Global Perception Based Autoregressive Neural Processes

Jinyang Tai*
School of Computer Engineering and Science, Shanghai University, Shanghai, China
a1203146411@gmail.com

Abstract

Increasingly, autoregressive approaches are being used to serialize observed variables based on specific criteria. The Neural Processes (NPs) model variable distribution as a continuous function and provide quick solutions for different tasks using a meta-learning framework. This paper proposes an autoregressive-based framework for NPs, based on their autoregressive properties. This framework leverages the autoregressive stacking effects of various variables to enhance the representation of the latent distribution, concurrently refining local and global relationships within the positional representation through the use of a sliding window mechanism. Autoregression improves function approximations in a stacked fashion, thereby raising the upper bound of the optimization. We have designated this framework as Autoregressive Neural Processes (AENPs) or Conditional Autoregressive Neural Processes (CAENPs). Traditional NP models and their variants aim to capture relationships between the context sample points, without addressing either local or global considerations. Specifically, we capture contextual relationships in the deterministic path and introduce sliding window attention and global attention to reconcile local and global relationships in the context sample points. Autoregressive constraints exist between multiple latent variables in the latent paths, thus building a complex global structure that allows our model to learn complex distributions. Finally, we demonstrate the effectiveness of the NPs or CFANPs models for 1D data, Bayesian optimization, and 2D data.

1. Introduction

Neural processes (NPs) are different from traditional stochastic processes such as Gaussian processes (GPs) [15, 20] in that GPs are difficult to use due to the necessity of selecting an appropriate kernel function. In practice, the selection of the right kernel function for different distributions is often a matter of specialized knowledge and experience. In addition, different kernel functions can result in different computational complexity and model accuracy. In contrast, NPs provide a more flexible approach to data modeling by eliminating the need to select a specific kernel function. This approach can lead to more efficient and accurate predictions without requiring extensive knowledge of the data distribution. NPs [5] is a new model for the combination of parametric [19, 24] Neural Networks and Stochastic Processes. In the given data, NPs are modeled as a new class of Stochastic Processes with a flexible approach to the original data distribution. In particular, NPs deal with non-trivial function distributions for which it is difficult to find a suitable prior representation of the GPs function. In this sense, GPs perform data modeling by driving a prior distribution, while NPs perform the same task by driving the data. While maintaining the flexible properties of the model during the training process, the NPs are implemented with a meta-learning framework so that they can be quickly adapted to new functional tasks.

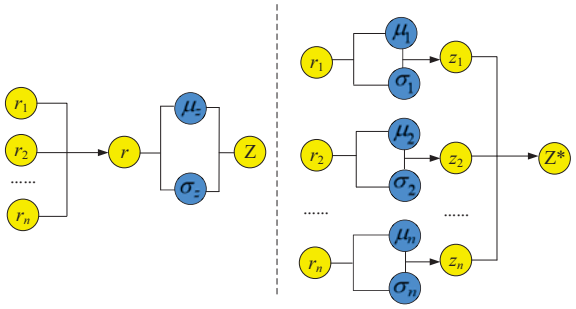

Figure 1. The NPs model [5] (left) represents the context sample points denoted as r using an average value aggregation. In this model, the latent distribution is represented by the mean μ_z and variance σ_z normal distribution of a single variable. On the other model, the BANPs model [14] (right) represents the latent distribution of each context sample point as a single variable with a mean μ_n and variance σ_n normal distribution.

Despite the increasing attention in Stochastic Processes methods, the NPs model still has several limitations that hinder its development [13]. One of the main problems

*Corresponding author

is that the NPs encoder maps context sample points to a fixed-length latent representation, while the decoder maps the above latent representation and the target sample point input to the target sample point output. However, the encoder achieves a fixed-length representation of the context sample points through an average aggregation module that assigns equal weight to each context sample point. As a result, it is difficult for the decoder to determine which context sample points provide relevant information for predicting the target sample points, leading to underfitting [11]. Another limitation of NPs is their susceptibility to noise in real data, which causes the sampled context sample points to contain more interference information, resulting in deviations in the predicted location of the target sample points. This is because NPs fail to capture the embedding relationships between the context sample points [12]. Furthermore, the distribution in the real world is complex and constantly changing, making it difficult for the NPs model to express its latent distribution by a single latent variable [6]. Existing NPs and their variants represent the latent distribution by stacking single or multiple latent variables [14, 13, 5]. However, this approach has limitations that need to be addressed to further improve the effectiveness and accuracy of the NPs model.

In this paper, we introduce two frameworks, Autoregressive Neural Processes (AENPs) and Conditional Autoregressive Neural Processes (CAENPs), which jointly model the global structure by using multiple latent Gaussian variables in an autoregressive manner. Our approach provides better modeling performance, especially for complex distribution modeling processes. While existing NPs and their variants typically use self-attention implementations as the deterministic path encoding process, this can be computationally expensive and may not effectively capture the relationships between the local context sample points. To overcome this, we propose a combination of sliding window attention and global attention that is implemented autoregressively, allowing us to capture relationships between the global and local context sample points while avoiding high computational cost. We demonstrate the advantages of our approach on 1D and 2D datasets, as well as in Bayesian optimization. Our approach improves the latent distribution performance of multiple latent variables, leading to better modeling performance for complex distribution modeling processes.

2. Background

2.1. Neural Processes

Given a sequence of regression tasks represented as $\mathcal{D} = (X, Y)$, where $X = \{x_i\}_{i=1}^{n} \in \mathbb{R}^d$ is the observation set and $Y = \{y_i\}_{i=1}^{n} \in \mathbb{R}^d$ is the corresponding label set. we can define a context sample point set as $(X_C, Y_C) := \{(x_i, y_i)\}_{i \in C}^{n}$, where C is the set of indices for the context sample points. We also have a target set $X_T := \{x_i\}_{i \in T}^{m}$, where T is the set of indices for the target sample points. The goal of the NPs model is to learn a function f that maps the input observation x to the output label y given the context sample points set X_C, Y_C. In other words, we want to learn the conditional distribution $p(Y_T|X_T, X_C, Y_C)$. To achieve this, the NPs model maps each a pair of context sample point (x_C, y_C) to a corresponding representation r_C using a Multi-Layer Perceptron (MLP) [27]. The above processes can be expressed as follows:

$$r_C := MLP\left(x_C, y_C\right) \tag{1}$$

In practice, each context sample point (x_C, y_C) is mapped by the MLP function to r_C, which is used to maintain dimensionality consistency, and further processed to take an average value denoted as r (as shown in Figure 2). The latent distribution in the NPs model is represented by a single variable Z, for which the Gaussianisation factor is decomposed into a mean μ_z and variance σ_z. Meanwhile, the above latent distribution Z plays the role of a global variable in the NPs model, acting as a representation of the uncertainty function. So, the latent path for Z is derived from $s_C := s\left(\boldsymbol{x}_C, \boldsymbol{y}_C\right)$. We have incorporated the latent distribution into the equation as follows.

$$p\left(y_T \mid x_T, x_C, y_C\right) := \int p\left(y_T \mid x_T, r_C, Z\right) q\left(Z \mid s_C\right) dZ \tag{2}$$

where $Z = \mathcal{N}(\mu_z, \sigma_z)$. The NPs model infers the target sample point y_T by the likelihood method. The encoder and decoder parameters in the NPs model are adjusted by maximizing the ELBO as follows.

$$\begin{aligned}\log p\left(y_T \mid x_T, x_C, y_C\right) \geq \mathbb{E}_{q(Z|s_T)}\left[\log p\left(y_T \mid x_T, r_C, Z\right)\right] \\ -D_{\mathrm{KL}}\left(q\left(Z \mid s_T\right) \| q\left(Z \mid s_C\right)\right)\end{aligned} \tag{3}$$

The primary function of NPs is to acquire the skill of target reconstruction, with Kullback-Leibler (KL) divergence regularization serving to facilitate approximation between the context and target sample points.

2.2. Autoregressive

For predicting a sequence of target sample points, it is customary to organize the samples following specific guidelines in order to establish a sequential relationship. For example, a structured sequence $x_1, x_2, \cdots, x_n$ can be represented using a p-order autoregressive model (AR). This model formulates x_t as a function involving a linear combination of the preceding p terms within the series along

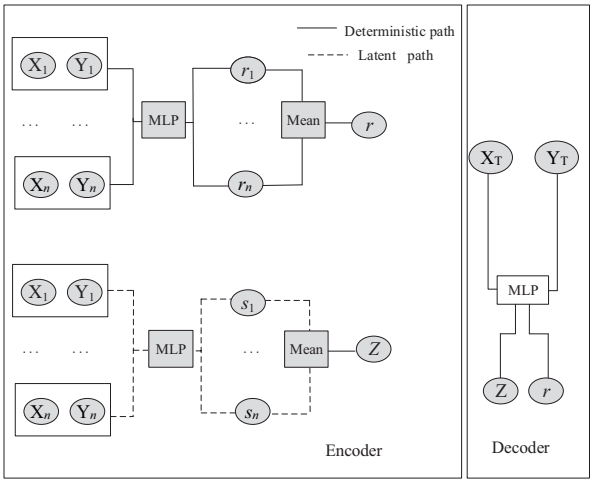

Figure 2. NPs model architecture. The left side indicates the process of encoding the context sample points. The right side indicates the process of decoding the context sample points.

with an associated error term. This methodology inherently captures the probability distribution and is commonly employed with structured data types, such as dimensions, features, temporal instances, and more.

$$y_t = \Phi_0 + \Phi_{1y_{t-1}} + \Phi_{2y_{t-2}} + \ldots + \Phi_p y_{t-p} + \varepsilon_t \quad (4)$$

where Φ_0 is a constant term, $\Phi_1, \cdots, \Phi_p$ are model parameters, and ε_t is noise with mean 0 and variance σ.

The autoregressive approach provides an implicit modeling of the likelihood function and is typically applied to regular data, such as dimensions, features, or time, among others. In this paper, the authors divide context sample points sampled according to NPs into different numbers to construct different dimensions or number levels for ranking purposes. For example, given n context sample points represented as $(x_i, y_i)_{i \in C}^{n}$, they can be divided into different dimensions such as $\{(x_1, y_1)\}$, $\{(x_1, y_1), (x_2, y_2)\}$, $\cdots$, $\{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\}$. These divisions are used to establish the dependencies of the different levels of context sample points and to establish the joint probability distribution, which is then used to infer the latent distribution of the target sample points.

3. Related Work

Garnello et al [4] identified several challenges within Gaussian Processes (GPs), including substantial computational demands and the intricacy of selecting an appropriate prior. In the present landscape, widely adopted neural networks achieve precise distributional representations through gradient descent optimizationt[7, 16]. The authors combine the two to produce Conditional Neural Processes (CNPs), which can be extended to large data sets only by observing a few context sample points. CNPs use fixed-dimension input in the data encoding process, resulting in a lack of flexibility in output. On this basis, NPs[5] enrich their encoder representation by introducing latent variables. The aggregation approach of encoders in NPs causes the context sample points to underfit. Attentive Neural Processes(ANPs)[11] dynamically assign power to context sample points in this aggregation with multi-headed attention. In practice, NPs have serious flaws in sequential decision making, combined with the temporal function in the current transformer to produce Transformer Neural Processes (TNPs)[17]. Convolutional Conditional Neural Processes (CCNPs) solve NPs to model the translational equivariance in the data (ie, time series, spatial data, text data). This model extends data processing from finite dimensions to infinite dimensions.

Autoregressive models have witnessed noteworthy advancements within the realm of deep learning research. In recent investigations [21, 22], scholars have treated individual image pixels as sequences, wherein each pixel's value relies on the value of its preceding pixel. This dependency structure permits prediction via a pixel-by-pixel strategy. Termed autoregressive modeling, this methodology leverages the predicted value at a given point as input for forecasting the subsequent value. An additional study [9] amalgamated an order-independent autoregressive model with a discrete absorption diffusion model, yielding comparable performance with fewer iterative steps compared to conventional diffusion models. Notably, this composite model has showcased enhanced prowess in image compression, surpassing the capabilities of the standard diffusion model.

4. Autoregressive Neural Processes

The NPs [5] model a continuous function probability distribution conditional on partial observation of the context sample points, which is constrained by the latent distribution. In this section, we enhance the representation of the latent distribution by replacing the representation of a single variable normal distribution with Autoregressive in the NPs model. The new model generated above is referred to as Autoregressive Neural Processes (AENPs) or Conditional Autoregressive Neural Processes (CAENPs).

4.1. The Latent Distribution of Autoregressive

The encoder for AENPs has the same two components as the NPs model: the deterministic path and the latent path. The latent path is used to describe the latent distribution of context sample points. The model is combined with Stochastic Processes to produce a latent distribution for each context sample point under this path. NPs and variants models [14, 13, 5] use single or multiple Gaussian distributions as a latent representation of each context sample

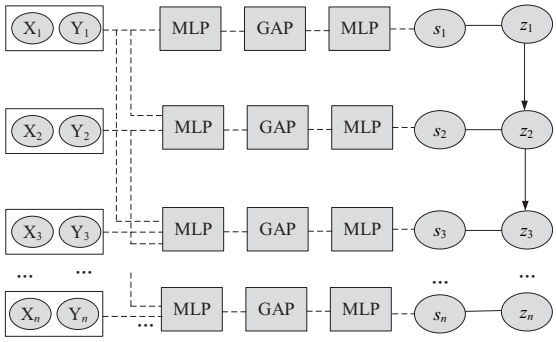

Figure 3. The latent distribution representation of the AENPs model. The context sample points complete the mapping relationship with the help of two MLPs and one GAP combination. At the same time, the ability to constrain between different latent variables is realized by autoregressive methods.

point.

We find that the representation of the latent variable Z in the latent distribution $p(s_C \mid Z)$ depends on the outcome of observing the context sample points (x_C, y_C) mapping s_C. The NP averages the results of the mapping s_C and then expresses them as a single variable normal distribution, resulting in a weak expression of its latent distribution [11, 5] (as shown in Figure 2). The mapping context sample points $s_C = (s_1, s_2, \cdots, s_n)$ are treated in the AENPs model to avoid the problem of a single representation of the latent distribution by combining their different levels with the latent variable $Z = (z_1, z_2, \cdots, z_n)$ (as shown in Figure 3). We incorporate autoregression to enhance the expressiveness of the latent distribution. The conditional probability based on the autoregressive constraint on the latent variable $z_1, z_2, \cdots, z_n$ is as follows.

$$p(Z) = \prod_{i=1}^{n} p\left(z_i \mid z_1, \cdots, z_{n-1}\right) = \prod_{n=1}^{n} p\left(z_n \mid z_{1:n-1}\right) \tag{5}$$

Finally, the AENPs model goal is predicted by x_C, y_C, x_T referring to the corresponding target sample points denoted as $p(y_T \mid x_C, y_C, x_T)$.

The specific implementation detail of the AENPs model is shown in Figure 3. We build a module consisting of two MLPs stacked with Global Average Pooling (GAP) [18]. This module processes the ith context sample point to form a new continuous input by combining the previous i-1 context sample points. The latent variable z_i representation of the context sample points (x_C, y_C) at each different level is through a Gaussian distribution. The process for the i-th context sample point is formulated as follows.

$$s_i = \text{MLP(GAP(MLP}((x_i, y_i), (x_{i-1}, y_{i-1}), \cdots, (x_1, y_1))) \tag{6}$$

Also, the formula for calculating the latent distribution of the i-th context sample point is calculated as follows.

$$p_\psi\left(z_i \mid s_i\right) = \mathcal{N}\left(\mu_{z_i}, \sigma_{z_i}^2\right) \tag{7}$$

This latent distribution between the different latent variables of the autoregressive constraint is calculated as follows.

$$z_i = \mu_i + \sigma_i \odot z_{i-1} \tag{8}$$

The process of training AENPs is to complete the computation of the conditional posterior distribution $q_\phi\left(z_n \mid z_{n-1}, x_C, y_C, x_T, y_T\right)$. The goal of the above equation is to calculate as follows.

$$s_n = \text{MLP(GAP(MLP}(x_C, y_C, x_T, y_T))) \tag{9}$$

$$q_\phi\left(z_n \mid z_{n-1}, s_n\right) = \mathcal{N}\left(\mu_{z_n}, \sigma_{\mathbf{z_n}}^2\right) \tag{10}$$

As in Equation (3), AENPs use KL divergence to measure the matching result between the true distribution and the approximate distribution as follows.

$$D_{KL} = \sum_{i=2}^{n} E_{q_\phi(z_n \mid x_C, y_C, x_T, y_T)}[D_{KL}[q_\phi(z_n \mid z_{n-1}, x_C, y_C, x_T, y_T) \| p_\psi(z_n \mid z_{n-1}, x_C, y_C, x_T)] + D_{KL}[q_\phi(z_1 \mid x_C, y_C, x_T, y_T) \| p_\psi(z_1 \mid x_C, y_C, x_T)] \tag{11}$$

The D_{KL} section and the reconstruction section above are as follows.

$$L \geq \mathbb{E}_{z_n \sim q_\phi(z_n \mid x_T, y_T)}[\log p_\theta(y_T \mid z_n, x_T, x_C, y_C)] - \beta \cdot D_{KL} \tag{12}$$

where β denotes hyperparameters to balance the KL and reconstructed parts. At the same time, the presence of β better captures the uncertainty in the model.

4.2. Conditional Dependence Between the Latent Distribution of Autoregressive

The AENPs model increases model representation capability by addressing the dependencies between multiple latent variables with an autoregressive approach. This approach ignores the different dependencies between the different context sample points corresponding to the latent distribution on the final representation distribution.

We introduce the gating mechanism from the (Long Short-Term Memory) LSTM [1] model between the different latent variables as shown in Figure 4. We call this model

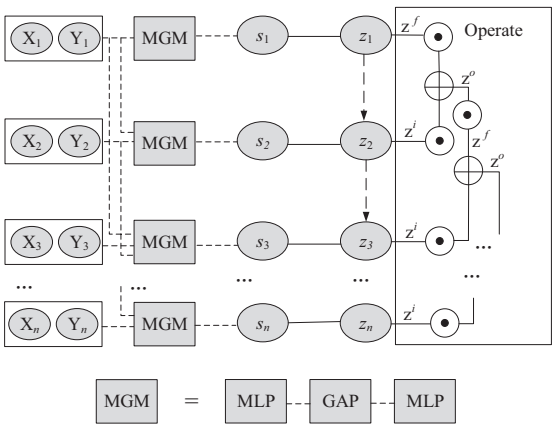

Figure 4. The latent distribution representation of the CAENPs model. The context sample points complete the mapping relationship with the help of two MLP and GAP combinations. The association between the different latent variables is achieved through forgetting gates, selection gates, and output gates, with the ultimate goal of achieving a representation of the latent distribution.

Conditional Autoregression Neural Processes (CAENPs). The CAENPs model and the AENPs model use the same treatment for the context sample point (x_C, y_C) to complete the mapping relationship. The association between the different latent variables $z_1, z_2, \cdots, z_n$ is done through the gating mechanism from the LSTM model. The gate mechanism in the LSTM contains mainly forgetting gates, selection gates, and output gates. The forgetting gate is required to discard the part of the distribution represented by the last latent variable. The selection gate is required to retain the part of the distribution represented by the current latent variable. The output gate is required to output the part of the distribution represented by the latent variable.

The forgetting gate for the latent variable of the ith context sample point is calculated as follows.

$$z^f = \text{sigmoid}(W^f \odot [z_{i-1}, s_i] + b^f) \tag{13}$$

where sigmoid denotes the activation function; b^f denotes a bias; W^f denotes the weight. The selection gate for the latent variable of the ith context sample point is calculated as follows.

$$z^i = \text{sigmoid}(W^i \odot [z_{i-1}, s_i] + b^i) \tag{14}$$

$$\tilde{z_i} = \tanh(W^c \odot [z_{i-1}, s_i] + b^c) \tag{15}$$

where sigmoid and tanh denote the activation function; b^i and b^c denote a bias; W^i and W^c denote the weight . The output gate for the latent variable of the ith context sample point is calculated as follows.

$$z^o = \text{sigmoid}(W^o \odot [z_{i-1}, s_i] + b^o) \tag{16}$$

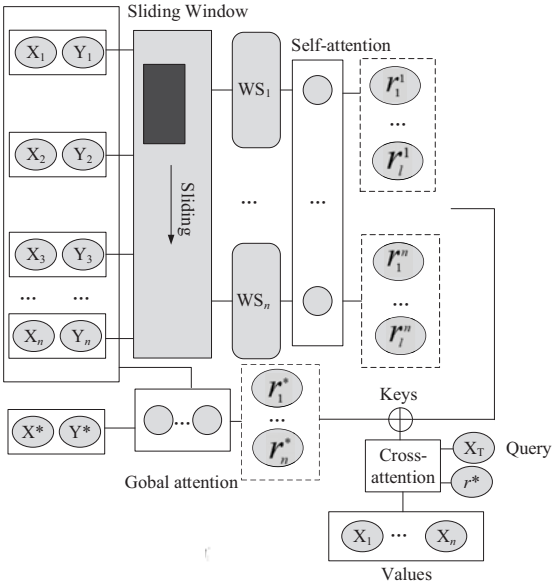

Figure 5. The deterministic path of the AENPs or CAENPs model. Capturing local information about the context sample points by sliding window self-attention. The target sample point x_T is retrieved by distance to the nearest context sample point (x_C, y_C) that is (x^*, y^*). The combination of sliding window attention and global attention replaces the self-attention inside [13].

$$z_i = z^o \odot \tanh(z^f \odot z_{i-1} + z^i \odot \tilde{z_i}) \tag{17}$$

where tanh and sigmoid denote the activation function; b^o denotes a bias; W^o denotes the weight. We inferred the final latent distribution by correlating the different latent variables with each other through a gating mechanism. The CAENPs model and the AENPs model are optimized in the same way by using equation (12).

The CAENPs and AENPs models are associated in an autoregressive way with the different latent variables in the latent path. This autoregressive approach focuses on discovering dependencies between the different latent variables and enhancing their representation of the latent distribution.

4.3. The Deterministic Path

The deterministic path is achieved in the NPs model by taking an average of the context sample points. This approach tends to cause underfitting of the context sample points [5]. [11] solves the underfitting problem in the NPs model by using an attention mechanism.

In the AENPs or CAENPs model, sliding windows[2] are used to capture the relationship between the local context sample points (As shown in Figure 5). The size of the sliding window WS is l and the ith read context sample points in its message is denoted as

$(x_i, y_i), \cdots, (x_{l+i-1}, y_{l+i-1})$. The content of each window is self-attention transformed to achieve $(r_1^i, \cdots, r_l^i)$ mapping relationships. The target sample points x_T are obtained as the nearest sample point from the context sample points (x_C, y_C) according to the Euclidean distance as (x_*, y_*). (x_*, y_*) is transformed with all the context sample points by equation (18) to obtain the mapping relationship $(r_1^*, \cdots, r_l^*)$. Global attention and sliding window attention are represented by summing the weights of the positional relationships into a matrix $A \in \mathbb{R}^{n*n}$[2]. Finally, the relationship between the x_T (Query), A (keys), x_C (values) is represented by the cross-attention mechanism with context sample points r^*.

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \qquad (18)$$

where Q denotes the Query; K denotes the Key; V denotes the Value; $\sqrt{d_k}$ denotes the scaling factor; Attention $K = Q = V$ converted to self-attention.

In the AENPs or CAENPs model, we capture relationships between local context sample points through a sliding window and global attention to capture relationships between the global context sample points. In this paper, we replace the original self-attention with sliding window attention and global attention to reduce the spatial complexity from $\mathcal{O}(n^2)$ to $\mathcal{O}(ln + n)$, in addition to enhancing the capture of the local context sample points relationship.

5. Experimental Result

CAENPs and AENPs are used as new Stochastic Process models to enrich the variants of the family of Neural Processes. We are required to demonstrate the performance of our AENPs and CAENPs models on both 1D and 2D datasets. This paper illustrates the situation from the 1D Gaussian distribution dataset and the 2D dataset (MNIST [5], CelebA [11]).

5.1. 1D Function Regression

As a first experiment, we evaluated the performance of the AENPs and CAENPs models on a 1D regression task. To generate the distribution, we chose a Gaussian kernel function for the 1D function. The context sample points sampled from this distribution were used as training data, and the associated target sample points were used as test data. The goal of the model is to take as input the context sample points $(x, y)_C$, target sample points x_T, and output the predicted values y_T^* corresponding to the ground truth values y_T. The range of values for each data point x in the 1D Gaussian distribution data was restricted to the interval [-3,3].

For this 1D data distribution, we explored the effect of using different ways of implementing the latent distribution

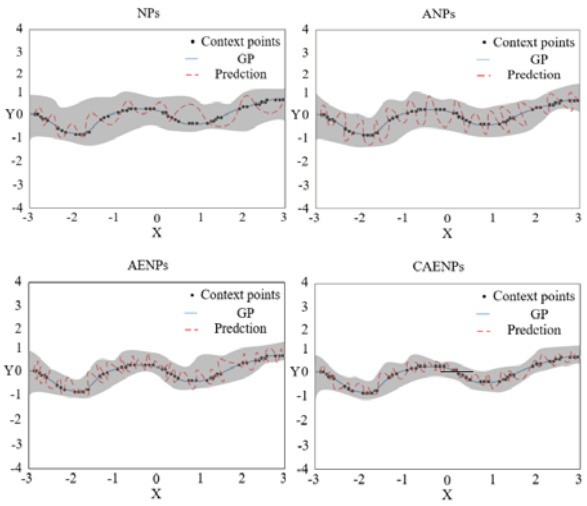

Figure 6. Visualization of NPs, ANPs, AENPs, CAENPs models on 1D Gaussian distribution prediction results. A total of 40 sample points are sampled by all the above methods. The true fit curve for Gaussian distribution is shown in blue, the context point samples is shown in black, and the prediction curve is shown in red.

to predict the target sample distribution in the AENPs and CAENPs. In order to illustrate the superior performance of the AENPs model and CAENPs model, the results of existing popular NPs [5][1], and ANPs [11][2] are required to be compared on 1D Gaussian distribution data (see Figure 6). The optimized loss is shown in Equation (12).

In Figure 6, We provide 1D Gaussian distribution data to represent the visualization of NPs, ANPs, AENPs, and CAENPs. The AENPs or CAENPs model provides superior results among the multiple latent variables in an autoregressive constrained approach. From figure 6, the CAENPs model is better than the AENPs model in terms of the way it constrains between the different latent variables through the gating mechanism.

We are training on 1D Gaussian distribution data using NPs, ANPs, AENPs, and CAENPs that require the minimum value in the Gaussian distribution prior to being extracted with the objective function. From the process, the NPs, ANPs, AENPs, and CAENPs are compared to the original kernel values in 1D Gaussian distribution data. The whole process uses the best simple regret [18], which is a method of discovering the difference between the best observed solution and the global optimal solution in the 1D Gaussian distribution data. The model considers the objective function of the agents of the previous NPs and variants. We use Thompson sampling [25] to extract the waiting function from the agents and actions. We have chosen 100

[1] https://github.com/EmilienDupont/neural-processes
[2] https://github.com/soobinseo/Attentive-Neural-Process

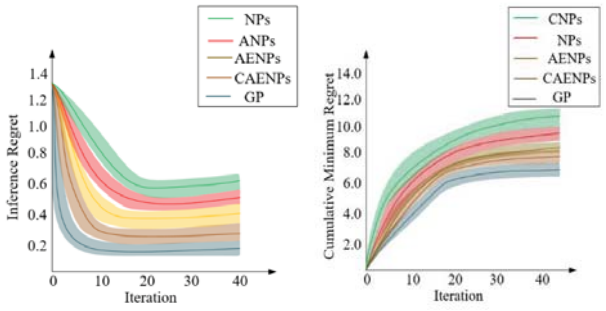

Figure 7. Simple and cumulative regret for Bayesian optimized.

Method	RBF kernels	Matérn 5/2	Periodic
CNPs	0.278 ± 0.003	0.310 ± 0.003	0.652 ± 0.001
NPs	0.282 ± 0.003	0.315 ± 0.003	0.650 ± 0.002
ANPs	0.193± 0.001	0.230 ± 0.000	0.703 ± 0.002
BNPs	0.269± 0.003	0.301 ± 0.003	0.649 ± 0.002
BANPs	0.260± 0.001	0.232 ± 0.001	0.613 ± 0.001
CCNPs	0.254± 0.003	0.228 ± 0.001	0.681± 0.002
TNPs	0.177 ± 0.001	0.222 ± 0.000	0.670 ± 0.009
VNPs	0.162 ± 0.003	0.201 ± 0.000	0.642 ± 0.007
AENPs(Ours)	**0.152 ± 0.003**	**0.217 ± 0.001**	**0.593 ± 0.005**
CAENPs(Ours)	**0.149 ± 0.001**	**0.199 ± 0.003**	**0.524 ± 0.001**

Table 1. We experimented with different kernel functions in 1D Gaussian distribution. The mean and standard deviation of the five runs are reported (MSE measures).

objective functions as the results shown in figure 8. From figure 8, the AENPs or CAENPs result in better predicted minimum values than the NPs, and ANPs.

We require different kernel functions in addition to the Gaussian kernel function data described above to account for the reliability of the AENPs or CAENPs model. We compare NPs[5], CNPs [4][3], ANPs[11], BNPs[13][4], BANPs[14], CCNPs[26][5], TNPs[17][6], VNPs[6][7] in terms of metrics: Mean Square Error (MSE), Calibration Error (CE) to measure performance. Furthermore, we use the data generated by GP data to compare different kernels (RBF kernels, Matérn 5/2, Periodic). From the results in Tables 1 and Tables 2 the AENPs, CAENPs, and other models showed the best results in terms of MSE and CE measures of their model performance. The final results show that the AENPs and CAENPs models are reliable.

5.2. 2D Function Data Images

In addition to the 1D data experiments described above, the AENPs model or CAENPs model requires validation on

[3]https://github.com/stratisMarkou/conditional-neural-processes
[4]https://github.com/juho-lee/bnp
[5]https://github.com/cambridge-mlg/convcnp
[6]https://github.com/tung-nd/TNP-pytorch
[7]https://github.com/ZongyuGuo/Versatile-NP

Method	RBF kernels	Matérn 5/2	Periodic
CNPs	0.078 ± 0.002	0.051 ± 0.000	0.143 ± 0.002
NPs	0.093 ± 0.002	0.056 ± 0.001	0.130 ± 0.007
ANPs	0.085 ± 0.001	0.169 ± 0.001	0.265 ± 0.002
BNPs	0.093 ± 0.003	0.054 ± 0.002	0.115 ± 0.004
BANPs	0.082± 0.001	0.232 ± 0.001	0.613 ± 0.001
CCNPs	0.094± 0.000	0.195 ± 0.001	0.162± 0.003
TNPs	0.048 ± 0.001	0.050 ± 0.001	0.155 ± 0.009
VNPs	**0.045 ± 0.000**	0.049 ± 0.001	0.140 ± 0.005
AENPs(Ours)	0.052 ± 0.002	**0.047 ± 0.002**	**0.137 ± 0.000**
CAENPs(Ours)	0.049 ± 0.003	**0.041 ± 0.000**	**0.122 ± 0.005**

Table 2. We experimented with different kernel functions in 1D Gaussian distribution. The mean and standard deviation of the five runs are reported (CE measures).

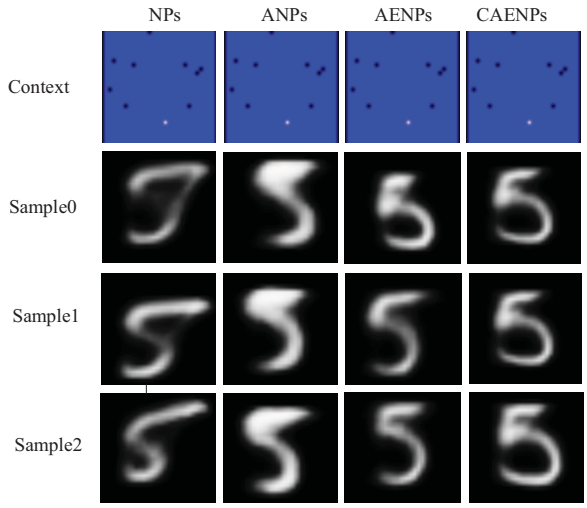

Figure 8. The NPs, ANPs, AENPs, and CAENPs models sampled 10 context sample points to complete the image complementation task. The results of 3 stages are selected for each model for presentation.

2D data to demonstrate its suitability for handling higher dimensional data. In this paper, 2D data is selected from common ordinary image data (AENPs model or CEFNPs covers all 2D data, not just images). The dependencies between pixel values in an image are considered to be the background values x_C and y_C, while predicting the target pixel y_T is considered to be the complete image problem. We compare the NPs[5], CNPs [4], ANPs[11], BNPs[13], BANPs[14], CCNPs[26], and TNPs[17],VNPs[6] on the MNIST dataset [5] and CelebA [13] dataset.

The results of visualizing NPs, ANPs, AENPs, and CAENPs on the MNIST dataset (As shown in Figure 8) and CelebA dataset (As shown in Figure 9). From Figures 8 and 9, we find that the AENPs or CAENPs models predicted better output pixel values than the ANPs and NPs models.

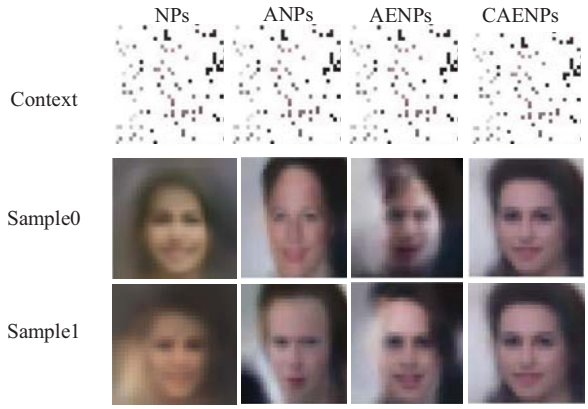

Figure 9. The NPs, ANPs, AENPs, and CAENPs model use a random sampling of 100 context sample points. We use NPs, ANPs, AENPs, and CAENPs methods to output predict image pixels through their learned distributions. Each model selects 2 stage samples for presentation.

The superiority of the AENPs and CAENPs models is analyzed from a quantitative perspective in Figure 8 or 9. The results are evaluated in quantitative form for the complete images. We use the Frechet Inception Distance (FID) [8] metric to measure the quality of the complete image. We have expressed the formula for the above metric as follows.

$$FID = ||\mu_r - \mu_g||_2^2 + Tr(\sum\nolimits_r + \sum\nolimits_g - 2(\sum\nolimits_r \sum\nolimits_g)^{1/2}) \tag{19}$$

where r denotes the ground truth image, g denotes the complete images, and μ denotes the distribution mean. The larger value of FID, the worse the quality of the complete images. On the contrary, the smaller value of FID, the better the quality of the complete images. From Table 7 and 4, it is found that the FID is lowest at 50, 100, 200, and 400 the context sample points for the complete images. The above results imply that the AENPs or CAENPs model is of the highest quality than the NPs, CNPs, ANPs, BNPs, BANPs, CCNPs, TNPs, and VNPs models of the complete images. It is illustrated from the results that constraining the latent variable in an autoregressive manner enhances the representation of the latent distribution. At the same time, the original self-attention was replaced by sliding-window attention and global attention, capturing the role of the local and the global between the context sample points on the deterministic path.

5.3. Ablation Study

In analyzing the performance of the AENPs or CAENPs models, we examined various factors that affect their performance. various influencing factors. Notably, autoregression plays a unique and important role in AENPs or CAENPs models.

Methods	50	100	200	400
	FID	*FID*	*FID*	*FID*
CNPs	90.41	87.23	74.10	63.92
NPs	87.56	79.03	71.88	65.35
ANPs	79.62	72.15	60.84	52.71
BNPs	74.00	63.69	52.42	49.38
BANPs	70.05	64.67	51.69	48.16
CCNPs	78.13	68.40	55.83	51.56
TNPs	67.42	54.51	47.29	39.50
VNPs	65.92	53.42	46.91	37.42
AENPs(Ours)	**58.16**	**43.57**	**34.83**	**19.00**
CAENPs(Ours)	**57.43**	**40.64**	**32.48**	**17.14**

Table 3. The FID result of selecting a different number of pixel points from the images as context sample points to the complete image (MNIST dataset).

Methods	50	100	200	400
	FID	*FID*	*FID*	*FID*
CNPs	94.13	79.04	68.26	51.72
NPs	87.52	76.11	60.34	48.60
ANPs	80.00	73.89	55.91	37.45
BNPs	68.95	61.54	56.07	45.52
BANPs	61.38	64.35	54.20	40.11
CCNPs	74.23	69.08	59.86	50.31
TNPs	63.18	59.24	57.99	40.99
VNPs	68.92	58.13	56.00	38.43
AENPs(Ours)	**60.84**	**55.72**	**41.09**	**31.55**
CAENPs(Ours)	**55.47**	**43.30**	**39.17**	**30.70**

Table 4. The FID result of selecting a different number of pixel points from the images as context sample points to the complete images (CelebA dataset).

Sliding window attention and Global attention. We employ a hybrid approach that incorporates both sliding windows and global paths to establish the deterministic path. In contrast, for the latent path, we leverage the (C)AENPs-NPs and (C)AENPs-VNPs models based on the NPs and VNPs frameworks respectively. The efficacy of these models is assessed using the 2D MNIST and CelebA datasets, and the summarized outcomes are presented in Table 6.

Autoregressive implementation method. The AENPs and CAENPs models implement the autoregressive approach in the form of chain multiplication and gating mechanisms in the LSTM. This paper further compares the results of common Bidirectional Long Short-Term Memory (bi-LSTM) [10] (CAENPs-biLSTM), Gate Recurrent

Method	50	100	200	400
	FID	*FID*	*FID*	*FID*
(C)AENPs-NPs	78.03	47.59	45.31	39.64
(C)AENPs-VNPs	63.04	43.54	39.53	32.57
AENPs(Ours)	**58.16**	**43.57**	**34.83**	**19.00**
CAENPs-biLSTM	**54.32**	**42.28**	**31.08**	**18.02**
CAENPs-GRU	**51.00**	**41.05**	**30.27**	**18.32**
CAENPs-Transformer	**48.03**	**37.04**	**28.55**	**17.49**

Table 5. Variations of Models for Modular Combinations of AENPs and CAENPs (MNIST dataset).

Method	50	100	200	400
	FID	*FID*	*FID*	*FID*
(C)AENPs-NPs	81.44	78.59	61.00	59.37
(C)AENPs-VNPs	77.32	74.48	59.05	48.96
AENPs(Ours)	**60.84**	**55.72**	**41.09**	**31.55**
CAENPs-biLSTM	**59.23**	**50.08**	**49.46**	**29.83**
CAENPs-GRU	**58.90**	**49.81**	**47.22**	**28.65**
CAENPs-Transformer	**49.54**	**58.28**	**45.93**	**20.70**

Table 6. Variations of Models for Modular Combinations of AENPs and CAENPs (celebA Dataset).

Unit (GRU) [3] (CAENPs-GRU), and Transformer [23] (CAENPs-Transformer) implementations in the latent path. From the results, it is clear that Transformer works better than the original LSTM and GRU methods with the different latent variables autoregressive constraints.

Different numbers of context sample points. We experimented with sampling ratios of 10%, 30%, 50%, 70%, and 90% in the MNIST and celebA datasets. We aim to compare the performance of AENPs and CAENPs models with different numbers of context sample points.

The effect of sliding window size on AENPs or CAENPs model. We experimented with different sliding windows on the dataset separately for MNIST and celebA effects. The final result is expressed as the value of the FID. Similarly, we take the number of contexT sample points to 50,100,200,400 (Results are shown in the table 7. We choose a sliding window size of 3 as the result in the AENPs or CAENPs model.

6. Conclusion

Neural Processes (NPs) are a new approach to modeling stochastic processes. The model is given only a portion of the context sample points, and the distribution of the target sample points is learned by means of function approximation to predict the target sample points. The model is given only a portion of the context sample points, and the distribution of the target sample points is learned by means of function approximation to predict the target sample points.

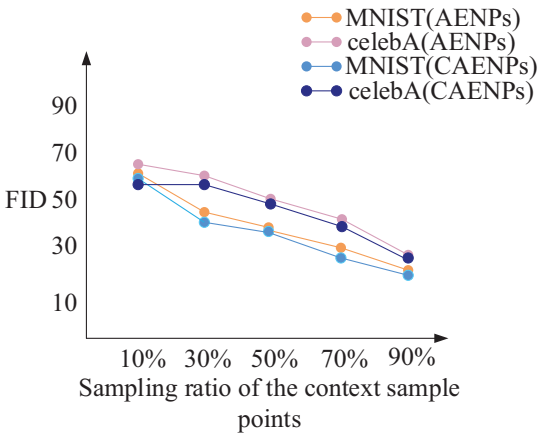

Figure 10. The FID values for the different proportions of sampled context sample points by the AENPs or CAENPs model.

Size of sliding window	50	100	200	400
	FID	*FID*	*FID*	*FID*
1	89.54	80.19	73.08	66.15
2	88.36	79.03	72.92	65.84
3	**58.16**	**43.57**	**34.83**	**19.00**
4	63.29	59.00	47.94	28.95
5	64.34	65.35	54.66	37.04
6	69.06	68.29	59.85	49.38

Table 7. The FID result of selecting a different number of pixel points from the images as context sample points to the complete images (MNIST dataset).

This paper reconciles the contradiction between local and global with sliding window attention and global attention for the NPs model and its variants on the relationship between sample points of contextual relations. At the same time, the spatial complexity of the original implementation in terms of self-attention is reduced. The latent representation draws on the properties of autoregression to achieve an overall structural representation of the latent variables stacked according to different levels. We implement the latent variable stacking by means of simple multiplication and gating mechanisms (LSTM). The gating mechanism is implemented by learning the dependencies between different levels of latent variables to achieve global perception. In future work, we plan to investigate the impact of superposition on the potential distribution of noisy data within the AENPs or CAENPs methods. Additionally, we aim to explore the performance of these models on other types of datasets and tasks, such as natural language processing and reinforcement learning.

References

[1] Henrique Aguiar, Mauro Santos, Peter Watkinson, and Tingting Zhu. Learning of cluster-based feature importance for electronic health record time-series. In *Proceedings of the 39th International Conference on Machine Learning*, pages 161–179, 2022.

[2] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv:2004.05150*, 2020.

[3] Junyoung Chung, Çaglar Gülçehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *ArXiv*, abs/1412.3555, 2014.

[4] Marta Garnelo, Dan Rosenbaum, Christopher Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo Rezende, and S. M. Ali Eslami. Conditional neural processes. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1704–1713. PMLR, 10–15 Jul 2018.

[5] Marta Garnelo, Jonathan Schwarz, Dan Rosenbaum, Fabio Viola, Danilo J. Rezende, S. M. Ali Eslami, and Yee Whye Teh. Neural processes. *ICML 2018 workshop*, abs/1807.01622, 2018.

[6] Zongyu Guo, Cuiling Lan, Zhizheng Zhang, Yan Lu, and Zhibo Chen. Versatile neural processes for learning implicit neural representations. In *The Eleventh International Conference on Learning Representations*, 2023.

[7] Ayoub El Hanchi, David Stephens, and Chris Maddison. Stochastic reweighted gradient descent. In *ICML*, pages 8359–8374, 2022.

[8] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. In *Advances in Neural Information Processing Systems*.

[9] Emiel Hoogeboom, Alexey A. Gritsenko, Jasmijn Bastings, Ben Poole, Rianne van den Berg, and Tim Salimans. Autoregressive diffusion models. In *International Conference on Learning Representations*, 2022.

[10] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. *ArXiv*, abs/1508.01991, 2015.

[11] Hyunjik Kim, Andriy Mnih, Jonathan Schwarz, Marta Garnelo, S. M. Ali Eslami, Dan Rosenbaum, Oriol Vinyals, and Yee Whye Teh. Attentive neural processes. volume abs/1901.05761, 2019.

[12] Mingyu Kim, Kyeong Ryeol Go, and Se-Young Yun. Neural processes with stochastic attention: Paying more attention to the context dataset. In *International Conference on Learning Representations*, 2022.

[13] Juho Lee, Yoonho Lee, Jungtaek Kim, Eunho Yang, Sung Ju Hwang, and Yee Whye Teh. Bootstrapping neural processes. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

[14] Minsub Lee, Junhyun Park, Sojin Jang, Chanhui Lee, Hyungjoo Cho, Minsuk Shin, and Sungbin Lim. Neural bootstrapping attention for neural processes, 2022.

[15] Zhihang Li, Teng Xi, Jiankang Deng, Gang Zhang, Shengzhao Wen, and Ran He. Gp-nas: Gaussian process based neural architecture search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[16] Wu Lin, Frank Nielsen, Khan Mohammad Emtiyaz, and Mark Schmidt. Tractable structured natural-gradient descent using local parameterizations. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 6680–6691. PMLR, 18–24 Jul 2021.

[17] Tung Nguyen and Aditya Grover. Transformer neural processes: Uncertainty-aware meta learning via sequence modeling. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 16569–16594. PMLR, 17–23 Jul 2022.

[18] Liming Pan, Cheng Shi, and Ivan Dokmanić. Neural link prediction with walk pooling. In *International Conference on Learning Representations*, 2022.

[19] Mohammad Pezeshki, Oumar Kaba, Yoshua Bengio, Aaron C Courville, Doina Precup, and Guillaume Lajoie. Gradient starvation: A learning proclivity in neural networks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 1256–1272, 2021.

[20] Michalis K Titsias, Jonathan Schwarz, AG de G Matthews, Razvan Pascanu, and Yee Whye Teh. Functional regularisation for continual learning using gaussian processes. *ICLR 2020*, 2020.

[21] Aaron van den Oord, Nal Kalchbrenner, Lasse Espeholt, koray kavukcuoglu, Oriol Vinyals, and Alex Graves. Conditional image generation with pixelcnn decoders. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2016.

[22] Aäron van den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. *CoRR*, abs/1601.06759, 2016.

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[24] Ruosi Wan, Zhanxing Zhu, Xiangyu Zhang, and Jian Sun. Spherical motion dynamics: Learning dynamics of normalized neural network using sgd and weight decay. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[25] Zhi Wang, Chicheng Zhang, and Kamalika Chaudhuri. Thompson sampling for robust transfer in multi-task bandits. In *Proceedings of the 39th International Conference on Machine Learning*, pages 23363–23416, 2022.

[26] Zesheng Ye and Lina Yao. Contrastive conditional neural processes. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9677–9686, 2022.

[27] Wentao Zhang, Ziqi Yin, Zeang Sheng, Yang Li, Wen Ouyang, Xiaosen Li, Yangyu Tao, Zhi Yang, and Bin CUI. Graph attention multi-layer perceptron. In *ICLR*, 2022.