}}}\,\big)$$

The original ANP loss in [2] has only the first two components–reconstruction and global latents–$R+G$ (and that too, without the need for local latents c_is in the reconstruction term). Also, it is important to note that all latent variable parameterizations are from the Gaussian family (Refer to the Appendix for mathematical formulations). Inference is done via simple backpropagation, and we use an Adam optimizer in all our models. We use the re-parameterization trick to generate latent variable samples while still maintaining full differentiability of the computation graph. [4]. All the latent variables in ANP^{++} are parameterized by Gaussian families whose means and variances are learned from input contexts.

$$\begin{aligned}
q(z|s_C;\phi) &= \mathcal{N}(\mu(s_C), \sigma(s_C); \phi) \\
q(z|s_T;\phi) &= \mathcal{N}(\mu(s_T), \sigma(s_T); \phi) \\
q(c_i|s_T;\lambda) &= \mathcal{N}(\mu(s_T), \sigma(s_T); \lambda) \\
q(c_i|s_i;\lambda) &= q(c_i|s_C;\lambda) = \mathcal{N}(\mu(s_i), \sigma(s_i); \lambda)
\end{aligned}$$

B Schematic description of NP and ANP models



(a) Neural Process (NP) (b) Attentive Neural Process (ANP)

Figure 3: Schematic description of Neural Process (NP) and Attentive Neural Process (ANP)

C Results: Target and Context NLL curves for Synthetic Data

Figures 5 and 6 show the target and context negative log-likelihood (NLL), respectively, for data generated from a GP with Squared Exponential (SE) and Periodic (PER) kernels. Figure 7 shows the target and context NLL for contexts generated from periodic non-stationary sine curves. We see that in all cases, our model ANP^{++} has as good, if not better, target prediction and context reconstruction performance compared to ANP.

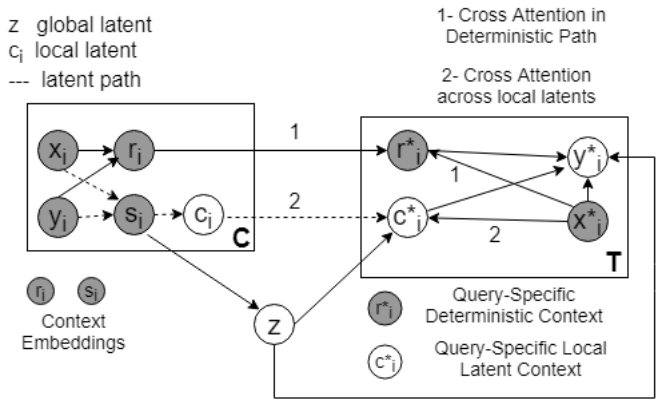


Figure 4: A sketch of the underlying graphical model for the ANP^{++}

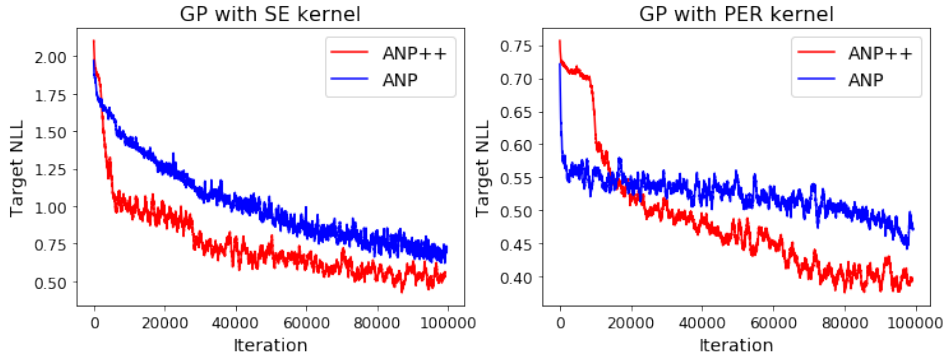


Figure 5: Target negative log-likelihood for synthetic data realized from a GP using SE and PER kernels with varying kernel parameters.

D Prediction samples for non-stationary sines data

Figure 8 shows a few examples from non-stationary sines data to demonstrate the distribution of functions fitted by ANP and ANP^{++}.

E Experiment : MNIST results

To see if model performance translates from a synthetic dataset to a real dataset, we also run an 2-D image completion task with MNIST data. MNIST data provides us two advantages: it is easily trainable with a simple feed-forward net, and the properties of the data are well-known. This problem can be thought of as a pairwise regression problem between the 2-dimensional position x_i and the corresponding pixel value y_i. Figure 9 shows some realizations of our model for different numbers of context points. We see that the model is capable of performing few-shot regression well, even with very few context points ($|C| = 10$). As a sanity check, we also test the model's reconstructive ability when the entire context is provided ($|C| = 784$) and we see that the output realizations almost entirely resemble the input contexts. Figure 10 again contains different samples realized for a number of different context points for a given global latent variable.

Another factor to ascertain in the ANP^{++} is whether the global latent variable z provides diverse stochastic realizations of the data-generating process. To do so, we generate samples with 10 context points as inputs and generate decoded samples with a different global latent variable z sampled each time. Figure 11 shows that the model predicts different digits for the same input context. This example indicates that the latent dimensions of the global latent variable z are expressive enough to induce decoding of diverse samples with limited context information.

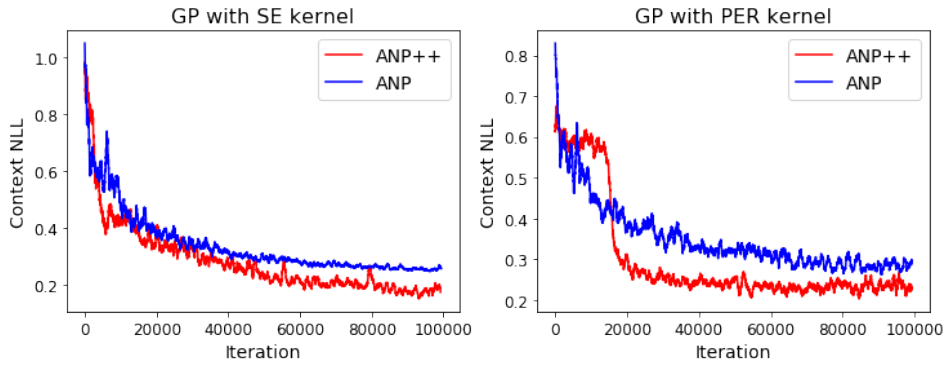


Figure 6: Context negative log-likelihood for synthetic data realized from a GP using SE and PER kernels with varying kernel parameters.

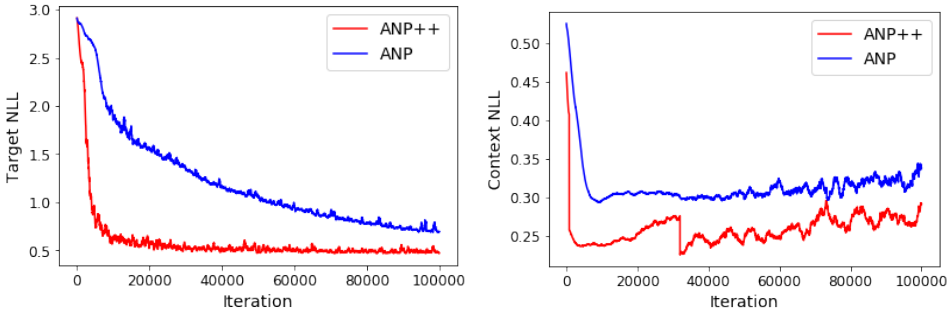


Figure 7: Target (left) and context (right) negative log-likelihood for synthetic data realized from periodic non-stationary sines whose period varied as function of x.

F Discussion

Time Complexity The significantly low computational complexity at inference time is what distinguishes NPs from GPs (linear versus quadratic/cubic). Yet, as witnessed both in this work and [2], attention mechanisms (both self- and cross-attention) are necessary to prevent NPs' drastic under-fitting of context points. One downside of the incorporation of attention into the NP framework is the supralinear scaling–assuming n_C context points and n_T "different" target points (members of $T \backslash C$), the cross-attention mechanism in the deterministic path entails $n_C + n_T$ queries attending to n_C key-value pairs. The complexity of this operation is $O(n_C(n_C + n_T))$; with the ANP^{++}, the self-attention in the decoder entails a slightly increased complexity of $O((n_C + n_T)^2)$. Therefore, we lose the desirable linear scaling of NPs; however, this quadratic scaling is not a major concern as the attention operations are all some form of matrix multiplication which could be massively parallelized with a GPU. Thus, the actual runtime (wall-clock) does not climb so dramatically in practice. Furthermore, our experiments indicate that the more sophisticated model (ANP$^{++} > ANP > NP$) converges in a fewer number of iterations, thus offsetting any time complexity overheads caused by these attention mechanisms.

NP vs. GP We have already established the fact that ANPs are more flexible (besides being computationally more efficient) than GPs and that they implicitly learn a "kernel" from the data. On the other hand, these advantages come at the cost of any neural process being just an approximation of the conditionals of the true stochastic process. In the 1-D synthetic experiments with a GP data, we also compare the learned uncertainties of the neural process models with respect to the original data-generating Gaussian Process. A commonality we observe (largely by visual inspection of the plots) is that all the neural processes (NP, ANP, ANP^{++}) have lower variance estimates for the targets than the posterior variance of the GP for the same context points. In other words, the NPs seem to be overconfident at the target points compared to the data-generating Gaussian Process. We believe that this could be explained by a few reasons which need further investigation: (1) the perceived overconfidence is an artifact of Gaussian variational approximations and performing backpropagation with reparameterization–similar to the observations in a BNN trained with Bayes by Backprop [5]; (2) the similarity among the different training tasks (curves) could be perceived by the model to be high enough and this could lead to the overconfidence.

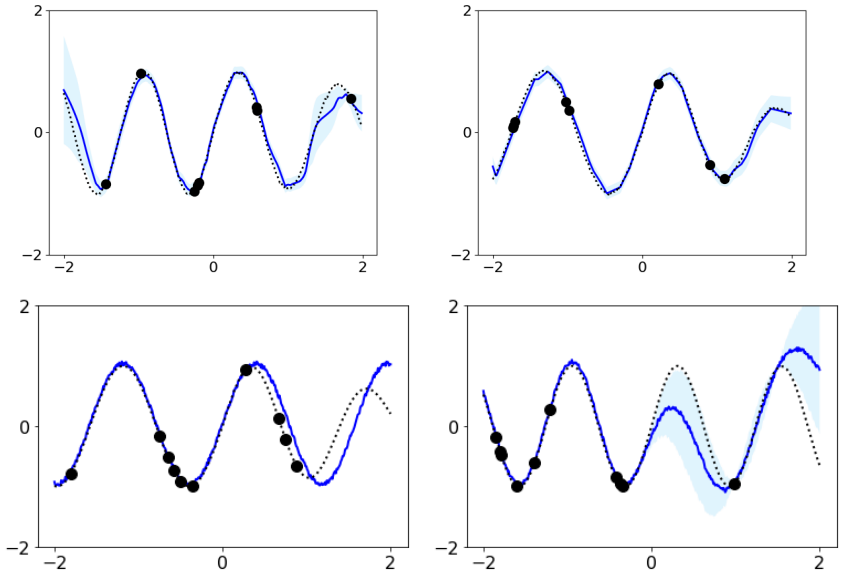


Figure 8: Sample predictions of ANP^{++} (top) and ANP (bottom) and for non-stationary sines data

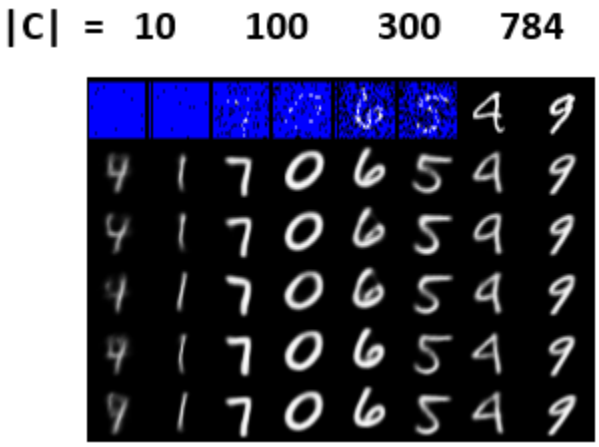


Figure 9: MNIST image completion results for different numbers of context points. The top row of the table indicates the context passed to the model. The samples in each column are generated by the predicted means and standard deviations of the model. Note that each column has been sampled with a single global latent vector z.

Marginal Contributions of the components Both the ANP and ANP^{++} have a deterministic and a latent path with ANP^{++} having an additional set of local latents. Since ANP^{++} also learns latent representations for each of the context points c_i, the deterministic embeddings learned for the context points r_i are expected to have a significant informational overlap with the learned local latents. To verify this, we ran a quick performance check by turning off the deterministic path in the 1-D synthetic data experiment with a squared-exponential kernel, and the context NLL and target NLL numbers seem fairly similar with and without the deterministic embeddings r_i. This observation means that the marginal value of deterministic embeddings and local latents needs to be investigated more systematically in future work. Neither the original ANP work [2] nor our work does any systematic ablation study in order to understand the incremental value each component (self-attention in deterministic path, self-attention in latent path, self-attention in the decoder, cross-attention in deterministic path, cross-attention in latent path). Besides removing redundant components and trimming computational overheads, such studies are also necessary to determine the real scope of our work and find fitting real-world applications.

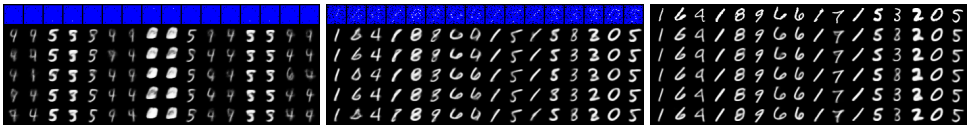


Figure 10: Some sample predictions for ANP^{++} with MNIST data for 0, 100 and 784 context points. In each column, the model uses the same latent z variable (one single sample).

Figure 11: MNIST image completion results for the same context, but with different realizations of the global latent z.

Extensions and Future Work Our work in this paper provides a few initial steps at improving the existing ANP model, and the initial experiments show promise. Having said that, there is huge scope for future investigation about the properties of ANP^{++}. Beyond the simple 1-D GP and sine curve experiments, more targeted experiments should continue to be conceived in order to build stronger evidence regarding this proposed advantage of the novel features of ANP^{++}. One other missing element in our work is experiments to understand the marginal effects of our two additions–self-attention in decoder and cross-attention across local latents. Further experiments need to be designed and executed to study the individual contributions of each of these changes as they both cause computational overheads to the model. In terms of real-life experiments on image completion, we wish to run the model on larger datasets such as CelebA and run experiments to check how the model behaves when the context points are not provided at random but rather based on a predetermined pattern (e.g. top half of the image, first half of a time series' data). Furthermore, one can also think about applications of the ANP^{++} model toward other domains–natural language (e.g. sentence completion) and reinforcement learning (e.g. transfer applications, exploration in bandits and RL) are two such categories of problems for which these neural process models could be a good fit.

G Related Work

Gaussian Processes The key advantage of regression learning with GP is the model's ability to incorporate similarity between points through the covariance structure using kernel functions. While NPs do not have this implicit ability, ANPs learn the similarity between points in their domain via the attention mechanisms. Also, the choice of kernel is arguably a detrimental factor in the training of GPs, whereas all members of the neural process family learn the kernels directly from the context data. On the other hand, GPs are more principled in the sense that their posterior predictive covariances and marginal variances can be expressed in a fully analytic form, while the family of neural processes has no such guarantees about the learned distributions. There are other works that learn data-generating processes with algorithms that lie somewhere in the spectrum between neural networks and Gaussian Processes. Variational implicit processes [6], for instance, define the stochastic process with a similar decoder and a finite-dimensional latent setup as the NP but ultimately learn the posterior with a GP approximation. Similar to NPs, matching networks [7] and deep kernel learning [8] are two other methods that extract representations directly from data, but both pass these representations to an explicit kernel and perform learning in a GP framework. Along the spectrum from neural networks to GPs, the neural process family resides closer to neural networks in terms of inference and computational complexity but possesses an ability to learn a distribution over functions.

Meta-learning Meta-learning models share the fundamental motivations of NPs as they shift some workload from training time to test time. Assuming we are given input-output pairs from any function at test time, NPs have the ability to probabilistically reason about this function conditioning on the given input-output pairs, making them ideal candidates for few-shot density estimations. On one hand, few-shot classification has received extensive focus from the research community, including attention based models in the last few years such as [9, 10, 11]. On the other hand, few-shot pairwise regression is comparatively less explored (e.g. [12]). In this work, we use attention mechanisms to learn the similarity between domain points in the few-shot pairwise regression setting, building on top of [1, 2] with the addition of cross-attention across local latents and a more expressive decoder compared to ANP.

Conditional Latent Variable Models One can group all the models similar to the family of neural processes under the canopy of conditional latent variable models. Such models learn the conditional distribution $p(X_T|X_C, z)$ where z is a latent variable that is sampled to generate different stochastic realizations. The well-known conditional VAEs [4] are an example of this class of models which use a conditional input apart

from a latent variable sample in order to generate an output. A more sophisticated version of CVAE that uses a hierarchical latent structure with global and local latents is the Neural Statistician [13], which uses z to capture global uncertainty, but instead of representing a distribution over possible functions, it learns simply an unconditional distribution over the sets in the input space. The NS does not generate different y values conditioned on inputs in a pairwise setting like a GP or NP but rather generates different samples x using the global latent z and additionally sampling local latents z_T. Notably, unlike in NPs, the prior of the global latent variable in the NS is not conditioned on z. Another variant of the NS model is the variational homoencoder [14], which has a similar hierarchical structure as [13] but uses a separate subset of data points for the context and predictions. In this family, neural processes have their own standing due to their ability to perform more targeted sampling of the latent distribution using x in a regression setting, while some of the other models described earlier have been demonstrated to be useful only in classification settings. An ability to do this kind of targeted sampling has interesting potential applications–e.g. conditional generation and completion tasks (image, natural language etc.) that do not fit fully well into the framework provided by other models.

Attention and Transformer Models The idea of attention in sequence-to-sequence models has been extensively pursued as a mechanism to aggregate value representations based on the similarity between a query and a set of keys that correspond to the values [15, 16, 17]. Transformers [3] use a combination of self-attention and cross-attention modules for sequence transduction, thus creating a fully attention-based model, dispensing with the usual recurrent encoders and decoders used for this purpose. Image transformers [18] extend this work to a sequence-modeling formulation of image generation with a tractable likelihood. In this work, we leverage self-attention modules in both the encoder and the decoder and two cross-attention modules between the encoder and the decoder, producing (just) the model architecture similar to the transformer models, yet applied in entirely different settings.