Efficient Queries Transformer Neural Processes

Leo Feng[*] [1,2] **Hossein Hajimirsadeghi** [1] **Yoshua Bengio** [2] **Mohamed Osama Ahmed** [2]
[1] Borealis AI [2] Mila – Université de Montréal

Abstract

Neural Processes (NPs) are popular methods in meta-learning that can estimate predictive uncertainty on target datapoints by conditioning on a context dataset. Previous state-of-the-art method Transformer Neural Processes (TNPs) achieve strong performance but require quadratic computation with respect to the number of context datapoints per query, limiting its applications. Conversely, existing sub-quadratic NP variants perform significantly worse than that of TNPs. Tackling this issue, we propose Efficient Queries Transformer Neural Processes (EQTNPs), a more computationally efficient NP variant. The model encodes the context dataset into a set of vectors that is linear in the number of context datapoints. When making predictions, the model retrieves higher-order information from the context dataset via multiple cross-attention mechanisms on the context vectors. We empirically show that EQTNPs achieve results competitive with the state-of-the-art.

1 Introduction

Meta-learning aims to learn a model that can adapt quickly and computationally efficiently to new tasks. Neural Processes (NPs) are a popular method in meta-learning that models a conditional distribution of the prediction of a target datapoint given a set of labelled (context) datapoints, providing uncertainty estimates. NP variants (Garnelo et al., 2018a; Gordon et al., 2019; Kim et al., 2019) adapt via a conditioning step in which they compute embeddings representative of the context dataset. NPs can be divided into two categories: (1) computationally efficient (sub-quadratic complexity) but poor performance and (2) computationally expensive (quadratic complexity) but good performance. Early NP variants were especially computationally efficient, requiring only linear computation in the number of context datapoints but suffered from underfitting and as a result, overall poor performance. Tackling underfitting, recent state-of-the-art methods such as TNPs have proposed to use self-attention mechanisms. However, these state-of-the-art methods are computationally expensive in that they require quadratic computation in the number of context datapoints per query. The quadratic computation makes the method inapplicable in settings with limited resources.

In this work, we introduce Efficient Queries Transformer Neural Processes (EQTNPs), a NP variant which has a per target datapoint query complexity that is linear in the number of context datapoints. Our experiments show that EQTNPs achieve results competitive with the state-of-the-art methods.

2 Background

2.1 Meta-Learning for Predictive Uncertainty Estimation

In meta-learning, models are trained on a distribution of tasks $\Omega(\mathcal{T})$. During each meta-training iteration, a batch of B tasks $\mathbf{T} = \{\mathcal{T}_i\}_{i=1}^{B}$ is sampled from a task distribution $\Omega(\mathcal{T})$. A task $\mathcal{T}_i$ is a tuple ($\mathcal{X}$, $\mathcal{Y}$, $\mathcal{L}$, q), where $\mathcal{X}$ is the input space, $\mathcal{Y}$ is the output space, $\mathcal{L}$ is the task-specific

[*]Correspondence to: Leo Feng <leo.feng@mila.quebec>

Method	Computational Complexity (Big-O) Training Step	Evaluation Condition	Query
CNP (Garnelo et al., 2018a)	$N + M$	N	M
CANP (Kim et al., 2019)	$N^2 + NM$	N^2	NM
NP (Garnelo et al., 2018b)	$N + M$	N	M
ANP (Kim et al., 2019)	$N^2 + NM$	N^2	NM
BNP (Lee et al., 2020)	$N + M$	N	M
BANP (Lee et al., 2020)	$N^2 + NM$	N^2	NM
TNP-D (Nguyen & Grover, 2022)	$(N + M)^2$	—	$(N + M)^2$
EQTNP (Ours)	$N^2 + NM$	N^2	NM

Table 1: Computational Complexity in Big-O notation of the model with respect to the number of context datapoints (N) and the number of target datapoints per batch (M). It is important that the cost of performing the query step is low.

loss function, and $q(x, y)$ is a distribution over data points. During each meta-training iteration, for each $\mathcal{T}_i \in \mathbf{T}$, we sample from $q_{\mathcal{T}_i}$: a context dataset $\mathcal{D}_i^{\text{context}} = \{(x, y)^{i,j}\}_{j=1}^{N}$ and a target dataset $\mathcal{D}_i^{\text{target}} = \{(x, y)^{i,j}\}_{j=1}^{M}$, where N and M are the fixed number of context and target datapoints respectively. The context data is used to perform an update to the model f such that the type of update differs depending on the model. In Neural Processes, these updates refer to its conditioning step where embeddings of the dataset are computed. Afterwards, the update is evaluated on the target data and the update rule is adjusted. In this setting, we use a neural network to model the probabilistic predictive distribution $p_\theta(y|x, \mathcal{D}^{\text{context}})$ where θ are the parameters of the NN.

2.2 Neural Processes

Neural Processes (NPs) are a class of models that define an infinite family of conditional distributions where one can condition on arbitrary number of context datapoints (labelled datapoints) and make predictions for an arbitrary number of target datapoints (unlabelled datapoints), while preserving invariance in the ordering of the contexts. Several NP models have proposed to model it as follows:

$$p(y|x, \mathcal{D}_{\text{context}}) := p(y|x, r_C) \tag{1}$$

with $r_C := \text{Agg}(\mathcal{D}_{\text{context}})$ where Agg is a deterministic function varying across the variants that aggeregates $\mathcal{D}_{\text{context}}$ into a finite represention with permutation invariance in the context dataset. These NPs aim to maximise to likelihood of the target dataset given the context dataset. Conditional Neural Processes (CNPs) (Garnelo et al., 2018a), proposes an aggregator using DEEPSETs (Zaheer et al., 2017) and Conditional Attentive Neural Processes (CANPs) (Kim et al., 2019) (a deterministic variant of ANPs without the latent path) proposes an aggreagator using a single self-attention layer. In these NP variants, for a new task, we first (in a conditioning step) compute the embedding of the context dataset r_C. Afterwards, in a separate querying step, the model makes predictions for target datapoints.

Neural Processes have several desirable properties (1) **Scalability**: NPs generates predictions more computationally efficiently than Gaussian Processes which scale cubically with the size of the context dataset. In Table 1, we show the complexity of NP variants relative to the size of the context dataset. It suffices to perform the conditioning step once; however, for each prediction, the querying step needs to be performed independently. As such, the complexity of the query step is the most important when considering the model's efficiency. (2) **Flexibility**: NPs can condition on an arbitrary number of context datapoints and make predictions on an arbitrary number of target datapoints, and (3) **Permutation invariance**: the predictions are order invariant in the context datapoints.

2.2.1 Transformer Neural Processes

Transformer Neural Processes (TNPs) (Nguyen & Grover, 2022) comprises of a transformer-like architecture that takes both the context dataset and the target dataset as input. However, unlike previous NP variants that model the predictive distribution of target datapoints independently, i.e., $p(y_i|x_i, \mathcal{D}_{\text{context}})$, TNPs propose to model the predictive distribution of all target datapoints in conjunction $p(y_{1,\ldots,M}|x_{1,\ldots,M}, D_{\text{context}})$ as follows:

$$p(y_{1,\ldots,M}|x_{1,\ldots,M}, D_{\text{context}}) := p(y_{1,\ldots,M}|r_{C,T}) \tag{2}$$

with $r_{C,T} := \text{Agg}(x_C, y_C, x_T)$ where Agg comprises of multiple Self-Attention layers with a masking mechanism such that the context datapoints and target datapoints only attend to the context datapoints. The output of the aggregator $r_{C,T}$ that corresponds to the target datapoints embeddings are passed to a predictor model for prediction.

In contrast to previous NP variants, TNPs computes its embeddings with both the context dataset and target dataset. Thus, there is no conditioning step for TNPs. Instead, when making queries for a batch of target datapoints, the self-attention on both context and target datapoints must be re-computed, requiring quadratic computation $O((N+M)^2)$ in both the number of context datapoints N and number of target datapoints M. Importantly, this is quadratic even in very small batch query scenario. For example, if $M = 1$, then the prediction is still quadratic $O((N+1)^2)$. The quadratic complexity of a query regardless of the size of the target batch makes the model computationally costly for deployment.

3 Methodology

In this section, we introduce Efficient Queries Transformer Neural Processes (EQTNPs), an NP variant that achieves competitive results while being more computationally efficient.

3.1 Reducing the querying complexity of TNP

Due to passing both the context and query dataset through a self-attention mechanism, for TNP, each prediction requires a quadratic amount of computation in the number of context datapoints regardless of the number of target datapoints. We propose to compute an embedding for each self-attention layer encompassing the information of the context dataset. This way, we can perform cross-attention to retrieve information from the context dataset for predictions. Unlike in TNP whose queries require quadratic computation $O((N+M)^2))$ in both the number of context and target datapoints, the queries by EQTNPs can instead be made in $O(NM)$. Importantly, this means the model has a per target datapoint query complexity that is linear in the number of context datapoints, greatly benefiting the settings with small batch queries. For example, if $M = 1$, then TNP requires $O(N^2)$ while EQTNP requires only $O(N)$. Figure 1 describes how the architecture of TNP is modified to give a more efficient variant that we name Efficient Queries TNPs (EQTNPs).

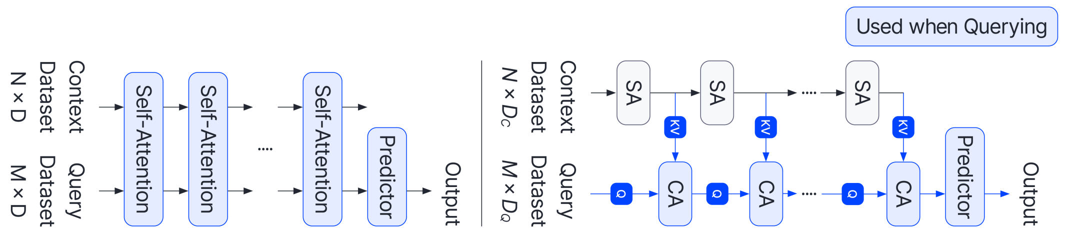

Figure 1: Model architecture for TNPs (left) and EQTNPs (right). The blue highlights the parts of the architecture used when querying. SA refers to Self-Attention and CA refers to CrossAttention.

Conditioning Phase: In the conditioning phase, we compute the embeddings representative of the context dataset

$$\begin{aligned}\text{CEMB}_0 &= \text{CONTEXT}\\ \text{CEMB}_i &= \text{SelfAttention}(\text{CEMB}_{i-1})\end{aligned}$$

Since the number of context datapoints is $O(N)$, the computational cost per SelfAttention layer is $O(N^2)$. Let K be the number of self-attention layers. Then the embeddings $\{\text{CEMB}_1, \ldots, \text{CEMB}_K\}$ encode the model's knowledge of the context dataset.

Querying Phase: When making predictions, we make predictions by retrieving information from the context dataset, specifically from $\{\text{CEMB}_1, \ldots, \text{CEMB}_K\}$, via multiple cross-attention layers.

$$\begin{aligned}\mathrm{QEMB}_0 &= \mathrm{QUERY}\\ \mathrm{QEMB}_i &= \mathrm{CrossAttention}(\mathrm{QEMB}_{i-1}, \mathrm{CEMB}_i)\\ \mathrm{Output} &= \mathrm{Predictor}(\mathrm{QEMB}_\mathrm{K})\end{aligned}$$

Each CrossAttention layer is linear in CEMB. Since there is an embedding per datapoint, the computational cost is $O(NM)$. The multiple cross-attention layers that retrieve information from the context dataset embeddings allows the predictor to retrieve higher-order information just like in TNPs.

4 Experiments

We evaluate EQTNPs on the meta-regression and image completion setting. These experiment settings have been used extensively to benchmark NP models in prior works (Garnelo et al., 2018a; Kim et al., 2019; Lee et al., 2020; Nguyen & Grover, 2022).

In this experiments section, we focus on comparisons with the well-established vanilla versions of NP models. Specifically, for the case of TNPs, we compare with TNP-D. Due to its masking mechanism and architecture, TNP-D models the probabilistic predictive distribution identical to that of previous NP variants. As such, it offers a fair comparison to prior NP works. In contrast, TNP-ND models the predictive distribution differently. This choice of modelling by TNP-ND benefits settings where the predictive uncertainty of all target datapoints are evaluated in conjunction such as in image completion settings. However, this does not benefit and can instead be detrimental to settings where the uncertainty of individual datapoints is evaluated independently such as in multi-armed bandits or bayesian optimization. For a fair comparison of TNP-ND with prior works, it would require modifying the existing NP methods into ND variants (e.g., CNP-ND, CANP-ND, and BNP-ND). However, this is outside the scope of our work.

4.1 1-D Regression

In this experiment, the model observes a set of N context points pertaining from an unknown function f. The model is expected to make predictions for a set of M target datapoints that comes from the same function. In each training epoch, $B = 16$ functions are sampled from a GP prior with an RBF kernel $f_i \sim GP(m, k)$ where $m(x) = 0$ and $k(x, x') = \sigma_f^2 \exp\left(-\frac{(x-x')^2}{2l^2}\right)$. The hyperparameters l and σ_f are randomly sampled per task. The hyperparameters are sampled as follows: $l \sim \mathcal{U}[0.6, 1.0)$, $\sigma_f \sim \mathcal{U}[0.1, 1.0)$, $N \sim \mathcal{U}[3, 47)$, and $M \sim \mathcal{U}[3, 50 - N)$.

Method	RBF	Matern 5/2
CNP	0.26 ± 0.02	0.04 ± 0.02
CANP	0.79 ± 0.00	0.62 ± 0.00
NP	0.27 ± 0.01	0.07 ± 0.01
ANP	0.81 ± 0.00	0.63 ± 0.00
BNP	0.38 ± 0.02	0.18 ± 0.02
BANP	0.82 ± 0.01	0.66 ± 0.00
TNP-D	1.39 ± 0.00	0.95 ± 0.01
EQTNP	1.32 ± 0.01	0.92 ± 0.01

Table 2: 1-D Meta-Regression Experiments with log-likelihood metric (higher is better). All experiments are run with 5 seeds.

At test-time, the models are evaluated on unseen functions sampled from GPs with RBF and Matern $5/2$ kernels. Specifically, the methods are evaluated according to the log-likelihood of the target datapoints. The number of context datapoints N and the number of evaluation points M are sampled according to the same distribution as in training.

Results: As shown in Table 2, EQTNPs achieve results comparable to that of TNPs.

4.2 Image Completion

In this problem, the model observes a subset of pixel values of an image and predicts the remaining pixels of the image. As a result, the problem can be perceived as a 2-D meta-regression problem in which x is the coordinates of a pixel and y is the value of the pixel. The image can be interpreted as a unique function in itself (Garnelo et al., 2018b). For these experiments, the x values are rescaled to [-1, 1] and the y values are rescaled to $[-0.5, 0.5]$. Randomly selected subsets of pixels are selected as context datapoints and target datapoints. For these experiments, we consider two datasets: EMNIST (Cohen et al., 2017) and CelebA (Liu et al., 2015).

EMNIST comprises of black and white images of handwritten letters of 32×32 resolution. In total, 10 classes are used for training, $N \sim \mathcal{U}[3, 197)$ context datapoints are sampled, and $M \sim \mathcal{U}[3, 200-N)$ target datapoints are sampled. CelebA comprises of colored images of celebrity faces that are down-sampled to 32×32 and $N \sim \mathcal{U}[3, 197)$ and $M \sim \mathcal{U}[3, 200 - N)$.

Results: As shown in Table 3, EQTNPs performs comparably with TNPs.

Method	CelebA	EMNIST	
	32x32	Seen (0-9)	Unseen (10-46)
CNP	2.15 ± 0.01	0.73 ± 0.00	0.49 ± 0.01
CANP	2.66 ± 0.01	0.94 ± 0.01	0.82 ± 0.01
NP	2.48 ± 0.02	0.79 ± 0.01	0.59 ± 0.01
ANP	2.90 ± 0.00	0.98 ± 0.00	0.89 ± 0.00
BNP	2.76 ± 0.01	0.88 ± 0.01	0.73 ± 0.01
BANP	3.09 ± 0.00	1.01 ± 0.00	0.94 ± 0.00
TNP-D	3.89 ± 0.01	1.46 ± 0.01	1.31 ± 0.00
EQTNP	3.91 ± 0.10	1.44 ± 0.00	1.27 ± 0.00

Table 3: Image Completion Experiments. Each method is evaluated with 5 different seeds according to the log-likelihood (higher is better).

5 Related Work

The first NP model was Conditional Neural Processes (CNPs) (Garnelo et al., 2018a). CNPs encode the context dataset by encoding the context datapoints via a deep sets encoder. NP (Garnelo et al., 2018b) proposes to use a global latent variable to encode functional stochasticity. Later, other variants proposed to build in translational equivariance (Gordon et al., 2019), attention to tackle underfitting (Kim et al., 2019), transformer attention (Nguyen & Grover, 2022), bootstrapping (Lee et al., 2020), modeling predictive uncertainty (Bruinsma et al., 2020), and tackling sequence data (Singh et al., 2019; Willi et al., 2019). From a practical perspective, NPs have also been applied to a large range of problems, including modeling sequences of stochastic processes (Singh et al., 2019), sequential decision making settings (Galashov et al., 2019), and modeling stochastic physics fields (Holderrieth et al., 2021). Mathieu et al. (2021) propose a contrastive learning framework to learn better representations for NPs.

6 Conclusion and Future Work

Prior NP methods have generally been split between (1) computationally efficient variants of Neural Processes but poor overall performance and (2) computationally expensive attention-based variants of Neural Processes that perform well. In this work, we propose Efficient Queries Transformer Neural Processes (EQTNPs). EQTNPs fix the slow querying of the previous state-of-the-art method (TNPs) by proposing to apply self-attention only on the context dataset and retrieve information to the target datapoints via cross-attention mechanisms. Each query datapoint only requires linear computation in the number of context datapoints. In the small batch query scenario this is important since in contrast TNPs will require quadratic computation in both the number of context and target datapoints. In our experiments, we show that EQTNPs achieve results comparable to state-of-the-art. However, both EQTNPs and TNPs require quadratic computation to perform the conditioning step, limiting their scalability. As such, an interesting future direction would be to develop methods that are competitive while requiring linear computation in the conditioning step.

References

Wessel Bruinsma, James Requeima, Andrew YK Foong, Jonathan Gordon, and Richard E Turner. The gaussian neural process. In *Third Symposium on Advances in Approximate Bayesian Inference*, 2020.

Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *2017 international joint conference on neural networks (IJCNN)*, pp. 2921–2926. IEEE, 2017.

Alexandre Galashov, Jonathan Schwarz, Hyunjik Kim, Marta Garnelo, David Saxton, Pushmeet Kohli, SM Eslami, and Yee Whye Teh. Meta-learning surrogate models for sequential decision making. *arXiv preprint arXiv:1903.11907*, 2019.

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

Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J Rezende, SM Eslami, and Yee Whye Teh. Neural processes. *arXiv preprint arXiv:1807.01622*, 2018b.

Jonathan Gordon, Wessel P Bruinsma, Andrew YK Foong, James Requeima, Yann Dubois, and Richard E Turner. Convolutional conditional neural processes. In *International Conference on Learning Representations*, 2019.

Peter Holderrieth, Michael J Hutchinson, and Yee Whye Teh. Equivariant learning of stochastic fields: Gaussian processes and steerable conditional neural processes. In *International Conference on Machine Learning*, pp. 4297–4307. PMLR, 2021.

Hyunjik Kim, Andriy Mnih, Jonathan Schwarz, Marta Garnelo, Ali Eslami, Dan Rosenbaum, Oriol Vinyals, and Yee Whye Teh. Attentive neural processes. 2019.

Juho Lee, Yoonho Lee, Jungtaek Kim, Eunho Yang, Sung Ju Hwang, and Yee Whye Teh. Bootstrapping neural processes. *Advances in neural information processing systems*, 33:6606–6615, 2020.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Emile Mathieu, Adam Foster, and Yee Teh. On contrastive representations of stochastic processes. *Advances in Neural Information Processing Systems*, 34:28823–28835, 2021.

Tung Nguyen and Aditya Grover. Transformer neural processes: Uncertainty-aware meta learning via sequence modeling. In *International Conference on Machine Learning*, pp. 16569–16594. PMLR, 2022.

Gautam Singh, Jaesik Yoon, Youngsung Son, and Sungjin Ahn. Sequential neural processes. *Advances in Neural Information Processing Systems*, 32, 2019.

Timon Willi, Jonathan Masci, Jürgen Schmidhuber, and Christian Osendorfer. Recurrent neural processes. *arXiv preprint arXiv:1906.05915*, 2019.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.