

Supporting Information for “AI models of unstable flow exhibit hallucination”

Governing equations and DNS

We consider miscible viscous fingering in a two-dimensional permeable domain of length L_x , width L_y , uniform porosity ϕ , and isotropic permeability κ . The medium is initially saturated with a more viscous fluid of viscosity μ_0 , while a less viscous fluid of viscosity μ_1 is injected from the inlet boundary. The outlet is treated as a free-flow boundary or a prescribed pressure boundary. The fluids are assumed to be fully miscible at all concentrations, without chemical reactions or thermal effects, and gravitational forces are neglected. Following [4], the system is modeled using the theory of single-phase flow through porous media. The continuity equation and the advection–dispersion equation (ADE) govern the transport of concentration, while Darcy’s law relates velocity and pressure:

$$\frac{\partial c}{\partial t} + \nabla \cdot \left(\mathbf{u}c - \frac{1}{\text{Pe}} \nabla c \right) = 0, \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0, \quad (2)$$

$$\mathbf{u} = -\frac{1}{\mu(c)} \nabla p, \quad (3)$$

$$\mu(c) = e^{-Rc}, \quad (4)$$

$$R = \log M, \quad M = \frac{\mu_0}{\mu_1}, \quad (5)$$

where $0 \leq c \leq 1$ is the normalized concentration field of the injected fluid, and $\mathbf{u} = [u_x, u_y]$ is the dimensionless velocity field. The Péclet number is defined as

$$\text{Pe} = \frac{VL_y}{D}, \quad D = \phi D_m + \alpha_d V, \quad (6)$$

Note that the effective dispersion coefficient D above includes both molecular diffusion D_m and pore scale dispersion α_d [7]; hence the governing transport equation is the advection–dispersion equation. We note that the effective dispersion coefficient D used in this study is a scalar representation of hydrodynamic dispersion. A fully tensorial representation of hydrodynamic dispersion is physically appropriate with multi-directional species transport and should be considered in future DNS. We also note that setting the hydrodynamic dispersion parameter to the molecular diffusion coefficient (suitable for bulk mixing of

the two fluids) would correspond to an open, non-porous domain. This regime is outside the scope of the present study, which is built around DNS data generated for porous media viscous fingering, but we acknowledge this as a valuable direction for future work. The viscosity law (4) is exponential in concentration, consistent with dilute, chemically inert tracers in groundwater [3, 5].

The system is governed by two dimensionless groups: the viscosity ratio M and the Péclet number Pe . Boundary conditions are: constant injection with $c_{\text{in}} = 1$, natural outflow at the outlet, and no-flux with zero normal concentration gradient at the transverse boundaries. The transversely averaged outlet concentration $\bar{c}_{\text{out}}$ defines the breakthrough curve.

Direct Numerical Simulation (DNS) becomes challenging at high viscosity ratios (e.g., $M > 50$) and large Péclet numbers (e.g., $Pe > 500$), due to steep concentration gradients, high fingertip velocities and vorticity [5] and a rapid growth of discretization-related errors [8]. To address these issues, we adopt a sequential solution scheme for DNS, where the pressure equation is solved using a second-order finite volume method with two-point flux approximation and the ADE is discretized using a sixth-order compact finite difference scheme [5].

Data generation details

DNS was carried out on a finite-volume grid of size $(n_x, n_y) = (512, 256)$ over 600 time steps. This temporal resolution is sufficient to capture the evolution of viscous fingers prior to breakthrough, during breakthrough, and in the post-breakthrough regime.

The initial condition is constructed by first defining a two-dimensional spatial grid, then introducing small random perturbations through shifted cosine waves. The complementary error function is subsequently applied to generate a sharp concentration front, resulting in a heterogeneous and physically realistic initial field for diffusion and mixing simulations.

Each DNS realization produces data of shape $(600, 3, 256, 512)$, corresponding to 600 time steps and three physical states: horizontal velocity u_x , vertical velocity u_y , and concentration field c , as illustrated in Fig. S1. To prepare the data for deep learning models, the simulations were temporally downsampled by retaining every fourth time step, yielding 150 frames per case. Spatial fields were further resized to $(n'_x, n'_y) = (128, 64)$ using bilinear interpolation. For training purposes, only the concentration field c was utilized.

In total, 30 simulations with distinct random initial conditions were generated for viscosity ratios $M \in [11.5, 55.0]$, sampled at uniform increments of 1.5. The dataset was evenly split into 15 training and 15 testing cases. Specifically, M at intervals of three, beginning with $M = 11.5$, were assigned to the training set, while the remaining values were reserved for testing.

Intrinsic stochasticity of fingering

Viscous fingering is governed by an inherently unstable set of equations and is highly sensitive to small perturbations, e.g., in rock permeability, and numerical

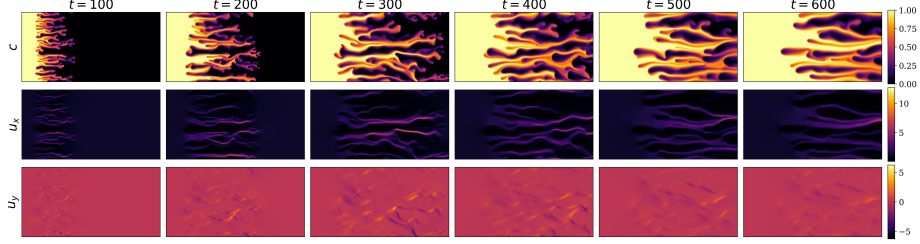


Figure S1: Snapshots of the DNS for $M = 49$ with shape $(600, 3, 256, 512)$, corresponding to 600 time steps and three physical states: horizontal velocity u_x , vertical velocity u_y , and concentration field c . The size of grid discretizations size $(n_y, n_x) = (256, 512)$.

errors, e.g., discretization errors. This sensitivity becomes more pronounced as the M increases, due to the exponential viscosity relation in (4). This is illustrated by Fig. S2 where the same initial conditions lead to different simulation results. Despite identical inputs, the simulations diverge and produce noticeably different outcomes. This demonstrates that the DNS solution exhibits intrinsic variability at high M , even in the absence of stochastic forcing. As discussed in Fig. 6 of the manuscript, our proposed AI framework, DeepFingers, correctly captures the propagation of such uncertainty into fluid mixing and spreading metrics.

Data preparation

The dataset comprises normalized concentration fields $c(x, y)$ defined over a two-dimensional (2D) domain of size $(n_x \times n_y)$, generated via DNS across a range of viscosity ratios M and $Pe = 3000$. The DeepFingers model uses the concentration field c at time t to serve as the input for predicting the c field at the subsequent time step $t + 1$. The branch network receives inputs with shape $(B, 1, 64, 128)$, where B denotes the batch size. The trunk network is provided with two scalar inputs: the viscosity ratio M and the time at $t + 1$, which are concatenated into a tensor of shape $(B, 2)$. The model output corresponds to the predicted c field at time $t + 1$, also formatted as $(B, 1, 64, 128)$.

Model training and inference

DeepFingers is trained for 10^5 iterations using an L_1 loss and the Adam optimizer (initial learning rate 10^{-3} with stepwise decay every 5000 iterations), with a batch size of 32. The model is evaluated on test cases with viscosity ratios M not used during training.

Training the model over the full range of M values requires 1.5 hours on a single Nvidia A100 GPU. DNS has no training phase and solves each case directly. For inference, DNS requires 6071 seconds for $M = 13$ and 19264

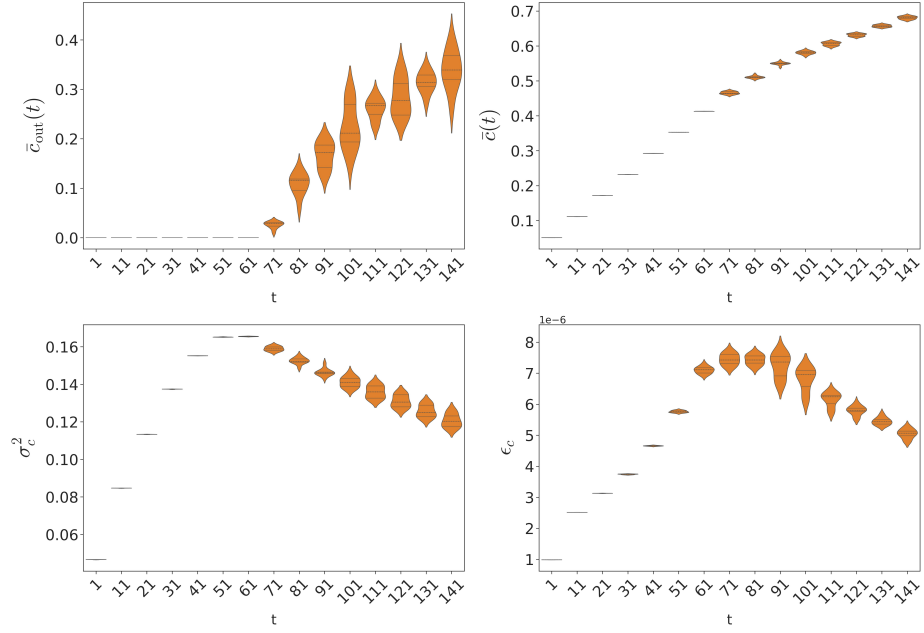


Figure S2: Distribution of four physics-based metrics for $M = 55$ obtained by running DNS ten times with identical initial conditions. The variability across runs highlights the inherent instability of the VF system at high M . Because pixel-wise comparison becomes unreliable under such sensitivity, we focus instead on physics-informed metrics and spectral analysis when comparing DNS with surrogate models.

Table S1: Training configuration.

Setting	Value
Optimiser	Adam
Initial LR	10^{-3}
LR schedule	$\times 0.9$ every 5000 iterations
Loss	L_1
Iterations	100,000
Batch size	32
Latent width	64
Fourier modes	20×20
Spectral blocks	4
Merge	multiplicative
L_2 regularisation	3×10^{-6}
Backend	PyTorch (DeepXDE)

Table S2: DeepFingers training and inference time for all time steps. Hardware: 16 CPU cores, 128 GB RAM, NVIDIA A100 GPU.

Method	Training hours	Inference (seconds)	
		$M = 13$	$M = 55$
DNS	N/A	6071	19264
DeepFingers	1.5	2.13	2.22

seconds for $M = 55$, reflecting the increased numerical effort at higher M . DeepFingers produces all time-step predictions in approximately 2 seconds for both cases, showing no meaningful dependence on M and providing a speed-up of more than three orders of magnitude.

Model evaluation: metrics and significance

To quantify the mixing and spreading associated with fingering instabilities, we evaluate four domain-wide indicators: the mean or domain-averaged concentration, the concentration variance, the mean scalar dissipation rate, and the outlet concentration.

Average concentration field

In viscous fingering, the displacement of a more viscous fluid by a less viscous fluid leads to the development of complex patterns in the $c(x, y, t)$ field. Although the c map reveals the local morphology of the fingers, a more global and physically meaningful metric of mixing can be obtained by examining the domain-averaged concentration field, $\bar{c}(t)$. In the context of oil recovery and other subsurface engineering applications, the average concentration is closely related to the sweep efficiency, a key measure of how effectively the injected

fluid contacts and displaces the in-place fluid. This scalar quantity condenses the complex spatio-temporal evolution of the c field due to finger splitting, merging, and channeling into a single measure, allowing for the quantification of the overall progression of fluid displacement and mixing. This enables a direct comparison across different M and facilitates the evaluation of AI model performance in terms of the sweep efficiency relative to DNS.

Concentration variance

Denoted by σ_c^2 , it quantifies the degree of segregation or un-mixing between the fluids and reflects the spatial spread of the concentration field [5]. High σ_c^2 implies low degree of mixing [4].

$$\sigma_c^2(t) = \overline{c^2} - \bar{c}^2, \quad (7)$$

where the overbar symbol denotes the spatial average over the domain. Large values of σ_c^2 indicate strong concentration contrasts and limited mixing, while small values correspond to a more homogeneous fluid distribution. Owing to its role in measuring concentration fluctuations, this metric is often referred to as the scalar energy, analogous to kinetic energy in the case of momentum conservation via Navier-Stokes equation.

Mean scalar dissipation rate

Denoted by ϵ_c , it characterizes the rate of interface stretching and mixing in the system [5].

$$\epsilon_c = \frac{1}{\text{Pe}} \overline{|\nabla c|^2}. \quad (8)$$

It captures the combined effects of advective stretching, which enhances concentration gradients, and mass diffusion, which acts to smooth them. As fingering structures develop, the interface is stretched, amplifying gradients and increasing ϵ_c , while diffusion gradually reduces these gradients.

The mean scalar dissipation rate ϵ_c is a measure of the rate of mixing; high ϵ_c means high mixing rate. At early times, before breakthrough, $c_{\text{out}} = 0$, the concentration variance σ_c^2 initially rises due to the injection of the less viscous fluid and the formation of fingering structures. As the fingers develop and diffusion begins to homogenize the concentration field, σ_c^2 decreases, resulting in a non-monotonic temporal trend. Fig. S5 illustrates this behavior for both DNS and DL models. The mean scalar dissipation rate, ϵ_c , exhibits a similar non-monotonic evolution in Fig. S6. It increases during early-time finger formation as advective stretching amplifies concentration gradients and decreases at later times as diffusion smooths the interfaces. Together, σ_c^2 and ϵ_c provide complementary perspectives on the mixing and spreading dynamics of fingering, capturing both the growth of fingers and the subsequent smoothing induced by diffusion.

Breakthrough concentration

Also known as the outlet-averaged concentration, $\bar{c}_{\text{out}}(t)$ tracks breakthrough dynamics at the outflow boundary of the domain. It is essential to examine $\bar{c}_{\text{out}}$, which often holds a special meaning in real-world applications [1, 2, 6]. At the reservoir scale, the outlet boundary can signify a production well and, at the laboratory scale, the outlet boundary can signify a rock sample’s outlet connected to an effluent collector. The temporal evolution of $\bar{c}_{\text{out}}(t)$ captures the effluent concentration profile. The breakthrough time defined as the moment when $\bar{c}_{\text{out}}(t)$ exceeds a prescribed threshold. This provides a direct measure of the timing and extent of fluid displacement.

For a fluid displacement process, such as in viscous fingering, the onset of nonzero concentration at the outlet marks the breakthrough time, the moment in time when the injected fluid first reaches the production end of the domain. During viscous fingering, this happens as fingers arrive at the right boundary of the domain, which is most clearly shown by column $t = 75$ of Fig. 2 of the manuscript. Finger arrival at the boundary is an episodic process, i.e., fingers arrive at different times and $\bar{c}_{\text{out}}(t)$ can jump with each arrival. This gives rise to a non-monotonic trend in $\bar{c}_{\text{out}}(t)$ in contrast to the monotonically increasing trend of $\bar{c}(t)$ in the previous section.

The breakthrough time is a critical parameter for reservoir management, particularly in fluid injection-driven subsurface operations. In these processes, an earlier breakthrough generally indicates inefficient sweep and the formation of high-conductivity flow channels as opposed to a uniform, piston-like displacement. Monitoring the evolution of the outlet concentration also enables the assessment of post-breakthrough displacement performance, such as the rate of increase in the concentration of the injected fluid, which reflects the severity of channeling and delay in achieving full sweep.

By comparing the predicted $\bar{c}_{\text{out}}(t)$ curves with DNS results for different viscosity ratios M , we can evaluate the AI model’s ability to reproduce both the breakthrough time and the subsequent concentration trends. Accurate prediction of these features is particularly valuable for optimizing flooding strategies, estimating recovery factors, and designing operational controls in reservoir management.

Fig. S4 shows that the DeepFingers model closely approximates the breakthrough curve across the range of M values, performing more reliably than the other DL models with hallucination issues.

The four metrics described above capture complementary aspects of interface dynamics and mixing efficiency during fluid displacement. Together, they provide a comprehensive description of mixing, spreading, and breakthrough behavior during the evolution of fingering.

DNS software

The DNS solver is implemented in MATLAB. The complete set of routines is summarized in Table S3. No third-party MATLAB toolboxes are required

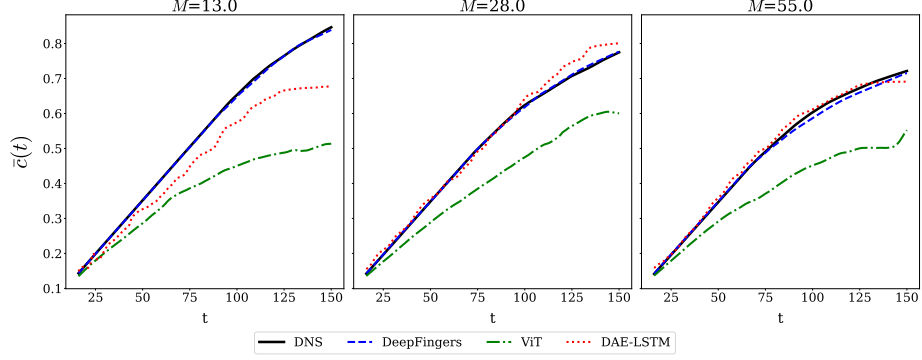


Figure S3: Comparison of the average concentration $\bar{c}(t)$ over time for DNS, DeepFingers, ViT, and DAE-LSTM at $M = 13, 28$, and 55 (left to right). The average concentration quantifies sweep efficiency, representing the fraction of the domain displaced or swept by the injected fluid. Among the models, DeepFingers most closely matches DNS, accurately capturing the key displacement dynamics across all mobility ratios.

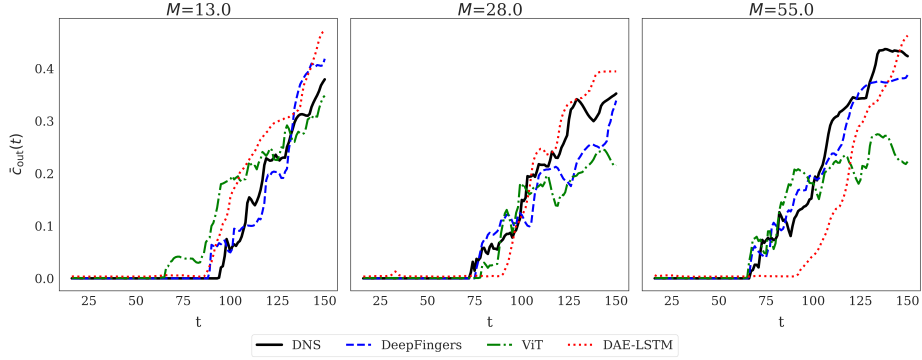


Figure S4: Comparison of the outlet-averaged concentration $\bar{c}_{out}(t)$ over time for DNS, DeepFingers, ViT, and DAE-LSTM, across mobility ratios $M = 13.0, 28.0$, and 55.0 (left to right). The outlet concentration provides critical insight into breakthrough time, a key indicator of sweep efficiency in displacement processes. Among the models, DeepFingers most accurately captures both the onset of breakthrough and the subsequent evolution of the produced concentration across M values. As M increases, the flow becomes increasingly unstable, broadening the range of possible solution behaviors and making accurate modeling more challenging.

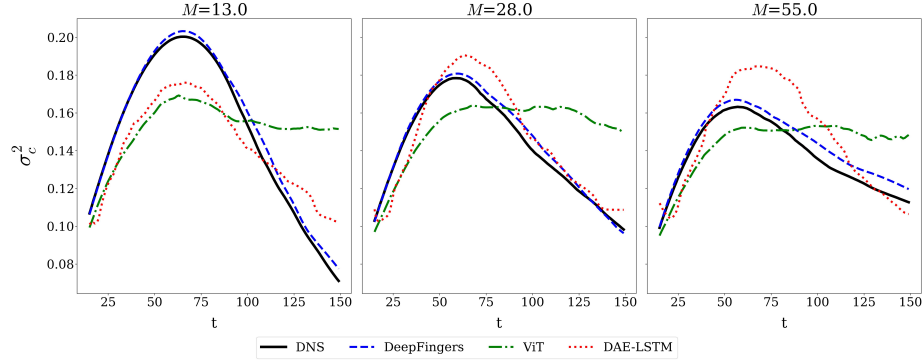


Figure S5: Comparison of the concentration variance σ_c^2 (a measure of the degree of mixing) between DNS, DeepFingers, ViT, and DAE-LSTM. σ_c^2 increases at early times as the less viscous fluid enters the domain and starts to spread due to fingering. As the less viscous fluid continues to occupy more space and the mean concentration increases, σ_c^2 starts to decrease.

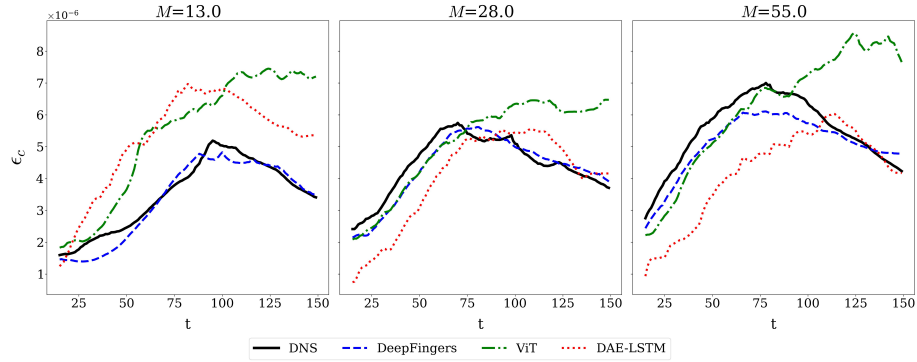


Figure S6: Comparison of the mean scalar dissipation rate ϵ_c (a measure of interface stretching during VF) between DNS, DeepFingers, ViT, and DAE-LSTM.

Table S3: MATLAB routines of the DNS solver.

Routine	Function
<code>miscible2D.m</code>	Main driver: grid/parameter setup, initial condition, time loop.
<code>solve_press.m</code>	Assembles and solves the linear system for the cell-centered pressure vector, returns Darcy velocities based on two-point flux approximation.
<code>residual_c.m</code>	Evaluates the advection–dispersion residual.
<code>MatComp.m</code>	Builds sixth-order compact finite-difference matrices.
<code>der.m</code>	Applies the compact-FD operator via the Thomas algorithm.

beyond base MATLAB: the solver relies on FFT (`fft/ifft`), sparse matrix construction (`spdiags`), and the sparse direct solver (MATLAB’s backslash operator).

DeepFingers packages

DeepFingers follows the standard DeepONet formulation, in which the solution operator is represented through two subnetworks: a branch network that encodes the input field and a trunk network that encodes the physical parameters. On top of this classical branch–trunk structure, DeepFingers incorporates a U-FNO decoder that maps the merged latent representation to the predicted concentration field. The model uses the Python packages in Table S4.

Table S4: Python packages used by the DeepFingers surrogate.

Package	Role
<code>DeepXDE</code>	Operator-learning framework providing model wrapper, training loop, batching utilities, callbacks, and using PyTorch backend.
<code>U-FNO</code>	U-FNO spectral–convolution decoder implementation from https://github.com/gegewen/ufno .
<code>torch</code>	Core tensor library for autodiff, neural-network layers, FFT operations, and optimisation.
<code>numpy</code>	Array manipulation, numerical utilities, and metric computation.
<code>h5py</code>	Reading and writing HDF5 datasets used for training and testing.

Model framework comparison

Table S5, which provides a high-level comparison between DeepFingers and the two baseline model families. This table highlights the core architectural distinctions framework type, parameter conditioning, latent representation, temporal

treatment, and decoder design—that differentiate DeepFingers from existing approaches. These differences clarify how DeepFingers enables improved multi-scale fidelity and parameter-aware generalization.

References

- [1] M Alasker, Z Liu, F Ghazvini, FPJ de Barros, and B Jha. Scalar transport during miscible viscous fingering experiments in consolidated porous media. *Physics of Fluids*, 37(11), 2025.
- [2] A. Bonazzi, B. Jha, and F.P.J. de Barros. Influence of initial plume shape on miscible porous media flows under density and viscosity contrasts. *Journal of Fluid Mechanics*, 972:A19, 2023.
- [3] G.M. Homsy. Viscous fingering in porous media. *Annual Review of Fluid Mechanics*, 19(1):271–311, 1987.
- [4] B. Jha, L. Cueto-Felgueroso, and R. Juanes. Fluid mixing from viscous fingering. *Physical review letters*, 106(19):194502, 2011.
- [5] B. Jha, L. Cueto-Felgueroso, and R. Juanes. Quantifying mixing in viscously unstable porous media flows. *Physical Review E*, 84(6):066312, 2011.
- [6] C. Nicolaides, B. Jha, L. Cueto-Felgueroso, and R. Juanes. Impact of viscous fingering and permeability heterogeneity on fluid mixing in porous media. *Water Resources Research*, 51(4):2634–2647, 2015.
- [7] Adrian E Scheidegger. Statistical hydrodynamics in porous media. *Journal of Applied Physics*, 25(8):994–1001, 1954.
- [8] C.T. Tan and G.M. Homsy. Viscous fingering with permeability heterogeneity. *Physics of Fluids A: Fluid Dynamics*, 4(6):1099–1101, 1992.

Table S5: High-level comparison of DeepFingers and baseline models.

Aspect	DeepFingers	DAE-LSTM	ViT
Framework type	Single-stage operator network: branch + trunk with U-FNO decoder.	Two-stage pipeline: CNN encoder $\rightarrow$ compact latent $\rightarrow$ LSTM for temporal dynamics.	Single-stage transformer predicting next frame from patch embeddings.
Training stages	End-to-end operator learning (single stage).	Autoencoder training followed by LSTM temporal training.	One-stage supervised next-step prediction.
How M and time are used	M and target time are explicit trunk inputs (parameter conditioning).	M appended as an extra input channel; M is also used in parameterization of Koopman operator; time implicit via step index.	M provided as an input channel; time implicit via step index.
Latent representation	Spatially resolved spectral latent.	Compact low-dimensional bottleneck latent.	Patch/embedding tokens.
Temporal modeling	Time explicitly conditioned via trunk.	LSTM evolves latent state; assumes uniform step frequency.	Direct next-step prediction; assumes uniform step frequency.
Decoder / reconstruction	U-FNO decoder (spectral convolution with U-Net-style multiscale skips).	CNN decoder reconstructs from compact latent.	Transformer head / patch decoder.
Primary design goal	Preserve multiscale spectral fidelity; explicit parameter conditioning.	Efficient latent temporal modeling and compact representation.	High-capacity spatial attention for image prediction.
Model parameter count	19,331,601	DAE : 441,577 LSTM : 1,545,728	17,726,595