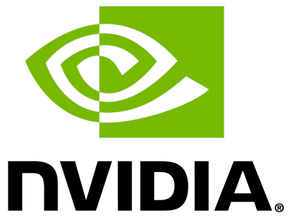
# NEURAL OPERATORS: ACCELERATING SCIENTIFIC SIMULATIONS BY LEARNING ON FUNCTION SPACES

Anima Anandkumar

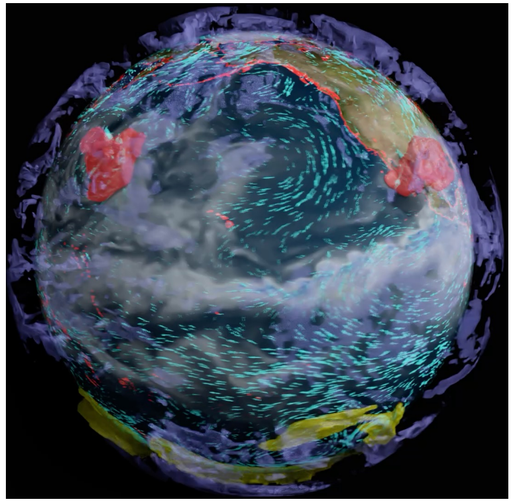
Bren Professor, Caltech CMS

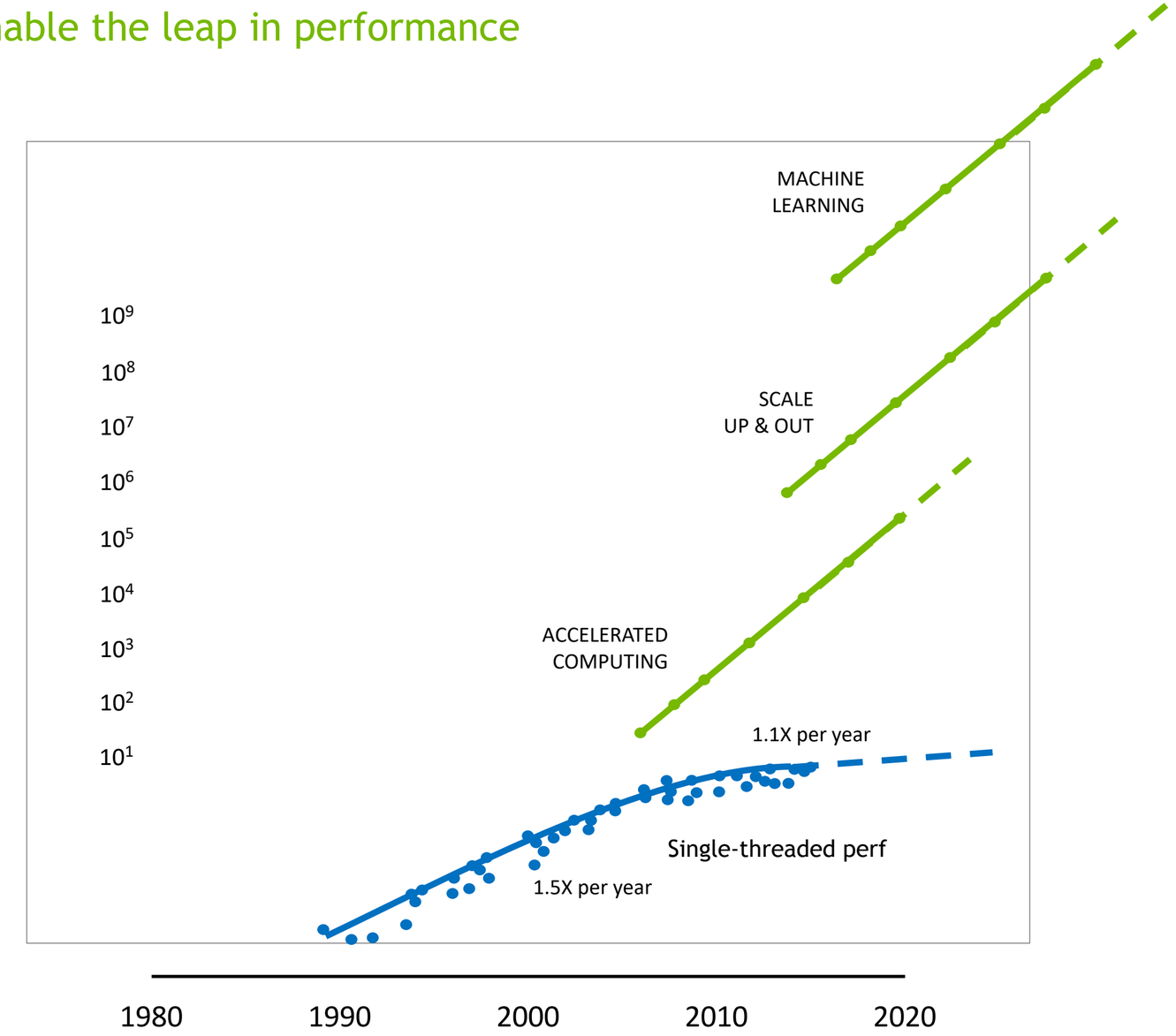
Senior Director of AI Research, NVIDIA




# MILLION-X LEAP IN SCIENTIFIC COMPUTING

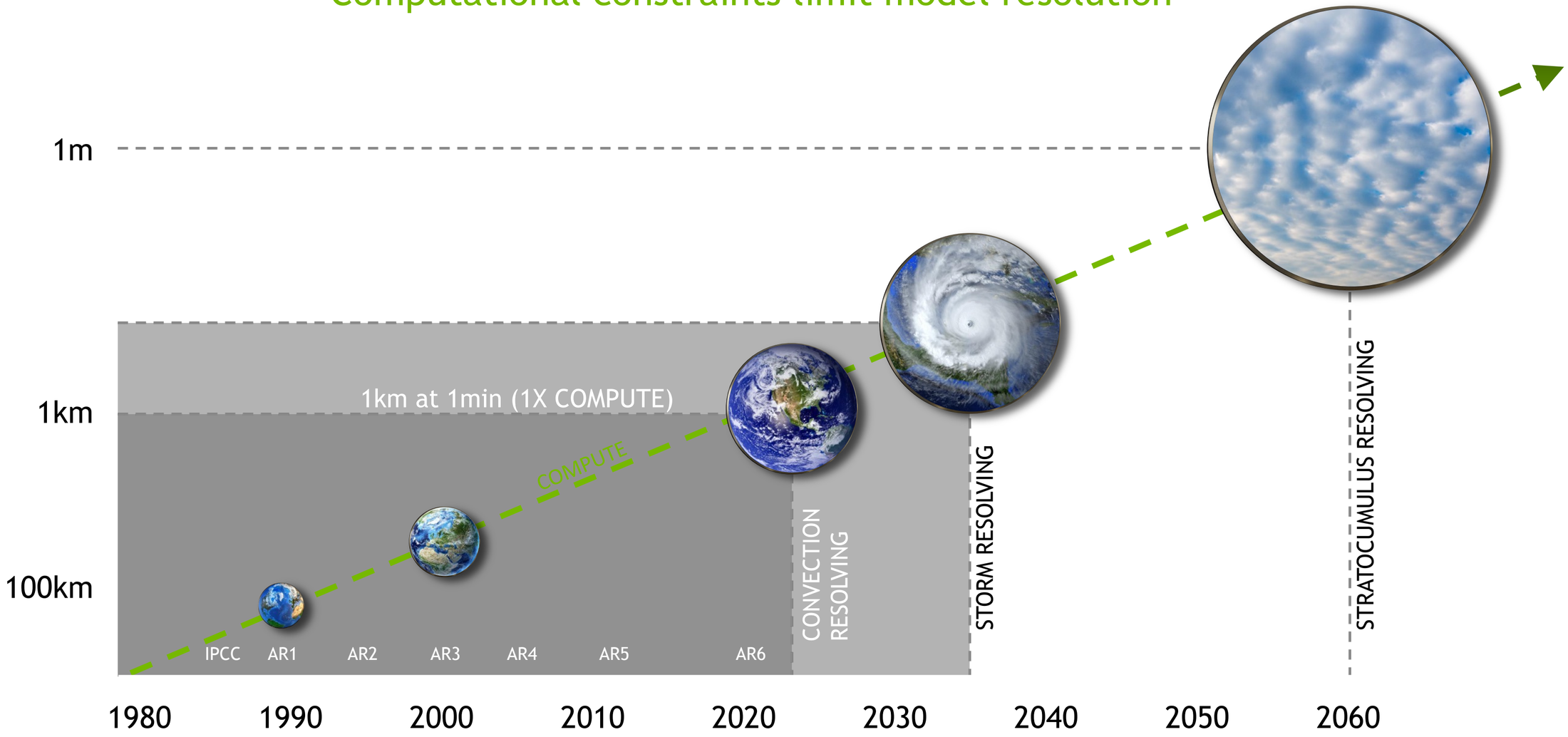
## AI/ML to enable the leap in performance





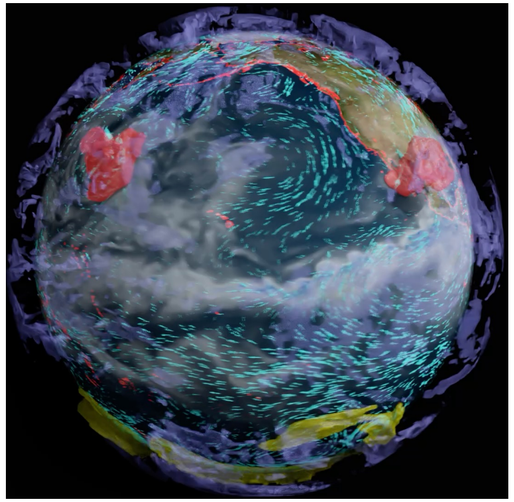
# CLIMATE MODELING REQUIRES MILLION-X SPEEDUPS

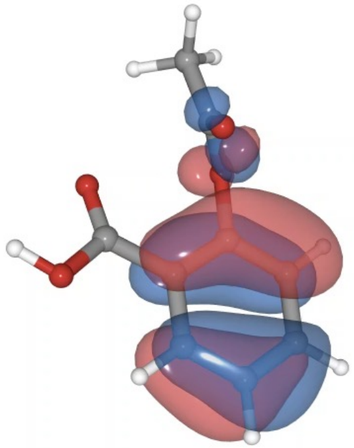
## Computational constraints limit model resolution

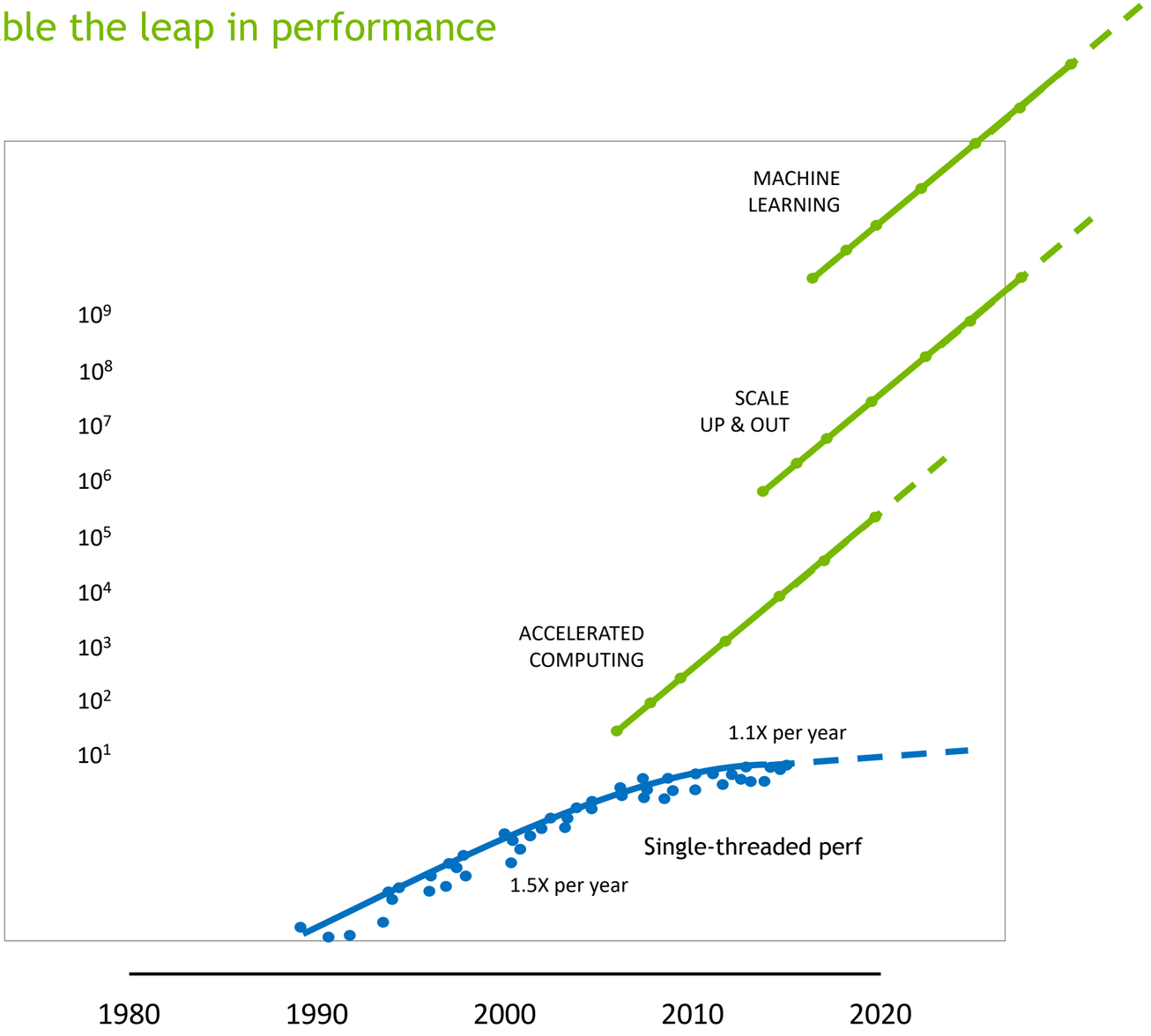


# MILLION-X LEAP IN SCIENTIFIC COMPUTING

## AI/ML to enable the leap in performance







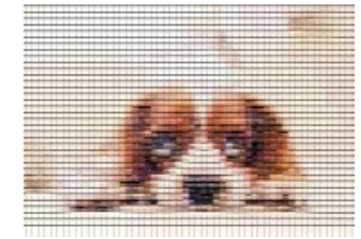
# FOURIER NEURAL OPERATOR

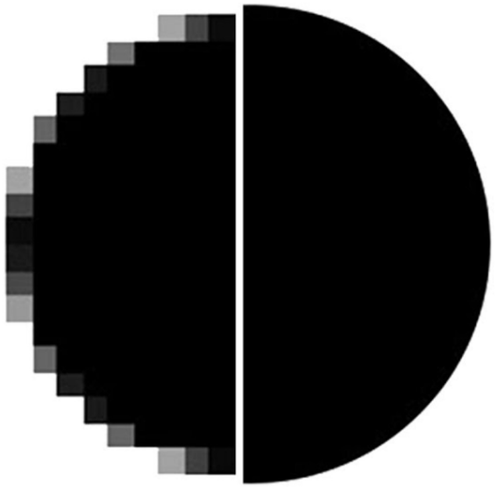
# DISCRETIZATION-INVARIANT LEARNING

## One AI model for any discretization: no re-training

### Neural Network

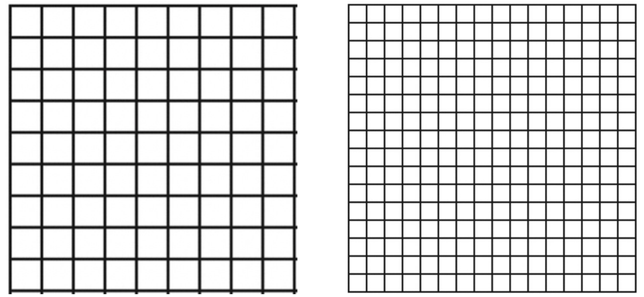
Input and output at fixed resolution

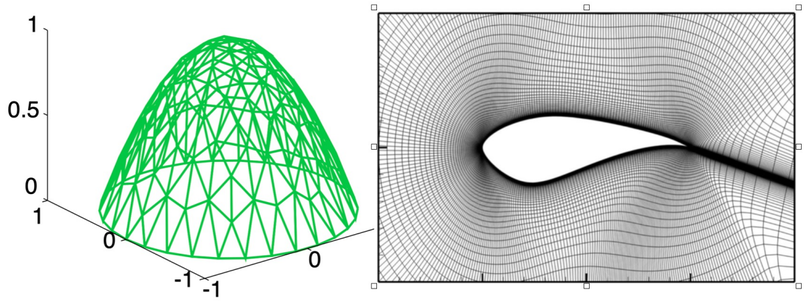




### Neural Operator

Input and output at any points in domain



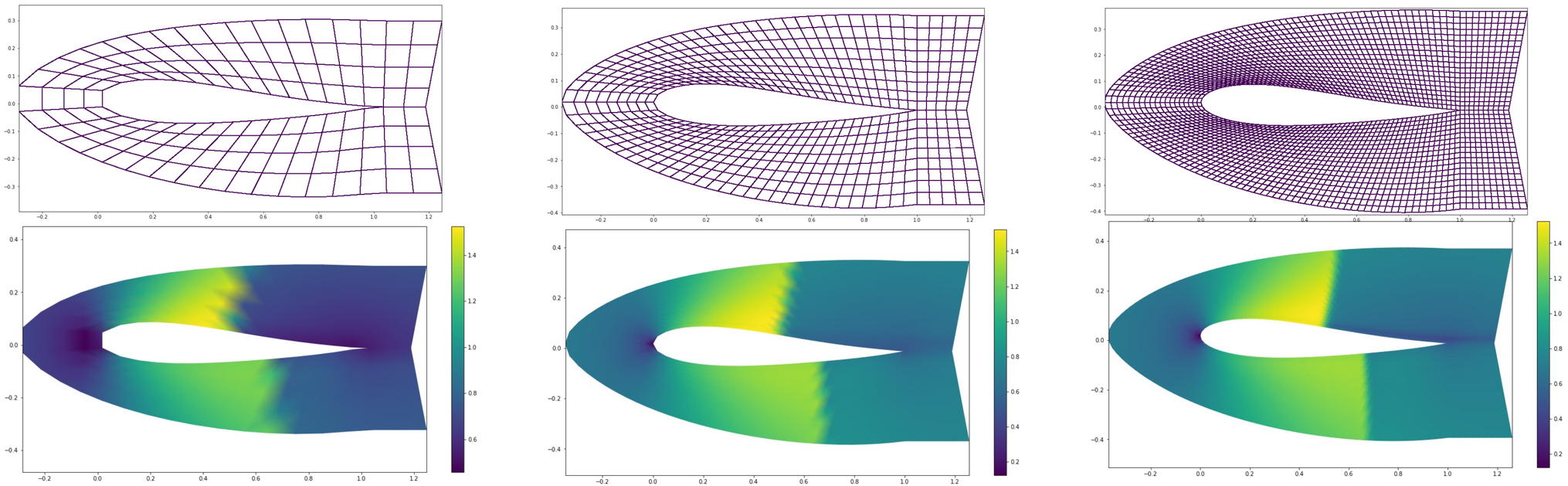


# DISCRETIZATION-INVARIANCE OF NEURAL OPERATOR

Definition: a trained AI model is discretization-invariant if

- We can query at any point.
- Converges upon mesh refinement to a limit.

Mesh refinement

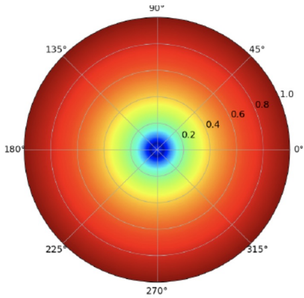


Converging solution

# INTEGRAL OPERATOR FOR SOLVING LINEAR PDE

Input → Integral Linear Operator → Output

$$\int \kappa(x, y)\, v(y)\, dy$$

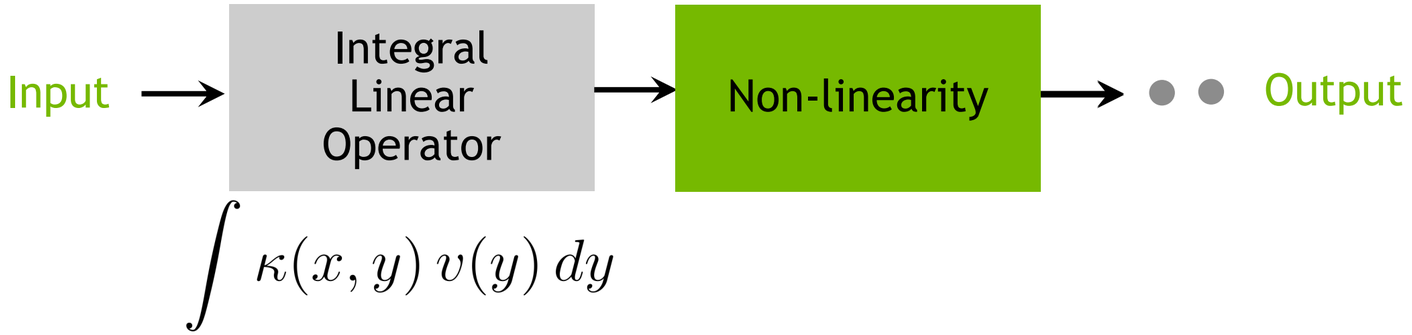



$K(x, y)$

Kernel of integral operator
For heat diffusion

- Integral operator outputs functions (not just finite-dimensional vectors).
- Integral operator is discretization invariant.

# NEURAL OPERATOR: A GENERAL FRAMEWORK



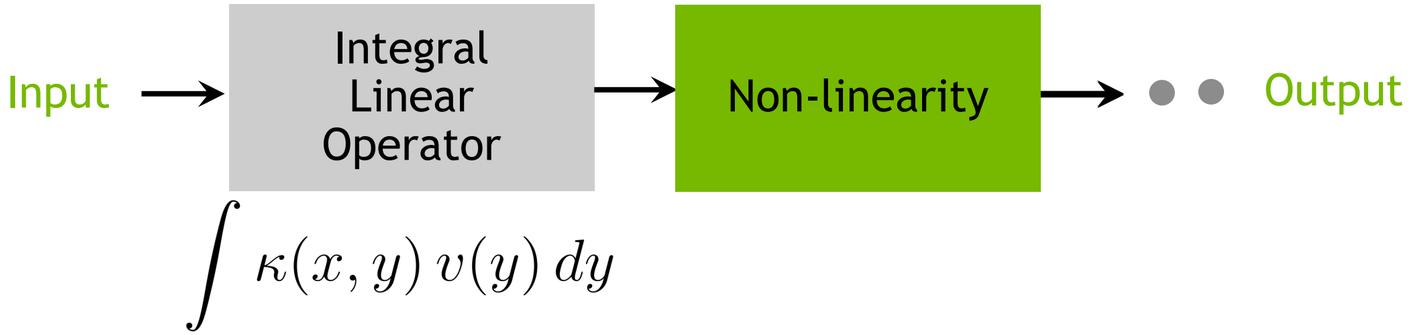

- Integral operator outputs functions (not just finite-dimensional vectors).
- Integral operator is discretization invariant.

# DISCRETIZATION-INVARIANCE

| Property \ Model | CNNs | DeepONets | CNNs+Interpolation | Neural Operators |
|---|---|---|---|---|
| Discretization Invariance | ✗ | ✗ | ✓ | ✓ |
| Query at any point | ✗ | ✓ | ✓ | ✓ |
| Input at any point | ✗ | ✗ | ✓ | ✓ |
| Universal Approximation | ✗ | ✓ | ✗ | ✓ |

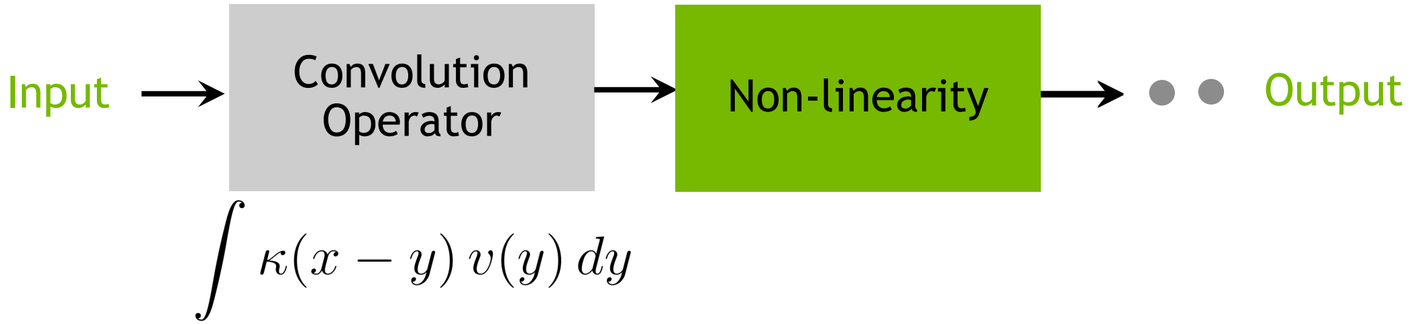
- Neural operators are discretization-invariant.
- Neural operators are universal approximators in function spaces.

# NEURAL OPERATOR: A GENERAL FRAMEWORK



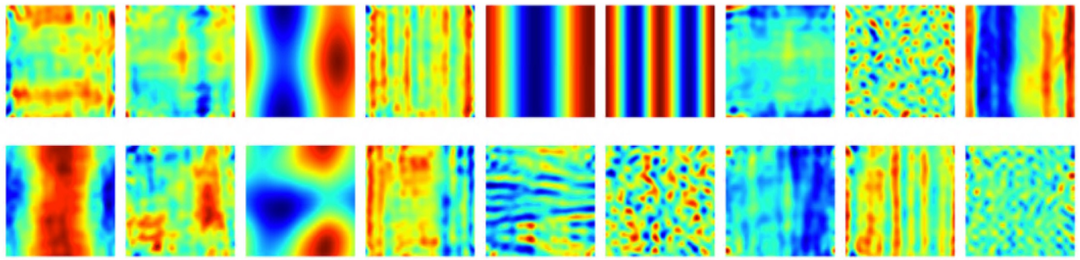

- Integral operator outputs functions (not just finite-dimensional vectors).
- Integral operator is discretization invariant.

# FOURIER NEURAL OPERATOR: EFFICIENT FRAMEWORK




- Special case of integral linear operator: convolution
- Global (continuous) convolution over the domain

Filters in CNN



Fourier Filters

# FOURIER TRANSFORM FOR GLOBAL CONVOLUTION

Integral linear operator

$$\int \kappa(x, y)\, v(y)\, dy$$

Convolution operator
(special case of integral operator)

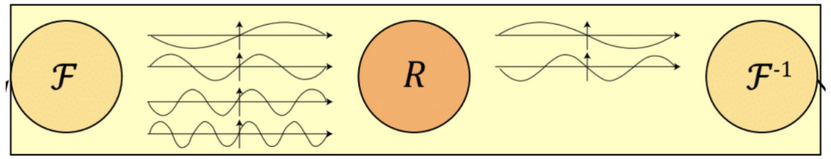
$$\int \kappa(x - y)\, v(y)\, dy$$

Solving convolution in Fourier domain

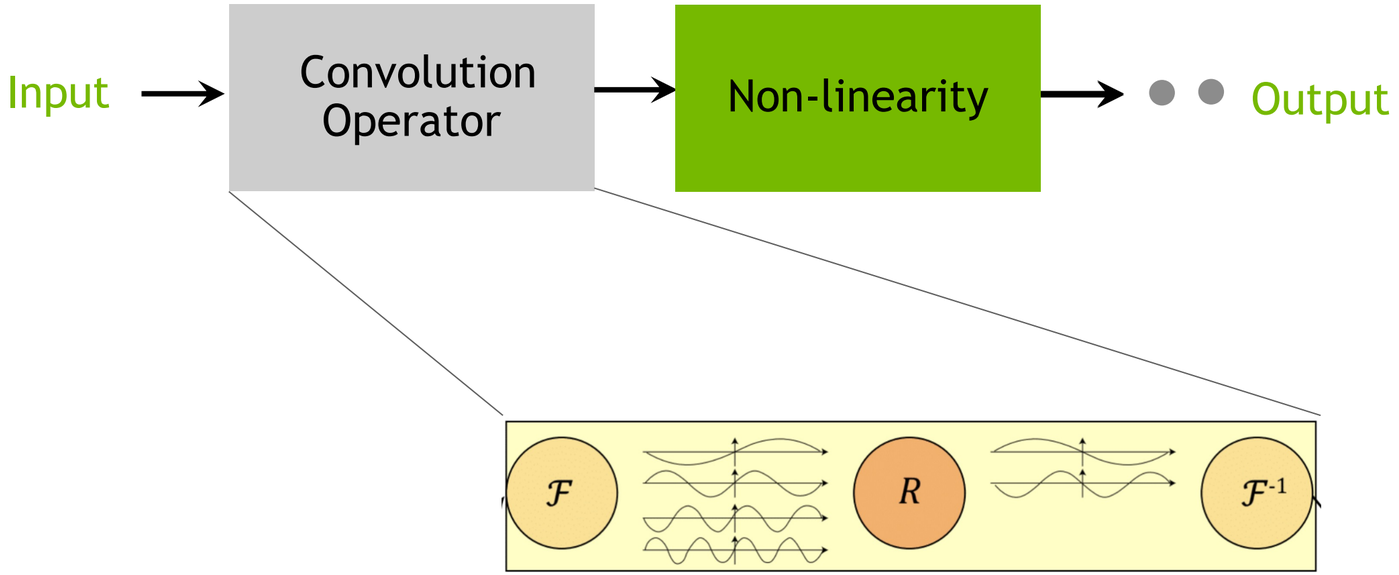
$$\mathcal{F}^{-1}(\, \mathcal{F}(\kappa)\, .\, \mathcal{F}(v)\, )$$

$$R := \mathcal{F}(\kappa)$$

Learn weights *R* in Fourier Domain



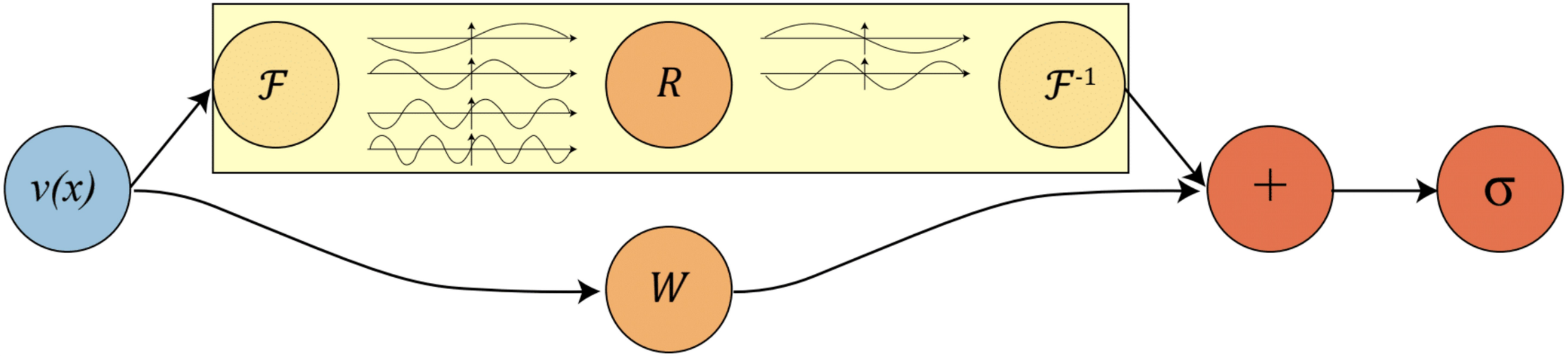
# FNO: FOURIER NEURAL OPERATOR




- Convolution = multiplication in frequency domain.
- Learning weights in frequency domain.
- Fourier Transform implements convolution and also discretization invariant.

# FOURIER LAYER IN NEURAL OPERATOR

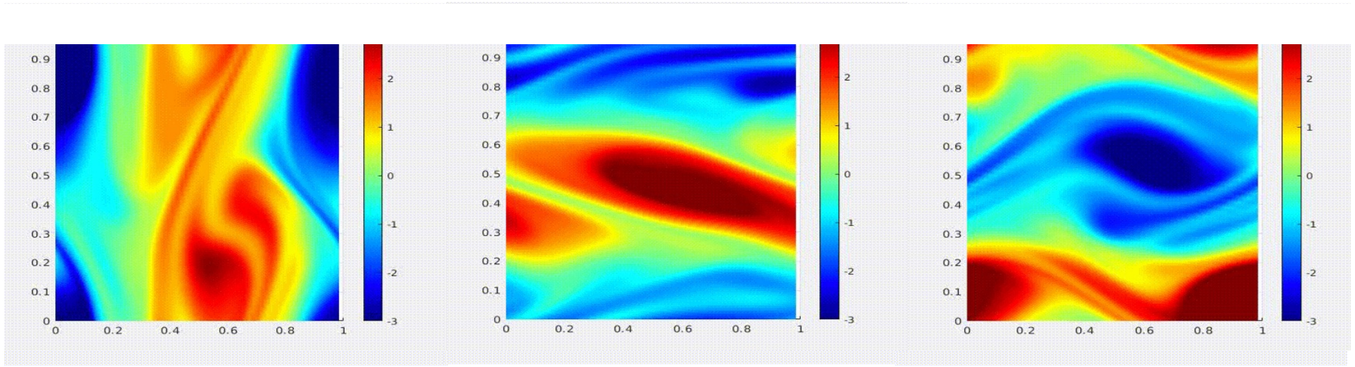
The linear transform *W* is pointwise operation at each location x that can help learn residual from Fourier layer, e.g., non-periodic boundary



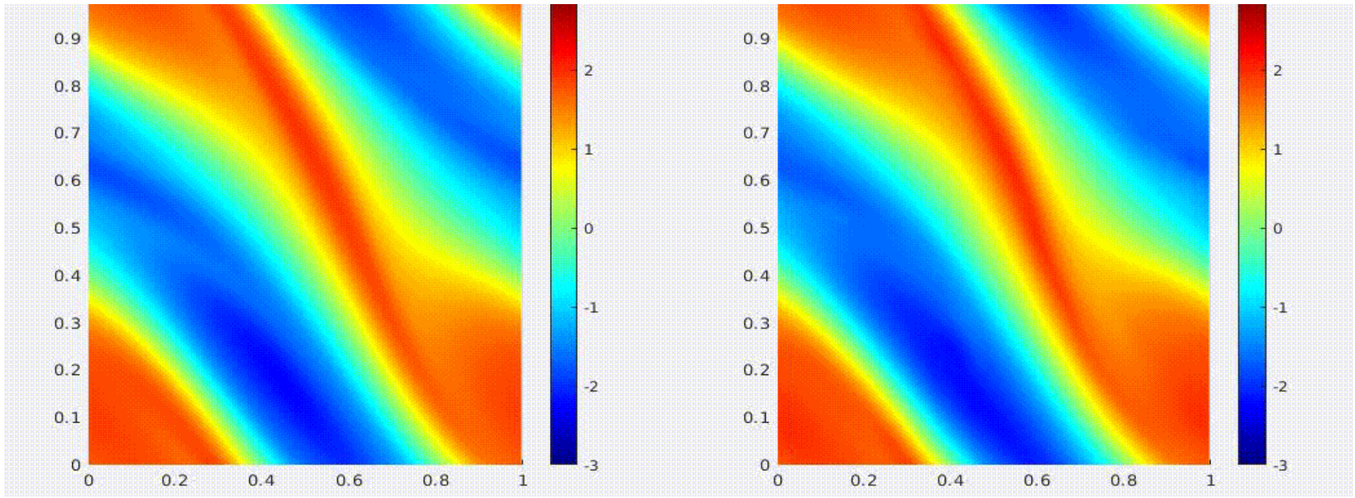
# DEMONSTRATING DISCRETIZATION INVARIANCE OF FNO

## Zero-shot super-resolution

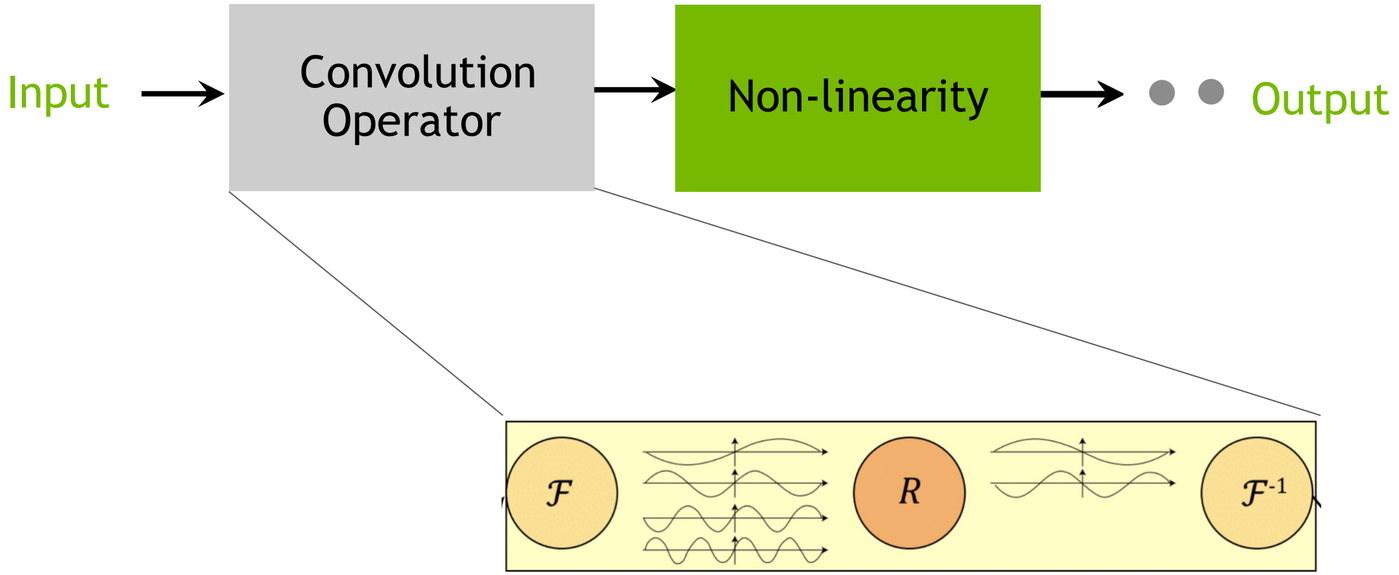
Train using coarse resolution data



Directly evaluate on higher resolution (no re-training)



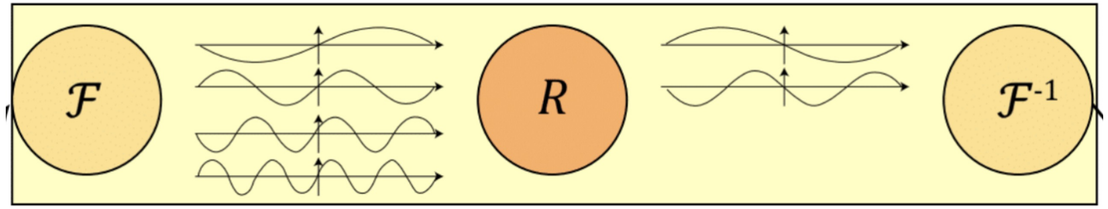
# FNO: FOURIER NEURAL OPERATOR




- Convolution = multiplication in frequency domain.
- Learning weights in frequency domain.
- Fourier Transform implements convolution and also discretization invariant.

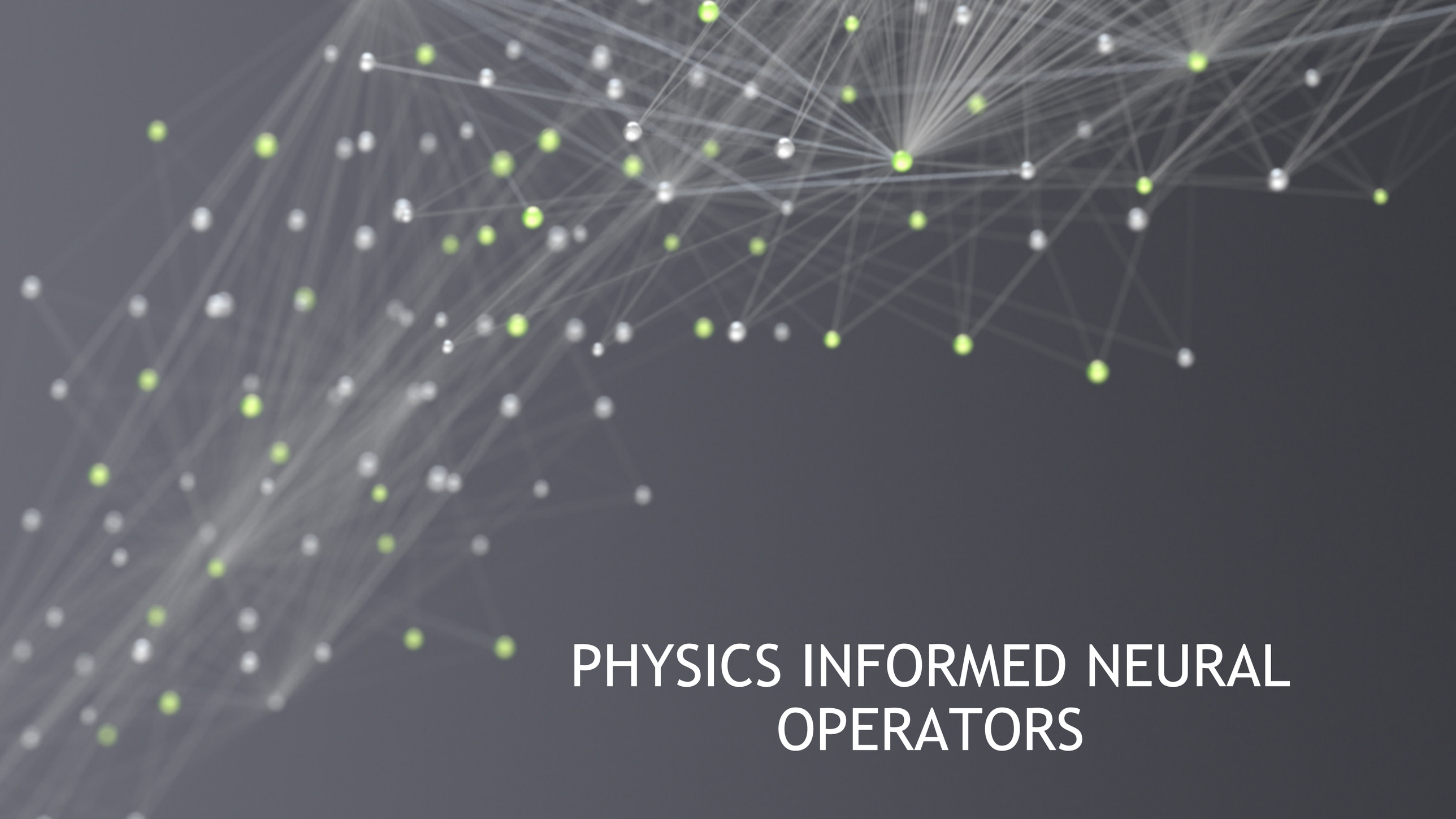
# OPTIMIZATION OF FOURIER NEURAL OPERATOR

- For successful learning, need appropriate selection of
  - Number of frequency modes
  - Resolution of training data



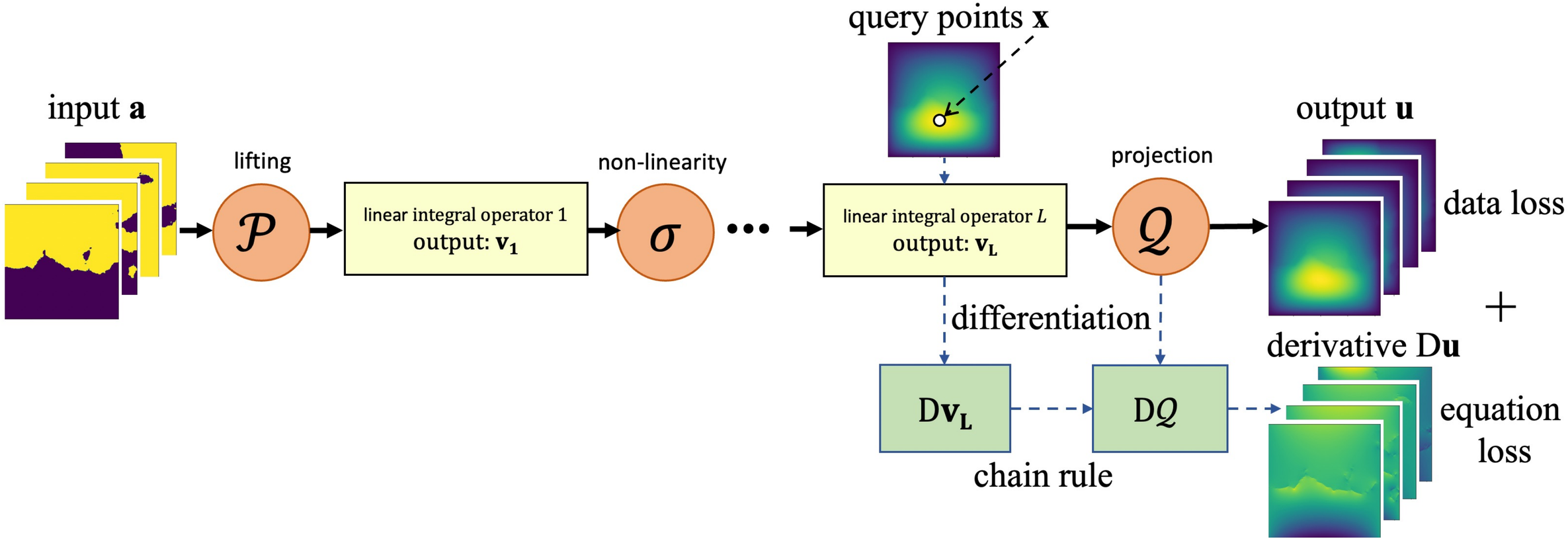

- Too few frequency modes and low-resolution training data can cause underfitting.
- Too many frequency modes can lead to overfitting.
- High resolution training data is computationally expensive to obtain and train on.

- Proposed: Incrementally augment both frequency modes and training resolution
- Is faster to train and better generalization.



# PHYSICS INFORMED NEURAL OPERATORS

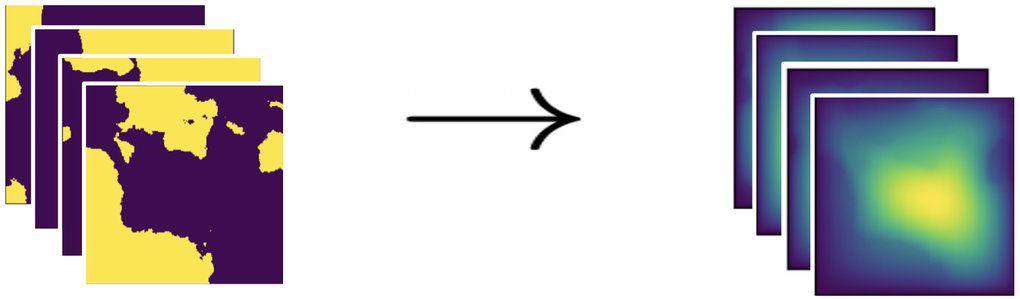
# PINO: PHYSICS-INFORMED NEURAL OPERATOR



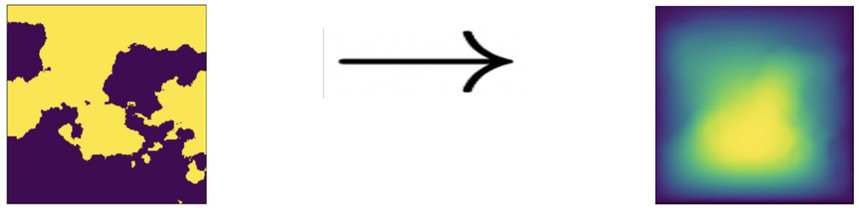
# PINO: PHYSICS-INFORMED NEURAL OPERATOR

PINO can learn solution operator for a family of equations and fine-tune on an instance

Operator learning

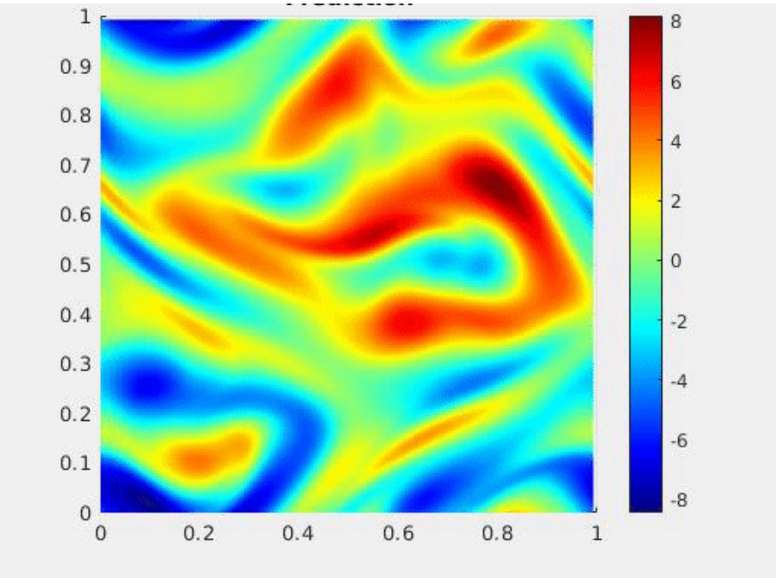


Instance-wise finetuning

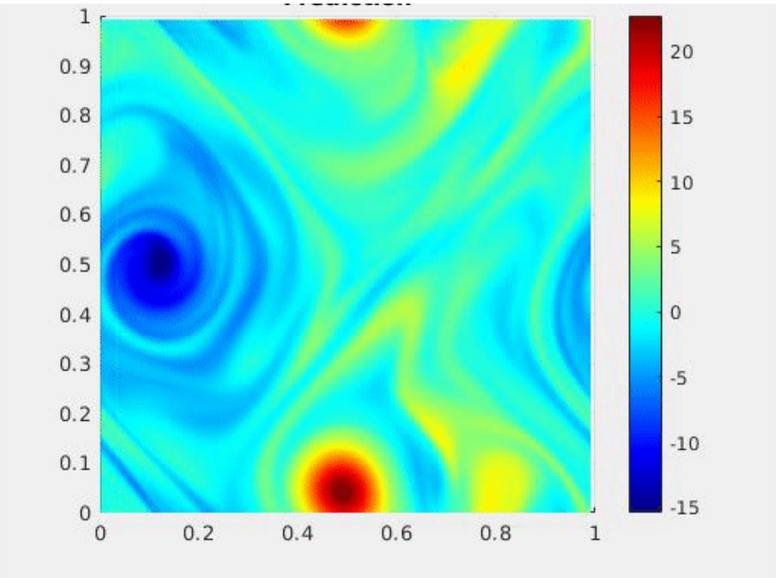


# TRANSFER LEARNING WITH PINO

Operator learned on Re100, fine-tune to Re500. Converges 3x faster.



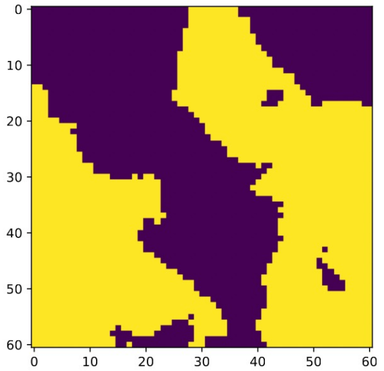

Re100



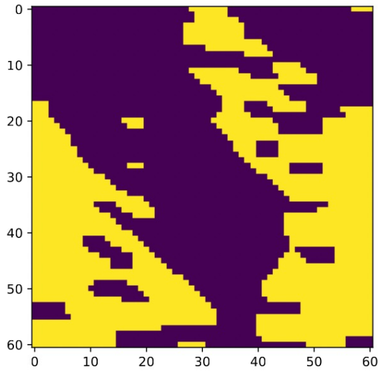

Re500

# INVERSE PROBLEMS WITH PINO

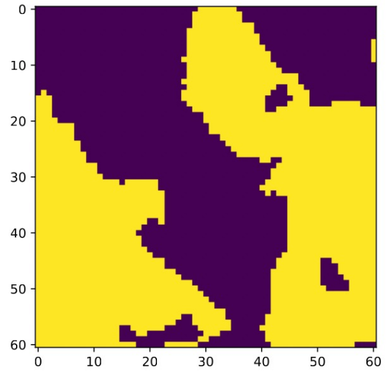
Inverse problem: given solution of forward simulation, recover input.
PINO makes the inverse prediction more physically valid.



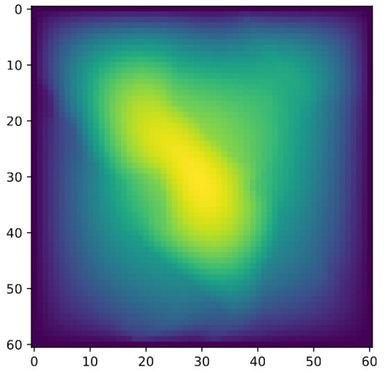

(a) Ground truth input $a$



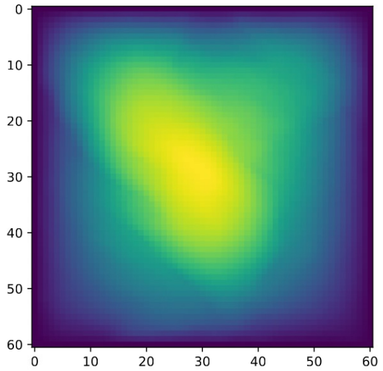

(b) Inversion using only data constraint



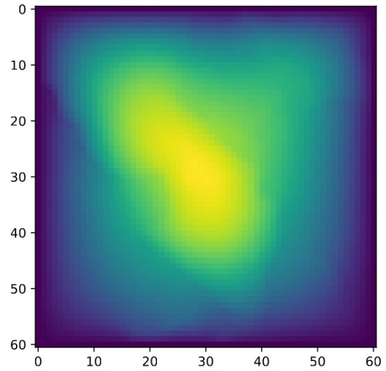

(c) Inversion using data and PDE constraints




(d) Observed output function



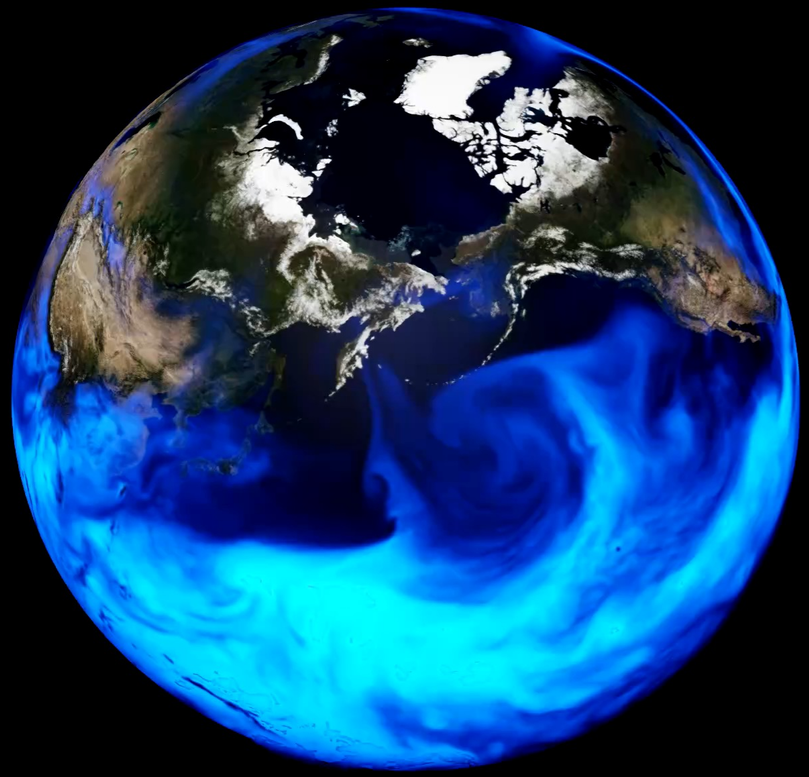

(e) Output function of inversion using only data constraint



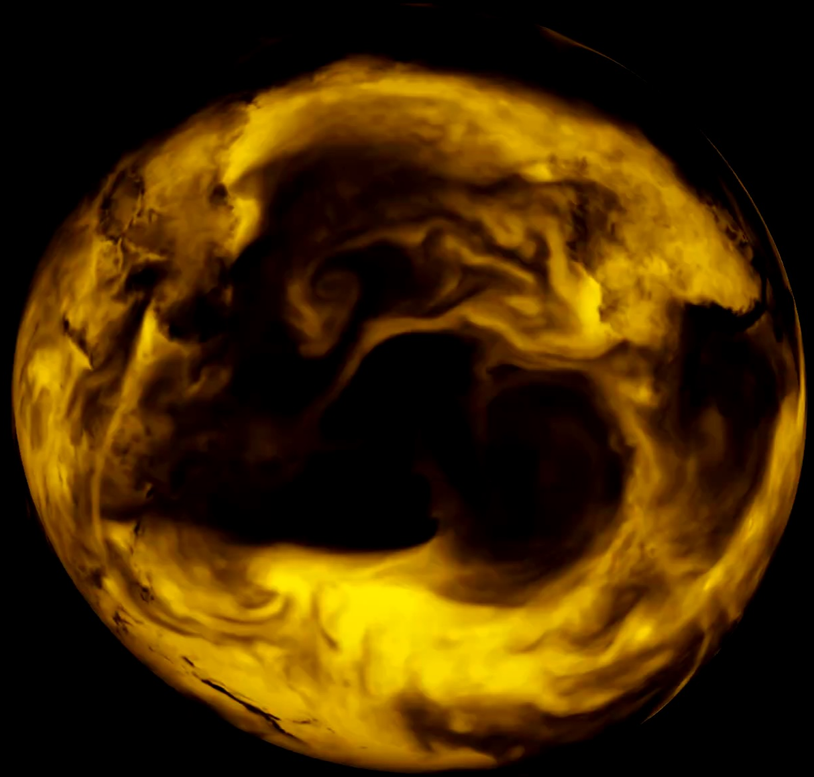

(f) Output function of inversion using data and PDE constraints

# APPLICATIONS OF FNO

Ground Truth



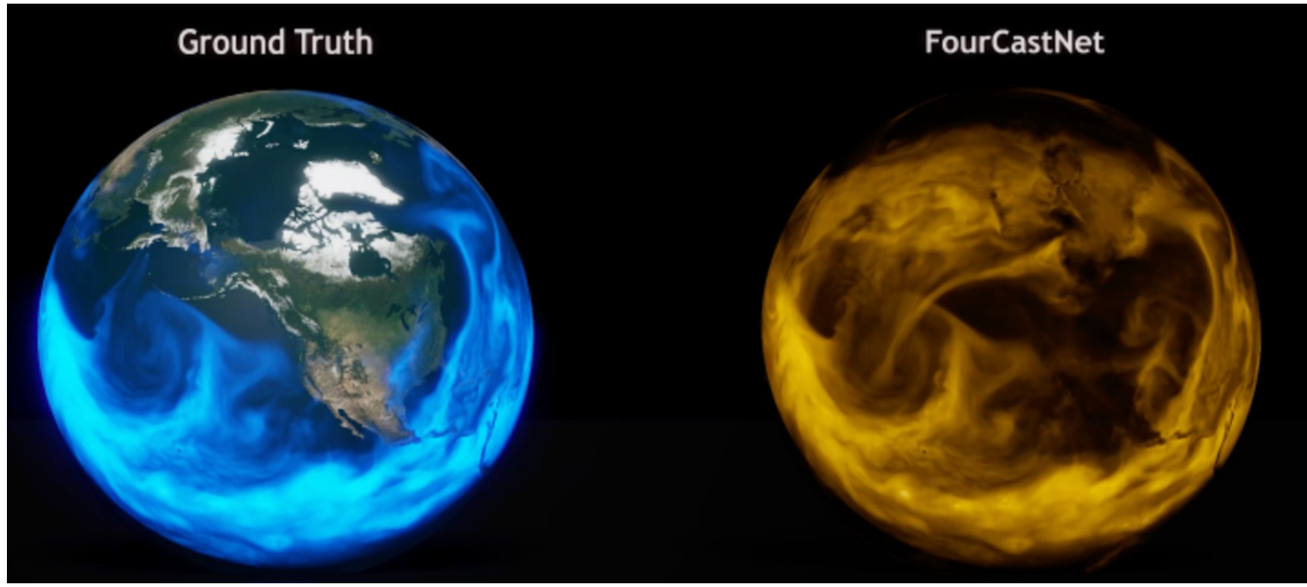
FourCastNet



Our AI (FourCastNet) is **45,000 times** faster than current weather models

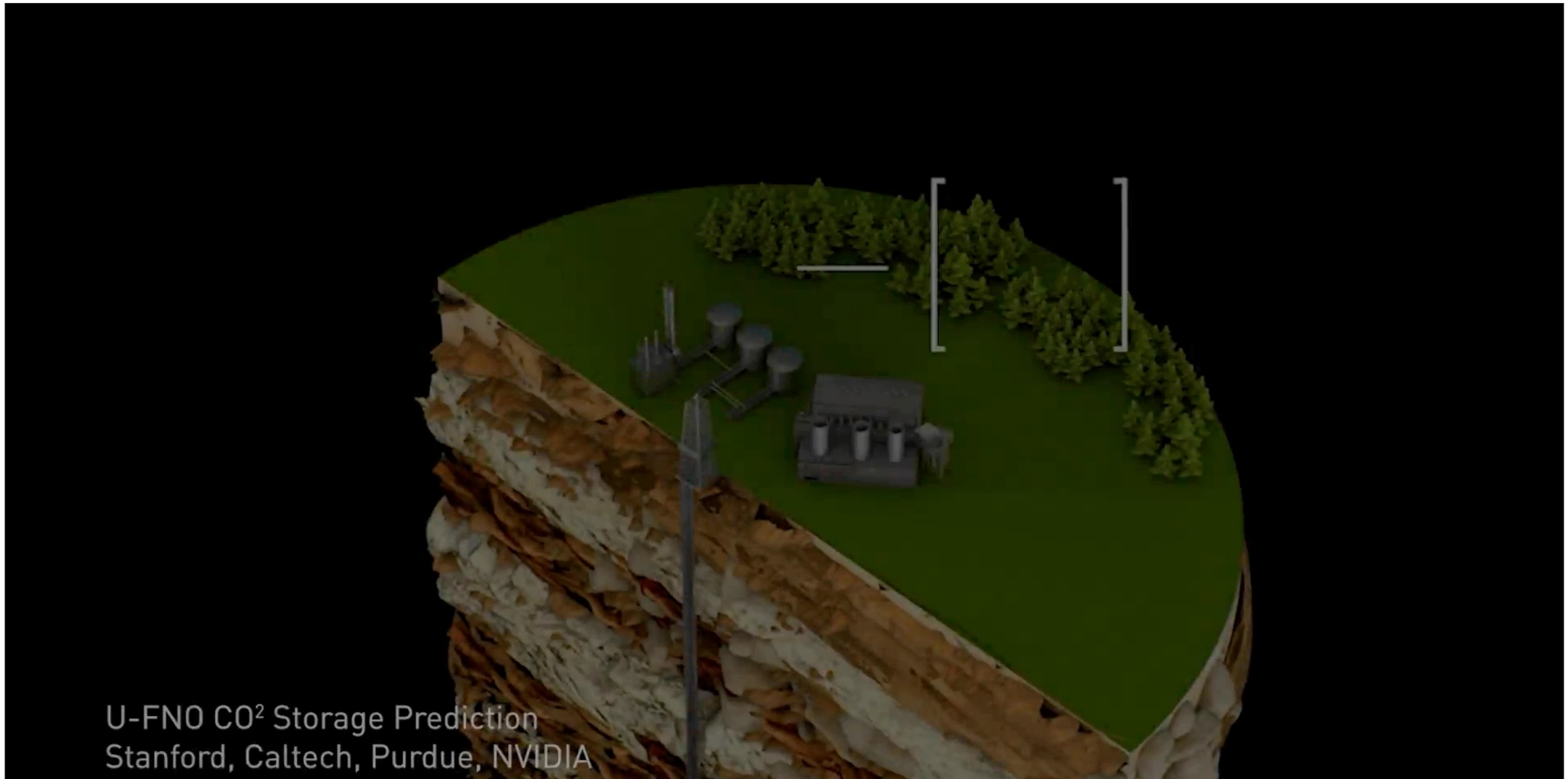
# FOURCASTNET FOR WEATHER PREDICTION

- Trained on 10 TB of weather data
- **8x** higher resolution than any other AI model for weather forecasting.
- **45,000x** speedup
- **25000x** smaller energy footprint.
- 1000-member ensemble in a fraction of a second.
- Unparalleled accuracy of surface winds and precipitation up to one week.



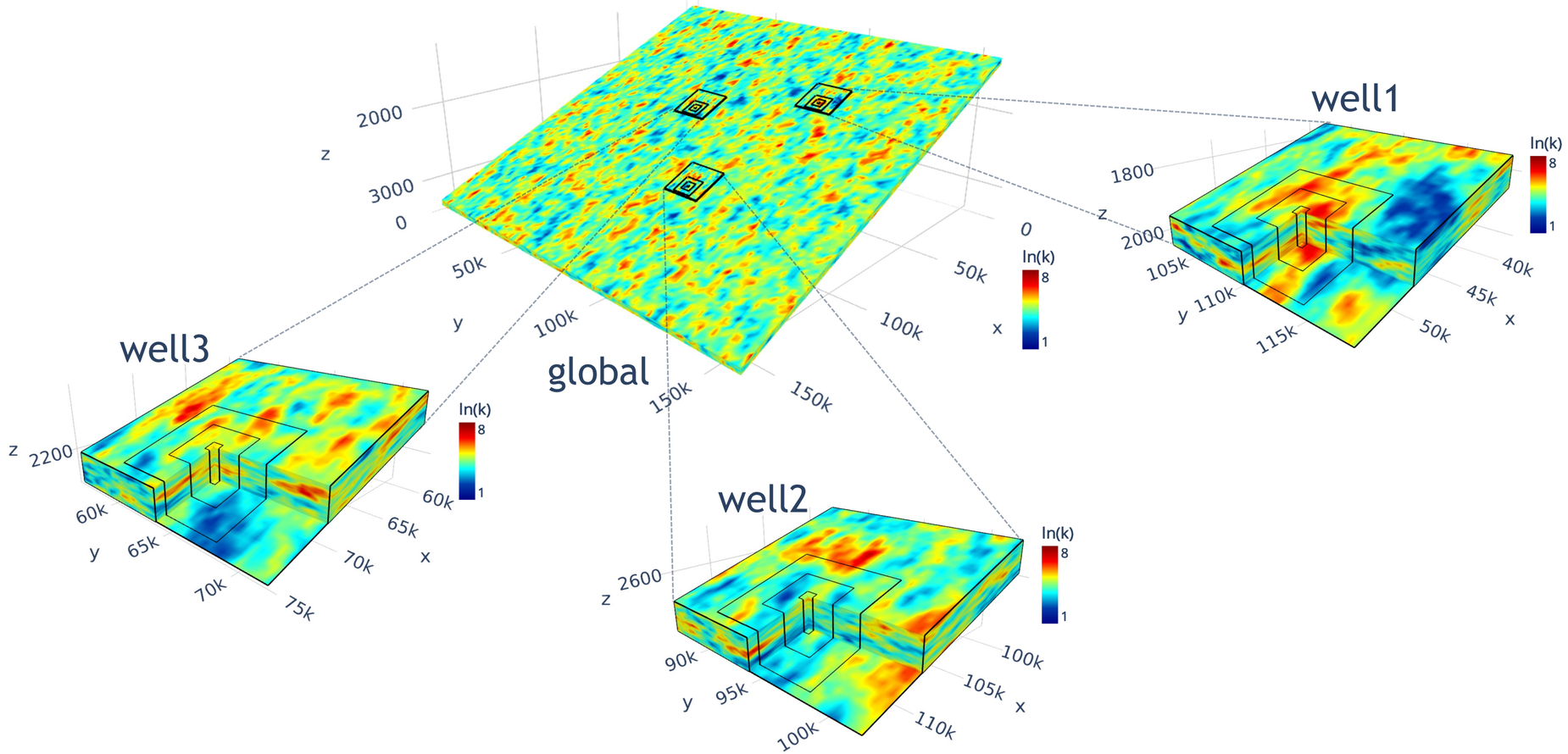
# CLIMATE CHANGE MITIGATION: MODELING $CO_2$ STORAGE

Our AI Method accelerates by 700,000 times



# FOUR-DIMENSIONAL CCS MODELING WITH AI (FNO)

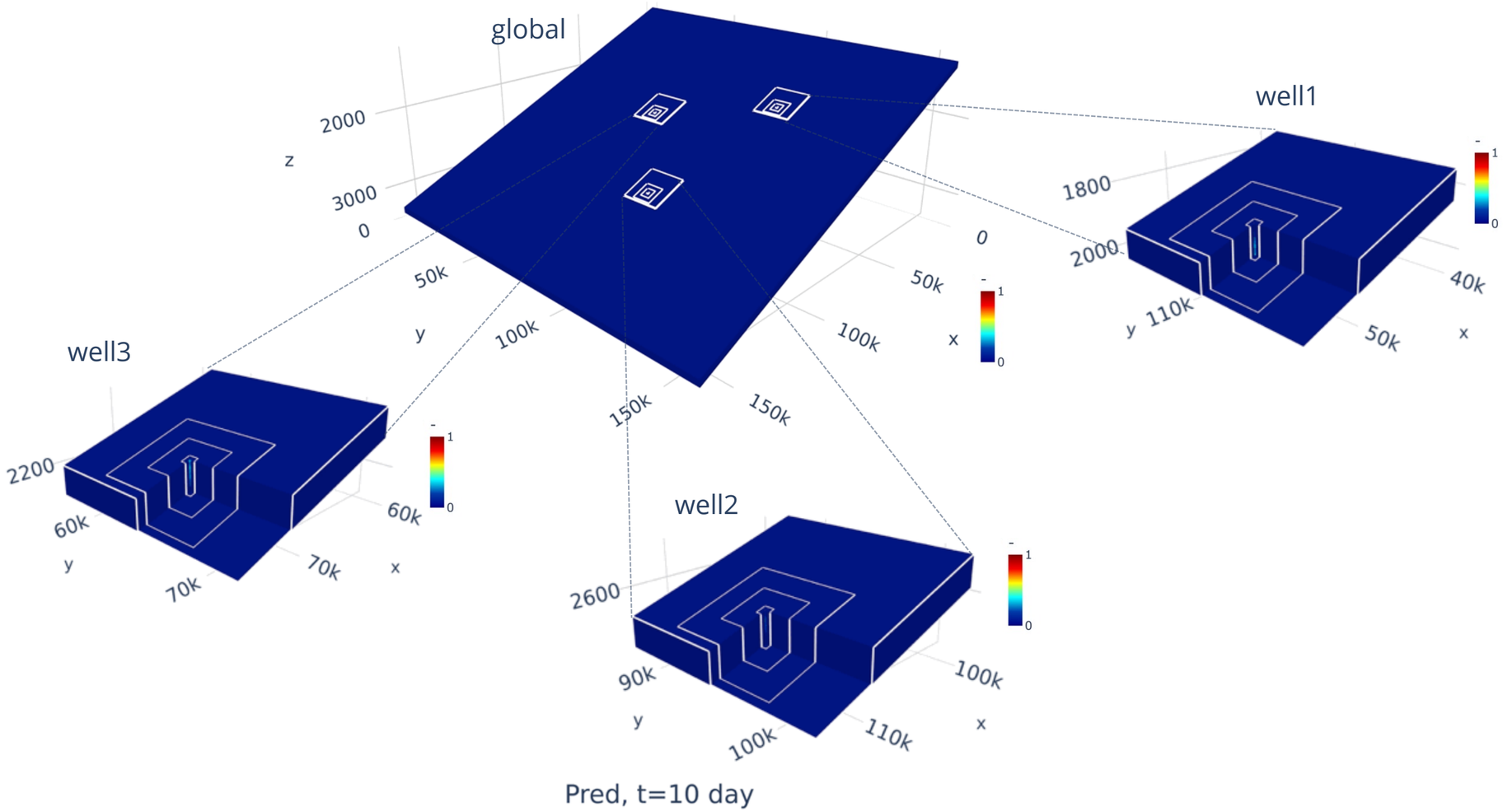
## Our AI Method accelerates by 700,000 times




Permeability Heat Map

# FOUR-DIMENSIONAL CCS MODELING WITH AI (FNO)

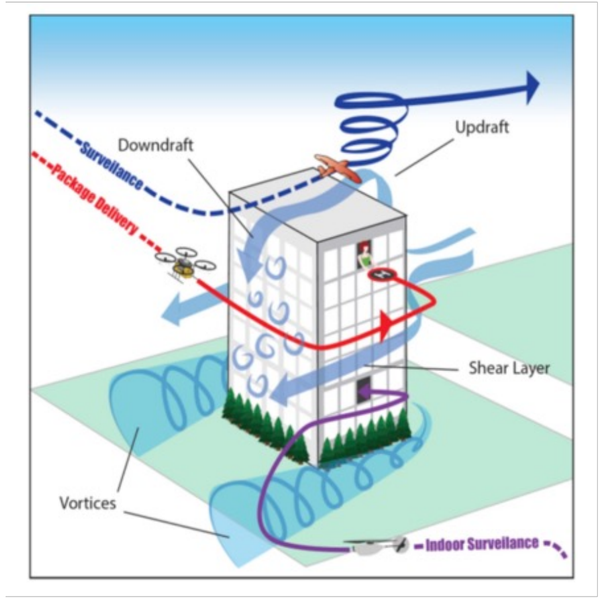
## Our AI Method accelerates by 700,000 times

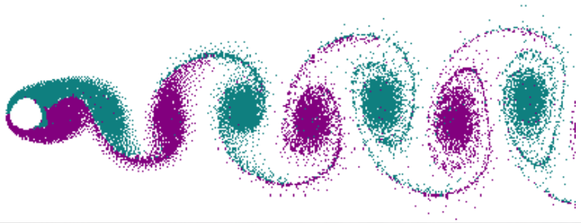


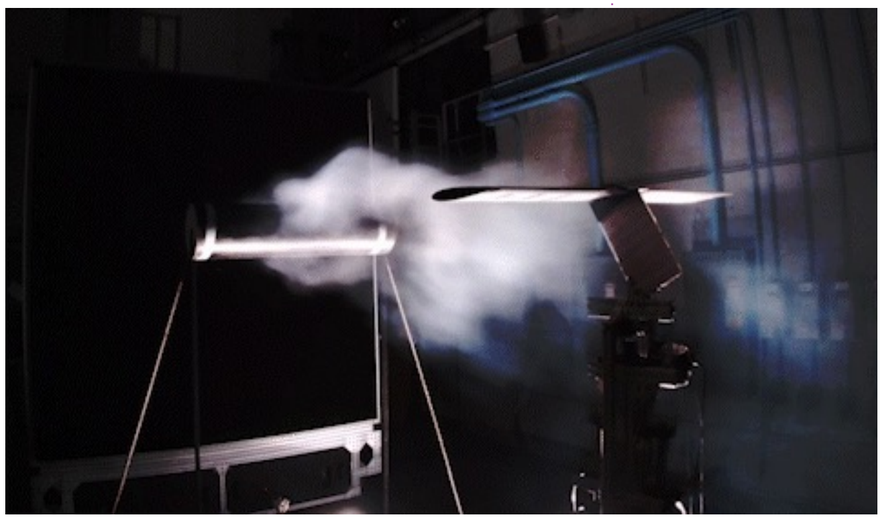

Gas Saturation

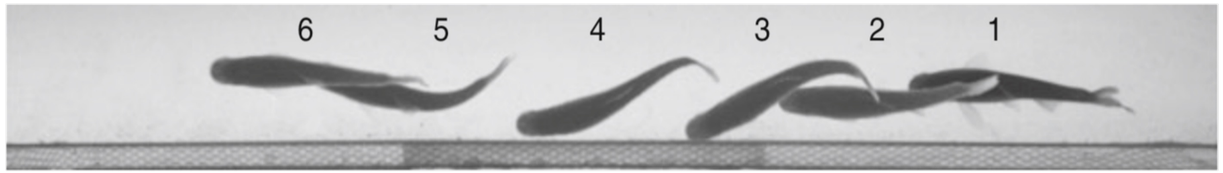
# FNO for RAPID ADAPTATION TO TURBULENCE

## Real-time flow prediction in drones





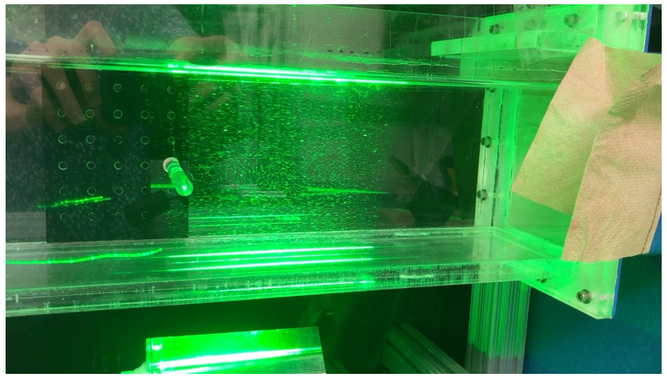


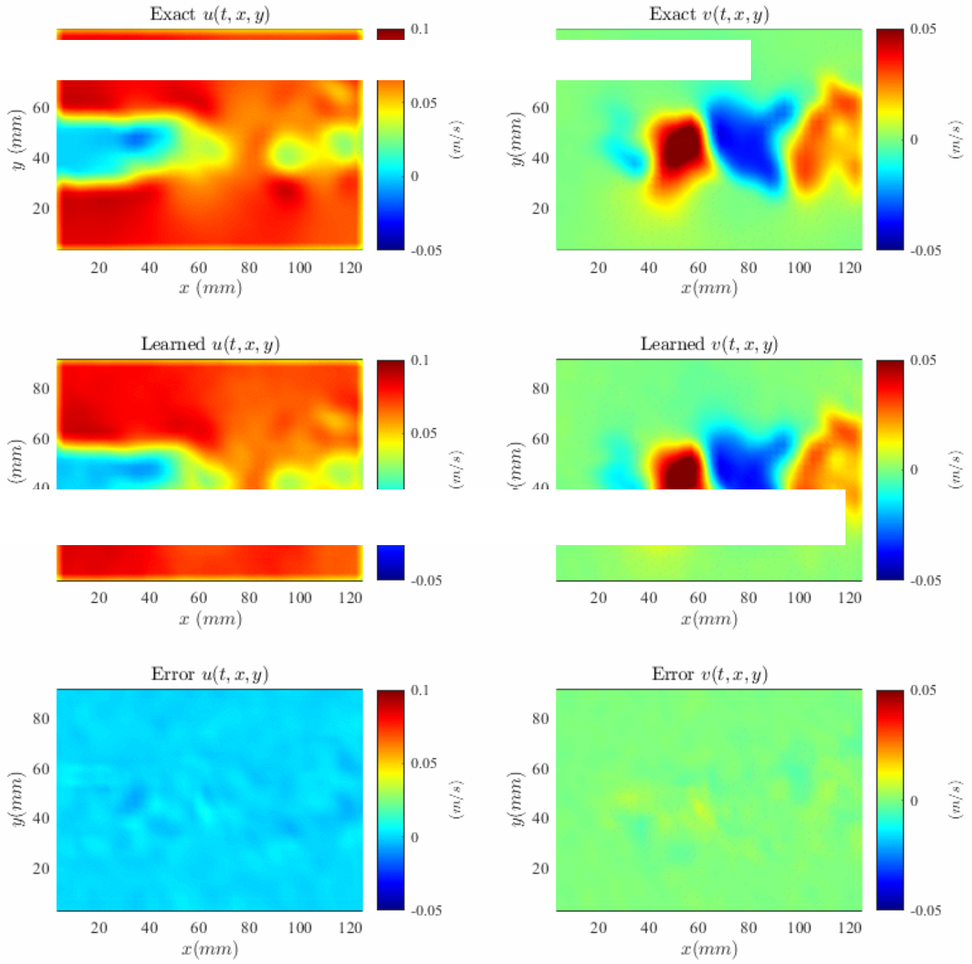

*Watkins et al., *Ten questions concerning the use of drones in urban environments*, Building and Environments, 2019

*Windsor, S.P., *Hydrodynamic Imaging by Blind Mexican Cavefish*, Flow Sensing in Air and Water, 2014

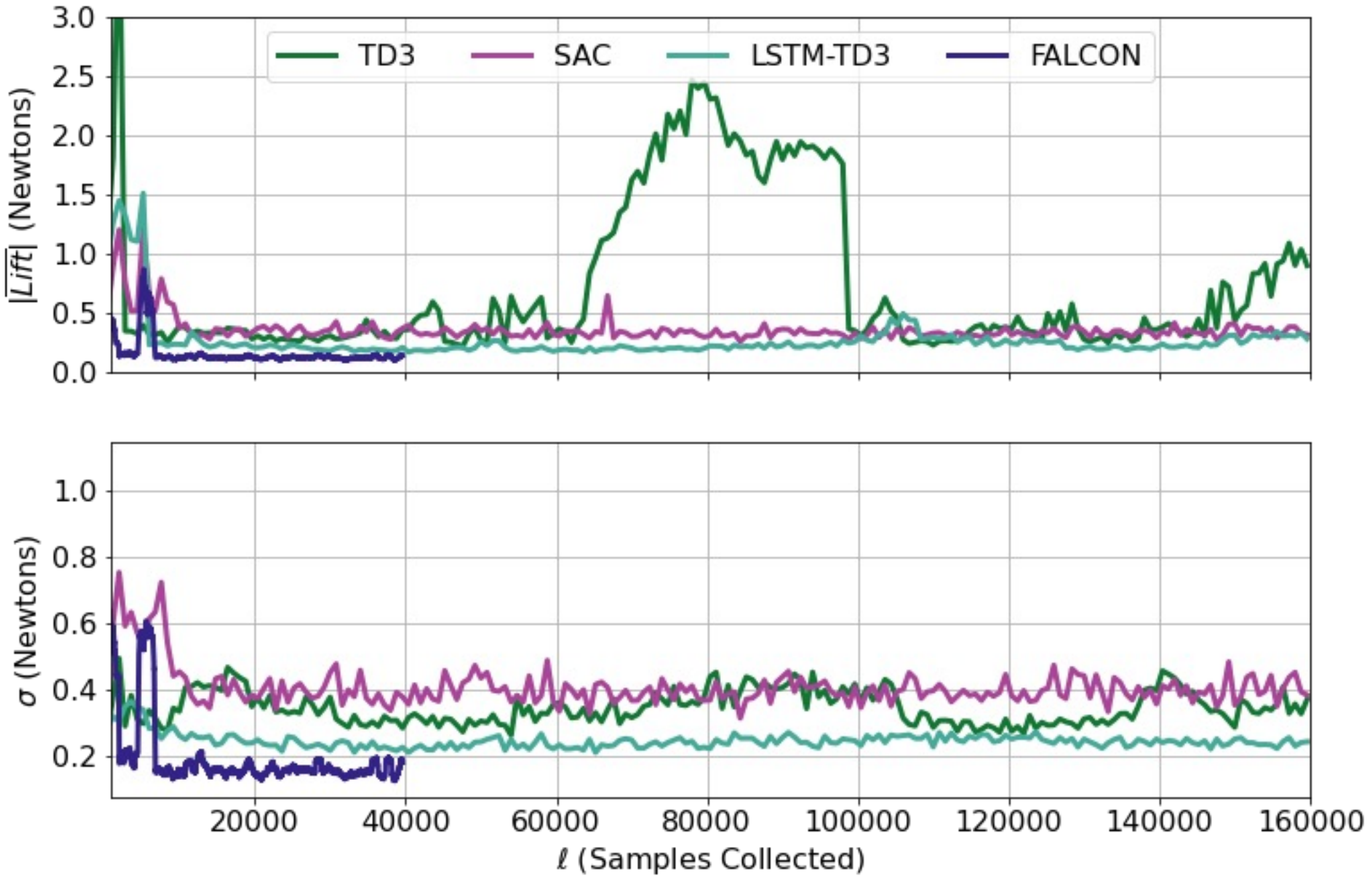
# FIRST REAL-TIME FLUID FLOW PREDICTION

## FNO learns directly from noisy experimental data



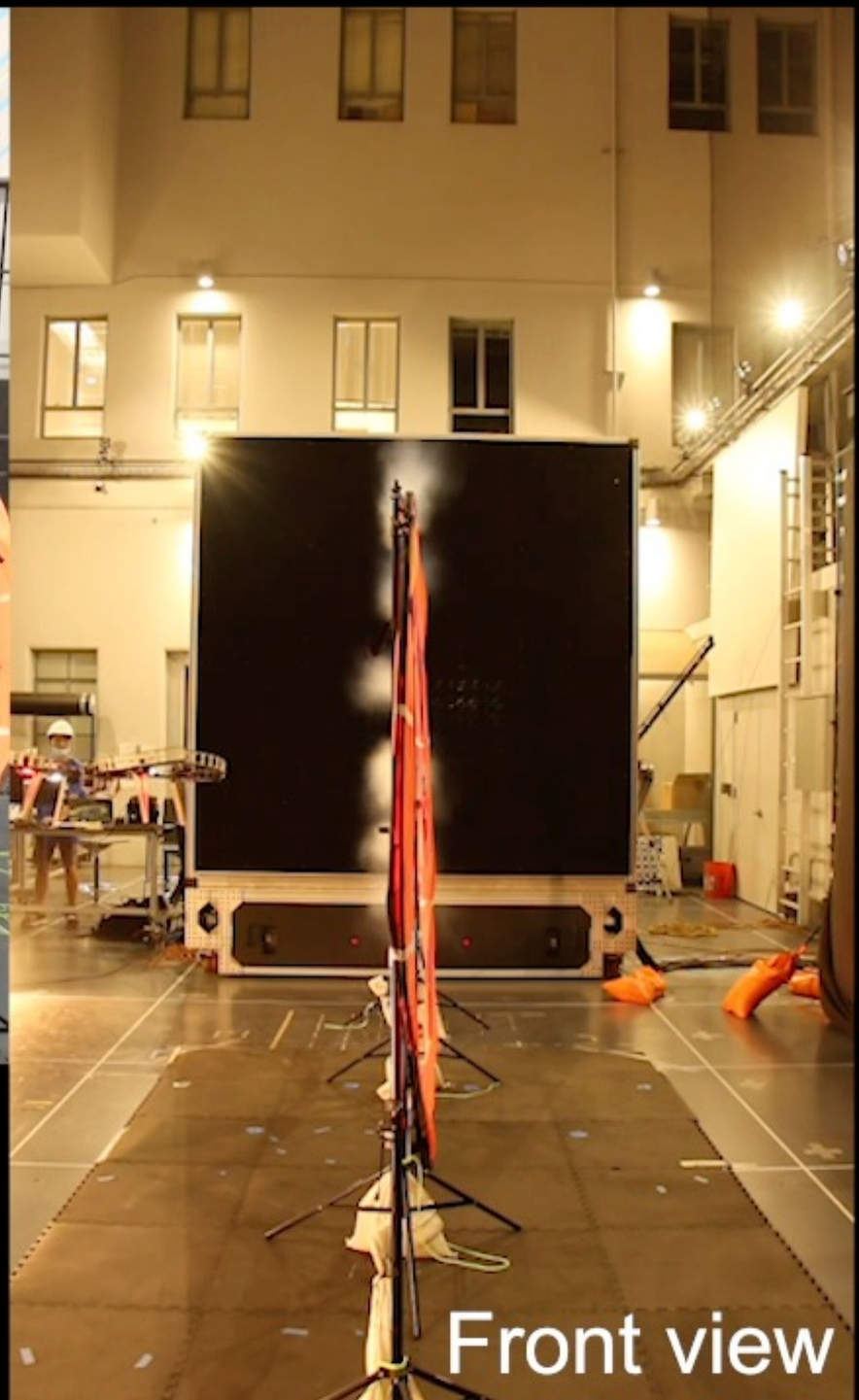


# RAPID ADAPTATION TO TURBULENT CONDITIONS




| Method | Mean(N) | σ(N) | # of Samples |
|---|---|---|---|
| **FALCON** | **0.120** | **0.158** | **$2.36 \times 10^4$** |
| LSTM-TD3 | 0.212 | 0.233 | $1.52 \times 10^5$ |
| PID | 0.208 | 0.260 | NA |
| TD3 | 0.904 | 0.373 | $1.60\times 10^5$ |
| SAC | 0.435 | 0.395 | $1.60\times 10^5$ |

**State-of-the-art stabilization performance with an order of magnitude less samples in falcon**


With our method, the UAV precisely follows an ellipse through two narrow gates in wind.

# OPTIMIZATION CHALLENGES: DETERMINISTIC INITIALIZATION

# RANDOM INITIALIZATION IN NEURAL NETWORKS

Requires careful selection of **variance of initial weights** at different layers

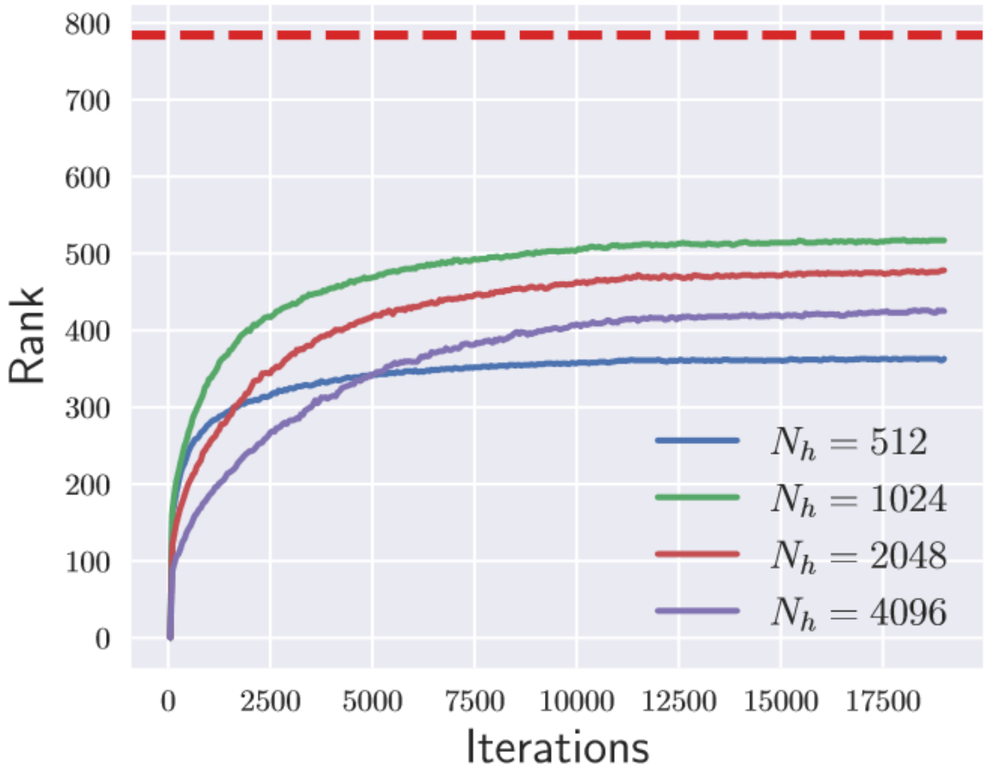
- Variance too high: can lead to gradient explosion.
- Variance too low: can lead to vanishing gradients.
- Need of handcrafted initialization for different architectures (e.g., Xavier, Kaiming, Fixup)
- Batch normalization usually required to stabilize the signal propagation.

Weak reproducibility

- Large training variation over repeated experiments with different random seeds
- Interferes with accurate uncertainty quantification

# IDENTITY INITIALIZATION

- Identity initialization: Initialize all weight matrices as identity
- But when network layers have different widths:
  - Identity initialization leads to rank constraint during entire training.
  - Rank of all weight matrices is bounded by min( input dimension, all layers except output layer)
- Under identity initialization, increasing width of layers doesn't improve expressivity. **Leads to underfitting.**




No matter how wide ($N_h$) the hidden layer is, its rank is always bounded by input dimension $N_x = 784$.

# OUR PROPOSAL: ZERO INITIALIZATION

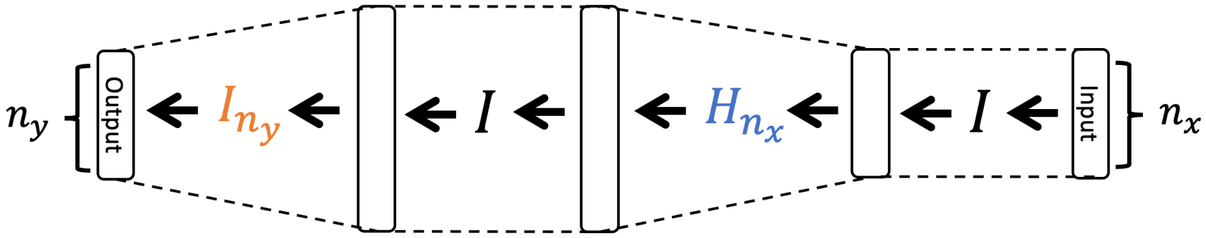
ZerO: Initialize all weights with only zeros and ones

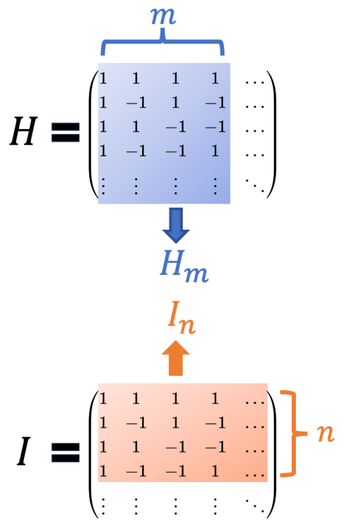
All layers are initialized as identity or Hadamard transforms

Hadamard transform breaks rank constraints of identity initialization

Benefits

- Universally applicable to all architectures
- State-of-art accuracy
- Training ultra deep networks without batch normalization
- Better training reproducibility
- Incremental learning trajectory with low-rank solutions at convergence.

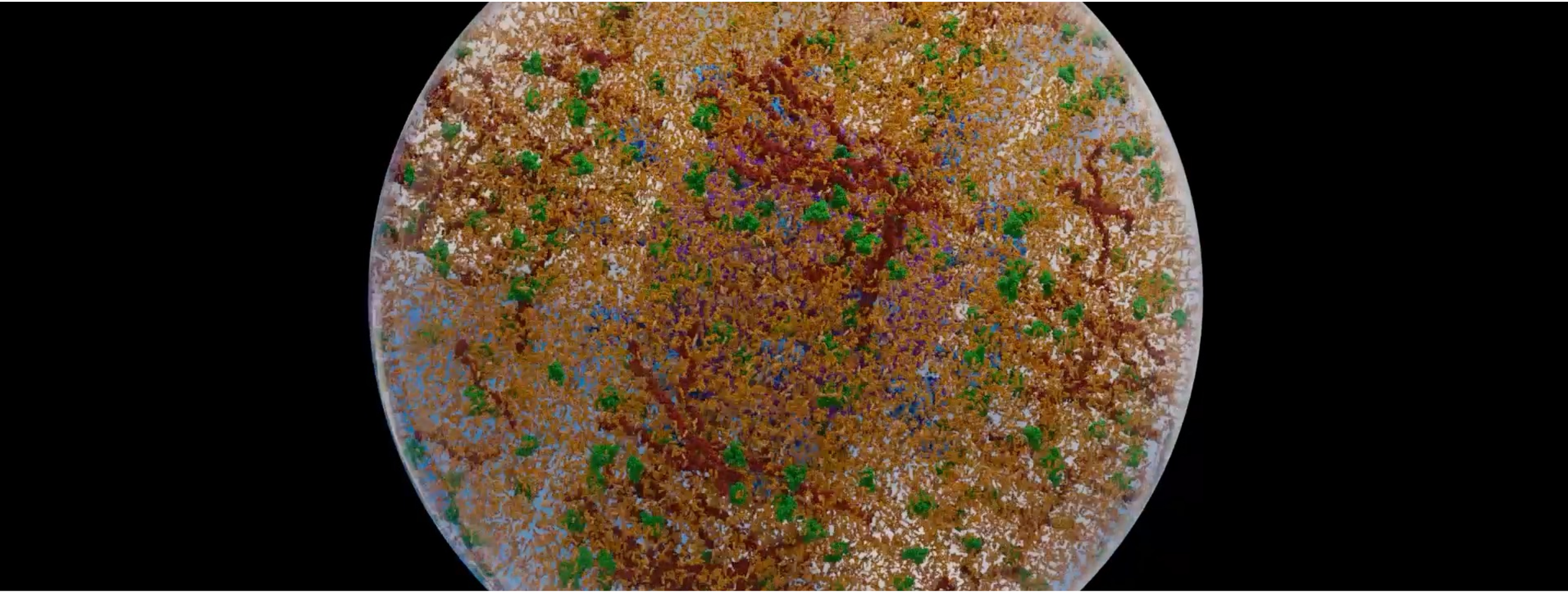


# AI FOR CHEMISTRY

# AI IS TRANSFORMING DRUG DISCOVERY

AI Accurately Predicts Quantum-level Molecular Properties with 1000x Speedup



**RESPIRATORY AEROSOL + SARS-COV-2 VIRUS**

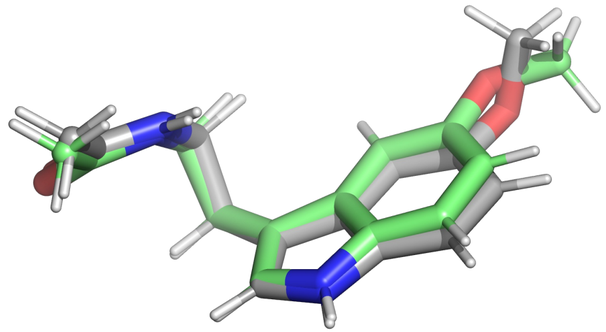
Gordon-Bell Special Prize Finalist

# AI IS TRANSFORMING DRUG DISCOVERY

AI accurately predicts quantum-level molecular properties with 1000x speedup

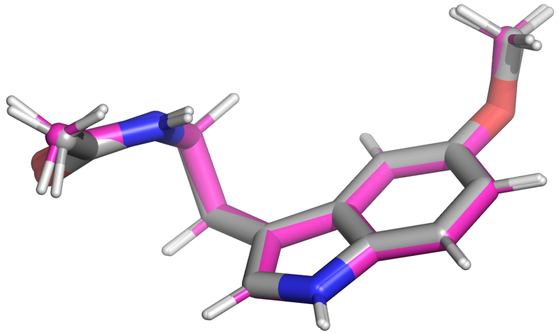
Geometry optimization for Melatonin

Traditional Approximation

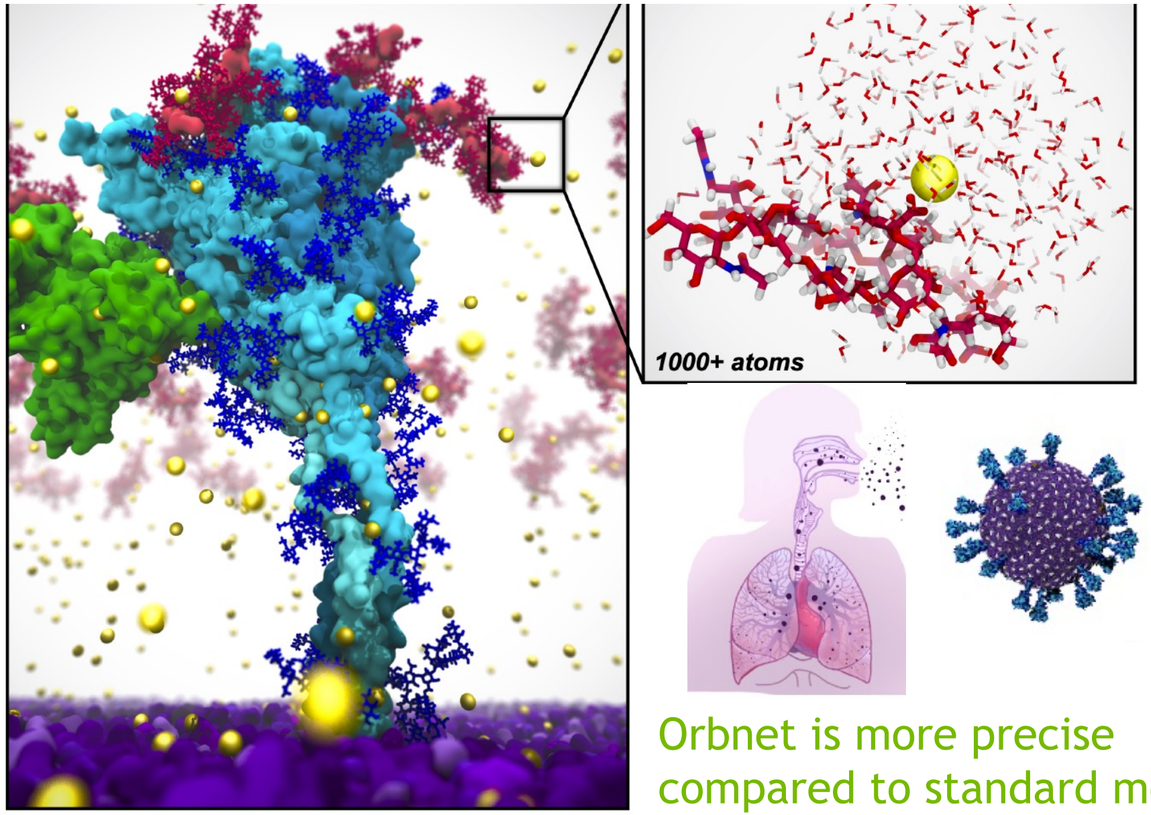


AI is 200 times faster and more accurate

AI (Orbnet)



Respiratory aerosol + SARS-CoV-2 virus



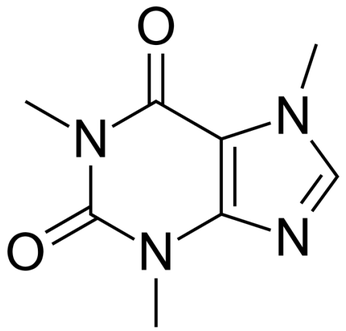

Orbnet is more precise compared to standard methods

# AI FOR CHEMISTRY WITH QUANTUM FEATURES

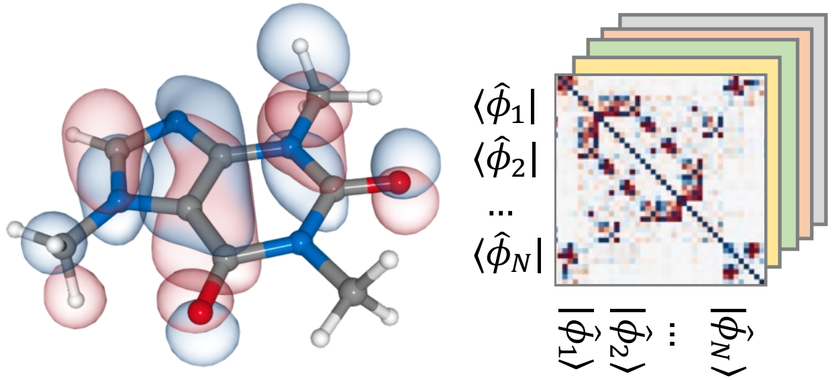
Informatic Representation:
SMILES Strings

CN1C=NC2=C1C(=O)N(C(=O)N2C)C

Chemists' Representation:
Graphs




Physicists' Representation:
Orbitals & Quantum Interactions




Transferability

Low → High

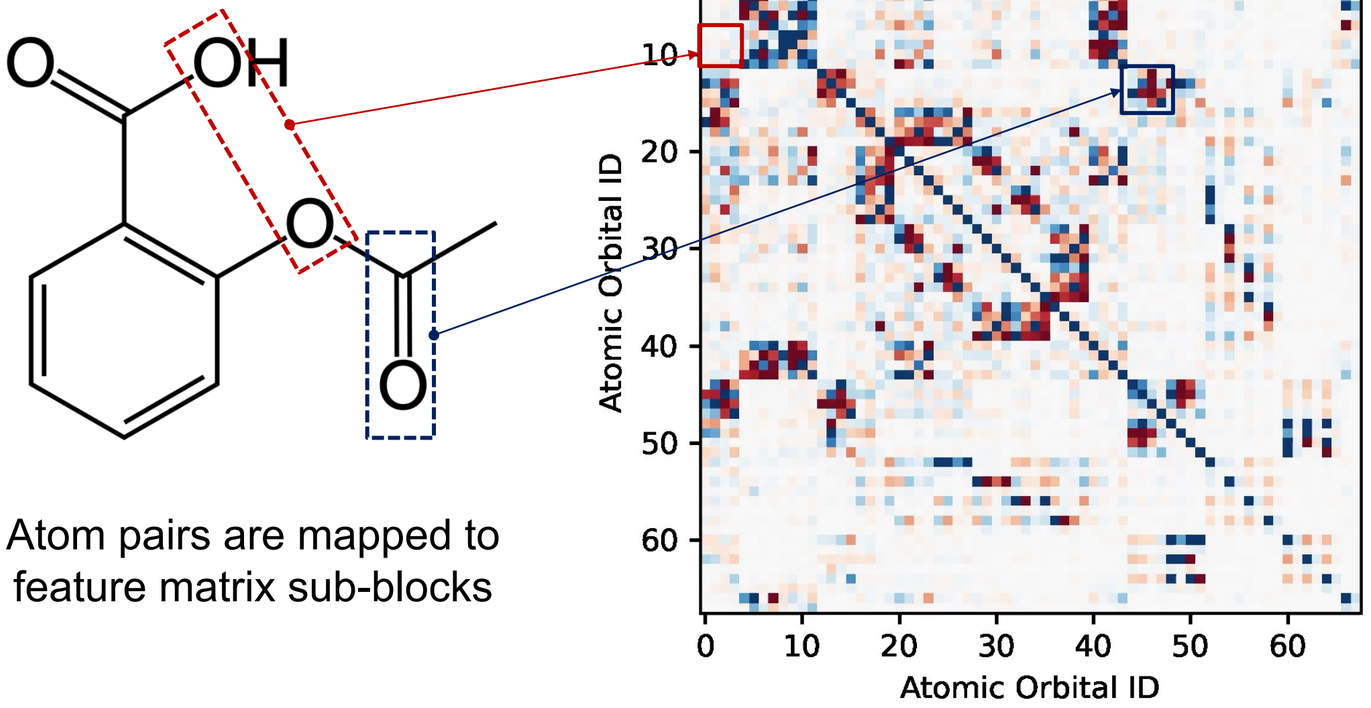
Quantum-mechanical features lead to data efficiency

ENTOS

Caltech

# ORBITAL-BASED FEATURES FOR MOLECULES

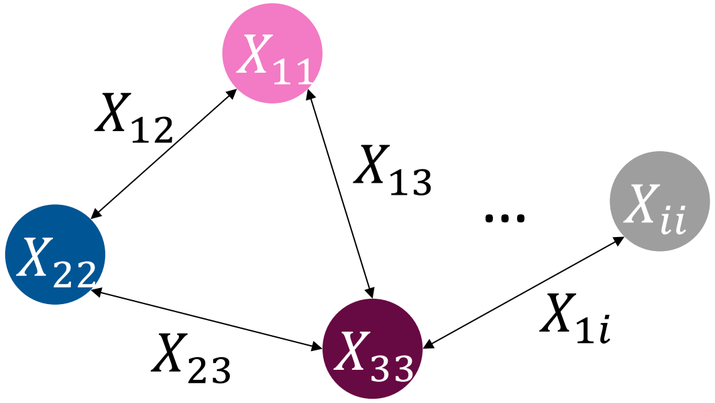
Pairwise features: quantum operators evaluated in atomic orbital basis




Atom pairs are mapped to feature matrix sub-blocks

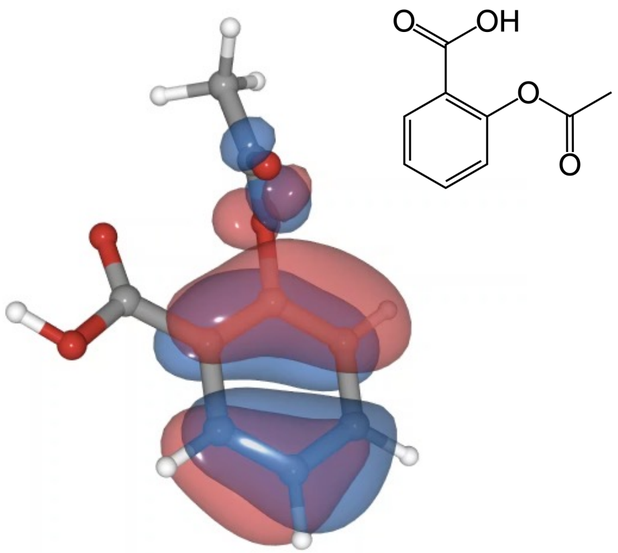
Graph view:

- Nodes - diagonal feature sub-blocks
- Edges - off-diagonal feature sub-blocks

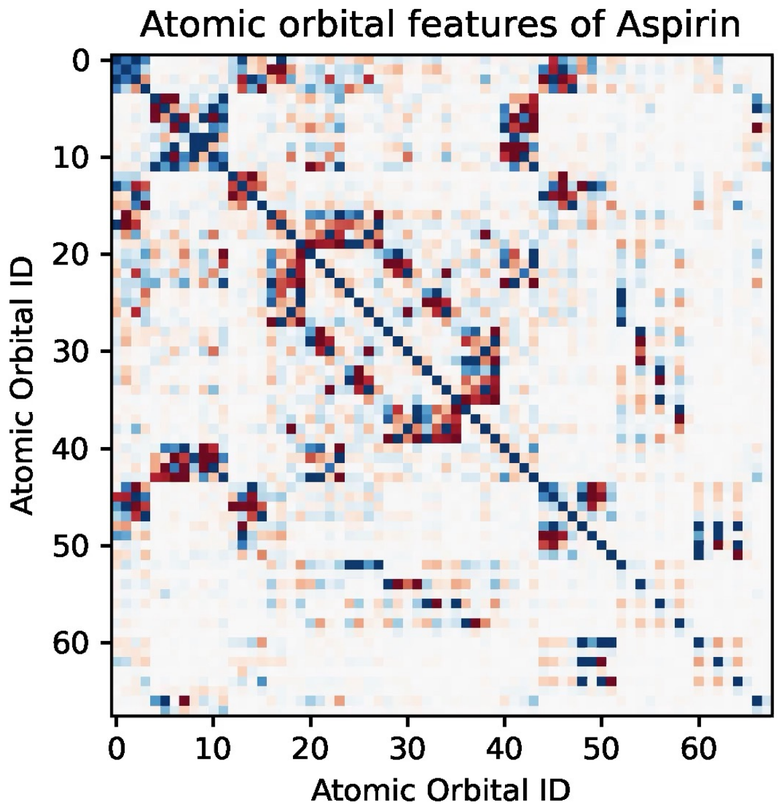


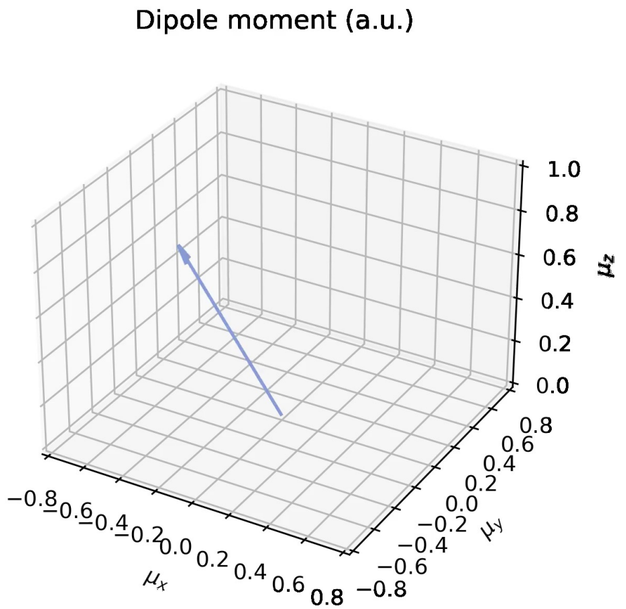
# 3D SYMMETRY IN MOLECULES & ORBITALS

Rotating molecule in 3D





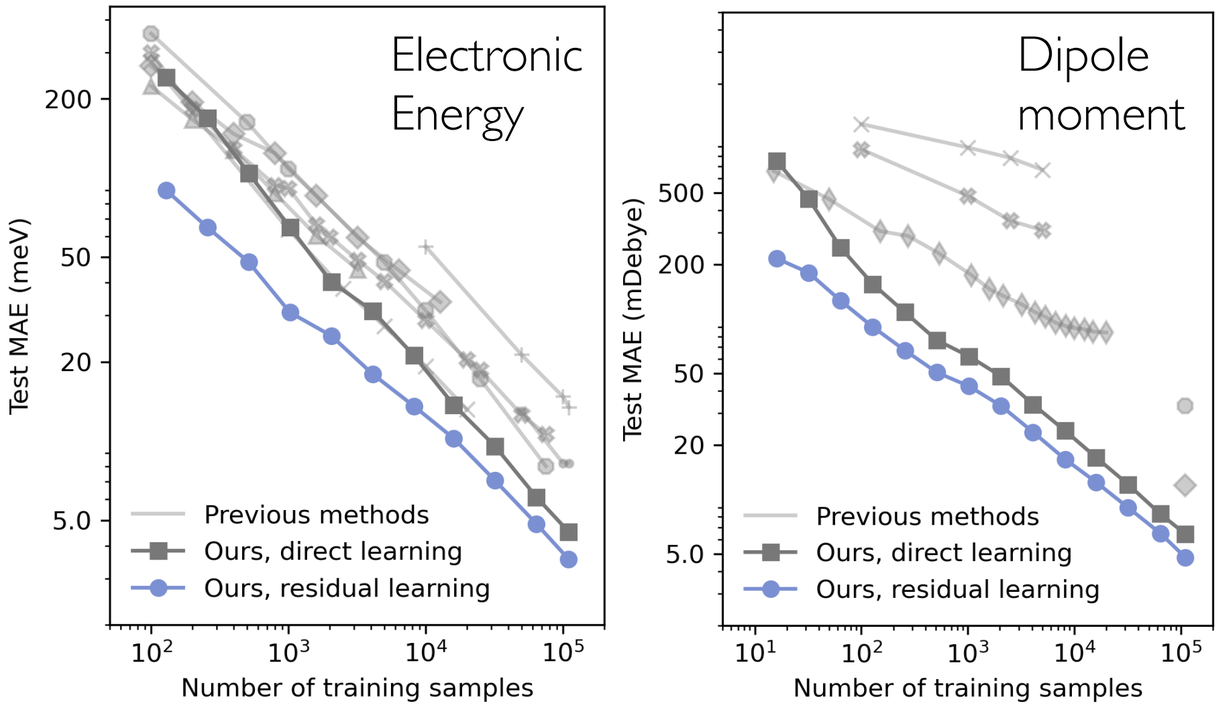




Designing neural network equivariant to continuous symmetries in quantum features

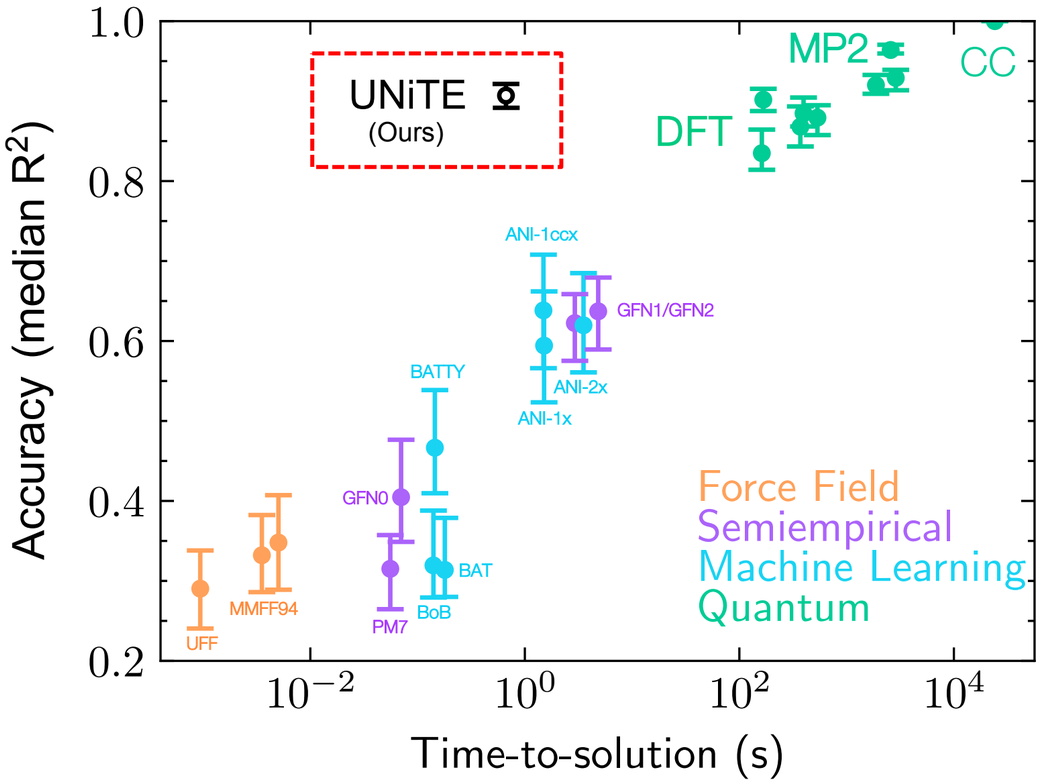
# QUANTUM CHEMISTRY PREDICTIONS

No loss of accuracy + 1000x speedup relative to traditional methods

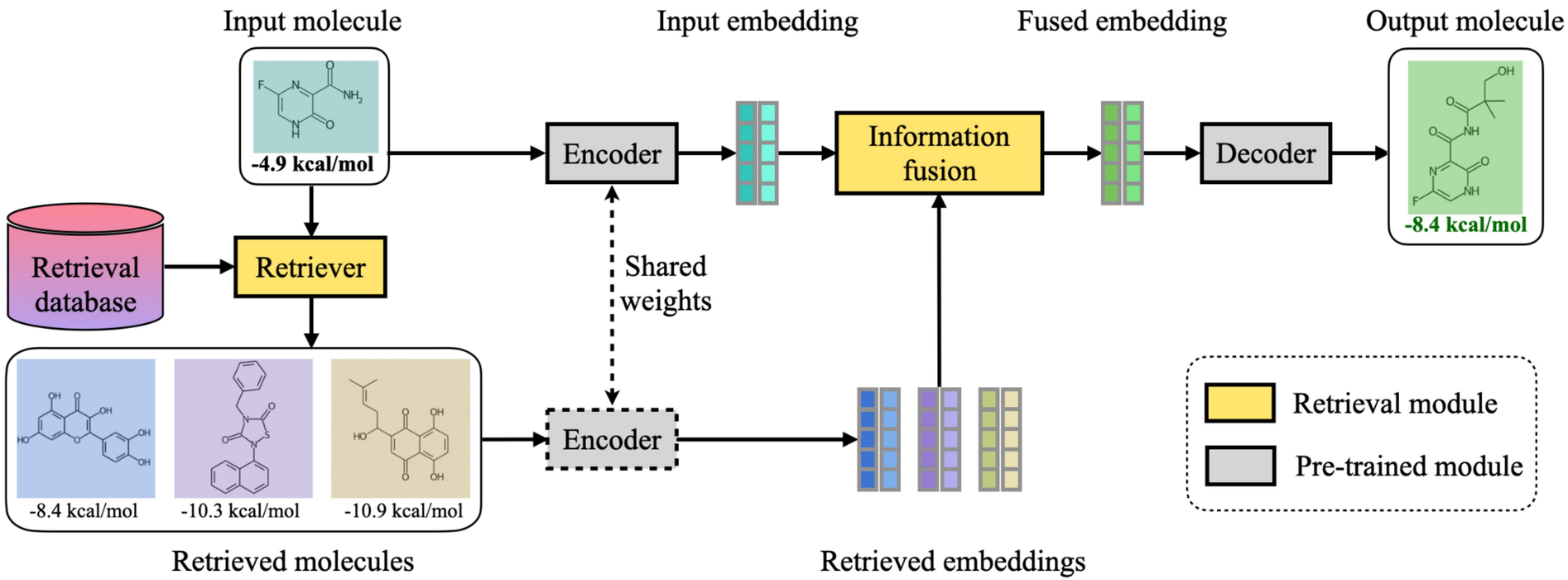
## State-of-the-art data efficiency




## Drug-molecule conformer rankings



# RETMOL: RETRIEVAL-BASED FRAMEWORK FOR MOLECULE GENERATION

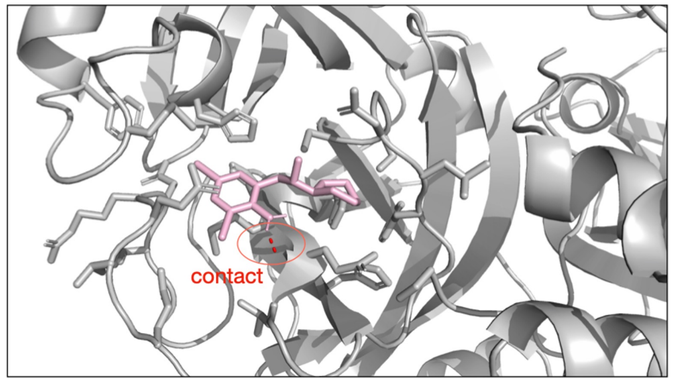



An example of optimizing the binding affinity for an existing potential drug, Favipiravir, for better treating the COVID-19 virus (SARS-CoV-2 main protease, PDB ID: 7L11) under various other design criteria

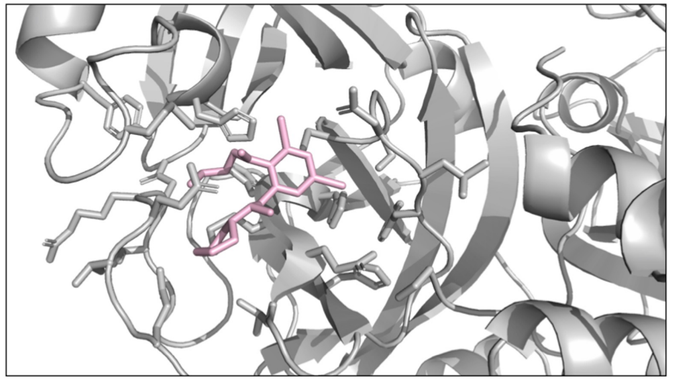
*It plugs in a retrieval module (i.e., retriever and information fusion) into a pre-trained generative model (i.e., the encoder and decoder)*

# OPTIMIZING EXISTING DRUGS FOR SARS-COV-2 MAIN PROTEASE

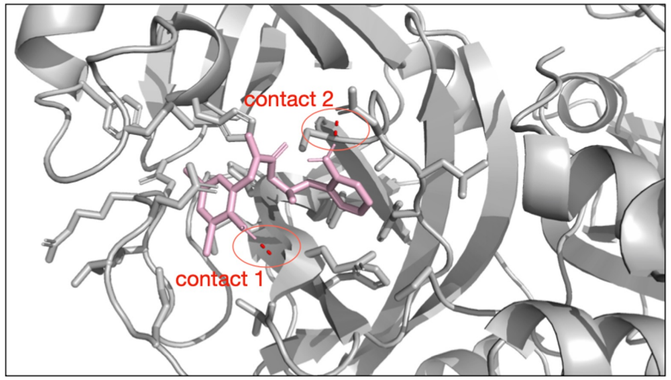
3D visualizations of comparing RetMol with Graph GA in optimizing the original inhibitor, Bromhexine, that binds to the SARS-CoV-2 main protease




Original
(docking score=-9.64)



Graph GA
(docking score=-11.83)

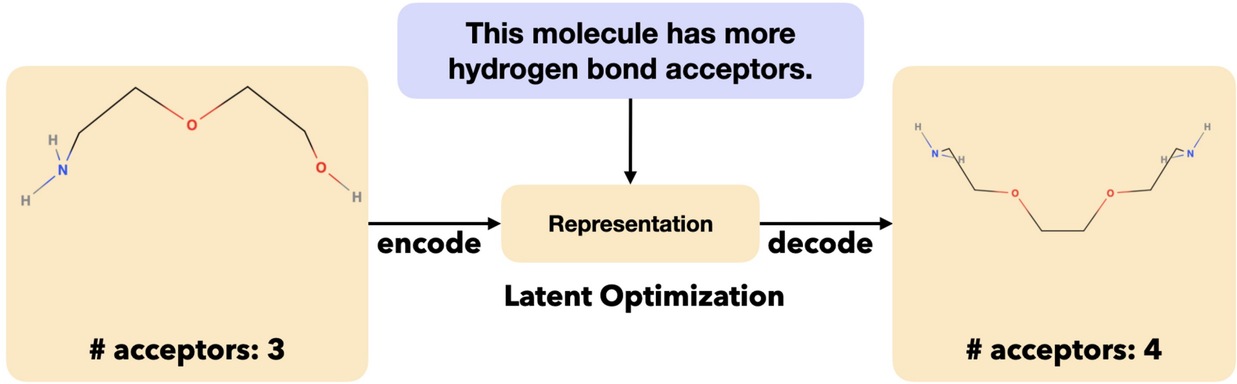


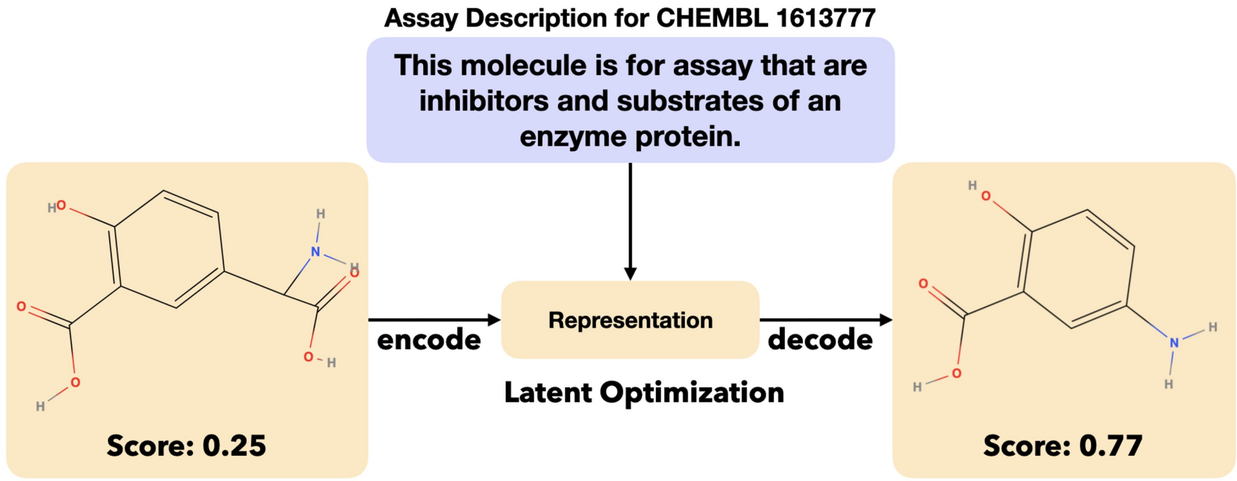

RetMol (Ours)
(docking score=-12.65)

*The optimized inhibitor in RetMol has more polar contacts (red dotted lines) and also more disparate binding modes with the original compound*

# LANGUAGE-GUIDED MOLECULE EDITING

## Editing on Drug Property and Structure-aware Bioactivities

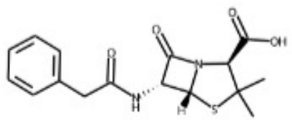
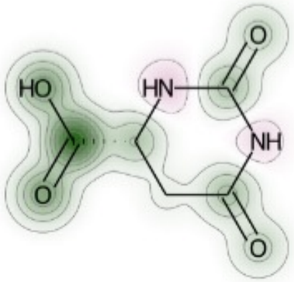
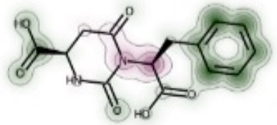
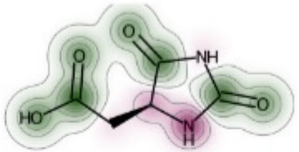
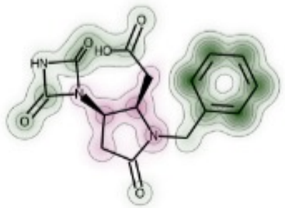
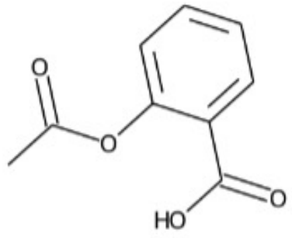
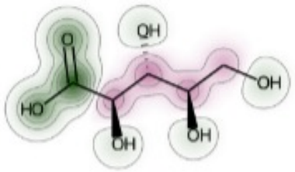
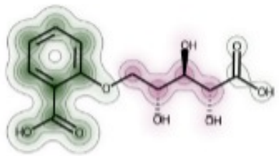
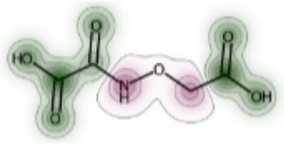
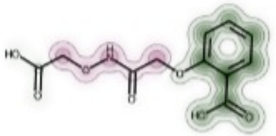


# LANGUAGE-GUIDED MOLECULE EDITING

Editing on Drug Property and Structure-aware Bioactivities

Table 9: Similarity map for editing towards common molecules.

| ground-truth: Penicillin | **Text Prompt: This molecule looks like Penicillin.** | | **Text Prompt: This molecule looks like Penicillin.** | |
|---|---|---|---|---|
| | Input Molecule | Output Molecule | Input Molecule | Output Molecule |
|  |  |  |  |  |
| | Tanimoto: 0.23 | Tanimoto: 0.32 | Tanimoto: 0.16 | Tanimoto: 0.27 |
| ground-truth: Aspirin | **Text Prompt: This molecule looks like Aspirin.** | | **Text Prompt: This molecule looks like Aspirin.** | |
| | Input Molecule | Output Molecule | Input Molecule | Output Molecule |
|  |  |  |  |  |
| | Tanimoto: 0.11 | Tanimoto: 0.43 | Tanimoto: 0.16 | Tanimoto: 0.42 |

# Quantum optimization

# QUANTUM SEMI-DEFINITE PROGRAMMING

SDP: Convex program over symmetric, positive semidefinite matrices

## Classical

$$\begin{aligned}&\text{minimize}_{X\in\mathbb{S}^+} \ \langle W, X\rangle \\ &\text{subject to } \langle A_\mu, X\rangle = b_\mu, \ \ \forall \mu \leq M \\ &X \succeq 0,\end{aligned}$$

## Quantum

$$\begin{aligned}&\text{minimize} \ \ \langle W, \rho\rangle = \text{minimize} \ \ \langle \psi | W | \psi \rangle \\ &\text{subject to } \langle A_\mu, \rho\rangle = b_\mu, \ \ \forall \mu \leq M \\ &\rho \succeq 0 \ \ (\text{by definition})\end{aligned}$$

SDP is naturally implemented in quantum systems:

- Density matrices in quantum systems are inherently positive semidefinite.
- Inner products are quantum expectation values.

# QUANTUM SEMIDEFINITE PROGRAMS

Quantum SDPs hold a lot of promise – solve **$O(2^n)$ variables** with **n** qubits.

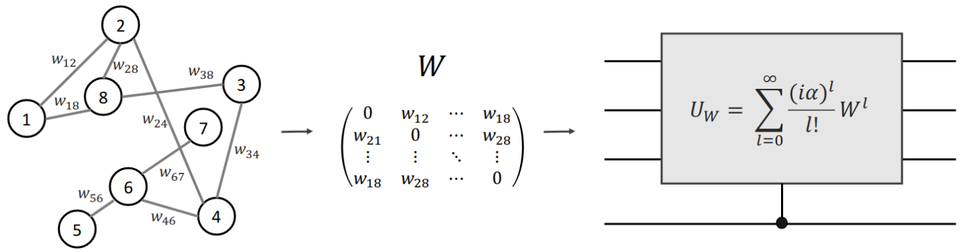
But challenges:

- Gibbs State Sampling: requires a large no. of low-error gates. Far term.
- Existing Variational Q-SDP inefficient: optimizing quantum circuit needs **exponentially** many measurements per epoch, as well as auxiliary classical operations.

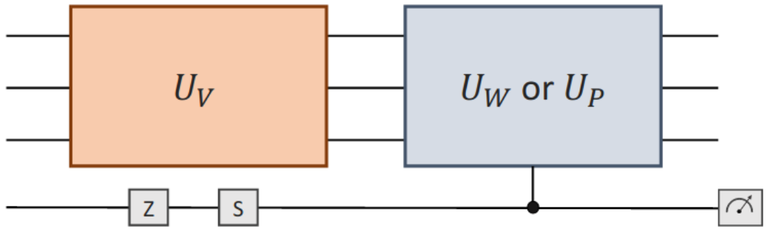
Ours: Approximate Quantum SDP

- Linear number **$O(n)$** of measurements.
- Polynomial **$O(n^2)$** of expectation values.

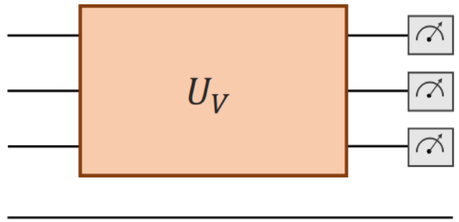
# HTAAC-QSDP (OUR METHOD) - IN A NUTSHELL




Encode O(2ⁿ)-variable objective function (e.g., adjacency matrix) as an n-qubit unitary.



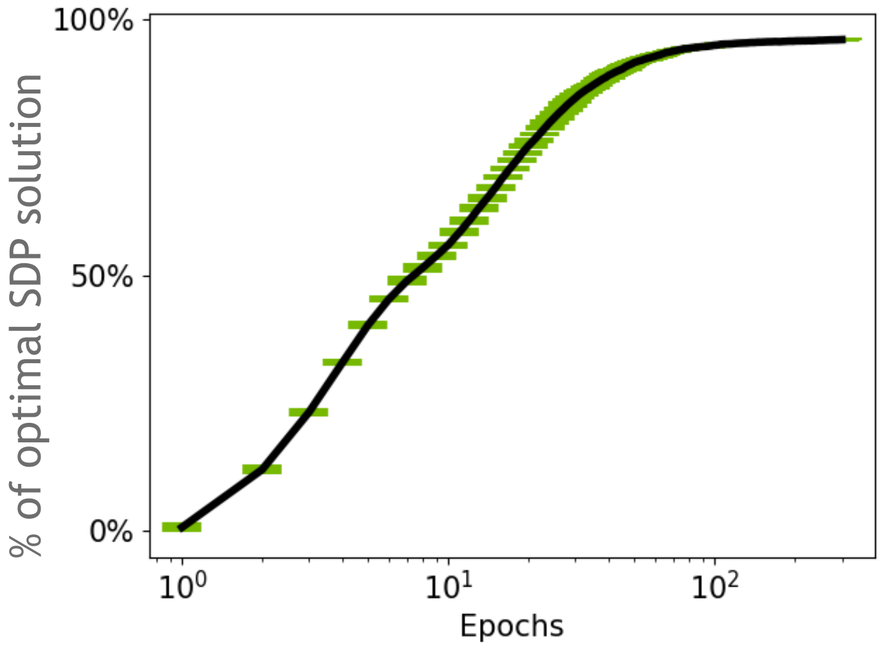

Estimate objective function as a single expectation value on the n+1th (auxiliary) qubit with Hadamard Test.




Approximate O(2ⁿ) variable constraints by calculating O(n²) amplitude constraints (marginal distributions an ensemble).

# 20,000 VERTEX MAXCUT RESULTS WITH NVIDIA CUQUANTUM

- Even though our method is approximate, it is close to exact SDP solver (>96%).
- Our method is efficient: only 15 qubits for 20,000 vertex Maxcut.
- 18x faster with NVIDIA CuQuantum.



# CONCLUSION

- AI4science is the future of science
- Principled algorithms for zero-shot generalization
- Operator learning extends neural networks to learning in infinite dimensional spaces
- Orders of magnitude speedup while maintaining accuracy