U-FNO - an enhanced Fourier neural operator based-deep learning model for multiphase flow

Gege Wen¹, Zongyi Li², Kamyar Azizzadenesheli³, Anima Anandkumar², Sally M. Benson¹

¹Stanford University, ²California Institute of Technology, ³Purdue University
CO₂ storage problem: multiphase flow governed by highly nonlinear PDEs

**Gas saturation:** \( S_G(x, t) \)

**Pressure buildup:** \( dP(x, t) \)

**Mass conservation**
\[
\frac{\partial M^\kappa}{\partial t} = -\nabla \cdot \mathbf{F}^\kappa + q^\kappa \\
\kappa = \text{CO}_2 \text{ or water}
\]

**CO₂ component**
\[
\frac{\partial M^{\text{CO}_2}}{\partial t} = -\nabla \left[ \mathbf{F}^{\text{CO}_2} \bigg|_{\text{adv}} + \mathbf{F}^{\text{CO}_2} \bigg|_{\text{dif}} \right] + q^{\text{CO}_2}
\]

**Each phase governed by Darcy’s law**
\[
\frac{\partial \left( \phi \sum_p S_p \rho_p X_p^{\text{CO}_2} \right)}{\partial t} = \\
-\nabla \left[ \sum_p \left( -k_{r,p} \rho_p \left( \frac{\mu_p}{\rho_p} (\nabla P + P \mathbf{e}) - \rho_p \mathbf{g} \right) \right) + \mathbf{F}^{\text{CO}_2} \bigg|_{\text{dif}} \right] + q^{\text{CO}_2}
\]

\( p = \text{liq or gas} \)

---

2 Background → Model → Dataset → Results → Discussion

Stanford University
Numerical simulation is the essential tool for solving these PDE, however, very **time consuming**

**Key challenges**
- Highly nonlinear
- Multi-physics in the problems
- Multiscale heterogeneity
- Need for high grid resolution
- Inherent uncertainty in geology

\[ S(x, t), P(x, t) = f(x, t, k(x), q(x), Q, T, P_{init}, k_r, P_c, \ldots) \]

- \( x \): grid discretization
- \( t \): time
- \( Q \): injection rate
- \( q(x) \): injection location
- \( k(x) \): permeability map
- \( P_{init} \): initial reservoir pressure

---

**Background**

- Model
- Dataset
- Results
- Discussion
### Summary to existing approaches for solving the CO₂ storage subsurface flow problems

<table>
<thead>
<tr>
<th>Approach</th>
<th>Example</th>
<th>Method</th>
<th>Advantage</th>
<th>Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numerical solver</td>
<td>ECLIPSE</td>
<td>Solve the equation by discretizing the space</td>
<td>PDE-based</td>
<td>Expensive</td>
</tr>
</tbody>
</table>

---
### Summary to existing approaches for solving the CO$_2$ storage subsurface flow problems

<table>
<thead>
<tr>
<th>Approach</th>
<th>Example</th>
<th>Method</th>
<th>Advantage</th>
<th>Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numerical solver</td>
<td>ECLIPSE</td>
<td>Solve the equation by discretizing the space</td>
<td>PDE-based</td>
<td>Expensive</td>
</tr>
<tr>
<td>Neural-FEM</td>
<td>Physics-informed neural networks (Raissi et al, 2019; Fuks et al, 2020)</td>
<td>Formulate PDE/initial cond. in loss function</td>
<td>PDE-based</td>
<td>Expensive, convergence</td>
</tr>
</tbody>
</table>
Summary to existing approaches for solving the CO$_2$ storage subsurface flow problems

<table>
<thead>
<tr>
<th>Approach</th>
<th>Example</th>
<th>Method</th>
<th>Advantage</th>
<th>Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numerical solver</td>
<td>ECLIPSE</td>
<td>Solve the equation by discretizing the space</td>
<td>PDE-based</td>
<td>Expensive</td>
</tr>
<tr>
<td>Neural-FEM</td>
<td>Physics-informed neural networks (Raissi et al, 2019; Fuks et al, 2020)</td>
<td>Formulate PDE/initial cond. in loss function</td>
<td>PDE-based</td>
<td>Expensive, convergence</td>
</tr>
<tr>
<td>Data-driven CNN</td>
<td>CCSNet (Wen et al, 2021)</td>
<td>Learn empirical input-output mapping</td>
<td>Very fast prediction</td>
<td>Lots of data</td>
</tr>
</tbody>
</table>
## Summary to existing approaches for solving the CO₂ storage subsurface flow problems

<table>
<thead>
<tr>
<th>Approach</th>
<th>Example</th>
<th>Method</th>
<th>Advantage</th>
<th>Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numerical solver</td>
<td>ECLIPSE</td>
<td>Solve the equation by discretizing the space</td>
<td>PDE-based</td>
<td>Expensive</td>
</tr>
<tr>
<td>Neural-FEM</td>
<td>Physics-informed neural networks</td>
<td>Formulate PDE/initial cond. in loss function</td>
<td>PDE-based</td>
<td>Expensive, convergence</td>
</tr>
<tr>
<td></td>
<td>(Raissi et al, 2019; Fuks et al, 2020)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data-driven CNN</td>
<td>CCSNet</td>
<td>Learn empirical input-output mapping</td>
<td>Very fast prediction</td>
<td>Lots of data</td>
</tr>
<tr>
<td></td>
<td>(Wen et al, 2021)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Neural operator</td>
<td>Fourier Neural Operator (FNO)</td>
<td>Learn infinite-dimensional integral operator with NN</td>
<td>Very fast prediction, data efficient</td>
<td>High frequency information</td>
</tr>
<tr>
<td></td>
<td>(Li et al, 2021)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Summary to existing approaches for solving the CO\textsubscript{2} storage subsurface flow problems

<table>
<thead>
<tr>
<th>Approach</th>
<th>Example</th>
<th>Method</th>
<th>Advantage</th>
<th>Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numerical solver</td>
<td>ECLIPSE</td>
<td>Solve the equation by discretizing the space</td>
<td>PDE-based</td>
<td>Expensive</td>
</tr>
<tr>
<td>Neural-FEM</td>
<td>Physics-informed neural networks (Raissi et al, 2019; Fuks et al, 2020)</td>
<td>Formulate PDE/initial cond. in loss function</td>
<td>PDE-based, convergence</td>
<td></td>
</tr>
<tr>
<td>Data-driven CNN</td>
<td>CCSNet (Wen et al, 2021)</td>
<td>Learn empirical input-output mapping</td>
<td>Very fast prediction</td>
<td>Lots of data</td>
</tr>
<tr>
<td>Neural operator</td>
<td>Fourier Neural Operator (FNO) (Li et al, 2021)</td>
<td>Learn infinite-dimensional integral operator with NN</td>
<td>Very fast prediction, data efficient</td>
<td>High frequency information</td>
</tr>
</tbody>
</table>

This work: U-FNO
Let’s take a closer look at the model architecture of *original* Fourier Neural Operator (Li et al, 2021)
Closer look at the model architecture of the *original* Fourier Neural Operator (Li et al, 2021)
The transform is conducted utilizing Fast Fourier Transform (FFT).

After the transform, the discrete pixel data becomes \textit{continuous function}.

\[
(\mathcal{F}f)_j(k) = \int_D f_j(x)e^{-2i\pi \langle x,k \rangle} \, dx
\]
Closer look at the model architecture of the *original* Fourier Neural Operator (Li et al, 2021)

\[(\mathcal{F}f)_j(k) = \int_D f_j(x)e^{-2\pi \langle x, k \rangle} \, dx\]

\[(\mathcal{K}(\phi)v_t)(x) = \mathcal{F}^{-1}(R_\phi \cdot (\mathcal{F}v_t))(x)\]
Closer look at the model architecture of the original Fourier Neural Operator (Li et al, 2021)

\[
(\mathcal{F} f)_j(k) = \int_D f_j(x) e^{-2\pi \langle x, k \rangle} dx
\]

\[
(\mathcal{F}^{-1} f)_j(x) = \int_D f_j(k) e^{2\pi \langle x, k \rangle} dk
\]

\[
(\mathcal{K}(\phi) v_t)(x) = \mathcal{F}^{-1} \left( R_{\phi} \cdot (\mathcal{F} v_t) \right)(x)
\]
**U-FNO Architecture:** enhanced Fourier Neural Operator for solving the CO$_2$-water multiphase flow problem

Remark 1: 2-step mini U-Net to enhance higher frequency information
**U-FNO Architecture**: enhanced Fourier Neural Operator for solving the CO₂-water multiphase flow problem

**Remark 1:**
2-step mini U-Net to enhance higher frequency information

**Remark 2:** $L$ Fourier layers followed by $M$ U-Fourier layers, $L$ and $M$ are hyper-parameters
Anisotropic permeability and heterogeneous porosity data generation recipe

Step 1. Start with radial permeability map

Step 3. assign each material to a random anisotropy ratio (1-150)

Step 2. sample the random # of anisotropic materials (1-6) and divide the radial perm distribution to # of materials

Step 4. calculate vertical perm using anisotropy ratio

Background ➔ Model ➔ Dataset ➔ Results ➔ Discussion

Stanford University
Create porosity field based on experimental relationships

Step 1. Calculate permeability average

Step 2. Solve for porosity using permeability average

Step 3. Add a random noise $\varepsilon$ with std=0.005

(Pape et al, 2000)
Create porosity field based on experimental relationships

\[ k = 25 \text{D}, \phi = 32\% \]

\[ k = 1e^{-4} \text{mD}, \phi = 0.3\% \]

(Pape et al, 2000)

\[ \text{Cap } \phi_{\text{max}} = 32\% \]

\[ \text{Cap } \phi_{\text{min}} = 0.3\% \]

Background → Model → Dataset → Results → Discussion
Adding porosity and anisotropy increase the input dimension for 38,000 times

CCSNet dataset

New dataset
Result: CO$_2$ gas saturation plume prediction greatly improved with U-FNO comparing to CNN.
Result: U-FNO produces consistently superior CO$_2$ gas saturation plume prediction throughout the injection period.
**Result:** Pressure buildup prediction greatly exceeded CNN; low pressure cases significantly improved.
**Result:** U-FNO produces consistently superior pressure buildup prediction throughout the injection period.
Result: U-FNO has superior performance on both training and testing sets among CNN and original FNO

<table>
<thead>
<tr>
<th>Model</th>
<th>Gas saturation ($SG$) R2</th>
<th>Pressure buildup ($dP$) R2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Train</td>
<td>Test</td>
</tr>
<tr>
<td></td>
<td>mean</td>
<td>std</td>
</tr>
<tr>
<td>CNN</td>
<td>0.994</td>
<td>0.007</td>
</tr>
<tr>
<td>FNO</td>
<td>0.990</td>
<td>0.013</td>
</tr>
<tr>
<td>Conv-FNO</td>
<td>0.993</td>
<td>0.011</td>
</tr>
<tr>
<td>U-FNO</td>
<td><strong>0.997</strong></td>
<td><strong>0.005</strong></td>
</tr>
</tbody>
</table>
Remark: great data efficiency – more complex physics, less data required

U-FNO only requires 4500 data samples for training
CCSNet requires 20000 data samples to achieve similar performance
Computational efficiency: prediction speed up is 60000x vs. numerical simulation; even faster than CNN

Table 3: Summary of the number of parameters, training time, and testing times required for all four models. The testing times are calculated by taking the average of 500 random cases. The speed-up is compared with average numerical simulation run time of 10 mins.

<table>
<thead>
<tr>
<th>Model</th>
<th># Parameter ((-))</th>
<th>Training ((s/\text{epoch}))</th>
<th>Testing</th>
<th>Speed-up vs. numerical simulation ((\text{times}))</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN</td>
<td>33,316,481</td>
<td>562</td>
<td>Gas saturation ((s)): 0.050</td>
<td>1×10^4</td>
</tr>
<tr>
<td>FNO</td>
<td>31,117,541</td>
<td>711</td>
<td>Pressure buildup ((s)): 0.050</td>
<td>1×10^9</td>
</tr>
<tr>
<td>Conv-FNO</td>
<td>31,222,625</td>
<td>1,135</td>
<td></td>
<td>1×10^5</td>
</tr>
<tr>
<td>U-FNO</td>
<td>33,097,829</td>
<td>1,872</td>
<td></td>
<td>6×10^4</td>
</tr>
</tbody>
</table>
U-FNO - an enhanced Fourier neural operator based-deep learning model for multiphase flow

Background → Model → Dataset → Results → Discussion
Acknowledgement

G. Wen and S. M. Benson gratefully acknowledges the supported by ExxonMobil through the Strategic Energy Alliance at Stanford University and the Stanford Center for Carbon Storage.

Z. Li gratefully acknowledges the financial support from the Kortschak Scholars Program.

A. Anandkumar is supported in part by Bren endowed chair, LwLL grants, Beyond Limits, Raytheon, Microsoft, Google, Adobe faculty fellowships, and DE Logi grant.