A Bilevel Optimization Framework for Training Bregman Neural Operators

Abdel-Rahim MEZIDI[1], Rémi EMONET[1,2], Jordan FRECON[1],
Amaury HABRARD[1,2], Saverio SALZO[3], Marc SEBBAN[1]

[1]Université Jean Monnet Saint-Etienne, CNRS, Institut d Optique Graduate School, Inria, Laboratoire Hubert Curien
UMR 5516, F-42023, SAINT-ETIENNE, France

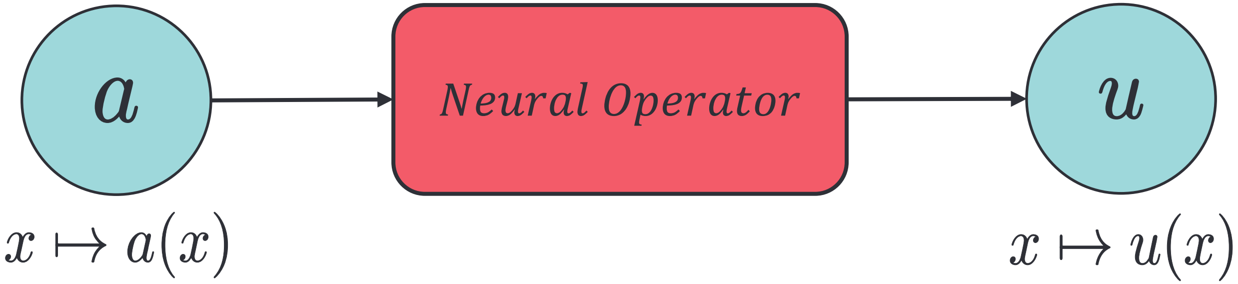
[2]Institut Universitaire de France (IUF)

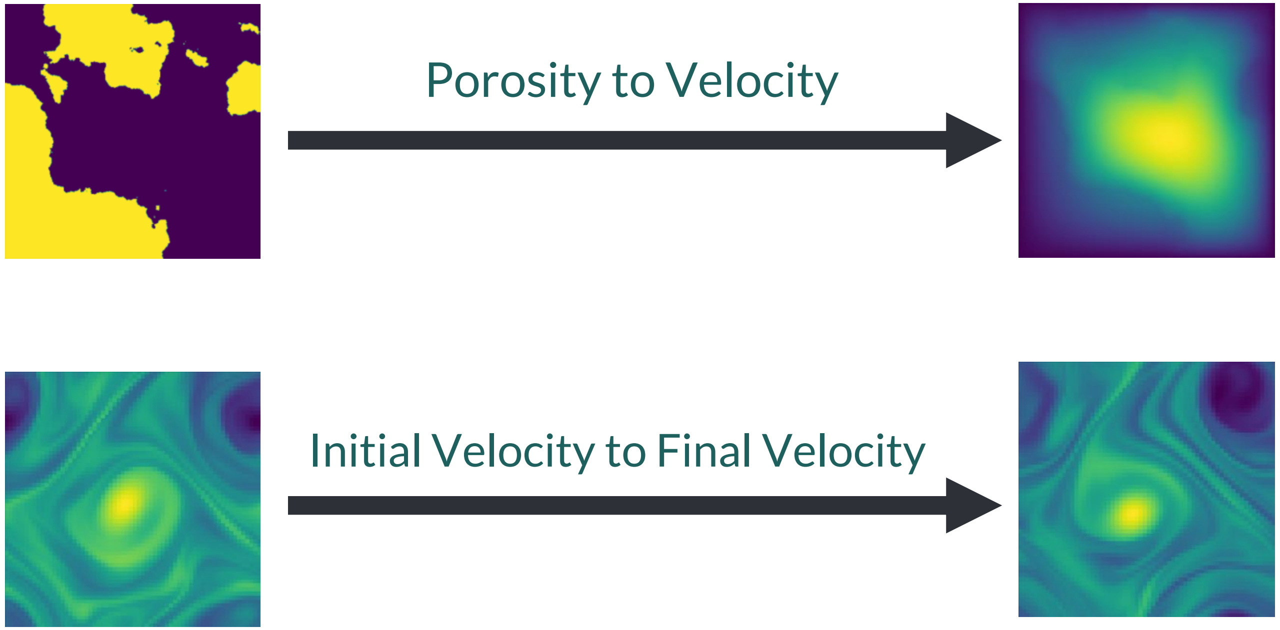
[3]DIAG, Sapienza University of Rome, 00185 Rome, Italy

Université Jean Monnet Saint-Étienne
cnrs
INSTITUT d'OPTIQUE GRADUATE SCHOOL ParisTech
Inria
Laboratoire Hubert Curien UMR • CNRS • 5516 • Saint-Étienne

Context: Neural Operators

Learn mappings between function spaces





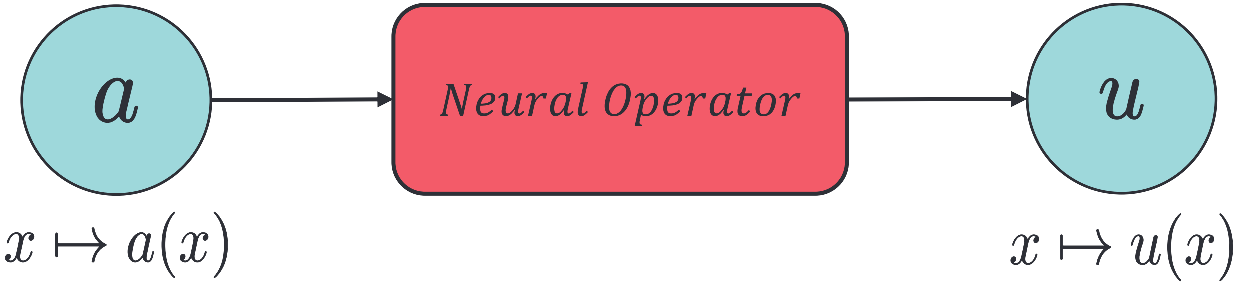
Context: Neural Operators

Physics-Informed Neural Network

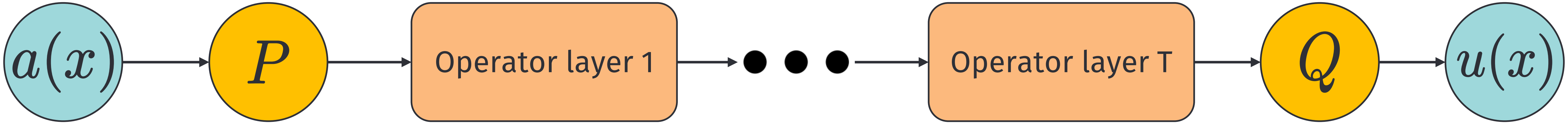
- Solve one instance of the PDE
- No data required, only PDE

Neural Operator

- Solve a family of PDEs
- No PDE required, only data



Architecture of Neural Operators

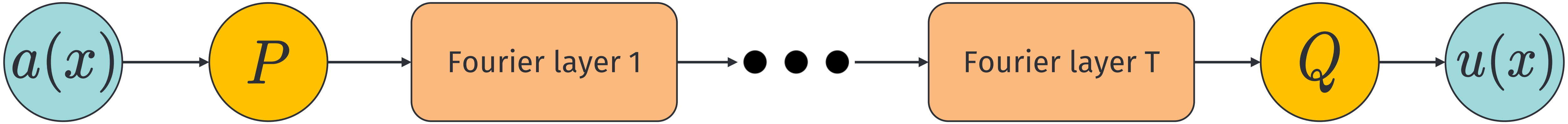


Lifting layer P

T Operator layers

Projection layer Q

Architecture of Fourier Neural Operators



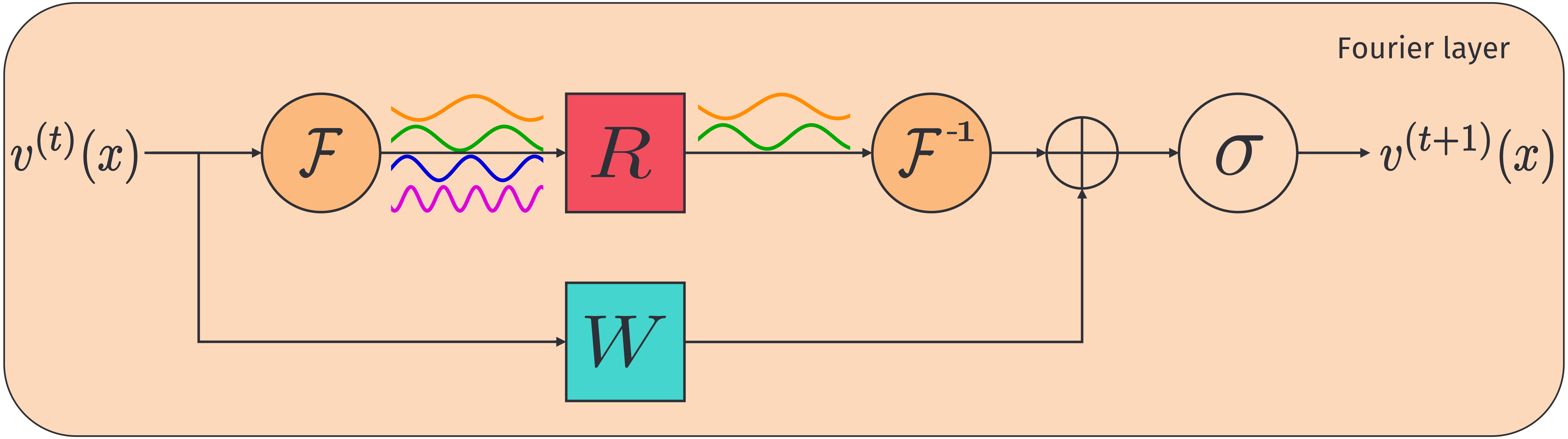
Lifting layer P

T Fourier layers

Projection layer Q

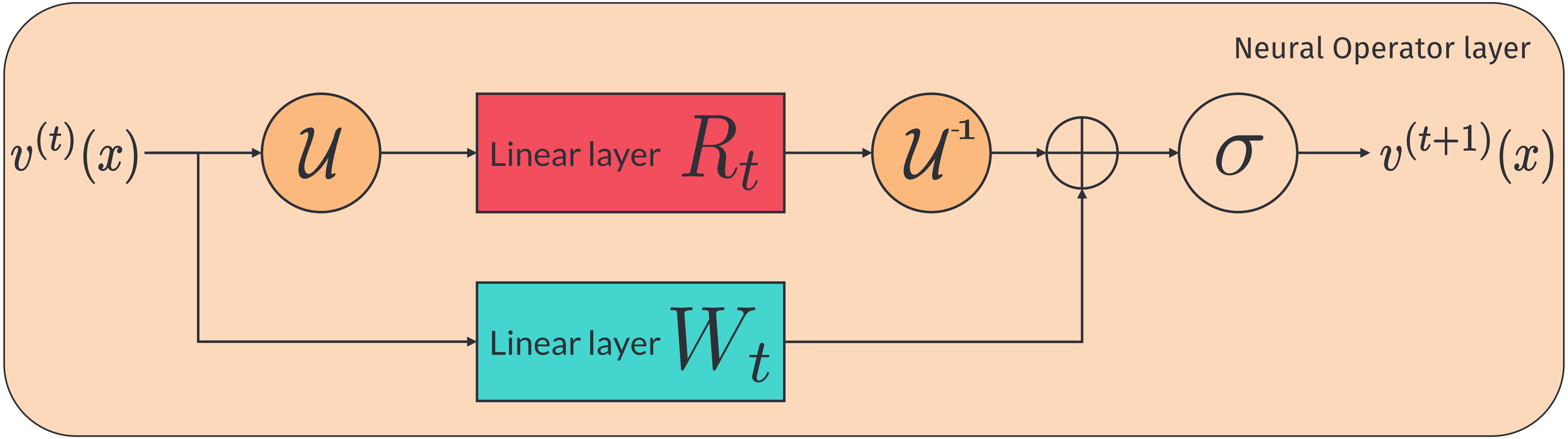
1: Li, Z. et al., Fourier Neural Operator for Parametric Partial Differential Equations, ICLR 2021

Architecture of Fourier Neural Operators



- Fourier transform $\mathcal{F}$
- Linear transform of the Fourier modes $\mathcal{R}$
- Inverse Fourier transform $\mathcal{F}^{-1}$
- Linear transform W
- Activation function σ

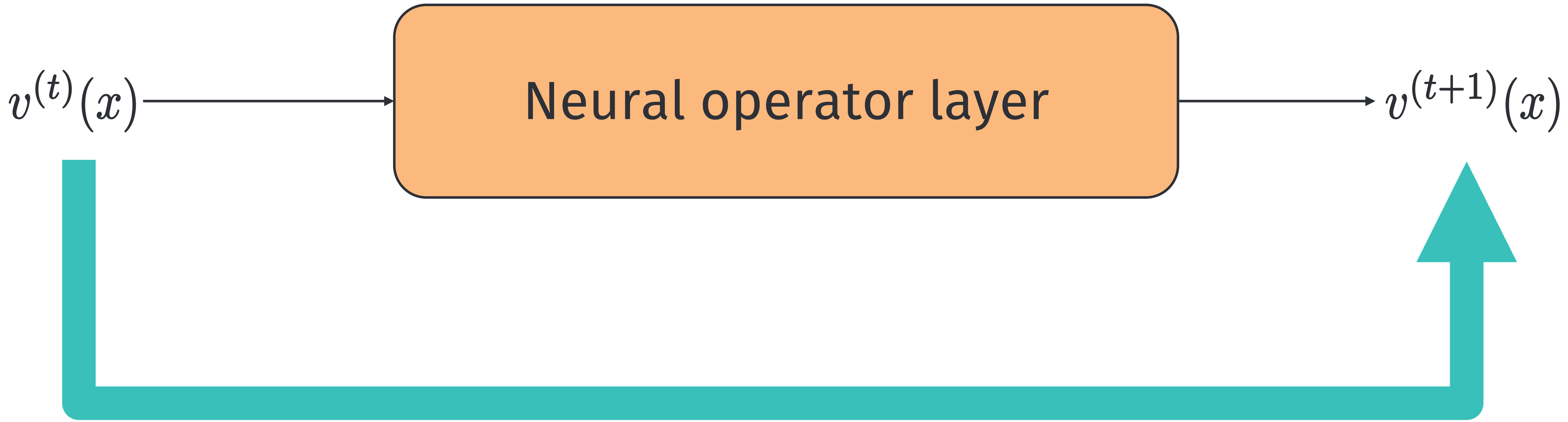
General architecture



- $\mathcal{U}$ and $\mathcal{U}^{-1}$ a unitary operator and its inverse
  - Fourier for FNO
  - Wavelet for WNO[1]
  - …

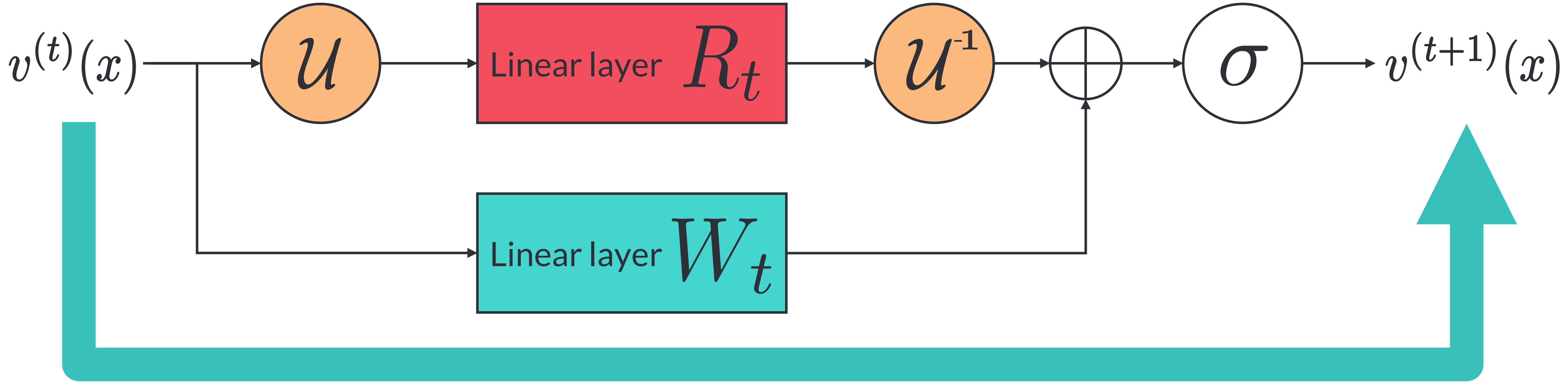
1: Tripura, T., & Chakraborty, S. (2023). Wavelet neural operator for solving parametric PDE in computational mechanics problems

Contribution: Proximal Optimization Viewpoint



$$v^{(t+1)} = \underset{v}{\operatorname{argmin}}\, \mathcal{L}(v, v^{(t)})$$

Contribution: Proximal Optimization Viewpoint



$$v^{(t+1)} = \underset{v}{\operatorname{argmin}}\ f_t(\mathcal{U}(v), \mathcal{U}(v^{(t)})) + g_t(v, v^{(t)}) + D(v, v^{(t)}) + h(v)$$

Learnable parameters: f_t, g_t

Fixed: Divergence D, regularizer h

Connections with Neural Operators

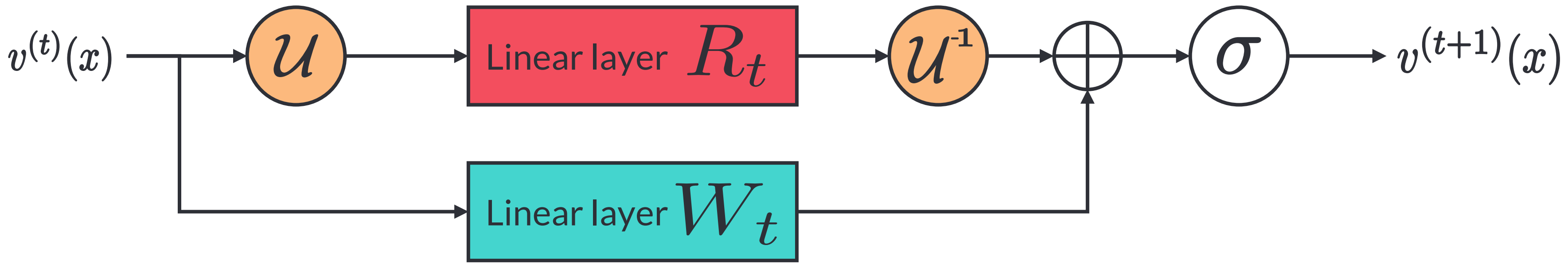
If f_t and g_t are *bi-linear*, D a *quadratic distance* and h a *convex function*, **Then:**

$$\nabla f_t \iff R_t$$

$$\nabla g_t \iff W_t$$

Proximal operator of h $\iff$[1] σ

$\Longrightarrow$ **Neural Operator layer**



$$v^{(t+1)} = \underset{v}{\operatorname{argmin}}\ f_t(\mathcal{U}(v), \mathcal{U}(v^{(t)})) + g_t(v, v^{(t)}) + D(v, v^{(t)}) + h(v)$$

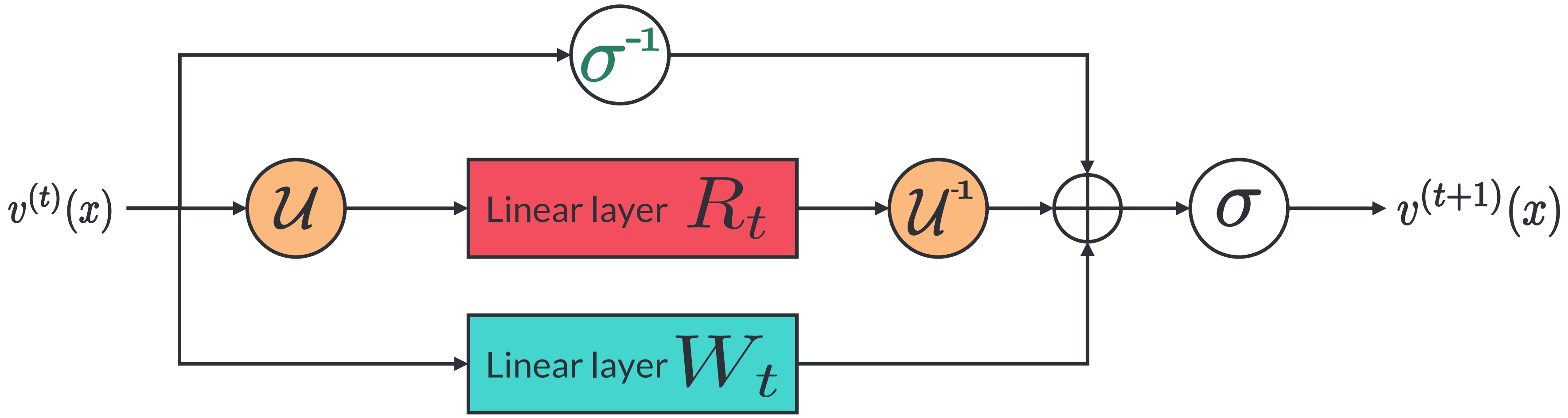
1: Combettes, P. L. and Pesquet, J.-C. (2020). Deep neural network structures solving variational inequalities. Set-Valued and Variational Analysis

New architecture: Bregman Neural Operators

If f_t and g_t are *bi-linear*, D a *Bregman divergence* and h a *convex function*, **Then:**

∇f_t	$\Longleftrightarrow$	R_t	*Proximal operator of h with Bregman divergence*	$\Longleftrightarrow$[1]	σ
∇g_t	$\Longleftrightarrow$	W_t	*Bregman divergence*	$\longleftrightarrow$	σ^{-1}

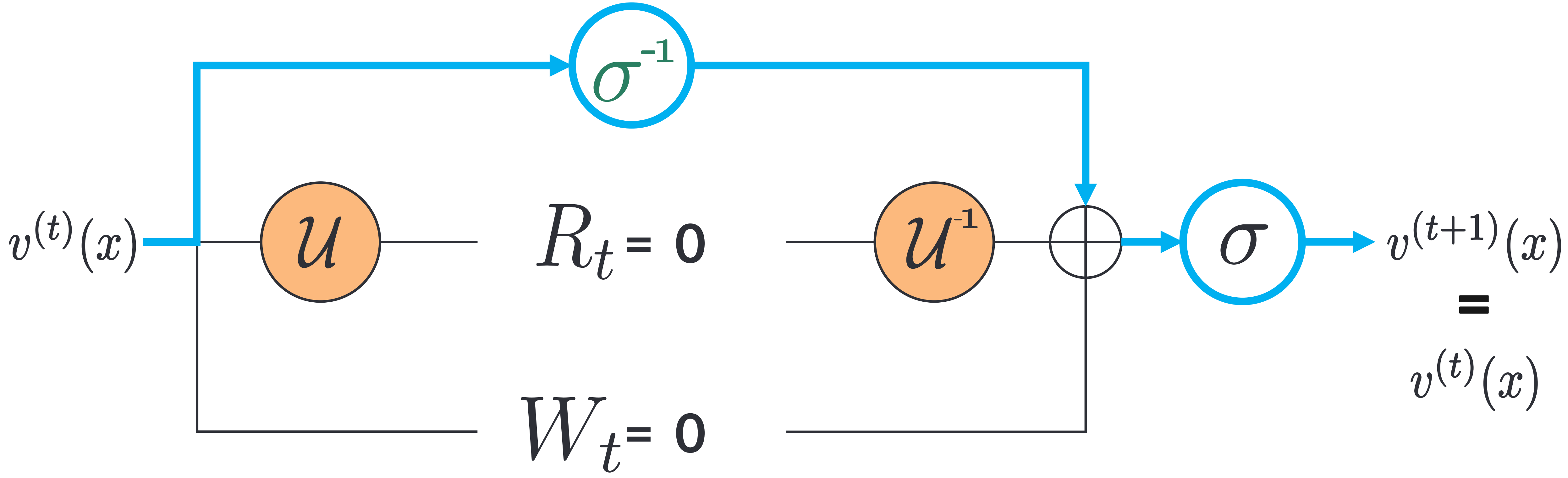
$\Longrightarrow$ Bregman Neural Operator layer



$$v^{(t+1)} = \operatorname*{argmin}_v f_t(\mathcal{U}(v), \mathcal{U}(v^{(t)})) + g_t(v, v^{(t)}) + D(v, v^{(t)}) + h(v)$$

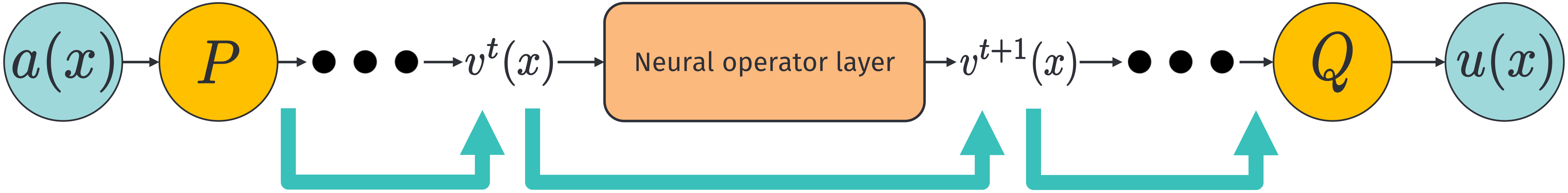
1: Frecon, J., et al. (2022). Bregman Neural Networks. ICML

Property of Bregman Neural Operators



When $\mathbf{R_t} = \mathbf{0}$ and $\mathbf{W_t} = \mathbf{0}$, the layer is equivalent to the **Identity Operator**

Bilevel Optimization



Upper-Level:

$$\min_{\{f_t, g_t\}_{t=0}^{T}} \ell(u^{obs}, u^{pred})$$

Lower-Level:

$$v^{(0)} = P(a)$$

$$\text{for } l = 0, 1, \ldots T - 1$$

$$v^{(t+1)} = \operatorname*{argmin}_{v} f_t(\mathcal{U}(v), \mathcal{U}(v^{(t)})) + g_t(v, v^{(t)}) + D(v, v^{(t)}) + h(v)$$

$$u(x) = Q(v^T)$$

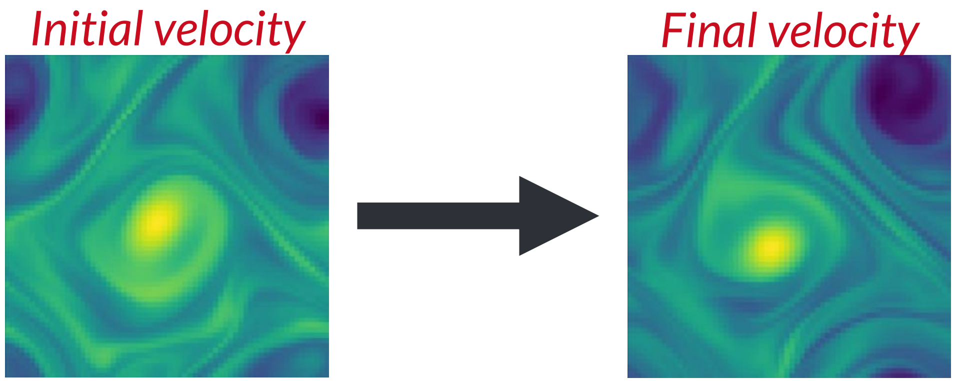
Experimental Setting

Models

- 2 Models
  - **Fourier Neural Operator (FNO)**
  - **Bregman FNO, layers initialized at 0**
- 32x32 Fourier modes
- 64 channels per Operator layer
- $\sigma =$ Softplus
- 1500 epochs

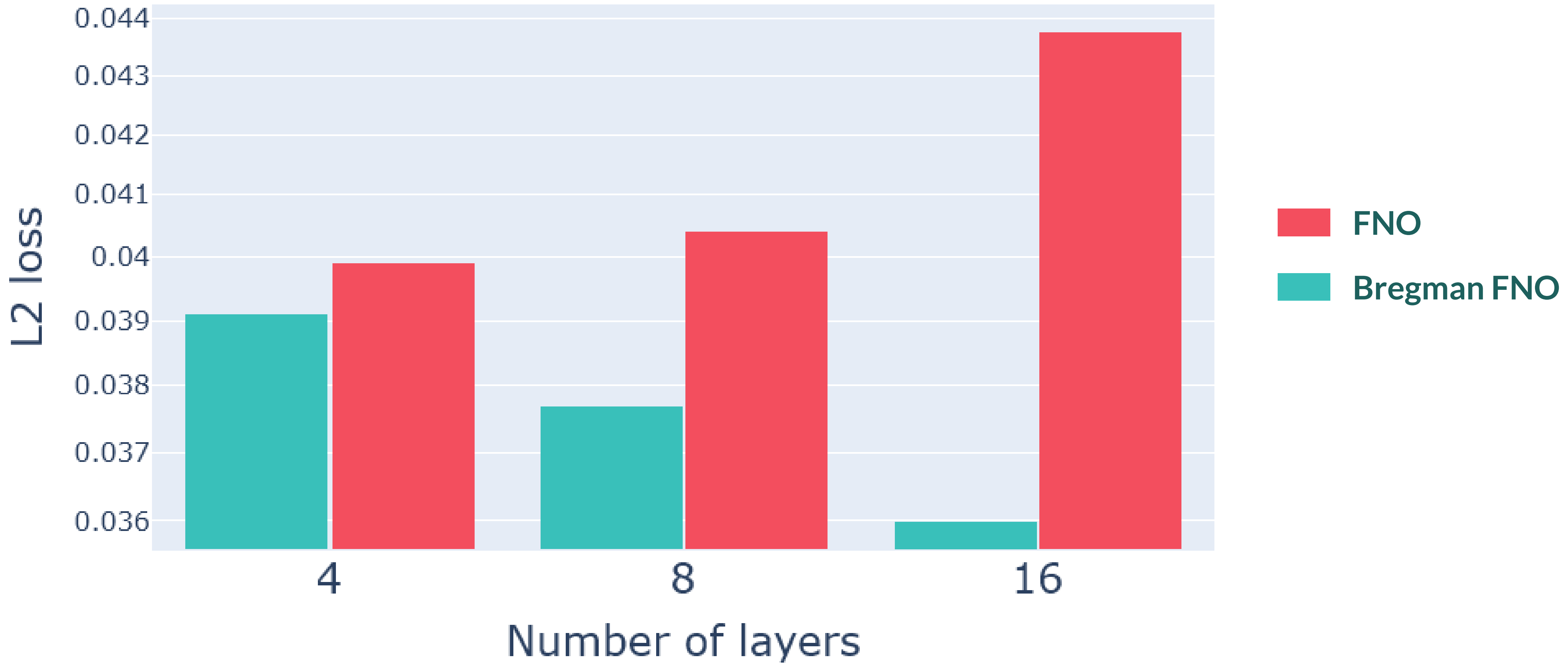
Dataset

- Navier-Stokes from PINO Github repository[1]
- temporal domain: $t \in \{0, 0.5\}$
- Resolution : 64x64
- Samples :
  - 2000 for training
  - 1000 for testing
  - 1000 for validation



1: https://github.com/neuraloperator/physics_informed

Numerical Experiments



Increasing performance with deeper models

Conclusion

Bilevel Optimisation Framework

$$\min_{\{f_t, g_t\}_{t=0}^{T}} \ell(u^{obs}, u^{pred})$$

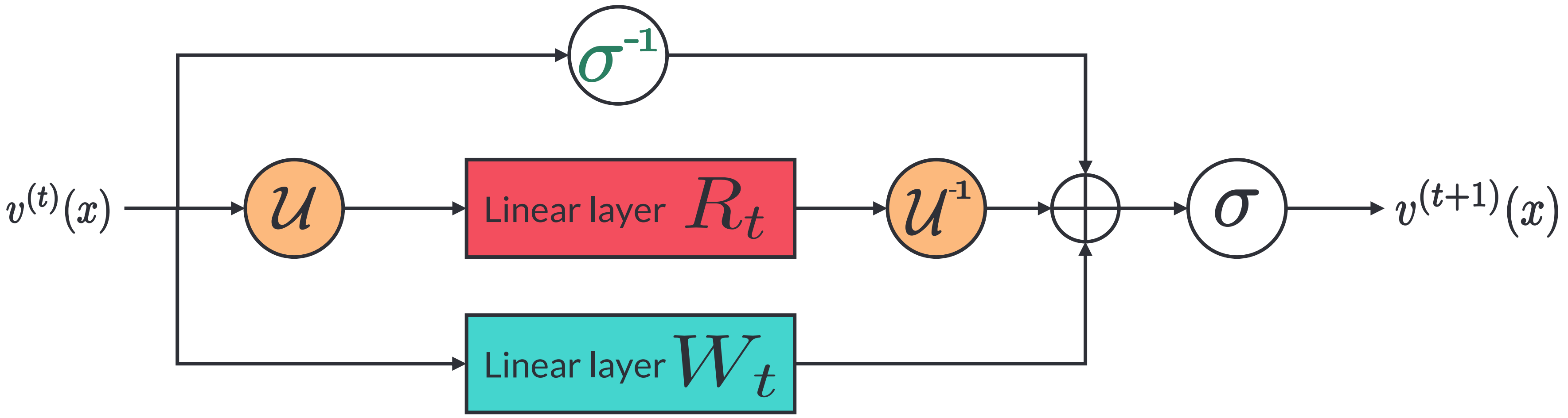
$$v^{(0)} = P(a)$$

for $l = 0, 1, \ldots T-1$

$$v^{(t+1)} = \operatorname*{argmin}_v f_t(\mathcal{U}(v), \mathcal{U}(v^{(t)})) + g_t(v, v^{(t)}) + D(v, v^{(t)}) + h(v)$$

$$u(x) = Q(v^T)$$

Bregman Neural Operators



Future work

Upper-Level:

$$\min_{\{f_t, g_t\}_{t=0}^{T}} \ell(u^{obs}, u^{pred}) \quad + \quad \ell_{Physics-Informed}$$

Lower-Level:

$$v^{(0)} = P(a)$$
for $l = 0, 1, \ldots T-1$
$$v^{(t+1)} = \operatorname*{argmin}_{v} f_t(\mathcal{U}(v), \mathcal{U}(v^{(t)})) + g_t(v, v^{(t)}) + D(v, v^{(t)}) + h(v)$$
$$u(x) = Q(v^T)$$

Future work

Upper-Level:

$$\min_{\{f_t, g_t\}_{t=0}^{T}} \ell(u^{obs}, u^{pred}) \quad + \quad \ell_{Physics-Informed}$$

Lower-Level:

$$v^{(0)} = P(a)$$
$$\text{for } l = 0, 1, \ldots T-1$$
$$\left\lfloor \; v^{(t+1)} = \operatorname*{argmin}_{v} f_t(\mathcal{U}(v), \mathcal{U}(v^{(t)})) + g_t(v, v^{(t)}) + D(\mathcal{U}(v), \mathcal{U}(v^{(t)})) + h(\mathcal{U}(v)) \right.$$
$$u(x) = Q(v^T)$$

Thank you !

Annex: Choice of couple (D,h)

Euclidean

$$\begin{cases} D &= D_{\frac{1}{2}\|\cdot\|^2} \\ h &= \phi - \frac{1}{2}\|\cdot\|^2 \\ \mathrm{prox}_h^D &\overset{1}{=} \sigma \end{cases}$$

$$v^{(t+1)} = \mathrm{argmin}_v(\ldots) + \phi(v) - \langle v^{(t)}, v\rangle$$

Bregman

$$\begin{cases} D &= D_{\phi} \\ h &= \imath_{\mathrm{dom}\phi} \\ \mathrm{prox}_h^D &= (\nabla\phi)^{-1} \overset{2}{=} \sigma \end{cases}$$

$$v^{(t+1)} = \mathrm{argmin}_{v\in\mathrm{dom}\phi}(\ldots) + \phi(v) - \langle \nabla\phi(v^{(t)}), v\rangle$$

The two optimization problems differ by a linear term (For *Bregman* $\nabla\phi(v^{(t)}) = \sigma^{-1}(v^{(t)})$)

Reminder: Bregman divergence $D_\phi(u,v) = \phi(u) - \phi(v) - \langle\nabla\phi(v), u - v\rangle$

→ In particular $D_{\frac{1}{2}\|\cdot\|^2}(u,v) = \frac{1}{2}\|u-v\|^2$

1: Combettes, P. L. and Pesquet, J.-C. (2020). Deep neural network structures solving variational inequalities. Set-Valued and Variational Analysis

2: Frecon, J., et al. (2022). Bregman Neural Networks. ICML

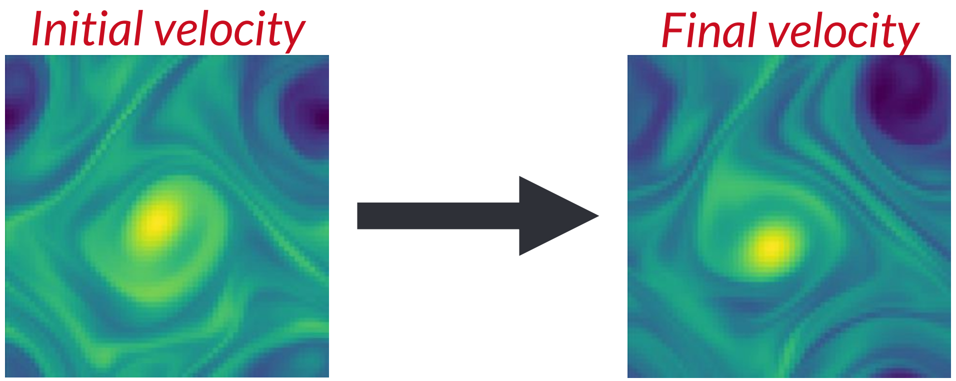
Experimental Setting

Models

- 2 Models
  - **Fourier Neural Operator (FNO)**
  - **Bregman FNO, layers initialized at 0**
- 32x32 Fourier modes
- 64 channels per Operator layer
- $\sigma =$ Softplus
- 1500 epochs

Dataset

- Navier-Stokes from PINO Github repository[1]
- Reynolds number: 500
- spatial domain: $x \in (0,2\pi)^2$
- temporal domain: $t \in \{0, 0.5\}$
- Resolution : 64x64
- Samples :
  - 2000 for training
  - 1000 for testing
  - 1000 for validation



1: *https://github.com/neuraloperator/physics_informed*