Machine learning for accelerating acoustic simulations

IOP Physical acoustics tutorial day

Antonio Stanziola

Biomedical Ultrasound Group
Department of Medical Physics and Biomedical Engineering
University College London

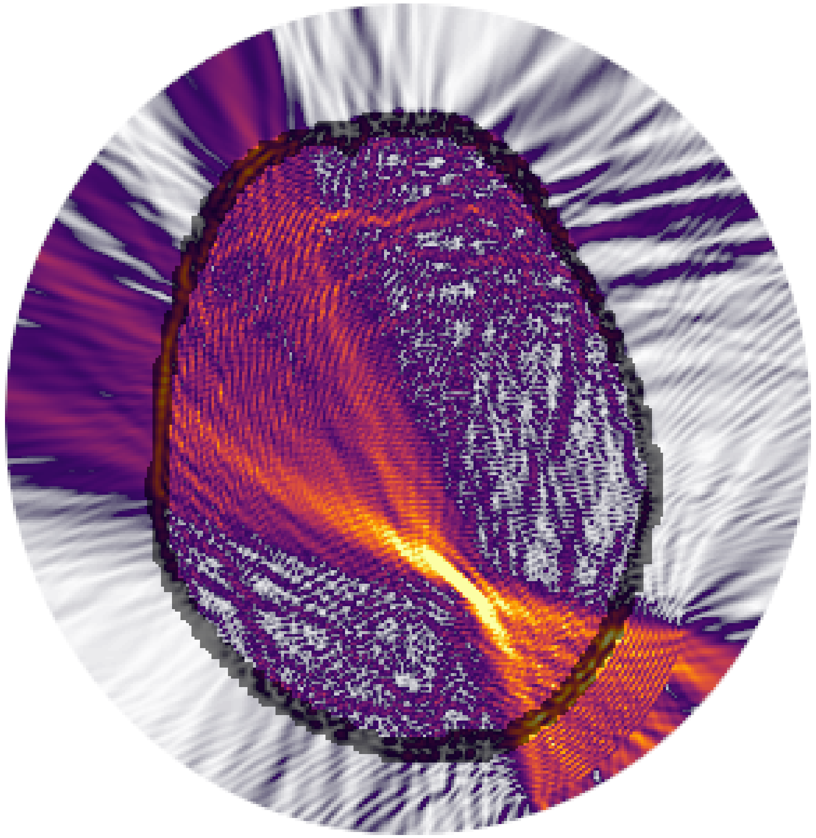
Transcranial ultrasound

A non-invasive method that uses acoustic energy to induce biological effects on the brain.

- Thermoablation
- BBB opening
- Neuromodulation

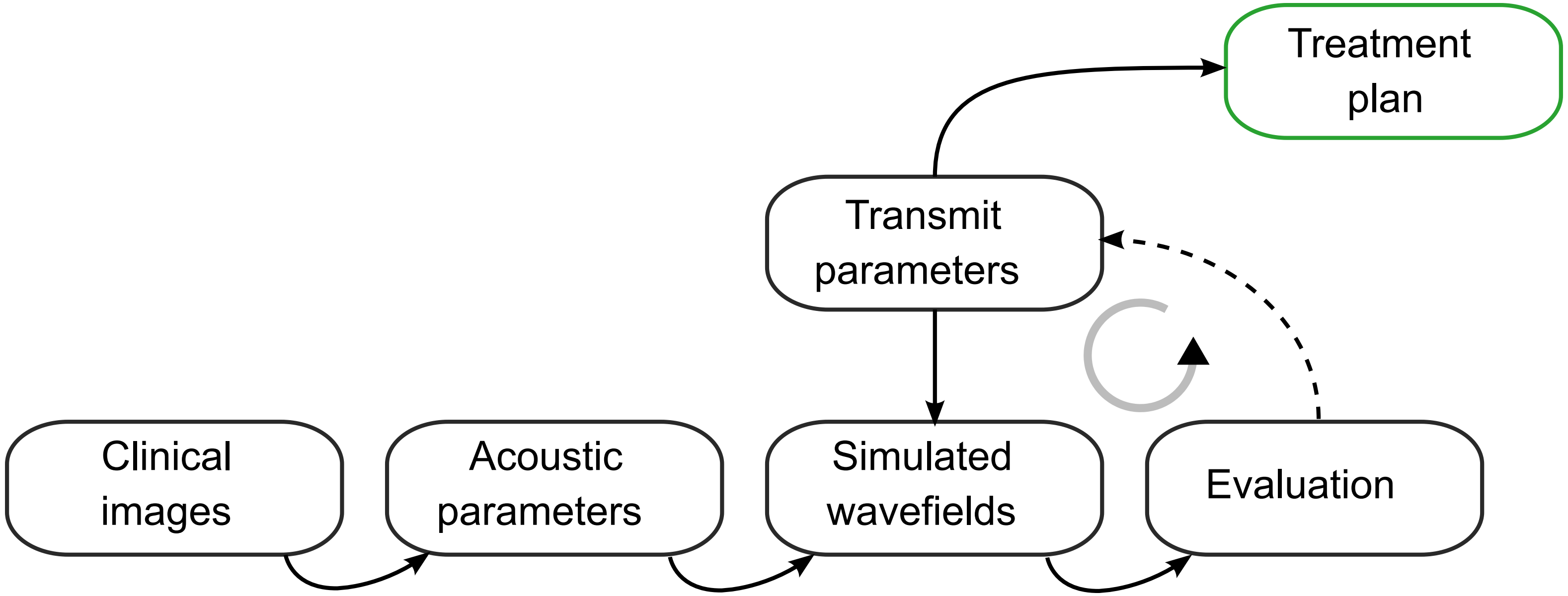
Requires a **treatment plan**:

- This includes the appropriate delays / phase shift for **focusing** the acoustic waves (beamforming).
- They must be tailored to the patient.
- Large intersubject variability of skull geometry and material properties.



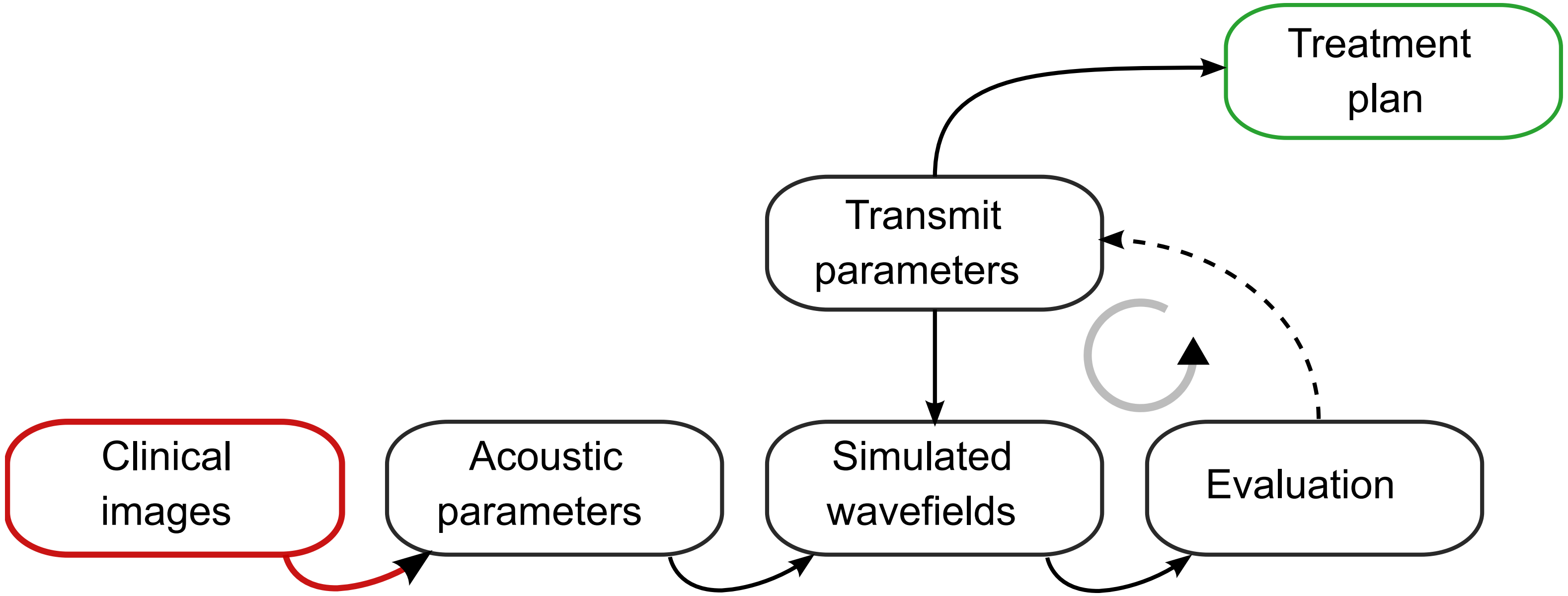
The ideal goal is **Real time treatment planning from MRI images**

In reality, there are many steps involved, some of which are uncertain while others are computationally demanding



The ideal goal is **Real time treatment planning from MRI images**

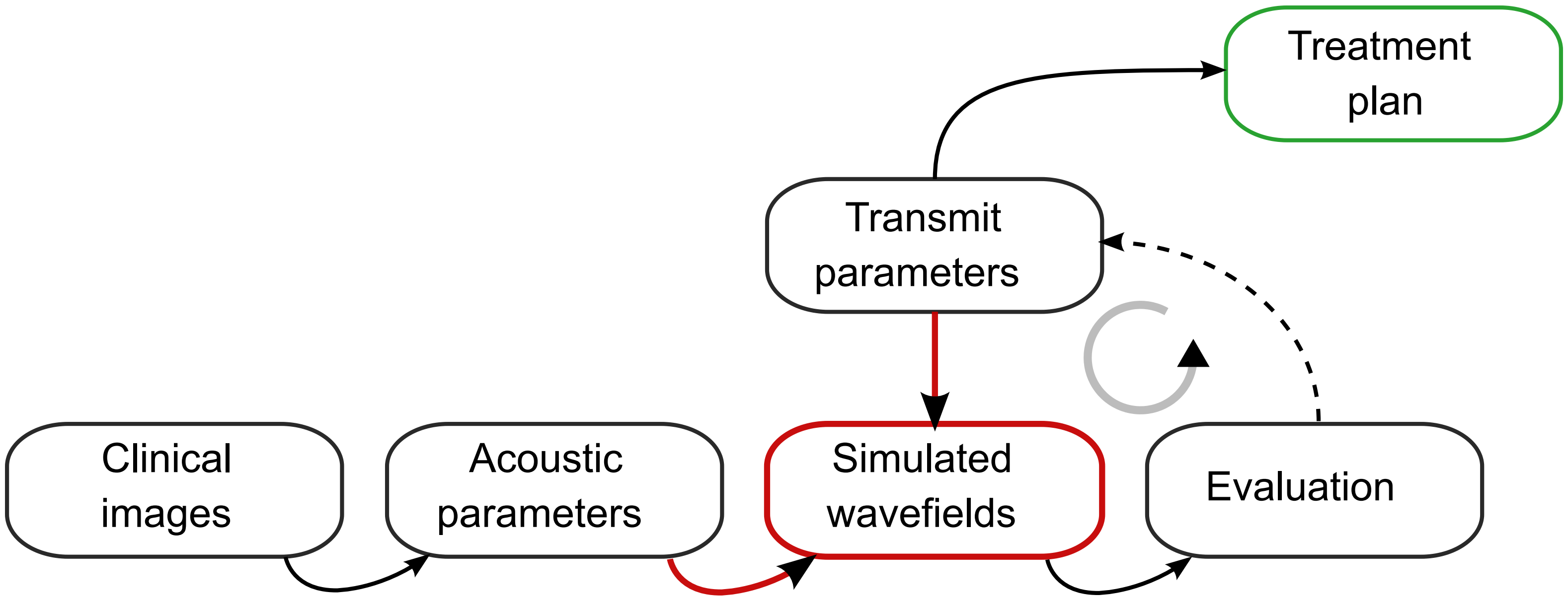
In reality, there are many steps involved, some of which are uncertain while others are computationally demanding



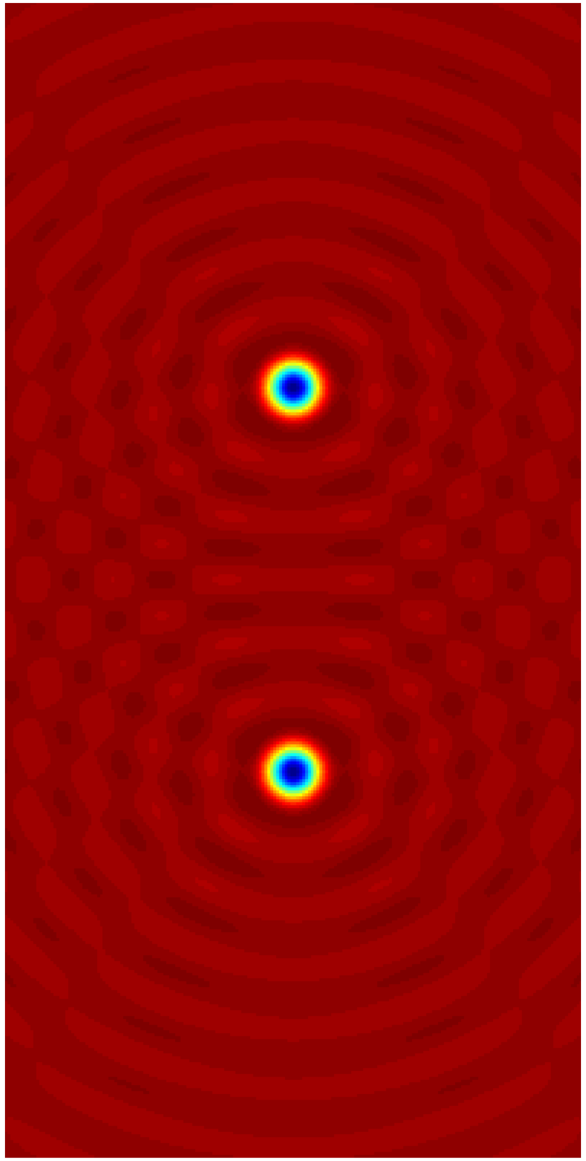
Acoustic parameters are currently indirectly derived from CT scans. Deep learing is being applied to do so from MRI acquisitions [1,2]

The ideal goal is **Real time treatment planning from MRI images**

In reality, there are many steps involved, some of which are uncertain while others are computationally demanding



We will focus on the simulation step, which mainly suffers from demanding computational requirements.



Helmholtz equation

Is derived from the wave equation by assuming single-frequency complex sources.

$$\mathcal{L}_c u = \left(\nabla^2 + \frac{\omega^2}{c^2}\right) u = s$$

The boundary conditions are given, at infinity, by the Sommerfield radiation condition. These can be approximated on a finite domain using an absorbing layer around the edges.

Despite depending on the speed of sound c, the operator $\mathcal{L}_c$ is **linear** over the space of complex functions: we can focus on solving the problem with a fixed source.

Challenges

Discretization:
Transform the problem into a its discrete counterpart

$$\mathcal{L}_c u = \rho \qquad \rightarrow \qquad \mathbf{L}_c \mathbf{u} = \boldsymbol{\rho}$$

- Finite Differences, **Pseudo Spectral**, Finite Elements, Neural Networks
- Generally heavy for real-life problems (~Gb per field)
- Finite Elements can reduce the size of the discrete problem: drawback is setup time for creating the mesh

Challenges

Solving the Helmholtz equation
Can solve the Helmholtz equation as the steady-state solution of the wave equation. The solution u^* is then approximated by running a wave simulator **long enough**.

Finding the solution u^* of the Helmholtz equation can also be casted as a minimization problem:

$$u^* = \arg\min_u \|\mathbf{L}_c \mathbf{u} - \boldsymbol{\rho}\|$$

Conditioning
Krylov methods, like GMRES, solve the problem in the subspace $\mathbf{K} = \{\boldsymbol{\rho}, \mathbf{L}_c\boldsymbol{\rho}, \mathbf{L}_c^2\boldsymbol{\rho}, \mathbf{L}_c^3\boldsymbol{\rho}, \dots\}$.
Convergence is slow or not guaranteed (restarted GMRES).

Domain decomposition methods [3]

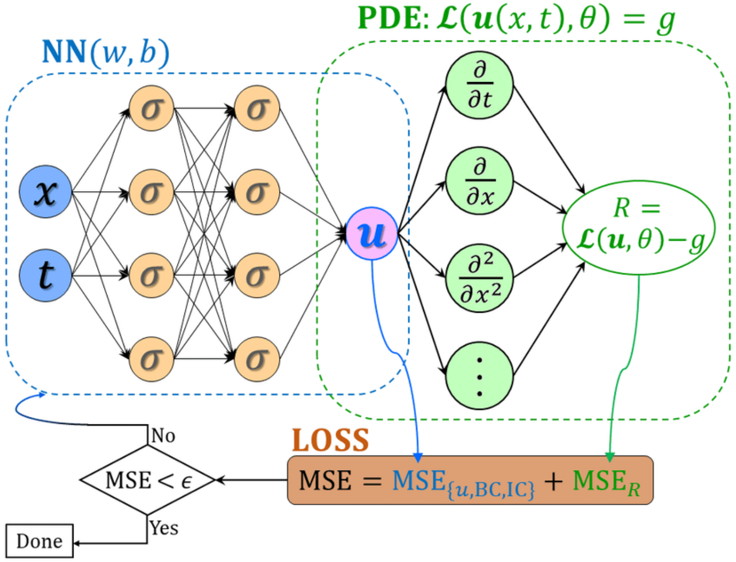
[3] *Gander, 2019*

Learned methods

Mesh-free

Use a neural network to represent the solution field [4]

from [Meng, 2019]



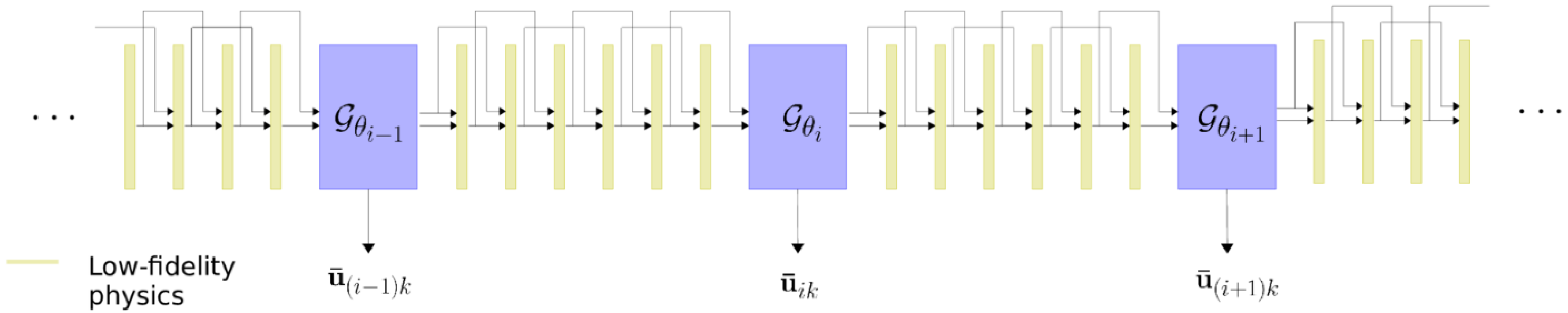
For the Helmholtz equation, sinusoidal activations [5] and attention mechanisms [6] have been suggested.

Learned methods

Error correction

Work by interleaving a neural network with an iterative solver

from [Siahkoohi, 2019]



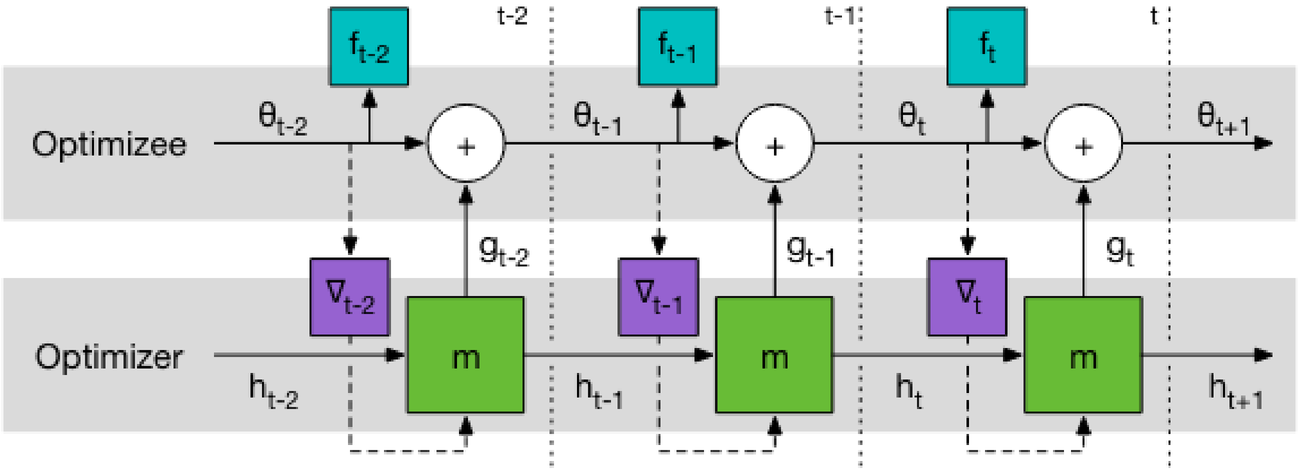
Proposed for the Helmholtz equation [7], time-domain wave solvers [8] and for parallel time integration [9]

Learned methods

Learning to minimize

By looking at the problem from the optimization point of view, an alternative approach is given by learning directly the minimization algorithm.

from [Andrychowicz, 2016]



Pioneered by [10], extended to inverse problems by [11]

[10] *Andrychowicz, 2016*; [11] *Putzky, 2017*

Learning a Helmholtz solver [12]

Learn an iterative update algorithm for the wavefield, conditioned on the heterogeneous speed of sound and source

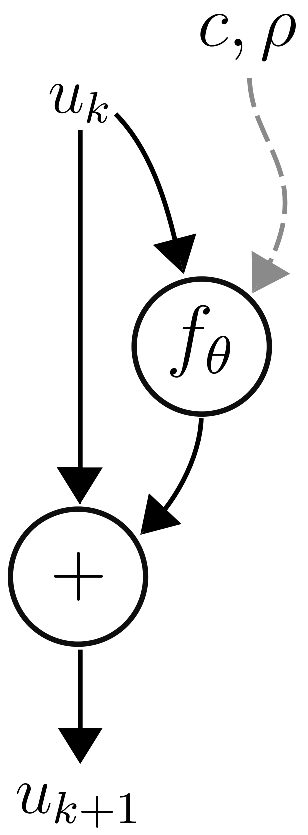
$$u_{k+1} = u_k + f_\theta(u_k, c, \rho)$$

We leverage the knowledge of the forward operator by making the dependene to c adn ρ implicit, using the **residual** e_k

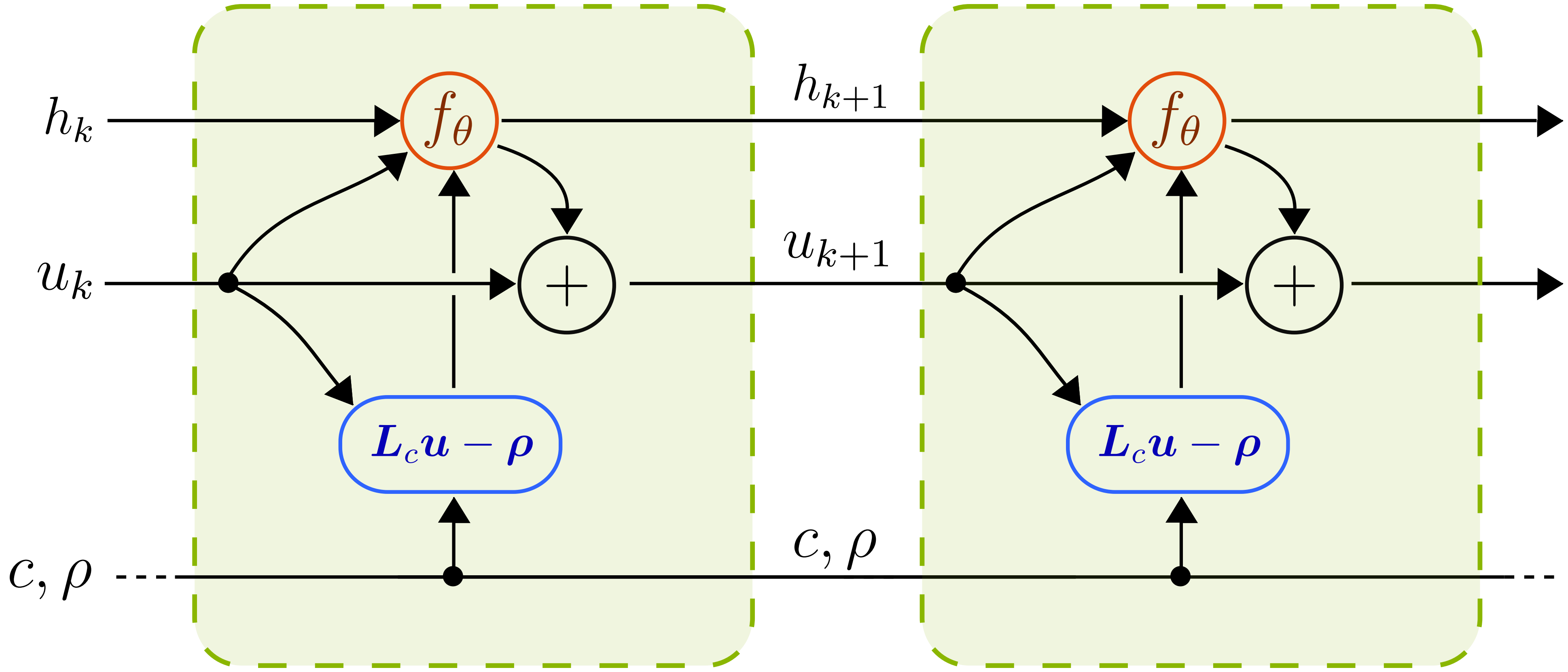
$$u_{k+1} = u_k + f_\theta(u_k, e_k) \qquad e_k = \mathbf{L}_c u_k - \rho$$

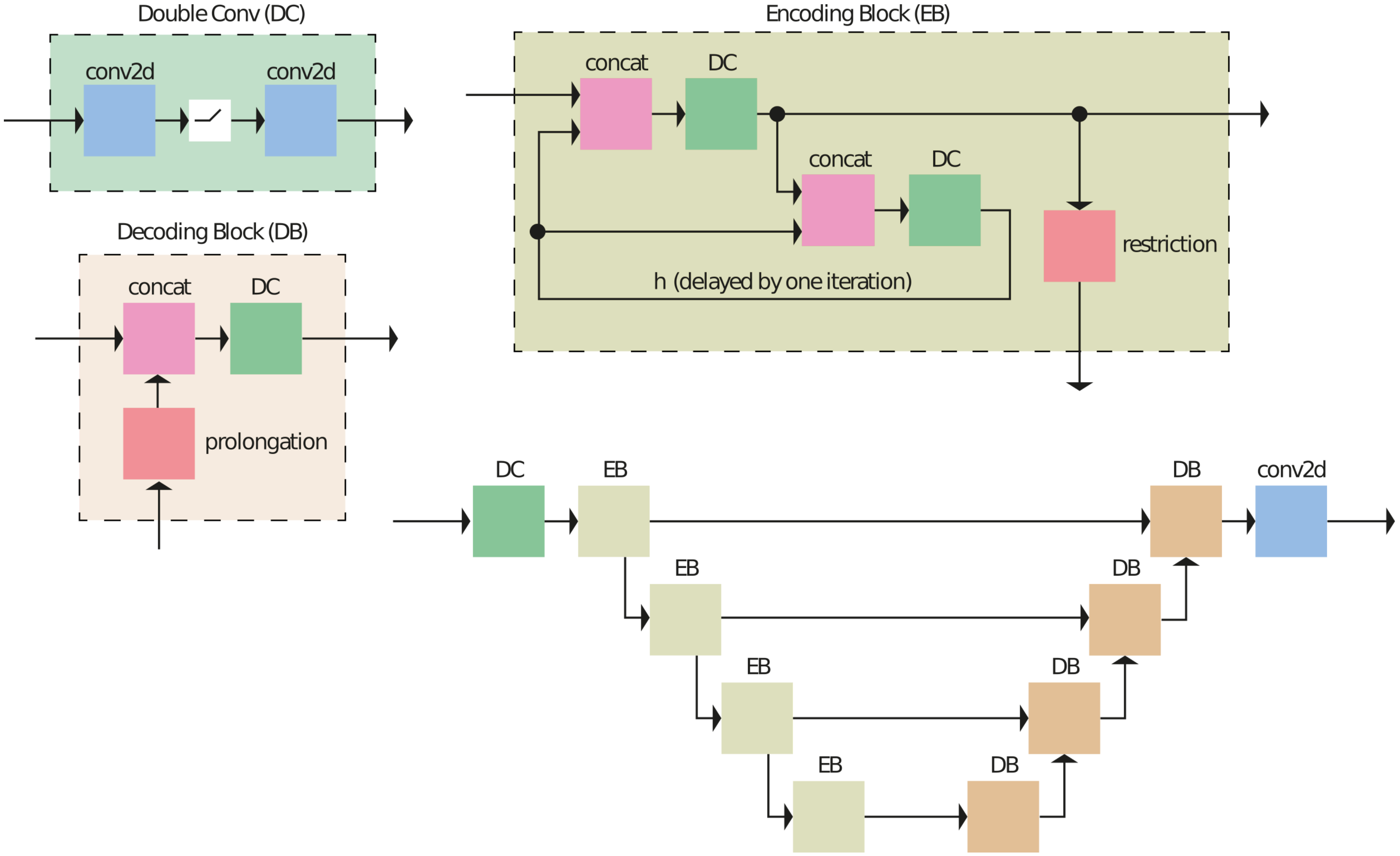
To strenghten the connection with iterative solvers, we add memory h_k of the previous states

$$u_{k+1} = u_k + \Delta u_{k+1} \qquad (\Delta u_{k+1}, h_{k+1}) = f_\theta(u_k, e_k, h_k)$$



The resulting model is a **recurrent neural network** which includes the discretized Helmholtz operator.





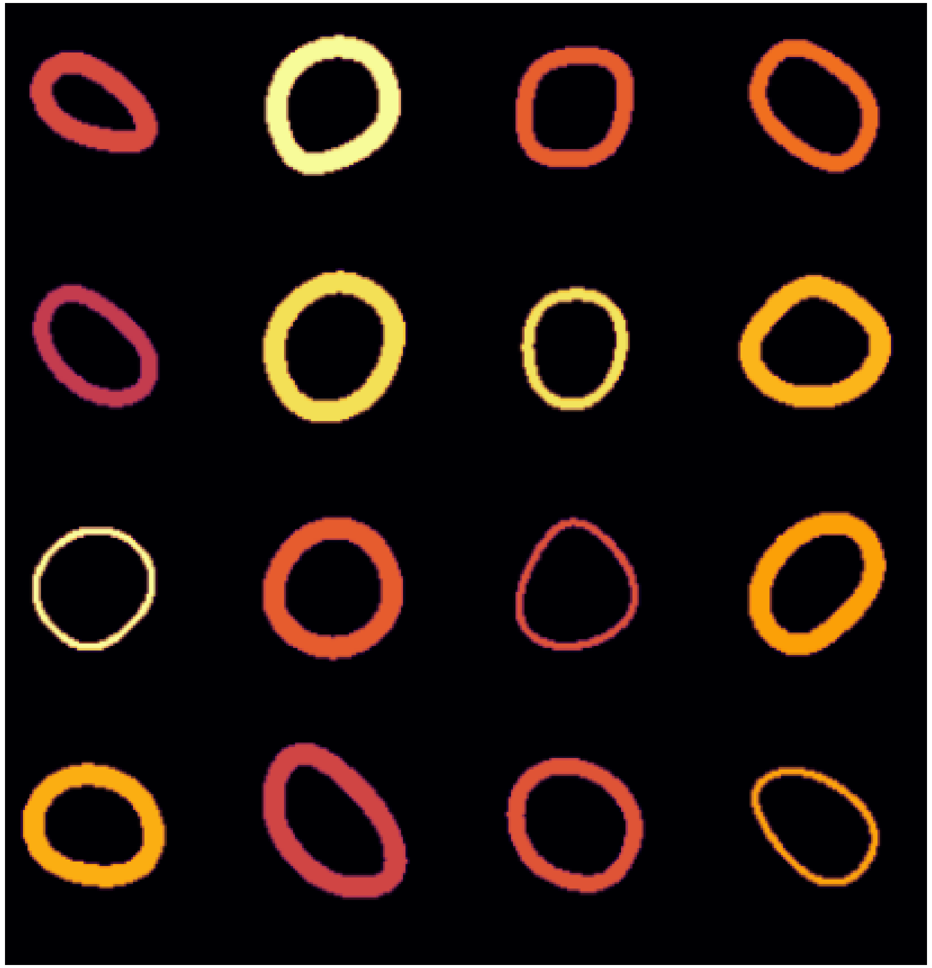
~47k parameters

Dataset

9000/1000/1000 **very** idealized skulls (96×96)

- Point-source at fixed location
  - Random location for validation
- Neural network with best validation loss at last iteration
  is saved
- No ground truth! Trained using physics loss

$$C_k = \|e_k\|^2$$

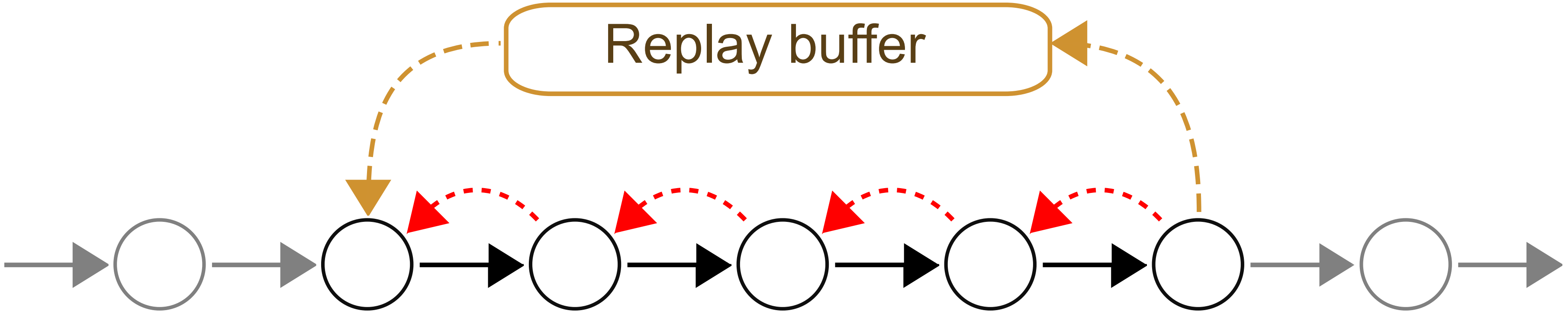


Training

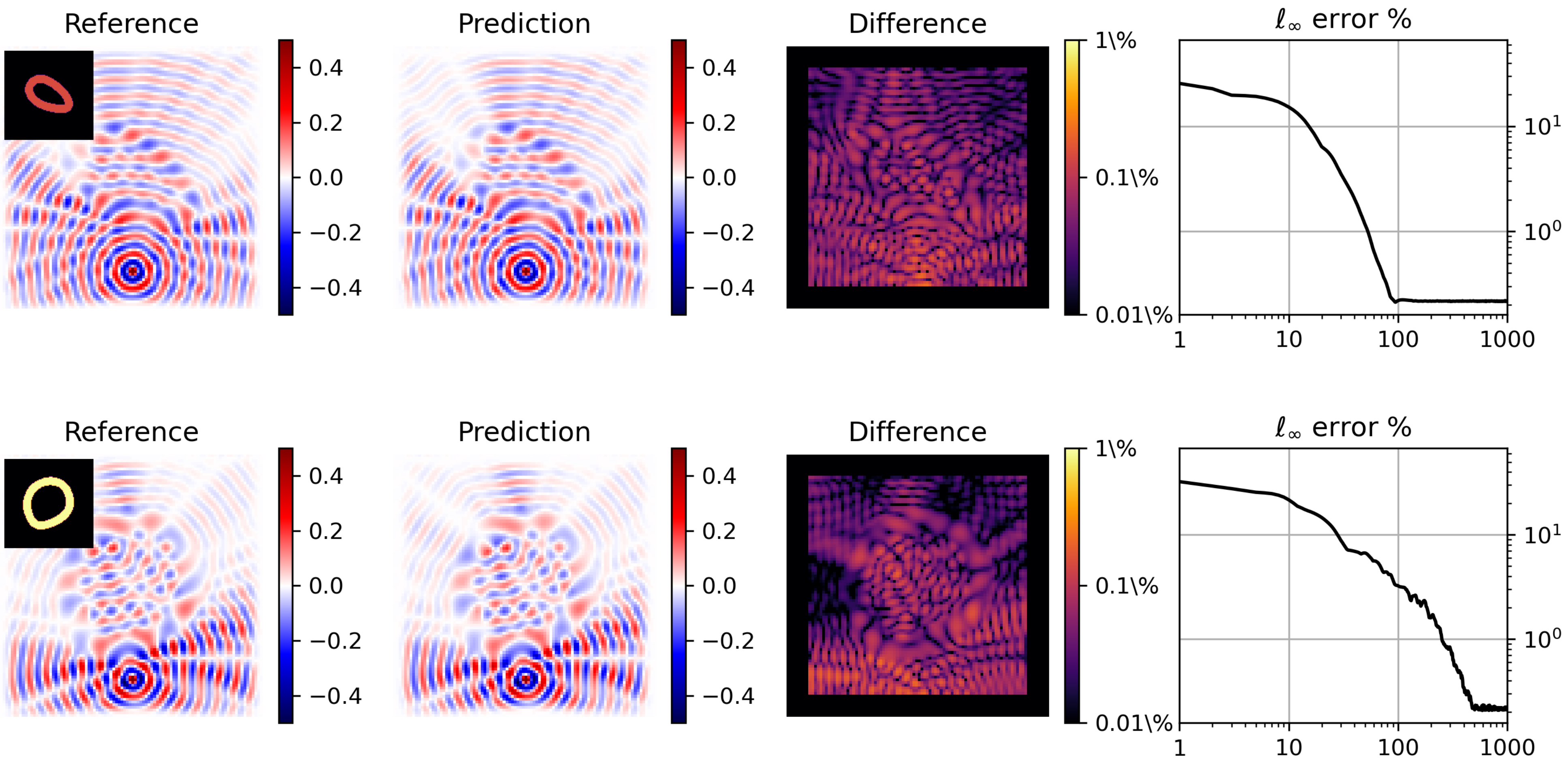
We use truncated backpropagation through time, to alleviate memory requirements

$$R = \sum_{k=0}^{T} C_k$$

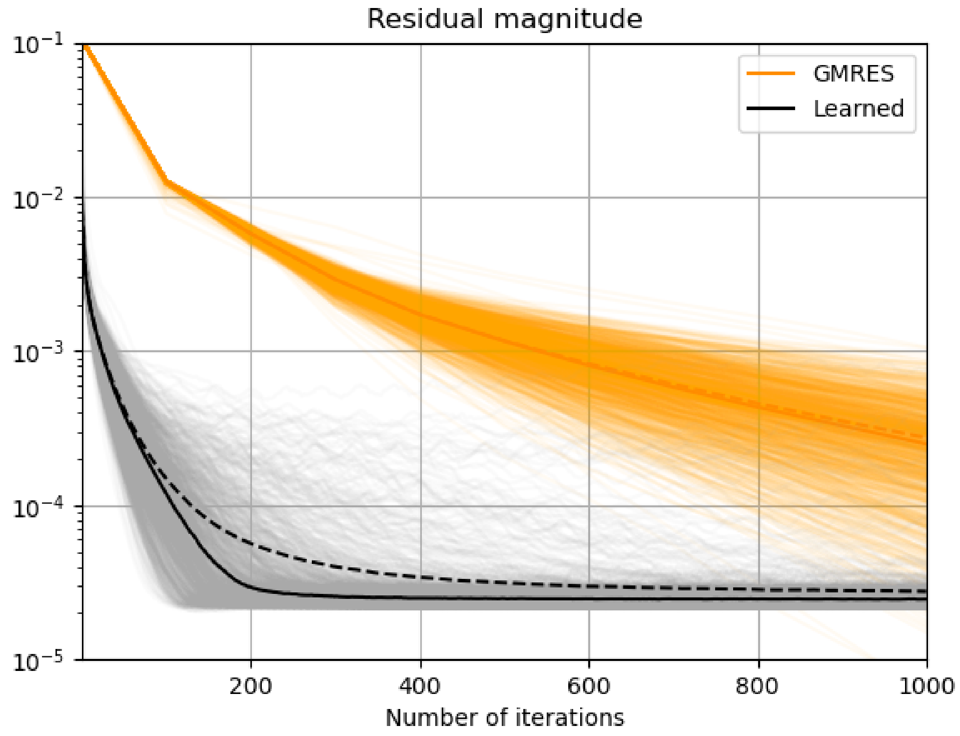
with $T = 10$, coupled with a replay buffer

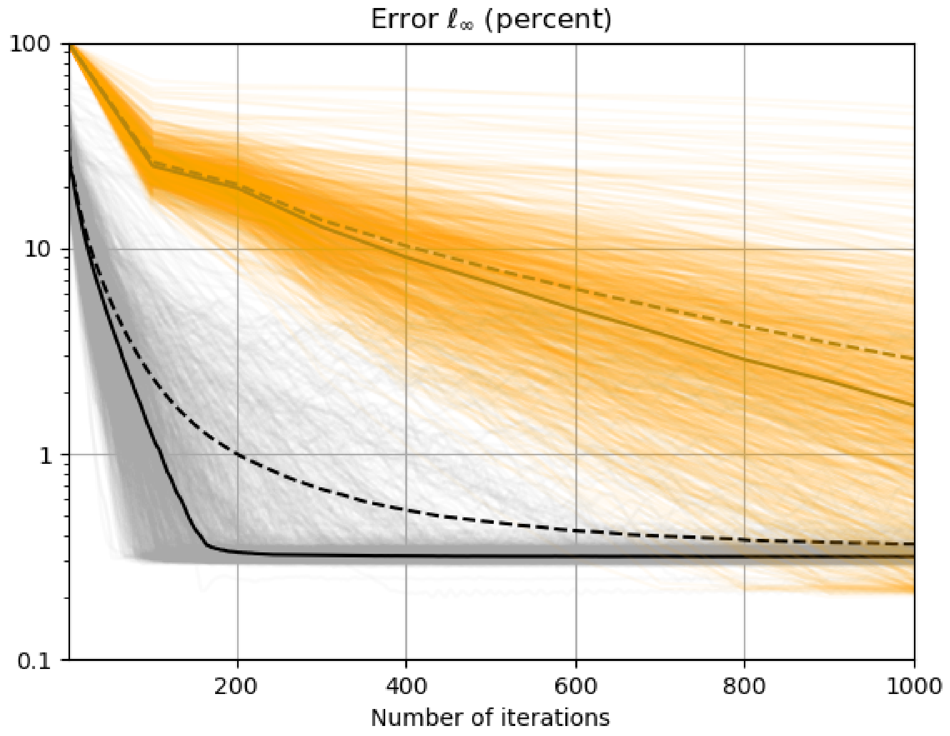


Results

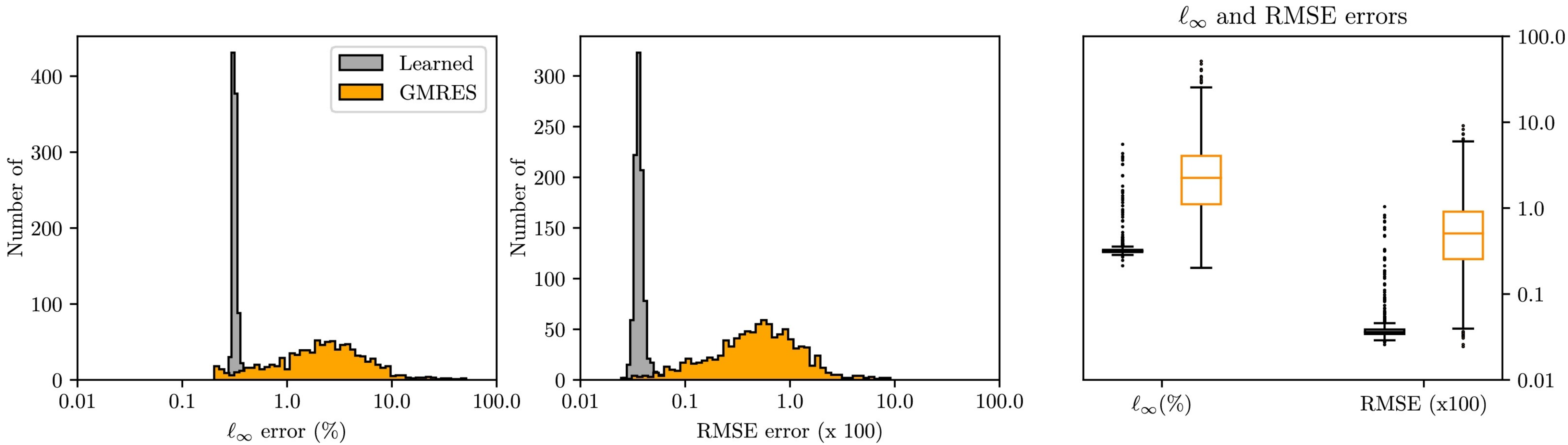

Reference
Prediction
Difference
ℓ_∞ error %
0.4
0.2
0.0
−0.2
−0.4
1\%
0.1\%
0.01\%
10^1
10^0
1
10
100
1000
Reference
Prediction
Difference
ℓ_∞ error %
0.4
0.2
0.0
−0.2
−0.4
1\%
0.1\%
0.01\%
10^1
10^0
1
10
100
1000

Convergence



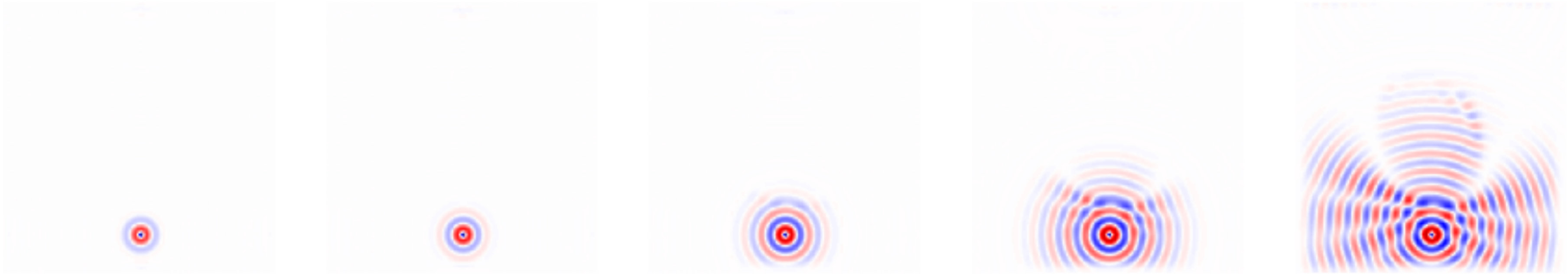


After 1000 iterations

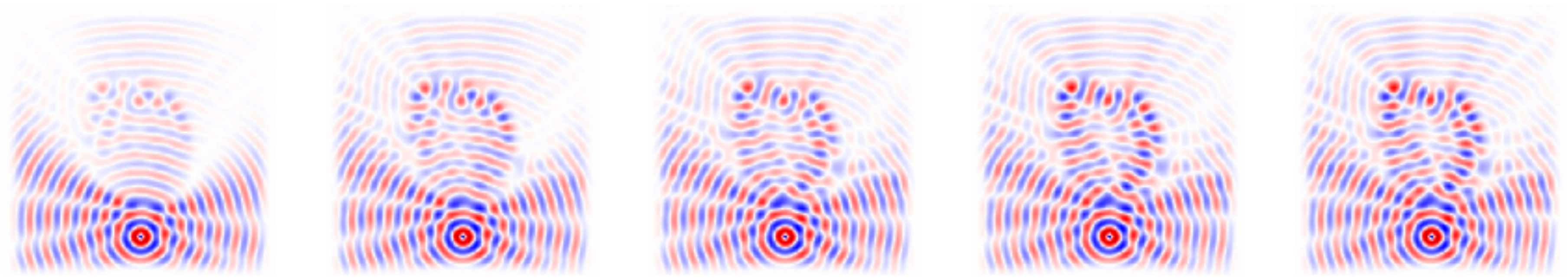


Compare with GMRES

Iterations: 10, 20, 50, 100, 250

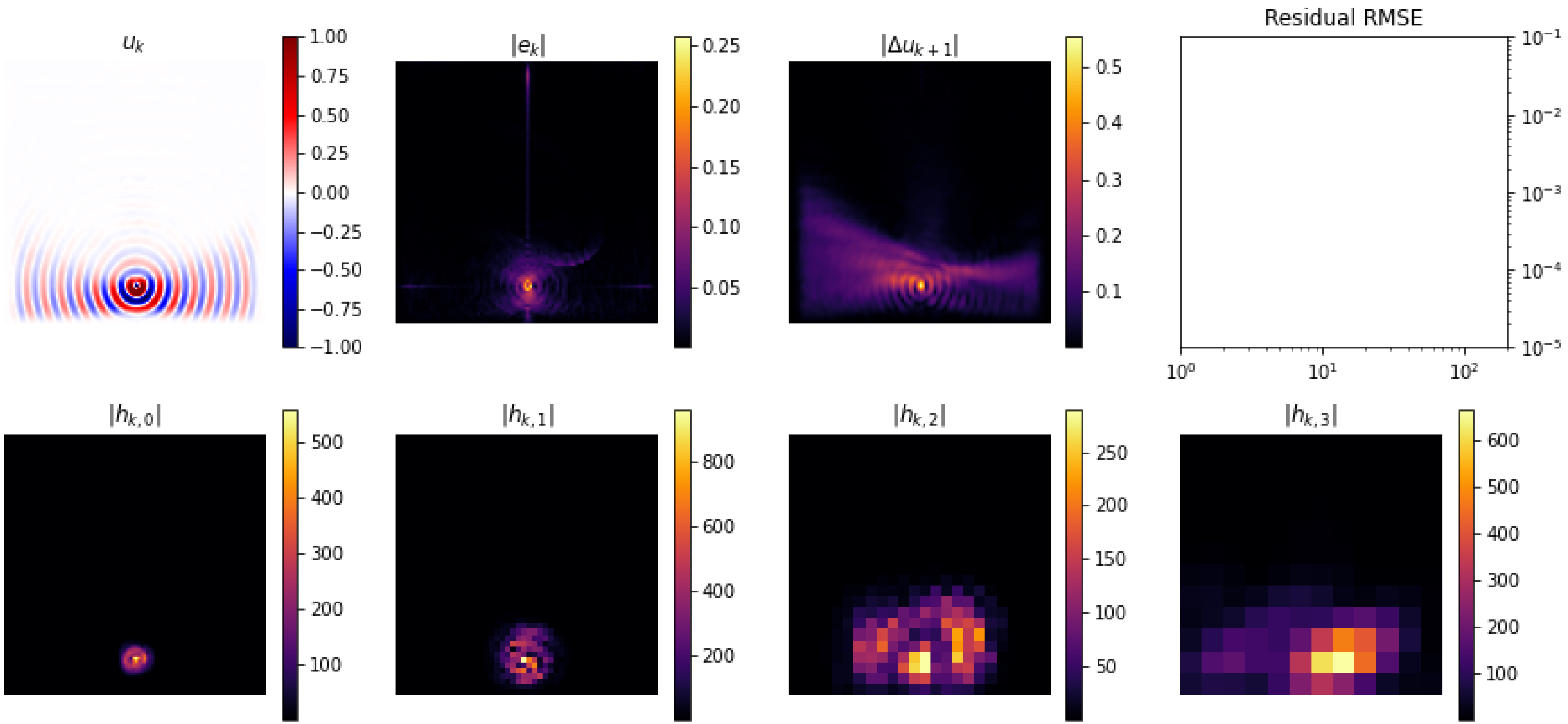


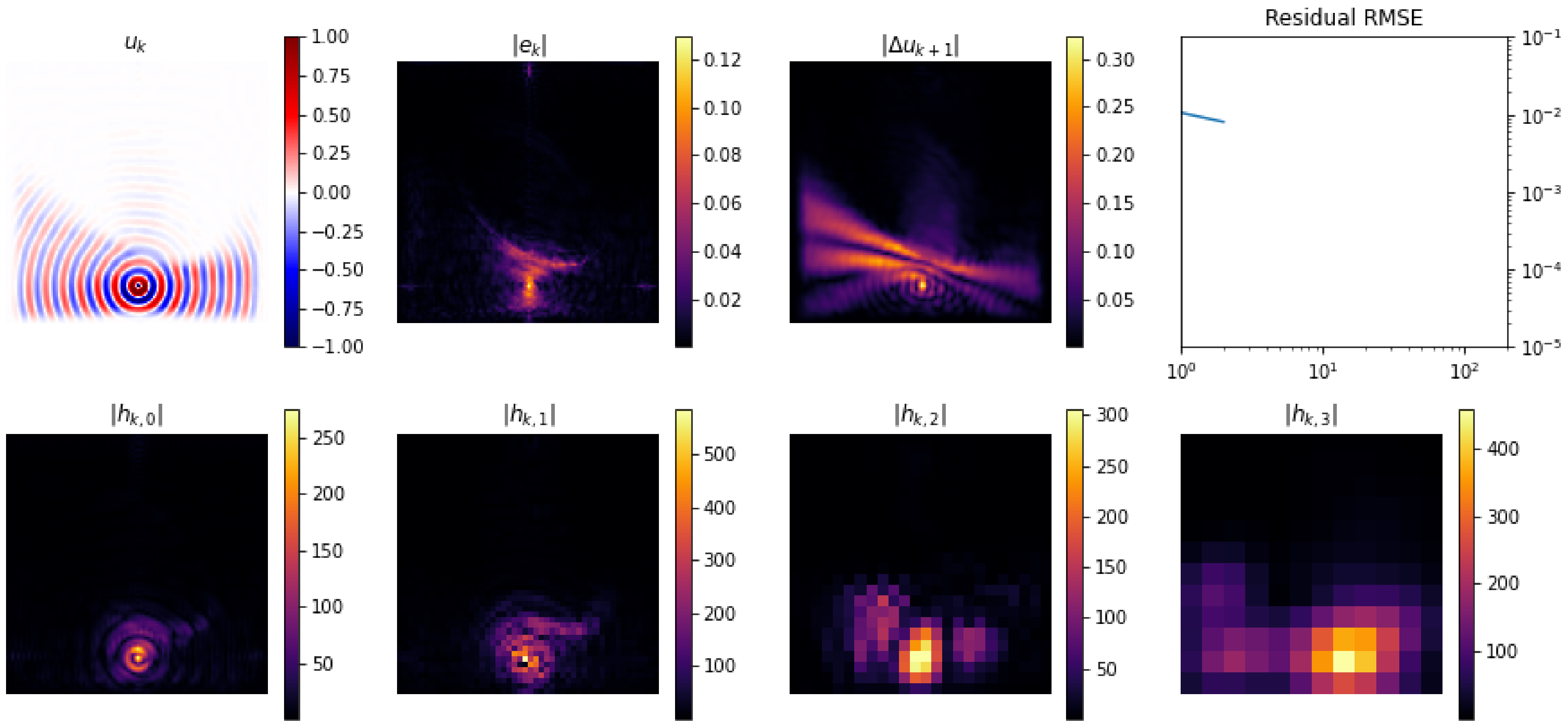
(a) Generalized minimal residual (GMRES) method.

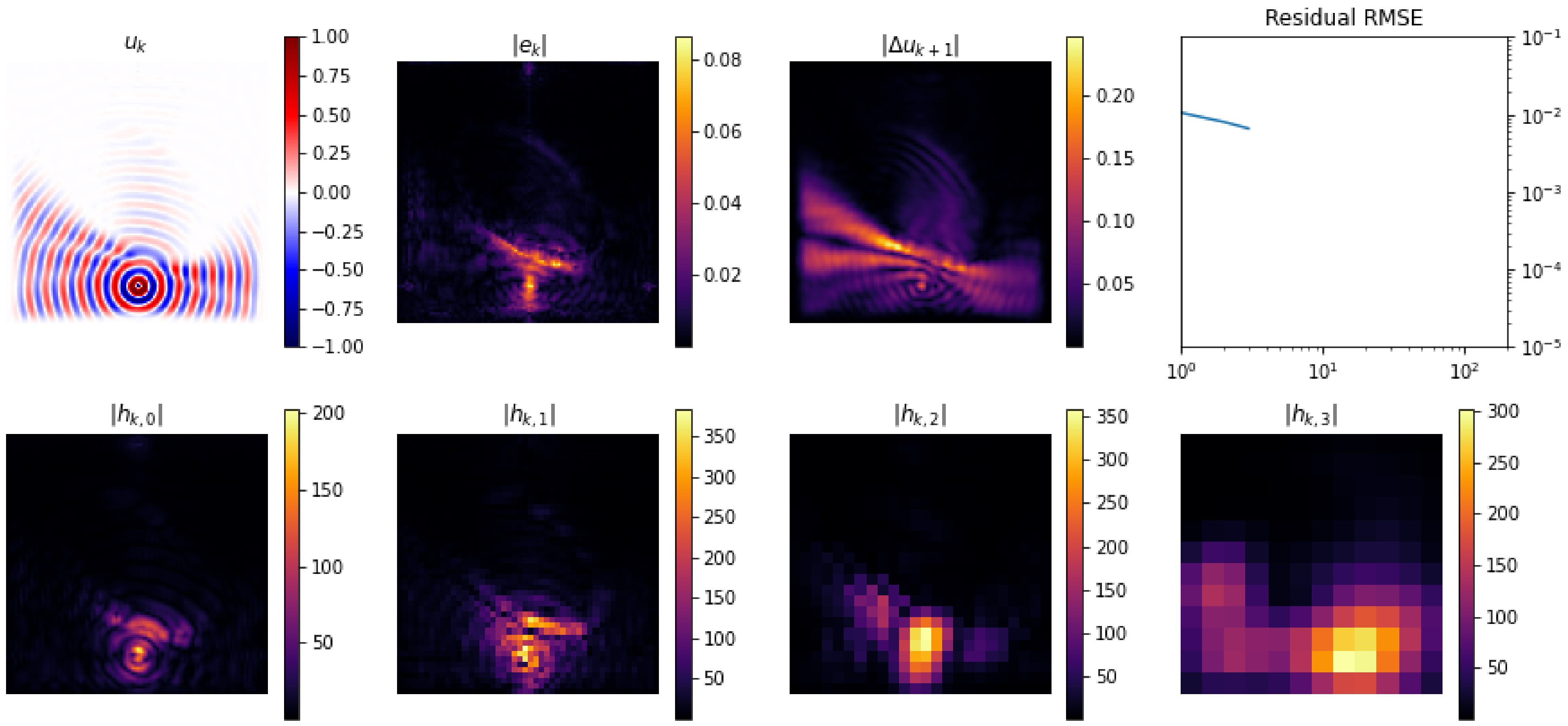


(b) Learned optimizer.

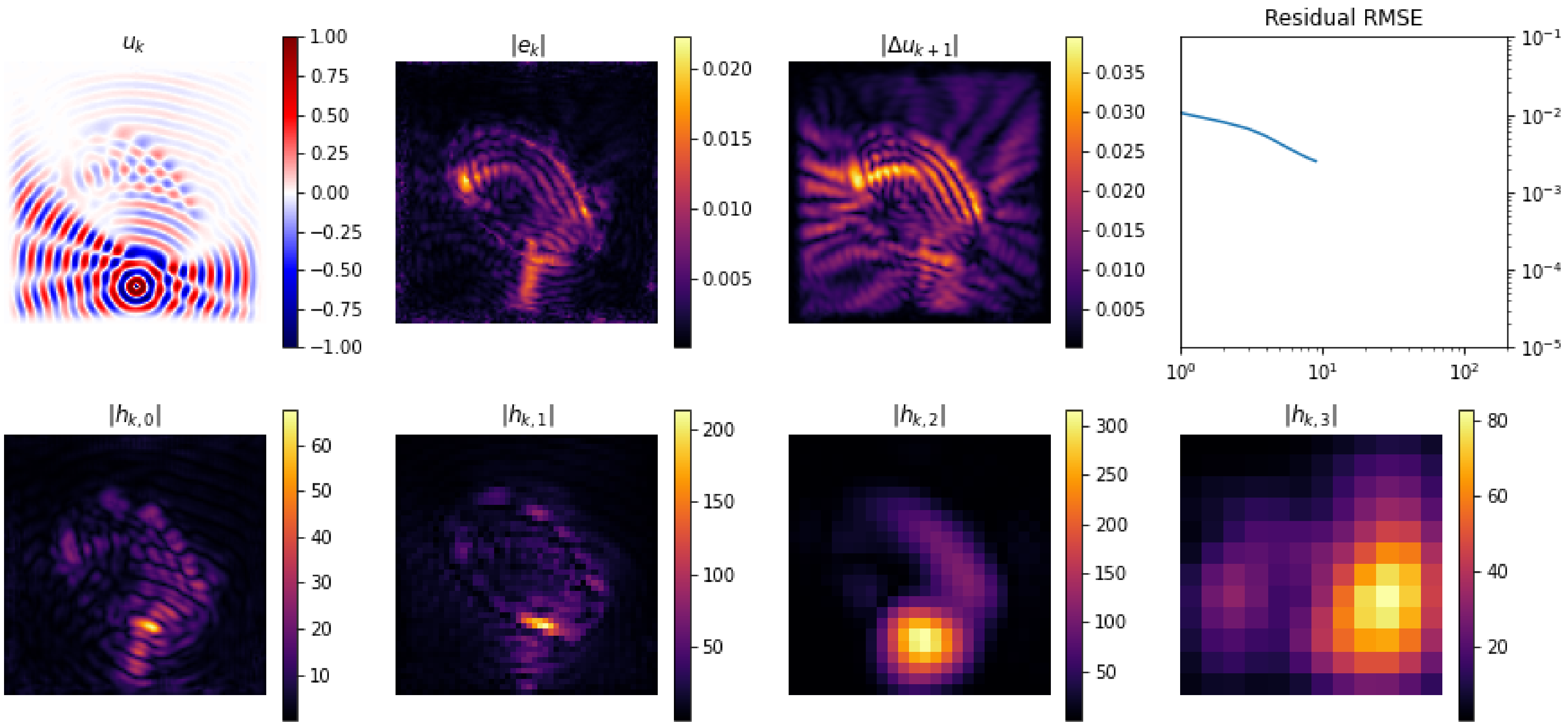

0:00 / 0:06


u_k
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
$|e_k|$
0.25
0.20
0.15
0.10
0.05
$|\Delta u_{k+1}|$
0.5
0.4
0.3
0.2
0.1
Residual RMSE
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^0
10^1
10^2
$|h_{k,0}|$
500
400
300
200
100
$|h_{k,1}|$
800
600
400
200
$|h_{k,2}|$
250
200
150
100
50
$|h_{k,3}|$
600
500
400
300
200
100

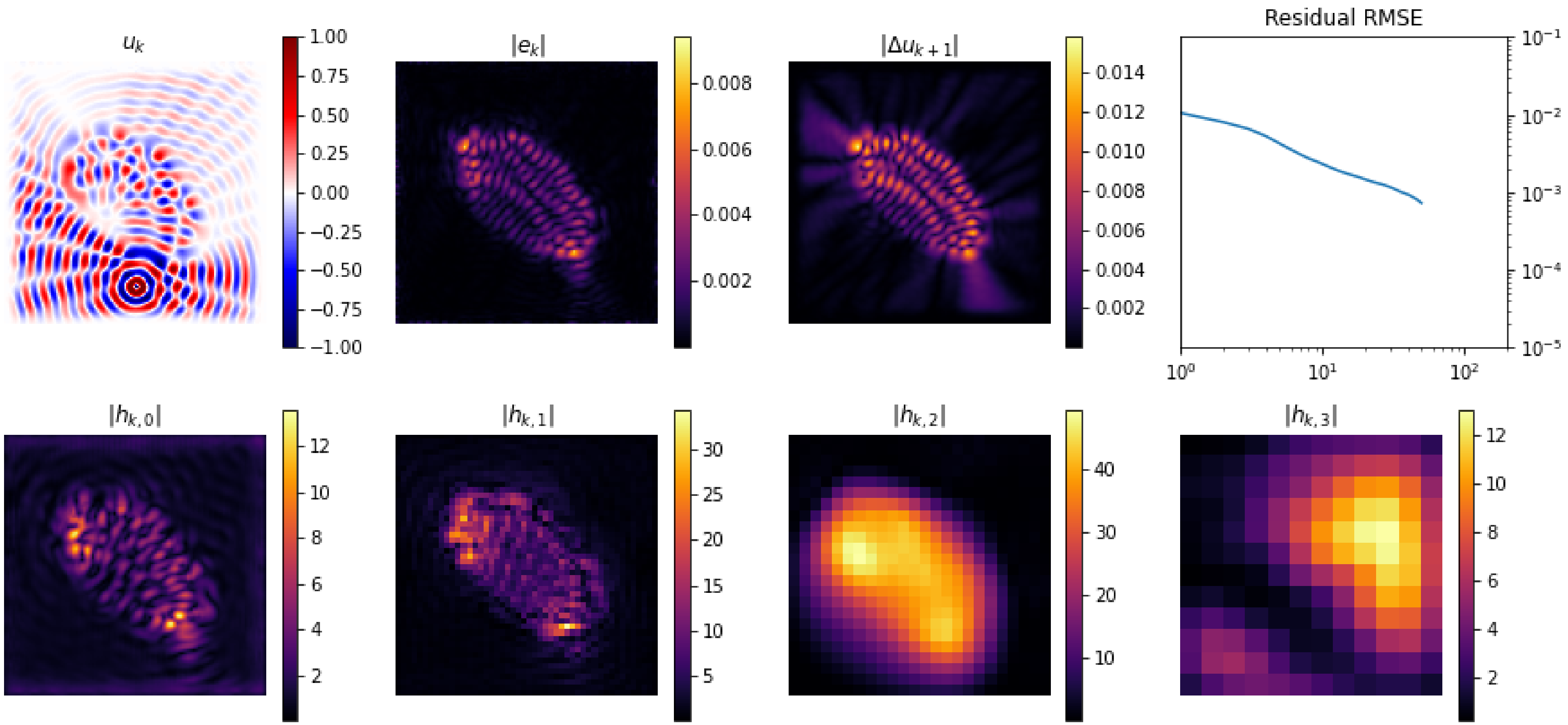

u_k
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
$|e_k|$
0.12
0.10
0.08
0.06
0.04
0.02
$|\Delta u_{k+1}|$
0.30
0.25
0.20
0.15
0.10
0.05
Residual RMSE
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^0
10^1
10^2
$|h_{k,0}|$
250
200
150
100
50
$|h_{k,1}|$
500
400
300
200
100
$|h_{k,2}|$
300
250
200
150
100
50
$|h_{k,3}|$
400
300
200
100



u_k
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
$|e_k|$
0.08
0.06
0.04
0.02
$|\Delta u_{k+1}|$
0.20
0.15
0.10
0.05
Residual RMSE
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^0
10^1
10^2
$|h_{k,0}|$
200
150
100
50
$|h_{k,1}|$
350
300
250
200
150
100
50
$|h_{k,2}|$
350
300
250
200
150
100
50
$|h_{k,3}|$
300
250
200
150
100
50



u_k
$|e_k|$
$|\Delta u_{k+1}|$
Residual RMSE
$|h_{k,0}|$
$|h_{k,1}|$
$|h_{k,2}|$
$|h_{k,3}|$

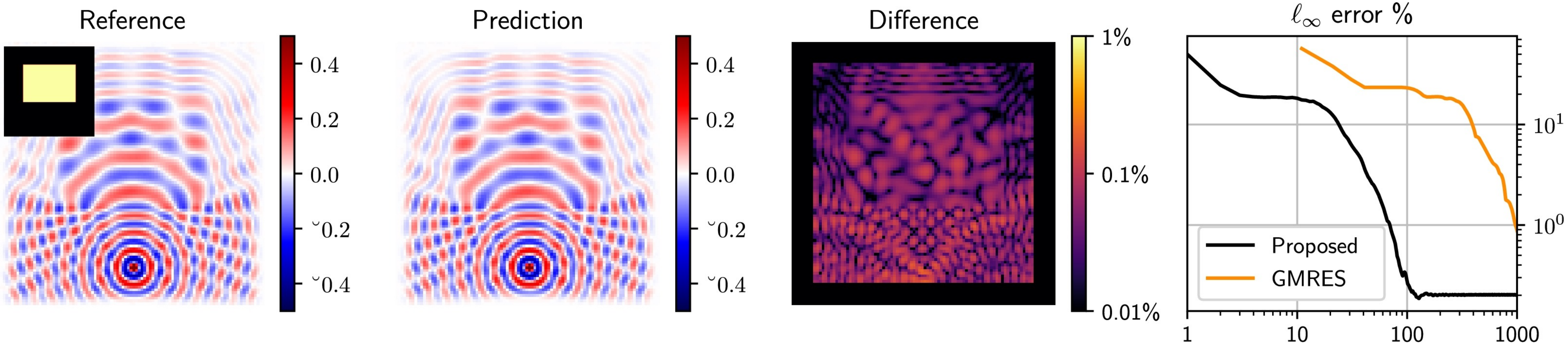


u_k
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
$|e_k|$
0.008
0.006
0.004
0.002
$|\Delta u_{k+1}|$
0.014
0.012
0.010
0.008
0.006
0.004
0.002
Residual RMSE
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{0}
10^{1}
10^{2}
$|h_{k,0}|$
12
10
8
6
4
2
$|h_{k,1}|$
30
25
20
15
10
5
$|h_{k,2}|$
40
30
20
10
$|h_{k,3}|$
12
10
8
6
4
2

Generalization

Different speeds of sound

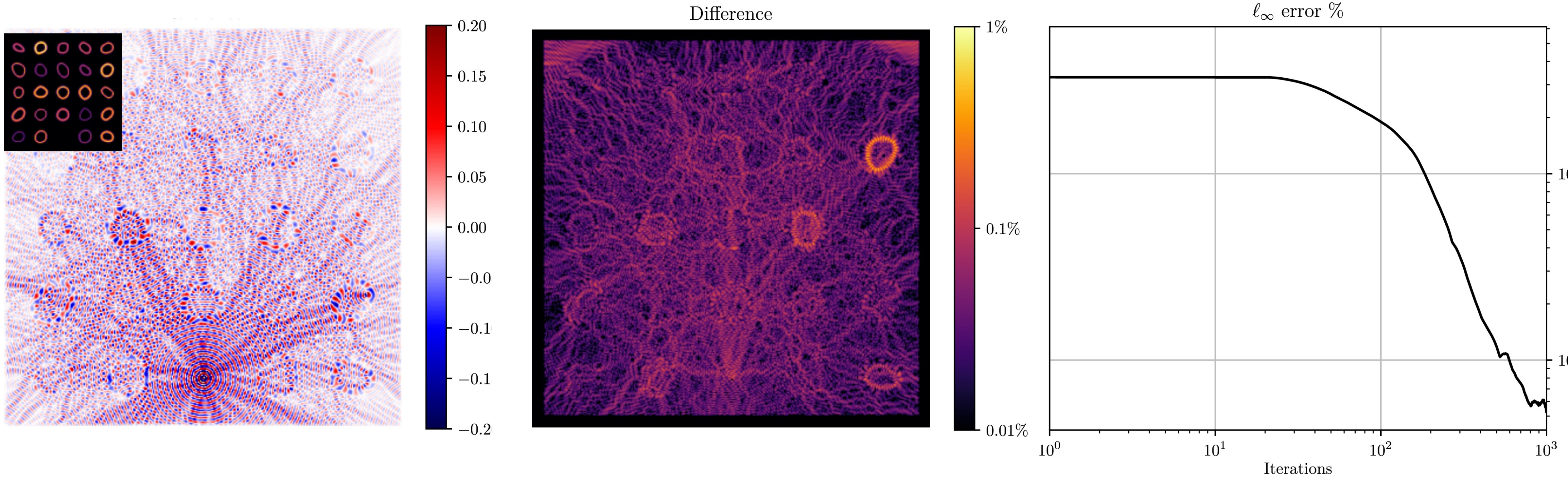
The network can generalize to speeds of sound maps not seen during traininng



It still maintains a speed advantage compared to GMRES.

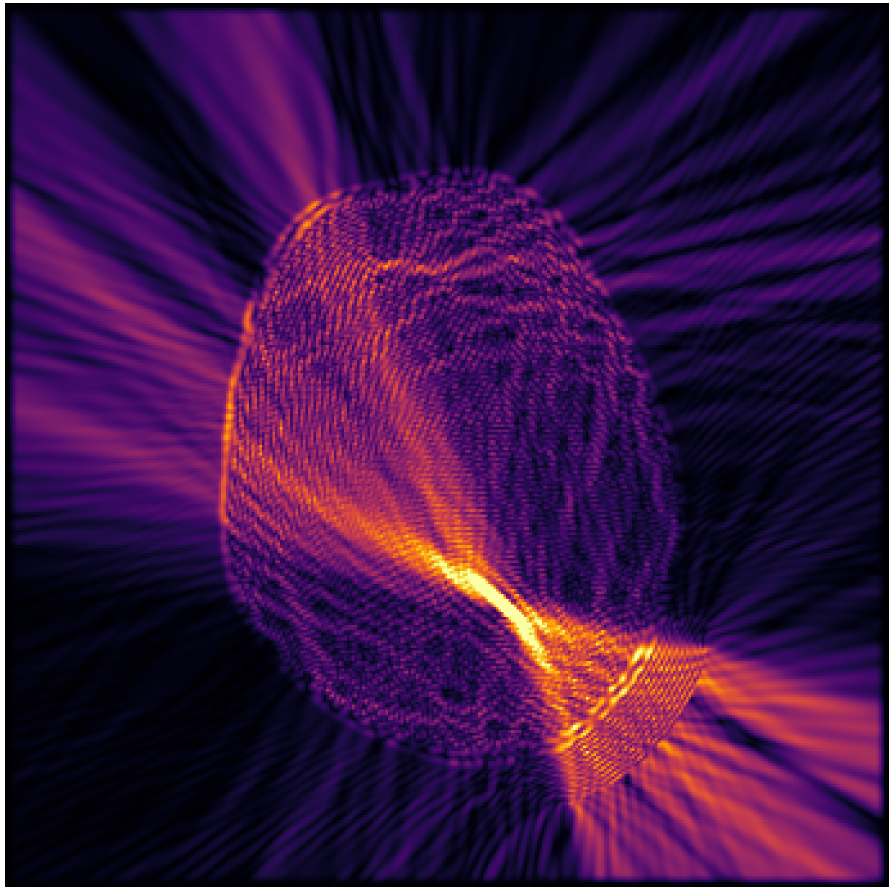
Generalization

Large domain

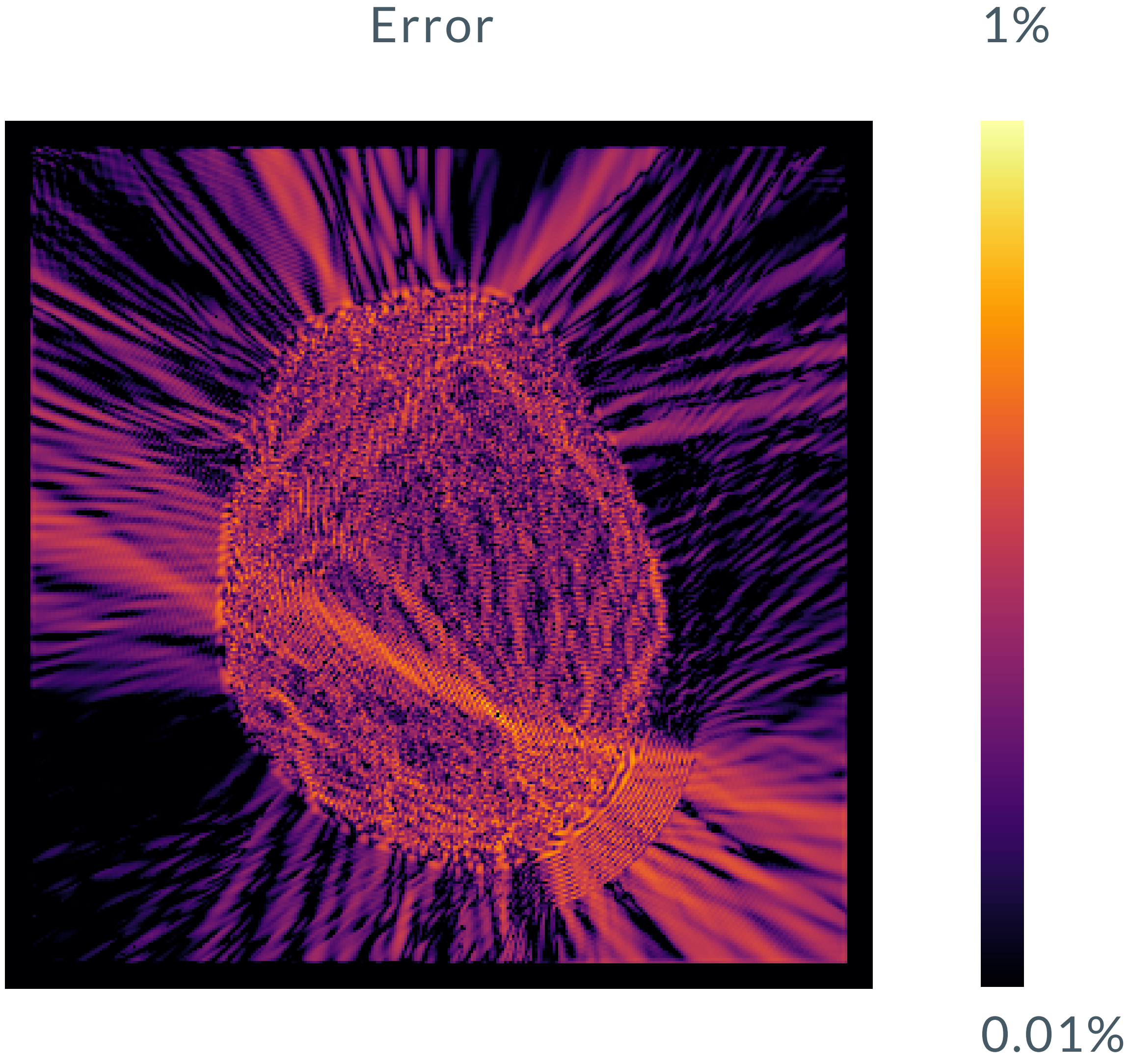



0:00 / 1:00

Predicted field

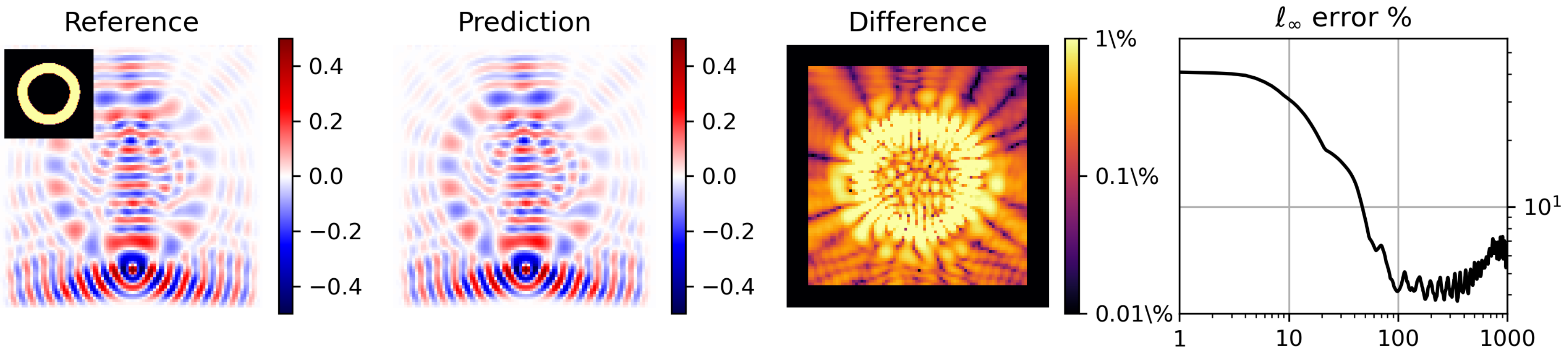


Error



Limitations

Some test-set outliers show large final errors



Presence of mode-like structures (similar to whispering gallery modes): it takes longer for time-domain solvers too in those examples

This doesn’t seem to happen on more complex sound speed distributions (e.g. real skull)

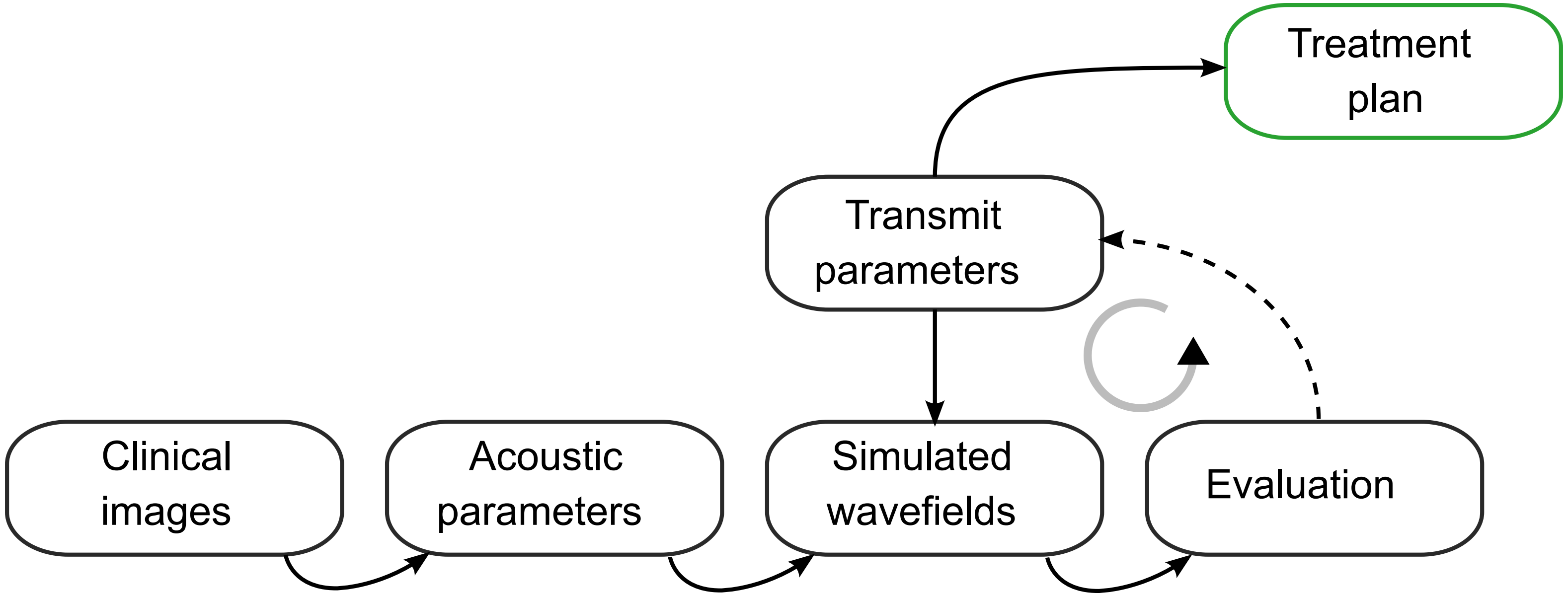
Conclusions

- Can learn a *lightweight* iterative solver for the heterogeneous Helmholtz equation
- No ground-truth samples are required
- The learned optimized can generalize well outside the training set.

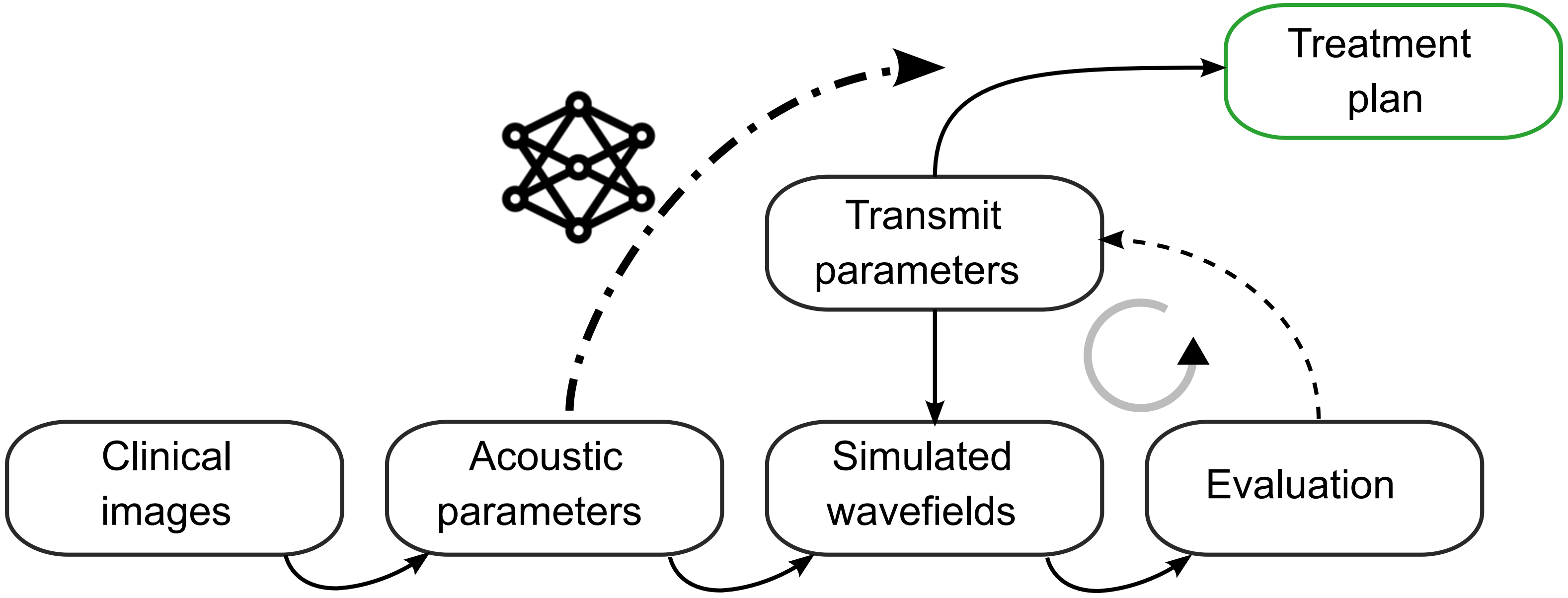
Open questions

- What's the best training strategy? Can the Replay Buffer be removed?
- How to ensure convergence of the iterative solver? What's a good loss function in absence of labeled data?
- How well does this translate in 3D?
- Can the problem be compressed to a more manageable size, rather than using small neural network models?
- Can we explicitly train for a fast solver?

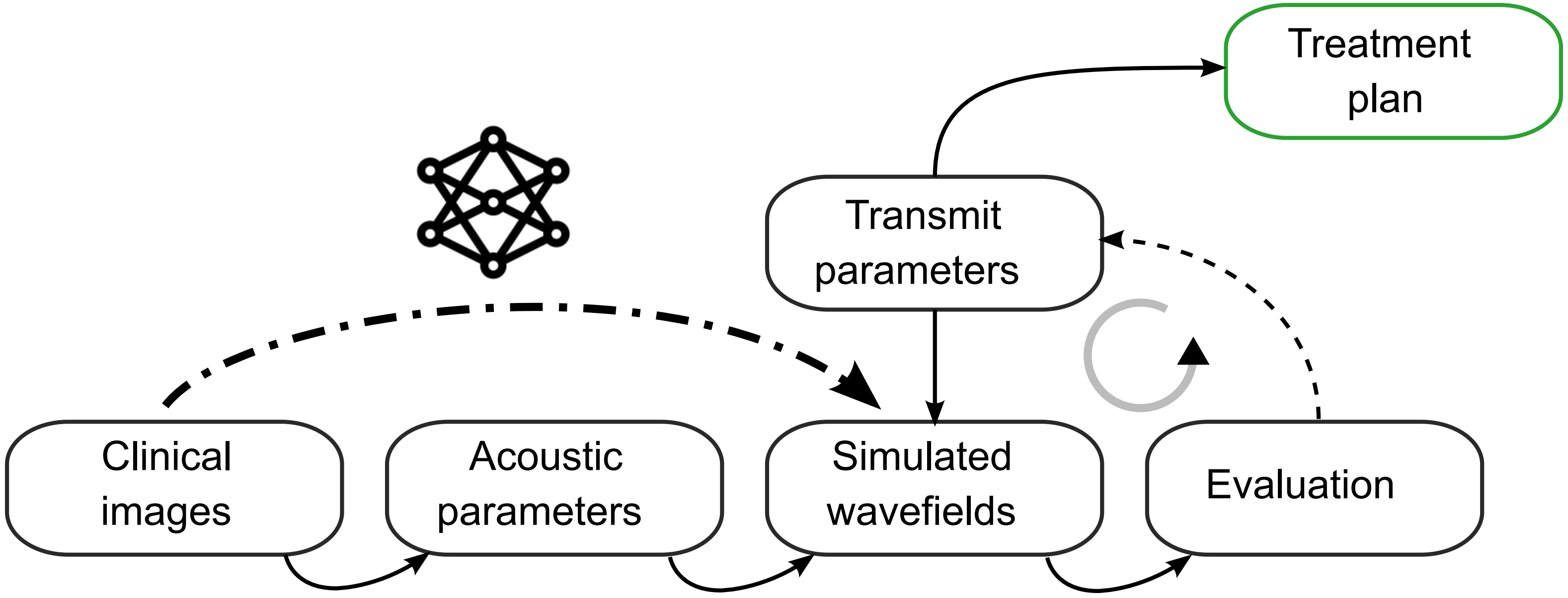
How can we get closer to the ideal goal?



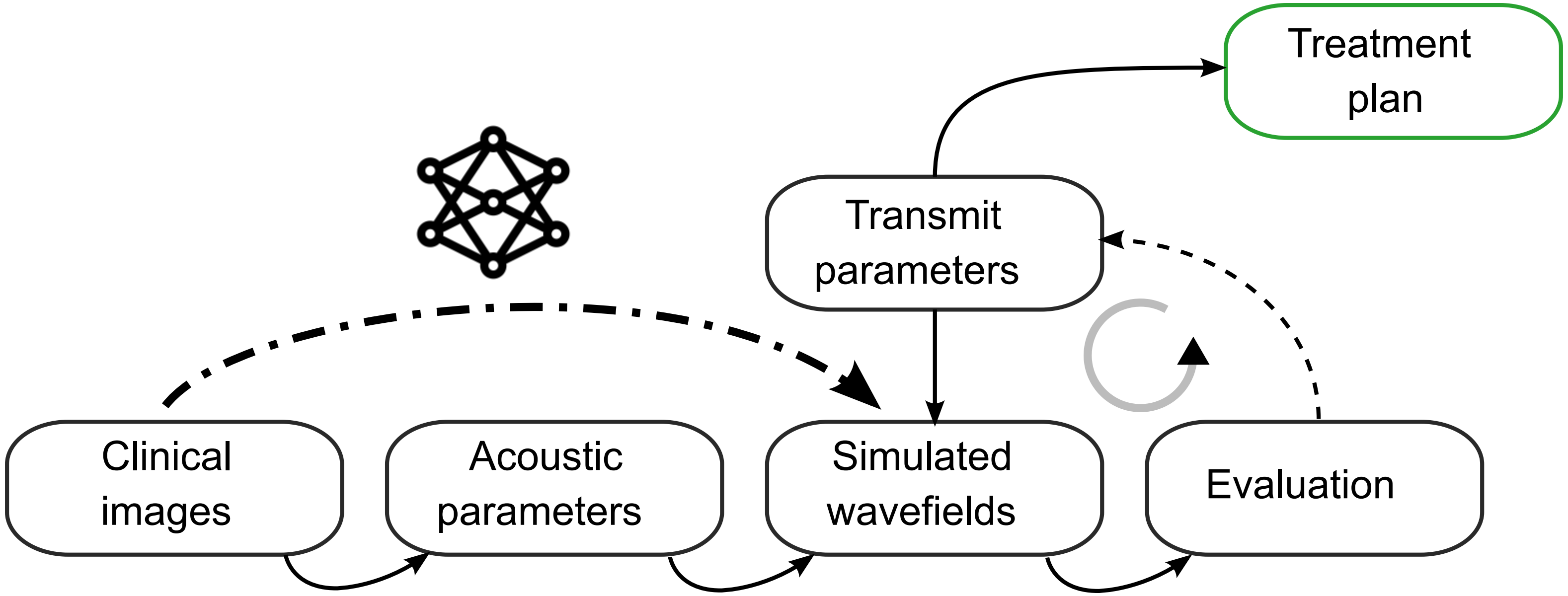
How can we get closer to the ideal goal?



How can we get closer to the ideal goal?

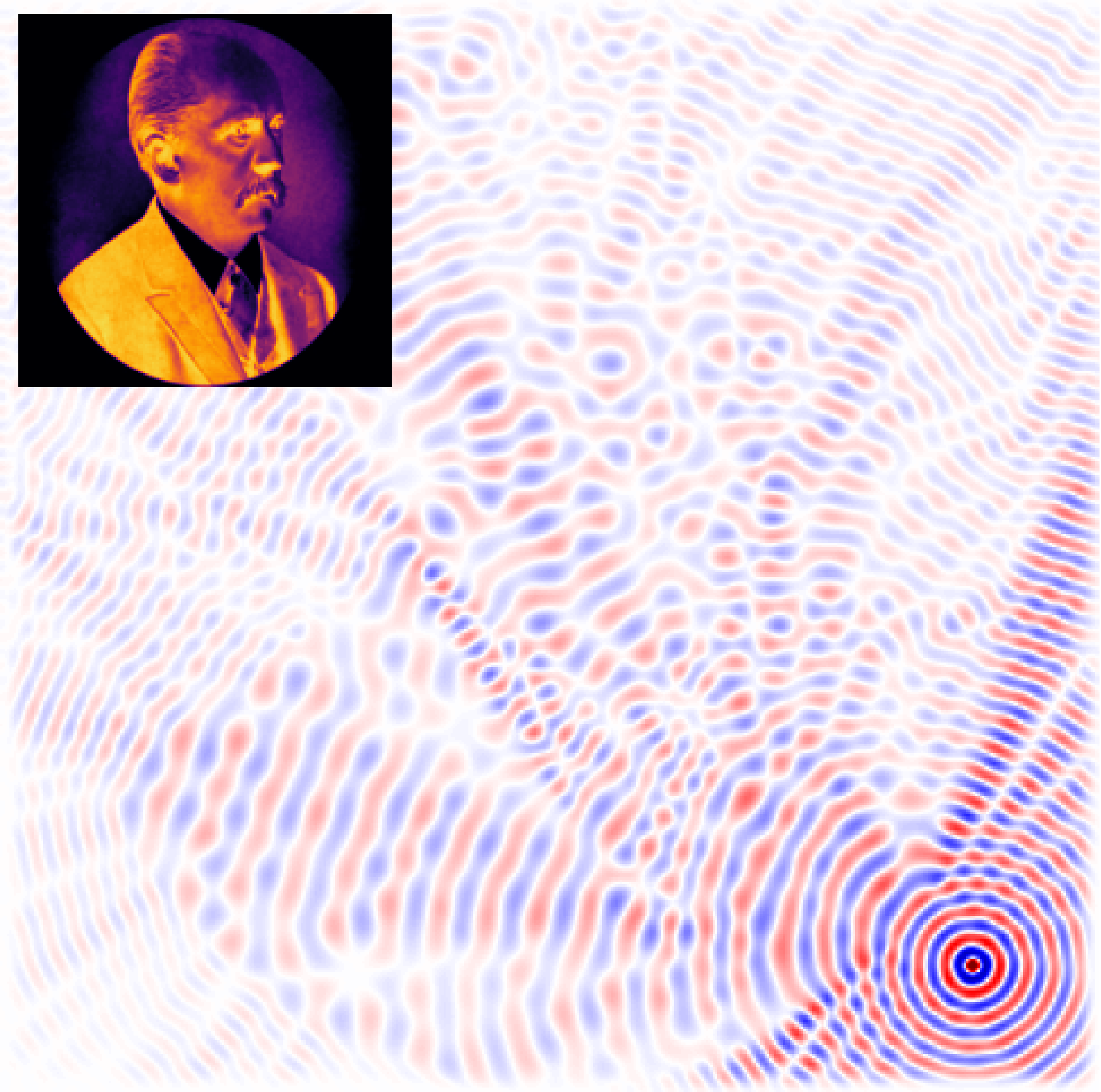


How can we get closer to the ideal goal?



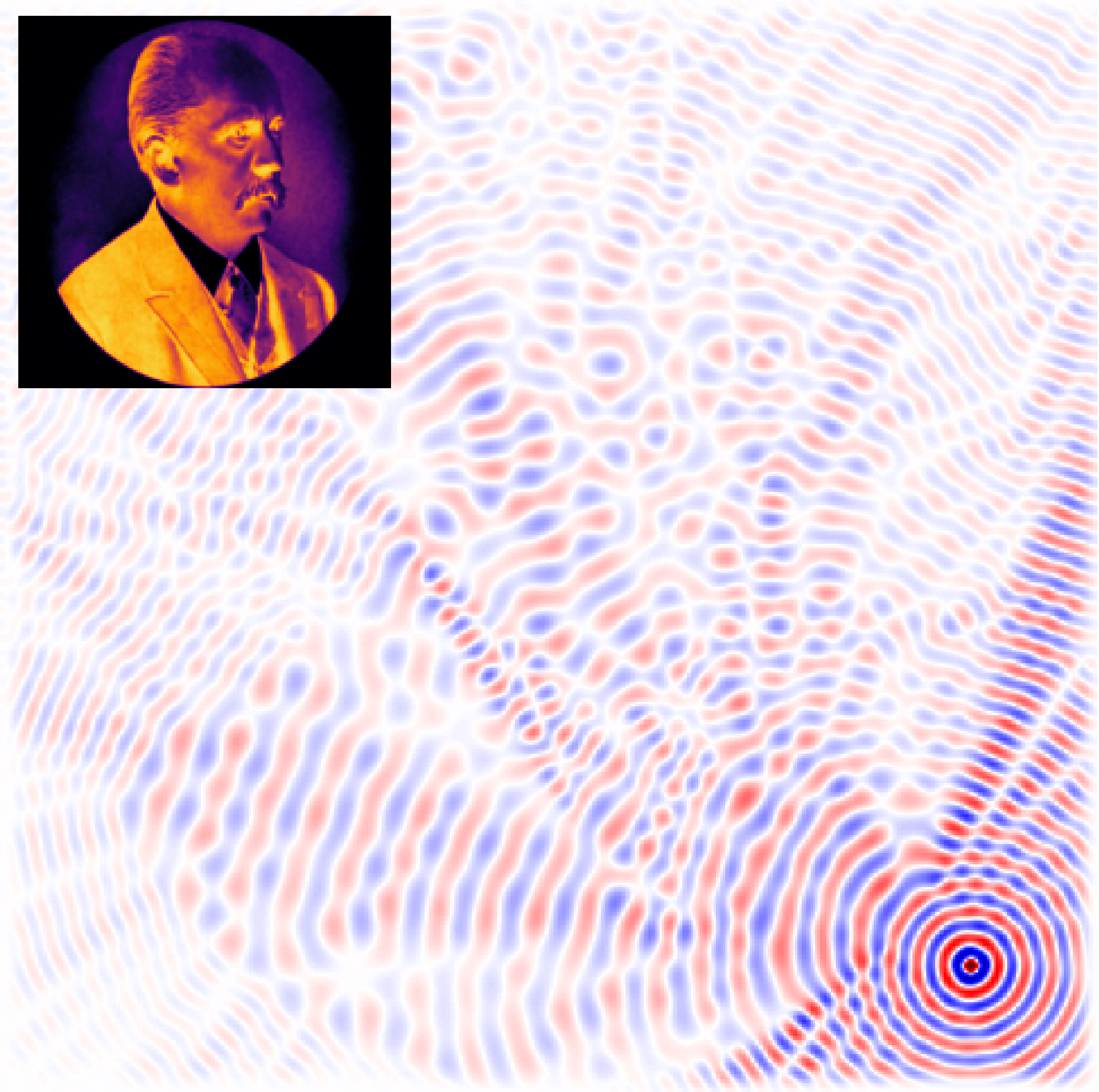
In either case, it is benefiical to have a **differentiable simulator** that can be dissected.

This is also used in several works on solving various PDEs, e.g. [13]



Conclusions

- Machine learning is increasingly being used to augment or learn acoustic simulators
- It is possible to learn a *lightweight* iterative solver for the heterogeneous Helmholtz equation
- More research is needed to understand the best training strategy and architecture
- End to end training can be done with a differentiable simulator



Conclusions

- Machine learning is increasingly being used to augment or learn acoustic simulators
- It is possible to learn a *lightweight* iterative solver for the heterogeneous Helmholtz equation
- More research is needed to understand the best training strategy and architecture
- End to end training can be done with a differentiable simulator

Thank you