Continuiti: A Generalized Framework for Neural Operators

appliedAI institute for europe

github.com/aai-institute/continuiti

WHY NEURAL OPERATORS

Direct Mapping Between Function Spaces
- Enhanced Flexibility: Neural operators map inputs to outputs as functions, offering a flexible framework ideal for problems expressed naturally as functions.
- Reduced Complexity: Avoids the need to discretize function spaces, simplifying model formulation and reducing computational complexity.
- Increased Accuracy: Directly handling functions improves generalization and accuracy.

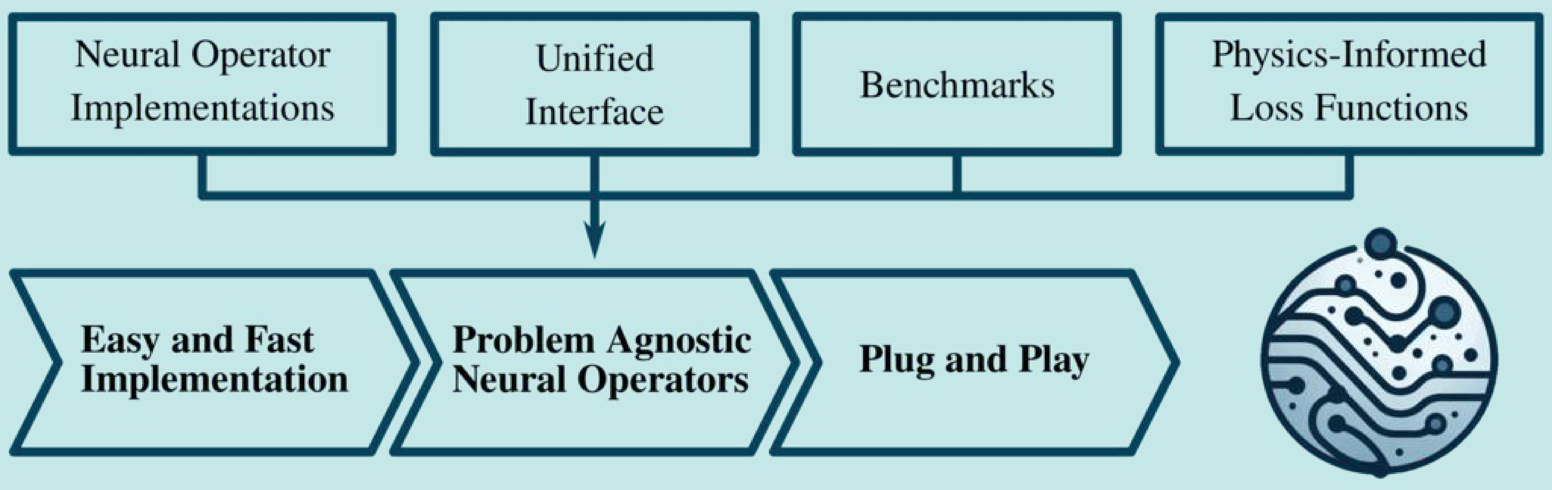
Discretization Independent
- The discretization of input and output functions can differ between samples.
- Neural operators can evaluate outputs at arbitrarily many points, in any location.

Physics Informed
- Seamless integration of physical constraints.
- Partial differential equations are naturally expressed using functions.

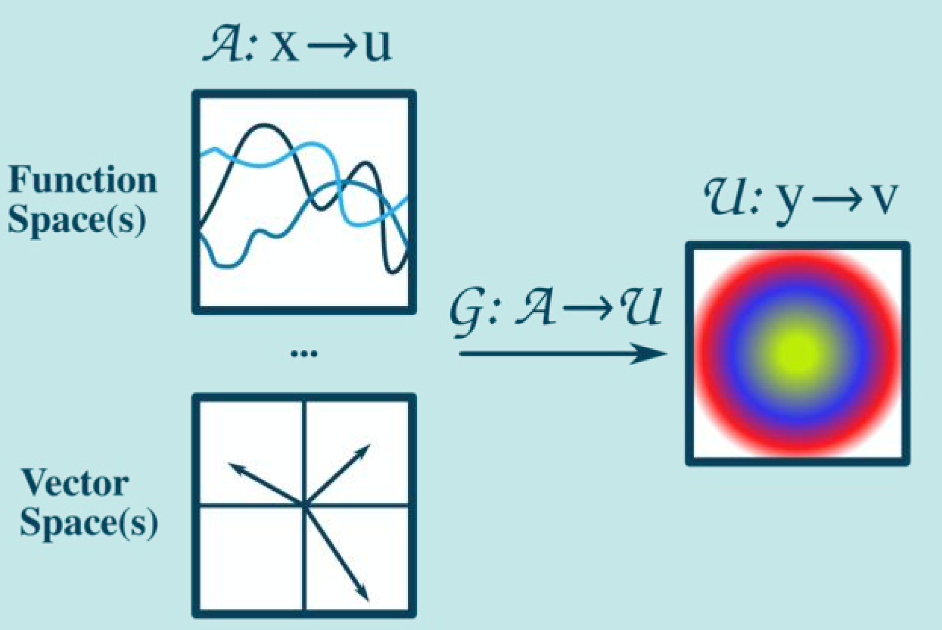
Wide Range of Applications
- Offers high flexibility and faster (even real-time) solutions.
- Provides robust performance across varied problems and datasets.
- Scalable: Effectively handles high-dimensional data and complex functions.
- Applicable in fields such as fluid dynamics, acoustics, structural mechanics, heat transfer, tomography, plasma physics, material design, seismology, optical systems, and many more.

CONTINUITI

Continuiti is an **open-source** Python package for learning function operators, emphasizing elegance and generality. It is based on the **PyTorch** framework, boosting familiarity and computational efficiency. The package includes **physics-informed loss functions** and **benchmarks** for operator learning tasks.



PROBLEM DESCRIPTION



- The problem is described by the mapping:

$$\mathcal{G}: \mathcal{A} \rightarrow \mathcal{U}$$

- The neural operator approximates the mapping:

$$\mathcal{G} \approx \mathcal{G}_\theta$$

- Neural operators are designed to learn mappings between infinite-dimensional function spaces rather than finite-dimensional vector-spaces.
- This approach allows to efficiently approximate solutions to partial differential equations and other functional problems.
- The resulting output function can be evaluated at arbitrarily many points, providing a flexible and powerful solution to the problem.

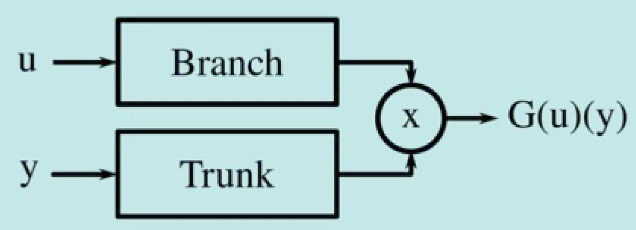
In Continuiti:
- The operator mapping is defined with:

```
>>> dno = DeepNeuralOperator()
>>> v = dno(x, u, y)
```

- Neural operator implementations can be swapped out seamlessly.
- Straightforward workflow through generalized datasets.

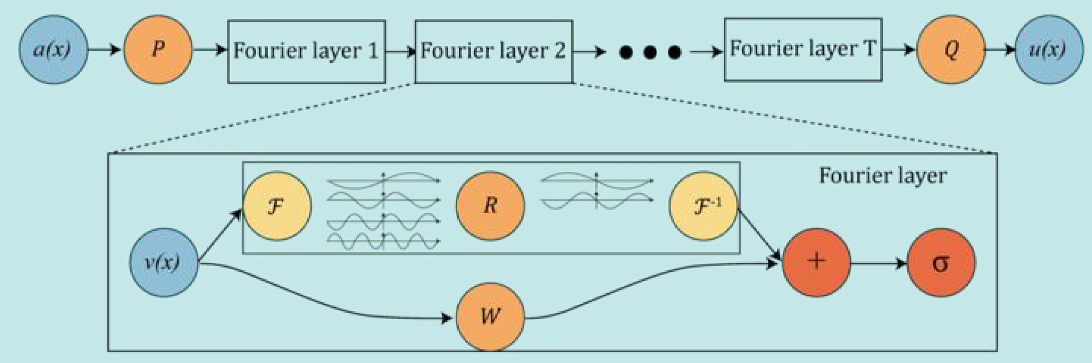
ARCHITECTURES

DeepONet [1]:
- First neural operator implementation.
- Based on the universal approximation theorem for operators by Chen et al. [2].



Other Implementations:
- Neural Operator [4]
- BelNet (Basis-enhanced neural operator) [5]
- Convolutional Neural Operator [6]
- Deep Neural Operator
- Deep Cat Operator
- Attention based models (GNOT [7], Zijie et al. [8])
- ... more to come :)

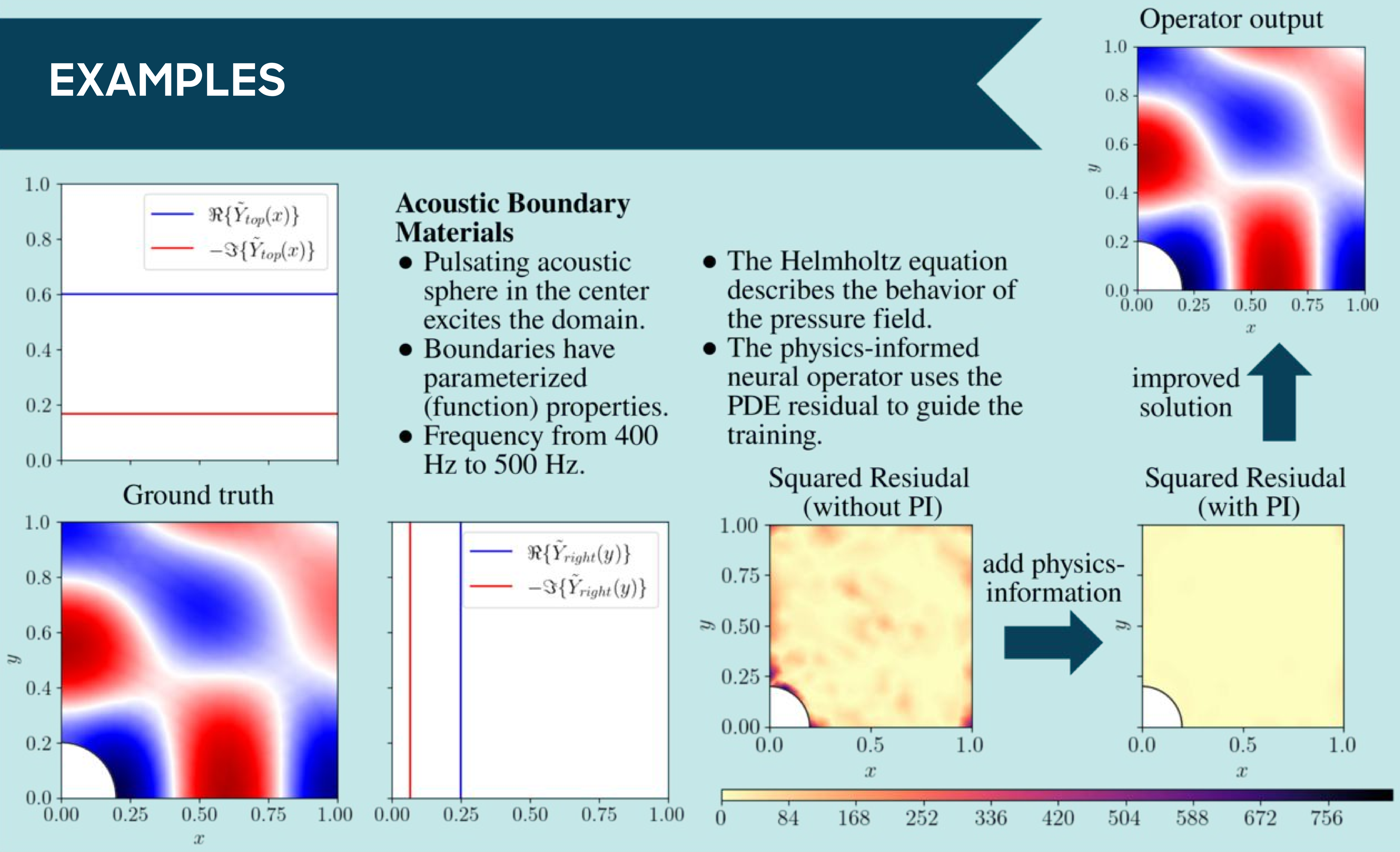
Fourier Neural Operator [3]:
- Utilizes Fourier transforms to learn the solution operator in the frequency domain.
- The fast Fourier transform allows for efficient implementations.



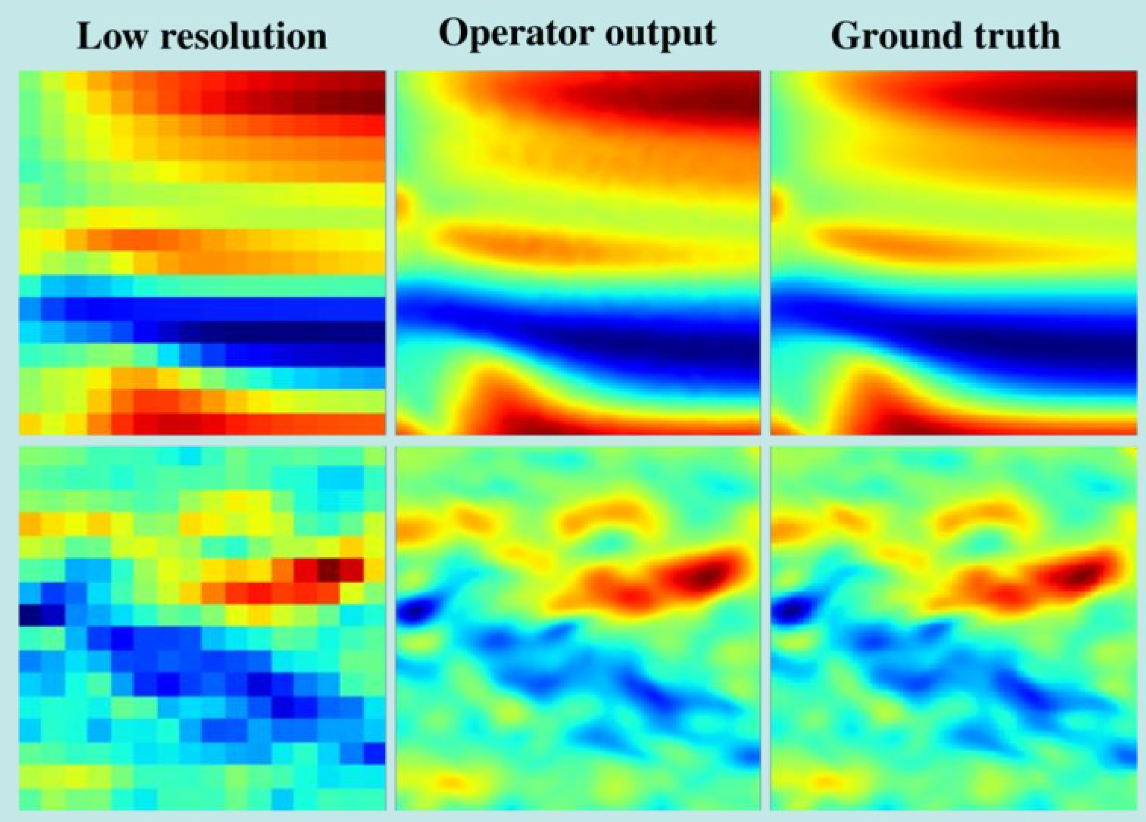
EXAMPLES

Acoustic Boundary Materials
- Pulsating acoustic sphere in the center excites the domain.
- Boundaries have parameterized (function) properties.
- Frequency from 400 Hz to 500 Hz.

- The Helmholtz equation describes the behavior of the pressure field.
- The physics-informed neural operator uses the PDE residual to guide the training.



Superresolution
- FLAME dataset: A set of flow samples of resolution 32x32 that should be upsampled to 128x128
- The dataset is available on Kaggle.
- The operator maps the low-resolution data to a continuous function.
- The trained operator can accurately predict the mapped function on a fine mesh, it achieved super-resolution.



[1] Lu, Lu, et al. "Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators." Nature machine intelligence 3.3 (2021): 218-229.
[2] Chen, Tianping, and Hong Chen. "Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems." IEEE transactions on neural networks 6.4 (1995): 911-917.
[3] Li, Zongyi, et al. "Fourier neural operator for parametric partial differential equations." arXiv preprint arXiv:2010.08895 (2020).
[4] Kovachki, Nikola, et al. "Neural operator: Learning maps between function spaces with applications to pdes." Journal of Machine Learning Research 24.89 (2023): 1-97.
[5] Zhang, Zecheng, Wing Tat Leung, and Hayden Schaeffer. "BelNet: Basis enhanced learning, a mesh-free neural operator." arXiv preprint arXiv:2212.07336 (2022).
[6] Raonic, Bogdan, et al. "Convolutional neural operators." ICLR 2023 Workshop on Physics for Machine Learning. 2023.
[7] Hao, Zhongkai, et al. "Gnot: A general neural operator transformer for operator learning." International Conference on Machine Learning. PMLR, 2023.
[8] Li, Zijie, Kazem Meidani, and Amir Barati Farimani. "Transformer for partial differential equations' operator learning." arXiv preprint arXiv:2205.13671 (2022).