Learning Physics, Not Coordinates

Equivariant Geometry-Informed Fourier Neural Operators for 3D PDEs

Sungwon Kim

Graduate School of Data Science, KAIST

Data Science and Artificial Intelligence Lab (DSAIL)

About the Speaker

Why this topic, and where it goes next

KAIST

Sungwon Kim

Ph.D. candidate, KAIST Graduate School of Data Science

Data Science and Artificial Intelligence Lab, advised by Chanyoung Park

Working toward simulation AI that transfers across geometry, a foundation model for physics in the geometric sense

Incoming Visiting Student Researcher, Caltech, with Anima Anandkumar

Selected publications sung-won-kim.github.io

ICML 2026	EqGINO: Equivariant Geometry-Informed Fourier Neural Operators for 3D PDEs
Water Research 2026	Physics-Embedded Graph Neural Operator for Colloidal Aggregation
ICML 2025	Thickness-aware E(3)-Equivariant 3D Mesh Neural Networks
ICLR 2025	Subgraph Federated Learning for Local Generalization **Oral**

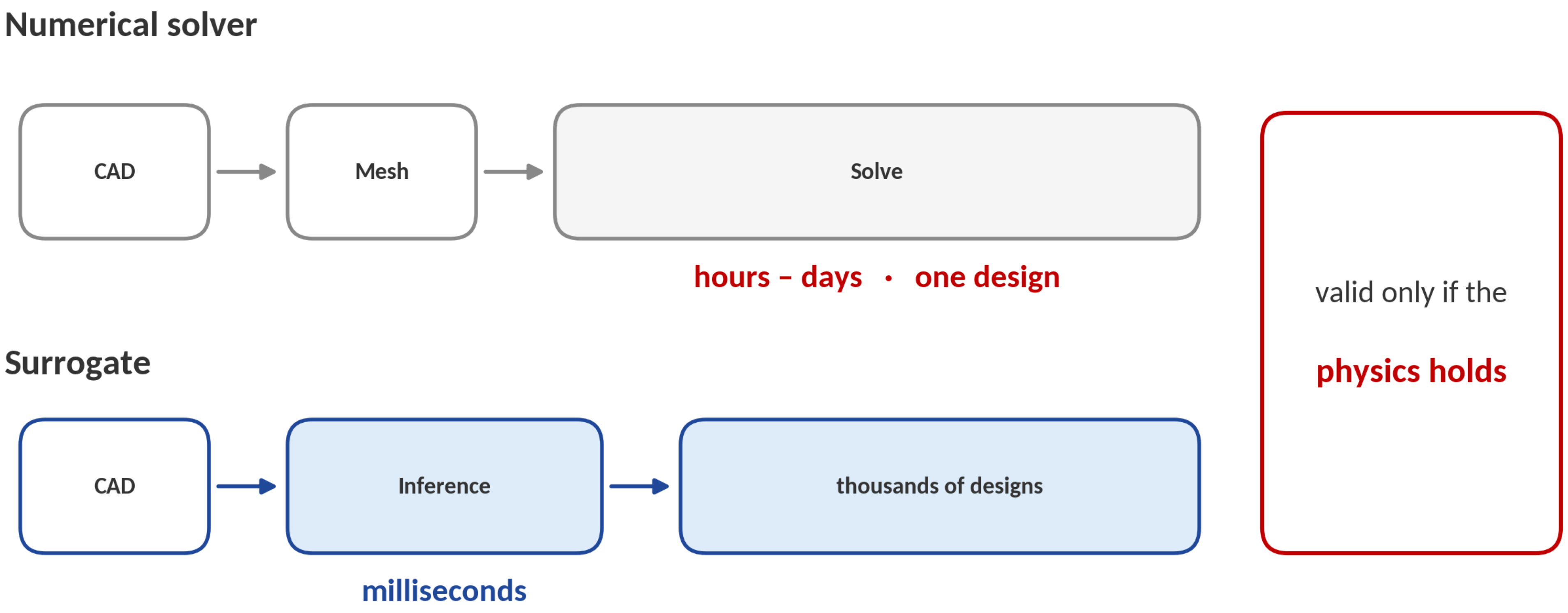
Contents

01 The Coordinate Shortcut

How absolute coordinates break 3D surrogates

Surrogates in the Design Loop

Why solver time is the constraint



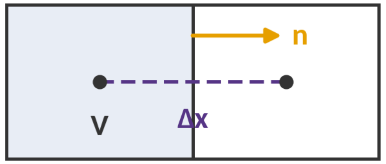
A surrogate replaces a solver only if it satisfies the same governing equations.

Coordinates in a Solver and in a Surrogate

What each one actually consumes

Numerical solver

finite volume RANS, finite element elasticity



face normals, areas, cell volumes, centroid distances

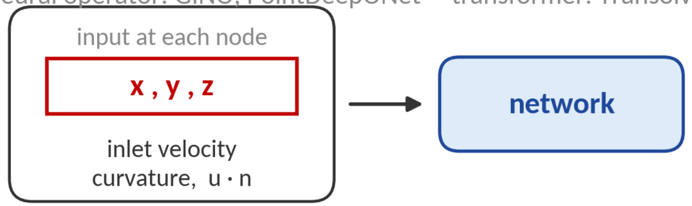
every one of these is a difference

move the origin and nothing changes

Surrogate

point cloud: PointNet, PointNet++ graph: MeshGraphNet

neural operator: GINO, PointDeepONet transformer: Transolver



the position itself is one of the inputs

nothing forces it to use only differences

move the origin and the prediction moves

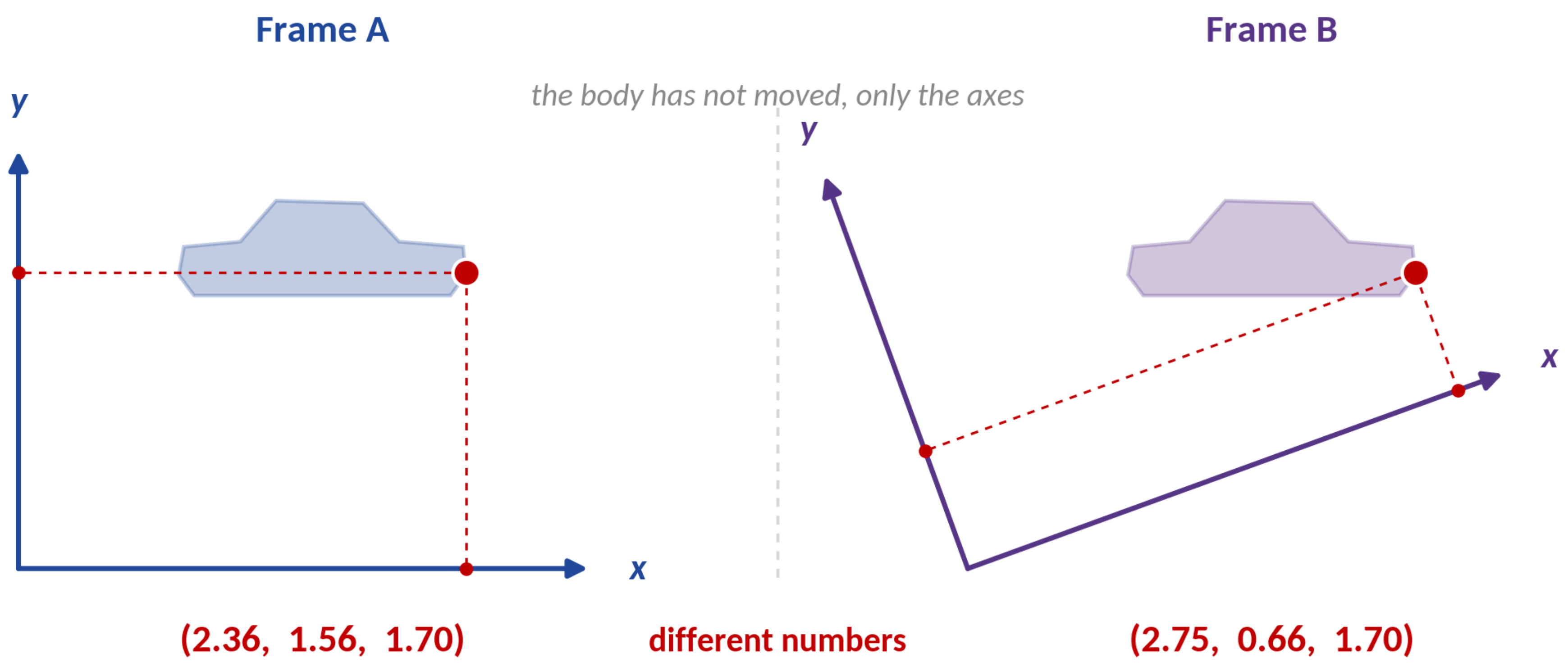
- Every one of these takes the node coordinate as an input feature, next to the physics

→ The absolute triple never reaches a solver, and here it is a network input

A solver never receives an absolute position; the surrogate is handed one as an input feature.

What a Coordinate Actually Is

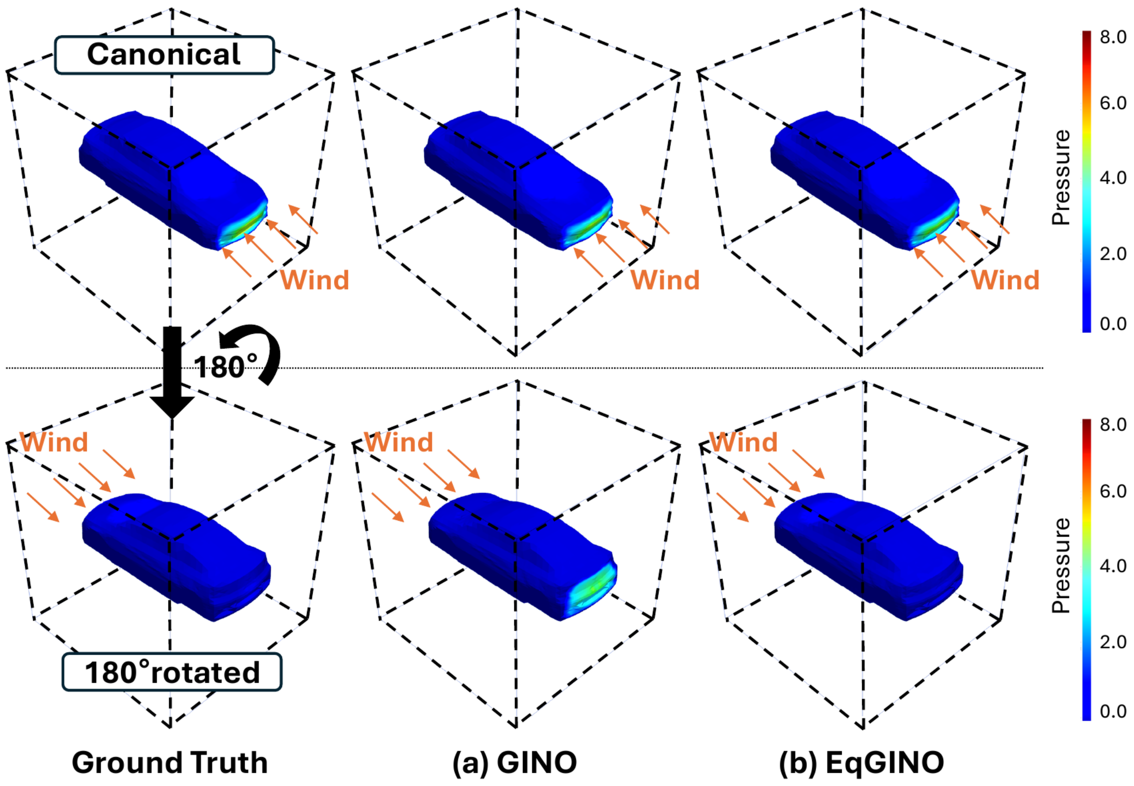
One point on one body, read by two observers



Cartesian coordinates describe the observer's frame, not the physical state.

ShapeNetCar Under a 180° Rotation

ShapeNetCar (CFD), surface pressure, trained on canonical poses only



Top row, canonical

- High pressure on the front bumper, in all three

Bottom row, rotated 180°

- **GINO** moves it to the *rear* bumper, the region the front bumper occupied during training

→ **Predicted from position, not from geometry**

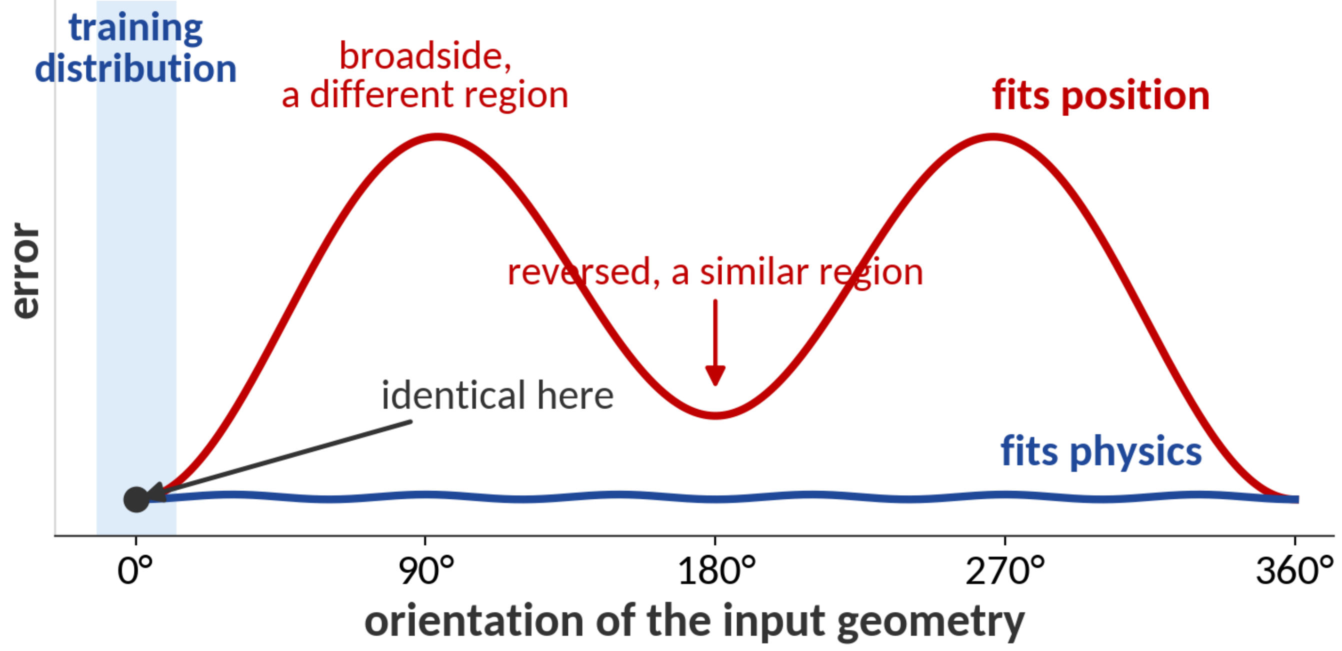
- **EqGINO** keeps it on the true front bumper

Under a 180° rotation GINO places the high-pressure region on the rear bumper.

GINO (Li et al., 2023) · ShapeNetCar (Umetani & Bickel, 2018)

Mechanism of the Failure

Why the error is structural, not a capacity problem



schematic, not measured

Two hypotheses

- Both fit the training poses equally well
- Only one of them is physics

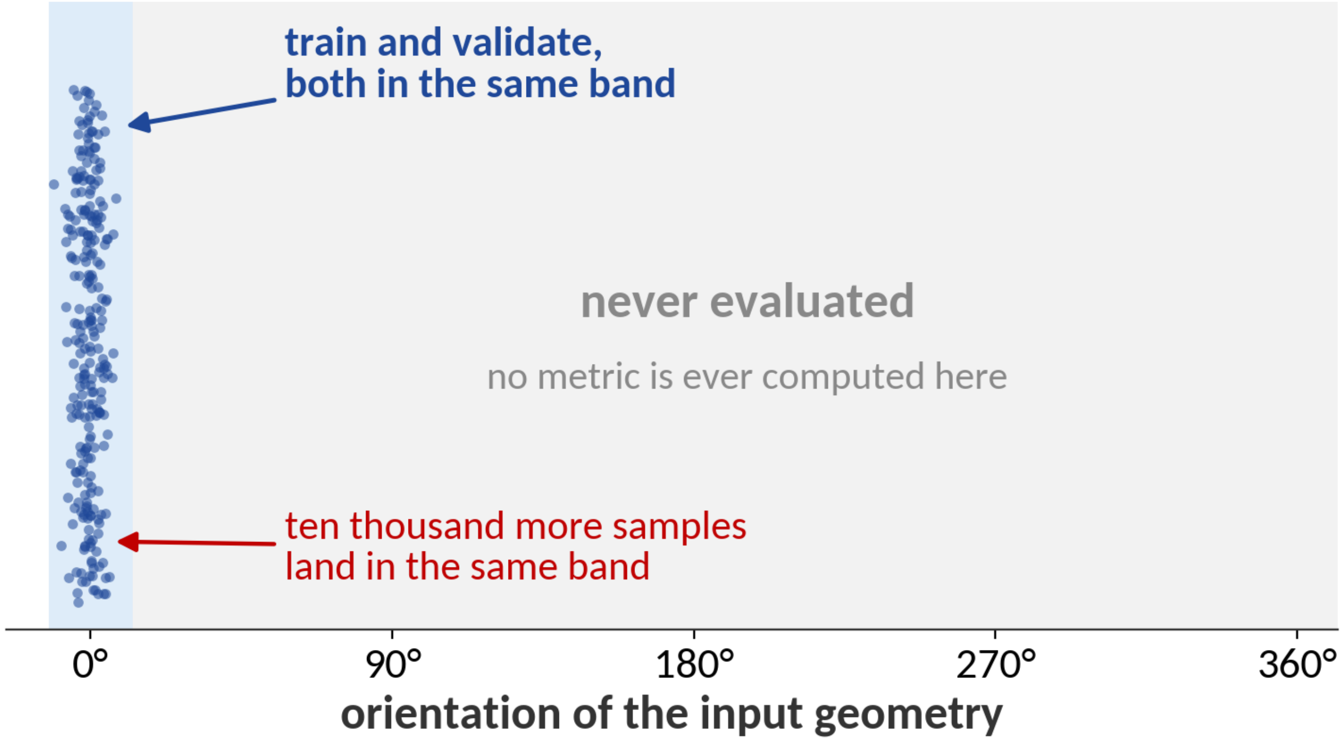
Consequence

→ **Nothing in the training data separates them**

The training distribution does not penalize the positional shortcut, so it is learned.

Why More Data Does Not Help

Training and validation sit at the same orientation



Scale

- Every canonical sample reinforces the shortcut
- Larger models learn it faster, not less

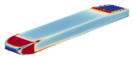
Measurement

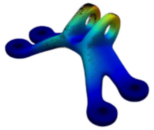
→ **The benchmark cannot expose this defect**

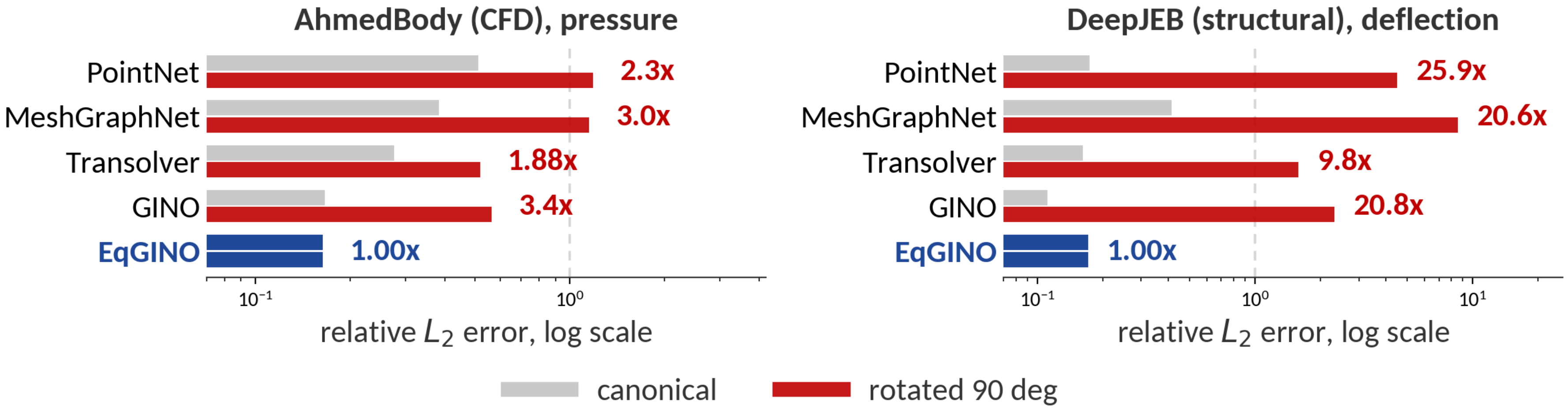
A metric computed at one orientation cannot detect a dependence on orientation.

Degradation Across Model Families

Relative L_2 error, trained canonical, tested under 90° multiples







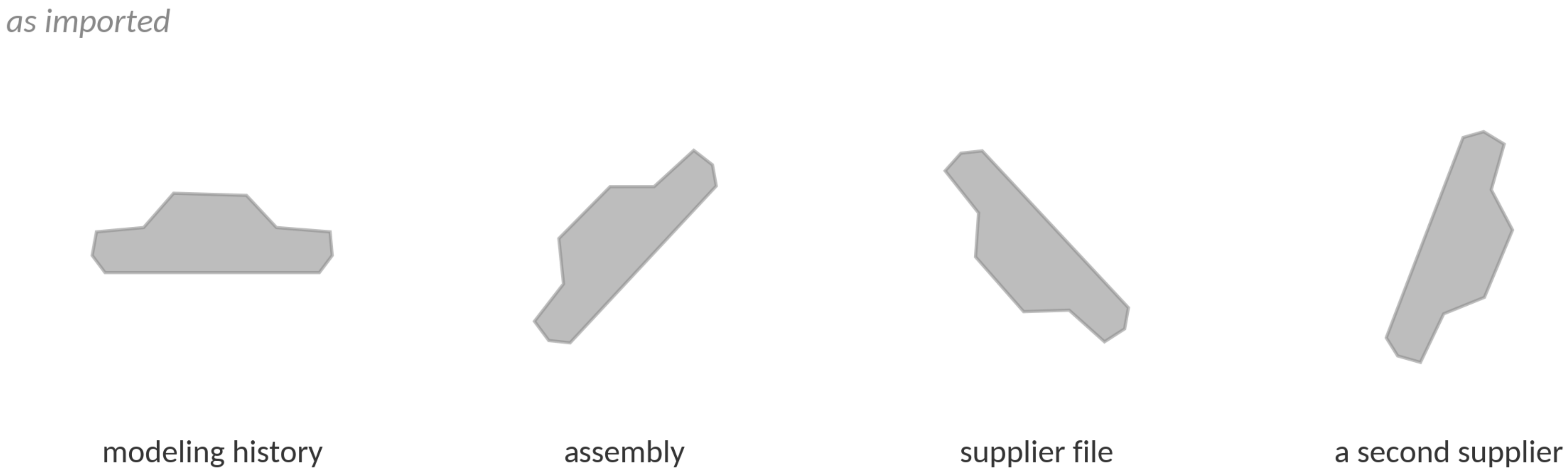
→ **EqGINO is unchanged in both panels, and lowest of all five on AhmedBody pressure**

Every non-equivariant model degrades under rotation, by 1.9x on AhmedBody pressure and up to 25.9x on DeepJEB deflection.

PointNet (Qi et al., 2017) · MeshGraphNet (Pfaff et al., 2020) · Transolver (Wu et al., 2024) · GINO (Li et al., 2023) · AhmedBody (Ahmed et al., 1984) · DeepJEB (Hong et al., 2025)

Orientation in a CAE Pipeline

Orientation is not controlled in a real CAE pipeline



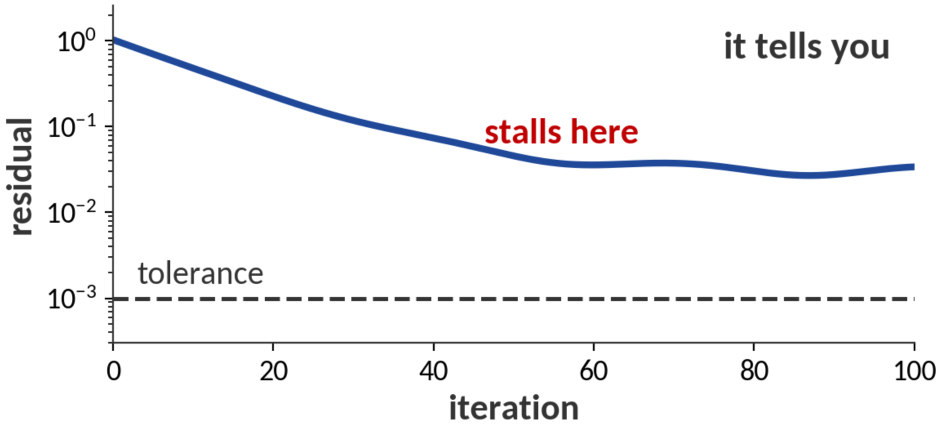
orientation follows the CAD history, not your training set

Nothing upstream of the model guarantees a consistent orientation.

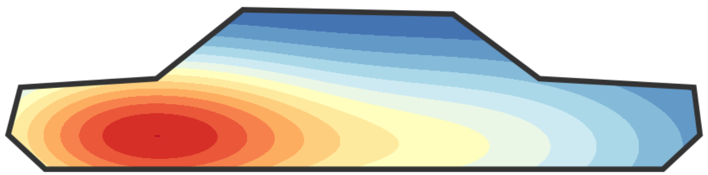
Failure Without a Warning

A solver reports its failure; a surrogate does not

a numerical solver



a neural surrogate



no warning, no divergence
smooth, plausible, and wrong

- No surrogate reports its own error. With coordinate dependence the trigger is the pose
- And the pose is the one thing your test set never changes

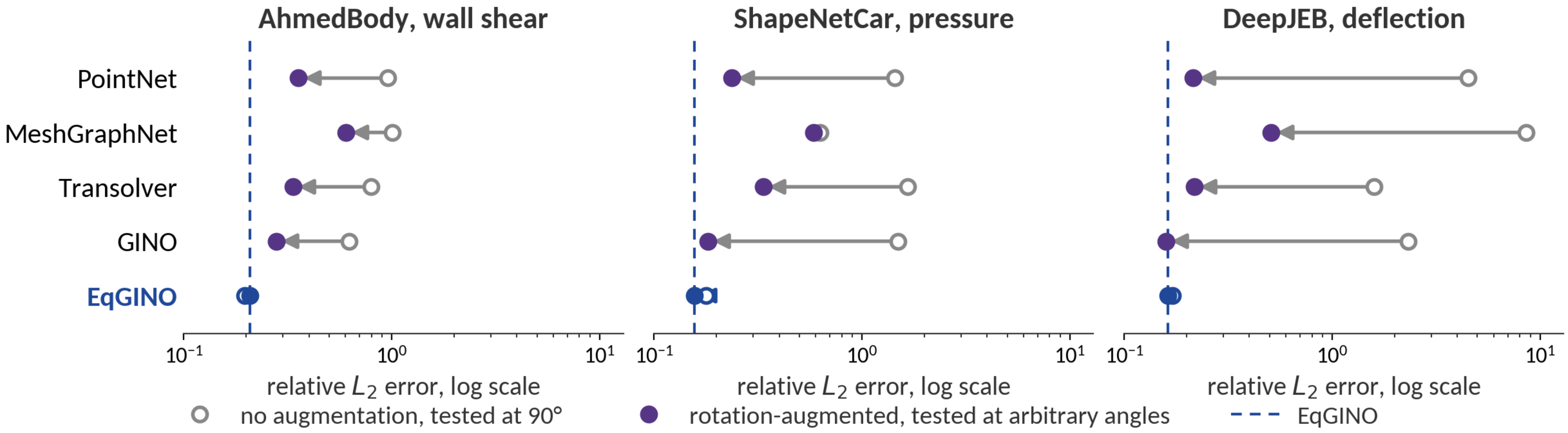
→ The usual fix is one alignment step, on every query, that is never allowed to be wrong

Canonical alignment moves the correctness requirement from the model to the user.

Augmentation as a Substitute

Every model, ours included, trained and tested on continuous SE(3) rotations

the same shape at random orientations, added to the training set of every model here, including ours



Rotation augmentation reduces the gap but provides no guarantee.

What Part 1 Establishes

Why this cannot be tuned away

does not remove it

- ✗ more data
- ✗ a bigger model
- ✗ rotation augmentation
- ✗ your in-distribution metric

what is left

→ **keep the coordinate out of the input**

→ **build layers that do not need it**

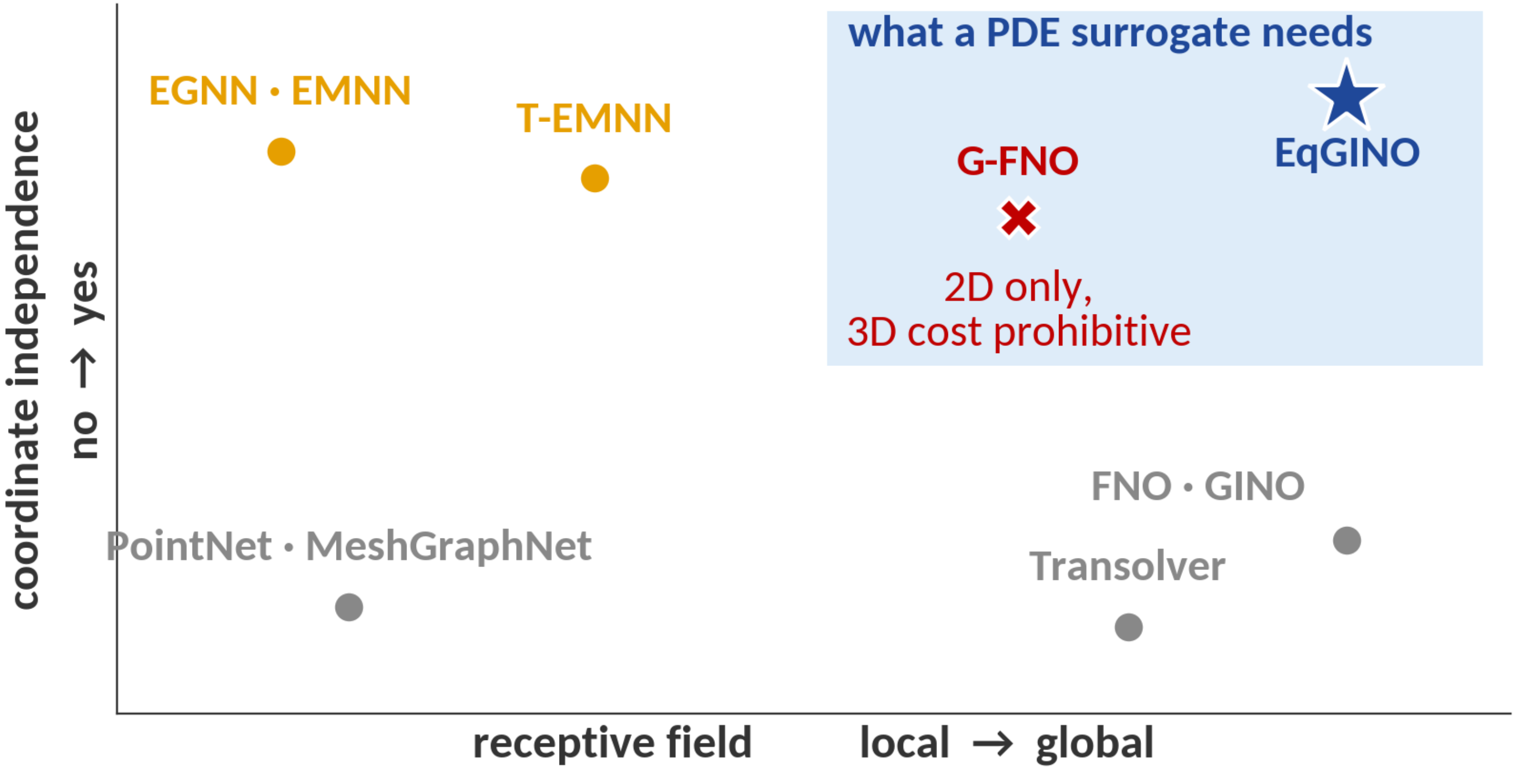
A model that can see the coordinate frame will fit to it, so the frame must not reach it.

02 Equivariance or Global Coupling

The existing routes give one, and a PDE needs both

Two Established Routes

Both are well developed, neither is sufficient alone

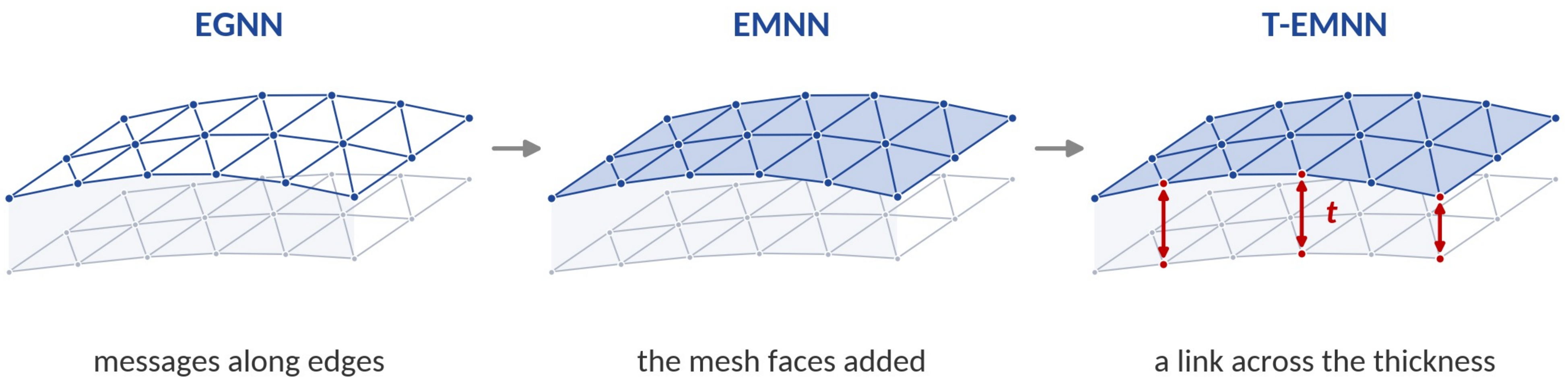


Equivariance and global context have been developed separately; PDE surrogates need both together.

EGNN (Satorras et al., 2021) · EMNN (Trang et al., 2024) · T-EMNN (Kim et al., 2025) · PointNet (Qi et al., 2017) · MeshGraphNet (Pfaff et al., 2020) · FNO (Li et al., 2020) · GINO (Li et al., 2023) · G-FNO (Helwig et al., 2023) · Transolver (Wu et al., 2024)

Route A: Equivariant Message Passing

One line of work, and what it gets right



- Features built only from invariants, so the model never sees the frame

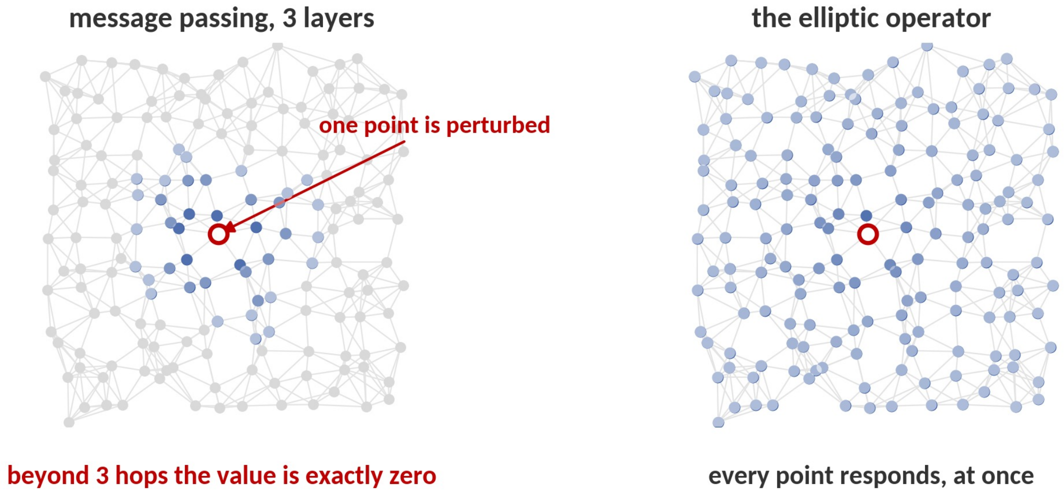
→ **Equivariance is exact, and it costs nothing at inference**

Equivariance can be built into message passing exactly, and this line of work has done it.

EGNN (Satorras et al., 2021) · EMNN (Trang et al., 2024) · T-EMNN (Kim et al., 2025)

Where the Reach Stops

Why the limit is a physics problem, not a compute problem



- One hop per layer, so the reach grows only with depth; deeper networks oversmooth
- Pressure is elliptic, so a solver settles it with a global solve at every step

→ Gain: equivariance, exactly. Cost: a reach that stops where the physics does not

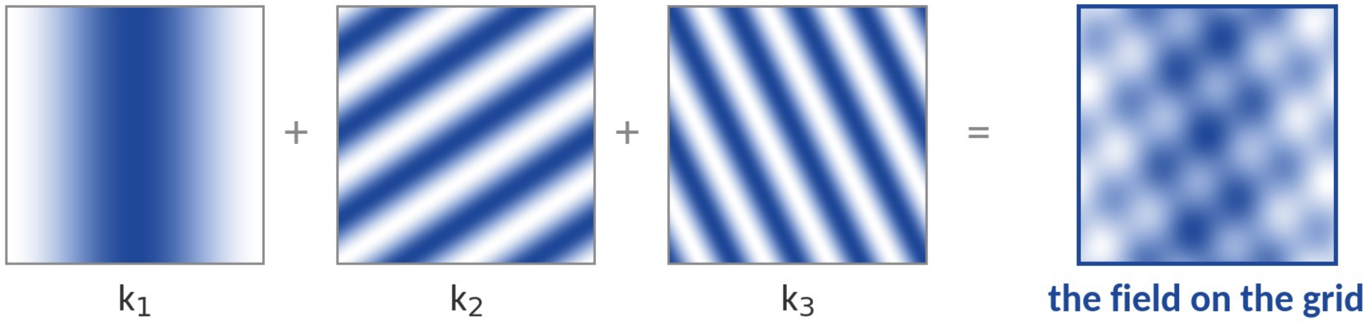
Local message passing cannot represent the instantaneous global coupling that PDE dynamics require.

oversmoothing (Li et al., 2018) · over-squashing (Alon & Yahav, 2020)

Route B: Spectral Operators

The receptive field an elliptic operator needs

Every Fourier mode spans the whole domain



change one coefficient
and every point moves

- One transform couples every point with every other, inside a single layer
- **FNO** on regular grids, **GINO** on irregular geometry

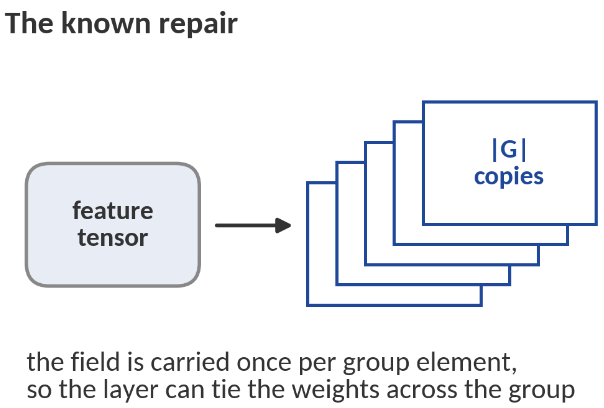
→ **The reach is solved. What is missing is the symmetry**

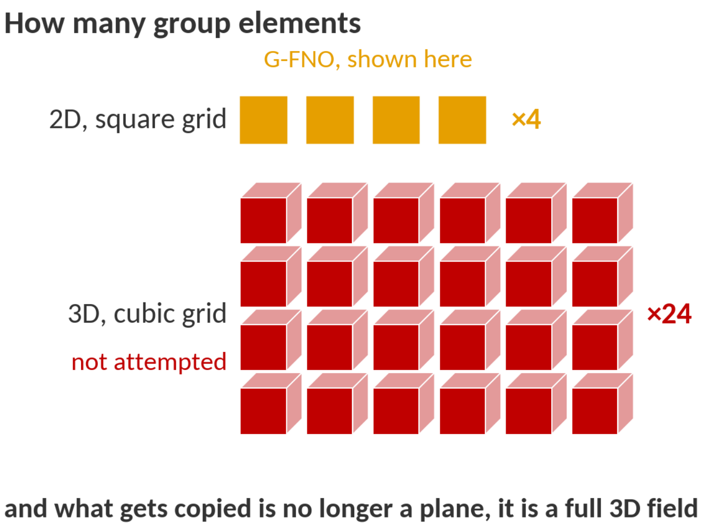
The spectral route supplies exactly the receptive field that local message passing cannot.

FNO (Li et al., 2020) · GINO (Li et al., 2023)

The Price of Making It Equivariant

Group convolution, and why it does not fit in 3D





- Twenty-four rotations of the cubic grid, forty-eight once reflections count

→ **Every one of them needs its own copy of a full 3D field, in every layer**

The spectral route has the receptive field a PDE needs, and the only known way to make it equivariant does not fit in 3D.

G-FNO (Helwig et al., 2023)

What Part 2 Establishes

One property each, and a PDE surrogate needs both

	Route A	Route B	Required
coordinate independence	●	○	●
global reach	○	●	●

Route A equivariant message passing, and the reach grows only with depth

Route B spectral operators, and the known repair copies the field twenty-four times

Each route supplies one of the two properties a PDE surrogate needs, and neither supplies both.

03 EqGINO

Equivariance enforced inside the spectral domain

Design Requirements

What the architecture has to satisfy simultaneously

1. No absolute coordinates anywhere in the path

- Encoder, spectral layers, decoder, one dependent module breaks the whole chain

2. Global receptive field preserved

- Equivariance must not be bought with locality

3. Baseline-level cost, on unstructured geometry

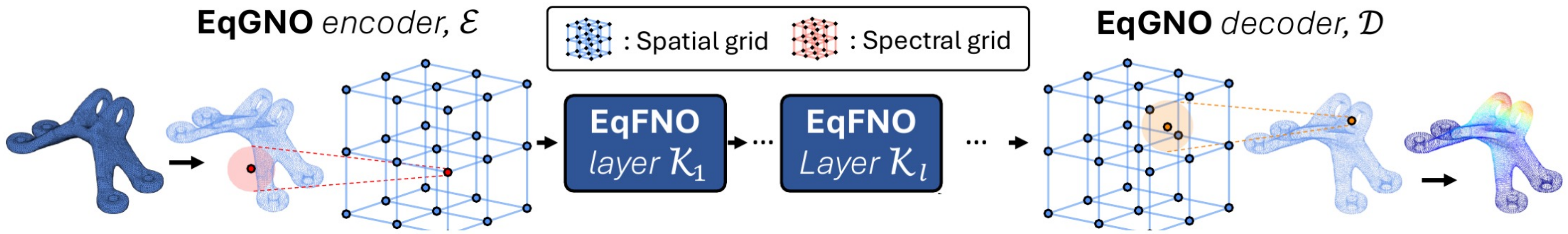
- No group convolution; yet the FFT needs the regular grid that real meshes lack

→ **Equivariance is a chain property, so every stage has to hold on its own**

End-to-end equivariance holds only if every stage is equivariant independently.

Architecture

Encode to a grid, propagate in the spectral domain, decode to the mesh



	GINO	EqGINO
encoder and decoder	kernel reads x, y	**kernel reads ‖x − y‖, ‖x − ȳ‖**
spectral layers	a free W(k) per mode	**W(Rk) = W(k), shared on orbits**

the three-stage skeleton itself is unchanged

→ **Same skeleton as GINO; both learnable units now read only rotation-invariant quantities**

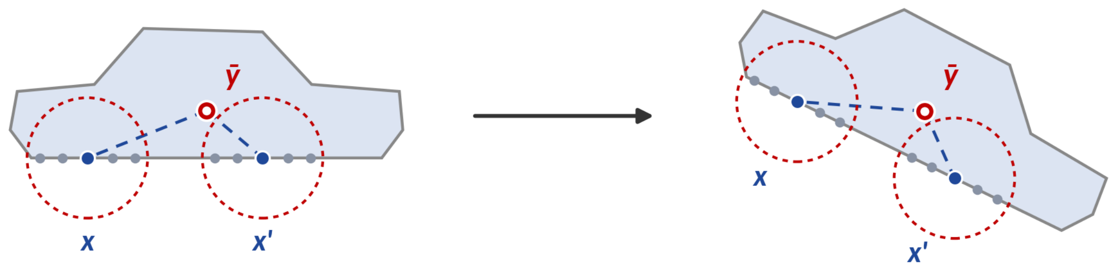
The structure follows GINO; the contribution is that each module is rebuilt to preserve symmetry.

GINO (Li et al., 2023)

EqGNO: Distances Instead of Positions

Making the mesh-to-grid transfer coordinate-free

canonical pose | rotated



the same local patch at both points,
so the neighbour distances agree

the body turns, the centroid turns with it,
and every number stays the same

red circle: the neighbourhood. Congruent, so $\|x - y\|$ cannot separate x from x'

dashed line: $\|x - \bar{y}\|$, which does separate them

- **GNO** reads x^{grid} and y_j themselves, so a rotated pose gives different latent features
- **EqGNO** reads $\kappa\left(\|x^{\text{grid}} - y_j\| ,\ \|x^{\text{grid}} - \bar{y}\| \right)$, two scalars, no position

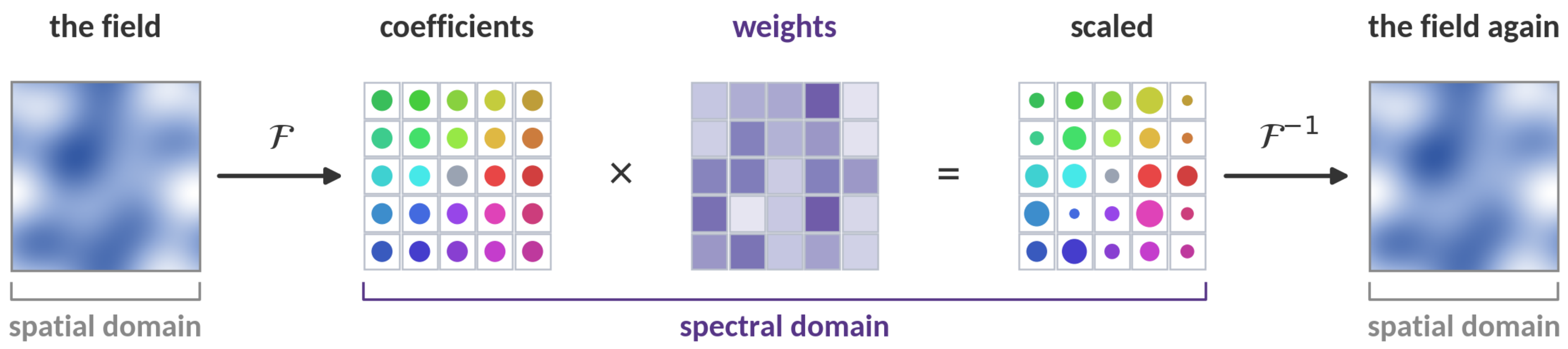
→ Rotation-invariant input, so encoder and decoder inherit equivariance

Replacing positions with distances to neighbors and to the centroid keeps the geometry and drops the frame.

GNO (Li et al., 2020)

Inside a Spectral Layer

Where the learnable weight actually sits



this multiplication is the FNO layer

$(Lv)(x) = F^{-1}[\, W(\mathbf{k}) \cdot (Fv)(\mathbf{k}) \,](x)$, one $W(\mathbf{k})$ per Fourier mode

→ **The weight belongs to the lattice slot, not to the coefficient**

A spectral layer multiplies every Fourier coefficient by a weight that belongs to its lattice slot.

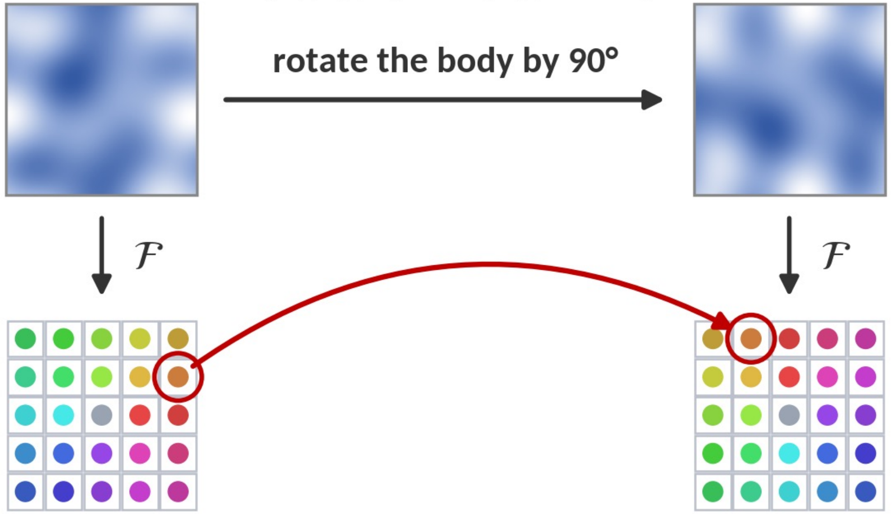
FNO (Li et al., 2020)

Rotation Moves the Coefficients

Why a rotation of the body becomes a reshuffle of the spectrum

$$\mathcal{F}[\mathcal{T}_R f](\mathbf{k}) = \mathcal{F}[f](R^{-1}\mathbf{k})$$

spatial domain



spectral domain

the same colours, turned by the same 90°

the coefficients keep their values and only change place

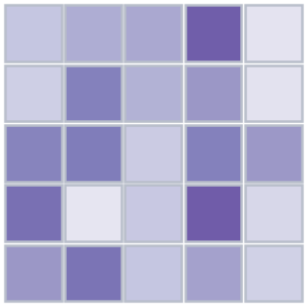
→ A rotation in space is a reshuffle of the lattice slots, and nothing here is learned

Rotation acts on the spectrum by moving coefficients between lattice slots without changing their values.

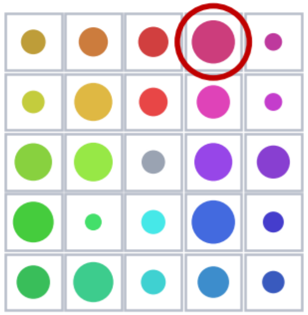
Where FNO Loses Equivariance

The coefficients moved, the weights did not

the weights never move

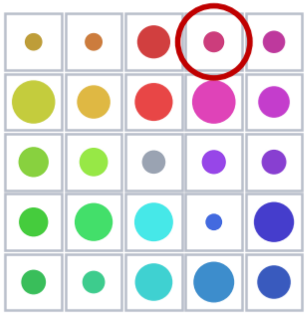


so the same coefficient meets a different weight



the layer applied to the rotated input

≠



the original output, rotated

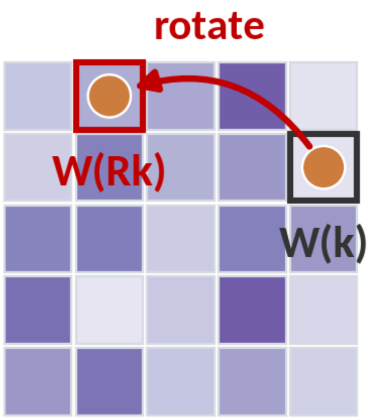
→ The layer applied to the rotated input is not the rotated output

Weights tied to lattice slots discard the rotation that the Fourier transform carried correctly.

FNO (Li et al., 2020)

One Coefficient, Two Weights

What would have to be true for the output to follow



one coefficient, two different weights

the output can only follow the rotation
if those two weights are the same number

$$W(R\mathbf{k}) = W(\mathbf{k})$$

- Before the rotation it meets $W(\mathbf{k})$, after it $W(R\mathbf{k})$, and the two are unrelated numbers

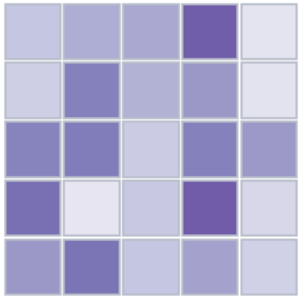
→ **One requirement, for every mode and every rotation, and the whole method comes from it**

The output can only follow the rotation if the weight at the rotated mode equals the weight at the original mode.

From Rotation to Isotropy

The condition for a spectral layer to commute with rotation

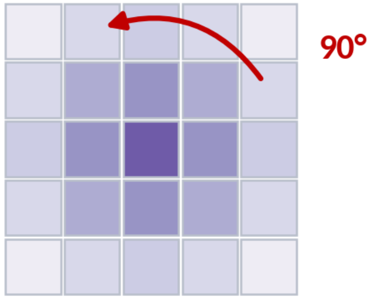
vanilla FNO



the weight may differ
in every direction

→

EqFNO



the weight may depend on ||k||
and on nothing else

$$W(R\mathbf{k}) = W(\mathbf{k})$$

- Require the layer to commute with rotation, and exactly one condition survives

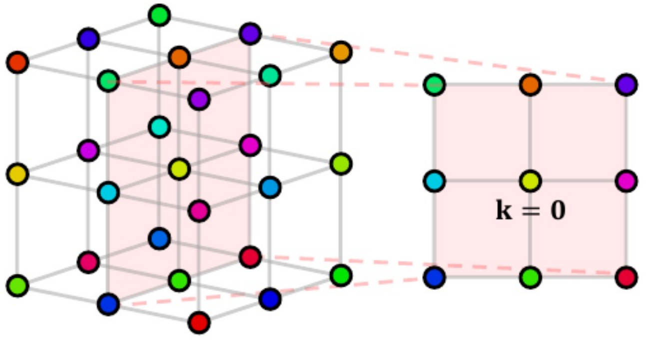
→ The weight may depend on $\|k\|_2$, never on the direction of k

Rotation equivariance reduces to a single requirement: the spectral weight depends only on frequency magnitude.

Orbit-Based Weight Sharing

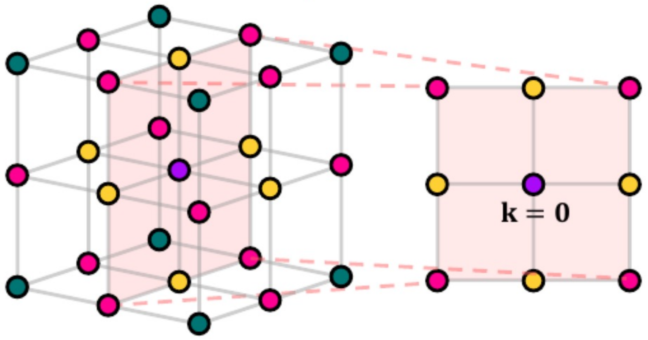
Reading the spectral lattice

Vanilla FNO



27 modes, 27 independent weights

EqFNO



27 modes, 4 weights

one dot is one Fourier mode k

the pulled-out square is the $k_z = 0$ slice

same colour is one shared weight for the whole shell ‖k‖, a complex 64 × 64 matrix held block-diagonal over G groups

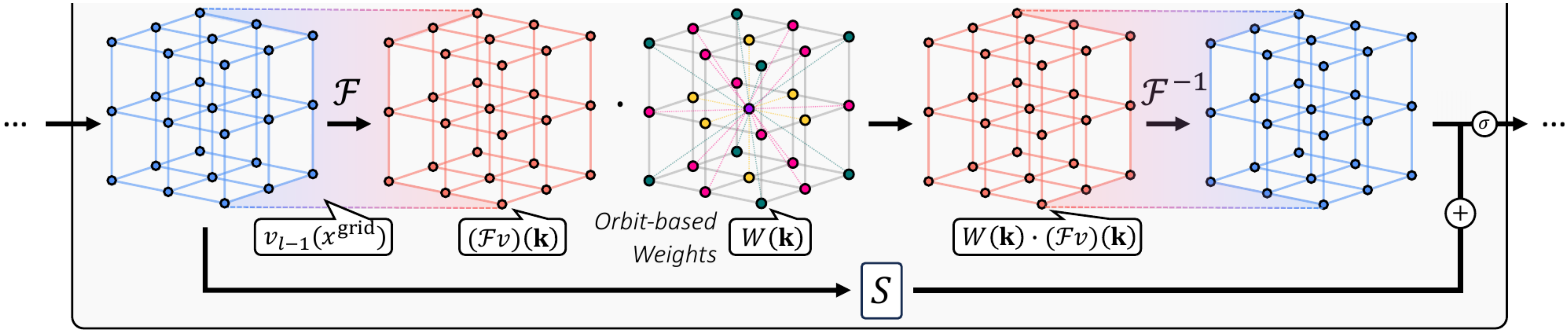
- In the trained model a 32^3 grid has 32,768 modes, sharing 464 orbit weights

→ Spectral parameter complexity falls from $O(K^3)$ to $O(K)$

Sharing one weight per radial orbit enforces the isotropy condition by construction.

Isotropic Spectral Convolution

One EqFNO layer, from grid features back to grid features



- Transform *F*, multiply by orbit-shared *W*(**k**), inverse transform

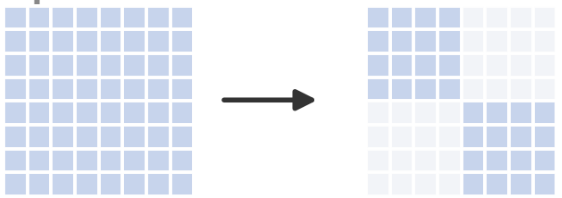
→ Every grid point interacts with every other one inside a single layer

The layer keeps FNO's global coupling and changes only how the spectral weights are tied.

What Weight Sharing Costs

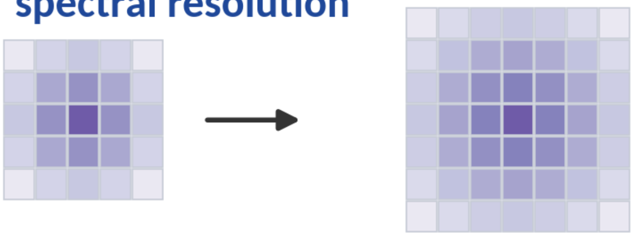
Trading channel interaction for spectral resolution

given up: channel interaction



one dense mix *G* groups, no mixing across

bought: spectral resolution



$K = 32$ $K = 40$, more shells

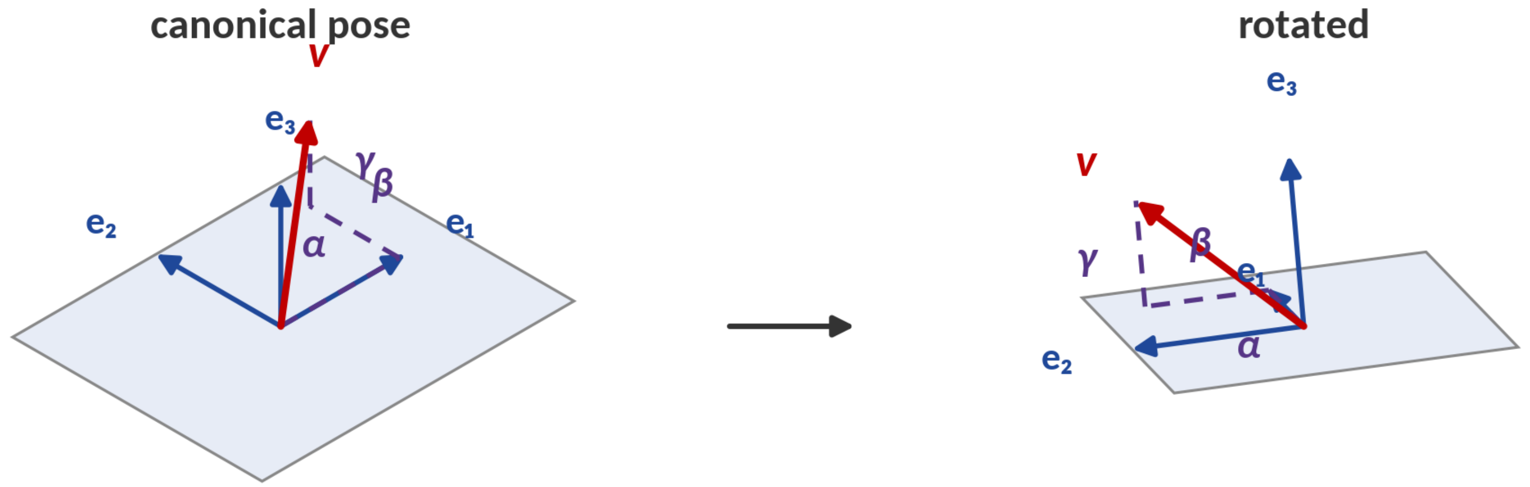
	parameters	Ahmed wall shear	Ahmed pressure	ShapeNetCar pressure
GINO, $K = 32$	285 M	0.1987	0.1666	0.1610
EqGINO, $K = 32$, $G = 2$	about 4 M	0.2078	0.1737	0.1878
EqGINO, $K = 40$, $G = 4$	**about 4 M**	**0.1958**	**0.1637**	**0.1773**

→ **Capacity restored, and the parameter count does not go up**

Restricting channel mixing pays for a finer spectral grid, which restores the capacity that orbit sharing removed.

Vector Fields via an Equivariant Local Basis

Extending an isotropic scalar backbone to deflection and wall shear stress



e_1 along the inflow, or the applied load
e_2 and e_3 from the geometry

the basis turns with the body,
and α, β, γ do not change

$\mathbf{v} = \alpha\,\mathbf{e}_1 + \beta\,\mathbf{e}_2 + \gamma\,\mathbf{e}_3$ — **the network predicts only the three scalars**

→ **Vector prediction reduced to scalar regression, backbone unchanged**

A geometry-derived local basis lets a scalar-isotropic backbone predict vector fields without losing equivariance.

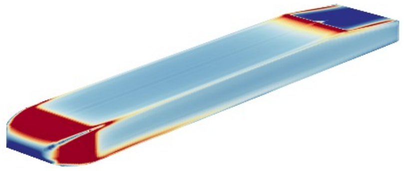
04

Testing the Claim

Three benchmarks, three rotation protocols, and what the numbers show

Three Benchmarks

Two fluid dynamics problems and one structural mechanics problem

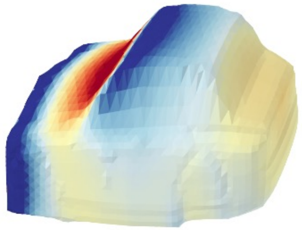


AhmedBody

CFD, turbulent external flow

wall shear stress (vector) + 4 scalars

train / val / test 413 / 45 / 50

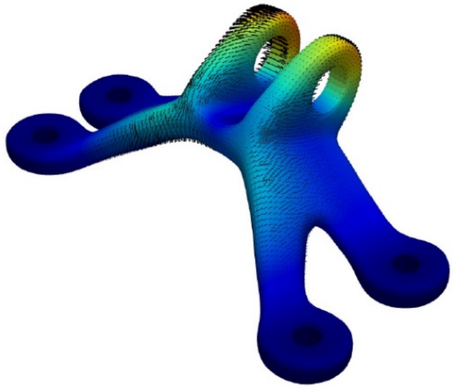


ShapeNetCar

CFD, 20 m/s, Re 5×10^6

surface pressure (scalar)

train / val / test 450 / 50 / 100



DeepJEB

linear static elasticity

deflection (vector), von Mises stress

train / val / test 1,776 / 162 / 197

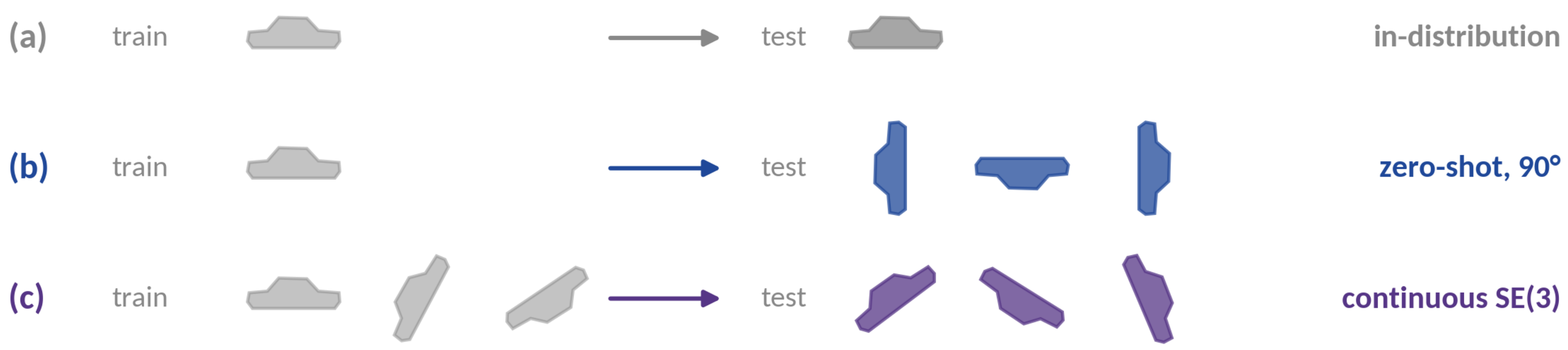
→ Two vector targets and three scalar targets, across fluids and structures

Two of the three are fluids and one is solid mechanics, so the claim is not about one physics domain.

AhmedBody (Ahmed et al., 1984) · ShapeNetCar (Umetani & Bickel, 2018) · ShapeNet (Chang et al., 2015) · DeepJEB (Hong et al., 2025) · SimJEB (Whalen et al., 2021)

Evaluation Protocols

Three rotation settings, one metric

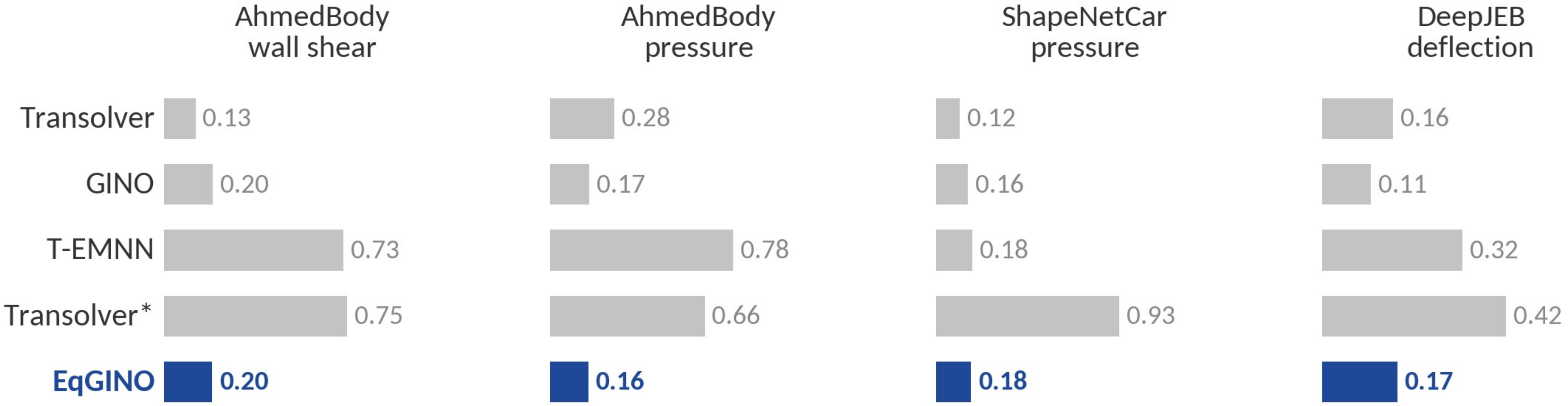


→ Rotation applied to the whole problem: geometry, inlet velocity, vector fields

One metric throughout, relative L2 error, across three benchmarks and three orientation settings.

In-Distribution Performance

Protocol (a) canonical → canonical, relative L_2 error, lower is better



- Relative L_2 error, lower is better, on four targets across the three benchmarks

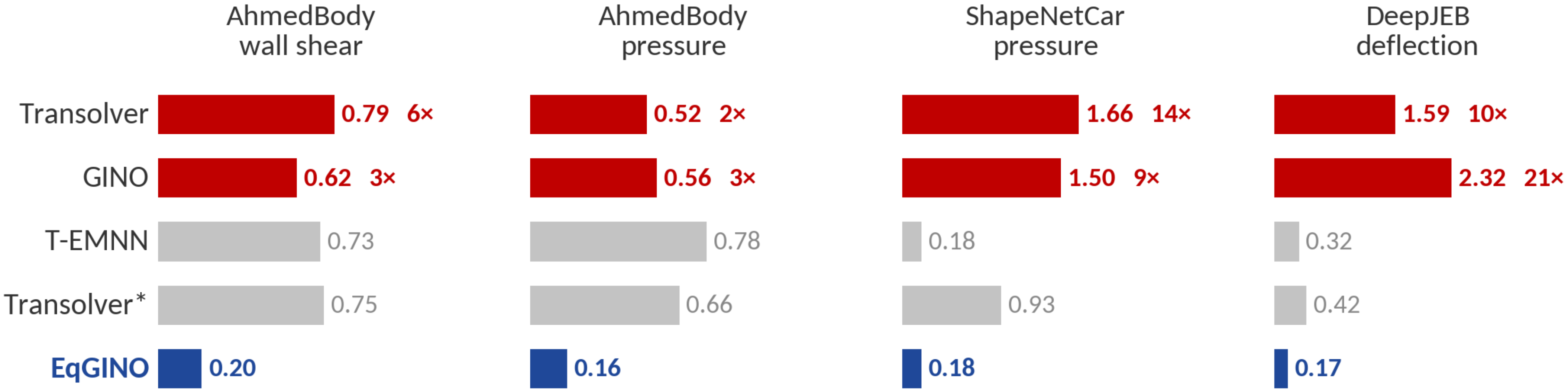
→ **Competitive with the coordinate-dependent models, far ahead of the equivariant ones**

EqGINO is competitive with the coordinate-dependent models and far ahead of the equivariant ones, using no coordinates.

Transolver (Wu et al., 2024) · GINO (Li et al., 2023) · T-EMNN (Kim et al., 2025)

Zero-Shot Under Rotation

Protocol (b): trained on canonical data, tested at 90° multiples



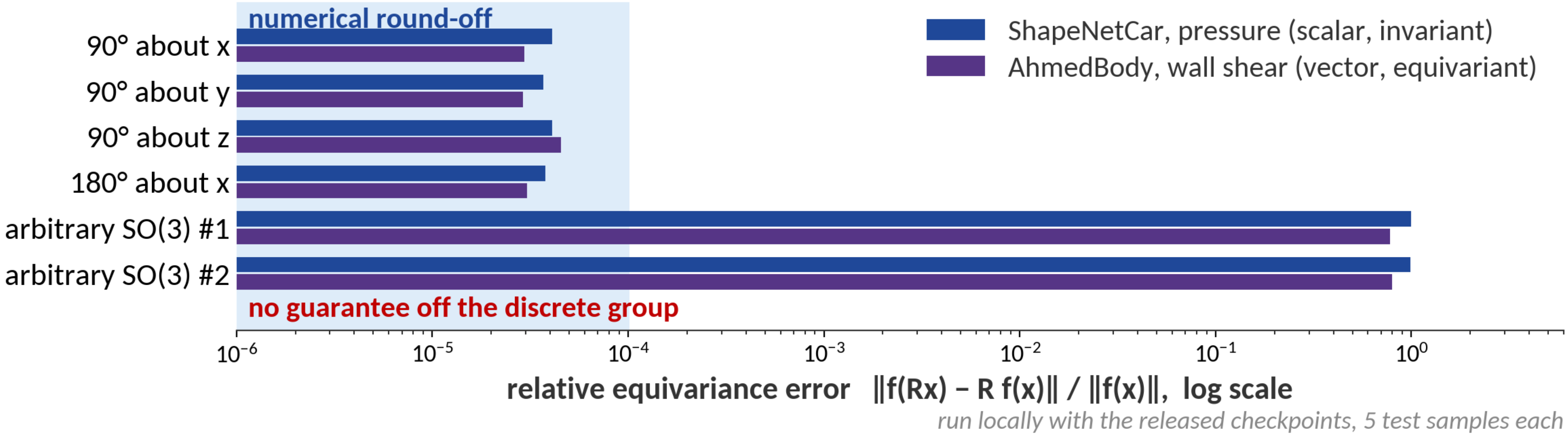
- Same four targets and the same models; only the test orientation changed

→ **Ours is the shortest bar in every panel now, and it was not before**

Rotation multiplies the error of coordinate-dependent models and leaves equivariant ones untouched.

Equivariance, Measured

Rotate the input, compare against the rotated output



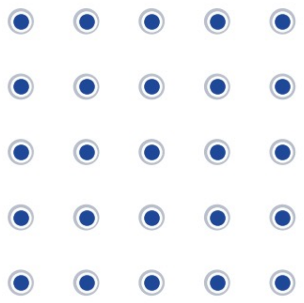
→ **Exact where it is claimed, absent where it is not, and both are measured**

The discrete guarantee holds to numerical round-off; off that group there is none.

Why the Guarantee Stops at 90°

The exactness comes from the grid

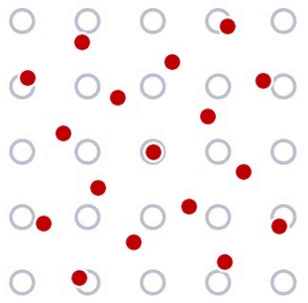
rotate by 90°



every point lands on a grid point,
so the modes are simply permuted

vs

rotate by any other angle



the points land in between,
so the field has to be resampled

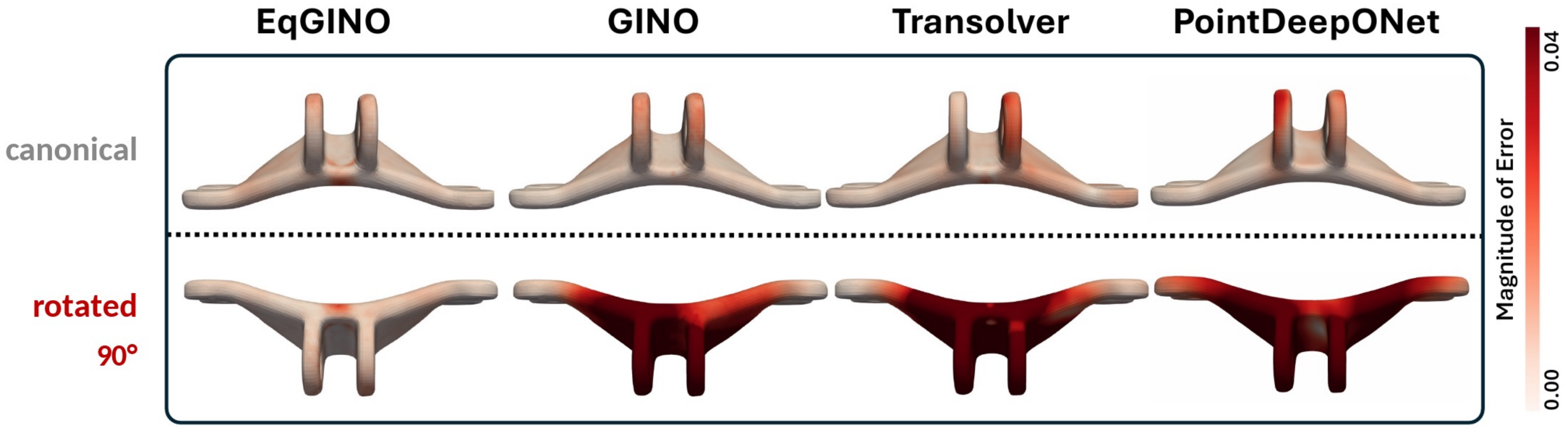
open circle: the grid. filled dot: where that grid point goes

→ **The guarantee holds exactly for the rotations the grid admits, and not beyond**

The exact guarantee comes from the grid, so it covers exactly the rotations the grid admits.

DeepJEB Error Maps

Deflection error on the jet engine bracket, darker is worse

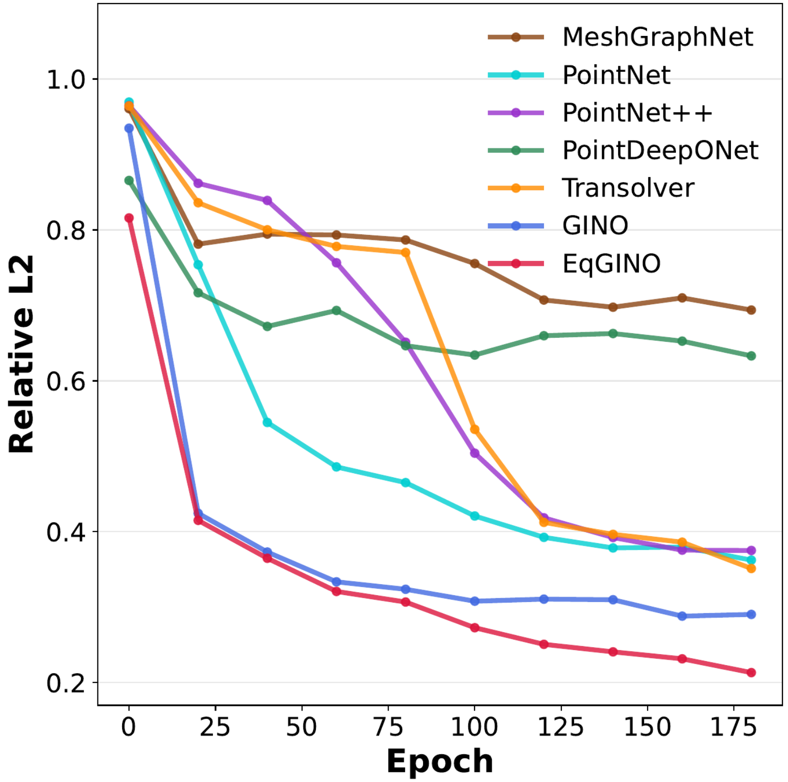


→ **Rotated, the three baselines are wrong over the entire part, not in one hard region**

Under rotation the error stays local for EqGINO and spreads over the entire part for the baselines.

Continuous Rotations

AhmedBody (CFD), wall shear stress, trained and validated under arbitrary SE(3) rotation



What is guaranteed

- Exact for 90° multiples only
- Arbitrary angles need augmented training

What we observe

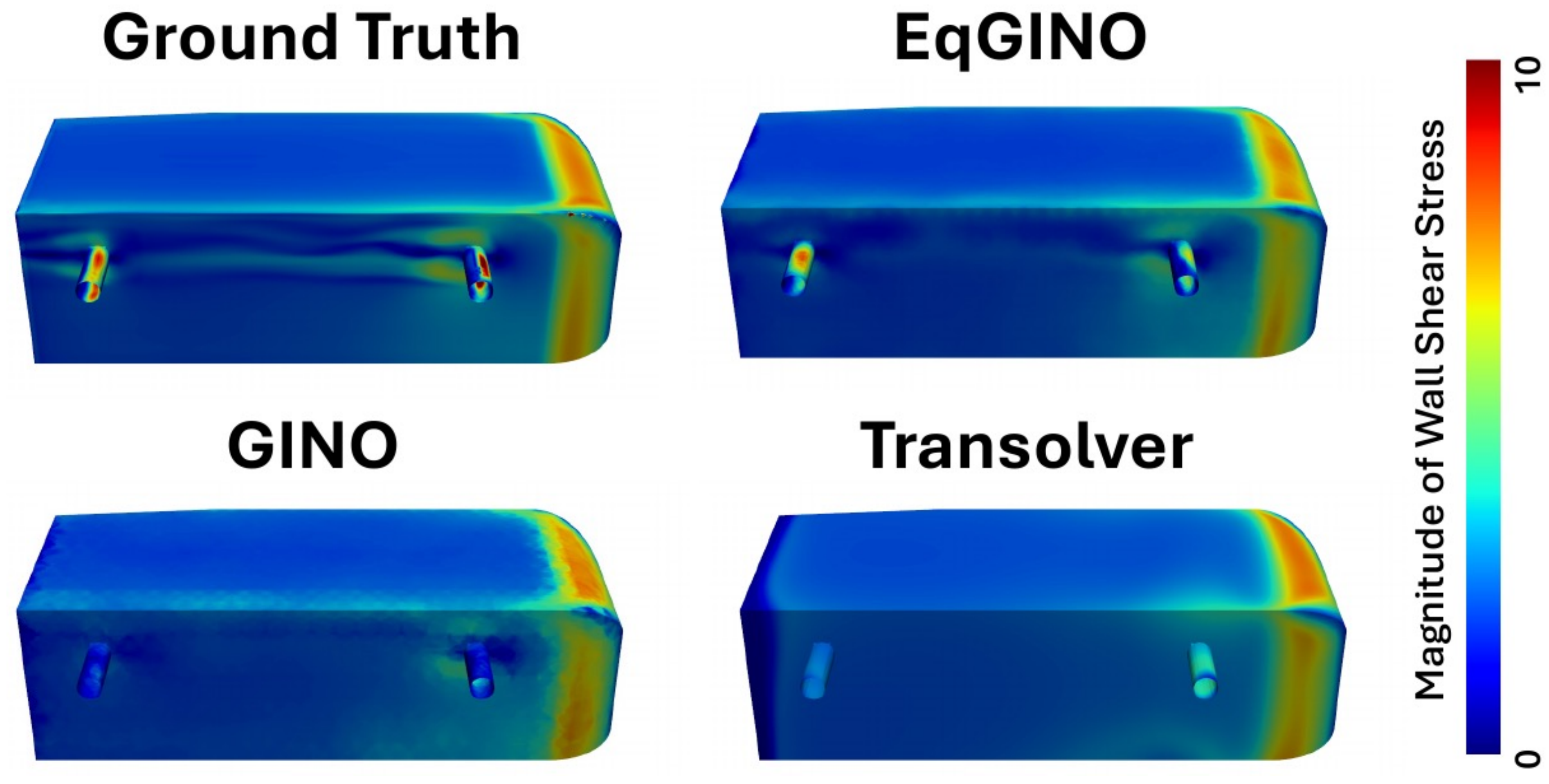
- **EqGINO** separates inside 20 epochs and converges lower

→ The discrete guarantee acts as a prior for the continuous group

The discrete guarantee reduces continuous rotation to a smaller learning problem.

Wall Shear Stress at 58.7°

AhmedBody (CFD), wall shear stress magnitude, test sample rotated 58.7° about *x*



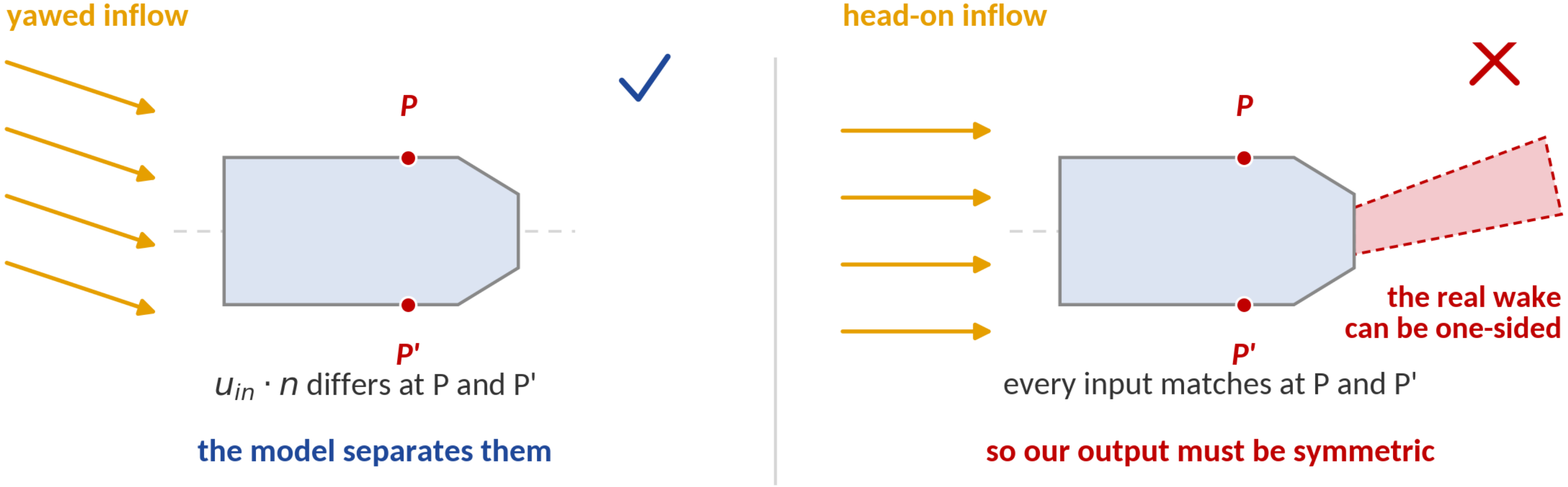
At an arbitrary angle EqGINO still recovers the pillar wakes and the separation band that the baselines lose.

05 Limits and What Comes Next

What is not solved, and where this goes next

Symmetric Geometry, Asymmetric Physics

Where an equivariant model is structurally unable to follow



- Node features include $u_{\mathrm{in}} \cdot \mathbf{n}$, the inflow projected on the surface normal

→ If the real flow breaks a symmetry that the input does not, we cannot represent it

When the whole input is mirror-symmetric, an equivariant model can only produce a symmetric output, whatever the physics does.

Where We Are Weakest

The two targets that go through the local basis

1. Where we do well

- A scalar target needs no reference direction, and those are our strongest columns

2. Where we trail

- Wall shear stress and deflection are vectors, and both go through the local basis
- On those two we are behind the best coordinate-dependent baseline

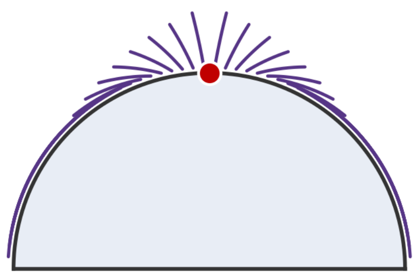
→ **Not a coincidence. The basis is the hard part**

Our weakest results are exactly the targets that have to pass through the local basis.

A Continuous Field, a Discontinuous Frame

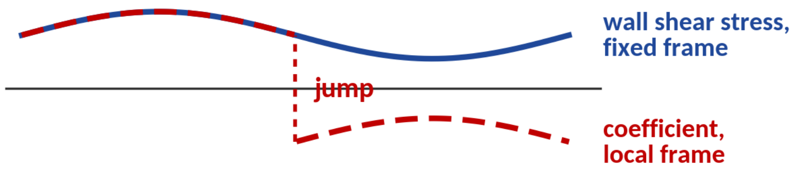
The frame flips where the relative position runs parallel to F

Comb the frame over the surface



the frame flips across this point

What the network has to regress



a discontinuity with no physical counterpart

- Scalars are unaffected. Pressure and von Mises stress need no frame at all

→ Hairy ball theorem: no pointwise recipe avoids it. A global basis is worth discussing

The field is smooth between neighbouring points, the local frame is not, so the regression target inherits a jump the physics does not have.

Scale Is Not Yet Industrial

Where this stops being usable today

1. What bounds us

- Grid resolution bounds the detail we can resolve
- Neighbour search in the graph operator scales as $O(N\, d\, r^3)$, inherited from **GINO**

2. What that means

- Our benchmarks carry a few thousand surface points; production meshes carry millions

→ **Million-cell meshes remain future work**

A transformer-based successor removes both this and the continuous-rotation limit. It is close to done and we expect it public soon.

The method is demonstrated at benchmark scale, and production mesh sizes are still open.

Implications for CAE Workflows

What changes if the surrogate is coordinate-free

1. No canonical alignment step

- And no silent failure when that step gets something wrong

2. A rotation test costs nothing

→ **Rotate your existing test set by 90°. Run it on any surrogate**

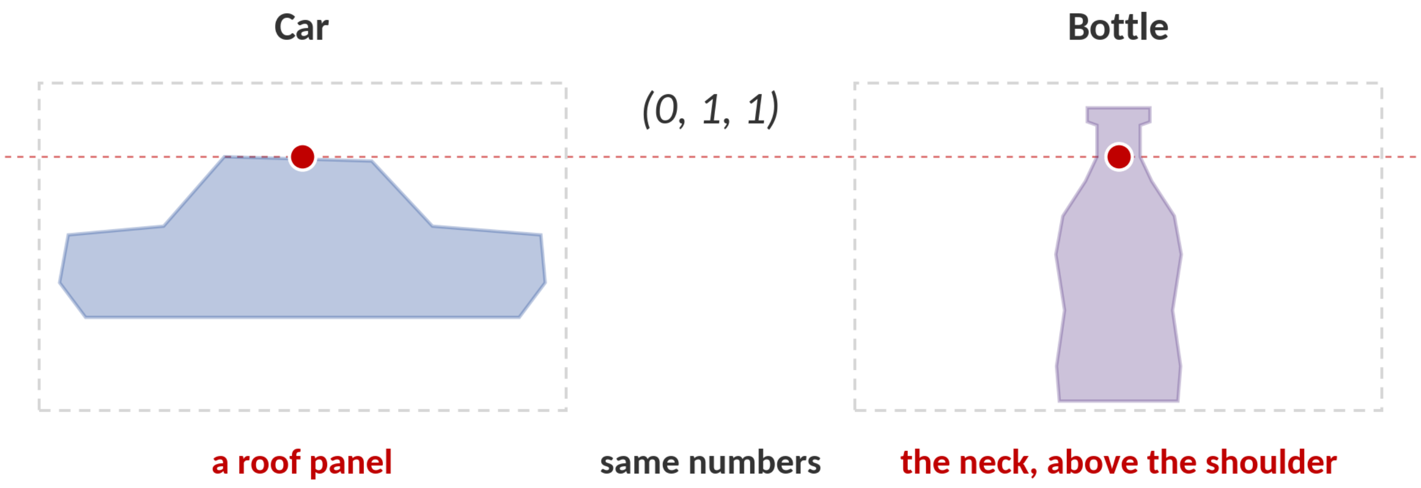
3. The two modules are separable

- **EqFNO** for any grid-based field in 2D or 3D, **EqGNO** for unstructured geometry

The practical gain is a surrogate whose accuracy does not depend on how the part was oriented in CAD.

Cross-Geometry, the Longer Programme

The same coordinate on a car and on a bottle

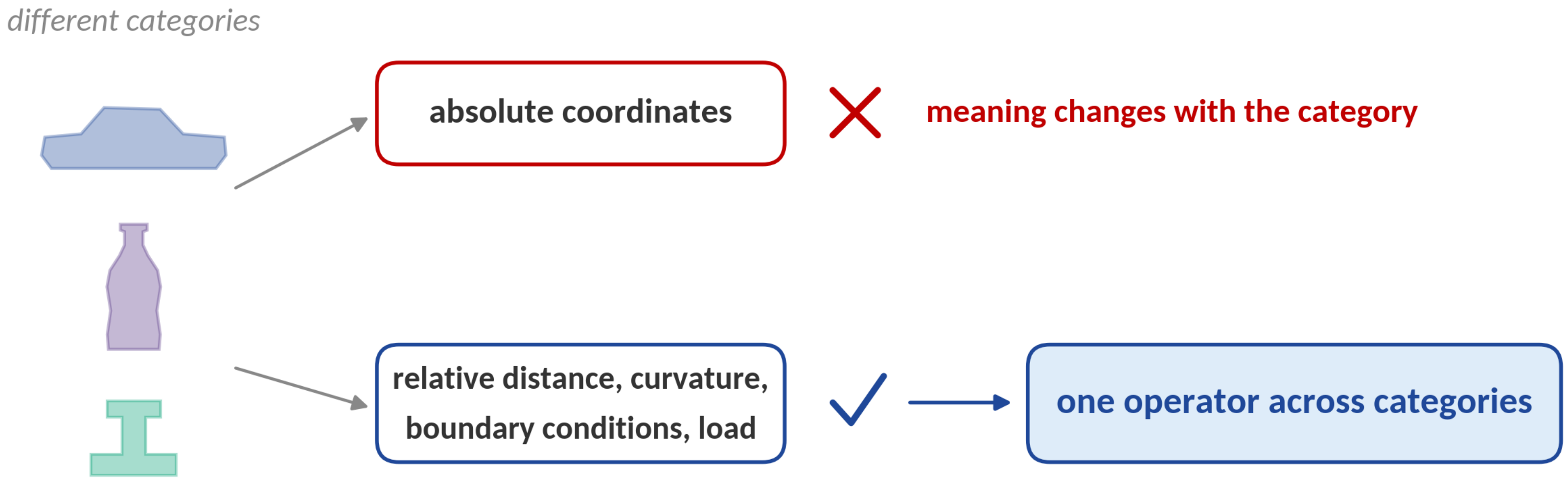


→ Navier-Stokes does not know which category it is looking at

A coordinate carries category-specific meaning, and the governing equations carry none. That is the programme, not a claim about this model.

Toward a Foundation Model for Physics

What one operator across categories would require



→ **EqGINO is the first step: the frame dependence is gone. The category dependence is not solved here**

Removing the coordinate is the first step toward one operator across categories, not an optimisation.

Summary

Diagnosis, construction, evidence

1. The diagnosis

- Absolute coordinates let a surrogate fit the observer's frame instead of the governing equation

2. The construction

- Isotropy in the spectral domain gives exact discrete equivariance
- Same constraint reduces spectral parameters from $O(K^3)$ to $O(K)$

3. The evidence

- Baselines lose one to two orders of magnitude under rotation; **EqGINO** loses nothing

→ **A model that cannot see the coordinate frame cannot overfit to it**

Accuracy under rotation requires equivariance in the architecture, not coverage in the data.

Thank you

Questions welcome

Code github.com/sung-won-kim/EqGINO

Web sung-won-kim.github.io

Mail swkim@kaist.ac.kr

LinkedIn

References

Works cited on the slides, in the form given by the paper's bibliography

Ahmed, S. R., Ramm, G., Faltin, G. Some salient features of the time-averaged ground vehicle wake. SAE Transactions, 1984.

Alon, U., Yahav, E. On the bottleneck of graph neural networks and its practical implications. arXiv:2006.05205, 2020.

Chang, A. X. et al. ShapeNet: an information-rich 3D model repository. arXiv:1512.03012, 2015.

Helwig, J. et al. Group equivariant Fourier neural operators for partial differential equations. arXiv:2306.05697, 2023.

Hong, S. et al. DeepJEB: 3D deep learning-based synthetic jet engine bracket dataset. Journal of Mechanical Design 147, 2025.

Kim, S. et al. Thickness-aware E(3)-equivariant 3D mesh neural networks. ICML, 2025.

Li, Q., Han, Z., Wu, X.-M. Deeper insights into graph convolutional networks for semi-supervised learning. AAAI, 2018.

Li, Z. et al. Neural operator: graph kernel network for partial differential equations. arXiv:2003.03485, 2020.

Li, Z. et al. Fourier neural operator for parametric partial differential equations. arXiv:2010.08895, 2020.

Li, Z. et al. Geometry-informed neural operator for large-scale 3D PDEs. NeurIPS 36, 2023.

Pfaff, T. et al. Learning mesh-based simulation with graph networks. ICLR, 2020.

Qi, C. R. et al. PointNet: deep learning on point sets for 3D classification and segmentation. CVPR, 2017.

Satorras, V. G., Hoogeboom, E., Welling, M. E(n) equivariant graph neural networks. ICML, 2021.

Trang, T. A. et al. E(3)-equivariant mesh neural networks. AISTATS, 2024.

Umetani, N., Bickel, B. Learning three-dimensional flow for interactive aerodynamic design. ACM TOG 37, 2018.

Whalen, E., Beyene, A., Mueller, C. SimJEB: simulated jet engine bracket dataset. Computer Graphics Forum 40, 2021.

Wu, H. et al. Transolver: a fast transformer solver for PDEs on general geometries. arXiv:2402.02366, 2024.