physical sciences forum

MDPI

Proceeding Paper

Efficient Representations of Spatially Variant Point Spread Functions with Butterfly Transforms in Bayesian Imaging Algorithms †

Vincent Eberle [1,2,*], Philipp Frank [1], Julia Stadler [1], Silvan Streit [3] and Torsten Enßlin [1,2]

[1] Max Planck Institute for Astrophysics, Karl-Schwarzschild-Straße 1, 85748 Garching, Germany
[2] Faculty of Physics, Ludwig-Maximilians-Universität München (LMU), Geschwister-Scholl-Platz 1, 80539 München, Germany
[3] Fraunhofer Institute for Applied and Integrated Security AISEC, Lichtenbergstraße 11, 85748 Garching, Germany
* Correspondence: veberle@mpa-garching.mpg.de
† Presented at the 41st International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering, Paris, France, 18–22 July 2022.

Abstract: Bayesian imaging algorithms are becoming increasingly important in, e.g., astronomy, medicine and biology. Given that many of these algorithms compute iterative solutions to high-dimensional inverse problems, the efficiency and accuracy of the instrument response representation are of high importance for the imaging process. For this reason, point spread functions, which make up a large fraction of the response functions of telescopes and microscopes, are usually assumed to be spatially invariant in a given field of view and can thus be represented by a convolution. For many instruments, this assumption does not hold and degrades the accuracy of the instrument representation. Here, we discuss the application of butterfly transforms, which are linear neural network structures whose sizes scale subquadratically with the number of data points. Butterfly transforms are efficient by design, since they are inspired by the structure of the Cooley–Tukey Fast Fourier transform. In this work, we combine them in several ways into butterfly networks, compare the different architectures with respect to their performance and identify a representation that is suitable for the efficient respresentation of a synthetic spatially variant point spread function up to a 1% error.

Keywords: response functions; spatially variant point spread functions; convolution; Bayesian imaging; butterfly matrices; Toeplitz matrices; sparse representations; neural networks

Citation: Eberle, V.; Frank, P.; Stadler, J.; Streit, S.; Ensslin, T. Efficient Representations of Spatially Variant Point Spread Functions with Butterfly Transforms in Bayesian Imaging Algorithms. *Phys. Sci. Forum* **2022**, *5*, 33. https://doi.org/10.3390/psf2022005033

Academic Editors: Frédéric Barbaresco, Ali Mohammad-Djafari, Frank Nielsen and Martino Trassinelli

Published: 14 December 2022

1. Introduction

Images of astronomical objects are the result of measurements by physical instruments and intricate postprocessing. In this procedure, instrument responses play an important role as they build the connection between the signal, i.e., the quantity of interest and the observables. Unfortunately, instrument responses are often non-trivial and hard to model in a simple and numerically efficient form. Examples for such instruments are the X-ray Observatories eROSITA [1] and Chandra [2]. Both are challenging to compute due to their inhomogeneous behavior in terms of space and energy. In order to efficiently make statistical field inferences, for example, by using NIFTy [3–5], a Python software package for Numerical Information Field Theory [6–9], these responses need to be fast and differentiable. One promising candidate for the efficient representation of instrument responses are butterfly transforms, a linear neural network structure inspired by the structure of the Fast Fourier Transform (FFT) algorithm, whose size scales with $\mathcal{O}(N \log N)$.

In many cases, the measurement equation for some data d, taken with an instrument response R of the signal s assuming additive noise n, can be formulated as $d = R(s) + n$.

Regarding photographic instruments this response R is a linear map that can be separated into two operations D and O. Here, D is describing the measurement process of the detector, while O represents the optical properties of the instrument. The latter is also referred to as point spread function (PSF). Since computers are used for the analysis of the experiments performed, the continuous signal space is approximated by a discrete pixelation and thus all operators can be represented as matrices.

If O can be approximated by a circulant matrix, a matrix consisting of cyclic permutations of the same row vector a, its matrix multiplication with any vector simplifies to a discrete convolution with a, meaning that it is spatially invariant and homogeneous, respectively. In many physically relevant cases, this homogeneity can approximately be assumed for a given observed area of the instrument. Additionally, the convolution theorem states that a convolution corresponds to a point-wise multiplication in Fourier space. Consequently, convolutional responses can be represented in an efficient way, due to the fact that one only has to store one N-entry vector instead of a N^2 matrix, as well as due to the efficiency one gains by replacing a discrete Fourier transformation by the Fast Fourier Transformation (FFT).

Often this assumption only holds up to a certain degree and in a limited field of view. For spatially variant PSFs and thus non-circulant responses efficient representations are urgently needed. In their paper about learning fast linear transforms algorithms [10], Dao et al. proposed a way to learn fast linear transformation algorithms using so-called butterfly factorizations, which are closely related to the butterfly transforms introduced in this paper. They were able to learn several fast linear transformations, e.g., FFT, discrete sine transform, etc., and showed that their approach can be applied as an efficient replacement of generic matrices in machine learning pipelines. We propose using butterfly transforms to represent spatially variant PSFs in order to build likelihoods for instruments such as eROSITA, Chandra, and many more.

In this paper, we present a way to parameterize butterfly transforms, combine them into networks and compare different network architectures in terms of their efficiency and accuracy. Section 2 describes how butterfly transformations are parameterized and how they are inspired by the structure of the Cooley–Tukey–FFT algorithm. Section 3 gives a short introduction to information field theory and Section 4 describes different designs of likelihoods. In Section 5, we define a metric in order to compare different butterfly network architectures with respect to their capability to represent the synthetic response defined in Section 6. The final results can be found in Section 7.

2. Methods

2.1. Fast Fourier Transformation

Due to the convolution theorem, Fourier transformation is one of the key elements of convolutional processes and thus the algorithm of FFT is highly relevant for the representation of instrument responses on regular grids. The main idea of the FFT is to split the sum in the discrete Fourier transform (DFT) into two sums, over even and odd indices [11]. By using the mathematical properties of the N-th primitive root $\omega_N = e^{\frac{-2\pi i}{N}}$, it can be shown that

$$\hat{f}_k = \frac{1}{\sqrt{2}}\hat{f}_k^{\text{even}} + \frac{1}{\sqrt{2}}\omega_N^k \hat{f}_k^{\text{odd}} \quad \text{and} \quad \hat{f}_{k+\frac{N}{2}} = \frac{1}{\sqrt{2}}\hat{f}_k^{\text{even}} - \frac{1}{\sqrt{2}}\omega_N^k \hat{f}_k^{\text{odd}} \,. \tag{1}$$

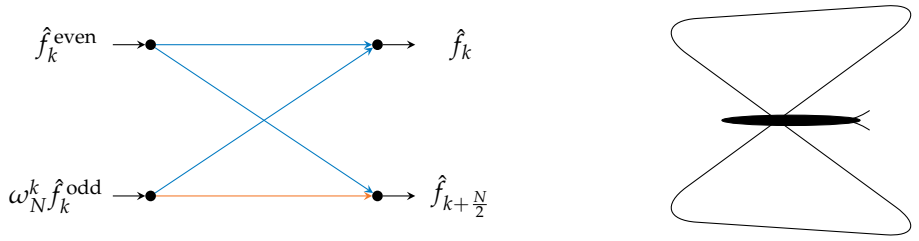
This means that an N-sized Fourier transform can be separated into two $N/2$-sized Fourier transforms along the even and odd indices [12]. The components $\hat{f}_k^{\text{even}}$ and $\hat{f}_k^{\text{odd}}$ can then be used to calculate f_k and $f_{k+\frac{N}{2}}$. Putting together the relations in Equation (1) yields.

$$\begin{pmatrix} \hat{f}_k \\ \hat{f}_{k+N/2} \end{pmatrix} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} \hat{f}_k^{\text{even}} \\ \omega_N^k \hat{f}_k^{\text{odd}} \end{pmatrix} . \tag{2}$$

The two smaller Fourier transforms can be separated in the same way, resulting in a divide and conquer algorithm. Assuming that the initial value of N is a power of 2, this splitting can be applied $\log_2(N)$ times. Inspired by machine learning language, these iterations are called layers in the following. With N additions in each of these layers, the total computational complexity is about $\mathcal{O}(N \log_2 N)$. Comparing this to a regular DFT with its computational complexity of $\mathcal{O}(N^2)$ (N components with N summands) the amount of saved time in the FFT algorithm is significant.

2.2. Butterfly Transform and Convolution

The data-flow diagram visualizing the algorithm of Equation (2) is often called a butterfly diagram, due to its appearance (see Figure 1). Therefore, the abstraction of the FFT algorithm, resulting in a similar data-flow diagram, is called butterfly transform in the following. As the butterfly diagrams always connect to two components, most of the descriptions used in the following concerning their parameterization are 2-dimensional, to keep the notation simple.



(**a**) Butterfly diagram (**b**) Butterfly pictogram

Figure 1. Comparison of: (**a**) butterfly diagram—blue lines indicate an addition, the orange line indicates subtraction; (**b**) pictogram of a butterfly with similar appearance.

In order to generalize the FFT while preserving its efficient structure, we decompose the operations in Equation (2) into a diagonal operator Φ and a mixing operator Θ as given in the following.

$$\Phi = \begin{pmatrix} 1 & 0 \\ 0 & \omega_N \end{pmatrix}, \quad \Theta = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad \text{and thus} \begin{pmatrix} \hat{f}_k \\ \hat{f}_{k+N/2} \end{pmatrix} = \Theta\, \Phi \begin{pmatrix} \hat{f}_k^{\text{even}} \\ \hat{f}_k^{\text{odd}} \end{pmatrix}. \tag{3}$$

For each component, we introduce free parameters that control how the operation deviates from an ordinary FFT. A general representation of Θ is obtained by parameterizing it by the sine and cosine of an angle θ,

$$\Theta_\theta = \begin{pmatrix} \cos\theta & \sin\theta \\ \sin\theta & -\cos\theta \end{pmatrix}. \tag{4}$$

To preserve the generality of the transformation within one layer, the θs for different connected pairs, denoted by the index k in Equation (3), are independent. That means that for an N-size transformation there are $N/2$ θs, in each layer, regulating the interaction between two connected data points. Considering this parameterization, we get the Θ from Equation (3), i.e., the one for an FFT, by inserting $\theta = \frac{\pi}{4}$. The operator Φ and an additional operator Γ are parameterized as

$$\Phi_\phi = \begin{pmatrix} e^{i\phi_1} & 0 \\ 0 & e^{i\phi_2} \end{pmatrix}, \quad \phi_j \in \mathbb{R} \quad \text{and} \quad \Gamma_\gamma = \begin{pmatrix} e^{\gamma_1} & 0 \\ 0 & e^{\gamma_2} \end{pmatrix}, \quad \gamma_j \in \mathbb{R}\,. \tag{5}$$

This parameterization of Φ makes it possible to recover the correct phases for an FFT, but also to change them in an arbitrary way. The combination of Θ and Φ is sufficient to represent an entire FFT transform. To obtain an even more general transformation, the

diagonal operator Γ is introduced, which accounts for the real-valued amplitudes. This leads to a loss of unitarity for $\gamma_1,\ \gamma_2 \neq 0$ in the combined transformation of Γ, Φ and Θ.

Now, we can build a generic butterfly-structured transformation B, using the layered structure of an FFT as a guiding example. The subscript of the operators refers to the layer in the FFT algorithm and thus implies that the correct components are connected.

$$B = \Gamma_0\Phi_0\Theta_0 \dots \Gamma_j\Phi_j\Theta_j \dots \Gamma_{n-1}\Phi_{n-1}\Theta_{n-1}\,. \tag{6}$$

Given this butterfly transformation and the structure of a convolution operation, based on the convolutional theorem, a butterfly convolution-like operator O can be formulated as

$$O = B^\dagger \Lambda B\,. \tag{7}$$

In this equation the Λ operator corresponds to the Fourier transformed PSF. Usually physically reasonable PSFs are real valued in position space and thus complex-valued in harmonic space. Therefore, the Λ operator is defined as a diagonal operator, with complex values,

$$\Lambda_\lambda = \begin{pmatrix} e^{\lambda_1} & 0 \\ 0 & e^{\lambda_2} \end{pmatrix}, \quad \lambda_j \in \mathbb{C}\,. \tag{8}$$

$B^\dagger$, in Equation (7), denotes the adjoint of B. For some experiments, the parameters of B and $B^\dagger$ were strictly coupled, called mirrored architecture in the following. For others, the parameters were independent, denoted by different indices, resulting in a non-mirrored architecture:

$$O = B_1^\dagger \Lambda B_2\,. \tag{9}$$

Since the butterfly structure is strongly related to the FFT, it would make sense to treat multidimensional butterfly transformations in the same way as multidimensional Fourier transformations. Therefore, butterfly transformations can be applied to each dimension separately. However, in this work the 2D application is slightly modified, in a way that the mixing operator Θ is applied to each axis separately (For the first axis all columns are transformed with the same θs, whereas for the second axis all rows transformations share the same θs.), but after this axis-wise Θ transformation the operators Φ and Γ are applied as diagonal operators. Another approach is to reduce the number of dimensions to one (in this case, as we are dealing with images, from 2D to 1D) and just perform one butterfly transform to this one dimension. For the case of 2-dimensional inputs the dimensionality reduction can be easily done by concatenating all the column vectors to one long vector, which will be called flattening from now on. These two different approaches differ in the number of layers needed by the butterfly algorithm as well as in the number of parameters per layer.

3. Information Field Theory

To reach a better understanding for the area of use for the efficient responses, a brief introduction to information field theory (IFT) [9] will be given. Information field theory is the application of information theory to physical fields. Probably the most important relation within information theory is Bayes' Theorem,

$$\mathcal{P}(s|d) = \frac{\mathcal{P}(d,s)}{\mathcal{P}(d)} = \frac{\mathcal{P}(d|s)\mathcal{P}(s)}{\mathcal{P}(d)}\,, \tag{10}$$

which connects a posterior with the likelihood, the prior, and the evidence. The likelihood can be computed from the noise statistic $\mathcal{P}(n|s)$ and the measurement equation, here in the form $\mathcal{P}(d|s,n) = \delta(d - R(s) - n)$. Thus, the likelihood is

$$\begin{aligned}\mathcal{P}(d|s) &= \int dn\, \mathcal{P}(d|s,n)\mathcal{P}(n|s) \\ &= \int dn\, \delta(d - R(s) - n)\mathcal{P}(n|s) = \mathcal{P}(n = d - R(s)|s)\ .\end{aligned} \tag{11}$$

The prior $\mathcal{P}(s)$ is chosen with respect to the physical knowledge one has about the observed quantity or situation. The evidence $\mathcal{P}(d) = \int ds\, \mathcal{P}(d|s)\mathcal{P}(s)$ is needed for the proper normalization of the posterior $\mathcal{P}(s|d)$. The information Hamiltonian is defined as the negative logarithm of the probability, $\mathcal{H}(d,s) = -\ln[\mathcal{P}(d,s)]$. Due to the properties of the logarithm and the product rule of probabilities the information Hamiltonian, $\mathcal{H}$ is an additive quantity $\mathcal{H}(d,s) = \mathcal{H}(d|s) + \mathcal{H}(s)$. Assuming Gaussian priors for signal, $\mathcal{G}(s,S)$, and noise, $\mathcal{G}(n,N)$, and using Equation (11) the Hamiltonians simplify to

$$\begin{aligned}\mathcal{H}(s) &= -\ln[\mathcal{G}(s,S)] = -\ln\left[\frac{1}{\sqrt{2\pi S}}\exp\left(-\frac{1}{2}s^\dagger S^{-1}s\right)\right] = \frac{1}{2}\ln|2\pi S| + \frac{1}{2}s^\dagger S^{-1}s\ , \\ \mathcal{H}(d|s) &= -\ln[\mathcal{G}(n,N)] = \frac{1}{2}\ln|2\pi N| + \frac{1}{2}(d - R(s))^\dagger N^{-1}(d - R(s))\ .\end{aligned} \tag{12}$$

One way to find an estimate for the signal s is to maximize the probability $\mathcal{P}(s|d)$. This can be achieved by minimizing the joint Hamiltonian $\mathcal{H}(d,s)$, with respect to the signal s. This is the maximum a posteriori (MAP) approximation. There are also ways to find an estimate for a signal with uncertainty quantification like metric Gaussian variational inference (MGVI) [13] or geometric variation inference (geoVI) [14]. As a minimization algorithm, Newton-CG [15] was used throughout all experiments. If the measurement process follows Poisson statistics, which is the case for realistic photographic measurements, a Poissonian likelihood model has to be used instead of a Gaussian likelihood model.

4. Parallel and Serial Likelihoods

Models for inference processes in NIFTy are built in a forward way, as so-called generative models. This means that a model of the physical signal is created first, followed by the instrument response, and finally by synthetic data. Applying the IFT formalism, described in Section 3, to a generative model with a butterfly convolution operator as a response yields a likelihood with dependencies on the signal s and the response parameters θ, ϕ, γ, and λ.

In addition to being able to use butterfly convolution operators with mirrored, non-mirrored, flat, and 2D configurations, they can be combined into a network built in parallel or in series. For the case of n multiple butterfly convolution operators in series, the response operator in Equation (12) is

$$R(s,\theta,\phi,\gamma,\lambda) = O_1 \dots O_n s\ , \tag{13}$$

while in the case of n butterfly convolution operators applied in a parallel architecture

$$R(s,\theta,\phi,\gamma,\lambda) = (O_1 + \dots + O_n)s\ . \tag{14}$$

Before using a butterfly network in an imaging application, the response R needs to be trained on signal-data pairs of the instrument. Using these signal-data pairs, the joint Hamiltonian $\mathcal{H}(d,s,\theta,\phi,\gamma,\lambda)$ is minimized with respect to the response parameters θ, ϕ, γ, and λ, resulting in a MAP approximation of the instrument. The initial values for these parameters, $\widetilde{\theta}, \widetilde{\phi}, \widetilde{\gamma}$, and $\widetilde{\lambda}$, are chosen such that all O_i correspond to a convolution with a delta peak ($\widetilde{\theta} = \pi/4$, $\widetilde{\gamma} = 0$, $\widetilde{\lambda} = 1$, and $\widetilde{\phi}$ according to the needed phases, see Section 2.1). The prior distribution of the parameters is assumed to be Gaussian, with means at the initial values and unit variance. The final goal of the minimization is to obtain an efficient digital twin of the real physical instrument.

Once a butterfly response is trained, it can be used for imaging with the corresponding instrument. For this, the response parameters are fixed to the inferred values $\underline{\theta}, \underline{\phi}, \underline{\gamma}$, and $\underline{\lambda}$, resulting in a response operator, which is linear in the signal s. The selection of a suitable generative model for s depends on the observation of interest. In order to obtain an estimate for the physical signal s, the inference algorithms MGVI or geoVI can be used.

5. Evaluation of the Response Approximation

Before using a trained butterfly response in an inference algorithm, it must be certified that the mapping done by the response representation is sufficiently accurate. Therefore, we compare the action caused by a signal, here a point source at position z, $s(x) = \delta(x - z)$, of the to-be-learned or simulated response with the butterfly response by their absolute difference. This will be called response approximation error

$$E(s) = \mathrm{abs}\Big[R_{\mathrm{sim.}}(s) - R_{\mathrm{but.}}(s)\Big] \,. \tag{15}$$

To keep the evaluation simple, unit brightness point sources at all signal domain locations $z \in \Omega$ are considered. In order to quantify the total error with respect to all mapping errors, we calculate the 2-norm ($\|s\|_2 = \sqrt{\sum_{x\in\Omega} |s_x|^2}$) of the 4D matrix $\mathbf{E}_z = E[\delta(x - z)]$, containing the error images for all posible z-values and normalize it by dividing with the 2-norm of the matrix $\mathbf{R}_z = R_{\mathrm{sim.}}[\delta(x - z)]$, containing all true simulated responses resulting in the the total error $\widehat{\zeta}$:

$$\widehat{\zeta} = \frac{\|\mathbf{E}\|_2}{\|\mathbf{R}_{\mathrm{sim.}}\|_2} \,. \tag{16}$$

6. Synthetic Response

In order to investigate whether and to which degree butterfly networks are capable of approximating spatially variant PSFs, they were trained to approximate a synthetic response. This synthetic response can be regarded as the convolution of the signal s, which is a point source located at the position z, $s(x) = \delta(x - z)$, with a rotational symmetric PSF with a position dependent shape,

$$(Rs)^y = \int_\Omega \mathrm{PSF}(y - x, x) s(x) \, dx \,. \tag{17}$$

For the PSF a zero centered Gaussian was chosen,

$$\mathrm{PSF}(x, z) = \mathcal{G}(\rho, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\big(-\frac{\rho^2}{2\sigma^2(z)}\big) \,, \tag{18}$$

with $\rho = \|x\|_2$, where x is the coordinate vector of the image plane and $\|x\|_2 = \sqrt{x_1^2 + x_2^2}$ is its length. The dependence on the position z of the point source is encoded in the variance $\sigma^2(z)$ of the Gaussian. To keep this spatial dependency simple, only the distance from the center of the image c to the point source z, $r = \|c - z\|_2$, influences the shape of the PSF. As this absolute value depends on the image resolution, r will be normalized by the maximal distance within the image, $\hat{r} = r/r_{\mathrm{max}}$, to get a relative measure for the distance being in the interval $[0, 1]$. As indicated, the variance σ^2 is a function of this relative distance $\hat{r}$ between the point source at z and the image center c,

$$\sigma^2(\hat{r}) = a \cdot \hat{r}^2 + \epsilon \,. \tag{19}$$

The two parameters are set to $a = 0.01$ and $\epsilon = 10^{-5}$. Following the Equation (19), larger distances $\hat{r}$ lead to larger values of the variance σ^2. This means that point sources with smaller values of $\hat{r}$ are convolved with a sharper Gaussian, while point sources being at far distance from the center are convolved with broader Gaussians (see Figure 2). This results in an spatially variant PSF, which can be used to examine the expressiveness of the butterfly architecture.

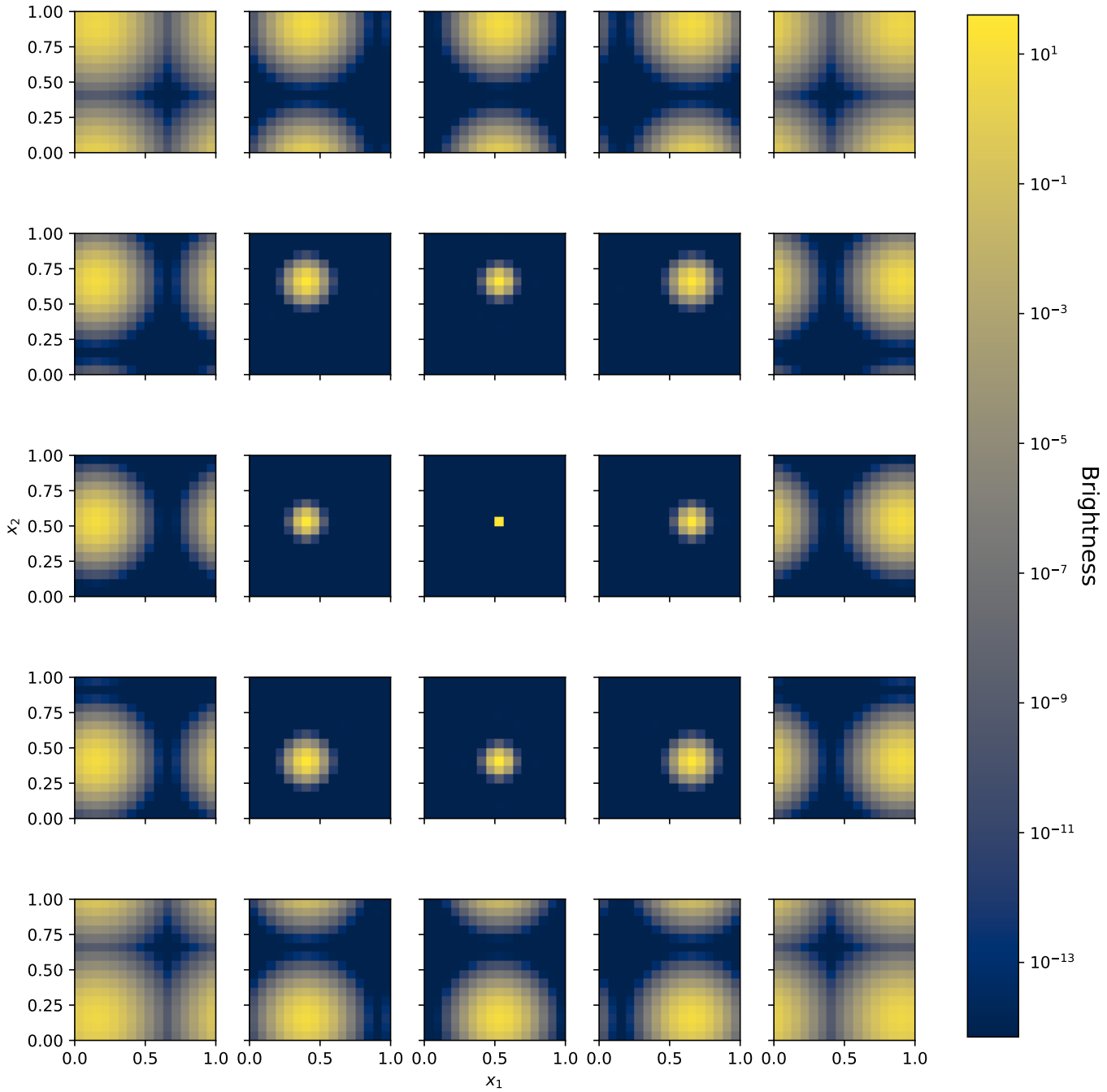


Figure 2. 25 signal responses $R(s)$ for point sources at different positions. For simplicity we used periodic boundaries for the kernels, which will be properly adressed in the future. The colors show the resulting brightness values.

7. Results

In search of a butterfly network capable of representing spatially variant point spread functions, various architectures were compared, in terms of their ability to represent the synthetic response, differing in their number of butterfly convolution operators (BCOs), mirrored (mr) or non-mirrored (nmr) architecture, flat or 2D network design, serial or parallel built likelihood (see Table 1). All of these networks were trained to approximate the synthetic response described in Section 6 until the optimization was sufficiently converged (300 Newton steps). As training data, a set of all possible PSFs within the given pixelation of 16×16 was used. The signals were fixed to be point sources with brightness values 40 at the corresponding positions and the noise covariance N was set to be diagonal with

entries of 10^{-6}. In order to get a better understanding of the influence of some of these properties on the total approximation behaviour, the networks are regarded separately and with respect to their final total approximation error $\hat{\zeta}$ in Table 1.

Table 1. Parameters and results for all seven network architectures. The density is here defined as the ratio of the number of parameters and the number of entries in a full matrix representation (16^4 = 65,536). A lower density indicates a higher efficiency of the representation.

Network Name	**Net$_1$**	**Net$_2$**	**Net$_3$**	**Net$_4$**	**Net$_5$**	**Net$_6$**	**Net$_7$**
# BCOs	1	2	3	3	3	3	3
architecture	mr	mr	mr	nmr	mr	nmr	nmr
design	flat	flat	flat	flat	2D	2D	flat
likelihood	serial	serial	serial	serial	serial	serial	parallel
$\hat{\zeta}$ in %	7.96	3.14	2.00	1.04	2.45	1.50	6.86
# parameters	5632	11,264	16,896	32,256	7872	14,208	32,256
Density in %	8.59	17.19	25.78	49.22	12.01	21.68	49.22

The comparison of the $\hat{\zeta}$ value of Net$_1$, Net$_2$, and Net$_3$ with 1, 2, and 3 BCOs, but otherwise the same properties, shows that a higher number of BCOs lowers the total error and thus increases the approximation capability. The second property of interest is the kind of architecture used, mirrored or non-mirrored. Therefore the $\hat{\zeta}$ value of Net$_3$, with its mirrored architecture, is compared to the one of Net$_4$, with its non-mirrored architecture, while their other properties are equivalent. This shows that the non-mirrored architecture is performing better than the mirrored one. The same conclusion can be drawn by comparing $\hat{\zeta}$ of Net$_5$ and Net$_6$, which also only differ in their state of mirroring. In a similar way the flattened and the 2D application can be examined. Since Net$_3$ and Net$_5$ only differ in this property, their error values suggest that the flat application is superior to the 2D application with respect to reconstruction capability. This is confirmed by regarding the error of Net$_4$ and Net$_6$, which are in a similar relationship.

Since more BCOs, flattening, and a non-mirrored architecture increase the number of parameters and thus lead to more degrees of freedom, it is assumed that these architectures are more flexible and can approximate the true response in a better way.

For the overall efficiency of the various networks it is not only important to approximate the synthetic response in a optimal way, but also to keep the number of parameters, and thus the network density (Network density is here defined as the ratio of network parameters and number of entries in a full matrix representation.), as low as possible (see Figure 3). In the examined cases, sparser architectures tend to perform worse in comparison to architectures with more parameters. Overall Net$_4$ approximates the synthetic reponse best with an 1% error. Net$_6$, however, has only 44% of the parameters of Net$_4$ and is therefore less dense. This goes hand in hand with a slightly increased approximation error by an absolute value of 0.46% (see Table 1). In the end, the number of parameters of butterfly networks still scales with $\mathcal{O}(N \log N)$. This means that they become less dense with increasing resolution.

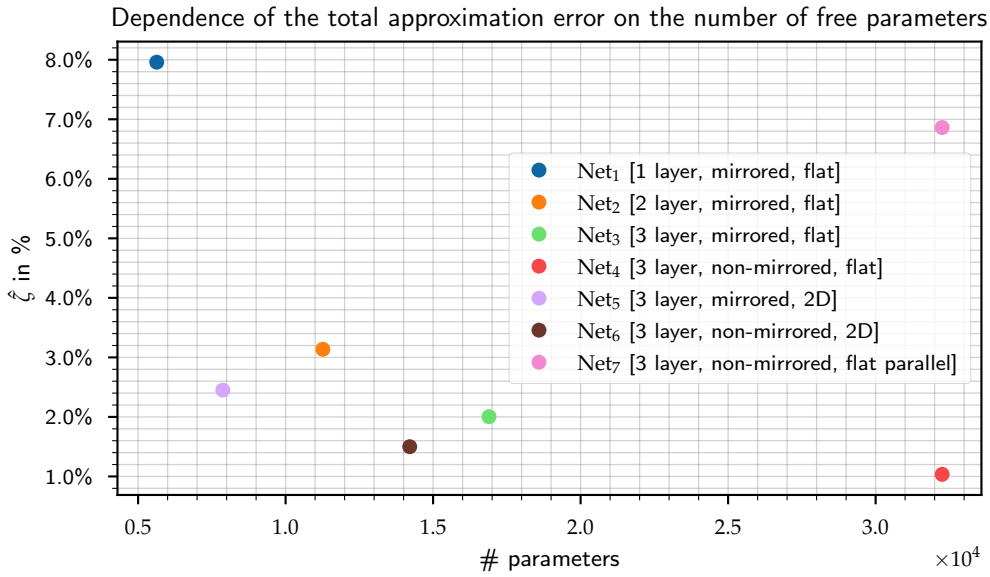


Figure 3. Total approximation error $\hat{\zeta}$ with respect to the number of parameters in the network. A combination of low error and low number of parameters is important for a good efficiency of the corresponding network.

8. Discussion

The need for efficient response representations in imaging led to the development of the models presented in this work, which were inspired by earlier research on butterfly matrices [10]. The efficient structure of butterfly matrices, inherited of Fast Fourier Transforms (FFT), results in a subquadratic algorithm scaling with $\mathcal{O}(N \log N)$ that is capable of representing an expensively simulated synthetic response up to 1% error. To this end, Net$_4$, a butterfly convolutional network with three butterfly convolution operators (BCOs) in series, non-mirrored architecture, and flat application is used, which is differentiable and thus suitable for the application as a response in generative models for measurement data. In order to improve the computational performance regarding support of GPUs and parallelization, more advanced machine learning platforms such as TensorFlow [16] or PyTorch [17] could be considered. After sufficient training, the corresponding butterfly network can be used to perform high-fidelity imaging using information field theory. Additonally, other fields of application with a connection to slightly inhomogeneous processes are imaginable. All in all, the method to represent instrument response functions introduced in this work is promising to improve imaging with complex photographic instruments and thus should be considered in further research.

Author Contributions: Conceptualization, V.E., P.F., J.S., S.S. and T.E.; methodology, V.E., S.S. and P.F.; software, V.E. and S.S.; validation, V.E. and S.S.; formal analysis, V.E. and T.E.; investigation, V.E.; resources, V.E.; data curation, V.E.; writing—original draft preparation, V.E.; writing—review and editing, V.E.; visualization, V.E.; supervision, T.E. and P.F.; project administration, T.E.; funding acquisition, T.E. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the German Aerospace Center and the Federal Ministry of Education and Research through the project "Universal Bayesian Imaging Kit—Information Field Theory for Space Instrumentation" (Förderkennzeichen 50OO2103).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We thank Philipp Arras for detailed feedback on the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Predehl, P.; Andritschke, R.; Arefiev, V.; Babyshkin, V.; Batanov, O.; Becker, W.; Böhringer, H.; Bogomolov, A.; Boller, T.; Borm, K.; et al. The eROSITA X-ray telescope on SRG. *arXiv* **2020**, arXiv:2010.03477.
2. Weisskopf, M.C.; Tananbaum, H.D.; Van Speybroeck, L.P.; O'Dell, S.L. Chandra X-ray Observatory (CXO): Overview. In *Proceedings of the X-ray Optics, Instruments, and Missions III*; SPIE: Bellingham, DC, USA, 2000; Volume 4012, pp. 2–16.
3. Selig, M.; Bell, M.R.; Junklewitz, H.; Oppermann, N.; Reinecke, M.; Greiner, M.; Pachajoa, C.; Enßlin, T.A. NIFTY—Numerical Information Field Theory—A versatile PYTHON library for signal inference. *Astron. Astrophys.* **2013**, *554*, A26. [CrossRef]
4. Steininger, T.; Dixit, J.; Frank, P.; Greiner, M.; Hutschenreuter, S.; Knollmüller, J.; Leike, R.; Porqueres, N.; Pumpe, D.; Reinecke, M.; et al. NIFTy 3—Numerical Information Field Theory: A Python Framework for Multicomponent Signal Inference on HPC Clusters. *Ann. Phys.* **2019**, *531*, 1800290. [CrossRef]
5. Arras, P.; Baltac, M.; Ensslin, T.A.; Frank, P.; Hutschenreuter, S.; Knollmueller, J.; Leike, R.; Newrzella, M.N.; Platz, L.; Reinecke, M.; et al. *Nifty5: Numerical Information Field Theory v5*; record ascl:1903.008; Astrophysics Source Code Library: College Park, MD, USA, 2019.
6. Enßlin, T.A.; Frommert, M.; Kitaura, F.S. Information field theory for cosmological perturbation reconstruction and nonlinear signal analysis. *Phys. Rev. D* **2009**, *80*, 105005. [CrossRef]
7. Enßlin, T. Astrophysical data analysis with information field theory. In Proceedings of the AIP Conference Proceedings, Canberra, ACT, Australia, 15–20 December 2013; American Institute of Physics: College Park, MD, USA, 2014; Volume 1636, pp. 49–54.
8. Enßlin, T. Information field theory. In Proceedings of the AIP Conference Proceedings, Garching, Germany, 15–20 July 2012; American Institute of Physics: College Park, MD, USA, 2013; Volume 1553, pp. 184–191.
9. Enßlin, T.A. Information theory for fields. *Ann. Phys.* **2019**, *531*, 1800127. [CrossRef]
10. Dao, T.; Gu, A.; Eichhorn, M.; Rudra, A.; Ré, C. Learning fast algorithms for linear transforms using butterfly factorizations. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; Volume 97, pp. 1517–1527.
11. Cooley, J.W.; Tukey, J.W. An algorithm for the machine calculation of complex Fourier series. *Math. Comput.* **1965**, *19*, 297–301. [CrossRef]
12. Wolberg, G. *Fast Fourier Transforms: A Review*; Department of Computer Science, Columbia University: New York, NY, USA, 1988.
13. Knollmüller, J.; Enßlin, T.A. Metric Gaussian Variational Inference. *arXiv* **2019**, arXiv:1901.11033.
14. Frank, P.; Leike, R.; Enßlin, T.A. Geometric variational inference. *Entropy* **2021**, *23*, 853. [CrossRef] [PubMed]
15. Nocedal, J.; Wright, S. *Numerical Optimization*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2006; pp. 168–170.
16. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015. Available online: tensorflow.org (accessed on 8 December 2022).
17. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*; Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2019; pp. 8024–8035.