UC San Diego

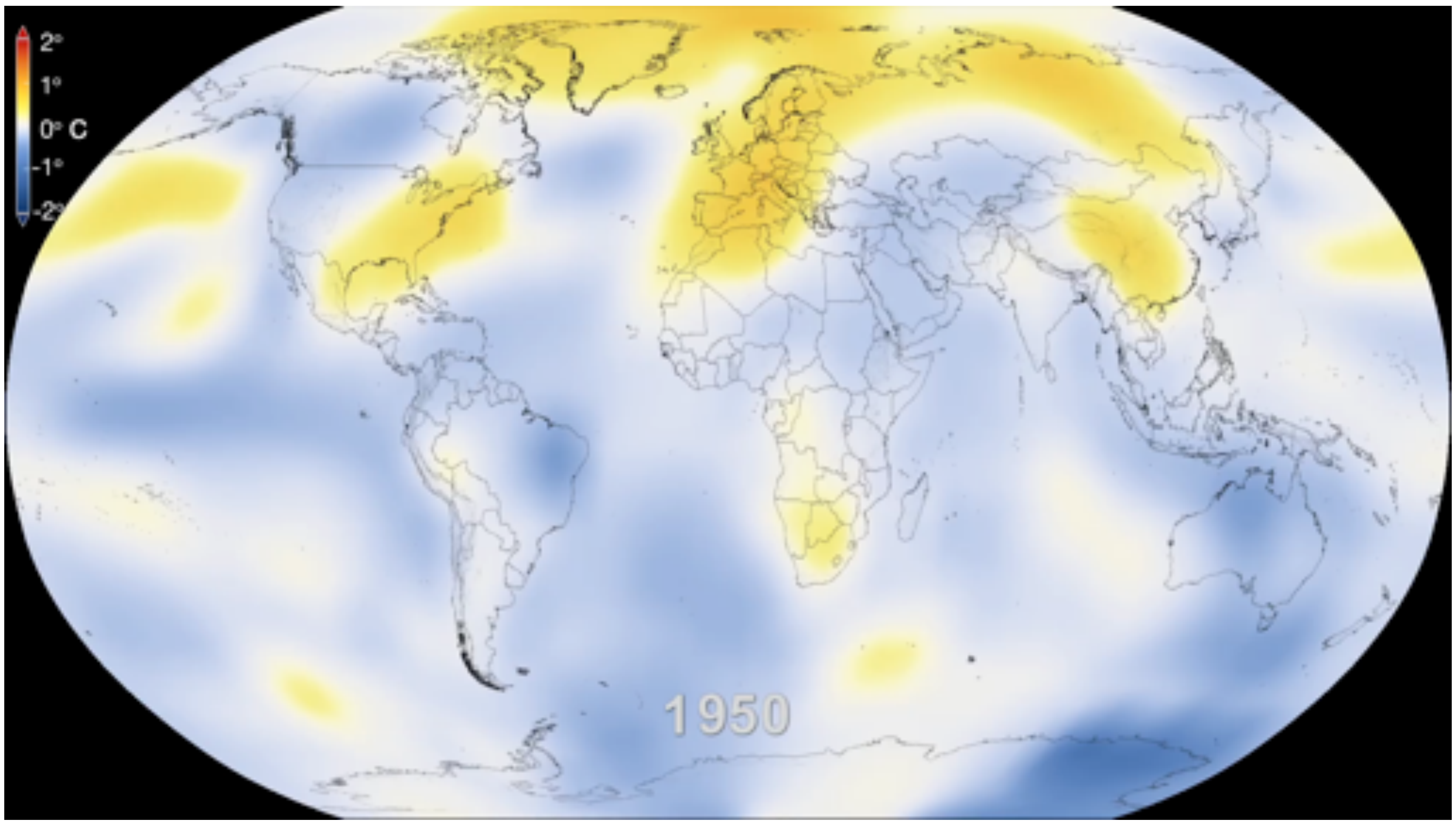
Physics-Guided AI for Learning Spatiotemporal Dynamics

Rose Yu

Assistant Professor
University of California, San Diego

Predicting Global Climate

100,000 stations, 180 countries



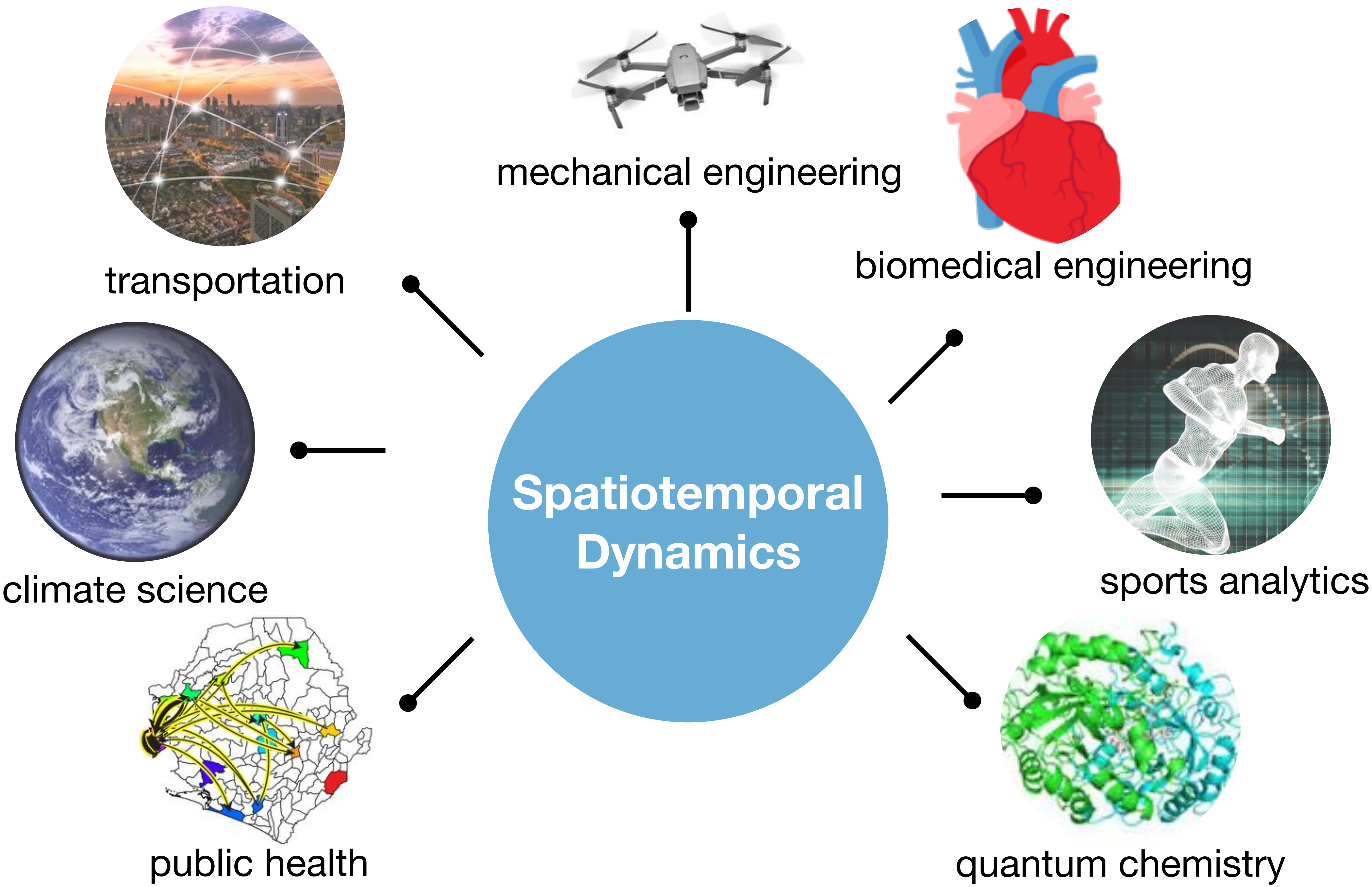
credit: NASA

Forecasting Daily Traffic

35,000 detectors, every 30 seconds

credit: Waze

Learning Spatiotemporal Dynamics



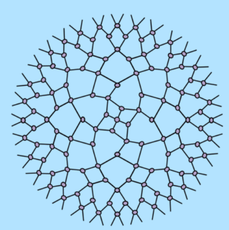
Physics-Guided AI

Physics

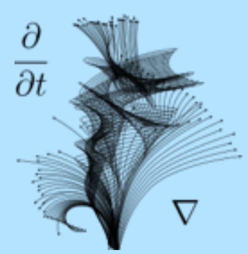
First Principles

Model-Based

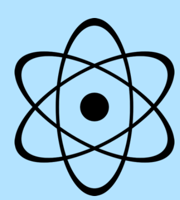
tensor network



differential equations



symmetry



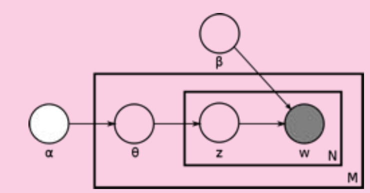
...

Learning

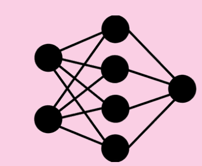
Statistical Inference

Data-Driven

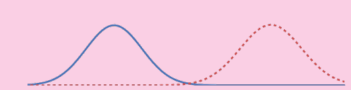
graphical model



neural networks



variational Bayes



...

+

Encode Inductive Bias

Improve Generalization

Reduce Sample Complexity

Increase **Trust** in AI

Trainable Operator

- Given input time series $(x_1, \cdots, x_t)$
- Goal: Learn a mathematical operator parameterized by deep neural nets

$$f : x_t \longrightarrow y_t$$

$$L\{f\}(x) = \int_0^\infty e^{-xt} f(t)dt$$

Trainable Weights

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

Related Work

- **Turbulence Modeling** [Ling et al. 2016, Raissi et al. 2017, Fang et al. 2018, Kim and Lee 2019, Chertkov et al. 2019, Wu et al. 2019]
    - no external force, spatial modeling
    - require boundary condition inputs
- **Fluid Animation** [Tompson et al. 2017, Chu and Thuerey, 2017, Xie et al. 2018, Thuerey et al. 2019]
    - emphasize simulation realism
    - lack physical interpretation
- **Video Prediction** [Wang et al. 2015, Finn et al. 2016, Xue et al. 2016, Denton et al. 2018]
    - complex noisy data
    - unknown physical processes

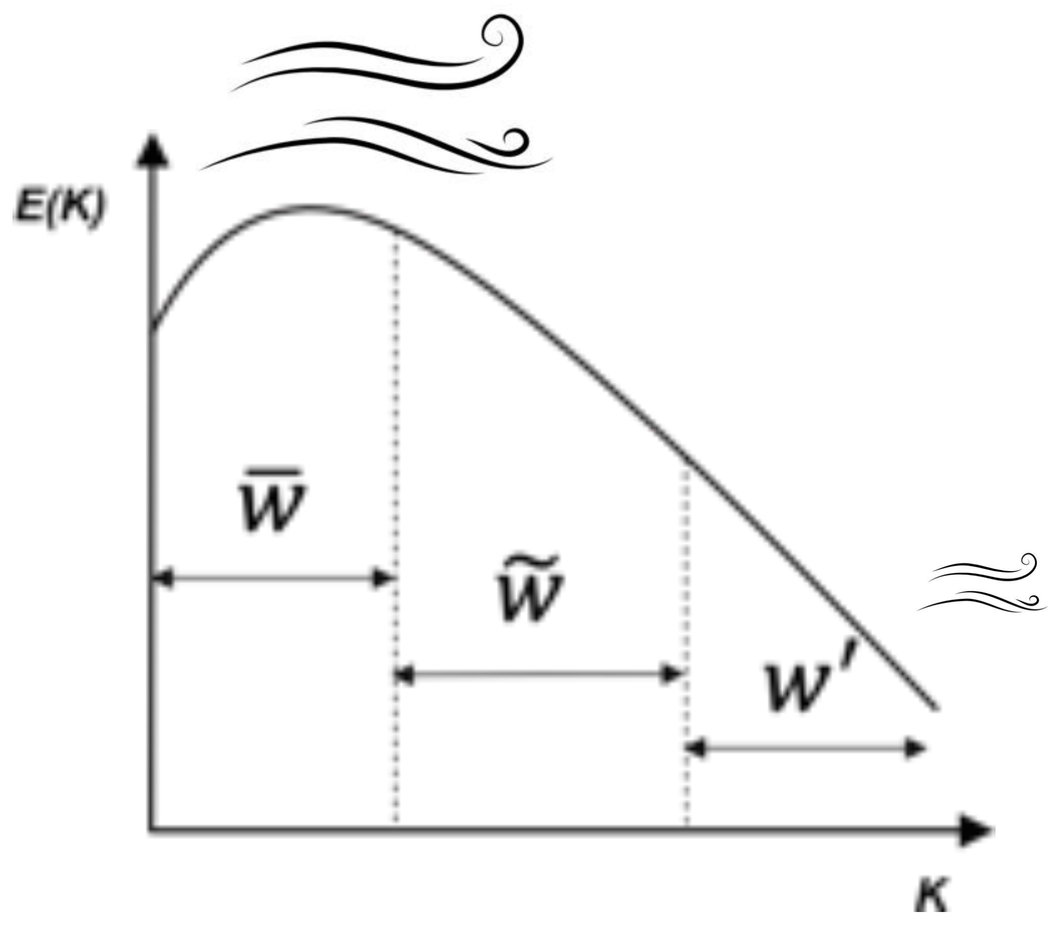
Hybrid Learning Framework

- 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)

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

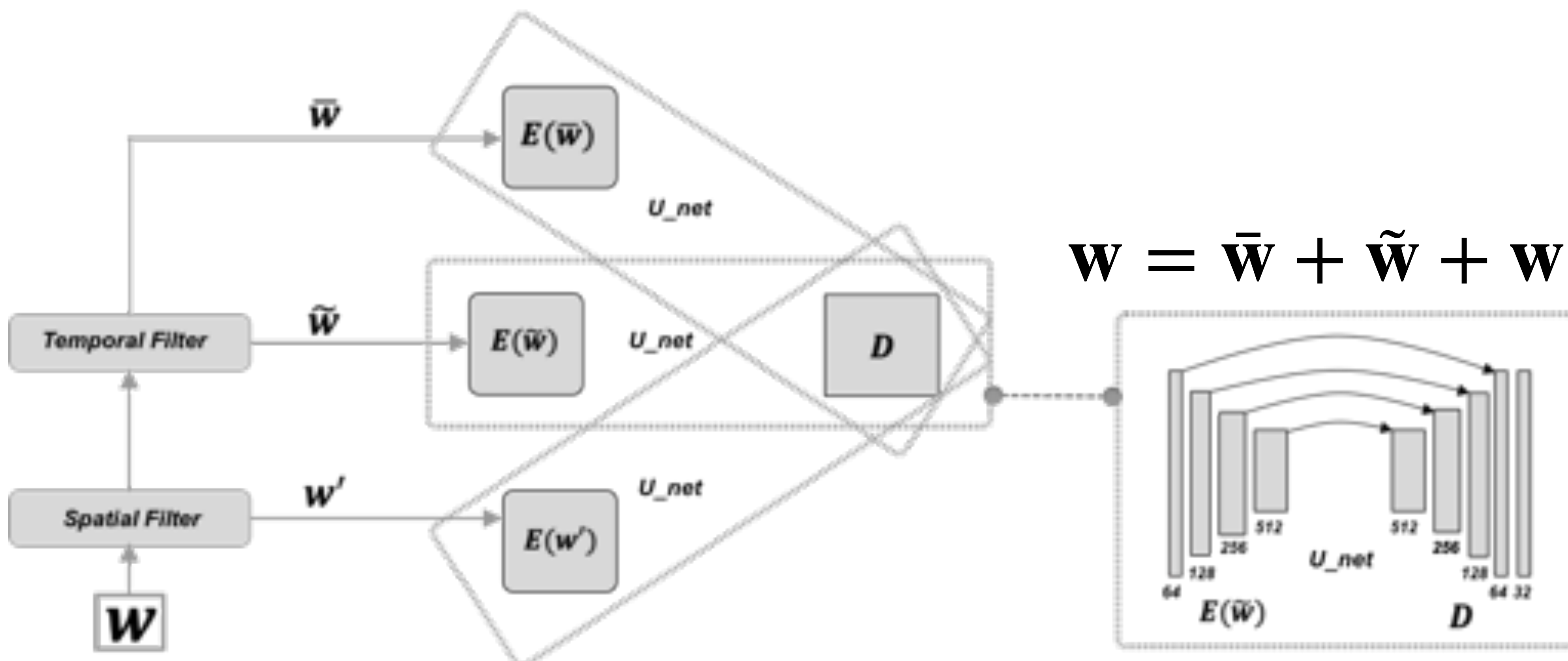


Turbulent-Flow Net

- RANS-LES Coupling

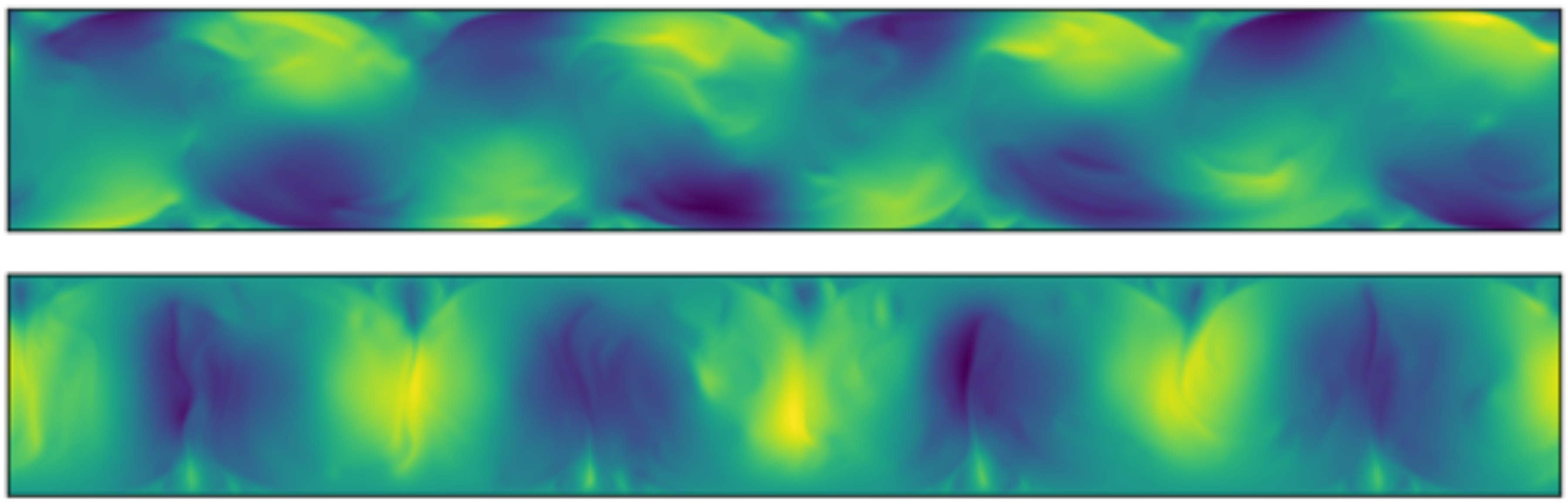
Spatial Filter **Temporal Filter**

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



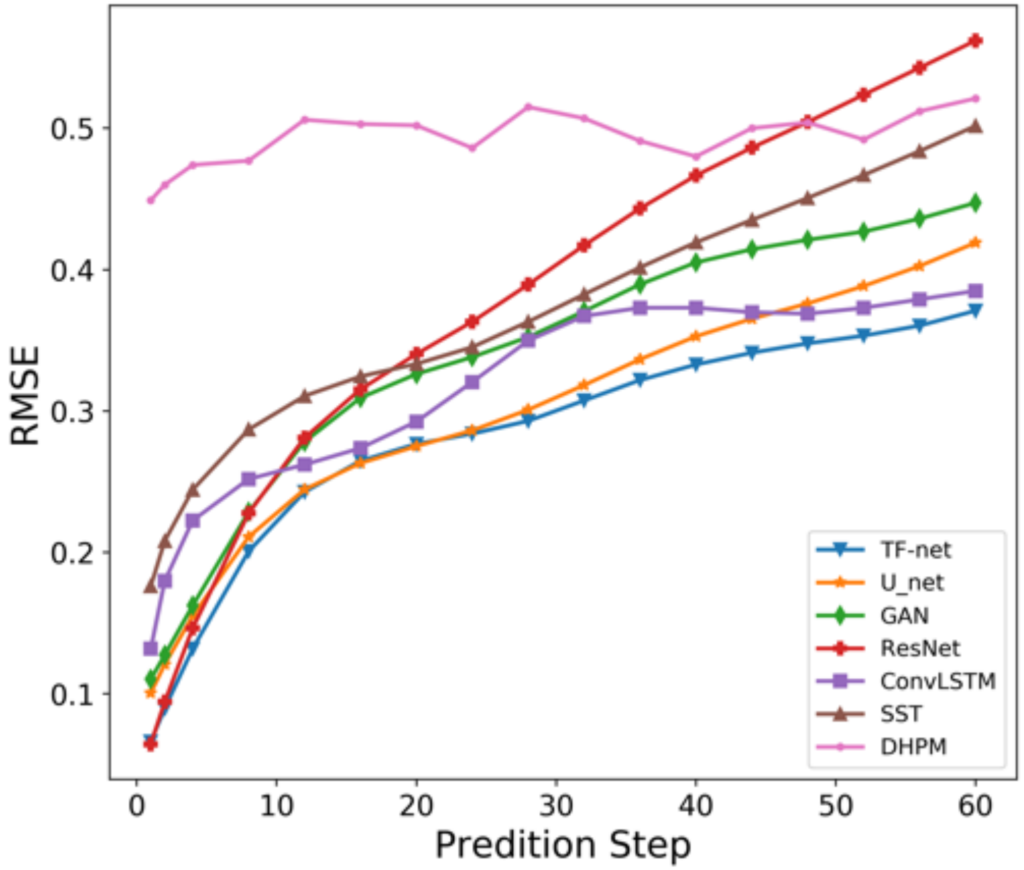
$$\mathbf{w} = \bar{\mathbf{w}} + \tilde{\mathbf{w}} + \mathbf{w}'$$

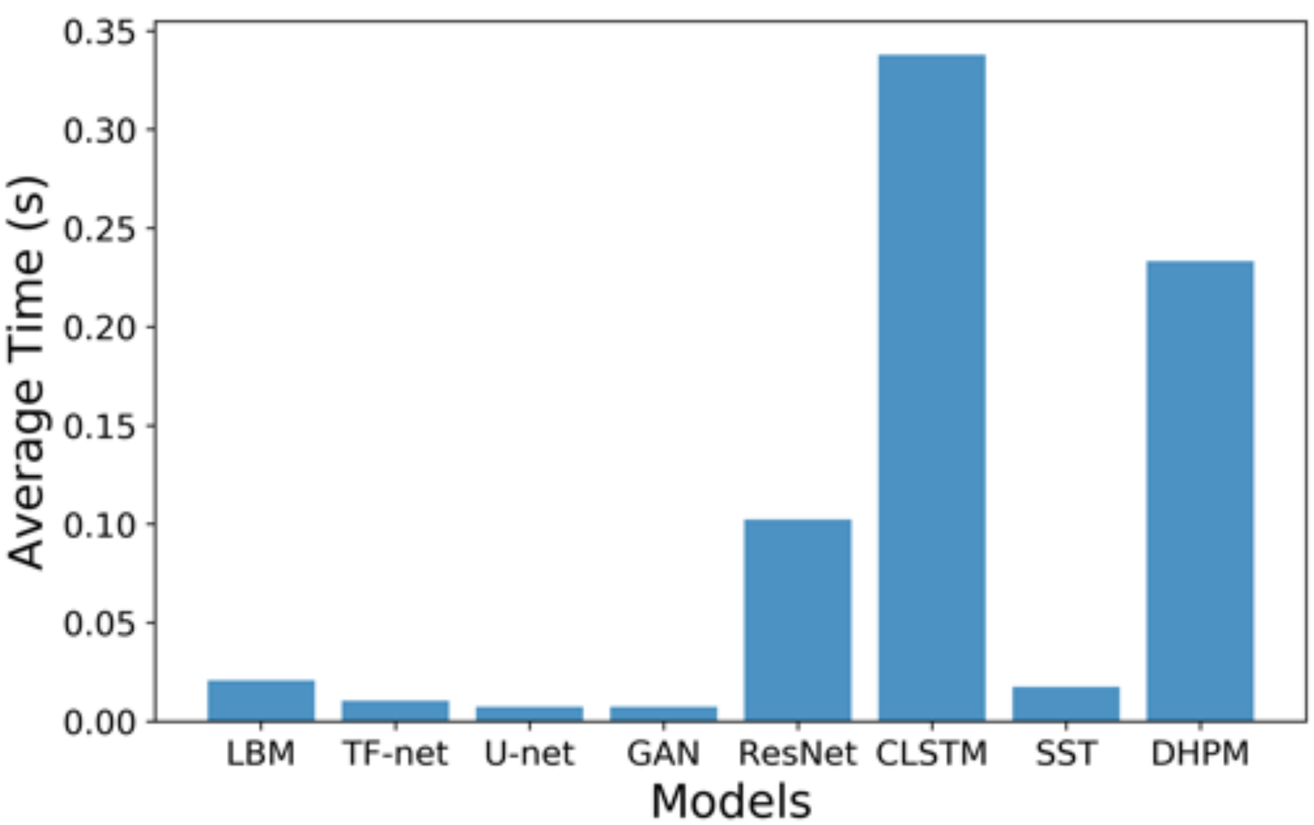
Data Description



- 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

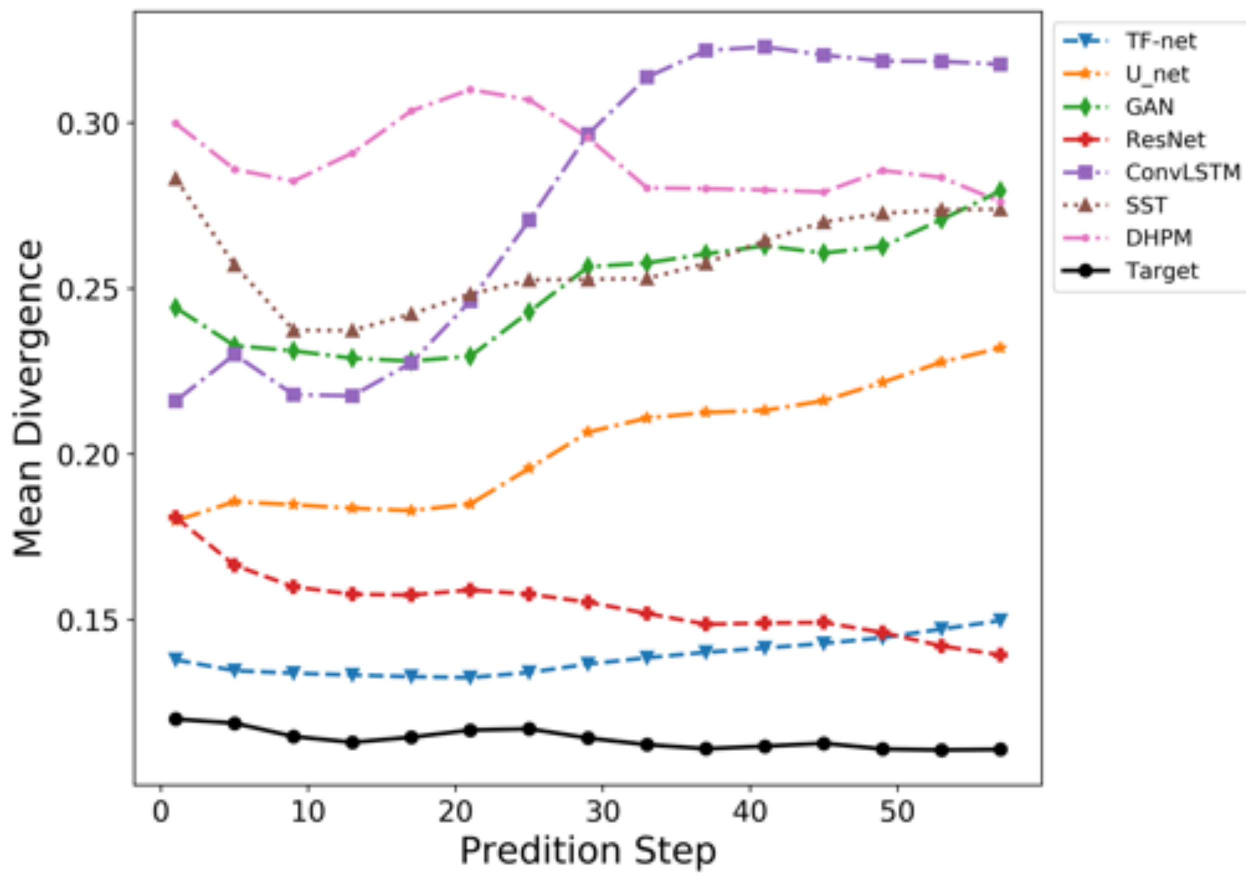
Prediction Performance

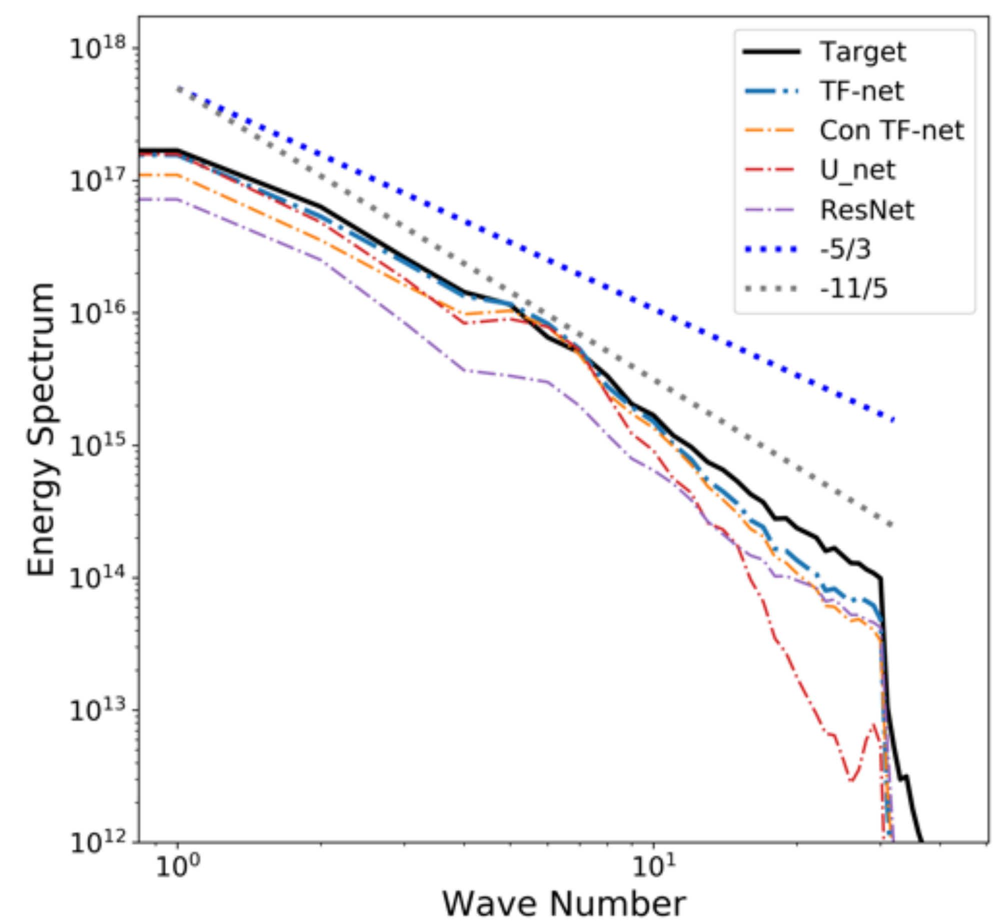




- TF-Net consistently outperforms baselines on forward prediction RMSE
- 2X faster than Lattice Boltzmann method (LBM)

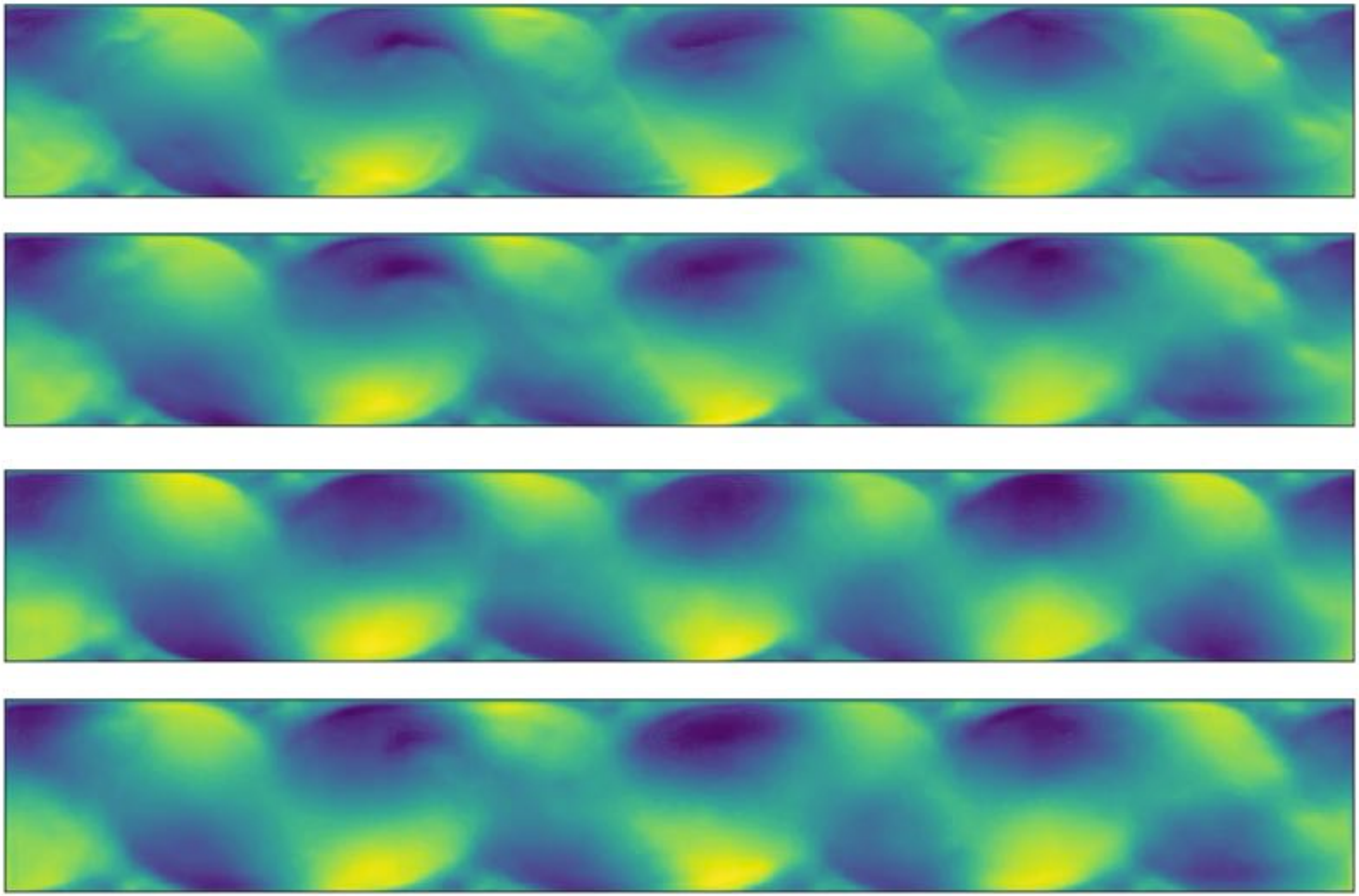
Physical Consistency





- 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



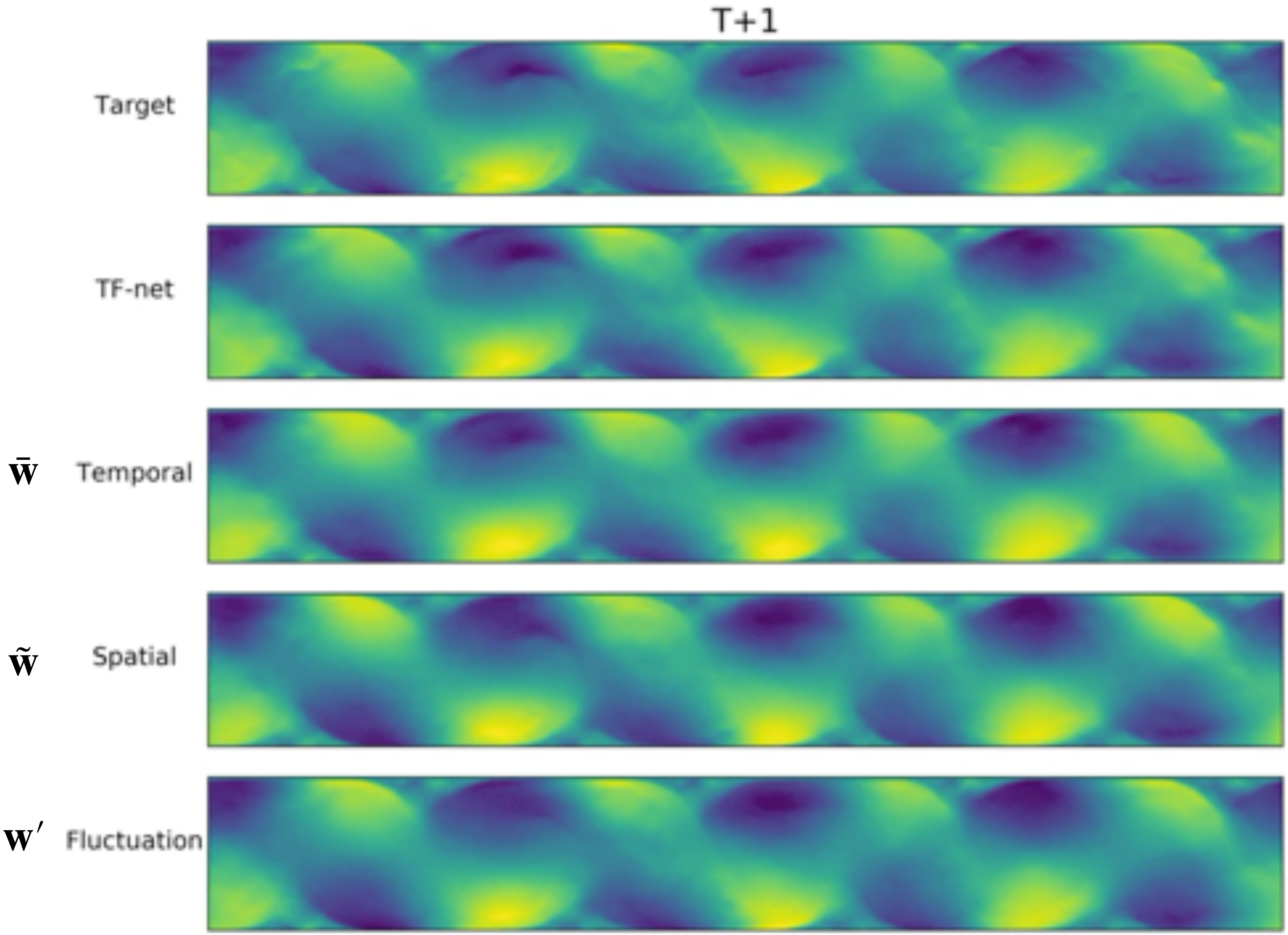
Target

TF-Net

ResNet

GAN

Ablation Study



Residual Learning

- Given input time series $(x_1, \cdots, x_t)$
- Goal: Learn a dynamics model f

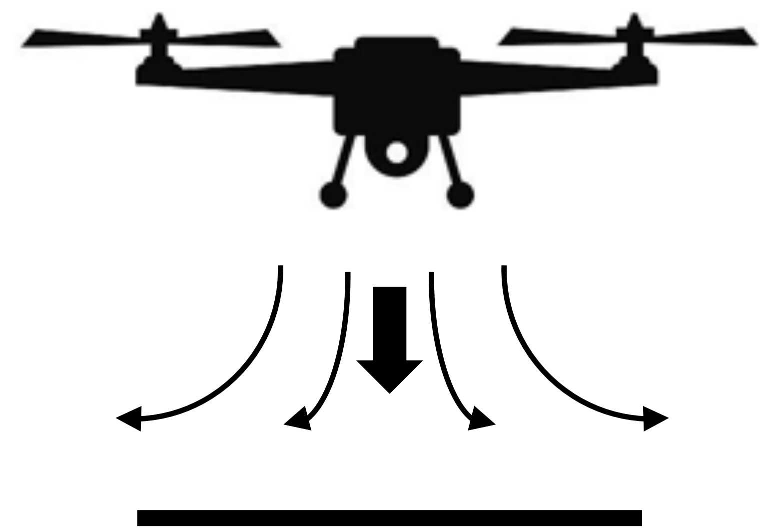
$$f : (x_0, \cdots, x_t) \longrightarrow (x_{t+1}, \cdots, x_{t+H})$$

$$f = g - r$$

g → physics-based model

r → machine learning model

Combating Ground Effect



Guanya Shi
Caltech

Kamyar Azizzadenesheli
Caltech

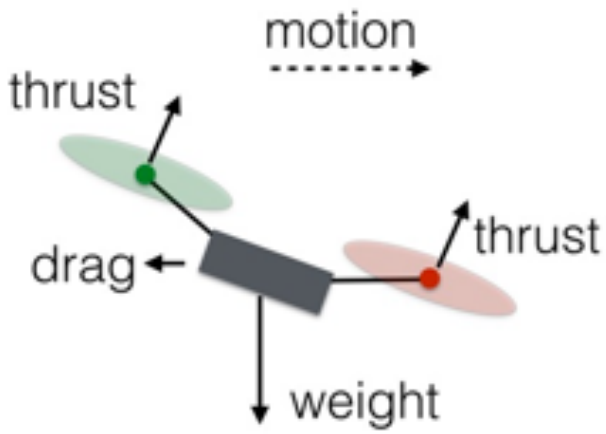
Soon-Jo Chung
Caltech

Anima Anandkumar
Caltech/NVIDIA

Yisong Yue
Caltech

Neural Lander: Stable Drone Landing Control using Learned Dynamics
Guanya Shi, Xichen Shi, Michael O'Connell, Rose Yu, Kamyar Azizzadenesheli, Animashree Anandkumar, Yisong Yue, and Soon-Jo Chung
International Conference on Robotics and Automation (ICRA), 2019

Hybrid Learning Framework



Position: p Velocity: v Angular Velocity: ω

Total Thrust, Torque: $\mathbf{f}_u, \tau_u$

Unknown Disturbance Force, Torque: $\mathbf{f}_a, \tau_a$

States: $\mathbf{s}_{t-1}$

+

Controls: $\mathbf{u}_{t-1}$

→

$$\dot{\mathbf{p}} = \mathbf{v}$$
$$m\dot{\mathbf{v}} = m\mathbf{g} + R\mathbf{f}_u + \mathbf{f}_a$$
$$\dot{R} = RS(\omega)$$
$$J\dot{\omega} = J\omega \times \omega + \tau_u + \tau_a$$

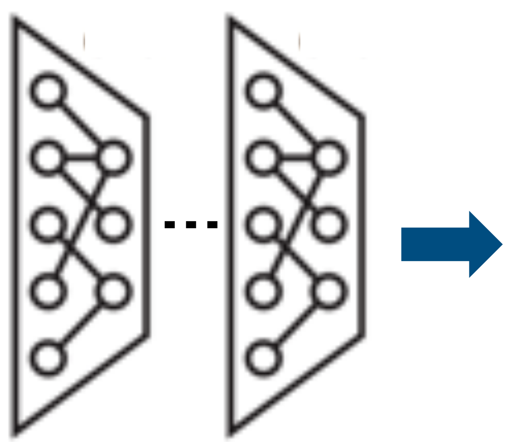
DNN with Spectral Normalization

→

New States: $\mathbf{s}_t$

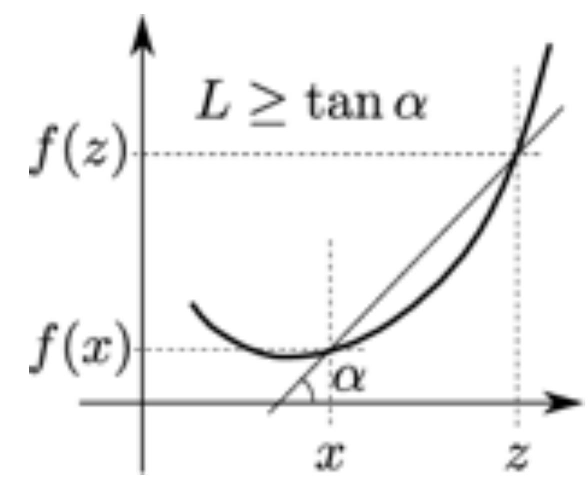
Learning Stable Dynamics

- **Spectral Normalization**: constrain the Lipschitz constant



$$f(\mathbf{x}) = g^L \circ g^{L-1} \cdots g^1(\mathbf{x})$$

$$g^L(x) = \phi(W^L x)$$



Approximate the Lipschitz constant

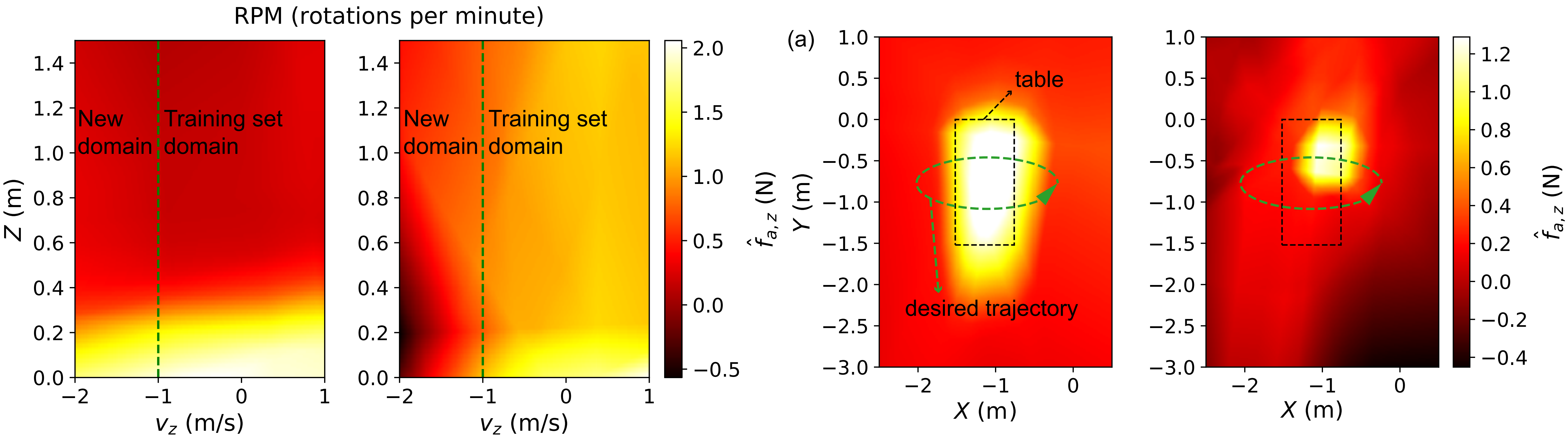
$$\|f\|_{Lip} \leq \|g^L\|_{Lip} \cdot \|\phi\|_{Lip} \cdots \|g^1\|_{Lip}(\mathbf{x}) = \prod_{l=1}^{L} \sigma(W^l)$$

Normalize the weights of a DNN by their singular values

$$\bar{W} = W/\sigma(W)$$

Combat Ground Effect

Spectral Normalization



- Spectrally normalized DNNs **generalize** well [Bartlett et al. 17], which is an indication of **stability** in machine learning

Equivariant Learning

- **Noether's theorem**: *For every symmetry, there is a corresponding conservation law.*

- Learn a function f that is G-equivariant w.r.t group G

$$f(\rho(g)x) = \rho'(g)f(x)$$

Sample Efficient Trajectory Prediction

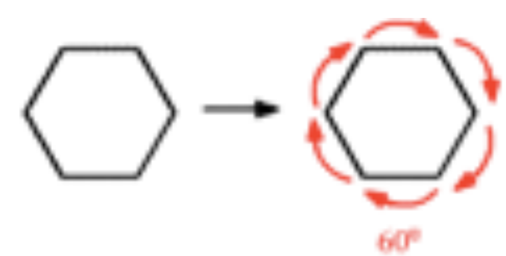
Jinxi (Leo) Li

Robin Walters

Trajectory Prediction using Equivariant Continuous Convolution
Walters, Robin, Jinxi Li, and Rose Yu.
International Conference on Learning Representations (ICLR), 2021.

Symmetry

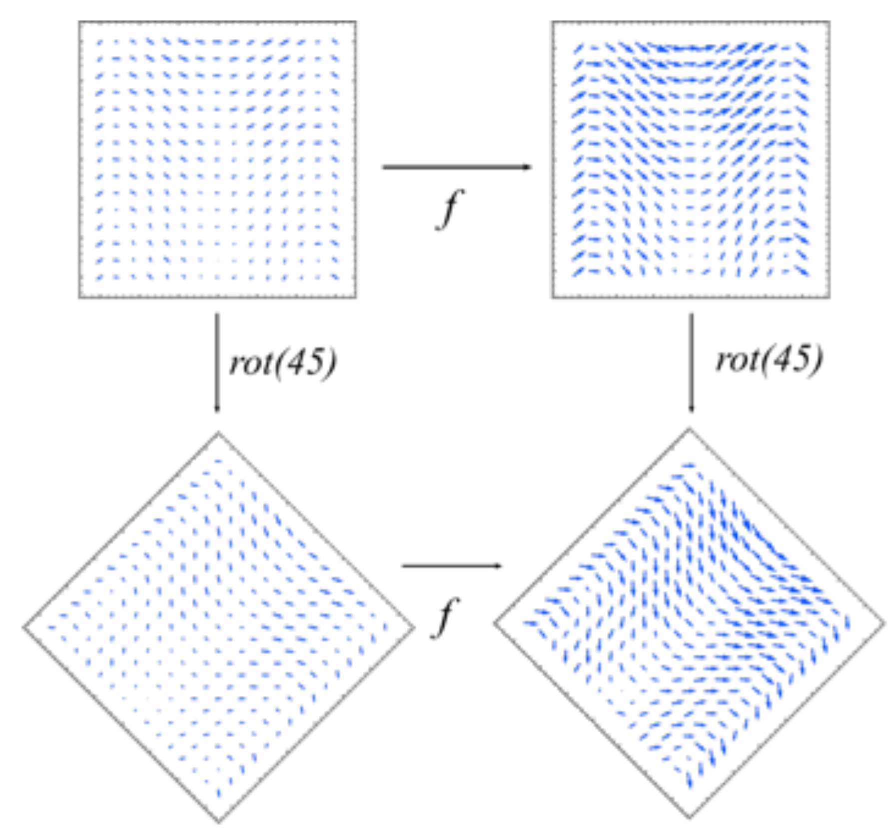
- **Group**: a set G and a composition map $\circ : G \times G \to G$
  - $1 \in G$ and $\forall g \in G, \exists g^{-1} \in G$
  - SO(2): 2d rotation



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

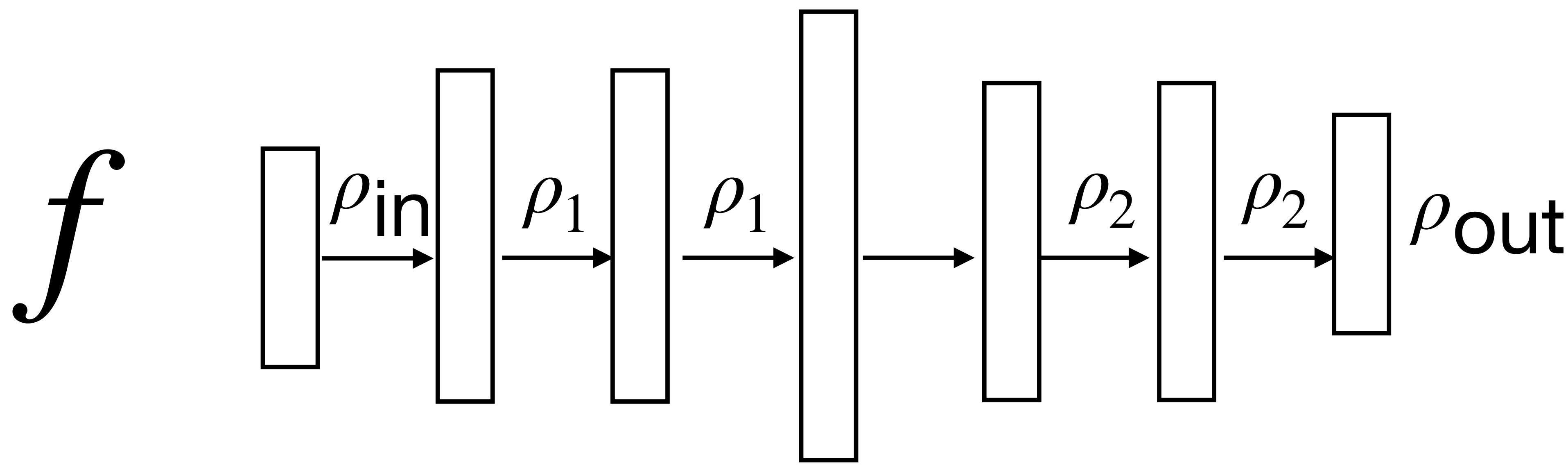
$$f(x, v) = (x, 2v)$$

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



Equivariant Networks

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

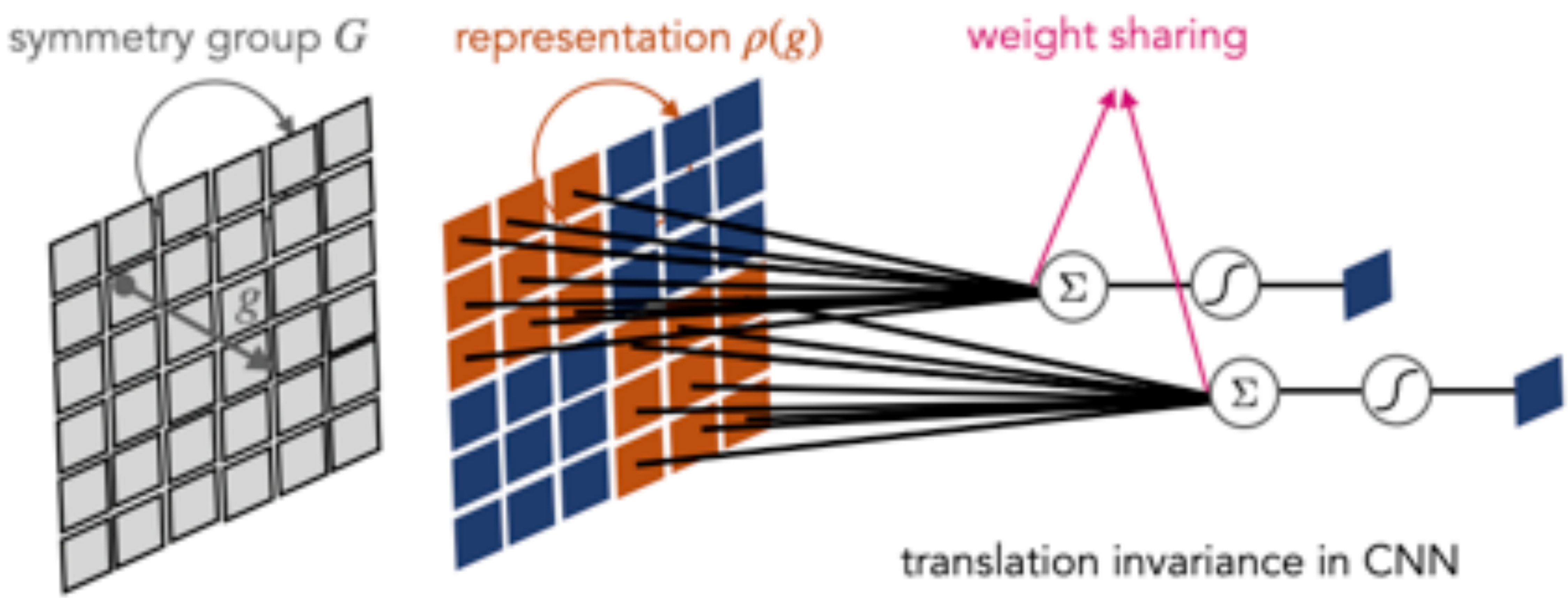


Proposition: Let the layer $V^{(i)}$ be a G-representation for $0 \leq i \leq n$. Let $f^{(ij)} : V^{(i)} \to V^{(j)}$ be G-equivariant for $i < j$. Define recursively $x^{(j)} = \sum_{0 \leq i \leq j} f^{(ij)}(x^{(i)})$, then $x^{(n)} = f(x^{(0)})$ is G-equivariant.

- If the maps between layers are equivariant, then the entire network is equivariant.
- Adding skip connections does not affect its equivariance with respect to 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} .

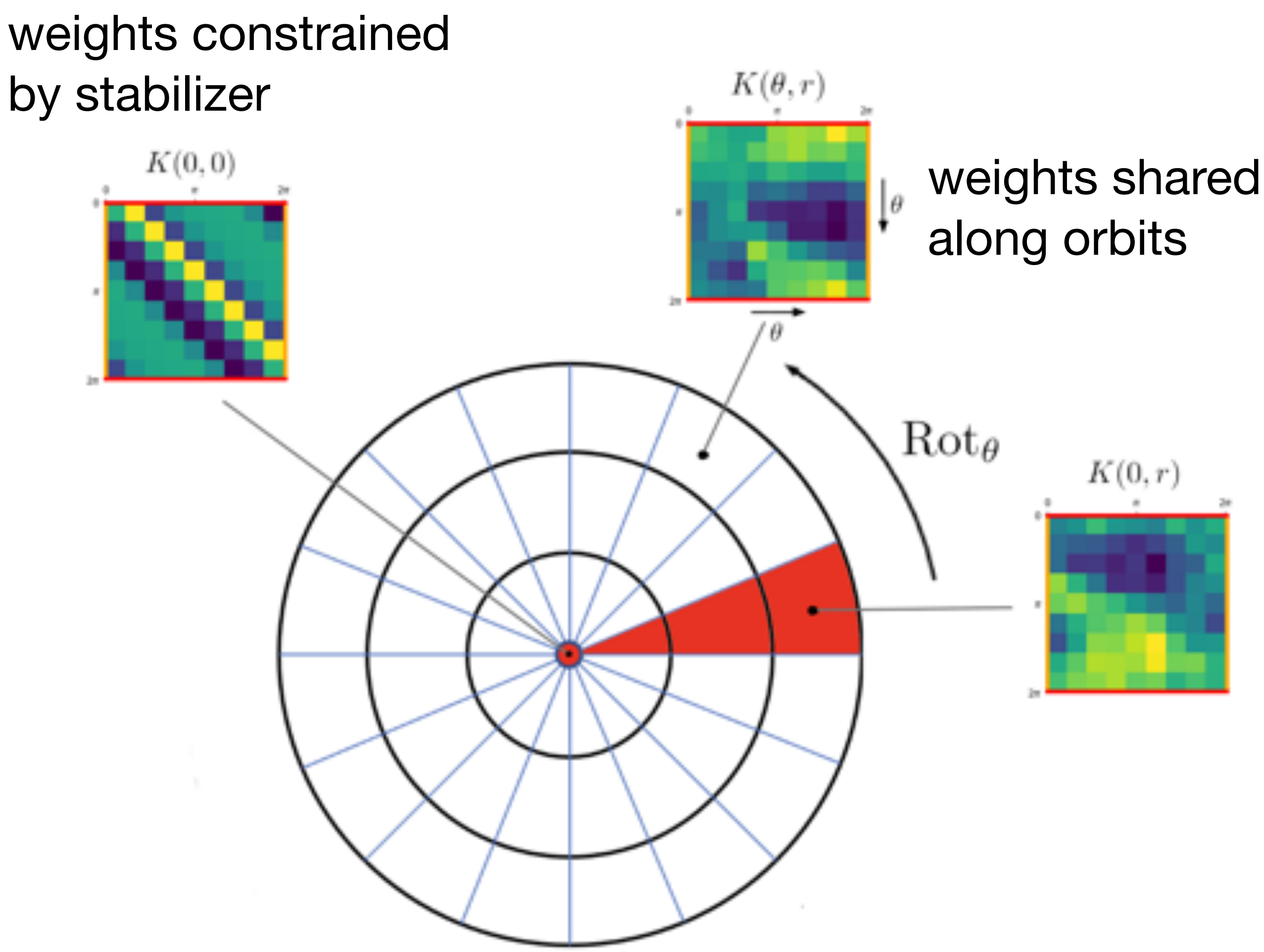


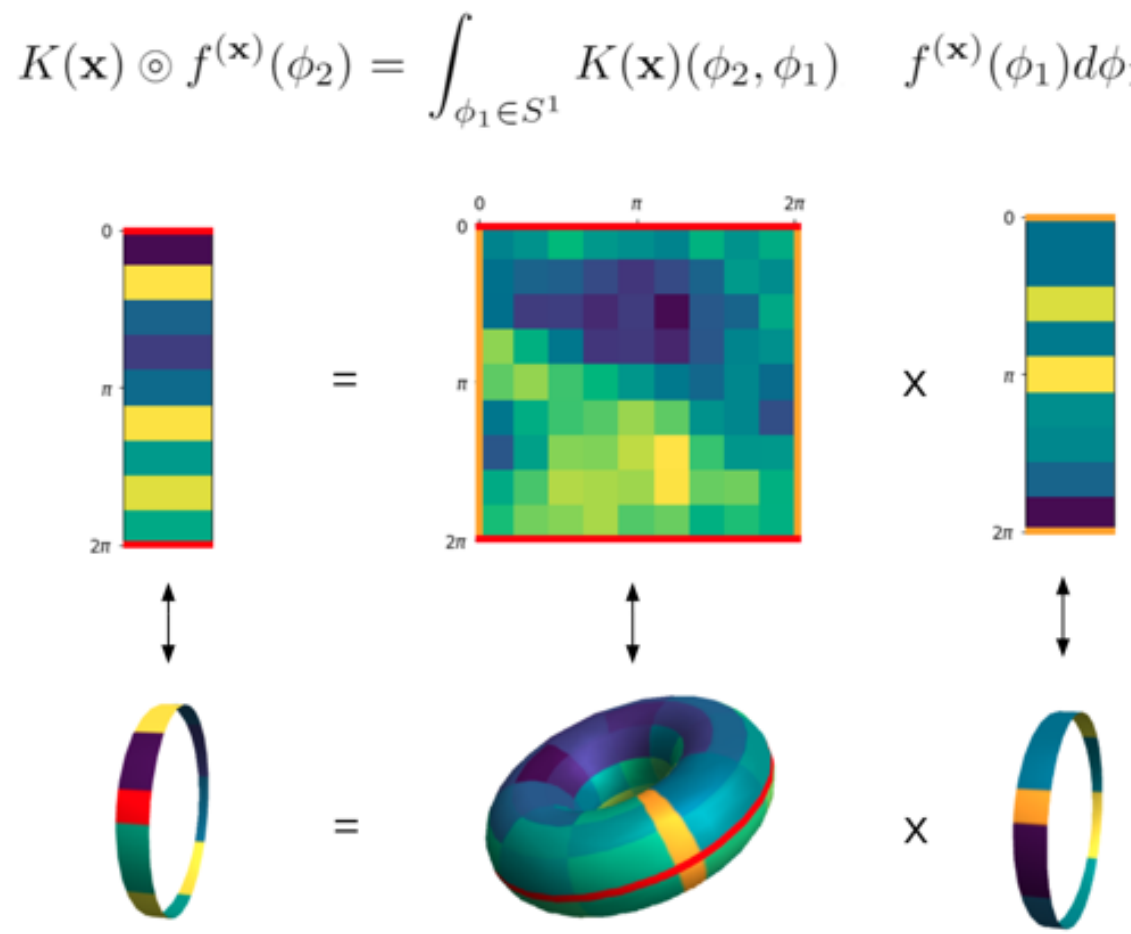
Rotation Symmetry

- Traffic dynamics resembles driven many-particle systems [Helbing 2000]
- Implicit rotation symmetry in vehicles
- Expect consistent predictions with different orientations

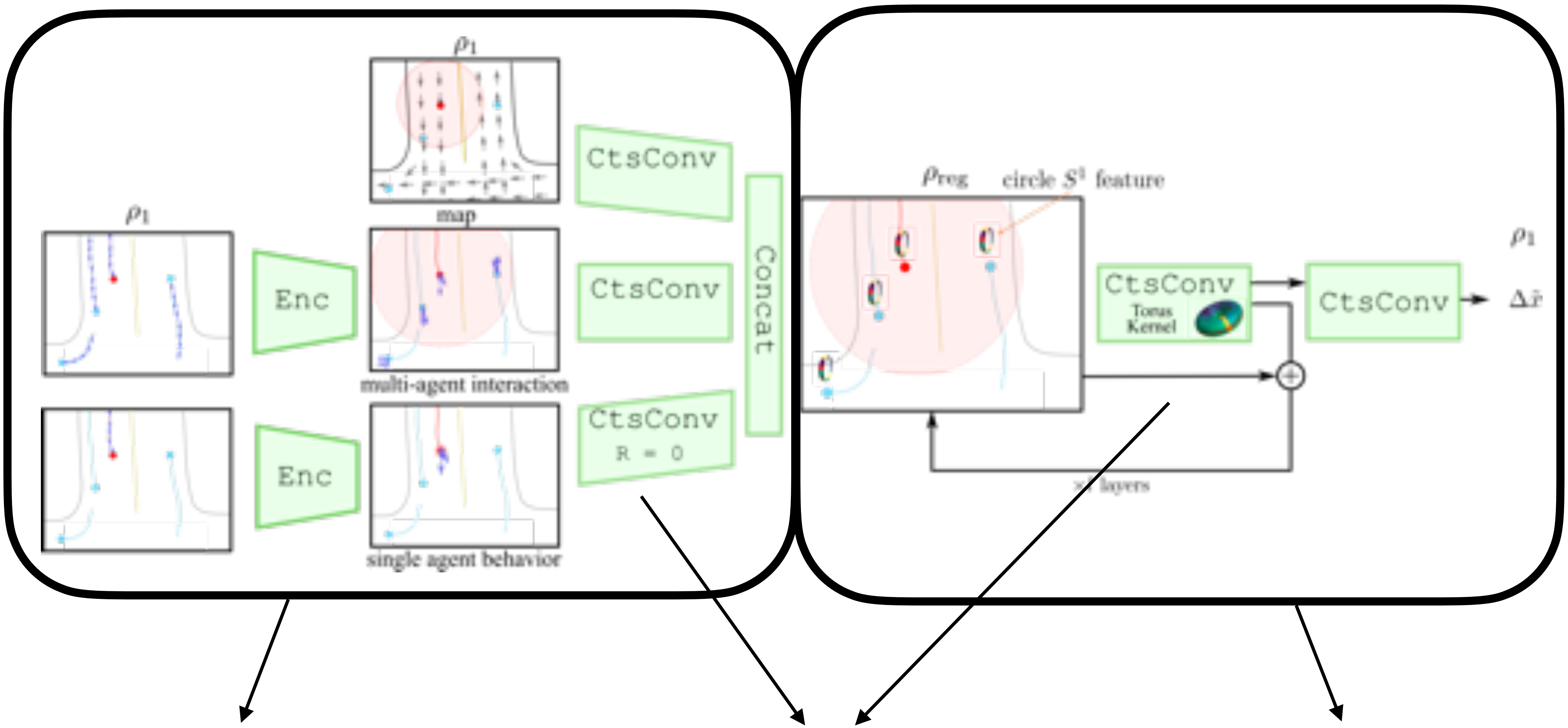
Equivariant Continuos Convolution (ECCO)

$$K(\theta+\phi, r) = \rho_{\text{out}}(\text{Rot}_\theta)K(\phi, r)\rho_{\text{in}}(\text{Rot}_\theta^{-1})\,.$$





ECCO



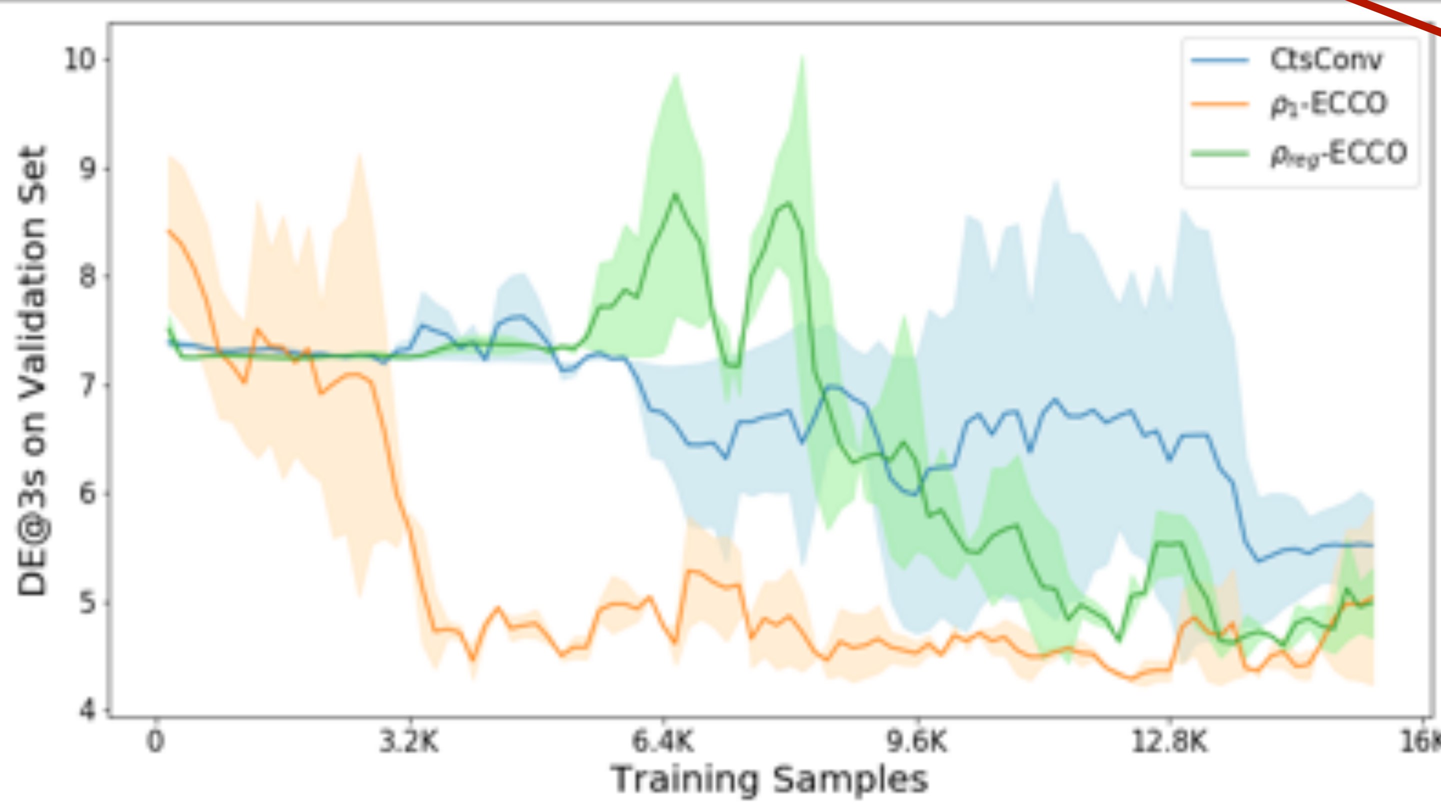
Encode Past

Continuous Convolution

Decode Future

Performance Comparison

Model	Argoverse ADE	Argoverse DE@1s	Argoverse DE@2s	Argoverse DE@3s	TrajNet++ ADE	TrajNet++ FDE	#Param
Constant Velocity	3.86	2.43	5.10	7.91	1.39	2.86	-
Nearest Neighbor	3.49	2.02	4.98	7.84	1.38	2.79	-
LSTM	2.13	1.16	2.81	4.83	1.11	2.03	50.6K
CtsConv	1.85	0.99	2.42	4.32	0.86	1.79	1078.1K
ρ_1-ECCO	1.70	0.93	2.22	3.89	0.88	1.83	51.4K
ρ_{reg}-ECCO	**1.62**	**0.89**	**2.12**	**3.68**	**0.84**	**1.76**	129.8K
VectorNet	1.66	0.92	2.06	3.67	-	-	72K + Decoder



95% params reduction

80% data reduction

Conclusion

- Incorporating Physical Principles in Deep Dynamics Models
  - **Trainable Operator**: replacing mathematical operators with trainable weights
  - **Residual Learning**: learning the correction terms of the physics-based models
  - **Equivariant Learning**: incorporating symmetry to guarantee laws of conservation
- Future Work
  - Stochastic dynamics and multi-agent interactions

“Time and space are not conditions of existence, time and space is a model of thinking.”

–Albert Einstein

Acknowledgment

Open Source Code and Data: roseyu.com

@yuqirose