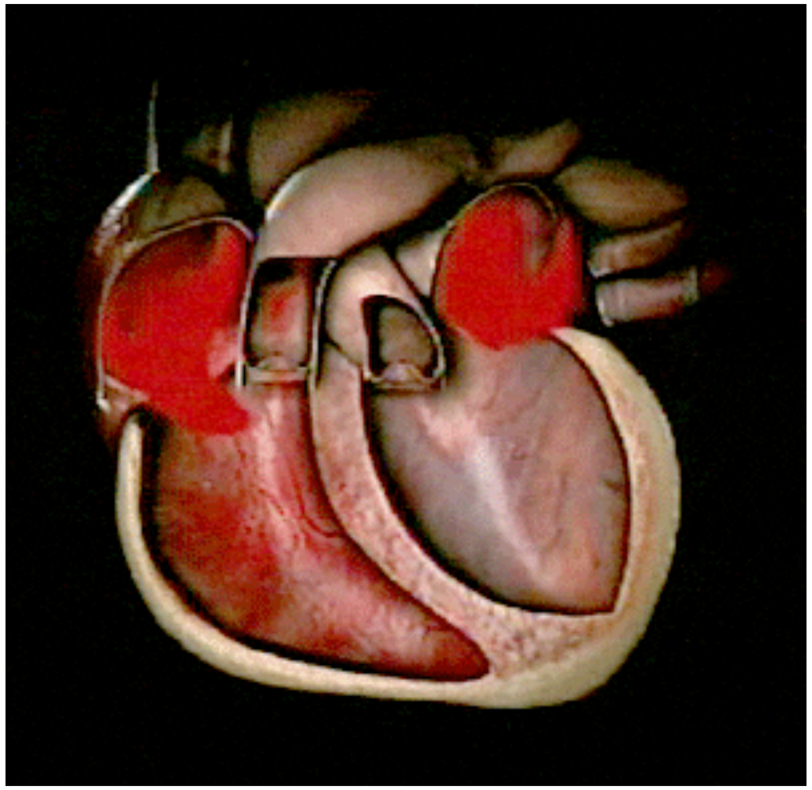
UC San Diego

Physics-Guided Deep Learning for Fluid Dynamics

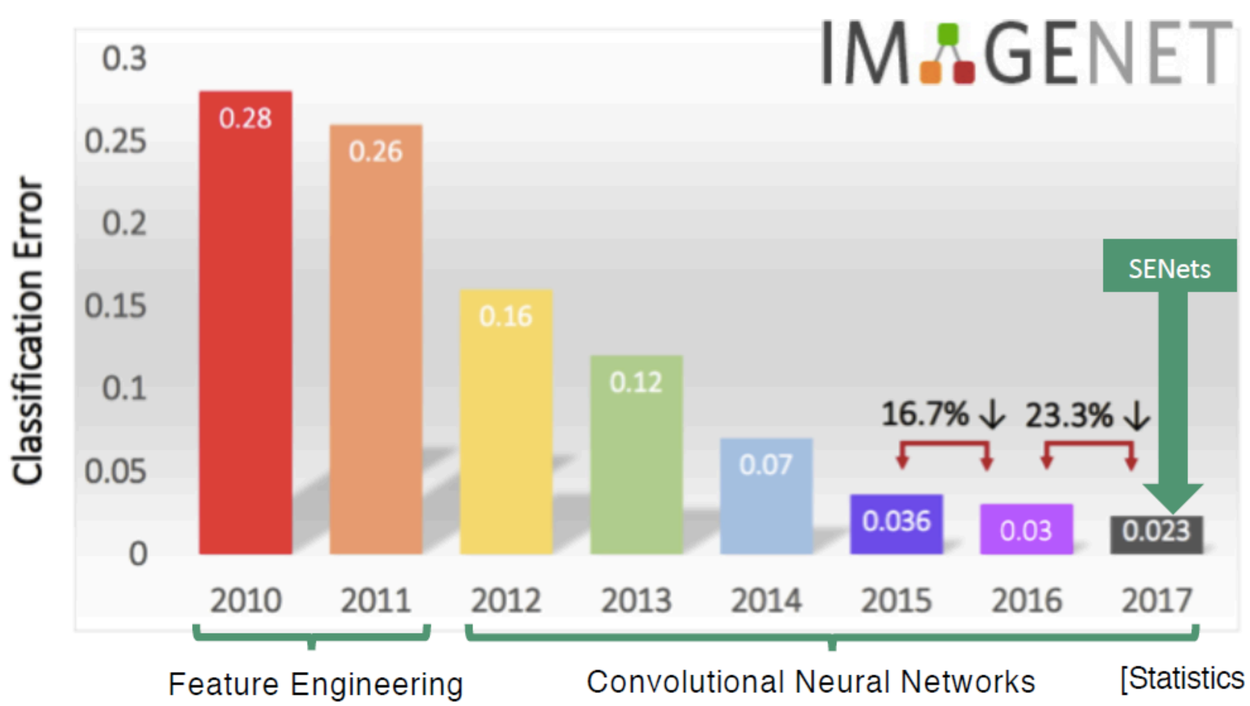
Rose Yu

Assistant Professor
University of California, San Diego

Fluid Dynamics



Promise of Deep Learning



[Statistics provided by ILSVRC]

Fluid Dynamics?

Deep Learning for Fluid Dynamics

- **Fluid Animation** [Tompson et al. 2017, Chu and Thuerey, 2017, Thuerey et al. 2019, Sanchez-Gonzalez et al. 2020]
    - emphasize simulation realism
    - lack physical interpretation
- **Data-Driven DL** [Chertkov et al. 2019, Mohan et al. 2020, Kochkov et al. 2021]
    - use DL as a function approximator
    - no explicit physical constraints
- **Physics-Guided DL** [Ling et al. 2016, Raissi et al. 2017, Kim and Lee 2019, Wu et al. 2019, Jiang et al, 2020]
    - no external force, require boundary condition inputs
    - only spatial modeling, no temporal dynamics

Brunton, Steven L., Bernd R. Noack, and Petros Koumoutsakos. "Machine learning for fluid mechanics." *Annual Review of Fluid Mechanics* 52 (2020): 477-508.

Accelerating Turbulence Simulation

Rayleigh-Bénard convection[1]

Rui Wang
UCSD

Karthik Kashinath
Lawrence Berkeley

Mustafa Mustafa
Lawrence Berkeley

Adrian Albert
Lawrence Berkeley

Towards Physics Informed Deep Learning for Spatiotemporal Modeling of Turbulence Flows
Rui Wang Adrian Albert, Karthik Kashinath, Mustafa Mustafa, Rose Yu
In ACM SIGKDD Conference on Knowledge Discovery and Data (KDD), 2020

Hybrid Learning Framework

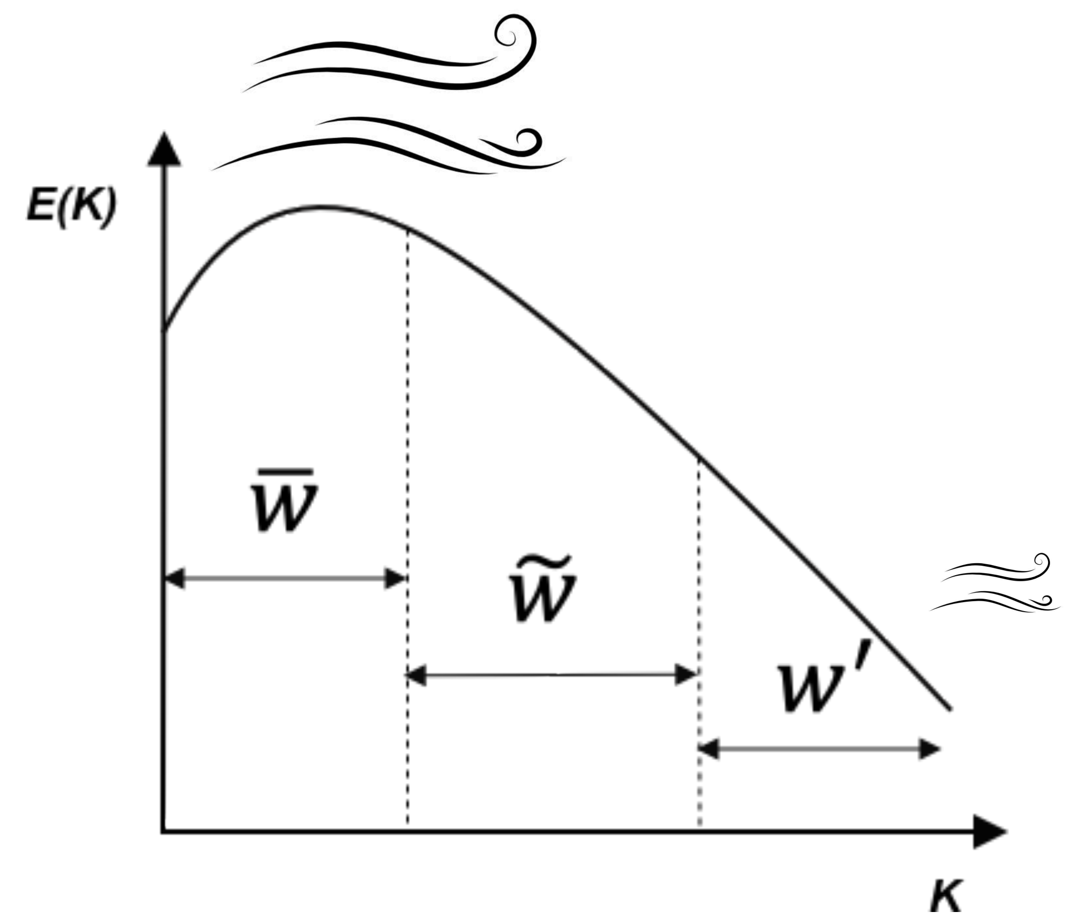
- Navier-Stokes equations: describe the motion of viscous fluids

- Reynolds Averaging (RANS)

$$\mathbf{w}(\mathbf{x}, t) = \bar{\mathbf{w}}(\mathbf{x}, t) + \mathbf{w}'(\mathbf{x}, t)$$
$$\bar{\mathbf{w}}(\mathbf{x}, t) = \frac{1}{T}\int_{t-T}^{t} G(s)\boldsymbol{w}(\mathbf{x}, s)ds$$

- Large Eddy Simulation (LES)

$$\mathbf{w}(\mathbf{x}, t) = \tilde{\mathbf{w}}(\mathbf{x}, t) + \mathbf{w}'(\mathbf{x}, t)$$
$$\tilde{\mathbf{w}}(\mathbf{x}, t) = \int G(\mathbf{x} \,|\, \xi)\mathbf{w}(\xi, t)d\xi$$



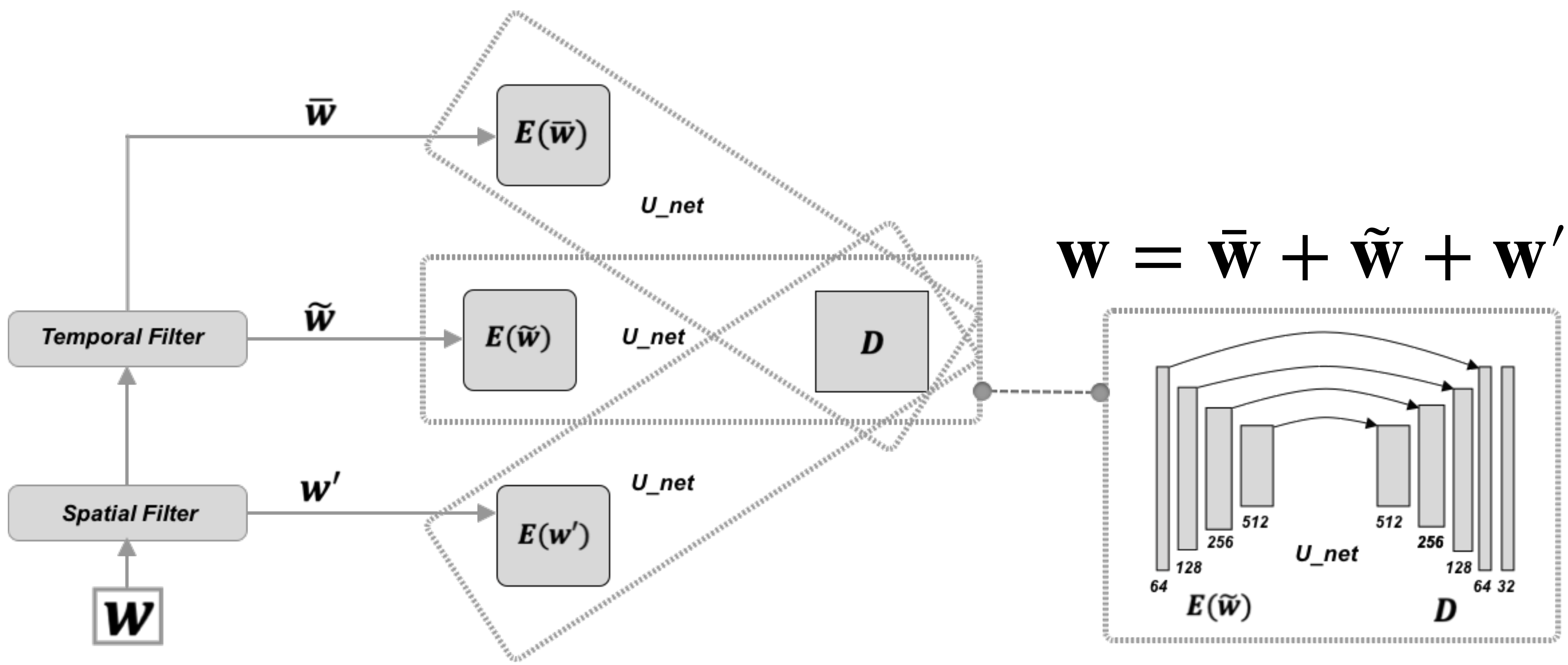
Turbulent-Flow Net

- RANS-LES Coupling

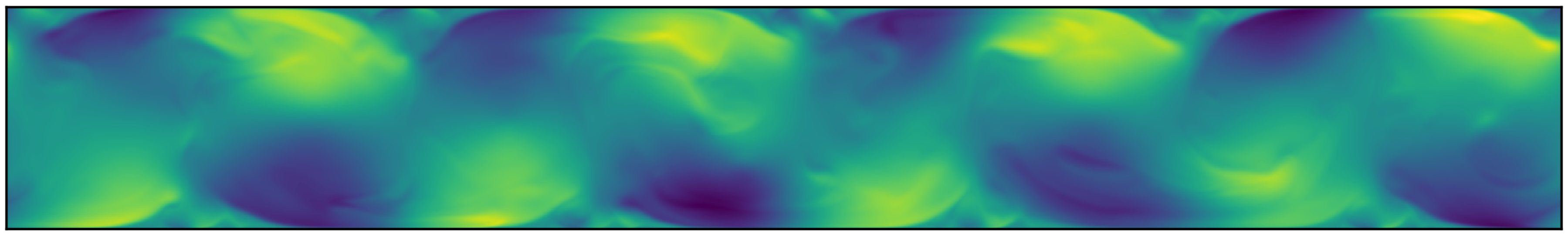
Spatial Filter **Temporal Filter**

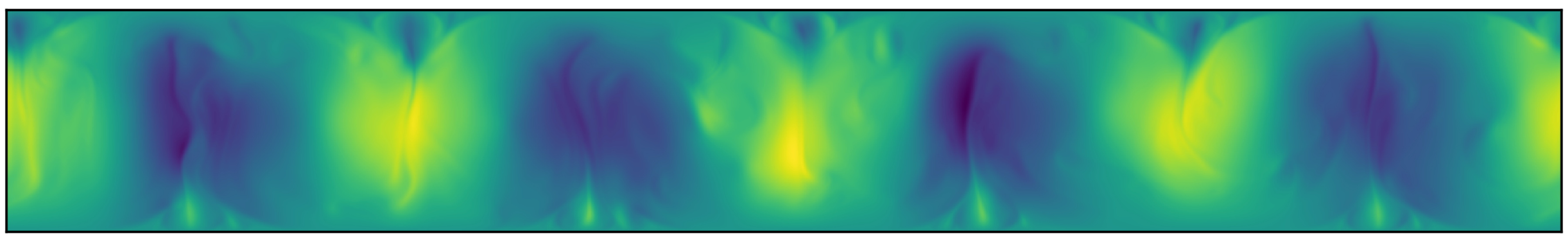
$$\mathbf{w}^*(\mathbf{x},\mathbf{t}) = \sum_{\xi} \boxed{G_1(\mathbf{x} \mid \xi)}\mathbf{w}(\xi, t) \qquad \bar{\mathbf{w}}(\mathbf{x},\mathbf{t}) = \frac{1}{T}\sum_{s=t-T}^{t} \boxed{G_2(s)}\mathbf{w}^*(\mathbf{x}, s)$$

$$\mathbf{w} = \bar{\mathbf{w}} + \tilde{\mathbf{w}} + \mathbf{w}'$$



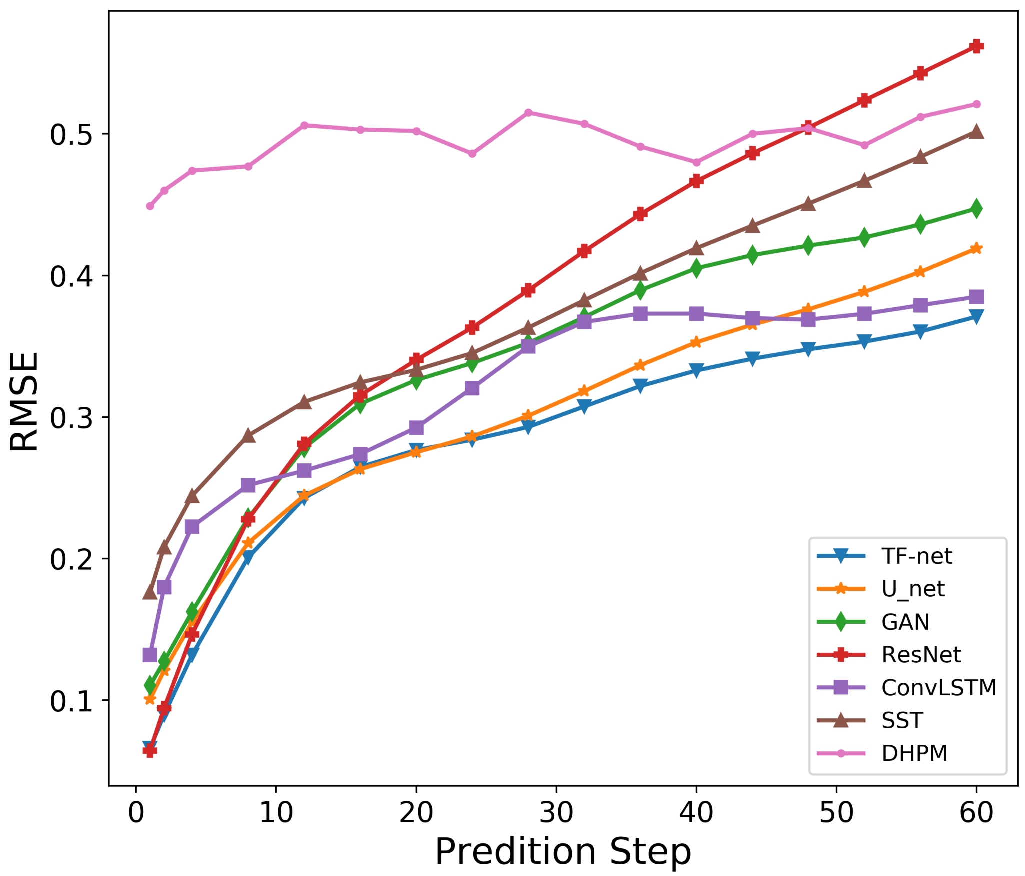
Data Description

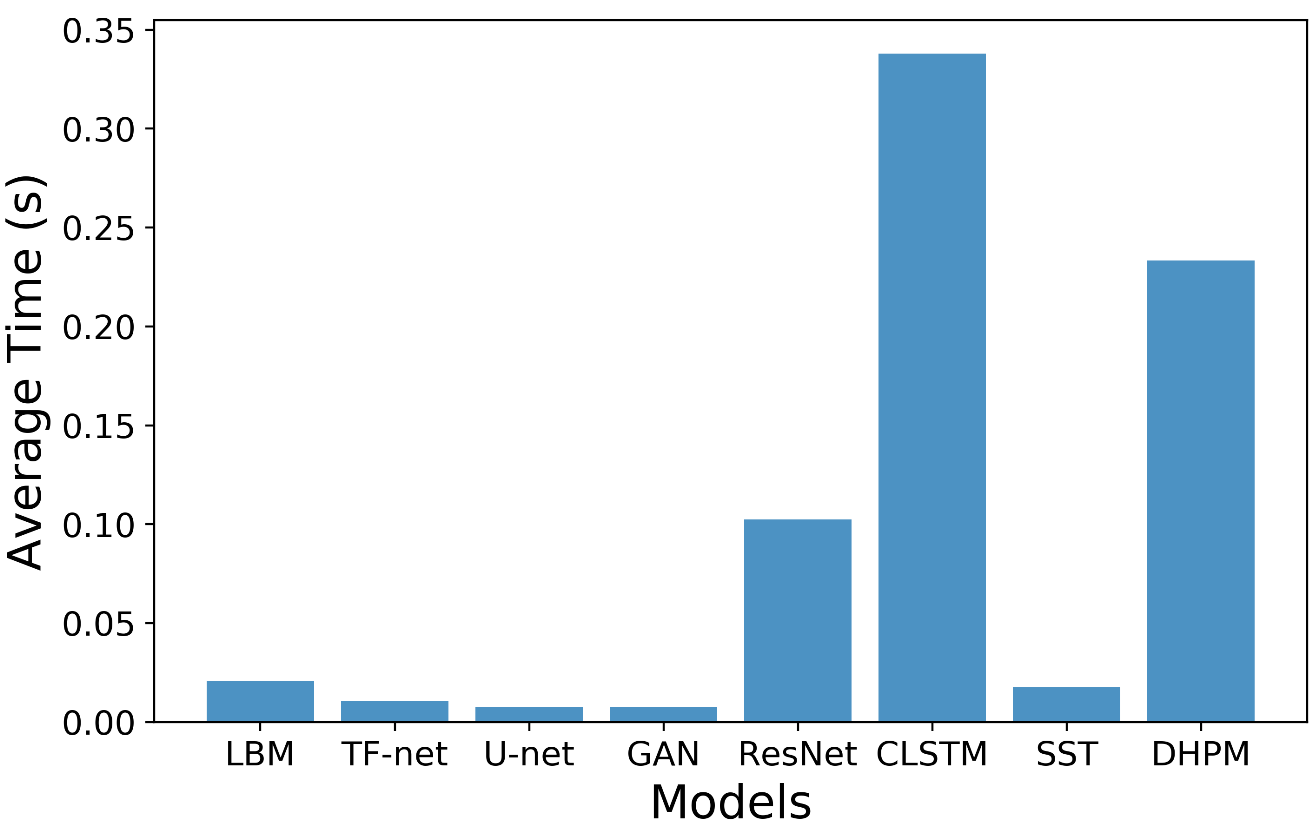




- RBC simulation with Prandtl number 0.71 and Reynolds number 2.5 x e8
- ~10k sequences, spatial resolution 64x64, time length 90
- 60 time step ahead prediction, results averaged over three runs

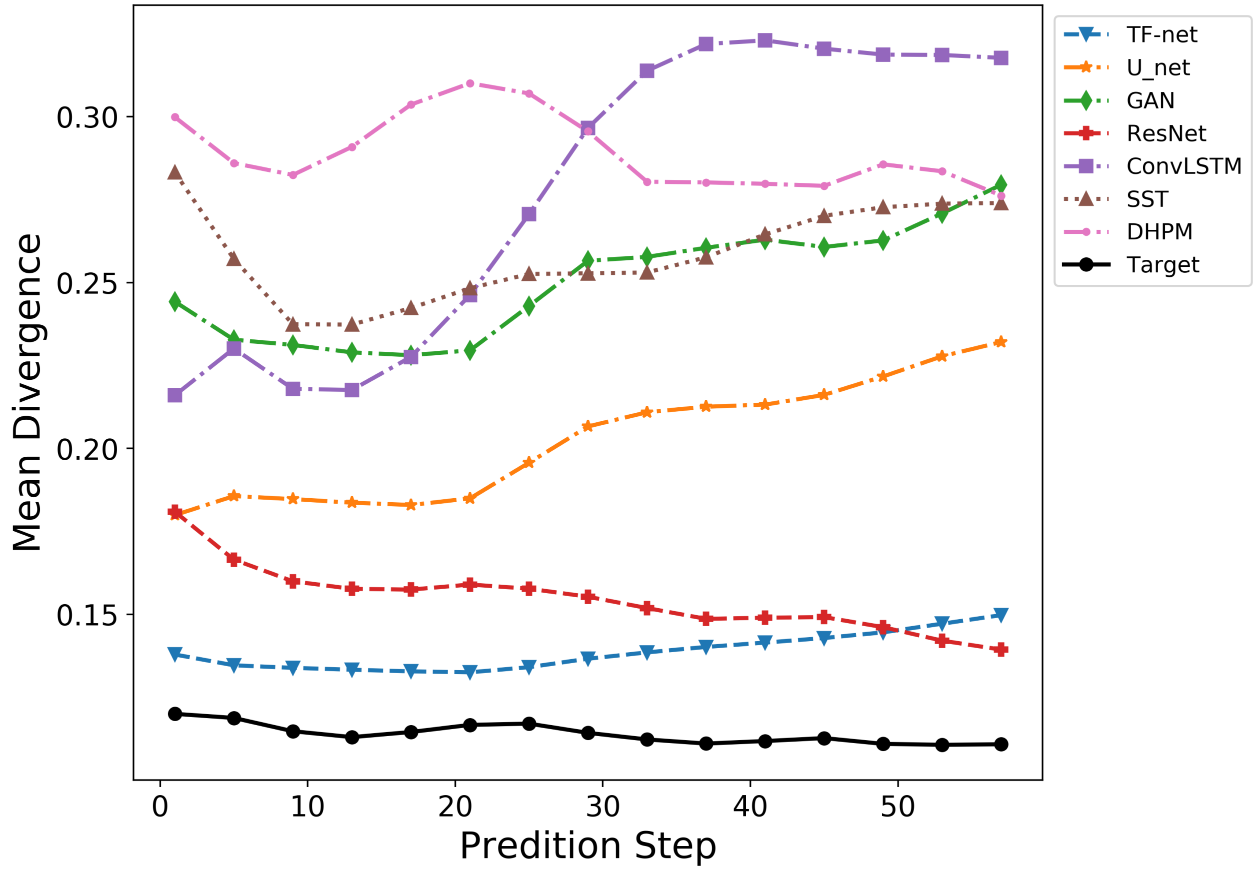
Prediction Performance

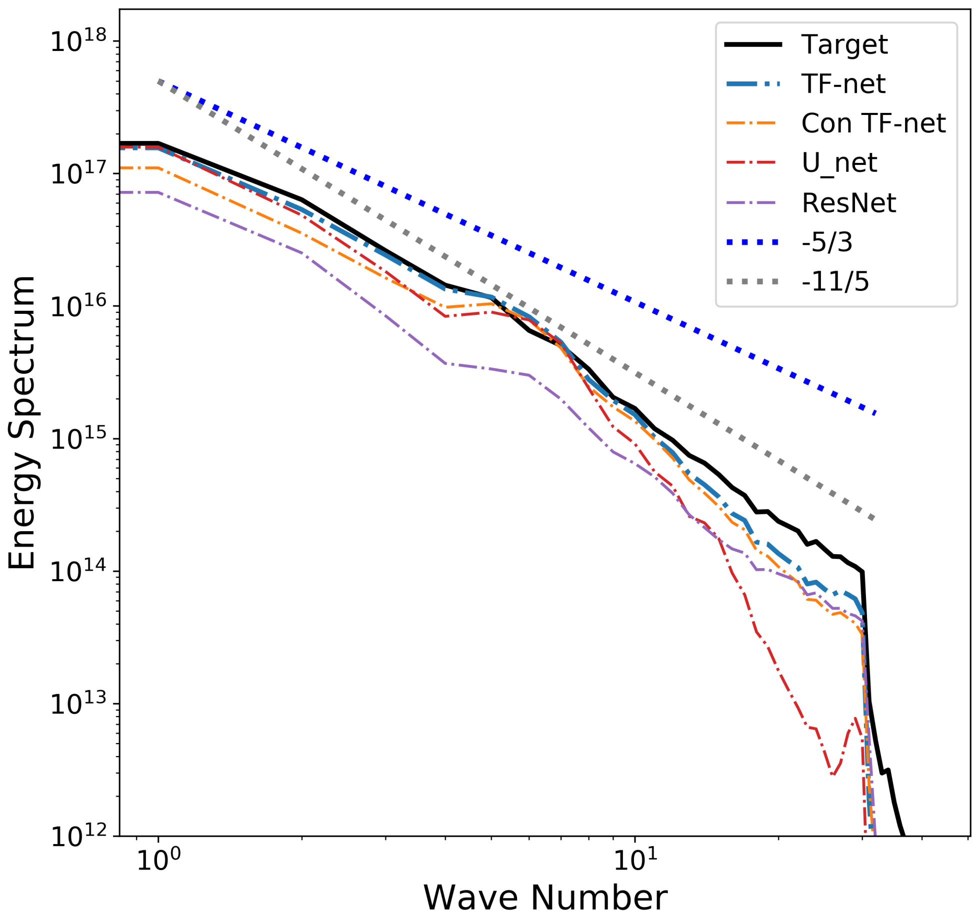




- TF-Net consistently outperforms baselines on forward prediction RMSE
- 2X faster than Lattice Boltzmann method (LBM)

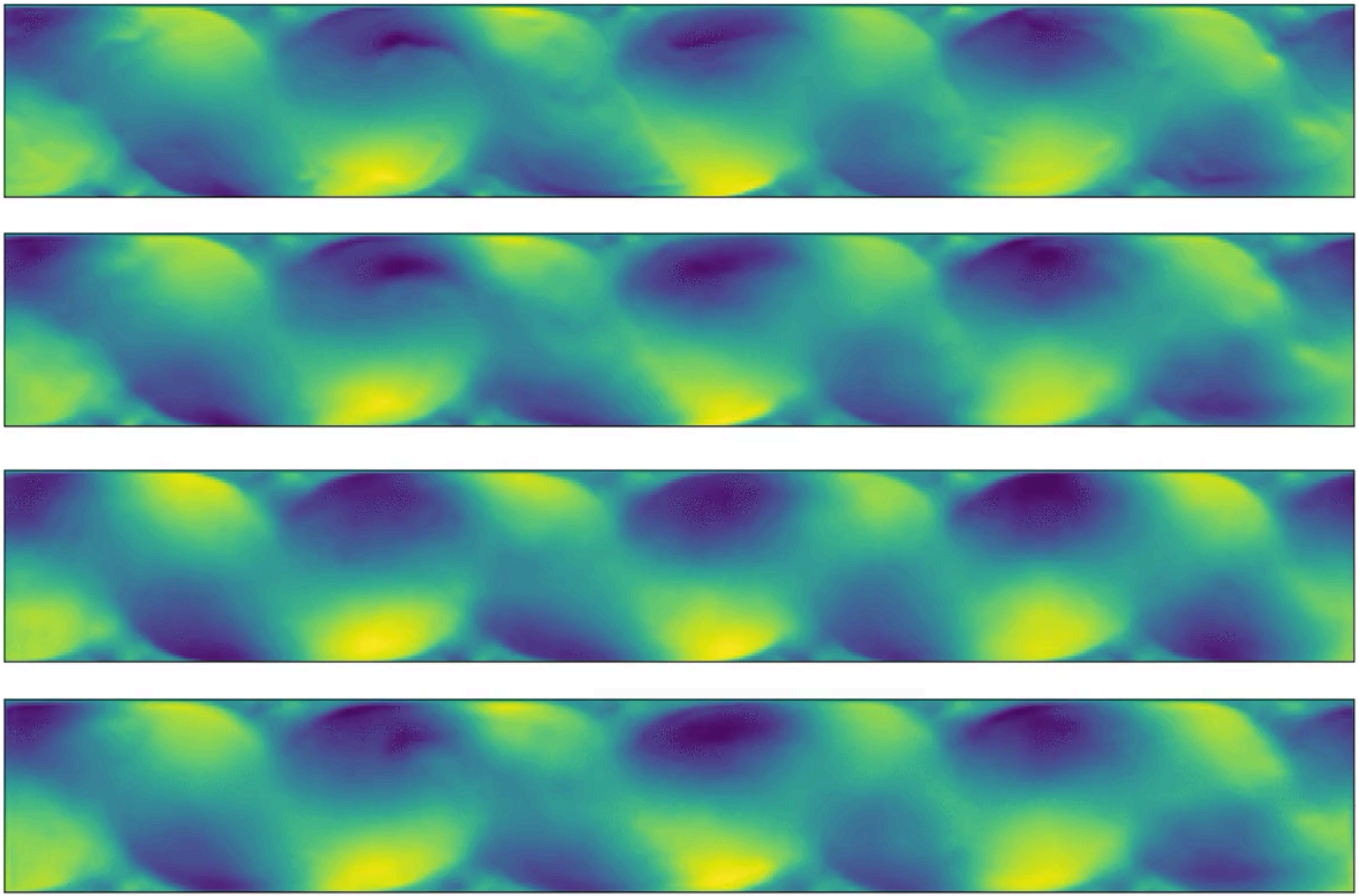
Physical Consistency





- TF-net predictions are closest to the target w.r.t. kinetic energy
- Video forward predictions methods (e.g. Unet, ConvLSTM) cannot capture physical properties

Prediction Visualization



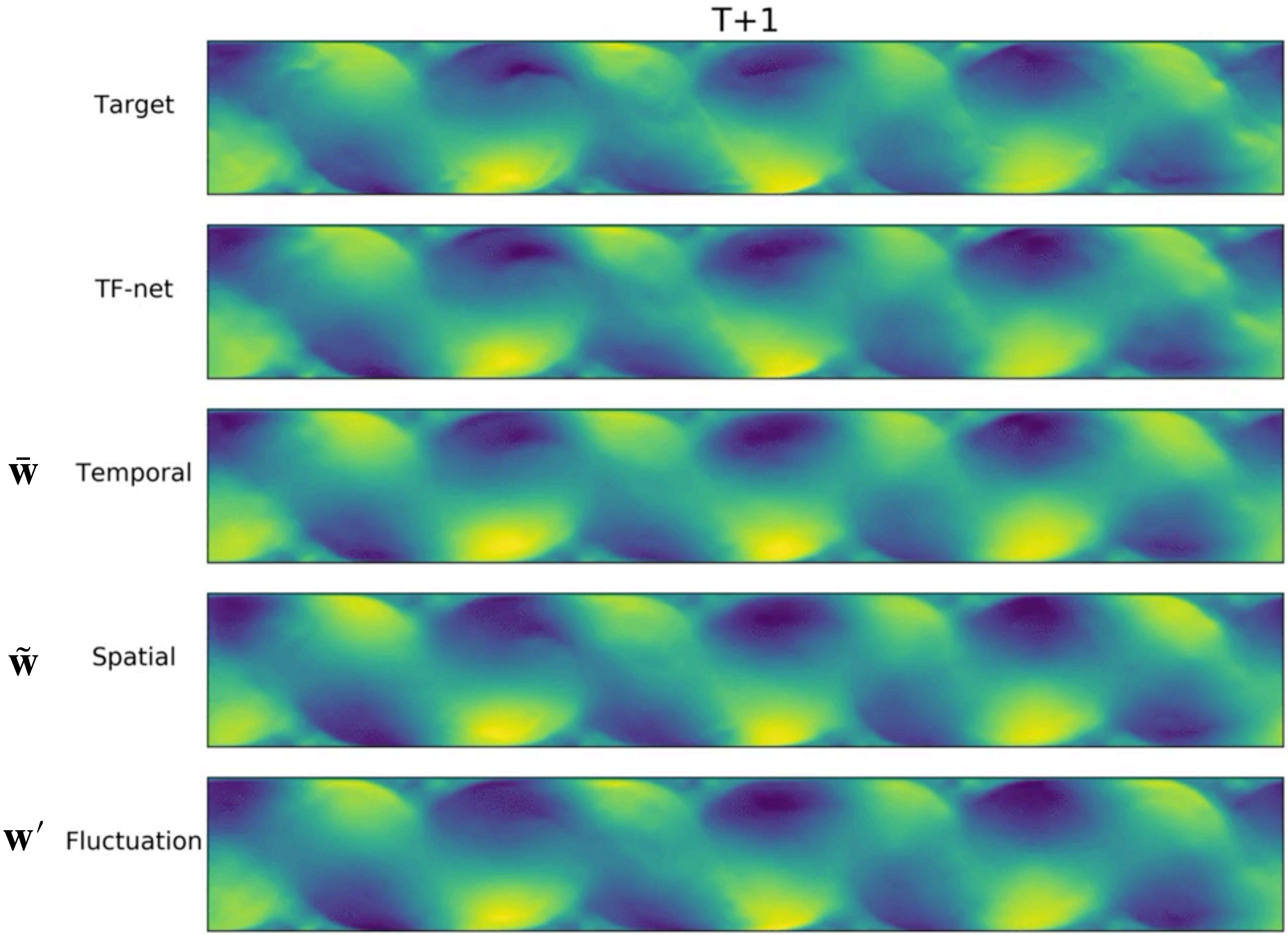
Target

TF-Net

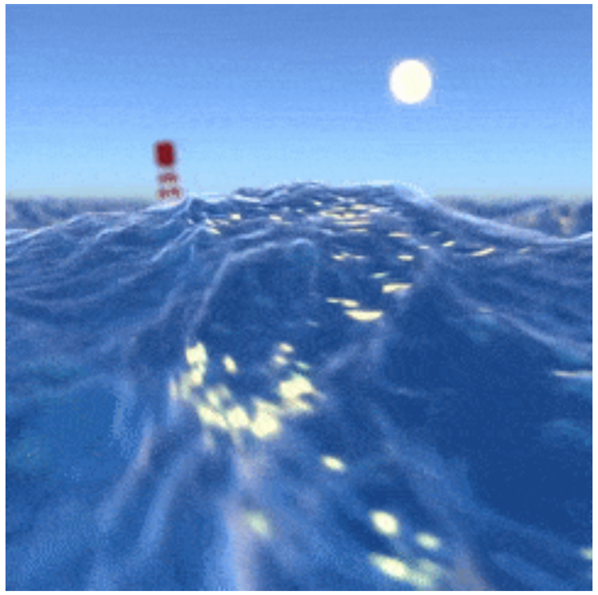
ResNet

GAN

Ablation Study



Incorporating Symmetry for Generalization



Rui Wang

Robin Walters

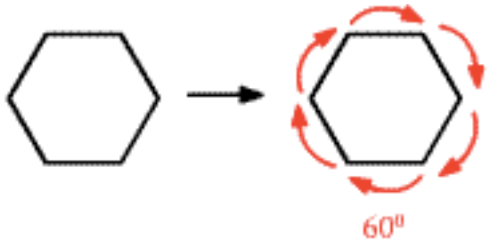
Incorporating Symmetry into Deep Dynamics Models for Improved Generalization
Rui Wang, Robin Walters, and Rose Yu
International Conference on Learning Representations (ICLR), 2021.

Symmetry

- Utilize symmetry to improve generalization
  - dynamics change but the laws of physics do not!
- Noether's theorem: *For every symmetry, there is a corresponding conservation law*
  - *translational* → conservation of momentum
  - *time invariance* →conservation of energy
- As inductive bias to
  - improve generalization
  - encode conservation laws

Group Equivariance

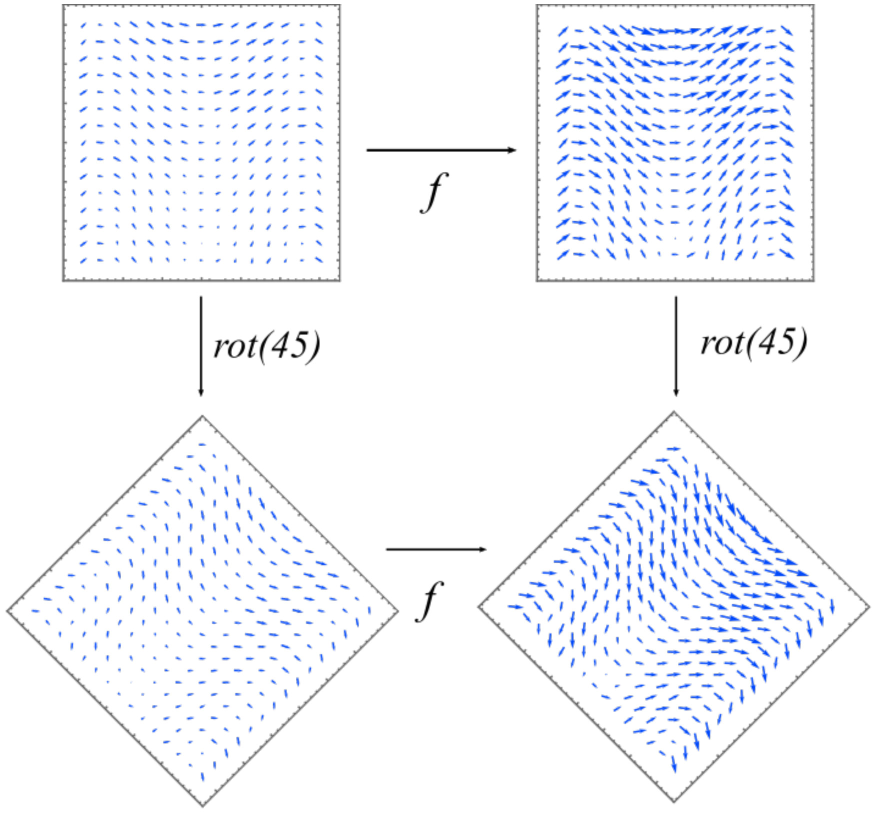
- **Group**: a set G and a composition map $\circ : G \times G \to G$
  - $1 \in G$ and $\forall g \in G, \exists g^{-1} \in G$
  - SO(2): 2d rotation



- **Invariance, Equivariance**: function f and group G
  - G-invariant: $f(g(x)) = f(x)$
  - G-equivariant: $f(gx) = gf(x)$

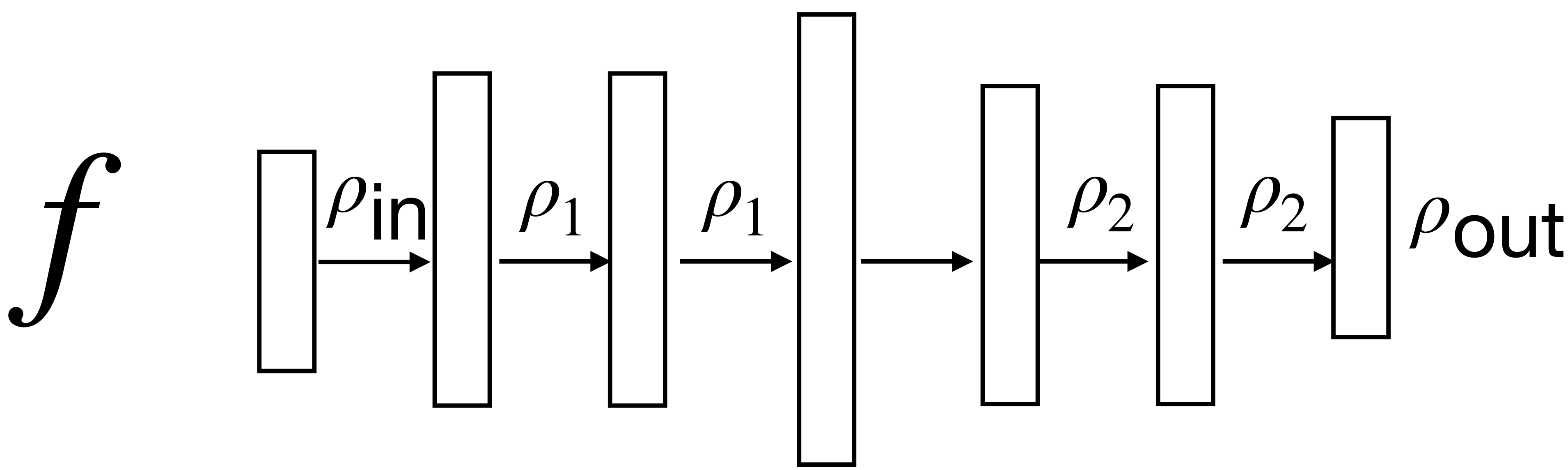
$$f(x, v) = (x, 2v)$$

$$\rho(Rot(\theta)) = \begin{pmatrix} \cos(\theta) & \sin(-\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}$$



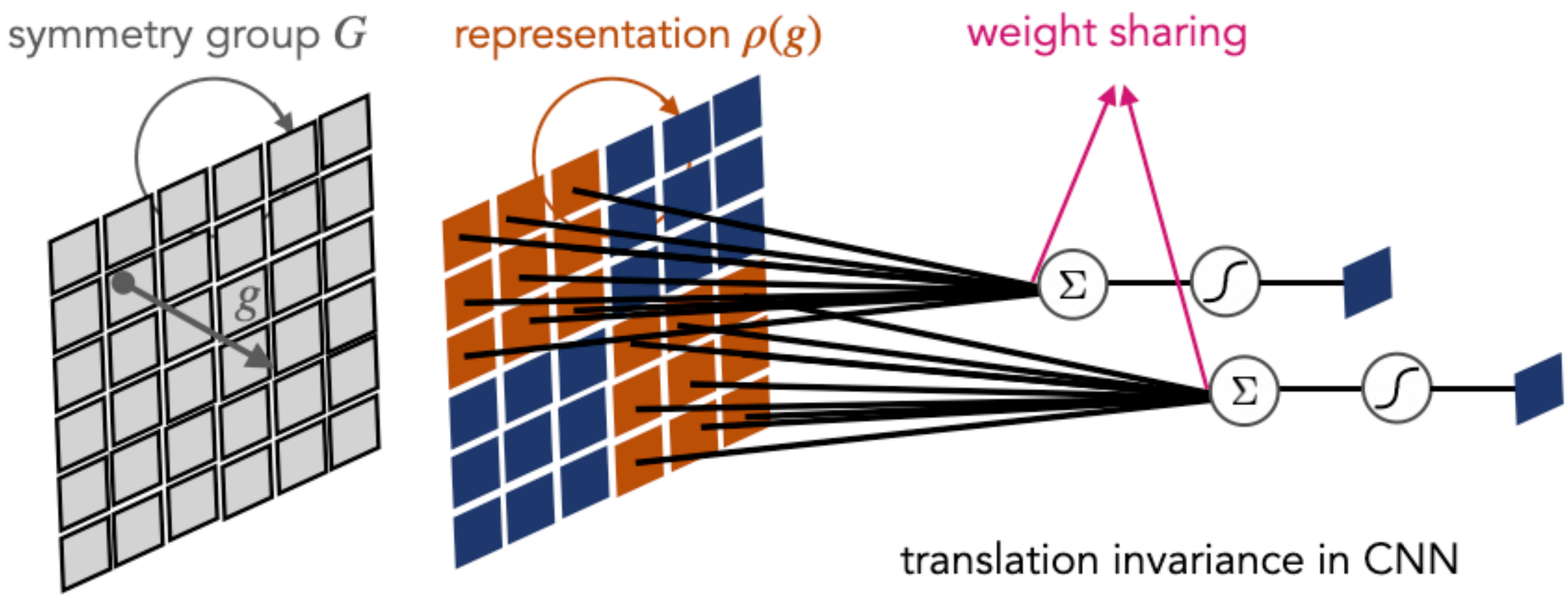
Equivariant Networks

- Use a neural network to learn f that is G-equivariant



- If the maps between layers are equivariant, then the entire network is equivariant.
- Adding skip connections does not affect its equivariance w.r.t. linear actions.

Weight Symmetry



Theorem (Weiler & Cesa 2019): a convolutional layer is G-equivariant if and only if the kernel satisfies $K(gv) = \rho_{out}^{-1}(g)K(v)\rho_{in}(g)$ for all $g \in G$, with action maps ρ_{in} and ρ_{out} .

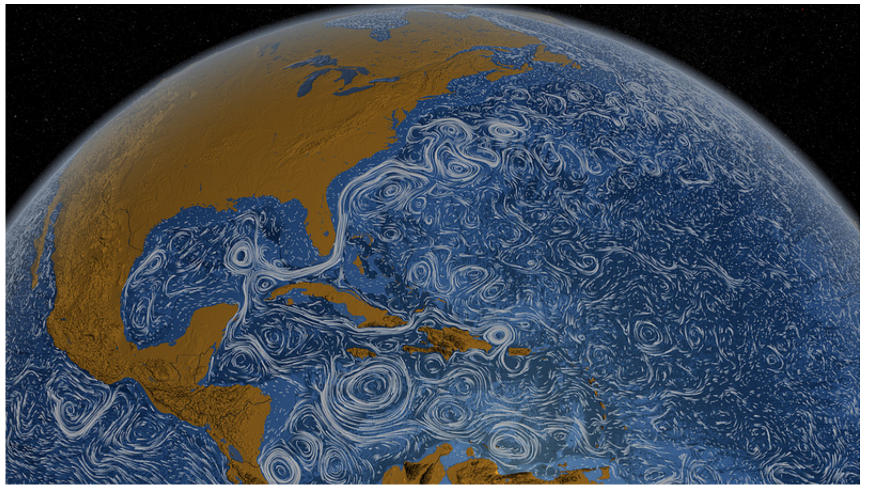
Symmetry of Differential Systems

- A system of differential operators $D = \{P_1, \cdots, P_r\}$
- if ϕ is a solution of D, then for all $g \in G$, $g(\phi)$ is also a solution

Symmetries	*Heat Equ.*	*NS Equ.*	*Params*
Space translation	$H(\boldsymbol{x} - \boldsymbol{v}, t)$	$\boldsymbol{w}(\boldsymbol{x} - \boldsymbol{v}, t)$	$\boldsymbol{v} \in \mathbb{R}^2$
Time translation	$H(\boldsymbol{x}, t - \tau)$	$\boldsymbol{w}(\boldsymbol{x}, t - \tau)$	$\tau \in \mathbb{R}$
Uniform Motion	$\eta H(x - 2\boldsymbol{v}t, t)$	$\boldsymbol{w}(\boldsymbol{x}, t) + \boldsymbol{c}$	$\boldsymbol{c} \in \mathbb{R}^2$
Reflect/rotation	$H(R\boldsymbol{x}, t)$	$R\boldsymbol{w}(R^{-1}\boldsymbol{x}, t)$	$R \in O(2)$
Scaling	$H(\lambda\boldsymbol{x}, \lambda^2 t)$	$\lambda\boldsymbol{w}(\lambda\boldsymbol{x}, \lambda^2 t)$	$\lambda \in \mathbb{R}_{>0}$

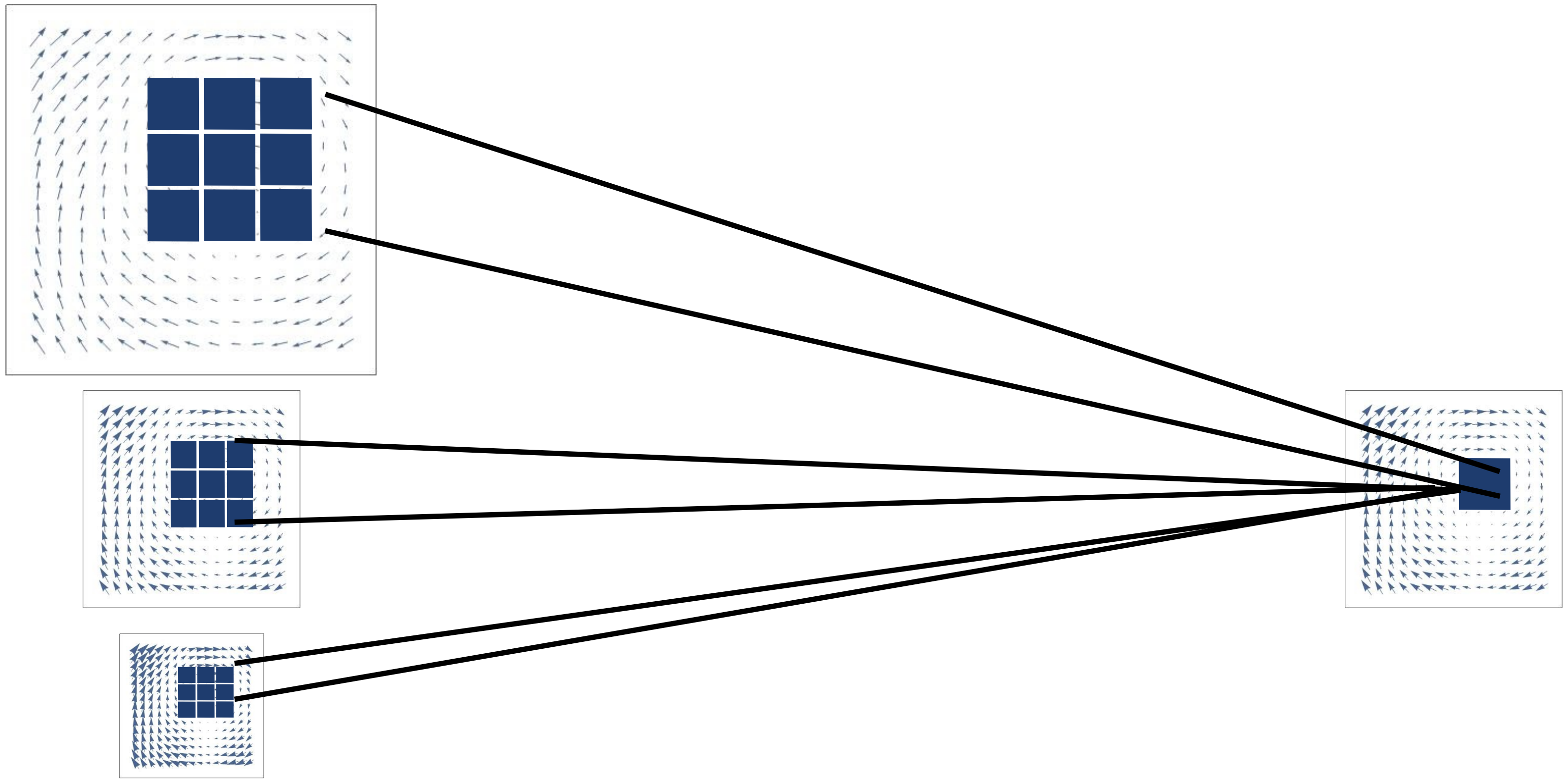
Scaling Law

Symmetry: Scaling



- Standard convolution shares weights across the input by translating a kernel across the input.
- For scale-equivariant convolution, we must translate and scale a kernel across the input

Symmetry: Scaling

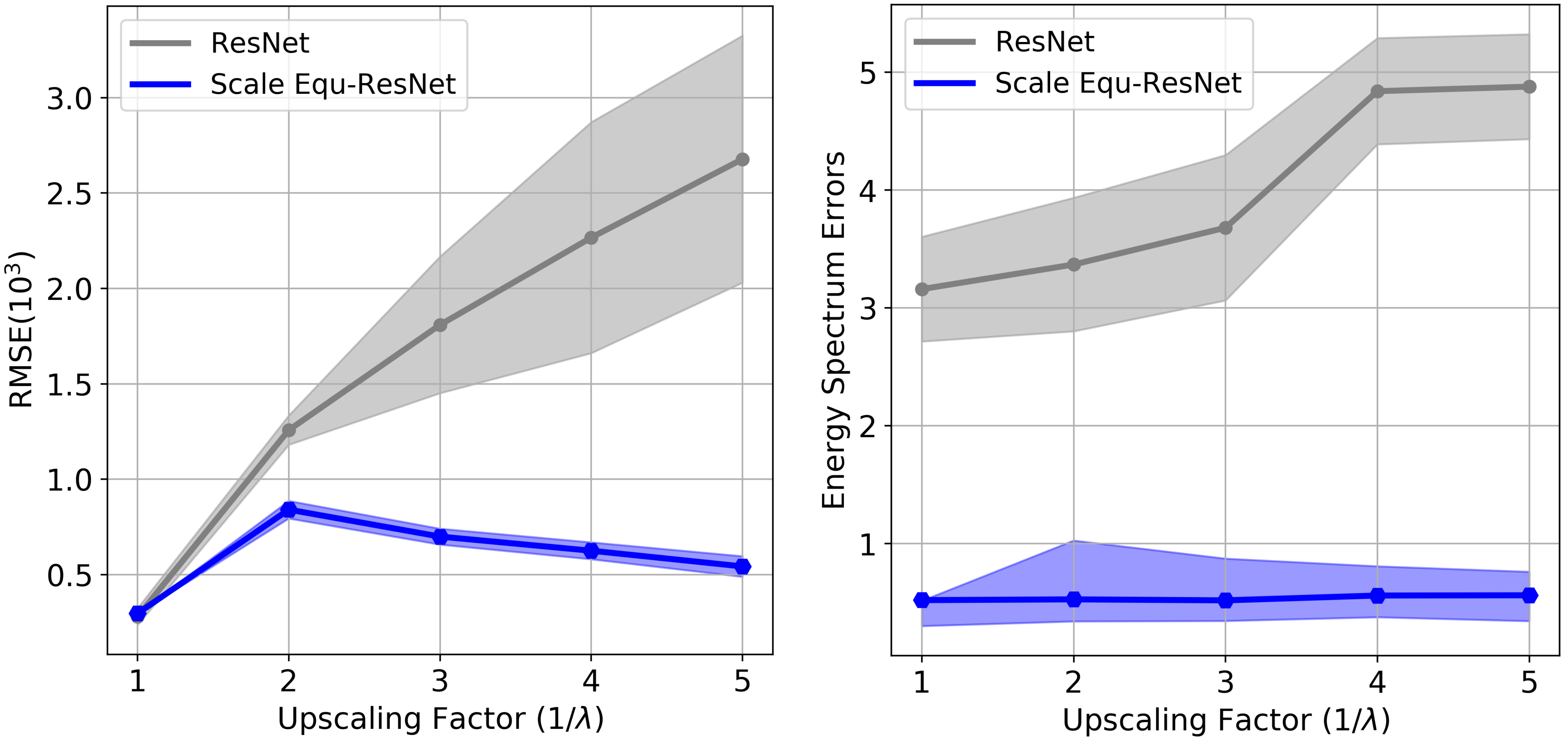


- *Scale* equivariant

$$\boxed{\boldsymbol{v}(p) = \sum_{\lambda \in \mathbb{Z}_{>0}, q \in \mathbb{Z}^2} (T_\lambda \boldsymbol{w})(p+q)(T_\lambda K)(q),}$$

$T_\lambda w(x,t) = \lambda w(\lambda x, \lambda^2 t)$

Ocean Currents Forecast



Conclusion

- Physics-Guided Deep Learning: Integrating first-principles into deep neural networks
  - **TF-Net**: Hybrid CFD-deep learning model for accelerating turbulence simulation
  - **Equ-Net**: symmetry-aware neural network for improved generalization
- Future work: flow control and optimization

Acknowledgment

Open Source Code and Data: roseyu.com

@yuqirose