Equivariant Learning of Stochastic Fields:
Gaussian Processes and Steerable Conditional Neural Processes

Peter Holderrieth * 1  Michael Hutchinson * 1  Yee Whye Teh 1,2

Abstract
Motivated by objects such as electric fields or fluid streams, we study the problem of learning stochastic fields, i.e. stochastic processes whose samples are fields like those occurring in physics and engineering. Considering general transformations such as rotations and reflections, we show that spatial invariance of stochastic fields requires an inference model to be equivariant. Leveraging recent advances from the equivariance literature, we study equivariance in two classes of models. Firstly, we fully characterise equivariant Gaussian processes. Secondly, we introduce Steerable Conditional Neural Processes (SteerCNPs), a new, fully equivariant member of the Neural Process family. In experiments with Gaussian process vector fields, images, and real-world weather data, we observe that SteerCNPs significantly improve the performance of previous models and equivariance leads to improvements in transfer learning tasks.

1. Introduction
In physics and engineering, fields are objects which assign a physical quantity to every point in space. They serve as a unifying concept for objects such as electro-magnetic fields, gravity fields, electric potentials or fluid dynamics and are therefore omnipresent in the natural sciences and their applications (Landau, 2013). Our goal in this work is to be able to predict the value of a field everywhere given some finite set of observations (see \( f \) in fig. 1). We will be interested in cases where these fields are not fixed but are drawn from a random distribution, which we term stochastic fields.

Viewing fields as usual mathematical functions, we can consider stochastic fields as stochastic processes. A well-known example of these are Gaussian processes (GPs) (Rasmussen & Williams, 2005) which have been widely used in machine learning. More recently, Neural Processes (NPs) and their related models were introduced as an alternative to GPs which enable to learn a stochastic process from data leveraging the flexibility of neural networks (Garnelo et al., 2018a;b; Kim et al., 2019).

When applying models to data from the natural sciences, it is a logical step to integrate scientific knowledge about the problem into these models. The physical principle of homogeneity of space states that all positions and orientations in space are equivalent, i.e. there is no canonical orientation or absolute position. A natural modelling assumption stemming from this is that the prior we place over a random field should be invariant, i.e. it should look the same from all positions and orientations. We will show that this implies that the posterior as a function of the observed points is equivariant.

Figure 1 illustrates this equivariance. The input is a discrete set of vectors at certain points in space, the red arrows, and the predictions a continuous vector field. If the space is...
homogenous, we would expect the predictions of a model to rotate in the same way as we rotate the data, i.e. we expect the model to be equivariant.

Translation equivariance in Gaussian processes has long been studied via stationary kernels. We extend this notion to derive Gaussian processes that are equivariant to more general transformations such as rotations and reflections. Recent work has also shown how to build translation equivariance into NP models (Gordon et al., 2020; Bruinsma et al., 2021). We will build on this work to introduce a new member of the Neural Process family that has more general equivariance properties. We do this utilising recent developments in equivariant deep learning (Weiler & Cesa, 2019). Imposing equivariance on deep learning models reduces the number of model parameters and has been shown to derive Gaussian processes that are equivariant to more general transformations such as rotations and reflections.

We describe all possible transformations of a feature map $F$ by a subgroup $G \subset \mathbb{E}(n)$. We assume that $G$ is the semidirect product of $T(n)$ and a subgroup $H$ of $O(n)$, i.e. every $g \in G$ is a unique composition of a translation $t_x$ and an orthogonal map $h \in H$:

$$g = t_x h$$  \hspace{1cm} (1)

As common, we call $H$ the fiber group. Depending on the inference problem, one would often pick $H = SO(n)$ or $H = O(n)$ (equivalently $G = SE(n),\mathbb{E}(n)$). However, using finite subgroups $H$ can be more computationally efficient, and give better empirical results (Weiler & Cesa, 2019). In particular, in dimension $n = 2$ we use the Cyclic groups $C_m$, comprised of the rotations by $\frac{k\pi}{m}$ ($k = 0, 1, \ldots, m-1$) and the dihedral group $D_m$ containing $C_m$ combined with reflections.

To describe transformations of a feature map $F$ via a group $G$, we need a linear representation $\rho : H \rightarrow \text{GL}(\mathbb{R}^d)$ of $H$ which we call fiber representation. The action of $G$ on a steerable feature map $F$ is then defined as

$$g.F(x) = \rho(h)F(g^{-1}x)$$  \hspace{1cm} (2)

where $g = t_xh, h \in G$. Figure 2 demonstrates for vector fields why the transformation defined here is a sensible notion to consider. In group theory, this is called the induced representation of $H$ on $G$ denoted by $\text{Ind}_G^H \rho$.

In allusion to physics, we use the term (steerable) feature field referring to the feature map $F : \mathbb{R}^n \rightarrow \mathbb{R}^d$ together with its corresponding law of transformation given by $\rho$ (Weiler & Cesa, 2019). We write $F_\rho$ for the space of these fields. Typical examples are:

1. **Scalar fields** $F : \mathbb{R}^n \rightarrow \mathbb{R}$ have trivial fiber representation $\rho = \rho_{\text{triv}}$, i.e. $\rho(h) = 1$ for $h \in H$, such that

$$g.F(x) = F(g^{-1}x)$$  \hspace{1cm} (3)

Examples are greyscale images or temperature maps.

2. **Vector fields** $F : \mathbb{R}^n \rightarrow \mathbb{R}^d$ have $\rho = \rho_{\text{Id}}$, i.e. $\rho(h) = h$ for $h \in H$, such that

$$g.F(x) = hF(g^{-1}x)$$  \hspace{1cm} (4)

Examples include electric fields or wind maps.

3. **Stacked fields**: given fields $F_1, \ldots, F_n$ with fiber representations $\rho_1, \ldots, \rho_n$ we can stack them to $F = (F_1, \ldots, F_n)$ with fiber representation as the direct sum $\rho = \rho_1 \oplus \cdots \oplus \rho_n$. Examples include a combined wind and temperature map or RGB-images.

Finally, for simplicity we assume that $\rho$ is an orthogonal representation, i.e. $\rho(h) \in O(d)$ for all $h \in H$. Since all fiber groups $H$ of interest are compact, this is not a restriction (Serre, 1977).
3. Equivariant Stochastic Process Models

In this work, we are interested in learning not only a single feature field \( F \) but a probability distribution \( P \) over \( \mathcal{F}_P \), i.e. a stochastic process over feature fields \( F \). For example, \( P \) could describe the distribution of all wind maps over a specific region. If \( F \sim P \) is a random feature field and \( g \in G \), we can define the transformed stochastic process \( g.P \) as the distribution of \( g.F \). We say that \( P \) is \( G \)-invariant if

\[
P = g.P \quad \text{for all} \quad g \in G \quad (5)
\]

From a sample \( F \sim P \), our model observes only a finite set of input-output pairs \( Z = \{ (x_i, y_i) \}_{i=1}^{n} \) where \( y_i \) equals \( F(x_i) \) plus potentially some noise. The induced representation naturally translates to a transformation of \( Z \) under \( G \) via

\[
g.Z := \{ (gx_i, \rho(h)y_i) \}_{i=1}^{n} \quad (6)
\]

In a Bayesian approach, we can consider \( P \) as a prior and given an observed data set \( Z \) we can consider the posterior, i.e. the conditional distribution \( P_Z \) of \( F \) given \( Z \). As the next proposition shows, equivariance of the posterior is the other side of the coin to invariance of the prior.

**Proposition 1.** Let \( P \) be a stochastic process over \( \mathcal{F}_P \). Then \( P \) is \( G \)-invariant if and only if the posterior map \( Z \mapsto P_Z \) is \( G \)-equivariant, i.e.

\[
P_{g.Z} = g.P_Z \quad \text{for all} \quad g \in G \quad (7)
\]

The proof of this can be found in appendix B.1.

In most real-world scenarios, it may not be possible to exactly compute the posterior and our goal is to build a model \( Q \) which returns an approximation \( Q_Z \) of \( P_Z \). However, often it is our prior belief that the distribution \( P \) is \( G \)-invariant. Given proposition 1, it is then natural to construct an approximate inference model \( Q \) which is itself equivariant.

We will see applications of these ideas to GPs and CNPs in sections 4 and 5.

4. Equivariant Gaussian Processes

A widely-studied example of stochastic processes are Gaussian processes (GPs). Here we will look at Gaussian processes under the lens of equivariance. Since we are interested in vector-valued functions \( F : \mathbb{R}^n \to \mathbb{R}^d \), we use matrix-valued positive definite kernels \( K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^{d \times d} \) (Álvarez et al., 2012).

In the case of GPs, we assume that for every \( x, x' \in \mathbb{R}^n \), it holds that \( F(x) \) is normally distributed with mean \( m(x) \) and covariances \( \text{Cov}(F(x), F(x')) = K(x, x') \). We write \( GP(m, K) \) for the stochastic process defined by this.

We can fully characterise all mean functions and kernels leading to equivariant GPs:

**Theorem 1.** A Gaussian process \( GP(m, K) \) is \( G \)-invariant, equivalently the posterior \( G \)-equivariant, if and only if

1. \( m(x) = m \in \mathbb{R}^d \) is constant with \( m \) such that
   \[
   \rho(h)m = m \quad \text{for all} \; h \in H \quad (8)
   \]

2. \( K \) fulfils the following two conditions:
   
   \((a)\) \: \( K \) is stationary, i.e. for all \( x, x' \in \mathbb{R}^n \)
   \[
   K(x, x') = K(x - x', 0) =: \hat{K}(x - x') \quad (9)
   \]
   
   \((b)\) \: \( K \) satisfies the angular constraint, i.e. for all \( x, x' \in \mathbb{R}^n \), \( h \in H \) it holds that
   \[
   K(hx, hx') = \rho(h)K(x, x')\rho(h)^T \quad (10)
   \]
   
   or equivalently, for all \( x \in \mathbb{R}^n, h \in H \) it holds that
   \[
   \hat{K}(hx) = \rho(h)\hat{K}(x)\rho(h)^T \quad (11)
   \]

   If this is the case, we call \( K \) \( \rho \)-equivariant.

The proof of this can be found in appendix B.2.

We note the distinct similarity between the kernel conditions in eqs. (9) and (10), and the ones found in the equivariant convolutional neural network literature (see for example eq. (2) in Weiler & Cesa (2019)). In contrast to convolutional kernels, we have the additional constraint that the kernel \( K \) must be positive definite.

A popular example to model vector-valued functions is to simply use \( d \) independent GPs with a stationary scalar kernel \( k : \mathbb{R}^n \to \mathbb{R} \). This leads to a kernel \( K(x) = k(x)I \) which can easily be seen to be \( E(n) \)-equivariant.

As a non-trivial example of equivariant kernels, we will also consider the divergence-free and curl-free kernels (see appendix C) used in physics introduced by Macêdo & Castro...
We note that the kernels considered in this work represent a vector-valued and equivariant case. Context sets maps are $G$-equivariant if and only if the mean and covariance feature field are $G$-equivariant, i.e. it holds for all $g \in G$ and context sets $Z$

$$m_{g,Z} = g \cdot m_Z$$
$$\Sigma_{g,Z} = g \cdot \Sigma_Z$$

with $\rho_m = \rho$ and $\rho_\Sigma = \rho \otimes \rho$ the tensor product with action given by

$$\rho_\Sigma(h)A = \rho(h)A\rho(h)^T, \quad A \in \mathbb{R}^{d \times d}$$

The proof can be found in appendix B.3.

In the following, we will restrict ourselves to perform inference from data sets of multiplicity 1, i.e., data sets $Z = \{ (x_i, y_i) \}_{i=1}^m$ where $x_i \neq x_j$ for all $i \neq j$. We denote the collection of all such data sets with $\mathcal{Z}_m$ meaning that they transform under $\rho$ (see eq. (6)).

Moreover, we assume that there is no order in a data set $Z$, i.e. we aim to build models which are not only $G$-equivariant but also invariant to permutations of $Z$. The following generalisation of the ConvDeepSets theorem of Gordon et al. (2020) gives us a universal form of all such conditional process models. We simple need to pick $\rho_m = \rho$ and $\rho_{\text{out}} = \rho_m \oplus \rho_\Sigma = \rho \oplus \rho_{\Sigma}$ in the following theorem.

**Theorem 2 (EquivDeepSets).** Let $\rho_m, \rho_{\text{out}}$ be the two fiber representations. Define the embedding representation as the direct sum $\rho_E = \rho_{\text{triv}} \oplus \rho_m$.

A function $\Phi : \mathcal{Z}_m \rightarrow \mathcal{F}_{\rho_{\text{out}}}$ is $G$-equivariant and permutation invariant if and only if it can be expressed as

$$\Phi(Z) = \Psi(E(Z))$$

for all $Z = \{ (x_i, y_i) \}_{i=1}^m \in \mathcal{Z}_m$ with

1. $E(Z) = \sum_{i=1}^m K(\cdot, x_i)\phi(y_i)$
2. $\phi(y) = (1, y)^T \in \mathbb{R}^{d+1}$
3. $K : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^{(d+1) \times (d+1)}$ is a $\rho_E$-equivariant strictly positive definite kernel (see theorem 1).
4. $\Psi : \mathcal{F}_{\rho_E} \rightarrow \mathcal{F}_{\rho_{\text{out}}}$ is a $G$-equivariant function.

Additionally, by imposing extra constraints (see appendix B.4), we can also ensure that $\Phi$ is continuous.
Figure 4. SteerCNP model illustration. (a) Embed the context set into a function. (b) Discretise this embedding on a regular grid. (c) Predict the mean and covariance of the conditional stochastic process on the grid of points. (d) Use kernel smoothing to predict the mean and covariance at target locations.

The proof of this can be found in appendix B.4. Using this, we can start to build SteerCNPs by building an encoder $E$ and a decoder $\Psi$ as specified in the theorem.

The form of the encoder only depends on the choice of a kernel $K$ which is equivariant under $\rho_E$. An easy but effective way of doing this is to pick a kernel $K_0$ which is equivariant under $\rho$ (see section 4) and a scalar kernel $k : \mathbb{R}^n \to \mathbb{R}$ and then use the block-version $K = k \oplus K_0$.

5.1. Decoder

By theorem 2, it remains to construct a $G$-equivariant decoder $\Psi$. To construct such maps, we will use steerable CNNs (Cohen & Welling, 2017; Weiler & Cesa, 2019; Weiler et al., 2018). In theory, a layer of such a network is an equivariant function $\Psi : \mathcal{F}_{\rho_a} \to \mathcal{F}_{\rho_{out}}$ where we are free to choose fiber representations $\rho_a, \rho_{out}$.

Steerable convolutional layers are defined by a kernel $\kappa : \mathbb{R}^n \to \mathbb{R}^{c_{in} \times c_{out}}$ such that the map

$$[\kappa \ast F](x) = \int \kappa(x, x') F(x') dx \quad (17)$$

is $G$-equivariant. These layers serve as the learnable, parameterisable functions.

Steerable activation functions are applied pointwise to $F$. These are functions $\sigma : \mathbb{R}^{c_{in}} \to \mathbb{R}^{c_{out}}$ such that

$$\sigma(\rho_m(h)x) = \rho_{out}(h)\sigma(x) \quad (18)$$

As a decoder of our model, we use a stack of equivariant convolutional layers composed with equivariant activation functions. The convolutions in eq. (17) are computed in a discretised manner after sampling $E(Z)$ on a grid $G \subset \mathbb{R}^n$. Therefore, the output of the neural network will be a discretised version of a function and we use kernel smoothing to extend the output of the network to the whole space.

5.2. Covariance Activation Functions

The output of a steerable neural network has general vectors in $\mathbb{R}^c$ for some $c$ as outputs. Therefore, we need an additional component to obtain (positive definite) covariance matrices in an equivariant way.

We introduce the following concept:

**Definition 1.** An equivariant covariance activation function is a map $\eta : \mathbb{R}^c \to \mathbb{R}^{d \times d}$ together with a fiber representation $\rho_\eta : H \to GL(\mathbb{R}^c)$ such that for all $y \in \mathbb{R}^c$ and $h \in H$

1. $\eta(y)$ is a symmetric, positive semi-definite matrix.
2. $\eta(\rho_\eta(h)y) = \rho_\Sigma(h)\eta(y)$

In our case, we use a quadratic covariance activation function which we define by

$$\eta : \mathbb{R}^{d \times d} \to \mathbb{R}^{d \times d}, \quad \eta(A) = AA^T$$

Considering $A = (a_1, \ldots, a_D) \in \mathbb{R}^{d^2}$ as a vector by stacking the columns, the input representation is then $\rho_\eta = \rho \oplus \cdots \oplus \rho$ as the $d$-times sum of $\rho$. It is straightforward to see that $\eta$ is equivariant and outputs positive semi-definite matrices.

5.3. Full model

Finally, we summarise the architecture of the SteerCNP (see fig. 4):

1. The encoder produces an embedding of a data set $Z$ as a function $E(Z)$.
2. A discretisation of $E(Z)$ serves as input for the decoder, a steerable CNN with input fiber representation $\rho_E$ and output fiber representation $\rho_\Psi$.
3. On the covariance part, we apply the covariance activation function $\eta$.
4. The grid values of the mean and the covariances are extended to the whole space $\mathbb{R}^n$ via kernel smoothing.
We train the model similar to the CNP by iteratively sampling a data set $Z$ and splitting it randomly in a context set $Z_C$ and a target set $Z_T$. The context set $Z_C$ is then passed forward through the SteerCNP model and the mean log-likelihood of the target $Z_T = \{(x_i', y_i')\}_{i=1}^m$ is computed. In brief, we minimise the loss

$$-\mathbb{E}_{Z_C, Z_T \sim p} \left[ \frac{1}{m} \sum_{i=1}^{m} \log \mathcal{N}(y_i'; \mathbf{m}_{Z_C}(x_i'), \Sigma_{Z_C}(x_i')) \right]$$

by gradient descent methods.

In sum, this gives a CNP model, which up to discretisation errors is equivariant with respect to arbitrary transformations from the group $G$ and invariant to permutations.

6. Related Work

Equivariance and symmetries in deep learning. Motivated by the success of the translation-equivariant CNNs (LeCun et al., 1990), there has been a great interest in building neural networks which are equivariant also to more general transformations. Approaches use a wide range of techniques such as convolutions on groups (Cohen et al., 2018; Kondor & Trivedi, 2018; Cohen & Welling, 2016; Hoogeboom et al., 2018; Worrall & Brostow, 2018), cyclic permutations (Dieleman et al., 2016), Lie groups (Finzi et al., 2020a) or phase changes (Worrall et al., 2016). It was in the context of Steerable CNNs and its various generalisations where ideas from physics about fields started to play a more prominent role. (Cohen & Welling, 2017; Weiler et al., 2018; Weiler & Cesa, 2019; Cohen et al., 2019). We use this framework since it allows for modelling of non-trivial transformations of features $F(x)$ via fiber representations $\rho$, as for example necessary to model transformations of vector fields (see eq. (4)).

Gaussian Processes and Kernels. Classical GPs using kernels such as the RBF or Matérn kernels have been widely used in machine learning to model scalar fields (Rasmussen & Williams, 2005). Vector-valued GPs, which allow for dependencies across dimensions via matrix-valued kernels, were common models in geostatistics (Goovaerts et al., 1997) and also played a role in kernel methods (Álvarez et al., 2012). With this work, we showed that many of these GPs are equivariant (see theorem 1) giving a further theoretical foundation for their applicability. In contrast to equivariant GPs, Reisert & Burkhardt (2007) consider the construction of equivariant functions from kernels, arriving at similar equivariance constraints as we do in theorem 1.

Neural Processes. Garnelo et al. (2018a) introduced Conditional Neural Processes (CNPs) as an architecture constructed out of neural networks which learns an approximation of stochastic processes from data. They share the motivation of meta-learning methods (Finn et al., 2017; Andrychowicz et al., 2016) to learn a distribution of tasks instead of only a single task. Neural Processes (NPs) are the latent variable counterpart of CNPs allowing for correlations across the marginals of the posterior. Both CNPs and NPs have been combined with other machine learning concepts, for example attention mechanisms (Kim et al., 2019).

Gordon et al. (2020) were the first to consider symmetries in NPs. Inspired by the prevalence of stationary kernels in the GP literature, they introduced a translation-equivariant NP model, along with a universal characterisation of such models. Our work can be seen as a generalisation of their method. By picking a trivial fiber group $H = \{e\}$ and $K$ as a diagonal RBF-kernel in SteerCNPs (see theorem 2), we get the ConvCNP as a special case of SteerCNPs.

During the development of this work, Kawano et al. (2021) also studied more general equivariance in CNPs, restricting themselves to scalar fields and focusing on Lie Groups. They only compared their approach with previous models on 1d synthetic regression tasks where their approach did not seem to outperform ConvCNPs, and in this case the added symmetry is redundant. In contrast, we focus on general scalar and vector-valued fields with non-trivial transformations $\rho$. As we show in the next section, this "steerable" approach leads to significant performance gains compared to previous models. The decomposition theorem presented in Kawano et al. (2021) can be seen as a special case of theorem 2 in this work by setting $\rho = \rho_{e_{tr}}$.

7. Experiments

Finally, we provide empirical evidence that equivariance is a helpful bias in stochastic process models. We focus on evaluating SteerCNPs since inference with the variety of GPs, which this work shows to be equivariant, has been studied exhaustively.

7.1. Gaussian Process Vector Fields

A common baseline task for CNPs is regression on samples from a Gaussian process (Garnelo et al., 2018a; Gordon et al., 2020), partially because one can directly compare the output of the model with the true posterior. Here, we consider the task of learning 2D vector fields $F: \mathbb{R}^2 \rightarrow \mathbb{R}^2$ which are samples of a Gaussian process $GP(0, K)$ with 3 different $E(2)$-equivariant kernels $K$: the diagonal RBF-kernel, the divergence-free kernel and the curl-free kernel (see fig. 3a and appendix C).

We run extensive experiments comparing the SteerCNP with the CNP and the translation-equivariant ConvCNP. On the SteerCNP, we impose various levels of rotation and reflection equivariance by picking different fiber groups $H$. As usual for CNPs, we use the mean log-likelihood as a measure of performance. The maximum possible log-likelihood of the target $Z$ forward through the SteerCNP model and the mean log-likelihood of the target $Z_T = \{(x_i', y_i')\}_{i=1}^m$ is computed. In brief, we minimise the loss

$$-\mathbb{E}_{Z_C, Z_T \sim p} \left[ \frac{1}{m} \sum_{i=1}^{m} \log \mathcal{N}(y_i'; \mathbf{m}_{Z_C}(x_i'), \Sigma_{Z_C}(x_i')) \right]$$

by gradient descent methods.

In sum, this gives a CNP model, which up to discretisation errors is equivariant with respect to arbitrary transformations from the group $G$ and invariant to permutations.
Table 1. Results for experiments on GP vector fields. Mean log-likelihood ± 1 standard deviation over 5 random seeds reported. Last row shows GP-baseline (optimum).

<table>
<thead>
<tr>
<th>Model</th>
<th>RBF</th>
<th>Curl-free</th>
<th>Div-free</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNP</td>
<td>-4.24±0.00</td>
<td>-0.750±0.004</td>
<td>-0.752±0.006</td>
</tr>
<tr>
<td>ConvCNP</td>
<td>-3.88±0.01</td>
<td>-0.541±0.004</td>
<td>-0.533±0.001</td>
</tr>
<tr>
<td>SteerCNP (SO(2))</td>
<td>-3.93±0.05</td>
<td>-0.550±0.005</td>
<td>-0.552±0.008</td>
</tr>
<tr>
<td>SteerCNP (C4)</td>
<td>-3.66±0.00</td>
<td>-0.461±0.003</td>
<td>-0.464±0.007</td>
</tr>
<tr>
<td>SteerCNP (C8)</td>
<td>-3.70±0.01</td>
<td>-0.479±0.004</td>
<td>-0.478±0.004</td>
</tr>
<tr>
<td>SteerCNP (C16)</td>
<td>-3.71±0.02</td>
<td>-0.476±0.005</td>
<td>-0.480±0.008</td>
</tr>
<tr>
<td>SteerCNP (D4)</td>
<td>-3.72±0.03</td>
<td>-0.471±0.002</td>
<td>-0.477±0.005</td>
</tr>
<tr>
<td>SteerCNP (D8)</td>
<td>-3.68±0.03</td>
<td>-0.462±0.005</td>
<td>-0.467±0.008</td>
</tr>
<tr>
<td>GP</td>
<td>-3.50</td>
<td>-0.410</td>
<td>-0.411</td>
</tr>
</tbody>
</table>

The results in table 1 show the significant benefit of the additional rotation equivariance built into the models on a variant, called rotMNIST, produced by randomly rotating each digit in the dataset. We find that this does not improve results in the non equivariant models, in fact causing a decrease in performance as they try to learn a more complex distribution from too little data. It therefore seems to be the gain in parameter and data efficiency due to the imposed equivariance constraints what leads to better performance of SteerCNPs.

Table 2. Results for the MNIST experiments. Mean log-likelihood ± 1 standard deviation over 3 random model and dataset seeds reported.

<table>
<thead>
<tr>
<th>Train dataset</th>
<th>Test dataset</th>
<th>MNIST</th>
<th>rotMNIST</th>
<th>MNIST</th>
<th>extMNIST</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>MNIST</td>
<td>rotMNIST</td>
<td>MNIST</td>
<td>extMNIST</td>
<td></td>
</tr>
<tr>
<td>GP</td>
<td>0.39±0.30</td>
<td>0.39±0.30</td>
<td>0.72±0.17</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CNP</td>
<td>0.76±0.05</td>
<td>0.66±0.06</td>
<td>-1.11±0.06</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ConvCNP</td>
<td>1.01±0.01</td>
<td>0.95±0.01</td>
<td>1.08±0.02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SteerCNP(C4)</td>
<td>1.05±0.02</td>
<td>1.02±0.03</td>
<td>1.14±0.02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SteerCNP(C8)</td>
<td>1.07±0.03</td>
<td>1.05±0.04</td>
<td>1.16±0.03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SteerCNP(C16)</td>
<td>1.08±0.03</td>
<td>1.05±0.03</td>
<td>1.17±0.05</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SteerCNP(D4)</td>
<td>1.08±0.03</td>
<td>1.05±0.03</td>
<td>1.14±0.03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SteerCNP(D8)</td>
<td>1.08±0.03</td>
<td>1.04±0.04</td>
<td>1.17±0.02</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

7.2. Image in-painting

To test the SteerCNP on purely scalar data we evaluate our model on an image completion task. In image completion tasks the context set is made up of pairs of 2D pixel locations and pixel intensities and the objective is to predict the intensity at new locations. For further details see appendix D.3, along with qualitative results from the experiments.

MNIST and rotMNIST. We first train models on completion tasks from the MNIST data set (LeCun et al., 2010). The results in table 4 show that the equivariance built into the SteerCNP is useful, with the various SteerableCNPs outperforming previous models. To test whether this performance gain could be replicated by data augmentation, we trained the models on a variant, called rotMNIST, produced by randomly rotating each digit in the dataset. We find that this does not improve results in the non equivariant models, in fact causing a decrease in performance as they try to learn a more complex distribution from too little data. It therefore seems to be the gain in parameter and data efficiency due...
weather map consisting of temperature, pressure and wind in the region at one single point in time. We give the models the task to infer a wind vector field from a data set \( Z \) of pairs \((x, y)\) where \( y = (y^x, y^p, y^w) \in \mathbb{R}^4 \) gives the temperature, pressure and wind at point \( x \). In particular, the output features are only a subset of the input features. To deal with such a task, we can simply pick different input and output fiber representations for the SteerCNP:

\[
\rho_{\text{in}} = \rho_{\text{triv}} \oplus \rho_{\text{triv}} \oplus \rho_{d}, \quad \rho_{\text{out}} = \rho_{d}
\]

**Performance on US data.** As a first experiment, we split the US data set in a train, validation and test data set. Then we train and test the models accordingly. We observe that the SteerCNP outperforms previous models like a GP with RBF-kernel, the CNP and the ConvCNP with a significant margin for all considered fiber groups (see table 3). Again, we observe that a relatively small fiber group \( C_4 \) leads to the best results. Inference from weather data is clearly not exactly equivariant due to local differences such as altitude and distance to the sea. Therefore, it seems that a SteerCNP model with small fiber groups like \( C_4 \) enables us to exploit the equivariant patterns much better than the ConvCNP and CNP but leaves flexibility to account for asymmetric patterns.

**Generalising to a different region.** As a second experiment, we take these models and test their performance on data from China. This can be seen as a transfer learning task. Intuitively, posing a higher equivariance restriction on the model makes it less adapting to special local circumstances and more robust when transferring to a new environment. Indeed, we observe that the CNP, the ConvCNP and the model with fiber group \( C_4 \) have a larger loss in performance than SteerCNP models with larger fiber groups such as \( C_{16}, D_8, D_4 \). Similarly, while GPs had a significantly worse performance than ConvCNP on the US data, it outperforms it on the transfer to China data. In applications like robotics where environments constantly change this robustness due to equivariance might be advantageous.

**Table 3.** Results on ERA5 weather experiment trained on US data. Mean log-likelihood ± 1 standard deviation over 5 random seeds reported. Left: tested on US data. Right: tested on China data.

<table>
<thead>
<tr>
<th>Model</th>
<th>US</th>
<th>China</th>
</tr>
</thead>
<tbody>
<tr>
<td>GP</td>
<td>0.386±0.005</td>
<td>-0.755±0.001</td>
</tr>
<tr>
<td>CNP</td>
<td>0.001±0.017</td>
<td>-2.456±0.365</td>
</tr>
<tr>
<td>ConvCNP</td>
<td>0.898±0.045</td>
<td>-0.890±0.059</td>
</tr>
<tr>
<td>SteerCNP (C_4)</td>
<td>\textbf{1.255±0.019}</td>
<td>-0.578±0.173</td>
</tr>
<tr>
<td>SteerCNP (C_8)</td>
<td>1.038±0.026</td>
<td>-0.582±0.104</td>
</tr>
<tr>
<td>SteerCNP (C_{16})</td>
<td>1.094±0.015</td>
<td>-0.550±0.073</td>
</tr>
<tr>
<td>SteerCNP (D_4)</td>
<td>1.037±0.037</td>
<td>\textbf{0.429±0.067}</td>
</tr>
<tr>
<td>SteerCNP (D_8)</td>
<td>1.032±0.011</td>
<td>-0.539±0.129</td>
</tr>
</tbody>
</table>

**8. Limitations and Future Work**

Similar to CNPs, our model cannot capture dependencies between the marginals of the posterior. Recently, Foong et al. (2020) introduced a translation-equivariant NP model which allows to do this and future work could combine their approach with general symmetries considered in this work.

As stated earlier, our model works for Euclidean spaces of any dimension. The limiting factor is the development of Steerable CNNs used in the decoder. In our experiments, we focused on \( \mathbb{R}^2 \) as the code is well developed by Weiler & Cesa (2019). Recent developments in the design of equivariant neural networks also explore non-Euclidean spaces such as spheres and encourage exploration in this direction in the context of NPs (Cohen et al., 2019; 2018; Esteves et al., 2020). Additionally, in Euclidean space we can incorporate other symmetries such as scaling via Lie group approaches, similar to (Kawano et al., 2021), or symmetry to uniform motion for fluid flow (Wang et al., 2020) by choosing suitable equivariant CNNs.

One practical limitation of this method is the necessity to discretise the continuous RKHS embedding, which can be costly and breaks the theoretical guarantees found in this work in theorem 2. An alternative approach would be to move away entirely from the structure of theorem 2, and
build an architecture more similar to the original CNP, util-
ising equivariant point cloud methods (Finzi et al., 2020b; Hutchin-
son et al., 2020; Satorras et al., 2021) to produce the embedding of the context set. This would avoid the discreti-
sation of the embedding grid and may provide speedups, but loses the flexibility promised by theorem 2.

9. Conclusion

In this work, we considered the problem of learning stochas-
tic fields and focused on using their geometric structure. We
motivated the design of equivariant stochastic process
models by showing the equivalence of equivariance in the
posterior map to invariance in the prior data distribution. We
fully characterised equivariant Gaussian processes and intro-
duced Steerable Conditional Neural Processes, a model that
combines recent developments in the design of equivariant
neural networks with the family of Neural Processes. We
showed that it improves results of previous models, even for
data which shows inhomogeneities in space such as weather,
and is more robust to perturbations in the underlying distri-
bution.

Our work shows that implementing general symmetries
in stochastic process or meta-learning models could be a
substantial step towards more data-efficient and adaptable
machine learning models. Inference models which respect
the structure of fields, in particular vector fields, could fur-
ther improve the application of machine learning in natural
sciences, engineering and beyond.

Acknowledgements

Peter Holderrieth is supported as a Rhodes Scholar by the
Rhodes Trust. Michael Hutchinson is supported by the EP-
SRC Centre for Doctoral Training in Modern Statistics and
Statistical Machine Learning (EP/S023151/1). Yee Whye
Teh’s research leading to these results has received funding
from the European Research Council under the European
Union’s Seventh Framework Programme (FP7/2007-2013)
ERC grant agreement no. 617071.

We would also like to thank the Python commu-
nity (Van Rossum & Drake Jr, 1995; Oliphant, 2007) for
developing the tools that enabled this work, including Py-
torch (Paszke et al., 2017b), NumPy (Oliphant, 2006; Walt
et al., 2011; Harris et al., 2020), SciPy (Jones et al., 2001),
and Matplotlib (Hunter, 2007).

References

Álvarez, M. A., Rosasco, L., and Lawrence, N. D. Kernels
for vector-valued functions: A review. Found. Trends
Mach. Learn., 4(3):195–266, March 2012. ISSN 1935-
8237. doi: 10.1561/2200000036.

Andrychowicz, M., Denil, M., Colmenarejo, S. G., Hoffman,
M. W., Pfau, D., Schaul, T., and de Freitas, N. Learning
to learn by gradient descent by gradient descent. CoRR,
abs/1606.04474, 2016.

9780132413770.

Brault, R., Heinonen, M., and Buc, F. Random Fourier Fea-
tures For Operator-Valued Kernels. In Asian Conference
on Machine Learning, pp. 110–125. PMLR, November
2016. ISSN: 1938-7228.

Bröcker, T. and Dieck, T. Representations of Compact Lie
Groups. Graduate Texts in Mathematics. Springer Berlin

Bruinsma, W., Requeima, J., Foong, A. Y. K., Gordon, J.,
and Turner, R. E. The Gaussian Neural Process. In Third
Symposium on Advances in Approximate Bayesian
Inference, 2021.

Cohen, T. and Welling, M. Group equivariant convolutional
networks. In Proceedings of The 33rd International Con-
ference on Machine Learning, volume 48 of Proceedings
of Machine Learning Research, pp. 2990–2999, New
York, New York, USA, 20–22 Jun 2016. PMLR.

Cohen, T. S. and Welling, M. Steerable CNNs. In 5th Inter-
national Conference on Learning Representations, ICLR
2017, Toulon, France, April 24-26, 2017, Conference

Diesleman, S., Fauw, J. D., and Kavukcuoglu, K. Exploiting
cyclic symmetry in convolutional neural networks. In Proceed-
ings of The 33rd International Conference on Machine
Learning, volume 48 of Proceedings of Machine Learning
Research, pp. 1889–1898, New York, New York,
USA, 20–22 Jun 2016. PMLR.

Esteves, C., Makadia, A., and Daniilidis, K. Spin-weighted
spherical CNNs. In Advances in Neural Information

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-
learning for fast adaptation of deep networks. In Proceed-
ings of the 34th International Conference on Machine
Learning, volume 70 of Proceedings of Machine Learning
Equivariant Learning of Stochastic Fields: Gaussian Processes and Steerable Conditional Neural Processes


