Wavelet Conditional Neural Processes

Junyu Xuan[1,†] **Mengjing Wu**[1]

[1]Australian Artificial Intelligence Institute, University of Technology Sydney
[†]Correspondence to Junyu.Xuan@uts.edu.au

Abstract

Conditional neural processes (CNPs) are a family of stochastic processes defined by deep neural networks, characterized by the necessary properties of marginal consistency and exchangeability. Thanks to their generalization capabilities across tasks, popular applications of CNPs include meta-learning and multi-task learning. The existing CNPs map a context set to a vector or function space where all samples are considered homogeneously, which limits their representational power. In this paper, we introduce a Wavelet Conditional Neural Process (WaveCNP) as a new member of the CNP family, based on wavelet transform theory. We propose mapping the context set into a nested multiresolution function space sequence rather than a singular space, achieved through the efficient and adaptive discrete wavelet transform. We demonstrate that WaveCNP shows superior capability with existing CNPs in terms of conditional prediction and multiresolution prediction.

1. Introduction

Conditional neural processes (CNPs) (Garnelo et al., 2018a,b) are a family of stochastic processes defined by deep neural networks, characterized by the properties of marginal consistency and exchangeability. Similar to Gaussian processes (Rasmussen and Williams, 2006), which are popular stochastic processes for machine learning, a CNP defines a conditional predictive distribution $p_\theta(y|x, Z)$ over output variable y given input variable x and a context/measurement set $Z = \{x_c, y_c\}_{c=1}^{M}$. In contrast to Gaussian processes, CNPs can achieve more complex probability distribution approximations and efficient inference for large datasets, thanks to the advantages of deep neural networks. Due to their stochastic process nature, CNPs exhibit strong generalization capabilities for unseen samples or tasks. Popular applications include meta-learning (Ullrich et al., 2023), multi-task learning (Requeima et al., 2019), and transfer learning (Wei et al., 2019), among others.

The core component of CNPs is an encoder that maps a set of samples to a vector and is also considered a function on sets (Wagstaff et al., 2022). According to Theorem 2 of (Zaheer et al., 2017), any continuous, permutation-invariant function operating on sets can be expressed in a sum-decomposed form. While this sum-decomposed form defines a valid stochastic process via deep neural networks, it is challenging to encode translation equivariance, which is an important property in machine learning, especially in the field of computer vision. To achieve translation equivariance, the Convolutional CNP (ConvCNP) (Gordon et al., 2020) was innovatively proposed to define an encoder that maps a set to a function in a reproducing kernel Hilbert space (RKHS).

However, due to the finite depth and limited kernel size of the convolutional neural networks used in ConvCNP, modelling long-range dependencies remains difficult. Specifically, each convolutional layer only aggregates information from a local neighbourhood, and the effective receptive field grows gradually with network depth. As a result, capturing interactions between spatially distant observations requires either a very deep network or large convolutional kernels, both of which can increase

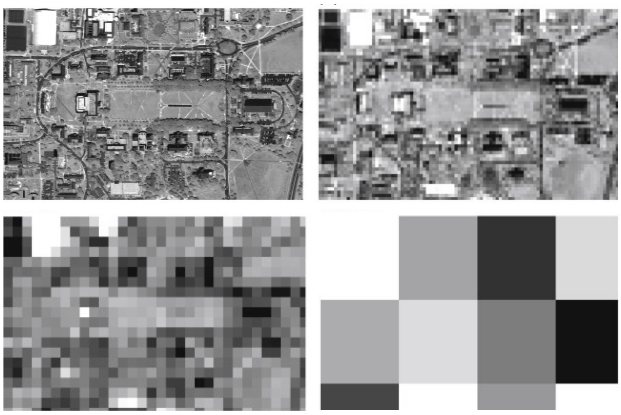

Figure 1: Examples of data with different resolutions of satellite images (Liang and Wang, 2020).

computational cost and optimisation difficulty. This limitation is particularly problematic when the target prediction depends on global contextual information, non-local correlations, or spatial structures distributed across distant regions. For example, in image-based regression or spatial modelling tasks, observations that are far apart may still be strongly correlated due to shared object-level structures, periodic patterns, global trends, or long-range physical dependencies. In addition, both CNP and ConvCNP map the data, or tasks in meta-learning, into a single representation space, either a vector space or a function space, and subsequently treat the representations as homogeneous through the same decoder. We argue that this homogeneous representation strategy limits their expressive power and, consequently, their ability to model complex conditional predictive distributions, as demonstrated in our experiments. Furthermore, data collected in practical applications may naturally contain mixed resolutions, as illustrated in Fig. 1, due to differences in input sources, sensing conditions, or imaging devices. Existing CNP-based methods unfortunately overlook this important characteristic.

In this work, we propose a new member of the CNP family - Wavelet CNP (WaveCNP) - to extend the representation capability of CNPs based on the wavelet transform theory. Specifically, we map sets to a shared multiresolution function space sequence, where a series of nested function spaces are constructed, and different spaces are connected by scaling functions and wavelets from the wavelet theory (Mallat, 1999). With the shared multiresolution vector space sequence, data with varying resolutions can be aligned correctly in the space, and functions at different resolutions are transformed using respective nonlinear transformations, such as Conv or LieConv. Meanwhile, since each resolution space is a Hilbert space spanned by specific scaling functions, the translation (Gordon et al., 2020) and group equivariance (Kawano et al., 2020) properties can be preserved. Our key contributions are summarized as follows:

- We propose a new member of the conditional neural process family, WaveCNP, which maps sets to a shared multiresolution space sequence, offering better representation power compared with singular vector or function spaces.
- We show that WaveCNP can model the conditional predictive distribution more efficiently than the baselines and make multiresolution predictions.

2. Background

Let $\mathcal{X} \subseteq \mathbb{R}^d$ and $\mathcal{Y} \subseteq \mathbb{R}^{d'}$ be the spaces of inputs and outputs[1], respectively. Define $\mathcal{Z}_M = (\mathcal{X} \times \mathcal{Y})^M$ as the collection of M input-output pairs. All notations used in this paper are listed and explained

1. Without loss of generality, we assume $d' = 1$ for the remainder of this paper unless specified otherwise.

in Table 4 in Appendix A. All CNPs assume there is an underlying stochastic process that defines a prior over the random functions $f : \mathcal{X} \to \mathcal{Y}$, where each function is referred to as a task in multi-task learning or a task in meta learning. Given a set of observation/context/measurement points $Z = \{\mathbf{x_c}, \mathbf{y_c}\} \subseteq \mathcal{Z}_M$ and some target inputs $\mathbf{x_t} \in \mathcal{X}$, the aim of CNPs is to model the predictive distribution $p(\mathbf{y_t}|\mathbf{x_t}, Z)$, where $\mathbf{y_t} \in \mathcal{Y}$. Two examples closely related to this work are introduced below (more detailed discussions on related works can be found in Appendix C), followed by the basic concepts of wavelet transform theory which will be used in our algorithm design.

2.1. Conditional neural processes (CNP)

CNP (Garnelo et al., 2018a) model the predictive distribution $p(\mathbf{y_t}|\mathbf{x}, Z) = p(\mathbf{y_t}|\Phi(\mathbf{x_t}, Z))$ as a Gaussian distribution $\mathcal{N}(y_t|\mu_t, \sigma_t)$ through a composition $\Phi = \rho \circ E$ of encoder E and decoder ρ,

$$\{(x_c, y_c)\} \xrightarrow{\text{by } E} \kappa = \oplus\{\kappa_c\} \xrightarrow{\text{by } \rho} \{(\mu_t, \sigma_t)\}$$

where $E : \mathcal{Z} \to \mathbb{R}^e$ *maps the set to an embedding* $\kappa = \oplus\{\kappa_c\} = \kappa_1 \oplus \kappa_2 \oplus \cdots \in \mathbb{R}^e$; $\oplus$ is a commutative operation that takes elements in $\mathbb{R}^e$ and maps them into a single element of $\mathbb{R}^e$ and needs to be permutation invariant; $\rho : \mathbb{R}^e \times \mathcal{X} \to \mathbb{R}^2$ where $\mathbb{R}^2$ is the space of mean and variance for the predictive Gaussian distribution; and μ_t, σ_t denote the predictive mean and variance.

2.2. Convolutional conditional neural processes (ConvCNP)

Since the equivariance with respect to input translations in $\mathcal{X}$ is not addressed in CNP, ConvCNP (Gordon et al., 2020) was proposed to *map a (context) set into a function space* $\mathcal{H}$ rather than a vector space $\mathbb{R}^e$ to achieve translation equivariant CNPs. The core of ConvCNP is a ConvDeepSet function defined in Theorem 1, which can be considered an extension of a finite-dimensional (vector) deep set to an infinite-dimensional (functional) space.

Theorem 1. *(Gordon et al., 2020) Let* $\mathcal{Z}'_{\leq M} \in \mathcal{Z}_{\leq M}$ *be topologically closed under permutations and translations with multiplicity* K *and* $\mathcal{Z}_{\leq M} = \bigcup_{m=1}^{M} \mathcal{Z}_m$. *For a function* $\Phi : \mathcal{Z}'_{\leq M} \to C_b(\mathcal{X}, \mathcal{Y})$ *(space of continuous, bounded functions* $\mathcal{X} \to \mathcal{Y}$ *endowed with supremum norm), the following conditions are equivalent:*

- Φ *is continuous, permutation-invariant and translation-equivariant.*
- *There exist a function space* $\mathcal{H}$ *and a continuous and translation-equivariant function* $\eta : \mathcal{H} \to C_b(\mathcal{X}, \mathcal{Y})$ *and a continuous kernel* $\ker : \mathcal{X}^2 \to \mathbb{R}$ *such that*

$$\Phi(Z) = \rho\left(\eta\left(h(Z)\right)\right), \; h(Z) = \sum_{i=1}^{m} \varphi_{K+1}(y_i) \ker(\cdot, x_i)$$

  where $\varphi_{K+1} : \mathcal{Y} \to \mathbb{R}^{K+1}$ *is defined by* $\varphi_{K+1}(y) = [1, \ldots, y^K]^T$.

The function Φ *of the above form is called ConvDeepSet.*

The forward process of ConvCNP can be simply summarized as

$$\{(x_c, y_c)\} \xrightarrow{①} \{(\hat{x}_i, h_i)\} \xrightarrow{②} \{(f_\mu(\hat{x}_i), f_\sigma(\hat{x}_i))\} \xrightarrow{③} \{(\mu_t, \sigma_t)\}$$

where ① is by kernel regression; ② is based on CNNs; ③ is by kernel regression; the function $h(Z)$ is represented by its values on grid supports $\{(\hat{x}_i, h_i)\}$. Group equivariant conditional neural processes (EquivCNP) (Kawano et al., 2020) further extend the translation equivariance to more general group equivariance, such as rotation and scaling. The idea is to replace the conventional convolutions in Theorem 1 with group convolutions (Finzi et al., 2020) to obtain the EquivDeepSet.

2.3. Wavelet transform theory

Before introducing the wavelet transform, we first introduce a classical signal processing concept: multiresolution vector space sequence.

Definition 1. *(Mallat, 1989) A multiresolution vector space sequence $\{V_j\}_{j\in\mathbb{Z}}$ is a set of vector spaces of measurable and square-integrable functions, satisfying the following properties: (1) $\forall j \in \mathbb{Z}, V_j \subset V_{j+1}$; (2) $\forall j \in \mathbb{Z}, g(x) \in V_j \iff g(2x) \in V_{j+1}$; (3) $\cup_{j=-\infty}^{+\infty} V_j$ is dense in $L^2(\mathbb{R})$ and $\cap_{j=-\infty}^{+\infty} V_j = \{\mathbf{0}\}$, where j is resolution level index; $g(x)$ is a function in $L^2(\mathbb{R})$ (square-integrable function space); the sequence of resolutions varies exponentially $(r^j)_{j\in\mathbb{Z}}$ $(r > 1)$; and $r = 2$ is chosen to ease the implementation.*

This multiresolution vector space sequence has a favourable property:

Theorem 2. *(Mallat, 1989) Let $\{V_j\}_{j\in\mathbb{Z}}$ be a multiresolution vector space sequence, then there exists a unique function $\phi(x)$ called a scaling function, such that for any $j \in \mathbb{Z}$ if $\phi^j(x) = \sqrt{2^j}\phi(2^j x)$ then $\{\phi^j(x - 2^{-j}n)\}_{n\in\mathbb{Z}}$ is an orthonormal basis of V_j.*

That is to say, we can construct a multiresolution vector space sequence using these orthonormal bases defined by a unique scaling function $\phi(x)$. Given this set of orthonormal bases, discrete wavelet transform (DWT) (Mallat, 1999) can map a (discrete) function, h, to a multiresolution vector space sequence through

$$h[\hat{x}_i] = \frac{1}{\sqrt{N}} \left(\sum_k W_\phi^{j_0}[k]\phi_k^{j_0}[\hat{x}_i] + \sum_{j=j_0}^{\infty} \sum_k W_\psi^j[k]\psi_k^j[\hat{x}_i] \right) \tag{1}$$

where the function is represented by N grid inputs/samples/points $\{\hat{x}_i\}_{i=1:N}$ and their function values, ψ is *wavelet function* that is orthogonal to ϕ, k is the filter length of the discrete scaling and wavelet functions, and

$$W_\phi^{j_0}[k] = \frac{1}{\sqrt{N}} \sum_{\hat{x}} h[\hat{x}]\phi_k^{j_0}[\hat{x}], \quad W_\psi^j[k] = \frac{1}{\sqrt{N}} \sum_{\hat{x}} h[\hat{x}]\psi_k^j[\hat{x}], \ j \geq j_0 \tag{2}$$

are called *approximation coefficients* and *detailed coefficients*, respectively. The wavelet transform could decompose the whole function space to $L^2(R) = V_0 \oplus W_0 \oplus W_1 \cdots$, where each subspace is spanned by the corresponding bases, e.g., V_0 is spanned by ϕ and W_0 is spanned by ψ^0. Hence, DWT provides a way to analyze signals at different levels of detail, making it useful for traditional applications like image compression and noise reduction. The process is reversible, meaning we can recover the function from the coefficients and wavelet functions in a procedure known as the inverse wavelet transform (IWT). Note that DWT and IWT are computationally efficient, with $O(N)$ complexity. This is one of the key advantages of our approach. More in-depth explanations of wavelet transformation can be found in (Pathak, 2009).

3. Wavelet deep sets

The problem of existing CNPs is that they map the context set to a singular vector or function space, which mixes information with different resolutions. Hence, our idea is to map the context set to a multiresolution function space instead. First, we map a set to a function in a space $h \in \mathcal{H}^{L2} = \mathcal{H} \bigcap L^2$, where $\mathcal{H}$ is RKHS and L^2 is the square-integrable space $\int |h|^2 \mathrm{dx} < \infty$, which is slightly different from the space of ConvCNP. This is mainly reflected in that not all kernel functions can be used here, as some are not square-integrable. For example, the polynomial kernel does not have this property. Fortunately, many candidates do, such as Gaussian (RBF), tricube, triweight, quadratic and cosine. Note that the default kernel used in ConvCNP is RBF, and in the following discussions,

we will use RBF as an example as well. Then, the above space requirement will not constitute a significant restriction.

We propose to map the sets into a shared multiresolution function space (Definition 1) by wavelet transformation. Before introducing the mapping details, let us look at the benefits of this mapping:

- The approximation functions with different resolutions are orthonormal projections in each space V_j, so the information with different resolutions is well separated, and each resolution r^j is characterised or captured by each V_j;
- This is a set of nested vector spaces, and the spaces are closely and consistently connected through properties i and ii, which not only ensures conceptual reasonability but also eases the mapping (explained later in detail);
- It can represent (cover) almost all the functions in the space L^2 as the resolution increases.

With the above beneficial properties, the information of the context set with different resolutions can be separated and then transformed respectively. This contrasts with CNP and ConvCNP, where information with different resolutions is mixed and transformed in the same way. To achieve the mapping, we use DWT by Eq. (1) to map the function from a context set to a multiresolution vector space sequence. Then, we have the following new deep set:

Definition 2. *Let $\mathcal{H}^{L^2}$ be a square-integrable function space, and let $\ker : \mathcal{X}^2 \to \mathbb{R}$ be a continuous kernel satisfying $\int \ker^2 < \infty$. Let ϕ and ψ denote the scaling function and wavelet function, respectively. Let $W_\phi^{j_0}[k]$ and $W_\psi^{j'}[k]$ be the approximation and detail coefficients associated with the multiresolution representation. Let J denote the total number of resolution levels in the multiresolution vector space sequence, and let $\{\eta_j : \mathcal{H}^{L^2} \to C_b(\mathcal{X}, \mathcal{Y})\}_{j=1}^J$ be a set of continuous mapping functions.*

For an input set $Z = \{x_c, y_c\}_{c=1}^N$, a WaveDeepSet Φ is defined as a function of the following form:

$$\Phi(Z) = \rho\left(\frac{1}{J}\sum_j \eta_j(E_j(h(Z)))\right),$$

where

$$h[\hat{x}_i] = \sum_{c=1}^{C} \varphi_{K+1}(y_c)\ker(\hat{x}_i, x_c), \qquad i = 1, 2, \ldots, N,$$

and

$$E_j(h) = \frac{1}{\sqrt{N}}\left(\sum_k W_\phi^{j_0}[k]\phi_k^{j_0}[\hat{x}_i] + \sum_{j'=j_0}^{j}\sum_k W_\psi^{j'}[k]\psi_k^{j'}[\hat{x}_i]\right).$$

Here, $\varphi_{K+1} : \mathcal{Y} \to \mathbb{R}^{K+1}$ is defined by

$$\varphi_{K+1}(y) = [1, y, y^2, \ldots, y^K]^T.$$

The function Φ defined above is called a WaveDeepSet.

Next, we outline several mathematical properties derived from Definition 2. These properties are essential for preserving the characteristics of a valid stochastic process and for handling data across multiple resolutions.

Corollary 1 (Permutation invariance). *WaveDeepSet is permutation-invariant. That is, for any permutation ν of the input samples, $\Phi(Z) = \Phi(\nu Z)$, where $\nu Z = \{(x_{\nu(1)}, y_{\nu(1)}), \ldots, (x_{\nu(N)}, y_{\nu(N)})\}$.*

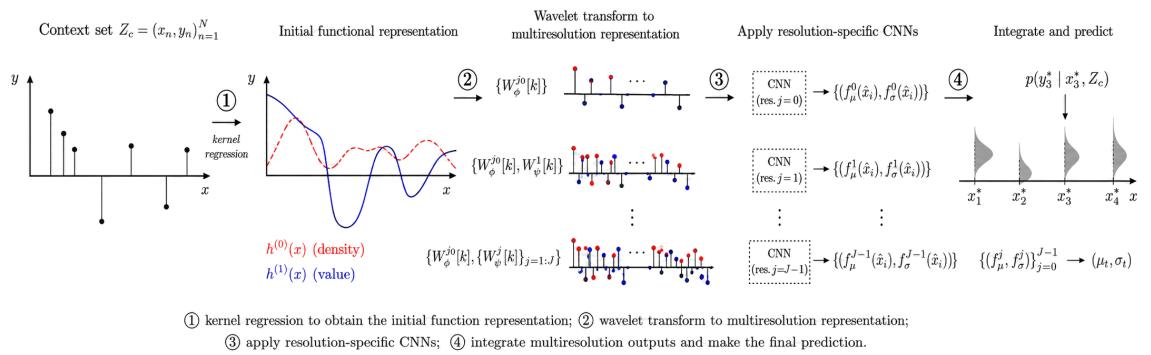

Figure 2: The forward process of WaveCNP.

Corollary 2 (Translation equivariance). *Assume that the kernel is translation-invariant, i.e.,* $\ker(x+\tau, x_c+\tau) = \ker(x, x_c)$*, and that the multiresolution operators* $\{E_j\}_{j=1}^J$ *are constructed using a stationary discrete wavelet transform, so that* $E_j(T'_\tau h) = T'_\tau E_j(h),\ j = 1, \ldots, J$*. Further assume that the mapping functions* $\{\eta_j\}_{j=1}^J$ *and the readout function* ρ *are translation-equivariant. Then WaveDeepSet is translation-equivariant. That is,* $\Phi(T_\tau Z) = T'_\tau \Phi(Z)$*.*

Corollary 3 (Resolution-wise alignment). *Assume that the multiresolution representation in Definition 2 is generated by an orthonormal wavelet decomposition* $V_J = V_{j_0} \oplus \bigoplus_{r=j_0}^J \mathcal{W}_r$*, where* V_{j_0} *is the coarse approximation space and* $\mathcal{W}_r$ *is the detail subspace at resolution level* r*. For each* $j = j_0, \ldots, J$*, define the cumulative resolution space* $V_j = V_{j_0} \oplus \bigoplus_{r=j_0}^j \mathcal{W}_r$*. Let* P_{V_j} *and* $P_{\mathcal{W}_r}$ *denote the orthogonal projections onto* V_j *and* $\mathcal{W}_r$*, respectively. Then the operator* E_j *in Definition 2 is the orthogonal projection onto* V_j*, namely* $E_j(h) = P_{V_j} h$*. Consequently, for any task* $h^{(\ell)} \in V_\ell$ *generated at resolution level* ℓ*, all finer-resolution detail components vanish:* $P_{\mathcal{W}_r} h^{(\ell)} = 0, \qquad r > \ell$*. Equivalently, its wavelet detail coefficients satisfy* $W_\psi^r[k] = 0,\ r > \ell$*.*

Please see the Appendix D for the proofs. The third property means that tasks with different resolutions are aligned in the multiresolution representation in the sense that each task is represented through its projection onto the corresponding nested resolution space, and no artificial finer-resolution components are introduced.

4. Wavelet conditional neural processes

In this section, we discuss the implementation and training details for WaveCNPs. The architectures for CONV, ρ, and η vary depending on the data and are provided in Appendix C. To maintain the permutation-invariant and translation-equivariant properties, we use CNNs for $\{\eta_j\}_{j=0:J-1}$. The forward process is shown in Fig. 2, and Algorithm 1 in Appendix B also shows the whole procedure, where the blue lines indicate the unique parts compared with ConvCNP.

Given a context set (Z_c in Line 2 of Algorithm 1), we first obtain its functional representation h through CONV. This step is the same as ConvCNP (Gordon et al., 2020), so we do not provide further explanations. Note that the number N of grid points $\{\hat{x}_i\}_{i=1:N}$ for the functional representation h is selected as an integer power of 2 to ease the wavelet transform.

To obtain functional representations at different resolutions, we use the DWT to obtain all approximation coefficients and detailed coefficients (W_ϕ and W_ψ in Line 5 of Algorithm 1). W_ϕ is a single matrix that stores the 'core' information, and W_ψ is a list (with length J) that stores the 'detailed' information at each resolution level. Hence, to obtain a functional representation at level j, we can remove the detailed information at levels $j' < j$ by setting $W_\psi^{j'} = \mathbf{0}$ and then use the inverse wavelet

transform (IWT) to recover the h_j based approximation coefficients and new detailed coefficients (Line 9 of Algorithm 1). The implementations of DWT and IWT here are based on Quadrature Mirror Filter bank (QMF) for higher computational efficiency. QMF (Mallat, 1999) relies on high-pass and low-pass filters, ξ^h and ξ^l, to compute the coefficients rather than the corresponding explicit wavelet and scaling functions[2].

Given the obtained functional representations at different resolutions, we transform them using the respective mapping functions $\{\eta_j\}$ (Line 10 of Algorithm 1), which is different from the uniform transformation used in ConvCNP. Then, all functions are added together (Line 11 of Algorithm 1) to predict the mean and variance via ρ (Line 13 of Algorithm 1). Similar to other CNPs, WaveCNPs also use the negative log-likelihood defined by predictive conditional distribution on test data as the training loss function,

$$\log p(Y|X,Z) = \log \prod_t p(y_t|\Phi_\theta(Z)(x_t)) = \sum_t \log \mathcal{N}(y_t;\mu_t,\sigma_t), \tag{3}$$

where $\mu_t, \sigma_t = \Phi_\theta(Z)(x_t)$ and θ denotes its trainable model parameters.

One important factor is selecting the wavelet function (or its filters equivalently). The popular wavelet functions in the literature include, but are not limited to, Haar, Daubechies, Coiflet (Mallat, 1999), and Triglets (Saydjari and Finkbeiner, 2022). Different wavelets have different waveforms and may lead to different function approximation performance (Yger and Rakotomamonjy, 2011). The selection of appropriate wavelet functions is not trivial and requires strong prior knowledge of the underlying functions. Hence, we propose an adaptive wavelet transformation based on parameterised QMF via the Sherlock-Monro algorithm (Sherlock and Monro, 1998). Take $J = 3$ as an example for simplicity. We can parameterise QMF as

$$\xi^l[i] = \begin{cases} \frac{1-\cos(\beta)+(-1)^i\sin(\beta)}{2\sqrt{2}} & i = 0,3 \\ \frac{1+\cos(\beta)+(-1)^{i-1}\sin(\beta)}{2\sqrt{2}} & i = 1,2 \end{cases} \tag{4}$$

where $\beta \in [0, 2\pi]$ is the parameter to control the underlying wavelet form. For example, when $\beta = \pi/3$, we can recover the filters for Daubechies-2. Note that ξ^l is the low-pass filter, and we can obtain the high-pass filter via their relationship $\xi^h = \big(-\xi^l[3], \xi^l[2], -\xi^l[1], \xi^l[0]\big)$. This method ensures that the obtained filters are orthogonal to each other. In the implementation, we can optimize β to find the appropriate wavelet form.

Table 1: Predictive log-likelihood results of 1-D synthetic regression

Model/Param	Matern	RBF	Polynomial	RBF+Linear
CNP/149k	0.0542 ± 0.0270	0.4376 ± 0.0258	1.0571 ± 0.0730	0.4665 ± 0.0510
AttnCNP/248k	0.1167 ± 0.1996	0.1079 ± 0.0832	1.3214 ± 0.0796	0.8651 ± 0.4154
ConvCNP/154k	0.7609 ± 0.0762	0.9016 ± 0.1605	0.7602 ± 0.1742	0.4277 ± 0.7805
EquivCNP/161k	0.9672 ± 0.1470	1.3922 ± 0.1359	0.4606 ±0.1754	1.2440 ±0.2203
TNP/155k	1.1116 ± 0.0137	**1.5383 ± 0.0206**	1.0377 ± 0.0553	1.5415 ± 0.0113
WaveCNP/103k	0.9956 ± 0.0883	1.2348 ± 0.2074	1.2747 ± 0.1182	1.4486 ± 0.0847
WaveCNP-a/103k	**1.1234 ± 0.0379**	1.4909 ± 0.1070	**1.4299 ± 0.0913**	**1.5451 ± 0.0684**

5. Experiments

We evaluate the performance of WaveCNP on commonly used benchmark tasks (function regression and image completion). We compare it with closely related ConvCNP[3] (Gordon et al., 2020);

2. The filters of some popular wavelet functions can be found at https://wavelets.pybytes.com/wavelet/db2/
3. https://github.com/cambridge-mlg/convcnp

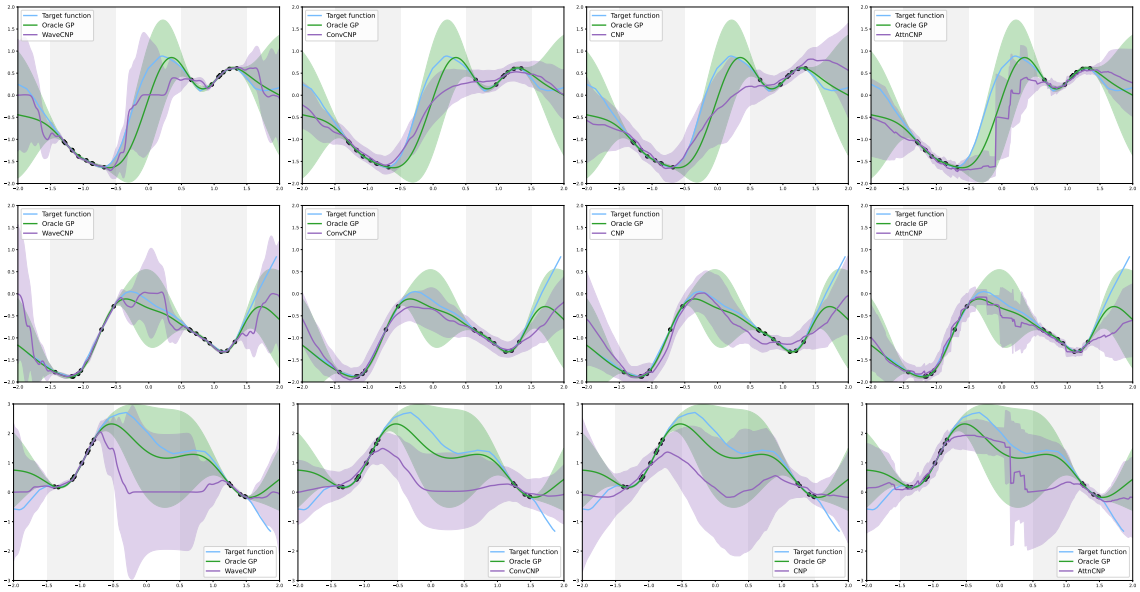

Figure 3: Examples of 1-D synthetic regression. Three blue target functions (corresponding to three rows) are sampled from a Gaussian process (GP); black dots are the randomly sampled context set from grey regions; an GP posterior is plotted with a green line and fill (2 standard deviations); the purple lines and fills in four columns are the predictions from WaveCNP, ConvCNP, CNP, and AttnCNP, respectively.

Table 2: Predictive log-likelihood results on 2-D image completion

Model/Param	MNIST	CIFAR	SVHN	CELEBA
CNP/149k	1.3705 ± 0.0451	0.6216 ± 0.0085	1.0754 ± 0.0080	0.6418 ± 0.0105
AttnCNP/166k	1.0290 ± 0.0671	1.0322 ± 0.0111	1.7561 ± 0.0284	1.1071 ± 0.0215
ConvCNP/154k	1.7516 ± 0.1000	1.3651 ± 0.0170	2.1517 ± 0.0238	1.3878 ± 0.0202
EquivCNP/141k	1.8498 ± 0.0628	1.3742 ± 0.0117	2.1635 ± 0.0322	**1.4396 ± 0.0270**
TNP/155k	1.3974 ± 0.0516	1.0984 ± 0.0194	2.1882 ± 0.0223	1.3281 ± 0.0182
WaveCNP/145k	1.8650 ± 0.0966	1.3885 ± 0.0265	**2.2179 ± 0.0065**	1.4077 ± 0.0051
WaveCNP-a/145k	**1.9361 ± 0.0953**	**1.4015 ± 0.0170**	2.2022 ± 0.0105	1.4290 ± 0.0128

two standard neural process family members, CNP (Garnelo et al., 2018a) and AttnCNP (Kim et al., 2019); EquivCNP[4] (Kawano et al., 2020), and a recent state-of-the-art Transformer NP (TNP[5]) (Nguyen and Grover, 2022). For WaveCNP, we used the Daubechies-2 (db2)[6]. The full implementation details can be found in our codes[7] and Appendix E.

5.1. 1-D synthetic regression

The tasks here are functions sampled from a Gaussian process (GP) (Rasmussen and Williams, 2006), where different tasks/functions are assumed to share some information. At the training stage, a random function is first drawn from a GP, and then several context and target points are sampled from this function to feed the model, while the log-likelihood of some sampled target points is used as the loss function. At the test stage, several functions are sampled from the same GP,

4. https://github.com/makora9143/EquivCNP
5. https://github.com/tung-nd/TNP-pytorch
6. https://pytorch-wavelets.readthedocs.io/
7. https://github.com/JunyuXuan/wavecnp-master

Table 3: Predictive log-likelihood results on 2-D image completion with varying task resolutions.

Model	MNIST	CIFAR	SVHN	CELEBA
CNP	2.0925 ± 0.065	0.8249 ± 0.072	1.2052 ± 0.034	0.7950 ± 0.066
AttnCNP	2.0905 ± 0.108	1.3769 ± 0.062	1.8300 ± 0.095	1.3290 ± 0.059
ConvCNP	2.6598 ± 0.062	1.7244 ± 0.081	2.3388 ± 0.076	1.6459 ± 0.084
EquivCNP	1.8482 ± 0.056	1.7877 ± 0.051	2.2908 ± 0.089	1.6332 ± 0.073
TNP	1.6187 ± 0.013	1.5171 ± 0.001	1.8332 ± 0.002	1.5473 ± 0.007
WaveCNP	**2.9176 ± 0.048**	1.7417 ± 0.063	2.4868 ± 0.094	**1.6922 ± 0.070**
WaveCNP (adapt)	2.3738 ± 0.051	**1.7947 ± 0.077**	**2.6033 ± 0.091**	1.6892 ± 0.089

and the log-likelihood of target points given a random context set is used as the final evaluation metric. In our experiments, we used 256 random functions for training, 60 functions for validation, and 2048 functions for testing. We used GP with different kernel functions to simulate different functions, including Matern, RBF, Polynomial, and RBF+Linear kernels. The implementation of GP is based on GPyTorch[8]. All models shared the same settings (given in Appendix E), including training epochs, learning rate, weight decay, etc.

Some examples are given in Fig. 3, where we show the predictions (purple line and fill) of four models (CNP, AttnCNP, ConvCNP, and our WaveCNP) on three functions (blue line, randomly drawn from a GP). A GP posterior is given for comparison (green line and fill), and the context set (black dots) is purposely sampled from two grey regions. From this figure, we can observe that our WaveCNP can make accurate predictions/fitting of the target function in the grey regions with small variances and outputs large variances in the regions without observed data. The full results are given in Table 1. We also performed a paired t-test to verify the significance of these results, comparing ours with the results from the other three models. Each model had 5 independent runs with different seeds (but shared by different models), and the mean and standard deviation are reported in the table. We can see that our WaveCNP achieved the best predictions overall. Note that the non-adaptive WaveCNP had already achieved relatively good performance, and the adaption could further improve the prediction. All tests show that the results are statistically significant (p-values are < 0.05) with only two exceptions: the results of AttnCNP and ours on Polynomial kernel (the p-value was 0.053, marked by underlines in the table); the results of TNP and ours on RBF+Linear kernel. Hence, we can conclude that the proposed multiresolution functional representation is helpful for predictive conditional distribution modelling.

5.2. 2-D image completion

Beyond 1-D data, we also evaluate the proposed model on 2-D image (also known as on-the-grid) data. Image completion is a standard evaluation method of the neural process family, where the task is an image and the goal is to predict the values of some target pixels given some context pixels. We used the images from benchmark datasets[9]: MNIST (LeCun et al., 1998), CIFAR (Krizhevsky and Hinton, 2009), SVHN (Netzer et al., 2011), and CELEBA (Liu et al., 2015). For each dataset, we randomly selected 1,000 images for training, 200 for validation, and 5,000 for testing. Full details are in Appendix E. The results are shown in Table 2, where each model had 5 independent runs, and the mean and standard deviation are reported in the table. We can see that 1) our WaveCNP consistently had a generally good performance among all models (even on SVHN and CELEBA, ours was not the best but comparable with the best), and 2) the adaptation generally led to performance improvement with only one exception on SVHN. Hence, we argue that the multiresolution functional representation is beneficial for image completion.

8. https://gpytorch.ai/
9. https://pytorch.org/vision/stable/datasets.html

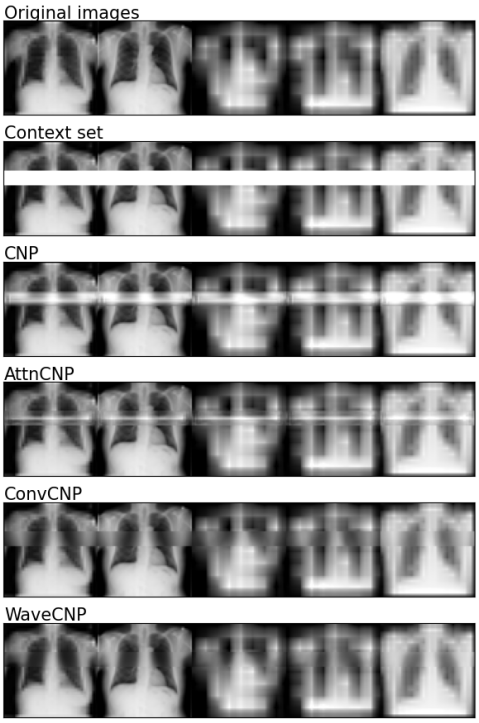

Figure 4: Examples of 2-D image completion. The white stripe in each image (Context set) is the prediction/completion target.

We also tested images with varying resolutions from the original images (with uniform resolution). In this task, we also include real-world data: NIH-chest-xrays[10] (Wang et al., 2017). For each image from the benchmark image datasets, we first applied the wavelet transform to obtain its varying resolutions and then randomly selected a resolution to form a new dataset. The final dataset is an image set with varying resolutions, which was fed to the models. An example of this task is shown in Fig. 4, where we used the part of an image as the context set and predicted the white stripe area. We can see that WaveCNP could make relatively reasonable completions. The full results are shown in Table 3. Since we know the wavelet functions used for resolution generation (db2), we used it in WaveCNP and did not adapt it. We can observe from the table that our model can achieve the best performance among all models.

6. Conclusion

In this paper, we proposed a new member for the conditional neural process family - Wavelet CNP, which maps the context set to a more powerful multiresolution functional representation rather than the vector or functional representations used by existing CNPs. The classical wavelet transform theory was used to build a multiresolution functional space and map the context set into it. Experimental studies showed that our WaveCNP can outperform existing CNPs in terms of conditional distribution modelling and multiresolution prediction.

10. https://www.kaggle.com/datasets/nih-chest-xrays/data

References

Khaled A. Althelaya, Salahadin A. Mohammed, and El-Sayed M. El-Alfy. Combining deep learning and multiresolution analysis for stock market forecasting. *IEEE Access*, 9:13099–13111, 2021.

Matthew Ashman, Cristiana Diaconu, Junhyuck Kim, Lakee Sivaraya, Stratis Markou, James Requeima, Wessel P. Bruinsma, and Richard E. Turner. Translation equivariant transformer neural processes. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 1924–1944. PMLR, 2024.

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

Andrew Carr, Jared Nielsen, and David Wingate. Wasserstein neural processes. *arXiv preprint arXiv:1910.00668*, 2019.

Leo Feng, Frederick Tung, Hossein Hajimirsadeghi, Yoshua Bengio, and Mohamed Osama Ahmed. Memory efficient neural processes via constant memory attention block. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 13365–13386. PMLR, 2024.

Marc Finzi, Samuel Stanton, Pavel Izmailov, and Andrew Gordon Wilson. Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data. In *International Conference on Machine Learning*, 2020.

Andrew Foong, Wessel Bruinsma, Jonathan Gordon, Yann Dubois, James Requeima, and Richard Turner. Meta-learning stationary stochastic process prediction with convolutional neural processes. In *Advances in Neural Information Processing Systems*, 2020.

Marta Garnelo, Dan Rosenbaum, Christopher Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo J. 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, 2018a.

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

Jonathan Gordon, Wessel P. Bruinsma, Andrew Y. K. Foong, James Requeima, Yann Dubois, and Richard E. Turner. Convolutional conditional neural processes. In *International Conference on Learning Representations*, 2020.

Mohammad Hadi, Xuesong Zhou, David Hale, et al. Multiresolution modeling for traffic analysis: guidebook. Technical report, United States. Federal Highway Administration, 2022.

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*, 2021.

Daolang Huang, Manuel Haussmann, Ulpu Remes, ST John, Grégoire Clarté, Kevin Luck, Samuel Kaski, and Luigi Acerbi. Practical equivariances via relational conditional neural processes. In *Advances in Neural Information Processing Systems*, volume 36, pages 29201–29238, 2023.

Saurav Jha, Dong Gong, Xuesong Wang, Richard E. Turner, and Lina Yao. The neural process family: Survey, applications and perspectives. *arXiv preprint arXiv:2209.00517*, 2022.

Makoto Kawano, Wataru Kumagai, Akiyoshi Sannai, Yusuke Iwasawa, and Yutaka Matsuo. Group equivariant conditional neural processes. In *International Conference on Learning Representations*, 2020.

Hyunjik Kim, Andriy Mnih, Jonathan Schwarz, Marta Garnelo, Ali Eslami, Dan Rosenbaum, Oriol Vinyals, and Yee Whye Teh. Attentive neural processes. In *International Conference on Learning Representations*, 2019.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

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*, 2020.

Shunlin Liang and Jindi Wang, editors. *A systematic view of remote sensing*, pages 1–57. Academic Press, second edition edition, 2020.

Hui Liu, Rui Yang, and Zhu Duan. Wind speed forecasting using a new multi-factor fusion and multi-resolution ensemble model with real-time decomposition and adaptive error correction. *Energy Conversion and Management*, 217:112995, 2020.

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

Stéphane Mallat. *A Wavelet Tour of Signal Processing*. Elsevier, 1999.

Stephane G Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, 1989.

Stratis Markou, James Requeima, Wessel P. Bruinsma, Anna Vaughan, and Richard E. Turner. Practical conditional neural processes via tractable dependent predictions. In *International Conference on Learning Representations*, 2022.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

Tung Nguyen and Aditya Grover. Transformer neural processes: Uncertainty-aware meta learning via sequence modeling. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 16569–16594. PMLR, 2022.

Ram Shankar Pathak. *The Wavelet Transform*, volume 4. Springer Science & Business Media, 2009.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, 2006.

James Requeima, Jonathan Gordon, John Bronskill, Sebastian Nowozin, and Richard E. Turner. Fast and flexible multi-task classification using conditional neural adaptive processes. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 7957–7968, Red Hook, NY, USA, 2019. Curran Associates Inc.

Andrew K Saydjari and Douglas P Finkbeiner. Equivariant wavelets: Fast rotation and translation invariant wavelet scattering transforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2):1716–1731, 2022.

Barry G Sherlock and Donald M Monro. On the space of orthonormal wavelets. *IEEE Transactions on Signal Processing*, 46(6):1716–1720, 1998.

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

Lars Ullrich, Andreas Völz, and Knut Graichen. Robust meta-learning of vehicle yaw rate dynamics via conditional neural processes. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pages 322–327. IEEE, 2023.

Edward Wagstaff, Fabian B Fuchs, Martin Engelcke, Michael A Osborne, and Ingmar Posner. Universal approximation of functions on sets. *Journal of Machine Learning Research*, 23(151):1–56, 2022.

Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2097–2106, 2017.

Ying Wei, Peilin Zhao, Huaxiu Yao, and Junzhou Huang. Transferable neural processes for hyperparameter optimization. *arXiv preprint arXiv:1909.03209*, 2019.

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

Dongxia Wu, Matteo Chinazzi, Alessandro Vespignani, Yi-An Ma, and Rose Yu. Multi-fidelity hierarchical neural processes. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2029–2038, 2022.

Zhijie Wu, Yuhe Jin, and Kwang Moo Yi. Neural fourier filter bank. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 14153–14163, 2023.

Muyu Xu, Fangneng Zhan, Jiahui Zhang, Yingchen Yu, Xiaoqin Zhang, Christian Theobalt, Ling Shao, and Shijian Lu. Wavenerf: Wavelet-based generalizable neural radiance fields. In *IEEE International Conference on Computer Vision*, pages 18195–18204, 2023.

Zesheng Ye and Lina Yao. Contrastive conditional neural processes. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022.

Florian Yger and Alain Rakotomamonjy. Wavelet kernel learning. *Pattern Recognition*, 44(10-11): 2614–2629, 2011.

Jaesik Yoon, Gautam Singh, and Sungjin Ahn. Robustifying sequential neural processes. In *International Conference on Machine Learning*, 2020.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. In *Advances in Neural Information Processing Systems*, 2017.

Liang Zhao, Feng Chen, Chang-Tien Lu, and Naren Ramakrishnan. Multi-resolution spatial event forecasting in social media. In *International Conference on Data Mining*, 2016.

A. Appendix: Notation table

For most notations, we follow the CNP family. Below, we list the important notations used in the paper and a description of their meaning.

Table 4: Notation table

Notation	Description
$x \in \mathcal{X}/\mathbb{R}^d$	data input
$y \in \mathcal{Y}/\mathbb{R}^{d'}$	data output
$\hat{x}$	grid input
$\mathcal{Z}_M$	the collection of all M-size sets of input-output pairs.
$Z = \{(\mathbf{x_c}, \mathbf{y_c})\} \in \mathcal{Z}_M$	a context set with size M
$\mathbf{x_t}$	target input
θ	model parameters
ker	kernel function
E	an encoder
$\kappa \in \mathbb{R}^e$	representation of a context set
h	functional representation of a context set
ρ	mapping from representation to mean and variance
$C_b(\mathcal{X}, \mathcal{Y})$	the space of continuous, bounded functions $\mathcal{X} \to \mathcal{Y}$ endowed with the supremum norm
η	translation-equivariant CNN mapping
J	wavelet decomposition level
ϕ	scaling function
ψ	wavelet function
$(r^j)_{j\in\mathbb{Z}}$ $(r > 1)$	resolution sequence, we choose $r = 2$ in this paper
W_ϕ, W_ψ	approximation coefficients and detailed coefficients
ξ^h, ξ^l	high-pass and low-pass filters of wavelet decomposition
ν	permutation

B. Appendix: Training algorithm

Algorithm 1 WaveCNP training

Require: image: I, context M_c, target M_t

while θ not converge **do**
  $Z_c \leftarrow M_c \odot I$
  $h \leftarrow \mathrm{CONV}_\theta([M_c, Z_c]^T)$
  $h^{1:C} \leftarrow h^{1:C}/h^{(0)}$
  $W_\phi, W_\psi \leftarrow \mathrm{DWT}(h)$ ▷ wavelet transform
  $f_t = \mathbf{0}$
  for $j = 1 : J$ **do**
    $W_\psi^j = 0$
    $h_j \leftarrow \mathrm{IWT}(W_\phi, W_\psi)$ ▷ inverse wavelet transform
    $h_j \leftarrow M_t \odot \eta_\theta^j(h_j)$
    $f_t \leftarrow f_t + h_j$
  end for
  $\mu, \sigma \leftarrow \rho_\theta(f_t)/J$
end while

C. Appendix: Related work

Neural processes (Garnelo et al., 2018a,b; Jha et al., 2022) are a family of stochastic processes with strong generalization ability and scalability. Since they first appeared, there have been various interesting extensions with different aims.

One group of works targets the sequential data, where long short-term memory networks (LSTM) and probabilistic state space model are respectively embodied in Recurrent NP (Willi et al., 2019) and Sequential NP (Singh et al., 2019) to unveil dynamical patterns from sequential data and deal with non-stationarity.

Inspired by the translation equivariance of Gaussian process (GP) under a stationary kernel, one group focused on achieving translation equivariance, which is believed to be an important property for modeling real-world data in practice. As introduced in Section 2.2, ConvCNP (Gordon et al., 2020) was proposed to map a (context) set into a function space rather than a vector space to achieve translation equivariance. A similar idea was also adopted to achieve more general group equivariance in EquivCNP (Kawano et al., 2020) based on group convolutions (like LieConv (Finzi et al., 2020)). Relational Conditional Neural Processes (RCNP) (Huang et al., 2023) was proposed to resolve the input dimensionality limitation. At the same time, SteerCNP (Holderrieth et al., 2021) was also proposed for the general group equivariance, not only for scalar field input but also vector-valued field input.

Another group focused on using attention to encode input similarity, which is proven to increase the predictive performance. The first work in this group was on the ANP (Kim et al., 2019) and its memory-efficient variant (Feng et al., 2024), which builds self-attention between context points and cross-attention between context and target points. Such an idea was further extended by the transformer neural process (TNP) (Nguyen and Grover, 2022) and its translation-equivalence variant (Ashman et al., 2024) which borrows the powerful and complex attention structure of the transformer. Multi-fidelity hierarchical neural processes (MF-HNP) (Wu et al., 2022) is another interesting work in this stream with a roughly similar idea to ours, although the multi-fidelity data is different from our multi-resolution data. The input for their model includes specific fidelity as an additional input associated with each data point, but our model does not assume such 'resolution labels' are given. The same problem happens in SNP (Singh et al., 2019) and Attentive Sequential Neural Processes (ASNP) (Yoon et al., 2020) which proposed a Recurrent Memory Reconstruction (RMR) mechanism to not only resolve the underfitting problem but also improve the robustness under task shift.

There is also a group targeting modeling of the output dependency because the classical CNP (Garnelo et al., 2018a) assumes a (also known as 'mean-field') factorized predictive distribution on each target input whose corresponding outputs are assumed to be independent. To model the output dependency, neural process (NP) (Garnelo et al., 2018b) introduces a latent variable to the encoder and all the predictions are conditioned on such variable and then dependent on each other. A similar operation is also applied to ConvCNP (Gordon et al., 2020) by ConvNP (Foong et al., 2020), which introduces a latent function instead of a latent variable. However, such an idea leads to complicated model inference. The Gaussian Neural Processes (GNP) (Bruinsma et al., 2020) resolves this problem by minimizing the Kullback–Leibler (KL) divergence between a real posterior predictive map and an approximated Gaussian process, where the positive semi-definite matrix is from the CNP or ConvCNP encoder (without vector or function representation). The efficiency is further improved by fixing the kernel form and extended to non-Gaussian predictions by a Gaussian copula Neural Process (GCNP) (Markou et al., 2022).

Furthermore, there are still more interesting extensions, such as bootstrapping for functional uncertainty (Lee et al., 2020), constrastive learning for noisy observations (Ye and Yao, 2022), and Wasserstein distance (Carr et al., 2019). However, all the above models are ignorant of the resolution issues.

From the perspective of coordinate-based regression, recent advancements in neural radiance fields (NRF) (Wu et al., 2023; Xu et al., 2023) have also explored the benefits of wavelet representation to boost predictive accuracy and address representation limitations. These are interesting works on using wavelet transformation and representation although not for neural processes. In general, NRF is a different task compared with NP; the wavelet transformation is applied to the raw images in NRF, whereas our model applies it to the functional representation of the context set.

D. Appendix: Proof of three Corollaries in Section 3

Proof. (For **Corollary 1**) Let

$$h_Z(\hat{x}_i) = \sum_{c=1}^{N} \varphi_{K+1}(y_c)\mathrm{ker}(\hat{x} * i, x_c).$$

For a permuted input set νZ, we have

$$h_{\nu Z}(\hat{x}_i) = \sum_{c=1}^{N} \varphi_{K+1}(y_{\nu(c)})\mathrm{ker}(\hat{x}_i, x_{\nu(c)}).$$

Since ν is a permutation of $1, \ldots, N$, the index change $r = \nu(c)$ gives

$$h_{\nu Z}(\hat{x}_i) = \sum_{r=1}^{N} \varphi_{K+1}(y_r)\mathrm{ker}(\hat{x}_i, x_r) = h_Z(\hat{x} * i).$$

Thus,

$$h_{\nu Z} = h_Z.$$

Since the multiresolution operator E_j is applied to h_Z, it follows that

$$E_j(h_{\nu Z}) = E_j(h_Z), \qquad j = 1, \ldots, J.$$

Therefore,

$$\frac{1}{J}\sum_{j=1}^{J} \eta_j(E_j(h_{\nu Z})) = \frac{1}{J}\sum_{j=1}^{J} \eta_j(E_j(h_Z)).$$

Applying the same function ρ to both sides yields

$$\Phi(\nu Z) = \rho\left(\frac{1}{J}\sum_{j=1}^{J} \eta_j(E_j(h_{\nu Z}))\right) = \rho\left(\frac{1}{J}\sum_{j=1}^{J} \eta_j(E_j(h_Z))\right) = \Phi(Z).$$

Hence, WaveDeepSet is permutation-invariant. □

Proof. (For **Corollary 2**) Let

$$h_Z(x) = \sum_{c=1}^{N} \varphi_{K+1}(y_c)\mathrm{ker}(x, x_c).$$

For the translated input set $T_\tau Z$, we have

$$h_{T_\tau Z}(x) = \sum_{c=1}^{N} \varphi_{K+1}(y_c)\mathrm{ker}(x, x_c + \tau).$$

By the translation-invariance of the kernel, we have

$$\ker(x, x_c + \tau) = \ker(x - \tau, x_c).$$

Therefore,

$$\begin{aligned} h_{T_\tau Z}(x) &= \sum_{c=1}^{N} \varphi_{K+1}(y_c)\ker(x - \tau, x_c) \\ &= h_Z(x - \tau) \\ &= (T'_\tau h_Z)(x). \end{aligned}$$

Hence,

$$h_{T_\tau Z} = T'_\tau h_Z.$$

Since each multiresolution operator E_j is translation-equivariant, we obtain

$$E_j(h_{T_\tau Z}) = E_j(T'_\tau h_Z) = T'_\tau E_j(h_Z), \qquad j = 1, \ldots, J.$$

Since each mapping function η_j is translation-equivariant, it follows that

$$\eta_j(E_j(h_{T_\tau Z})) = \eta_j(T'_\tau E_j(h_Z)) = T'_\tau \eta_j(E_j(h_Z)).$$

Taking the average over all resolution levels gives

$$\frac{1}{J}\sum_{j=1}^{J} \eta_j(E_j(h_{T_\tau Z})) = \frac{1}{J}\sum_{j=1}^{J} T'_\tau \eta_j(E_j(h_Z)) = T'_\tau \left(\frac{1}{J}\sum_{j=1}^{J} \eta_j(E_j(h_Z)) \right),$$

where the last equality follows from the linearity of T'_τ.

Finally, using the translation-equivariance of ρ, we have

$$\begin{aligned} \Phi(T_\tau Z) &= \rho\left(\frac{1}{J}\sum_{j=1}^{J} \eta_j(E_j(h_{T_\tau Z})) \right) \\ &= \rho\left(T'_\tau \left(\frac{1}{J}\sum_{j=1}^{J} \eta_j(E_j(h_Z)) \right) \right) \\ &= T'_\tau \rho\left(\frac{1}{J}\sum_{j=1}^{J} \eta_j(E_j(h_Z)) \right) \\ &= T'_\tau \Phi(Z). \end{aligned}$$

Therefore, WaveDeepSet is translation-equivariant. □

Proof. (For **Corollary 3**) Since the wavelet representation is assumed to be orthonormal, the finest resolution space V_J admits the orthogonal decomposition

$$V_J = V_{j_0} \oplus \bigoplus_{r=j_0}^{J} \mathcal{W}_r.$$

Therefore, any $h \in V_J$ can be uniquely expanded as

$$h = \frac{1}{\sqrt{N}} \left(\sum_k W_\phi^{j_0}[k]\phi_k^{j_0} + \sum_{r=j_0}^{J} \sum_k W_\psi^r[k]\psi_k^r \right).$$

For each $j = j_0, \ldots, J$, the cumulative resolution space is

$$V_j = V_{j_0} \oplus \bigoplus_{r=j_0}^{j} \mathcal{W}_r.$$

The operator E_j in Definition 2 keeps the coarse component in $V * j_0$ and the detail components up to level j:

$$E_j(h) = \frac{1}{\sqrt{N}} \left(\sum_k W_\phi^{j_0}[k]\phi_k^{j_0} + \sum_{r=j_0}^{j} \sum_k W_\psi^r[k]\psi_k^r \right).$$

Since the decomposition is orthonormal, this is exactly the orthogonal projection of h onto V_j. Hence,

$$E_j(h) = P_{V_j} h.$$

Now let $h^{(\ell)} \in V_\ell$ be a task generated at resolution level ℓ. Since

$$V_\ell = V_{j_0} \oplus \bigoplus_{r=j_0}^{\ell} \mathcal{W}_r,$$

we can write

$$h^{(\ell)} = \frac{1}{\sqrt{N}} \left(\sum_k W_\phi^{j_0}[k]\phi_k^{j_0} + \sum_{r=j_0}^{\ell} \sum_k W_\psi^r[k]\psi_k^r \right).$$

For any finer level $r > \ell$, the subspace $\mathcal{W}_r$ is orthogonal to V_ℓ. Therefore, for every basis function $\psi_k^r \in \mathcal{W}_r$, we have

$$\left\langle h^{(\ell)}, \psi_k^r \right\rangle_{\mathcal{H}^{L^2}} = 0.$$

By the definition of the wavelet detail coefficient,

$$W_\psi^r[k] = \left\langle h^{(\ell)}, \psi_k^r \right\rangle_{\mathcal{H}^{L^2}},$$

and hence

$$W_\psi^r[k] = 0, \qquad r > \ell.$$

Equivalently,

$$P_{\mathcal{W}_r} h^{(\ell)} = 0, \qquad r > \ell.$$

Since $E_j = P_{V_j}$, we have

$$E_j(h^{(\ell)}) = P_{V_j} h^{(\ell)}.$$

If $j \geq \ell$, then $V_\ell \subseteq V_j$. Since $h^{(\ell)} \in V_\ell$, projecting $h^{(\ell)}$ onto V_j leaves it unchanged:

$$E_j(h^{(\ell)}) = P_{V_j} h^{(\ell)} h^{(\ell)}.$$

If $j < \ell$, then $E_j(h^{(\ell)}) = P_{V_j} h^{(\ell)}$ is the coarser projection of $h^{(\ell)}$ onto V_j.

Thus, a task generated at resolution level ℓ has no components in finer detail subspaces $\mathcal{W}_r$ for $r > \ell$, and its representations under $\{E_j\}_{j=j_0}^J$ are aligned through the nested multiresolution spaces $\{V_j\}_{j=j_0}^J$. □

E. Appendix: More details of experiments

All experiments are conducted on two Tesla V100 (32G) GPUs.

Table 5: Hyperparameters for 1-D synthetic regression

Name	**Value**
epoch number	200
batch size	16
grid point number	256
latent dimension	128
learning rate	1e-3
weight decay	1e-5
random seeds	123, 124, 125, 126, 127

Table 6: Hyperparameters for 2-D image completion

Name	**Value**
epoch number	200
batch size	16
context set number	100
target set number	300
latent dimension	128
learning rate	1e-3
weight decay	1e-5
random seeds	123, 124, 125, 126, 127

E.1. 1-D synthetic regression

The context and target numbers are random numbers from $2^{\lfloor \log_2(i) \rfloor}, i \sim \text{Uniform}[3, 50]$. The other hyperparameters for all models are listed in Table 5. The architectures for all models are as follows: For **WaveCNP**, the network of each η is Conv1d-ReLU-Conv1d-ReLU-Conv1d-ReLU; For **ConvCNP**, the CNN network is Conv1d-ReLU-Conv1d-ReLU-Conv1d-ReLU-Conv1d-ReLU-Conv1d-ReLU-Conv1d-ReLU-Conv1d-ReLU-Conv1d-ReLU-Conv1d-ReLU-Conv1d-ReLU; For **CNP**, the encoder network is BatchLinear-ReLU-BatchLinear-ReLU-BatchLinear-ReLU-BatchLinear-ReLU-BatchLinear-ReLU-BatchLinear-ReLU-BatchLinear-ReLU-BatchLinear, and the decoder network is BatchLinear-ReLU-BatchLinear-ReLU-BatchLinear; For **AttnCNP**, the encoder and decoder networks are same with CNP but with additional cross attention. The neural numbers of the above architectures were chosen to ensure that the model sizes are comparable and the WaveCNP was with the smallest size among all models (as shown in Table 1 of the paper).

E.2. 2-D image completion

The context and target numbers are randomly sampled from the whole image, and the other hyperparameters for all models are listed in Table 6. For simplicity, we transformed images from all datasets to grayscale images (single output channel). The architectures for all models are as follows: For **WaveCNP**, the network of each η is 3 $\times$ ResConvBlock; For **ConvCNP**, the CNN network is 6 $\times$ ResConvBlock; For **CNP**, the encoder network is BatchLinear-ReLU-BatchLinear-ReLU-BatchLinear-ReLU-BatchLinear-ReLU-BatchLinear-ReLU-BatchLinear-ReLU-BatchLinear-ReLU-BatchLinear, and the decoder network is BatchLinear-ReLU-BatchLinear-ReLU-BatchLinear; For **AttnCNP**, the encoder network is BatchLinear-ReLU-BatchLinear with cross attention, and decoder network is same with CNP. The neural numbers of the above architectures were chosen to ensure that the model sizes are comparable and the WaveCNP was with the smallest size among all models (as shown in Table 2 of the paper).

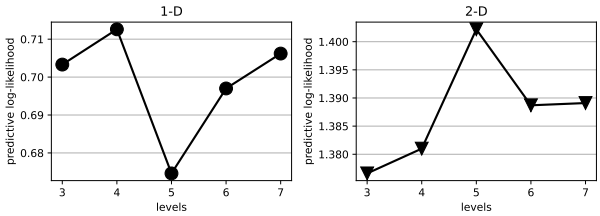

Figure 5: Effect of level on the WaveCNP performance (predictive log-likelihood) on two benchmark tasks

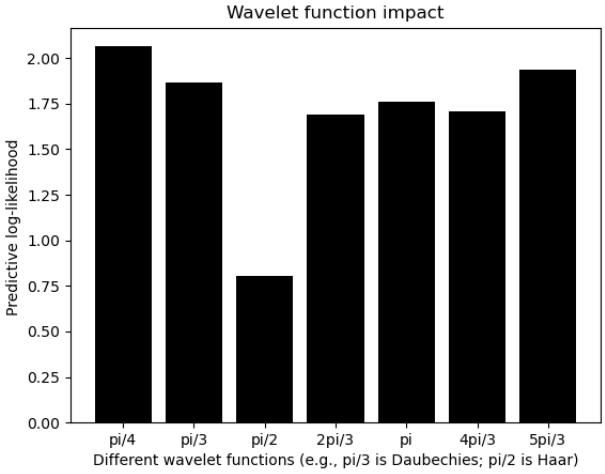

Figure 6: Effect of different wavelet functions

Here, we used 1,000 images/tasks for training rather than following the standard training-validation-testing split of these datasets. This choice is motivated by the meta-learning nature of conditional neural processes. In neural process models, each image is treated as a separate task, where the model observes only a subset of context points and is expected to predict the remaining target points. Therefore, the key objective is not simply to fit a large collection of images, but to evaluate whether the model can generalize to unseen tasks after learning from a limited set of training tasks. If all available training images were used, the experimental setting would become closer to conventional supervised learning and would weaken the assessment of task-level generalization. By restricting the training set to 1,000 images/tasks, we create a more challenging and appropriate meta-learning setting in which the model must learn transferable functional representations from a limited number of tasks and then adapt to previously unseen test images from only a small number of observed context points.

The batch size for EquivCNP is different from the others, set to 2 in our experiments. The reason is that this model is highly memory inefficient because some large matrices are kept in memory in their implementation. Therefore, we have to use a smaller batch size in our experiments.

F. Appendix: Ablation study

One hyperparameter of WaveCNP is the wavelet decomposition level. To show its effect on the final performance, we tested several options, from 3 to 7. The results are shown in Fig. 5, where 1D (Matern kernel) and 2D (CIFAR) datasets were used. We can observe that 1) there is no general trend pattern but only a fluctuation, and 2) there is not much difference across different levels. The

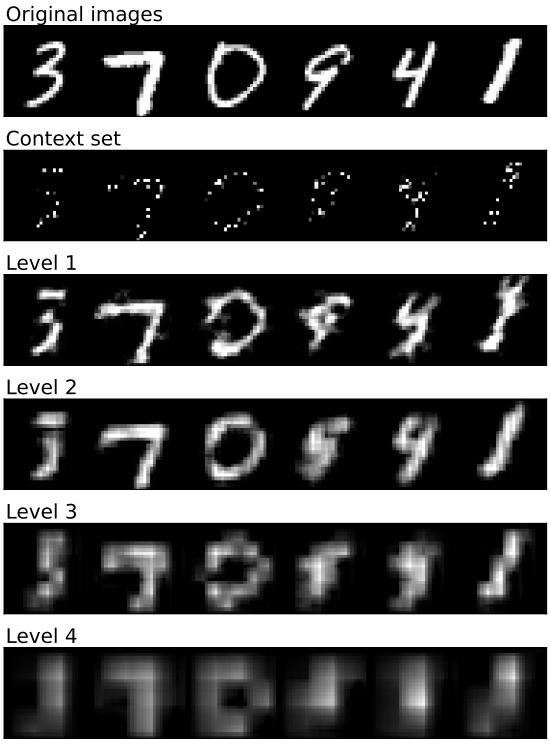

Figure 7: Multiresolution prediction by WaveCNP

other hyperparameter is the wavelet function selection. Even though we have given an adaptive strategy to avoid this selection, we also show the impact of different wavelet functions here. The results in Fig. 6 show the predictive log-likelihood of our algorithm (with fixed wavelet function) using the MNIST dataset, where different wavelet functions are from different β (note that some are with names but some are not).

G. Appendix: Multiresolution prediction

Multiresolution prediction is significant in various domains. One example is traffic management at various resolutions Hadi et al. (2022), where each resolution has specific advantages and disadvantages and can provide a different function in the management process. Macroscopic traffic predictions can assess regional transportation demand patterns involving large spatial scopes, while microscopic predictions are useful for studying operations and localized issues with limited spatial scopes. Other examples include, but are not limited to, stock market (Althelaya et al., 2021), wind speed (Liu et al., 2020), and social media events (Zhao et al., 2016). In this section, we show the capability of our WaveCNP on multiresolution prediction. WaveCNP maps the context set to a multiresolution function space, so we built the loss function for all resolutions of the image during training. Given a context set, the multiresolution predictions are shown in Fig. 8. The multiresolution prediction on MNIST dataset is shown in Fig. 7.

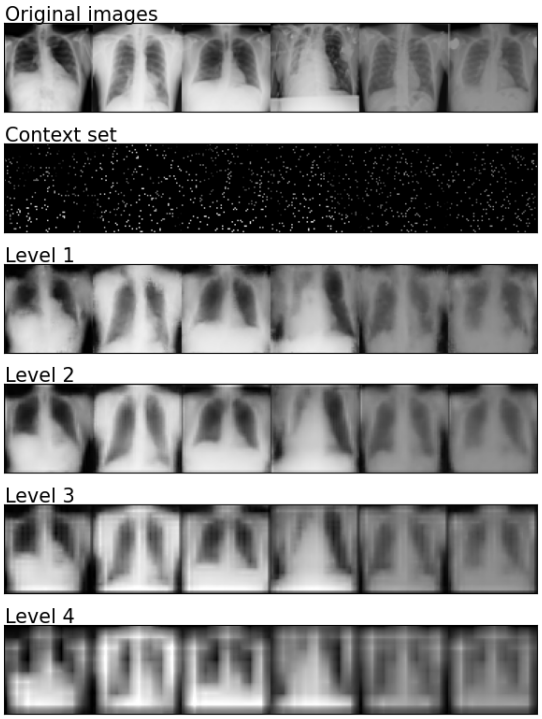

Figure 8: Multiresolution prediction by WaveCNP

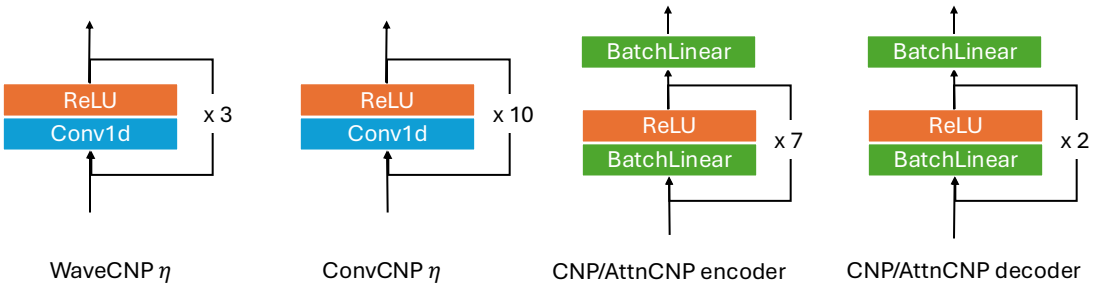

Figure 9: Visualizations of the model architectures for 1-D synthetic regression

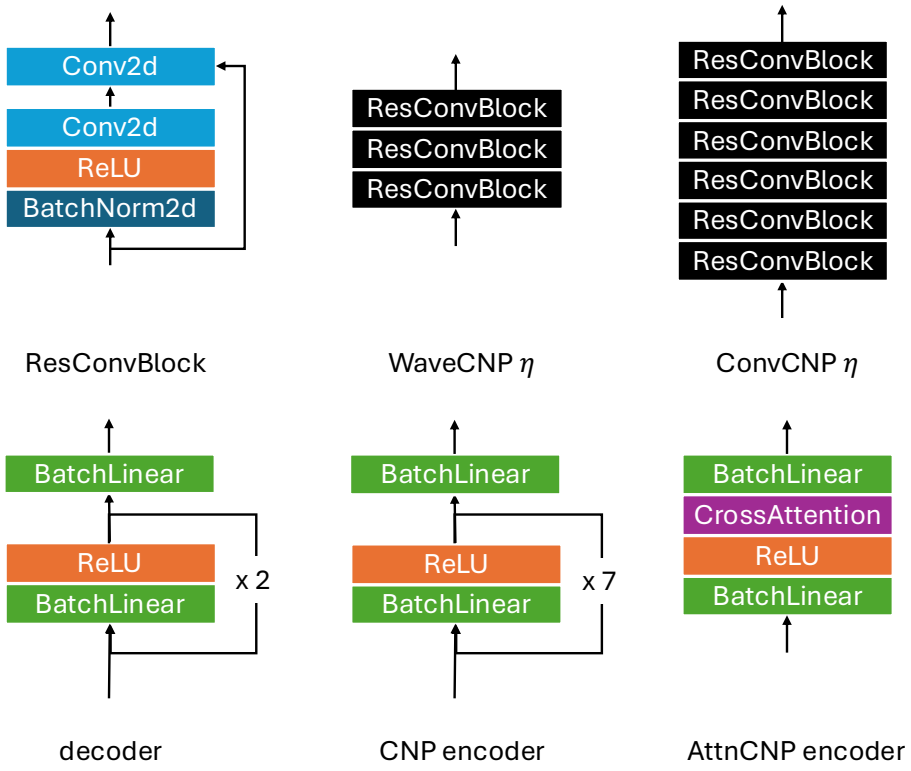

Figure 10: Visualizations of the model architectures for 2-D image completion